



PUBLIC (ОБЩЕДОСТУПНО)

Платформа SAP BusinessObjects Business Intelligence

Версия документа: 4.3 Support Package 4 – 2023-12-07

Руководство пользователя средства создания юниверсов

Содержание

1	История документа.	12
2	Общие сведения о средстве создания юниверсов.	13
2.1	Обзор.	13
2.2	Средство создания юниверсов и основные сведения о юниверсах.	13
	Что такое юниверс?.	14
	Какова роль юниверса?.	14
	Что составляет содержание юниверсов?.	14
	Окно юниверса.	16
	Корневой путь установки средства создания юниверсов.	17
2.3	Использование средства создания юниверсов для проектирования юниверсов.	18
	Как объекты генерируют программный код SQL?.	18
	Какие типы схем баз данных поддерживаются?.	19
	Как использовать юниверсы?.	19
2.4	Кто является дизайнером юниверса?.	20
	Требуемые навыки и знания.	20
	Каковы задачи дизайнера юниверса?.	21
2.5	Основные этапы создания юниверса.	21
2.6	Введение в процесс развертывания юниверса.	22
	Методология создания юниверса.	22
	Цикл разработки юниверса.	23
	Оптимизация юниверса планирует и поддерживает время.	25
2.7	Многоязычные юниверсы.	25
2.8	Определения языков и региональных стандартов.	26
2.9	Различные языковые стандарты.	27
2.10	Настройка языка продукта для пользовательского интерфейса средства создания юниверсов.	28
2.11	Использование многоязычных юниверсов.	28
2.12	Определение языкового стандарта замещения в связанных юниверсах.	28
2.13	Средство управления переводами.	29
2.14	Многоязычные данные.	29
2.15	Примеры для средства создания юниверсов.	29
	Демонстрационные базы данных.	30
	Демонстрационные юниверсы.	30
2.16	Использование юниверсов в средстве дизайна информации.	30
3	Выполнение основных операций.	32

3.1	Обзор.	32
3.2	Запуск средства создания юниверсов.	32
	Запуск средства создания юниверсов.	33
	Использование мастера Quick Design	33
3.3	Создание базового юниверса с помощью мастера нового юниверса.	34
	Зачем использовать мастер быстрого создания?	34
	Использование мастера быстрого дизайна.	34
	Внесение изменений в юниверс, созданный с помощью мастера нового юниверса.	42
3.4	Импорт юниверса.	42
	Импорт юниверса из репозитория.	42
	В чем разница между открытием и импортированием?	43
3.5	Открытие юниверса.	43
	Для открытия юниверса без импортирования.	43
3.6	Экспорт юниверса.	44
	Как расположены юниверсы в файловой системе репозитория?	44
	Экспортирование юниверса в репозиторий.	45
	В чем разница между экспортом и сохранением?	45
3.7	Сохранение юниверса.	46
	Имена файлов юниверсов в качестве идентификаторов.	46
	Сохранение юниверса.	46
	Сохранение определения юниверса в формате PDF.	47
3.8	Заккрытие юниверса.	48
3.9	Работа с несколькими дизайнерами.	48
	Блокировка юниверса.	48
	Номер редакции.	49
3.10	Пользовательский интерфейс средства создания юниверсов.	49
	Основные компоненты интерфейса пользователя.	49
	Пользовательский интерфейс средства создания юниверсов.	50
	Управление окнами	51
	Использование панелей инструментов.	51
	Выполнение действия или операции в средстве создания юниверсов.	52
3.11	Использование команд "Найти" и "Заменить".	54
	Использование команды "Найти".	54
	Использование быстрого поиска.	56
3.12	Организация отображения таблицы.	56
	Каким образом отображаются таблицы?	57
	Управление таблицами.	57
	Использование режима списка.	58
	Автоматическое упорядочивание таблиц.	59
	Изменение режима отображения таблицы.	59
3.13	Выбор параметров отображения схемы.	61

	Установка графических параметров для отображения панели структур.	62
	Просмотр значений таблиц и столбцов.	63
	Просмотр количества строк в таблицах базы данных.	66
3.14	Печать юниверса.	69
	Установка параметров печати.	70
4	Создание юниверса и настройка его параметров.	73
4.1	Что такое параметры юниверса?.	73
4.2	Создание нового юниверса.	74
	Создание нового юниверса из рабочей области.	74
4.3	Просмотр и ввод сводной информации.	76
4.4	Настройка параметров юниверса.	76
	Определение юниверса.	77
	Определение и редактирование соединений.	78
	Установка параметров сводки юниверса.	87
	Выбор стратегий.	88
	Определение средств управления ресурсами.	92
	Какие параметры системных ресурсов доступны?.	92
	Чтобы ввести информацию для управления ресурсом, выполните следующие действия.	93
	Ограничение времени выполнения запросов, создающих более одного SQL-оператора	93
	Определение ограничений SQL.	94
	Определение параметров для связанных юниверсов.	96
	Установка параметров генерации SQL.	96
	О параметрах генерации SQL.	99
	Изменение параметров динамической генерации SQL.	99
	Параметры SQL, задаваемые в пользовательском интерфейсе.	100
	Параметры SQL, задаваемые в PRM-файлах.	117
5	Создание схемы с помощью таблиц и объединений.	132
5.1	Общие сведения.	132
5.2	Что такое схема?.	132
	Дизайнер схемы является основным для успешного юниверса.	133
	Процесс разработки схемы и создания юниверса.	133
	Каковы стадии проектирования схемы?.	133
5.3	Вставка таблиц.	134
	Использование обозревателя таблиц.	134
	Упорядочивание таблиц на панели "Структура".	137
5.4	Использование производных таблиц.	138
	Добавление, редактирование и удаление производных таблиц.	138
5.5	Вложенные производные таблицы.	141

	Использование редактора "Производные таблицы".	141
	Как создать вложенную производную таблицу.	142
	Переименование вложенных производных таблиц.	142
5.6	Использование таблиц, имеющих столбцы ввода данных.	142
	Определение жестко запрограммированного списка значений.	143
	Определение списка значений для последующего их выбора или ввода пользователем	144
5.7	Определение объединения.	144
	Что такое объединение?.	145
	Почему объединения используются в схеме?.	145
	Объединение SQL.	145
	Что отсутствует в таблицах для объединения?.	146
	Объединение первичного и внешнего ключей.	146
	Понимание элементов объединения.	147
	Создание объединений.	148
	Свойства объединения.	153
	Редактирование объединения.	155
	Поддержка ANSI 92 для объединений в юниверсе.. . . .	159
	Удаление объединений.	163
5.8	Определение отдельных типов объединений.	163
	Создание равных объединений.	164
	Тета-объединения.	168
	Внешние объединения.	171
	Объединения быстрого доступа.	175
	Самоограниченные объединения.	176
5.9	Использование мощности множеств.	180
	Использование свойств связей в средстве создания юниверсов.	180
	Установка мощностей множеств вручную.	182
5.10	Проверка юниверса.	190
	Автоматическая Проверка целостности юниверса.	190
6	Устранение проблем объединений в схеме.	197
6.1	Обзор.	197
6.2	Что собой представляет проблема пути объединения?.	197
	Что собой представляет таблица соответствия.	198
	Что собой представляет таблица фактов.	198
	Какие типы путей объединения возвращают неверные результаты?.	198
	Определение и устранение проблем объединений.	199
6.3	Определение псевдонимов.	200
	Как использовать псевдонимы в схеме?.	200
	Создание псевдонимов.	201
6.4	Определение контекстов.	204

	Использование контекстов в схеме.	204
	Создание контекста.	205
	Редактирование контекста.	208
	Удаление контекста.	209
	Обновление контекстов.	209
	Пути объединения, которые предотвращают обнаружение контекста.	210
	Как контексты влияют на запросы?.	211
6.5	Устранение циклов.	215
	Что такое цикл?.	215
	Как цикл влияет на запросы?.	216
	Визуальное определение циклов.	223
	Автоматическое определение и устранение циклов.	224
	Функции средства проектирования для определения и устранения циклов.	224
	Примеры устранения циклов.	232
6.6	Устранение ловушек разногласия.	241
	Что такое ловушка разногласия?.	241
	Как ловушка разногласия завышает результаты?.	243
	Определение ловушки разногласия.	245
	Устранение ловушки разногласия.	245
6.7	Устранение ловушки раскрытия.	247
	Что такое ловушка раскрытия?.	247
	Как обнаружить ловушку раскрытия?.	250
	Устранение ловушки раскрытия.	250
6.8	Графическая демонстрация выявления проблем объединения.	254
	Вероятная ошибка разногласий.	254
	Вероятная ошибка раскрытия.	255
6.9	Проверка юниверса.	256
	Типы ошибок, определенные с помощью параметра "Проверить целостность".	257
	Проверка целостности юниверса вручную.	258
	Обновление структуры юниверса.	261
7	Создание юниверсов.	263
7.1	Обзор.	263
7.2	Введение в создание юниверса.	263
	Что такое объект?.	264
	Какие типы объектов используются в юниверсе?.	265
	Использование классов и объектов.	265
	Что такое класс?.	265
7.3	Использование области "Юниверс".	266
	Отображение классов и объектов или условий.	266
7.4	Основные операции над классами, объектами и условиями.	267
	Вырезать, копировать, вставить.	267

	Перемещение классов, объектов, или условий.	267
	Скрыть или отобразить классы, объекты и условия.	267
7.5	Определение классов.	268
	Создание класса.	268
	Свойства класса.	270
	Изменение класса.	271
	Использование подклассов.	271
7.6	Определение объектов.	272
	Создание объекта.	272
	Свойства объекта.	274
	Изменение объекта.	275
	Определение объекта.	275
	Свойства.	278
	Дополнительно.	279
	Определение поддержки индекса.	281
	Исходные сведения.	285
	Использование редактора SQL для определения объекта.	286
	Определение формата объекта.	289
	Просмотр таблицы, используемой в определении объекта.	290
	Определение измерения.	290
	Определение объекта-описания.	291
	Определение меры.	291
	Определение ограничений для объекта.	297
	Определение условных объектов.	302
	Использование самоограниченных объединений для применения ограничений.. . . .	308
	Применение ограничения путем логического вывода нескольких таблиц.	308
	Объединенные объекты.	311
7.7	Определение иерархий.	312
	Что такое многомерный анализ?.	313
	Определение иерархий.	313
	Установка иерархий.	314
7.8	Использование каскадных списков значений для иерархий.	317
	Создание каскадного списка значений.	318
7.9	Использование списков значений.	321
	Как использовать список значений?.	321
	Определение способа использования списка значений по отношению к объекту.	323
	Свойства и параметры списка значений.	323
	Редактирование списка значений.	327
	Экспорт списка значений.	331
	Обновление значений в списке значений.	334
	Использование данных в файле персональных данных.	334

	Администрирование списков значений в юниверсе.	336
	Оптимизация и настройка файлов списков значений.	337
7.10	Связывание юниверсов.	338
	Что представляют собой связанные юниверсы?.	338
	Различные способы связывания юниверсов.	340
	Преимущества связывания юниверсов.	341
	Требования по связыванию юниверсов.	342
	Ограничения при связывании юниверсов.	342
	Создание связи между двумя юниверсами.	342
	Редактирование производного юниверса.	346
	Удаление связи.	346
	Перемещение главного юниверса.	347
	Производные юниверсы и списки значений.	347
	Представление объектов в порядке их следования в главном юниверсе.	348
7.11	Включение одного юниверса в другой.	348
	Копирование главного юниверса в производный юниверс.	348
7.12	Создание юниверсов хранимых процедур.	349
	Хранимые процедуры в юниверсах Java bean.	350
	Создание юниверса на основе хранимых процедур.	350
7.13	Проверка юниверса.	354
	Проверка объектов в панели запросов.	354
	Проверка целостности юниверса.	354
	Проверка юниверса с помощью Web Intelligence.	355
8	Оптимизация юниверсов.	356
8.1	Обзор.	356
8.2	Использование таблиц агрегирования.	356
	Что такое поддержка агрегирования?.	356
	Применение поддержки агрегирования к хранилищам данных.	357
	Установка поддержки агрегирования.	357
	Создание объектов.	358
	Определение всех комбинаций объектов агрегирования.	359
	Расположение объектов в порядке агрегирования.	359
	Определение объектов агрегирования с помощью функции @Aggregate_Aware.	359
	Указание несовместимых объектов.	362
	Указание несовместимых объектов.	365
	Устранение циклов с помощью таблиц агрегирования.	368
	Тестирование поддержки агрегирования.	370
8.3	Использование функций @Function в SQL-коде для объекта.	370
	Вставка функции @Function в объект.	371
	Функция @Aggregate_Aware.	373

	@Prompt.	374
	@Select.	389
	@Variable.	390
	Оператор Where.	397
8.4	Использование внешних стратегий для настройки процесса создания юниверса.	399
	Перенос внешних стратегий в средство создания юниверсов.	400
	Общие сведения о внешних стратегиях.	400
	Что такое внешняя стратегия?.	401
	Создание справочного текста для внешних стратегий.	402
	Проверка на наличие описания файла внешней стратегии.	404
	Использование примера внешних стратегий.	405
	Какую структуру имеет файл стратегии (STG)?.	405
	Форматы вывода у стратегий.	407
	Создание внешней стратегии.	410
	Создание текстового файла для данных.	412
	Применение внешних стратегий в средстве создания юниверсов.	412
8.5	Использование аналитических функций.	413
	Что такое аналитические функции?.	414
	Какие преимущества дает использование аналитических функций?.	414
	Какие семейства аналитических функций поддерживаются?.	414
	Использование аналитических функций в средстве создания юниверсов.	415
	IBM DB2 UDB и Oracle.	416
	RedBrick (функции RSQL).	420
	Teradata (функции OLAP).	423
	Автоматическая вставка синтаксической конструкции в инструкции Select.	425
8.6	Использование функции добавления префиксов SQL.	426
	Добавление префикса перед SQL-инструкцией с помощью параметра юниверса BEGIN_SQL.	427
8.7	Оптимизация параметра выборки массива.	427
	Изменение параметра выборки массива.	428
8.8	Присвоение размера таблицы.	428
	Изменение файла PRM для присвоения размеров таблицы.	428
8.9	Изменение количества возвращенных строк для таблицы.	429
8.10	Использование объединений быстрого доступа.	429
9	Работа с юниверсами OLAP.	430
9.1	Юниверсы OLAP.	430
	Что такое юниверс OLAP?.	430
	Источники данных OLAP, которые можно использовать для создания юниверса.	431
9.2	Определение соединений для источника данных OLAP.	437
	Соединения с источниками данных OLAP.	437
	Запуск мастера нового соединения.	438

	Выбор программного обеспечения среднего яруса базы данных для соединения OLAP.	439
	Параметры входа для реляционных соединений – UDT.	440
	Параметры входа для соединений SAP BW OLAP.	442
	Параметры входа для соединений MSAS OLAP.	443
	Определение параметров входа в систему для соединений Essbase.	444
	Выбор куба или запроса источника для соединения OLAP.	445
	Определение параметров конфигурации для соединений OLAP.	445
	Определение пользовательских параметров для соединений Essbase.	447
9.3	Редактирование юниверсов OLAP.	447
	Создание юниверсов OLAP с дополнительными параметрами.	447
	Определение параметров OLAP для юниверса OLAP.	448
	Определение объектов в юниверсах OLAP.	449
	Функции средства создания юниверсов, поддерживаемые для юниверсов OLAP.	450
	Функция проекции "Делегированная база данных".	450
	Настройка делегированных мер для юниверсов OLAP.	452
	Установка проекции агрегирования для меры.	453
	Вычисляемые показатели в юниверсах OLAP.	454
	Функции MDX для запросов к кубу.	456
	XML-синтаксис для инструкций фильтра и WHERE.	457
	Предопределенные условия в юниверсах OLAP.	458
	Дополнительные подсказки в юниверсах OLAP.	463
	Повышение производительности определенных запросов для юниверсов SAP BW.	463
9.4	Управление жизненным циклом юниверса OLAP.	464
	Управление жизненным циклом юниверса OLAP.	464
	Обзор: взаимосвязь между состоянием объекта юниверса и состоянием объекта OLAP	465
	Обновление юниверса OLAP.	467
	Регенерация уровня 00 для юниверсов OLAP.	470
	Переименование уровня "L00" в уровень "BCE".	470
	Замена префиксов уровня юниверса OLAP.	471
	Синхронизация юниверса и куба OLAP.	471
	Управление измерениями при обновлении юниверсов OLAP.	472
	Управление иерархиями и характеристиками при обновлении юниверсов OLAP.	476
	Управление уровнями при обновлении юниверсов OLAP.	483
	Управление переменными SAP при обновлении юниверсов OLAP.	486
	Управление контрольными цифрами и мерами при обновлении юниверсов OLAP.	490
	Управление контрольными датами SAP при обновлении юниверсов OLAP.	493
9.5	Сопоставление различных кубов OLAP с юниверсами.	495
	Сопоставление и использование объектов SAP BW в юниверсе.	495
	Сопоставление кубов Essbase и компонентов юниверса.	504

	Сопоставление кубов MSAS с компонентами юниверса	505
10	Развертывание юниверсов.	507
10.1	Обзор.	507
10.2	Как выполняется развертывание юниверсов?.	507
	Определение юниверса в репозитории.	507
10.3	Предоставление доступа к юниверсу для всех пользователей.	508
10.4	Установка ограничений доступа к юниверсу.	508
	Что представляет собой ограничение?.	509
	Какие ограничения можно применять в юниверсе?.	509
	Как управлять ограничениями доступа.	510
	Создание ограничения.	511
	Применение ограничений доступа к юниверсу.	513
	Добавление группы пользователей в список доступных пользователей юниверса.	514
	Установка приоритета группы ограничений.	515
	Просмотр ограничений безопасности пользователей и групп пользователей.	517
10.5	Управление пользователями и регистрационными именами.	518
	Управление регистрационными именами.	518
	Управление паролями.	519
11	Использование образцов материалов.	520
11.1	Обзор.	520
11.2	База данных Клуба.	520
	Структура таблиц.	520

1 История документа

В следующей таблице описываются наиболее важные изменения документа.

Версия	Дата	Описание
Средство создания юниверсов SAP Business Objects 4.3	Июнь 2020 г.	Первоначальный выпуск
Средство создания юниверсов	Сентябрь 2021 г.	Информация об устаревании продукта обновлена в документации

2 Общие сведения о средстве создания юниверсов

2.1 Обзор

В этой главе приводятся общие сведения о *средстве создания юниверсов*, предназначенном для построения юниверсов. Приводится описание юниверсов, их содержимое, способ создания и роль юниверсов в бизнес-среде.

Описан обычный цикл работы среды юниверса, а также описаны наилучшие методы создания. В этой версии также приводится описание демонстрации баз данных и юниверсов.

В этой главе приводятся общие сведения о средстве создания юниверсов, структуре процесса разработки, а также различных языковых возможностях для юниверсов. Рассматриваются следующие темы:

📘 Примечание

Этот продукт устарел и больше не поддерживается и не поставляется после версии 4.3. При любых новых разработках настоятельно рекомендуется использовать *средство дизайна информации*; см. статью базы знаний SAP [3028832](https://support.sap.com/en/other/3028832.html) на портале SAP Support Portal.

Связанные сведения

[Средство создания юниверсов и основные сведения о юниверсах \[страница 13\]](#)

[Использование средства создания юниверсов для проектирования юниверсов \[страница 18\]](#)

[Кто является дизайнером юниверса? \[страница 20\]](#)

[Введение в процесс развертывания юниверса \[страница 22\]](#)

[Многоязычные юниверсы \[страница 25\]](#)

2.2 Средство создания юниверсов и основные сведения о юниверсах

Средство создания юниверсов BusinessObjects позволяет создавать юниверсы для пользователей, использующих такие клиенты BI, как Web Intelligence.

2.2.1 Что такое юниверс?

Юниверс – это файл, который содержит:

- Параметры соединения для одного или нескольких компонентов доступа к базе данных.
- Структурами SQL называются объекты, которые отображают действительные структуры SQL в базе данных, например столбцы, таблицы и функции баз данных. Объекты сгруппированы в классы. Объекты и классы доступны для просмотра пользователей Web Intelligence.
- Схема таблиц и объединений используется в базе данных. Объекты строятся из структур баз данных, которые включены в схему. Схема доступна только пользователям средства создания юниверсов. Она не отображается для пользователей, использующих клиентские инструменты BI, такие как Web Intelligence.

Пользователи Web Intelligence подключаются к юниверсу для выполнения запросов в базе данных. Они могут анализировать данные и создавать отчеты с помощью объектов в юниверсе, не видя основных структур данных в базе данных и ничего о них не зная.

2.2.2 Какова роль юниверса?

Роль юниверса заключается в предоставлении простоты использования и понимания интерфейса пользователям, не имеющим специальных технических знаний Web Intelligence для выполнения запросов в базе данных для создания отчетов и выполнения анализа данных.

Дизайнер юниверса может использовать средство создания юниверсов для создания объектов, представляющих структуры баз данных, например столбцы и функции баз данных, к которым пользователи выполняют доступ и запросы для получения информации, необходимой в соответствии с их бизнес-требованиями.

Объекты, создаваемые в юниверсе, должны соответствовать бизнес-среде и словарю конечного пользователя. Их роль заключается в предоставлении внешнего интерфейса, сконцентрированного на бизнесе для структур SQL в базе данных.

2.2.3 Что составляет содержание юниверсов?

Юниверс содержит следующие структуры:

- Классы
- Объекты

2.2.3.1 Классы

Класс – это логически сгруппированные объекты в юниверсе. Он представляет собой категорию объектов. Имя класса отображает категорию содержащихся в нем объектов. Класс может содержать подклассы, расположенные по порядку.

2.2.3.2 Объекты

Объект – именованный компонент, показывающий путь к данным или ошибкам данных в базе данных. Имя объекта необходимо взять из делового словаря целевой группы пользователей. Например, объектами менеджера по продукции в юниверсе могут быть "Продукт", "Жизненный цикл продукта" или "Дата выпуска". Юниверс финансового аналитика может содержать следующие объекты: "Размер прибыли" или "Окупаемость".

2.2.3.3 Типы объектов

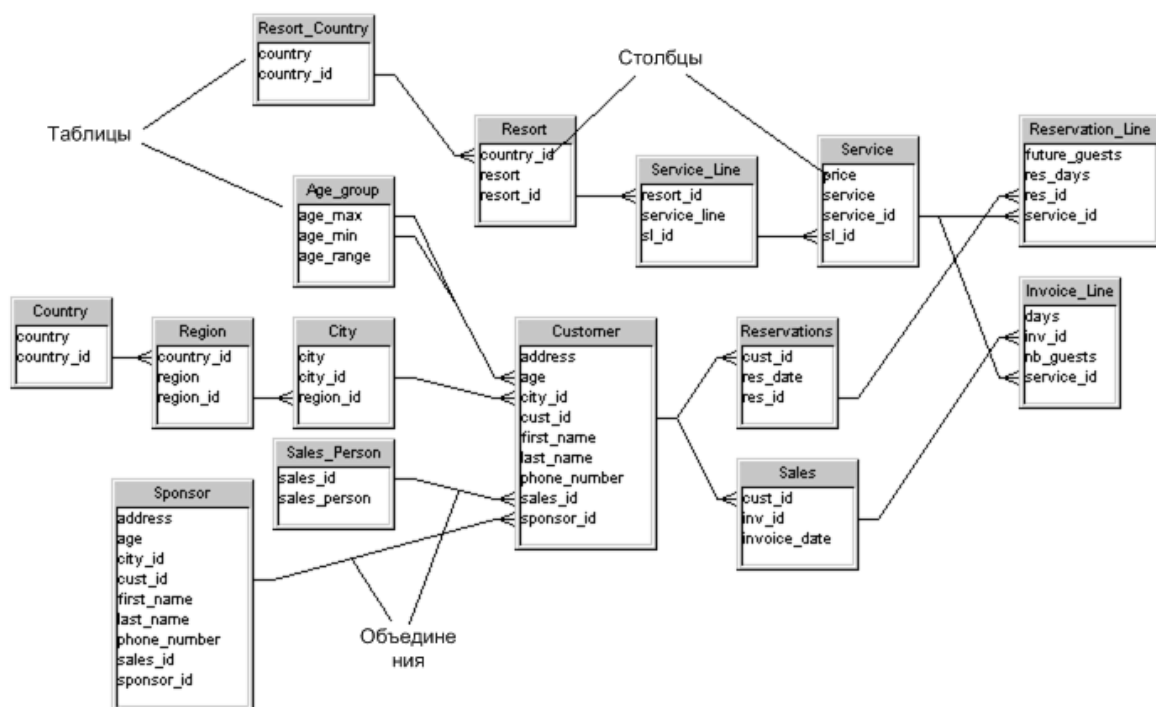
В средстве создания юниверсов объекты классифицируются по трем признакам: измерение, сведения и мера.

Тип объекта	Описание
Измерение	Параметры анализа. К измерениям обычно относятся расположенные по порядку данные, такие, как: географические данные, название продукта или временные данные. Пример. Фамилия и название города
Атрибут	Содержит описание измерения, но не является предметом анализа. Пример. Телефонный номер
Мера	Отражают числовые значения, которые используются для определения объекта измерения. Например, доходы с продаж

2.2.3.4 Структуры объединения объектов SQL отображаются в виде схемы

Пользователи программы Web Intelligence видят в юниверсах объединения SQL структуры, внесенные дизайнерами в схему базы данных. Дизайнер юниверса создает данную схему на основе таблиц и объединений, необходимых для отображения данных, нужных пользователям для анализа или создания отчета.

Схема – это часть файла юниверса, которая видна и доступна только в средстве создания юниверсов. Дизайнер создает схему в [Панели структуры Окна юниверса](#). Схема, показанная ниже – образец юниверса Beach.unv.

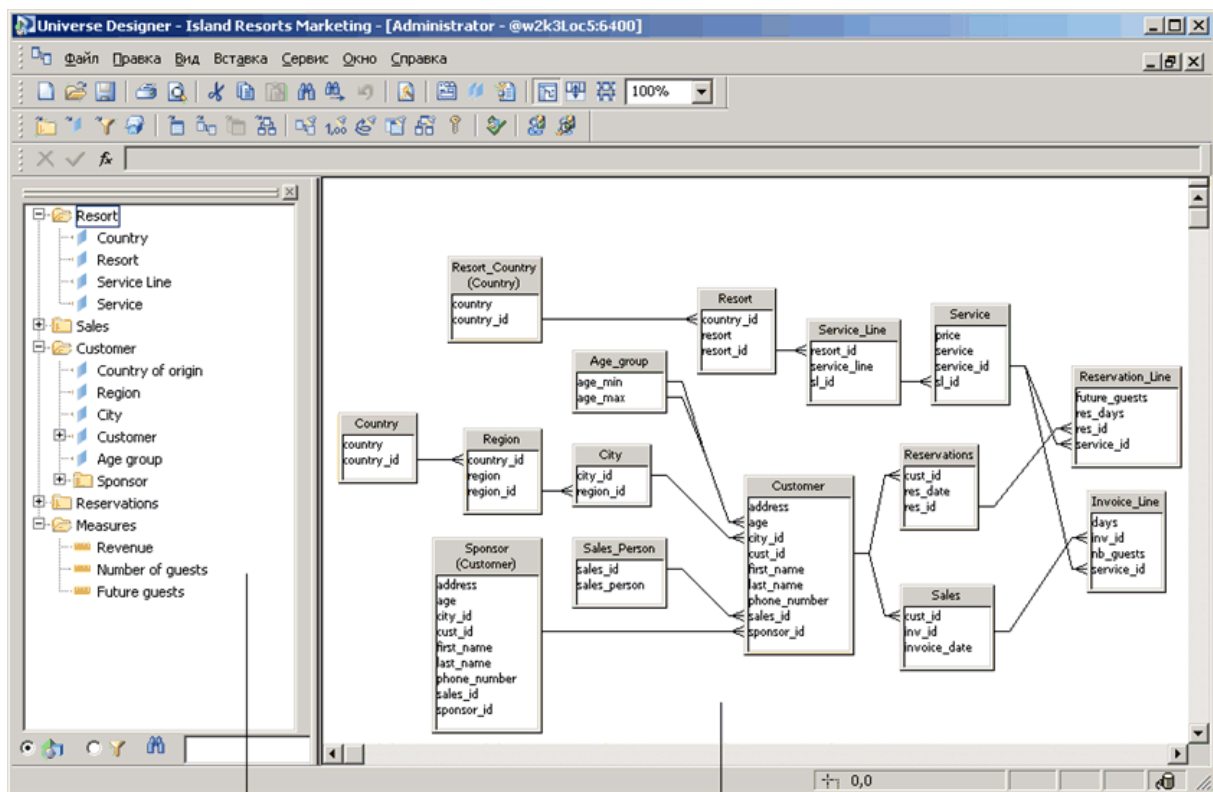


2.2.3.5 Как представлены объекты в юниверсе?

Объекты представляются в виде узлов в представлении *Древовидный проводник* на *панели юниверсов*. Дизайнер использует объекты меню для создания, удаления, копирования, просмотра и перемещения классов и объектов.

2.2.4 Окно юниверса

Ниже показано *окно юниверса* в средстве создания юниверсов. В нем содержится панель *Юниверс* (также отображаемая в приложении Web Intelligence) и панель *Структура* (отображаемая только в средстве создания юниверсов).



Панель
юниверсов

Панель структуры

2.2.5 Корневой путь установки средства создания юниверсов

В данном руководстве переменная \$INSTALLDIR – это корневой путь установки для файлов доступа к данным, используемых средством создания юниверсов и приложением Web Intelligence. Ниже показан путь установки Business Objects в подкаталог операционной системы, содержащий исполняемый файл средства создания юниверсов и драйверы доступа к данным.

В операционной системе Windows, \$INSTALLDIR = \\...\\Business Objects\\BusinessObjects Enterprise 12.0\\win32_x86.

Например, C:\\Program Files\\Business Objects\\BusinessObjects Enterprise 12.0\\win32_x86.

2.3 Использование средства создания универсов для проектирования универсов

В состав средства создания универсов входит мастер создания соединения с промежуточным ПО доступа к базе данных. С помощью этого средства можно создать несколько соединений, но для каждого универса может быть определено только одно соединение. Это соединение с базой данных сохраняется с универсом.

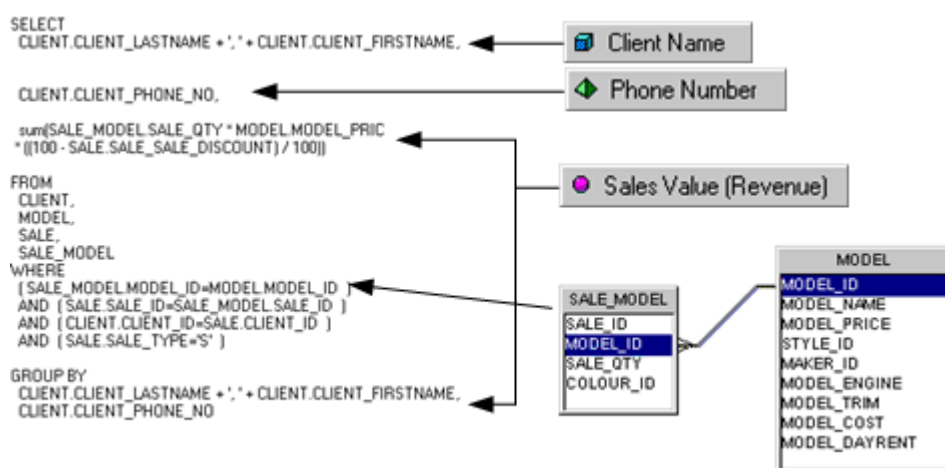
Средство создания универсов предоставляет графический интерфейс, с помощью которого можно выбрать и просмотреть таблицы в базе данных. Таблицы базы данных представлены в виде названий таблиц в схеме диаграммы. Интерфейс можно использовать для управления таблицами, создания объединений, связывающих таблицы, создания таблиц под псевдонимом, контекстов и циклов решения в схеме. Пользователи Web Intelligence не могут видеть эту схему.

В средстве создания универсов используется представление проводника объектов, *древовидное меню проводника*. Меню *Tree explorer* используется для создания объектов, показывающих путь к столбца и структурам SQL, представленным в схеме просмотра. Пользователи Web Intelligence управляют объектами для выполнения запросов в данных.

С помощью средства создания универсов можно распространять универсы путем их импорта в репозиторий CMS и экспорта из него.

2.3.1 Как объекты генерируют программный код SQL?

Пользователи Web Intelligence создают запросы путем перенесения объектов в рабочую область *Запрос*. Определение каждого объекта вычисляет инструкцию SELECT. При выполнении запроса, инструкция SELECT и необязательное утверждение WHERE для всех объектов запускаются по отношению к целевой базе данных.

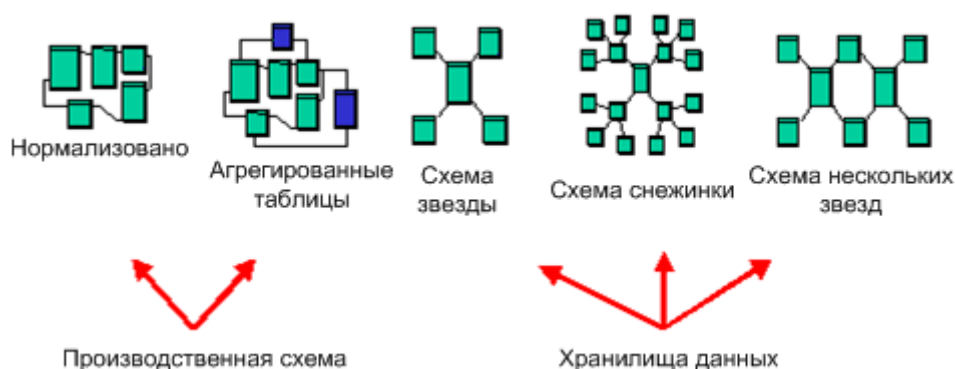


Когда пользователь включает в рабочую область *Запрос* объекты измерений и (или) сведений с объектом меры, утверждение GROUP BY, содержащее данные этих объектов измерений и сведений, автоматически добавляется к инструкции SELECT.

Таблицы, включаемые в утверждение FROM, и объединения в утверждении WHERE выводятся из схемы таблицы, созданной на панели [Структура](#).

2.3.2 Какие типы схем баз данных поддерживаются?

Средство создания юниверсов поддерживает большинство схем баз данных, включая показанные ниже. Перед использованием средства создания юниверсов не требуется переопределять или оптимизировать исходную базу данных.



2.3.3 Как использовать юниверсы?

Юниверсы используются пользователями Web Intelligence. Юниверсы хранятся в репозитории Центральной системы управления. Конечный пользователь подключается к юниверсам при помощи веб-обозревателя.

Соединение с базой данных определяется юниверсом и, таким образом, соединяясь с юниверсом конечный пользователь автоматически получает доступ к данным. Доступ к данным по очереди ограничивается объектами, доступными для юниверса. Данные объекты были созданы дизайнером юниверсов на основе профиля потребностей пользователя для целевой группы.

2.3.3.1 Предоставление необходимых данных

Юниверсы могут предоставлять необходимые данные конкретных приложений, систем или групп пользователей. Например, юниверс может содержать объекты, представляющие необходимые данные отделов маркетинга или бухгалтерии компании.

Юниверс может также предоставлять необходимые данные секции отдела или набора для организации процедур, таких, как платежная ведомость или система инвентаризации.

Примерами классов может служить информация о сотрудниках, информация о явке и информация об отделах.

2.3.3.2 Универсы и схемы баз данных

Схемы баз данных используются для создания трех универсов: ПЕРСОНАЛ, ТОВАРНЫЕ ЗАПАСЫ и ПРОДАЖИ. В каждом универсе содержатся классы и объекты. Каждый объект является отображением части структуры базы данных.

2.3.3.3 Кто пользуется универсами?

Пользователи Web Intelligence используют универсы для анализа и создания отчетов. В универсах можно создать классы и отчеты, относящиеся к их сфере деятельности.

2.4 Кто является дизайнером универса?

Для создания универсов в приложении Universe Designer используется средство создания универсов. Для дизайнера универса нет стандартного профиля. Человек, назначенный в компании дизайнером универса, может быть администратором базы данных, менеджером приложений или разработчиком, менеджером проекта или создателем отчета, обладающим достаточными техническими навыками для создания универсов других пользователей.

В компании может быть несколько дизайнеров универсов. Количество дизайнеров универса зависит от требований к данным компании. Например, один дизайнер универса может быть назначен для каждого приложения, проекта, отдела или функциональной области.

Если несколько человек создают универсы, важно определить набор правил или рекомендаций для соблюдения терминологии, чтобы объекты были представлены в точном соответствии друг другу.

2.4.1 Требуемые навыки и знания

Дизайнер универса должен обладать следующими навыками и уровнем технических знаний.

Навык или знание	Описание
Возможность анализировать потребности пользователя	Юниверсы создаются в соответствии с потребностями пользователя для данных. Дизайнер юниверса должен иметь навыки ведения анализа потребностей пользователя для создания классов и объектов, соответствующих словарю пользователя и развития юниверсов для удовлетворения потребностей сообщества пользователя. Сюда входит создание отчета и результаты запроса, которые подходят для анализа
Знание базы данных	Дизайнер юниверса должен иметь хорошие рабочие знания системы управления базы данных компании (СУБД), как разворачиваются базы данных, структуру логической базы данных и тип данных, хранимых в базах данных компании
Язык структурированных запросов (SQL)	Необходимы рабочие знания SQL

2.4.2 Каковы задачи дизайнера юниверса?

Дизайнер юниверса обычно несет ответственность за следующие задачи.

- Ведение анализа потребностей пользователя
- Дизайн и создание юниверса
- Распространение юниверса
- Поддержка юниверса

2.5 Основные этапы создания юниверса

Юниверс – это часть файла UNV, с помощью которого конечные пользователи средств отчетности создают отчеты. Это только часть файла UNV, видимая конечным пользователям. При создании юниверса следует выполнить следующие действия.

1. Создайте пустой файл юниверса и введите данные соединения и параметры.
2. Создайте схему таблиц баз данных, которые нужно использовать, и определите объединения между таблицами. Схема будет невидимой конечным пользователям, которые создают отчеты с помощью средств отчетности.
3. Разрешите в схеме все проблемы, связанные с объединениями.
4. Создайте классы и объекты, из которых будет состоять юниверс. Юниверс используется конечными пользователями, которые создают отчеты с помощью средств отчетности.

Связанные сведения

[Выполнение основных операций \[страница 32\]](#)

[Создание юниверса и настройка его параметров \[страница 73\]](#)

[Создание схемы с помощью таблиц и объединений \[страница 132\]](#)

[Устранение проблем объединений в схеме \[страница 197\]](#)

[Создание юниверсов \[страница 263\]](#)

[Оптимизация юниверсов \[страница 356\]](#)

2.6 Введение в процесс развертывания юниверса

В следующих разделах приведен обзор создания юниверса вручную и дано описание соответствующего создания юниверса для цикла развития обычного юниверса.

2.6.1 Методология создания юниверса

Методология дизайна юниверса, описанная в этом руководстве, состоит из одного этапа планирования и трех этапов реализации.

- Анализ бизнес-проблемы и планирование решения юниверса
- Разработка схемы
- Построение юниверса
- Распределение юниверса пользователям

Каждый этап реализации основан на допущении, завершеном на начальном этапе планирования. Этап планирования может быть выполнен без использования средства создания юниверсов и является решающим этапом, определяющим успешность создания юниверса. Плохо спланированный юниверс, не основанный на изучении потребностей пользователя, будет сложно разработать, реализовать, поддержать и он окажется бесполезным для целевых пользователей.

Каждая из этих фаз описана следующим образом.

2.6.1.1 Планирование юниверса до начала работы со средством создания юниверсов

Перед началом первого этапа необходимо потратить восемьдесят процентов времени, предназначенного для проекта создания юниверса, планируя юниверс. Необходимо придерживаться следующих пунктов.

- Анализировать анализ данных и требования в отчете целевой аудитории юниверса. Структуры, используемые для создания схемы, должны быть основаны на четко определенных потребностях пользователя для доступа к данным в таблицах и столбцах.

- Перед началом работы со средством создания юниверсов следует иметь четкое представление об объектах, которые необходимо создать. Не создавайте объекты при просмотре столбцов, доступных в базе данных, а указывайте столбцы, которые соответствуют обозначенному объекту из анализа потребностей пользователя.

2.6.1.2 Разработка схемы

Вы создаете схему для основной структуры базы данных юниверса. Эта схема включает таблицы и столбцы целевой базы данных и объединения, которыми они связаны. Возможно понадобится устранить проблемы объединения, например циклы, ловушки разногласия, ловушки раскрытия, которые могут произойти в структуре с использованием псевдонимов и контекстов. Протестируйте целостность общей структуры. В данном руководстве, разработка этапа схемы описана в главах [Создание схемы с помощью таблиц и объединений \[страница 132\]](#) и [Устранение проблем объединений в схеме \[страница 197\]](#).

2.6.1.3 Построение юниверса

Создайте объекты, которые выводят инструкции SELECT на основе компонентов схемы. Организуйте эти объекты в классы. Это объекты, которые определены из анализа потребностей пользователей. Можно создавать многие типы объектов для расширения возможностей отчета пользователя, многомерного анализа и оптимизировать производительность запроса.

Протестируйте целостность структуры юниверса. Также необходимо осуществить тестирования, выполнив отчеты в Web Intelligence.

Этап построения описан в главе [Создание юниверсов \[страница 263\]](#).

2.6.1.4 Распределение юниверса

Можно распределять юниверсы пользователям для тестирования и, со временем, для производства, экспортируя их в репозиторий центральной системы управления (CMS). Этот этап описан в главе [Развертывание юниверсов \[страница 507\]](#).

2.6.2 Цикл разработки юниверса

Разработка юниверса – это циклический процесс, который включает фазы планирования, проектирования, построения, внедрения и эксплуатации. Для проектирования и построения юниверсов используется средство создания юниверсов, однако эффективность любого юниверса напрямую зависит от того, насколько согласованы друг с другом другие этапы цикла разработки.

В этом разделе представлен обзор методологии проектирования юниверса, которую можно использовать для планирования и реализации проекта по разработке юниверса.

Ниже в таблице в общих чертах описан типичный цикл разработки юниверса.

Фаза разработки	Описание
Подготовка	<ul style="list-style-type: none">• Определение источника целевых данных и ознакомление с его структурой.• Ознакомление с данными, содержащимися в каждой таблице каждой из целевых баз данных.• Построение схемы объединений.• Оценка объема обрабатываемых данных (мощности множества).• Определение возможных направлений дальнейшей работы.
Анализ	<ul style="list-style-type: none">• Установление численности пользователей и структуры их распределения, например, разбивается ли группа пользователей по отделам или по задачам.• Определение информации, необходимой для пользователей.• Определение стандартных отчетов, которые им требуются.• Ознакомление с профессиональной терминологией для ее осмысленного использования.
Планирование	Определение стратегии проекта. Например, оценка количества юниверсов, которые необходимо создать, и решение вопроса о том, какие из них будут иметь возможность установления связи и на каком уровне.
Реализация	<ul style="list-style-type: none">• Построение юниверса с помощью соответствующего средства. В этом руководстве освещается эта часть цикла разработки юниверса, фактическое использование средства создания юниверсов.• Регулярная проверка правильности и достоверности создаваемого кода SQL во время процесса построения юниверса.
Проверка	Формирование небольшой группы квалифицированных пользователей Web Intelligence, которые имеют некоторое представление о той информации, которую они ожидают получить от юниверса. Предложение пользователям произвести всесторонние испытания, моделирующие реальное использование юниверсов.
Развертывание	Внедрение юниверса путем его экспорта в репозиторий центральной управляющей системы (Central Management System, CMS), где конечные пользователи смогут получать к нему доступ.

Фаза разработки	Описание
Развитие	Обновление и эксплуатация юниверса по мере изменения источников данных и роста требований пользователей.

📘 Примечание

При проектировании юниверса всегда следует учитывать в первую очередь требования пользователей, но НЕ структуру источников данных.

2.6.3 Оптимизация юниверса планирует и поддерживает время

Анализ требований пользователя и дизайнера являются наиболее важными этапами в процессе. Пользователи должны быть включены в развивающий процесс, если юниверсу необходимо выполнить требования к деловому языку, который используется для имени объекта и к данным которые могут быть доступными.

Реализация пройдет быстро и легко если первые три этапа выполнены правильно.

Можно потратить до 80% назначенного времени для развития юниверса на первые три этапа:

- Подготовка
- Анализ
- Планирование

Если потратить время на закладку фундамента для юниверса, то 20% времени расходуется на фактическое использование средства создания юниверсов для построения юниверса, и это будет значительно эффективнее, чем проектирование без необходимых предварительных этапов планирования и анализа.

2.7 Многоязычные юниверсы

Одной из основных функций SAP BusinessObjects Enterprise XI4 является способность создавать многоязычные метаданные и отчеты в одном и том же юниверсе. Данная функция позволяет руководствоваться одношаговым решением многоязычного отчета, зависящего от региональных стандартов и поддерживаемого единой моделью юниверса метаданных, и обеспечивает полную поддержку кодировки. Отчеты создаются в одном и том же юниверсе и отображаются на различных языках, заданных пользователем.

Метаданные юниверса, которые могут быть переведены:

- Имя юниверса
- Описание юниверса

- Имена классов
- Имена объектов
- Имена, описания и форматы объектов
- Имена пользовательских иерархий
- Вопросы столбцов ввода и подсказок

📌 Примечание

Только те подсказки, которые определены в метаданных юниверса, могут быть переведены. Подсказки, определенные с использованием функции @Prompt, нельзя переводить.

Юниверс может содержать переводы в нескольких языковых стандартах. При создании отчета на основе юниверса метаданные отображаются в языковом стандарте, зависящем от предпочитаемого пользователем языкового стандарта для просмотра.

Юниверс также определяет языковой стандарт замещения, который представляет собой языковой стандарт, использующийся в том случае, если другие стандарты недоступны.

Только языковые стандарты со статусом *Готово для использования* могут просматриваться пользователем, создающим отчет на основе юниверса.

В этих видимых языковых стандартах пользователи, создающие отчеты на основе юниверсов, могут видеть только те метаданные, которые имеют статус, принадлежащий к категории *Видимый перевод*. Категория *Видимый перевод* содержит метаданные, имеющие следующие статусы:

- NEEDS_REVIEW_TRANSLATION
- NEEDS_REVIEW_LOCALIZATION
- NEEDS_REVIEW_ADAPTATION
- TRANSLATED
- FINAL
- SIGNED_OFF

Перевод метаданных юниверса, установка языковых стандартов юниверсов и статусов метаданных выполняются в средстве управления переводами. Переводы и параметры языков содержатся в дополнительном XML-потоке и сохраняются в файлах формата .unv.

Интерфейс пользователя средства создания юниверсов также может быть отображен на различных языках. Далее следует описание функций многоязычного юниверса средства.

2.8 Определения языков и региональных стандартов

Некоторые языки используются в нескольких странах. Например, на французском языке (fr) говорят в следующих странах: Франция (FR), Бельгия (BE) и Швейцария (CH). В этом примере аббревиатура fr-FR, fr-BE, fr-CH обозначает французский (fr) как язык Франции (FR), Бельгии (BE) и Швейцарии (CH).

Язык	Страна
Французский	Франция

Язык	Страна
Французский	Бельгия
Французский	Швейцария

В то же время в одной стране может использоваться несколько языков (fr-CH, de-CH, it-CH). Например, в Швейцарии говорят на немецком, французском и итальянском.

Язык	Страна
Французский	Швейцария
Немецкий	Швейцария
Итальянский	Швейцария

Региональные стандарты определяют сочетание языка и географической области, а также способ сортировки данных. Даты и время часто отформатированы в конкретных форматах. Комбинация языка и страны (например, французский язык во Франции) в зависимости от операционной системы или приложения отображается следующим образом.

Операционная система	Формат региональных стандартов
Windows	Français (Франция) Региональные стандарты из параметров системы (страна)
Java	fr_FR
Sun Solaris	fr_FR.ISO8859-1

Для упрощения документации и приложения для обозначения языка и языковых стандартов может использоваться один термин "язык".

2.9 Различные языковые стандарты

Термины	Определение
Язык ПО	Это язык пользовательского интерфейса средства создания юниверсов. Меню и сообщения отображаются на этом языке.
Предпочитаемый язык просмотра (PVL)	Параметры предпочитаемого языка просмотра Эти параметры определяют язык отображения строк, текста и формата – частей содержания ресурса (документа или юниверса) или списка атрибутов – в приложении клиента InfoView или Web Intelligence Rich.
Языковой стандарт подстановки (замещения)	Языковой стандарт, использующийся в случае недоступности предпочтительного языкового стандарта.

Термины	Определение
Исходный язык	Языковой стандарт, в котором был создан документ.

2.10 Настройка языка продукта для пользовательского интерфейса средства создания юниверсов

На вкладке "Общие" настроек *Инструменты**Параметры* средства создания юниверсов выберите *язык* в списке доступных языков. Это язык продукта, известный как язык пользовательского интерфейса (UIL). Отсутствует воздействие на метаданные юниверса: имена объектов, контекстные имена и классы отображаются на оригинальном языке элементов базы данных. Для перевода метаданных юниверса используйте средство управления переводами.

2.11 Использование многоязычных юниверсов

Когда пользователь создает отчет на основе многоязычного юниверса, например, в Web Intelligence, языковые стандарты отображения метаданных зависят от доступности/статуса языковых стандартов и категории метаданных:

- Перевод метаданных в региональные параметры отображается только в том случае, если их статус находится в категории "Видимые", а языковые стандарты были определены как "Готово для использования".
- Языковые стандарты, в которых отображаются метаданные перевода, в порядке приоритета:
 - Предпочитаемый пользователем языковой стандарт для просмотра.
 - Если предпочитаемый языковой стандарт для просмотра недоступен, используется языковой стандарт замещения (если он был определен в данном юниверсе).
 - Если в этом юниверсе не было определено языкового стандарта замещения, то основным языковым стандартом является предпочитаемый пользователем языковой стандарт для просмотра.
 - Если же он недоступен, отображается исходное содержимое. Исходное содержимое представляет собой метаданные, как они определены в средстве создания юниверсов.

2.12 Определение языкового стандарта замещения в связанных юниверсах

Производный юниверс может повторно использовать метаданные из различных основных юниверсов. Когда производный юниверс и основные юниверсы определяют разные языковые стандарты замещения:

- Если языковой стандарт замещения определен на уровне производного юниверса, следует использовать данный языковой стандарт замещения.
- Если на уровне производного юниверса не определен языковой стандарт замещения, следует использовать такой стандарт замещения, принадлежащий первому основному юниверсу, определенному в производном юниверсе, если таковой имеется.
- Если в основных юниверсах не было определено языкового стандарта замещения, значит, в этом юниверсе нет языкового стандарта замещения.

2.13 Средство управления переводами

Средство создания юниверсов нельзя использовать для перевода метаданных юниверса или отображения переводов метаданных. В средстве создания юниверсов отображается только исходное содержимое юниверса. Для задач перевода в пакете BusinessObjects Enterprise содержится средство управления переводами. Этот инструмент представляет собой автономное приложение, доступное только на платформах Windows.

С помощью средства управления переводами разработчик юниверса может:

- Добавлять новые языковые стандарты в юниверс и помечать их как *Готовые для использования*.
- Определять языковой стандарт замещения юниверса.
- Переводить метаданные юниверса в добавленные языковые стандарты с помощью самого интерфейса.
- Задавать статус перевода метаданных в различных языковых стандартах.
- Экспортировать метаданные юниверса в файлы XLIFF для внешнего перевода и импортировать их обратно в приложение.

После того, как метаданные юниверса переведены, их можно сохранить заново, они могут понадобиться при создании многоязычных отчетов. Дополнительные сведения см. в руководстве по *средству управления переводами*.

2.14 Многоязычные данные

Посредством переменных PREFERRED_VIEWING_LOCALE и DOMINANT_PREFERRED_VIEWING_LOCALE разработчик может настраивать юниверс с целью фильтрации многоязычных данных и получать во время запроса только данные, соответствующие предпочитаемому пользователем языковому стандарту для просмотра. Это можно сделать, используя функцию @Variable

2.15 Примеры для средства создания юниверсов

В комплект поставки средства создания юниверсов входят следующие примеры:

2.15.1 Демонстрационные базы данных

Большинство примеров в данном руководстве основаны на базе данных "Клуб" построенный с помощью Microsoft Access 2000. Данная база данных используется менеджерами по продажам выдуманного предприятия, Island Resorts, для выполнения анализа дохода с продаж и маркетинга. Можно найти файл базы данных, Club.mdb, во вложенной папке баз данных в канале установки Business Objects.

Дополнительную информацию о структуре данной базы данных, см. в дополнении данного руководства.

База данных efashion также отправляется с данным выпуском. Данная база данных MS Access 2000 отслеживает 211 наименований продукта (663 наименование цветов), проданных более чем 13 магазинах (12 США, 1 в Канаде), в течении 3 лет.

База данных содержит:

- Главная таблица фактов, содержит 89,000 строк с информацией о доходах с продаж, на основе еженедельных продаж.
- Вторая таблица фактов содержащая продвижения.
- Две таблицы агрегирования, которые были установлены с навигацией агрегирования.

2.15.2 Демонстрационные юниверсы

Полный демонстрационный юниверс называется beach.unv и является составляющей частью во вложенной папке юниверсов в папке "Образец" канала установки Business Objects. Он был построен с помощью базы данных "Клуб" описанных ниже.

Этот юниверс можно использовать для знакомства с принципами построения определенных объектов и классов с помощью средства создания юниверсов.

В комплект поставки средства создания юниверсов также входит юниверс eFashion, построенный с использованием базы данных eFashion.

2.16 Использование юниверсов в средстве дизайна информации

Средство дизайна информации можно использовать для работы с юниверсами в формате .unv, созданными в средстве создания юниверсов, средстве создания юниверсов для настольных клиентов, в Universe Designer или в Universe Designer Personal. Эти файлы невозможно использовать непосредственно в средстве дизайна информации: их необходимо сначала преобразовать или обновить до более новой версии и преобразовать (в зависимости от версии файла). См. раздел средство дизайна информации, содержащий сведения о действиях, которые необходимо выполнять для использования различных версий файлов .unv, а также описание функций, поддерживаемых после преобразования файлов.

ⓘ Примечание

После преобразования файла .ipn для использования в средстве дизайна информации этот файл нельзя будет открыть в средстве, в котором он изначально был создан.

3 Выполнение основных операций

3.1 Обзор

В этой главе описываются основные операции, выполняемые в средстве создания юниверсов для создания, изменения и обновления юниверсов. Рассматриваются следующие темы:

- [Запуск средства создания юниверсов \[страница 32\]](#)
- [Импорт юниверса \[страница 42\]](#)
- [Открытие юниверса \[страница 43\]](#)
- [Экспорт юниверса \[страница 44\]](#)
- [Сохранение юниверса \[страница 46\]](#)
- [Создание юниверса и настройка его параметров \[страница 73\]](#)
- [Пользовательский интерфейс средства создания юниверсов \[страница 49\]](#)
- [Использование команд "Найти" и "Заменить" \[страница 54\]](#)
- [Организация отображения таблицы \[страница 56\]](#)
- [Выбор параметров отображения схемы \[страница 61\]](#)
- [Печать юниверса \[страница 69\]](#)

3.2 Запуск средства создания юниверсов

Средство создания юниверсов можно использовать только с репозиторием системы CMS. Перед запуском этого средства необходимо выполнить вход в репозиторий.

Перед первым запуском средства проектирования в том случае, если необходимо работать с уже существующими юниверсами, сначала необходимо открыть юниверс, сохранить его с защищенным соединением и экспортировать в репозиторий. Затем импортировать юниверс для обновления и экспортировать уже обновленные версии юниверсов в репозиторий. Таким образом будут синхронизированы версии в Центральной системе управления и локальные юниверсы.

После запуска средства создания юниверсов можно открыть юниверс одним из следующих способов:

- Создать новый юниверс
- Импортировать юниверс из репозитория Центральной системы управления
- Открыть юниверс непосредственно из файловой системы.

Юниверсы доступны пользователям Web Intelligence, только при их экспортировании в репозиторий. Самый распространенный способ работы со средством создания юниверсов – это импорт юниверса, внесение необходимых изменений и экспорт обновленной версии. Таким образом версии локальной файловой системы и репозитория Центральной системы управления будут синхронизированы.

❗ Примечание

Юниверс можно сохранить в локальной файловой системе. Это делается в случае, если необходимо обеспечить доступ к юниверсам для других пользователей, не имеющих прав доступа к Центральной системе хранения. Дополнительные сведения см. в разделе [Сохранение юниверса](#) [страница 46].

Для запуска средства создания юниверсов щелкните значок этого средства в панели задач, расположенный в группе установленных продуктов Business Objects. Перед запуском средства появится запрос на вход в CMS.

3.2.1 Запуск средства создания юниверсов

Запуск средства создания юниверсов:

1. Нажмите кнопку [Пуск](#) на панели задач.
2. Наведите указатель мыши на меню [Программы](#).
3. Выберите программу [средства создания юниверсов](#) в команде [BusinessObjects](#).
Откроется окно входа в CMS.
4. Введите следующую информацию. Как правило, эта информация предоставляется администратором BusinessObjects.

Данные для входа	Описание
Система	Имя сервера CMS.
Имя пользователя	Имя пользователя репозитория.
Пароль	Пароль для репозитория.
Аутентификация	Уровень безопасности

5. Нажмите кнопку [ОК](#).
Открывается экран запуска средства создания юниверсов, на котором представлен пустой сеанс. Имя пользователя и название репозитория появятся в строке заголовка.

В зависимости от параметров, установленных для средства создания юниверсов, мастер [быстрого запуска](#) средства проектирования может открываться автоматически при запуске этого средства. Нажмите кнопку [Отмена](#), чтобы закрыть мастер. Дополнительную информацию по отключению других параметров мастера см. в разделе [Отключение мастера нового юниверса](#) [страница 41]. При необходимости использовать мастер Quick Design см. раздел [Использование мастера Quick Design](#) [страница 33].

3.2.2 Использование мастера Quick Design

При первом запуске сеанса по умолчанию открывается [мастер нового юниверса](#). С помощью этого мастера можно быстро создать юниверс или ознакомиться со средством создания юниверсов. Однако

мастер не подходит для создания завершенного юниверса, соответствующего требованиям, которые конечный пользователь предъявляет к отчетам.

Рекомендуется отключить [мастер нового юниверса](#) и использовать его только для ознакомления со средством создания юниверсов, но не для создания юниверсов. Знакомство со сведениями и процедурами данного руководства относящимися к дизайну, построению и поддержке юниверсов во всех главах, кроме той в которой описывается непосредственно работа мастера, предполагает отключение мастера [Быстрый дизайн](#). [Использование мастера Quick Design \[страница 33\]](#) Дополнительные сведения об отключении прочих параметров мастера [Быстрый дизайн](#) см. в разделе [Отключение мастера нового юниверса \[страница 41\]](#).

3.3 Создание базового юниверса с помощью мастера нового юниверса

Для демонстрации или быстрой проверки юниверса, основанного на простой реляционной схеме, в [средстве проектирования юниверсов](#) представлен мастер нового юниверса для создания базового завершенного юниверса. Можно немедленно использовать юниверс результата или изменять и создавать новые сложные объекты. В данном случае можно постепенно уточнять качество и структуру юниверса.

В процессе разработки юниверса продукции его необходимо создавать вручную. В остальных главах этого руководства описывается, как создавать юниверс вручную. Это единственный раздел, в котором говорится об автоматическом создании юниверса.

3.3.1 Зачем использовать мастер быстрого создания?

Мастер быстрого создания оказывает помощь на всем протяжении создания юниверса. Он помогает установить соединение с базой данных, а затем позволяет создать простые классы и объекты. Мастер также предоставляет встроенные стратегии для автоматического создания объектов, объединений и таблиц.

Использование быстрого создания имеет следующие преимущества:

- Он поможет научиться работать с пользовательским интерфейсом и дизайном базового юниверса, если пользователь не знаком со средством создания юниверсов.
- Он сохранит время, автоматизировав многие процессы при создании юниверса демонстрации. С мастером можно быстро настроить рабочую модель юниверса, а затем можно настроить юниверс, чтобы он следовал нуждам целевой аудитории.

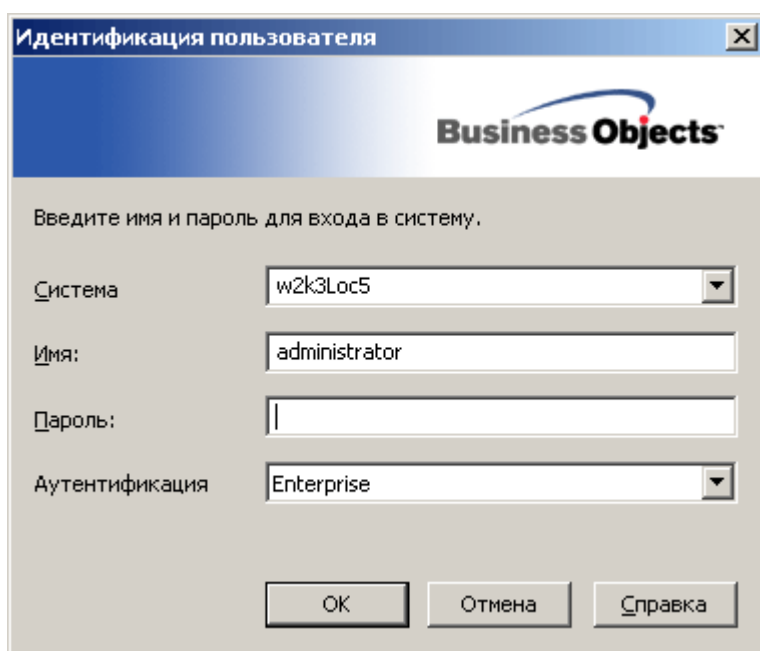
3.3.2 Использование мастера быстрого дизайна

Мастером быстрого создания называется мастер, который используется для автоматического создания юниверса. Каждый шаг мастера описан в каждом из следующих разделов.

3.3.2.1 Запуск мастера Quick Design

Чтобы запустить мастера быстрого дизайнера, выполните следующие действия:

1. Запустите средство создания юниверсов.
Отобразится диалоговое окно "Идентификатор пользователя".



2. Введите имя пользователя и пароль в диалоговом окне идентификатора пользователя.
3. Нажмите кнопку **OK**.
Отобразится экран приветствия мастера быстрого дизайна.

❗ Примечание

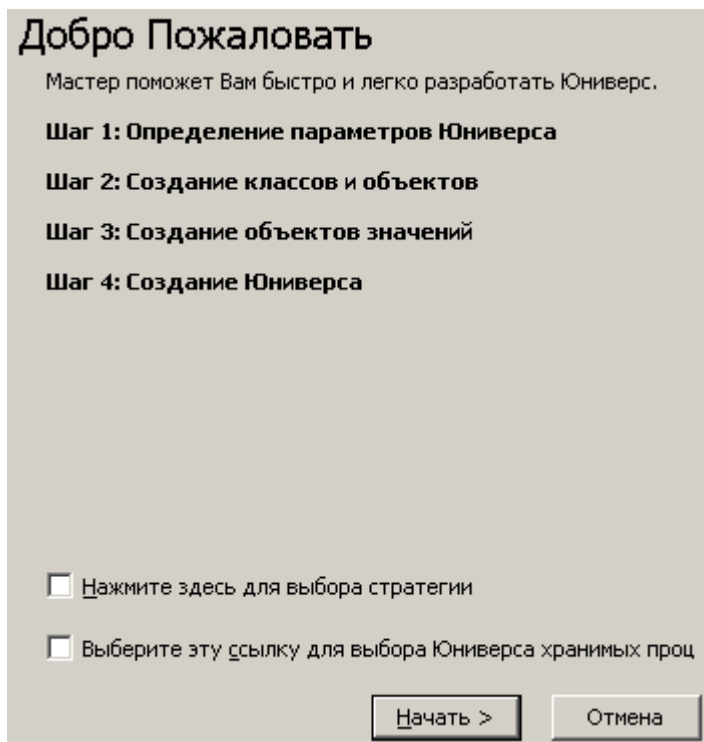
Если нет необходимости отображать мастер при следующем запуске сеанса, снимите флажок "Запускать мастер при загрузке". Кроме того, можно найти два параметра относящиеся к отображению мастера во вкладке "Общие" диалогового окна "Параметры": "Отобразить мастер приветствия" и "Запуск мастера быстрого дизайна" "Файл/Создать" (меню "Инструменты", команда "Параметры").

3.3.2.2 Экран приветствия

Экран приветствия отображает просмотр четырех шагов необходимых для создания основного юниверса. Он также предоставляет опцию параметра: Нажмите здесь для выбора стратегии. Если установить этот флажок, можно будет выбирать стратегии для создания юниверса; в противном случае средство создания юниверсов применяет встроенные стратегии по умолчанию.

В каждом из следующих диалоговых окон мастер быстрого создания запрашивает данные, необходимые для выполнения действия.

Для перехода между диалоговыми окнами используйте кнопку [Далее](#). Можно вернуться к предыдущему диалоговому окну, нажав кнопку [Назад](#). Можно завершить процесс и выйти из мастера быстрого создания в любой момент, нажав кнопку "Отмена".



Добро Пожаловать

Мастер поможет Вам быстро и легко разработать Юниверс.

Шаг 1: Определение параметров Юниверса

Шаг 2: Создание классов и объектов

Шаг 3: Создание объектов значений

Шаг 4: Создание Юниверса

☐ Нажмите здесь для выбора стратегии

☐ Выберите эту ссылку для выбора Юниверса хранимых проц

Начать > Отмена

При выборе опции "Нажмите здесь для установления параметра стратегий", диалоговое окно стратегий отобразится и выдаст список стратегий. Данное диалоговое окно описано в [Выбор стратегий \[страница 37\]](#). Можно выбрать стратегию или принять стратегии по умолчанию.

Нажмите [Начало](#), чтобы начать процесс создания.

3.3.2.3 Определение параметров юниверса

На этом шаге определяются параметры юниверса: имя юниверса и соединение с базой данных.

Можно указать для юниверса имя, длиной до 35 букв.

⚠ Предупреждение

Не используйте в имени юниверса следующие специальные символы: \ / : * ? " < > | &

Определение Параметров Универса

Для создания Универса Вам необходимо определить логическое имя и Соединение с базой.

◆ **Введите имя Универса**

Sales Analysis

◆ **Если Вы хотите создать Соединение, нажмите кнопку 'Новое...'.**

Новое...

◆ **Укажите Соединение с базой**

beach

Тест Изменить...

< Назад Далее > Отмена

Можно или создать соединение, или выбрать уже существующее. Чтобы создать соединение, нажмите кнопку "Создать" и установите необходимые параметры в следующих диалоговых окнах. Дополнительные инструкции по данным диалоговым окнам см. в разделе [Изменение параметров идентификации юниверсов \[страница 78\]](#)

Для проверки соединения нажмите кнопку "Проверить". Кнопка "Редактировать" позволит изменить параметры соединения.

Нажмите кнопку "Далее", чтобы перейти к следующему действию.

3.3.2.4 Выбор стратегий

При включении параметра стратегий на экране приветствия мастер быстрого создания предложит указать стратегии для создания объектов, объединений и таблиц.

Стратегия представляет собой скрипт, который считывает структурную информацию из файла базы данных или простого файла. Эти скрипты используются в средстве создания юниверсов для автоматического создания объектов, объединений и таблиц.

Укажите Стратегию

Мастер формирует классы и объекты, определяет связи и отношения в соответствии со следующей стратегией:

◆ **Укажите стратегию для объекта**

(Встр.) Станд. Переименование

Создает классы из имен таблиц и объекты из имен колонок. Заменяет '_' пробелами.

◆ **Укажите стратегию связей**

Редактировать вручную (нет)

◆ **Выберите стратегию таблиц**

(Встроенная) Стандартная

Читает структуру таблицы из системных таблиц.

< Назад Далее > Отмена

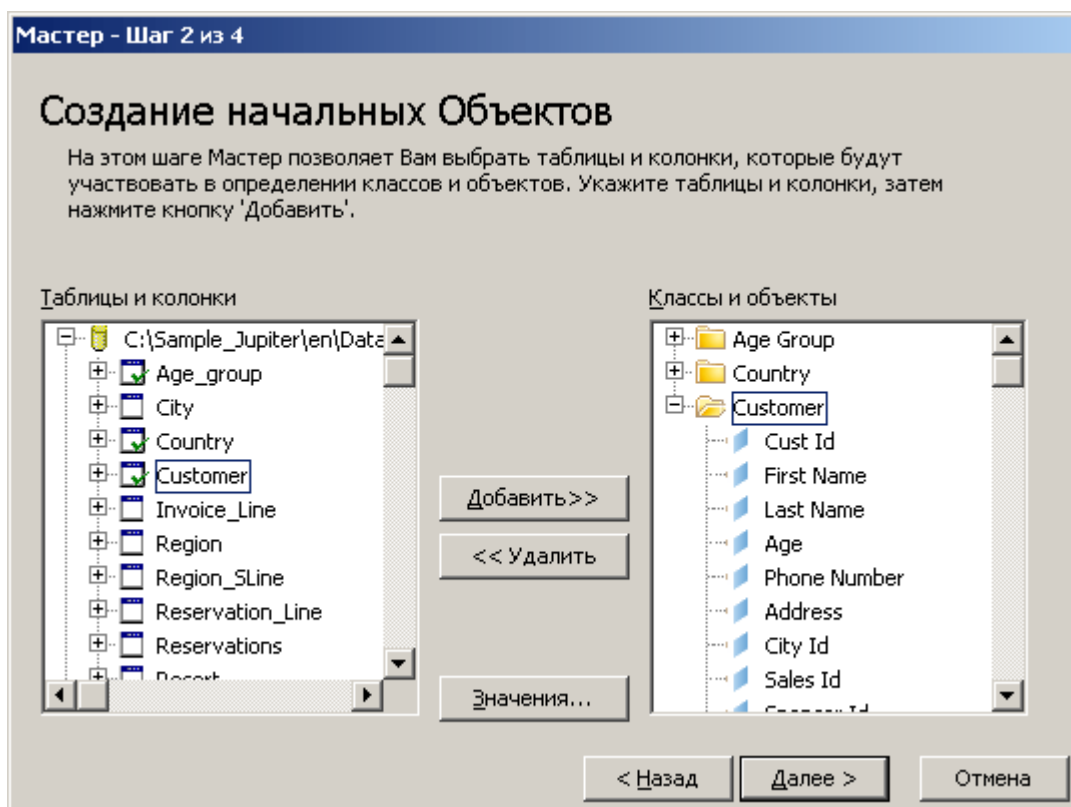
В поле списка можно выбрать другую стратегию, или не выбрать ничего. Краткие описания текущих стратегий отображаются ниже полей списка.

В дополнение к встроенным внутренним стратегиям, представленным в средстве создания юниверсов, можно создавать собственные внешние стратегии. Дополнительные сведения см. в разделе [Использование внешних стратегий для настройки процесса создания юниверса \[страница 399\]](#).

Нажмите кнопку "Далее", чтобы перейти к следующему действию.

3.3.2.5 Создание начальных классов и объектов

Основанный на параметрах соединения с базами данных, мастер представлен вместе со списком таблиц и столбцов баз данных. Для создания начальных классов и объектов выберите таблицы и столбцы из левой панели и добавьте их в панель классов и объектов юниверса справа.



Левая панель показывает имена таблиц по умолчанию. Можно также использовать следующие способы, чтобы переместить древовидную структуру файлов и добавить классы и объекты в правую панель:

- Для просмотра столбцов любой таблицы нажмите знак плюс (+) слева от названия таблицы.
 - Для просмотра значений данных любой таблицы или столбца нажмите таблицу или столбец, а затем нажмите кнопку "Просмотр значений".
 - Чтобы выбрать одну таблицу, нажмите таблицу, а затем нажмите кнопку "Добавить".
 - Чтобы выбрать несколько смежных таблиц, удерживайте клавишу "SHIFT", затем нажмите на первую и последнюю таблицу. Все таблицы между теми, которые нажали, будут выделены. Затем нажмите кнопку "Добавить".
 - Чтобы выбрать несколько несмежных таблиц, нажимайте таблицы по очереди, удерживая клавишу "CTRL". Нажмите кнопку "Добавить".
 - По-другому таблицы можно выбрать, перетаскивая их из левой панели в правую.
- При вставке таблицы средство создания юниверсов включает все содержащиеся в ней столбцы.

На правой панели имена и классы отображаются рядом со значком папки. Нажмите символ (+) рядом с именем класса для просмотра объектов. Можно переименовать класс или объект посредством двойного нажатия, а затем введением нового имени в диалоговом окне.

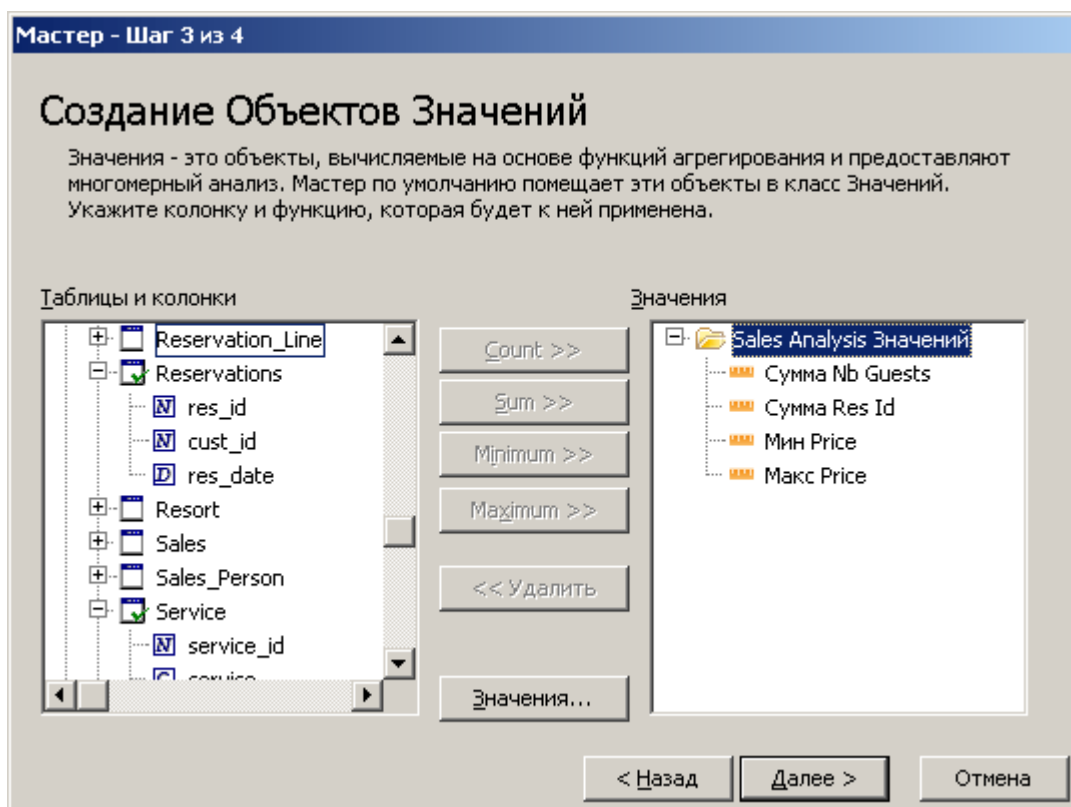
По умолчанию объект определяется как объект измерения, на что указывает символ куба, предшествующий имени объекта.

Чтобы удалить класс или объект, нажмите его, а затем нажмите кнопку "Удалить".

Нажмите [Далее](#) для перехода к следующему шагу.

3.3.2.6 Создание объектов мер

Объект мер является результатом функции агрегирования: Count, Sum, Minimum или Maximum. Этот тип объекта предоставляет цифровую информацию. Примеры объектов мер отображаются на правой панели диалогового окна ниже:



Если необходимо просмотреть значения данных какого-либо объекта, нажмите объект, а затем нажмите кнопку "Просмотр значений"

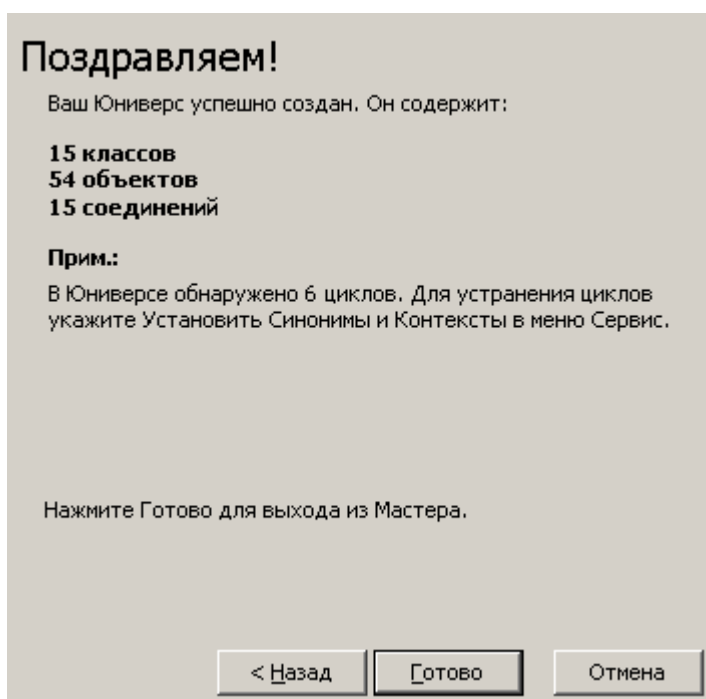
Что создать объект мер, нажмите соответствующий объект на левой панели, а затем нажмите кнопку "Агрегирование". Можно изменять имя любого созданного объекта мер.

Группирование объектов в один или более классов мер улучшает организационную структуру юниверса. Это также упрощает конечным пользователям перемещение по юниверсу. Дополнительные сведения об объектах мер, см. в разделе [Определение меры \[страница 291\]](#).

При нажатии кнопки [Далее](#) мастер Quick Design начинает создание юниверса.

3.3.2.7 Создание юниверса

Мастер Quick Design автоматически создает новый юниверс, основанный на установленных параметрах. Он определяет количество классов, объектов и объединений, созданных в юниверсе.



В диалоговом окне выше, находится сообщение о состоянии цикла существующего в объединениях юниверса. С помощью средства создания юниверсов можно устранять циклы с псевдонимами и контекстами. Дополнительные сведения см. в главе "Разработка схем".

При нажатии кнопки [Готово](#) отобразится панель юниверса и панель структуры нового юниверса.

3.3.2.8 Завершение рабочей сессии

Выберите Файл > Сохранить как, чтобы сохранить юниверс, затем Файл > Закр^ыть, чтобы закрыть юниверс.

При сохранении юниверса в средстве создания юниверсов отображается подсказка имени файла. Имя файла юниверса может содержать максимальное количество символов разрешенных операционной системой. Имеет расширение unv. По умолчанию эти файлы средства создания юниверсов хранятся в папке юниверса, вложенной в папку BusinessObjects. В Windows 2000 эта папка отображается ниже папки локальных данных вашей учетной записи.

Чтобы выйти из средства создания юниверсов, выберите "Файл > Выход".

3.3.2.9 Отключение мастера нового юниверса

При первом запуске сеанса [мастер нового юниверса](#) открывается по умолчанию. Можно предотвратить автоматическое отображение мастера при создании нового юниверса следующим образом:

Чтобы отключить мастера [Быстрый дизайн](#):

1. Выберите ► *Инструменты* ► *Параметры* .
Откроются *Параметры* диалогового окна на *Основной* странице.
2. Удалите флажок *Отображение мастера приветствия*. (Этот флажок уже удален, если ранее был удален флажок *Запустить мастер при загрузке* на начальной странице *Запуск мастера*.)
3. Удалите флажок *Запуск мастера быстрого дизайна Файл/Создать*.
4. Нажмите кнопку *ОК*.

Можно запустить мастер *Быстрый дизайн* в любой момент, выбрав выше флажков на странице *Общее* в диалоговом окне *Параметры*. Использование мастера *Быстрый дизайн* описано в разделе *Использование мастера Quick Design* [страница 33].

3.3.3 Внесение изменений в юниверс, созданный с помощью мастера нового юниверса

После создания базового юниверса с помощью мастера нового юниверса, может потребоваться редактирование связи и устранение всех циклов с помощью псевдонимов и контекстов. Также можно улучшить юниверс посредством более сложных компонентов, используя различные функции средства создания юниверсов. За необходимыми сведениями следует обратиться к соответствующему разделу.

3.4 Импорт юниверса

Можно импортировать один или несколько юниверсов, которые хранятся в папке юниверсов в репозитории. Импортировать можно только тот юниверс, который был экспортирован в репозиторий.

При выполнении импорта юниверса CMS проверяет версию юниверса в файловой системе репозитория. Если версия идентична, юниверс передается в средство создания юниверсов. Если версия юниверса в файловой системе репозитория более поздняя, чем версия CMS, откроется окно сообщения с просьбой переместить юниверс в папку. При ответе *Да* юниверс файловой системы репозитория перемещается с помощью более поздней версии в CMS.

3.4.1 Импорт юниверса из репозитория

3.4.1.1 Чтобы совершить импорт юниверса из репозитория,

1. Выберите ► *Файл* ► *Импорт* .
Откроется диалоговое окно *Импорт юниверса*.
 2. Выберите папку юниверсов в окне раскрывающегося списка.
- Или

Нажмите кнопку [Обзор](#) и выберите юниверс с помощью обозревателя папки.

Необходимо импортировать юниверс из этой папки.

3. Чтобы заблокировать юниверс, дважды нажмите имя юниверса.

Заблокированный юниверс отображается с символом замка. Чтобы разблокировать юниверс, дважды щелкните его еще раз.

4. Нажмите имя юниверса.

Это юниверс, который необходимо импортировать.

5. Проверьте путь файла для папки импорта в поле [Импорт папки](#).

Юниверс импортируется в данную папку.

6. Нажмите кнопку [ОК](#).

3.4.2 В чем разница между открытием и импортированием?

Данный юниверс можно открыть прямо из файловой системы. При сохранении данного юниверса, он сохраняется только в файловой системе, без обновления в Центральной системе управления. Пользователям Web Intelligence недоступно обновление данного юниверса.

При импорте юниверса текущая версия, доступная в репозитории, передается в средство создания юниверсов. После завершения ввода изменений юниверса, он экспортируется обратно в репозиторий. Центральная система управления обновляется в соответствии с последними изменениями.

3.5 Открытие юниверса

Юниверс открывается с помощью команды меню или нажатием кнопки [Открыть](#). При открытии юниверса без импортирования открывается версия локальной файловой системы, которая может отличаться от текущей версии в Центральной системе управления.

3.5.1 Для открытия юниверса без импортирования

1. Выберите [Файл](#) > [Открыть](#).

Диалоговое окно [Открыть](#) позволяет открыть каталог, указанный в качестве хранилище файла юниверса по умолчанию. Данный каталог можно установить на странице [Сохранить](#) диалогового окна [Параметры](#) ([Сервис](#) > [Параметры](#) > [Сохранить](#)).

2. Если необходимо, переместитесь в каталог с файлом юниверса (UNV).
3. Выберите файл юниверса и нажмите кнопку [Открыть](#).

Или

Дважды щелкните мышью файл юниверса.

Юниверс открывается в текущем окне средства создания юниверсов.

3.6 Экспорт юниверса

При экспортировании юниверса в репозиторий, он становится доступным для пользователей Web Intelligence и других дизайнеров.

При экспортировании юниверса он:

- Перемещается в выбранную папку юниверса в репозитории файловой системы и
- создается копия в Центральной системе управления

При каждом экспортировании в репозиторий, обновляется версия данного юниверса в Центральной системе управления. Эта версия доступна для пользователей Web Intelligence.

📌 Примечание

Сохранение юниверса и его экспортирование – это не одно и то же. При сохранении юниверса, обновляется его версия только в репозитории локальной файловой системы, но не в Центральной системе управления. Дополнительные сведения см. в разделе [В чем разница между экспортом и сохранением?](#) [страница 45].

3.6.1 Как расположены юниверсы в файловой системе репозитория?

Репозиторий хранит юниверсы в локальной файловой системе и на сервере Центральной системы управления. Работа ведется с версией юниверса локальной файловой системы. Локальная файловая система – это сервер, на котором установлено данное средство. По умолчанию юниверсы сохраняются в папке для юниверсов. Путь к папке юниверсов выглядит следующим образом:

```
\\Documents and Settings\<пользователь>\Application Data\Business Objects\Business Objects 12.0\universes\@<имя репозитория>\universe folder\<юниверс>.unv
```

Юниверсы, хранящиеся на сервере Центральной системы управления используются для управления версиями. При экспорте обновленного юниверса в репозиторий он также копируется на сервер Центральной системы управления.

3.6.2 Экспортирование юниверса в репозиторий

3.6.2.1 Для экспортирования юниверсов в репозиторий

1. Выберите **Файл** > **Экспорт**.

Откроется диалоговое окно **Экспорт юниверса**.

2. В раскрывающемся списке выберите нужную папку для юниверса.

Или

Нажмите кнопку **Обзор** и в проводнике папок выберите папку юниверса.

Необходимо экспортировать юниверс в данную папку.

3. Чтобы заблокировать юниверс, дважды нажмите имя юниверса.

Заблокированный юниверс отображается с символом замка. Чтобы разблокировать юниверс, дважды щелкните его еще раз.

4. Выберите группу в окне списка **Группы**. Это группа пользователей, которая использует экспортированные юниверсы.

5. Выберите юниверс в окне списка **Юниверсы**. В окне списка **Юниверсы** показаны имена активных юниверсов.

6. Если необходимо экспортировать другие неоткрытые юниверсы, нажмите кнопку **Добавить юниверс**, а затем выберите их с помощью проводника.

7. Нажмите кнопку **ОК**.

3.6.3 В чем разница между экспортом и сохранением?

При сохранении юниверса, обновляется версия юниверса в репозитории файловой системы. При этом в Центральной системе управления версия остается без изменений.

При экспорте юниверса, обновление версии в репозитории файловой системы синхронизируется с обновлением юниверса в Центральной системе управления.

Если юниверс сохраняется без экспортирования, юниверс в Центральной системе управления не обновляется. Сохраненный без экспортирования юниверс недоступен другим пользователям.

Каждый юниверс в репозитории назначается системным идентификатором. Дополнительные сведения об идентификаторах см. в [Определение юниверса в репозитории \[страница 507\]](#).

Нельзя экспортировать юниверс, если он был заблокирован в репозитории другим дизайнером.

Можно экспортировать только определенные защищенным соединением юниверсы.

3.7 Сохранение юниверса

Во время рабочего сеанса необходимо регулярно сохранять юниверсы. При сохранении юниверса средство создания юниверсов сохраняет его в локальной файловой системе как файл с расширением `.unv`.

В Web Intelligence пользователь задает имя юниверса (название).

При сохранении юниверса, данные изменения не сохраняются в Центральной системе управления. Для обновления юниверса в Центральной системе управления необходимо его экспортировать.

Можно использовать в имени юниверса следующее максимальное количество символов (полное имя) и расширение `.unv`:

Тип имени	Максимальное количество символов
Имя юниверса	100 (символы \, /, :, *, ?, ", <, >, и & не поддерживаются)
Имя <code>.unv</code>	Максимальное количество символов для операционной системы

3.7.1 Имена файлов юниверсов в качестве идентификаторов

Не изменяйте имя файла юниверса `.unv` после создания отчета на основе этого юниверса. При изменении имени файла отчеты, созданные на основе юниверса с прежним именем, не будут ссылаться на юниверсы, имена которых были изменены.

3.7.2 Сохранение юниверса

Имя юниверса может отличаться от имени с расширением `.unv`.

При использовании функции [Сохранить как](#) для сохранения юниверса под новым именем, имя юниверса в Центральной системе управления не изменяется автоматически. Для занесения данного юниверса в Центральную систему управления нужно экспортировать в нее новый юниверс.

Для сохранения юниверса нужно выполнить следующие действия:

- Выберите ► [Файл](#) ► [Сохранить](#) ► на панели меню
- Щелкните значок [Сохранить](#)
- Нажмите сочетание клавиш [CTRL+S](#).

3.7.3 Сохранение определения юниверса в формате PDF

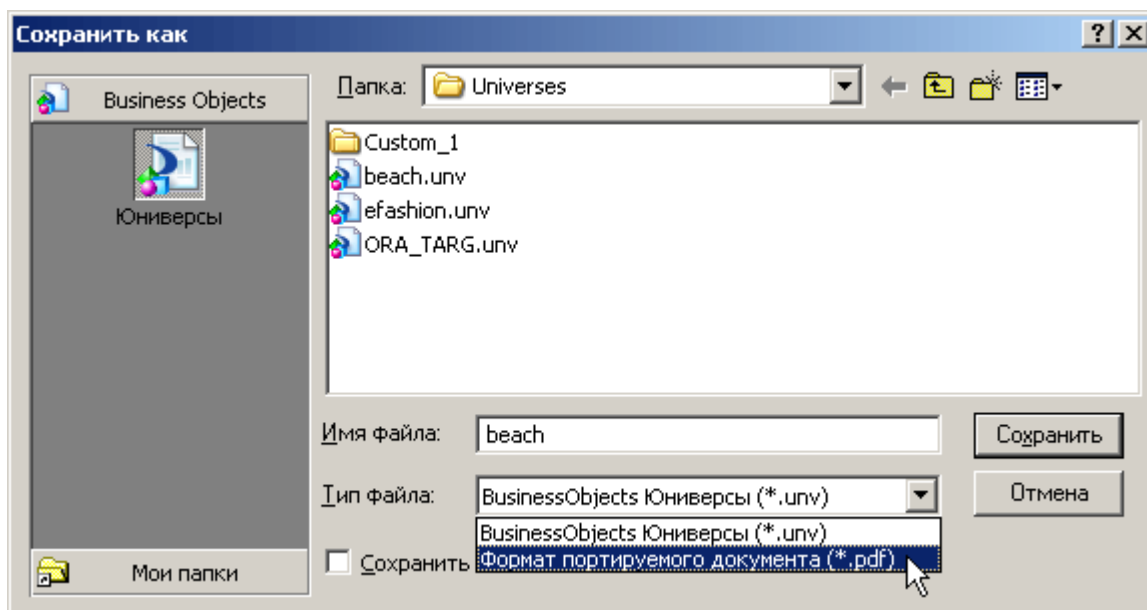
Можно сохранять сведения о юниверсе в виде файла Adobe PDF. Ту же информацию можно сохранять таким образом, чтобы потом вывести на печать для юниверса. Данные сведения содержат следующее.

- Общие сведения: параметры, связанные юниверсы и схема графических таблиц.
- Списки компонентов: списки компонентов в юниверсе, включающие объекты, условия, иерархии, таблицы, объединения и контексты.
- Описания компонентов: описания объектов, условий, иерархий, таблиц, объединений и контекстов в юниверсе.

В диалоговом окне "Параметры печати" можно настроить вывод определенных компонентов для отображения в PDF (► [Сервис](#) ► [Параметры](#) ► [Печать](#) ►). Дополнительные сведения по настройке параметров см. [Печать юниверса \[страница 69\]](#).

Для сохранения информации юниверса в формате файла PDF:

1. Выберите ► [Файл](#) ► [Сохранить как](#) ►.
2. В раскрывающемся списке *Тип файла* выберите *Формат переносимого документа (*.pdf)*.



3. Щелкните [Сохранить](#).

3.7.3.1 Установка параметров сохранения по умолчанию

По умолчанию в средстве создания юниверсов хранятся файлы, сохраняемые в подкаталоге "Юниверс" каталога установки приложения Business Objects. Можно указать другой каталог сохранения по умолчанию, как показано ниже.

1. Выберите ► *Инструменты* ► *Параметры* ▾
Откроется диалоговое окно *Параметры*.
2. Выберите вкладку *Сохранить*.
Откроется страница *Сохранить*.
3. Введите путь файла в текстовом окне "*Каталоги юниверса по умолчанию*".
Или
4. Просмотрите каталог, содержащий файлы .unv.
5. Если необходимо указать время автоматического сохранения, установите флажок "*Сохранить автоматически*" и выберите или введите число периода времени в окне выбора *значений минут*.
6. Нажмите кнопку *ОК*.

3.8 Заккрытие юниверса

Для закрытия юниверса можно использовать следующие способы.

Чтобы закрыть юниверс, выполните следующие действия.

- На панели меню выберите ► *Заккрыть* ► *Файл* ▾
- Нажмите кнопку "*Заккрыть окно*" в правом верхнем углу окна юниверса.
- Нажмите сочетание клавиш *CTRL+W*.

3.9 Работа с несколькими дизайнерами

Можно использовать средство создания юниверсов в многопользовательской среде, в которой несколько разработчиков могут работать в одинаковых юниверсах, не вызывая конфликтов в версиях.

Можно заблокировать юниверс так, что только один дизайнер может сразу выполнять изменения в юниверсе, а для юниверса назначается номер версии, чтобы сохранить отслеживание изменений.

3.9.1 Блокировка юниверса

Если юниверс хранится в папке юниверсов, его могут совместно использовать несколько дизайнеров, заверяя, что у них есть необходимые права пользователя.

Только один дизайнер может работать в данном юниверсе. Если дизайнеру необходимо работать в юниверсе, он может это сделать, только если другой дизайнер ранее не заблокировал данный дизайнер.

❗ Примечание

Заблокируйте юниверс в диалоговом окне "Импорт" или "Экспорт". Если юниверс заблокирован, рядом с именем данного юниверса отображается символ замка. Если другой дизайнер блокирует юниверс, символ замка отображается неактивным.

3.9.2 Номер редакции

При экспорте юниверса в папку юниверса средство создания юниверсов последовательно увеличивает номер выпуска юниверса. Это позволяет определять самую позднюю версию юниверса. Номер редакции отображается на вкладке "Итог" параметров юниверса ("Файл > Параметры юниверса > Итог").

3.10 Пользовательский интерфейс средства создания юниверсов

Пользовательский интерфейс средства создания юниверсов соответствует стандартам ОС Microsoft Windows. Его основными элементами являются окна, меню, панели инструментов, комбинации клавиш и интерактивная справка.

3.10.1 Основные компоненты интерфейса пользователя

Каждый юниверс содержится в собственном окне юниверса, которое находится в главном окне приложения средства проектирования.

Также существует независимое окно *Обозреватель таблиц*, в котором отображены все таблицы, доступные в подключенной базе данных.

3.10.1.1 Окно юниверса

Окно *юниверса* разделяется на две панели.

Панель	Отображает
<i>Структура</i>	Графическое представление лежащей в основании целевой базы данных юниверса. Оно включает в себя таблицы и объединения, в которые вносятся объекты, используемые конечными пользователями для запросов.
<i>Юниверс</i>	Классы и объекты, определенные в юниверсе. Существуют компоненты юниверса, которые пользователи Web Intelligence видят и используют для создания своих запросов.

3.10.1.2 Обзорщик таблиц

Обзорщик **таблиц** – это окно, в котором отображаются таблицы подключенной базы данных. Таблицы можно вставлять в панель **структуры**, перетаскивая их в панель **Структура** или дважды нажав необходимую таблицу в обзорщике **таблиц**.

Обзорщик **таблиц** можно отобразить одним из следующих способов.

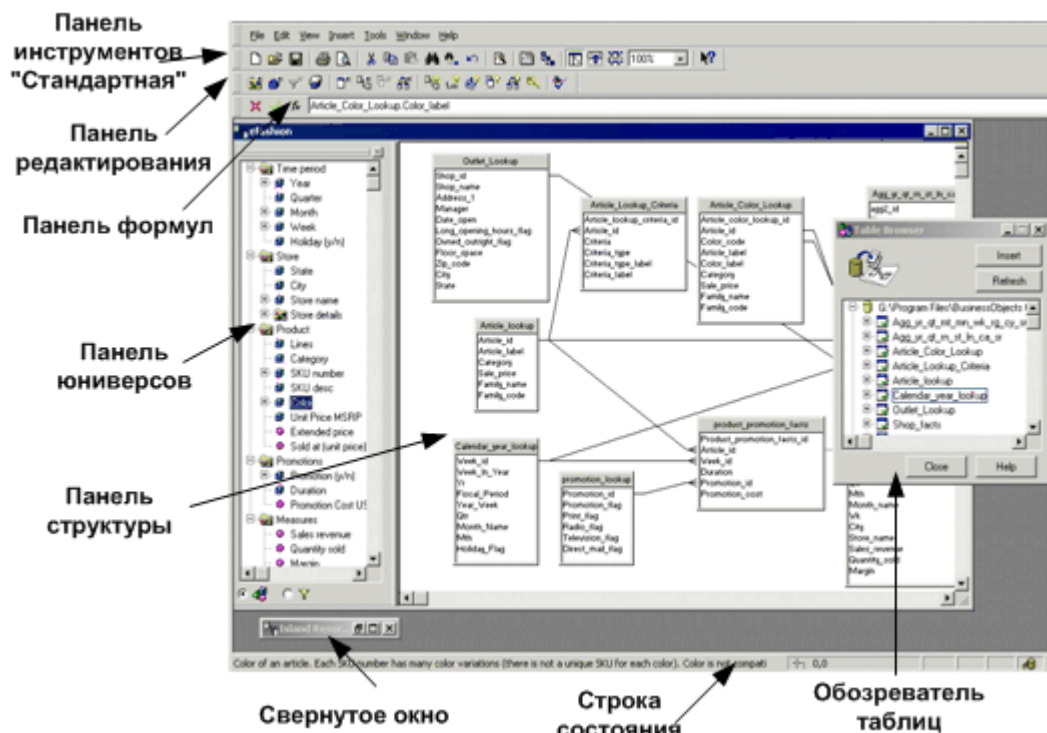
- Нажмите два раза фон панели **структуры**.
- Нажмите правой кнопкой мыши фон панели **структуры** и выберите в контекстном меню **Вставить таблицу**.
- Выберите ► **Вставить** ► **Таблицы** ►.

Примечание

Использование обзорщика таблиц полностью описано в главе "Проектирование схемы".

3.10.2 Пользовательский интерфейс средства создания юниверсов

Основные компоненты интерфейса приведены ниже.



3.10.3 Управление окнами

Окна пользовательского интерфейса можно использовать для следующих целей:

- Во время сеанса можно работать сразу с несколькими юниверсами. Средство проектирования отображает каждый юниверс на отдельных панелях [Структура](#) и [Юниверс](#).
- Недавно открытые юниверсы перечислены в нижней части меню [Файл](#). Изменить количество юниверсов в списке можно, выбрав ► [Инструменты](#) ► [Параметры](#) ► [Общие](#) ▾ и указав количество юниверсов в списке [Недавние файлы](#).
- Все окна в средстве создания юниверсов можно перемещать и сворачивать, также можно изменять их размер.
- Чтобы разместить окно в наиболее удобном месте выберите ► [Окно](#) ► [Упорядочить](#) ▾ и [Каскад](#), [Горизонтально](#) или [Вертикально](#).
- Чтобы выровнять окна, свернутые в окне средства проектирования, выберите ► [Окно](#) ► [Упорядочить значки](#) ▾.

3.10.4 Использование панелей инструментов

Окно средства создания юниверсов содержит два набора панелей инструментов: панель инструментов [Стандартная](#) и панель инструментов [Редактирование](#).

Доступные кнопки для каждой панели инструментов зависят от того, какая панель активна, панель [юниверса](#) или панель [структуры](#). Недоступные кнопки отображаются как неактивные.

Панели инструментов могут закрепляться. Можно перетащить панель инструментов и закрепить ее в любом месте окна юниверса.

3.10.4.1 Перемещение панели инструментов

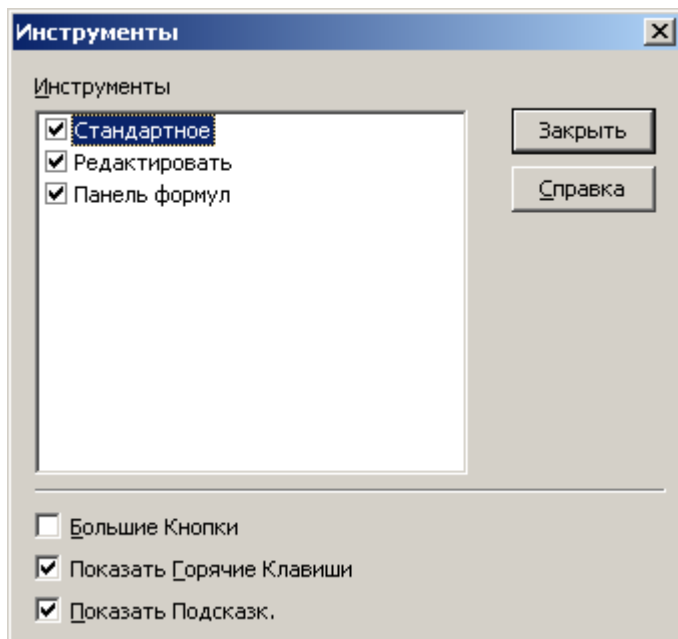
Чтобы переместить панель, выполните следующие действия.

1. Нажмите область в четырехугольнике, содержащим панель.
Эта область показана для обеих панелей на предыдущем изображении.
2. Удерживая левую кнопку мыши, перетащите панель инструментов в необходимое место.
3. Отпустите кнопку мыши.
Панель инструментов отображается независимо.

3.10.4.2 Скрытие и отображение панелей инструментов

Чтобы скрыть или отобразить панель инструментов, выполните следующие действия.

1. Выберите ► [Вид](#) ► [Панели инструментов](#) ▾.
Откроется диалоговое окно [панелей инструментов](#).



2. Установите или снимите флажок соответствующей панели инструментов.
3. Выберите или отмените выбор параметров для отображения кнопок панелей инструментов, подсказок и горячих клавиш, отображенных в нижней части диалогового окна.
4. Нажмите кнопку [OK](#).

3.10.5 Выполнение действия или операции в средстве создания юниверсов

В средстве создания юниверсов любые действия и операции можно выполнить следующими способами:

- Выбрав команду в меню.
- Нажав клавишу **Alt** и клавишу быстрого вызова команд на клавиатуре.
- Нажав кнопку на панели инструментов.

3.10.5.1 Использование мыши

В этом средстве можно использовать одиночный и двойной щелчок мышью следующим образом:

Одиночный щелчок

Можно использовать одиночный щелчок в следующих действиях.

- Выполнение стандартного действия (выбор команды или нажатие кнопки)
- Выбор элемента на панели [Юниверса](#), панели [Структуры](#), или в [Обозревателе таблиц](#).
- При выборе одного или нескольких компонентов в окне средства создания юниверсов с помощью одиночного нажатия правой кнопки мыши отображается всплывающее меню. Оно содержит команды, связанные с выбранными компонентами.

Дважды щелкните мышью

Можно дважды щелкнуть мышью следующие структуры юниверса для влияния изменения отображения или для изменения свойств.

Дважды щелкните мышью...	Результат...
Пустое место на панели Структура	Обозреватель таблицы отобразится.
Таблица на панели структуры	Изменяет отображение таблицы. Таблица и столбцы отображаются в одном из трех окон. Дополнительные сведения см. в разделе Использование режима списка [страница 58].
Объединение на панели структуры	Для объединения откроется диалоговое окно Редактировать объединение . В этом окне можно изменять свойства объединения.
Класс на панели Юниверса	Для класса отобразится диалоговое окно Редактировать свойства . В этом окне можно изменять свойства класса.
Объект на панели Юниверса .	Для объекта отобразится диалоговое окно Редактировать свойства . В этом окне можно изменять свойства объекта.
Объект условия в окне Условие на панели инструмента Юниверса	Для объекта условия отобразится диалоговое окно Редактировать свойства . В этом окне можно изменять свойства объекта.

3.10.5.2 Отмена действия

Можно отменить выполненное ранее действие двумя способами.

- Выберите ► [Редактировать](#) > [Отмена](#) ►.
- Выберите пункт [Отмена](#).

3.11 Использование команд "Найти" и "Заменить"

Можно использовать команду *Найти*, чтобы найти символы или текстовую строку в юниверсе и структуре панели. Можно использовать команду *Найти и Заменить*, чтобы найти и заменить символы или текст в именах и описаниях для любой структуры в юниверсе.

3.11.1 Использование команды "Найти"

В юниверсе и на панелях структур можно выполнить поиск текста, который содержится в структурах юниверса.

3.11.1.1 Установка параметров функции "Найти"

Доступные параметры функции *Найти* зависят от активности либо панели *Юниверса*, либо от *Структуры*.

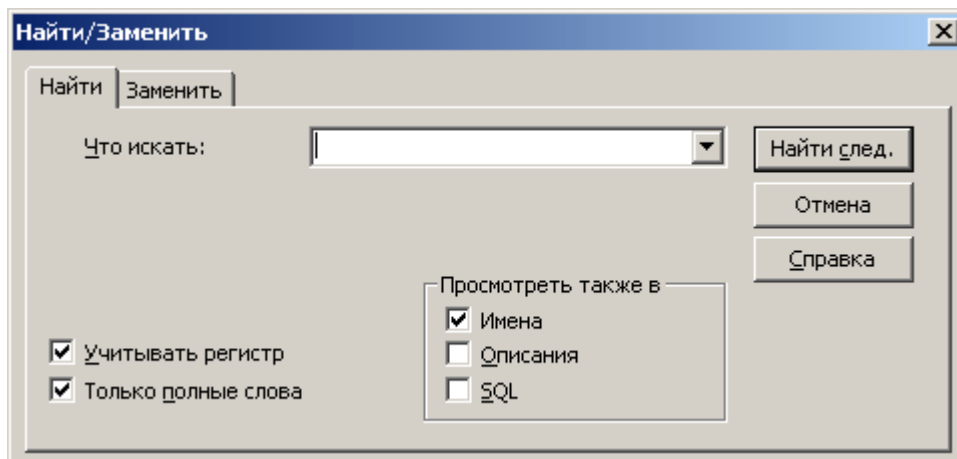
Для поиска строки можно выбрать следующие параметры поиска.

Параметр	Параметр доступен	Описание
Найти	При активности панели <i>Юниверса</i> или <i>Структуры</i>	Строка для поиска.
Учитывать регистр	При активности панели <i>Юниверса</i> или <i>Структуры</i>	Выполнить поиск с учетом верхнего и нижнего регистра.
Выполнить поиск целого слова	При активности панели <i>Юниверса</i> или <i>Структуры</i>	Выполнить поиск целой строки.
Искать также в имени	Когда активна панель <i>Юниверса</i>	Когда выбраны, классы поиска и имена объекта или только predetermined условия имен. Когда удалены, классы, объекты predetermined условия имен не включены в поиск.
Искать также в описаниях	Когда активна панель <i>Юниверса</i>	Когда выбраны, включая все описания структуры юниверсов в поиске.
Искать также в SQL	Когда активна панель <i>Юниверса</i>	Когда выбрано, включается в поиск определения SQL объектов объединений и других структур юниверса.

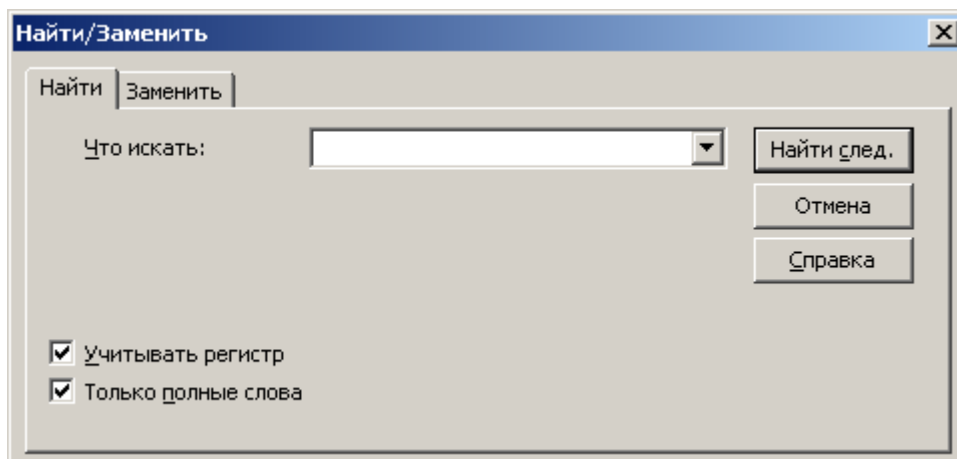
3.11.1.2 Поиск в юниверсе

Для поиска в юниверсе выполните следующие действия:

1. Щелкните на панели значок [Юниверс](#) или [Структура](#).
При необходимости найти строку на данной панели.
2. Выберите ► [Редактировать](#) ► [Найти](#) ►.
Отобразится окно [Найти/Заменить](#). Окно для активности панели [Юниверса](#) расположено ниже.



Окно для активности панели [Юниверса](#) отображается ниже.

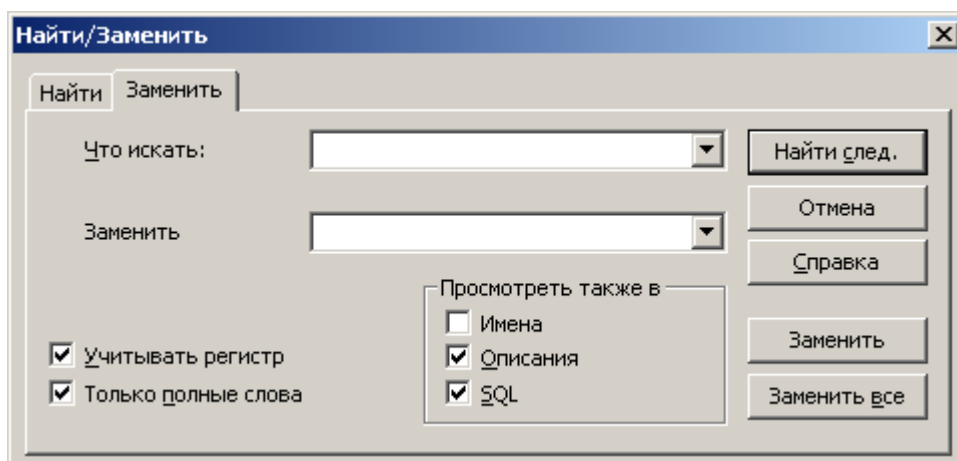


3. Введите символ или строку в текстовом поле [Найти](#).
4. Выберите или очистите текстовые поля параметров поиска.
5. Нажмите кнопку [Найти далее](#).
Если на панели юниверса найдены символ или строка, объект выделяется. При обнаружении экземпляра в описании объекта или определение SQL окно свойств объекта открывается автоматически, а символ или строка выделяется.
6. Нажмите кнопку [Найти далее](#), чтобы найти другой пример поиска строки.
7. Нажмите [Отмена](#), чтобы закрыть окно [Найти и Заменить](#).

3.11.1.3 Поиск и замена в юниверсе

Чтобы найти и заменить символ или строку в юниверсе, выполните следующие действия:

1. Выберите ► [Редактировать](#) ► [Заменить далее](#) ►.
Отобразится окно [Найти и Заменить](#).
2. Введите символ или строку в текстовом поле [Найти](#).



3. Введите символ или строку в поле [Заменить](#). Экземпляр в поле [Найти](#) будет заменен элементом текста, введенным в поле "Заменить".
4. Выберите или очистите текстовые поля параметров поиска.
5. Нажмите [Заменить](#), если требуется заменить элемент текста при каждом нахождении экземпляра. Или
Нажмите [Заменить все](#) для автоматической замены всех экземпляров в юниверсе.
При замене отдельных элементов окно свойств объекта открывается и становится активным автоматически, когда элемент отображается в описании объекта. Чтобы продолжить поиск щелкните окно [Найти и заменить](#).

3.11.2 Использование быстрого поиска

Можно найти активную панель путем ввода первой буквы строки поиска в окне поиска в нижней части панели "Юниверс".

Если панель юниверса активна, то поиск выполняется в именах класса и объекта.

Если панель структуры активна, то поиск выполняется в именах таблицы.

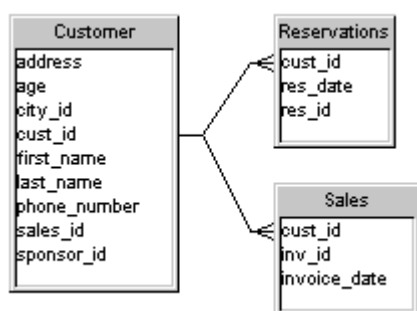
3.12 Организация отображения таблицы

Данный раздел описывает графические функции, которые можно использовать для организации и управления таблицами на панели структуры. Методология дизайнера, которая используется для

успешного создания дизайнера схемы, на панели [Структура](#), описана в главе [Создание схемы с помощью таблиц и объединений](#) [страница 132].

3.12.1 Каким образом отображаются таблицы?

На панели [Структура](#) таблицы графически представляются в виде прямоугольных символов. Имя таблицы выводится в полоске в верхней части прямоугольника. Список элементов внутри прямоугольника представляет столбцы таблицы. Линии, соединяющие таблицы, – это объединения.



3.12.2 Управление таблицами

Для управления таблицами на панели [Структура](#) можно выполнять следующие действия.

3.12.2.1 Выбор таблиц

Выбрать таблицу можно следующим образом.

Чтобы выбрать	выполните следующие действия...
одну таблицу	Щелкните таблицу.
несколько таблиц	<ul style="list-style-type: none">Вычертите границу выделения вокруг таблицы, удерживая левую кнопку мыши.Выберите несколько таблиц, удерживая клавишу SHIFT.
все таблицы одновременно	Выберите пункт меню Правка > Выделить все .

Для отмены выделения выведите курсор за пределы таблиц и сделайте щелчок.

3.12.2.2 Удаление таблиц

Чтобы удалить таблицу, выполните следующие действия.

1. Выберите таблицу.
2. Выполните любое из следующих действий.
 - Нажмите кнопку **Вырезать** на **стандартной** панели инструментов.
 - Выберите пункт меню **Правка > Вырезать**.
 - Нажмите клавишу **Delete**.

3.12.3 Использование режима списка

Можно использовать режим списка для вывода списка таблиц, объединений и контекстов, используемых в активном юниверсе. В режиме списка в средстве создания юниверсов выше панели **Структура** добавляются три панели. Эти панели называются **Таблицы**, **Объединения** и **Контексты**.

Режим списка можно использовать следующими способами.

Действие	Результат
Щелкните компонент списка в любой из панелей режима списка.	Компонент будет подсвечен на панели Структура .
Выберите таблицу, объединение или контекст на панели структуры.	Соответствующий компонент списка на панели Список будет подсвечен.
Дважды щелкните имя таблицы на панели таблиц.	Откроется окно "Переименование таблицы". Можно переименовать таблицу и в зависимости от базы данных изменить владельца таблицы и ее классификатор.
Дважды щелкните название объединения на панели Объединения .	Для данного объединения откроется окно Редактирование объединений . Можно отредактировать свойства объединения.
Дважды щелкните имя контекста на панели Контексты .	Откроется окно Редактирование контекста . Можно добавлять объединения к выбранному контексту, удерживая клавишу CTRL и выбирая объединения в списке.
Выберите компонент, затем щелкните треугольник между двумя панелями Список .	Отобразятся компоненты соседней панели списка, связанные с исходным компонентом. Все несвязанные компоненты отфильтровываются.
Щелкните разделительную линию между панелью Список и панелью Структура , а затем переместите ее вверх или вниз.	Размер панели Список будет увеличиваться или уменьшаться в зависимости от направления перемещения линии.

3.12.3.1 Использование треугольников между панелями для фильтрации компонентов списка

Маленькие треугольники, которые отображаются между панелями, действуют как фильтры при отображении компонентов. Например:

- Выберите имя таблицы на панели [Таблицы](#), а затем щелкните треугольник, указывающий на панель [Объединения](#). На панели объединений теперь отображаются только объединения выбранной таблицы.
- Выберите имя объединения на панели [Объединения](#), а затем щелкните треугольник, указывающий на панель таблиц. На панели [Таблицы](#) теперь отображаются только таблицы, связанные объединением.

3.12.3.2 Возвращение к нормальному режиму просмотра из режима списка

Режим [Список](#) можно выключить и вернуться к обычному просмотру двумя способами.

- Находясь в режиме списка, выберите пункт меню [Вид > Режим списка](#).
- Находясь в режиме списка, нажмите кнопку [Режим списка](#).

3.12.4 Автоматическое упорядочивание таблиц

Таблицы на панели структуры можно автоматически упорядочивать двумя способами.

- Выберите пункт меню [Вид > Упорядочить таблицы](#).
- Нажмите кнопку [Упорядочить](#).

3.12.5 Изменение режима отображения таблицы

Существует возможность отображать таблицы в трех различных представлениях. Каждый тип представления действует как фильтр на количество информации, показываемое в символе таблицы.

Каждое представление описывается следующим образом.

Представление таблицы	Описание
По умолчанию	Количество столбцов отображаемой таблицы может достигать восьми. Это значение можно изменять. Дополнительные сведения см. в разделе Выбор параметров отображения схемы [страница 61] .

Представление таблицы	Описание
Только имя	В символах таблиц отображаются только имена таблиц. Это снижает вероятность возникновения беспорядка на панели Структура при работе со множеством таблиц.
Столбцы объединения	В символе каждой таблицы показываются только столбцы, участвующие в объединении. Это обычно ключевые столбцы.

Каждое представление таблицы выводится следующим образом.

3.12.5.1 Представление таблицы по умолчанию

Символ таблицы с первыми восемью столбцами показан ниже.

Customer	
address	C
age	N
city_id	N
cust_id	N
first_name	C
last_name	C
phone_number	C
sales_id	N
...	

Многоточие (...) появляется после последнего столбца, когда количество столбцов в таблице превышает заданное по умолчанию значение. Если щелкнуть таблицу один раз, появляется полоса прокрутки. Можно расширять таблицу путем перетаскивания ее нижней границы вниз.

3.12.5.2 Представление в режиме имени таблицы

В символе таблицы можно отображать только имена таблиц, как изложено ниже.

- Дважды щелкните таблицу.

Отображается только имя таблицы.

3.12.5.3 Представление таблицы, образованной столбцами, участвующими в объединении

В символе таблицы можно выводить только столбцы из объединения, как разъясняется ниже.

- На панели [Структура](#) дважды щелкните таблицу, которая уже представлена в режиме имени. В таблице отображаются только столбцы объединения.

3.12.5.4 Изменение режима отображения для всех таблиц

Для одновременного изменения режима представления всех выбранных таблиц выполните следующие действия.

- Выберите пункт меню ► Вид ► [Изменить режим отображения таблицы](#) ►.

3.13 Выбор параметров отображения схемы

Существует возможность настройки формы и внешнего вида таблиц, столбцов, объединений и мощности множества на панели [Структура](#).


Для отображения компонентов на панели структуры используются следующие графические параметры.

Параметр	Описание
Форма объединения	Объединения могут быть представлены в виде простых линий различного типа или линий с индикаторами мощностей множеств, например, с разветвленными окончаниями или отношениями мощностей множеств.
Наилучшее положение	Будучи выбрано, объединение, связывающее две таблицы, автоматически оценивается с точки зрения лучшего отображения в левой или в правой части одной таблицы, оканчивающейся с левой или правой стороны другой таблицы, а также как имеющее наименьшую длину.
Таблицы	Таблицы могут выглядеть объемными, отображать псевдоним или количество строк. Для отображения количества строк в каждой таблице необходимо обновить отсчет строк в меню "Вид" > "Количество строк в таблице". Это описывается в разделе Изменение значения по умолчанию для количества возвращаемых строк [страница 65] .
Столбцы	Тип данных столбца может отображаться рядом со столбцом. Столбцы ключей могут быть подчеркнуты, а в символе таблицы столбцы могут располагаться в центре или у левого края.


Параметр	Описание
Количество столбцов по умолчанию	Можно ввести количество столбцов по умолчанию, которое будет отображаться в символе таблицы. Если количество столбцов в таблице будет больше, чем задано по умолчанию, в символе таблицы появится многоточие (...) в конце списка столбцов. Если щелкнуть таблицу один раз, у ее края появится линейка прокрутки.
Центрирование по выделению	Вид панели Структура определяется расчетом центральной точки.

3.13.1 Установка графических параметров для отображения панели структур

Параметры графического отображения для панели [Структура](#) устанавливаются следующим образом.

1. Выберите [Инструменты](#) [Параметры](#) .
Откроется диалоговое окно [Параметры](#).
2. Выберите вкладку [Графическое отображение](#).
Откроется страница [Графическое отображение](#). На данной странице перечисляются параметры графического отображения компонентов панели [Структура](#).
3. Выберите или введите параметры графического отображения.
4. Нажмите кнопку [ОК](#).

3.13.1.1 Примеры настройки параметров графического отображения

Ниже приводятся некоторые примеры возможного графического представления компонентов на панели структуры, для которых используются параметры графического отображения диалогового окна [Параметры](#) ([Инструменты](#) [Параметры](#) [Графическое отображение](#) ).

Имя с псевдонимом

Для выбранной таблицы с псевдонимом на панели [Структура](#) выводится как ее имя, так и имя таблицы в скобках, по отношению к которой она является производной.

"Показывать количество строк" и "Показывать формат"

Если установлен параметр *Показывать количество строк*, количество строк в каждой таблице отображается в нижней части ее символа. Чтобы обновлять количества строк для всех таблиц перед их выводом, необходимо выбрать пункт меню ► Вид ► *Количество строк в таблице* ▾.

Если выбрать пункт меню *Показывать формат*, рядом с названием столбца будет отображаться буква, представляющая тип столбца. Тип столбца обозначается следующим образом:

- С – символ;
- D – дата;
- N – число;
- T – подробный текст;
- L – блог (большой двоичный объект).

На панели *Структура* количества выводятся под нижним левым углом таблицы, а типы данных – рядом с именами столбцов.

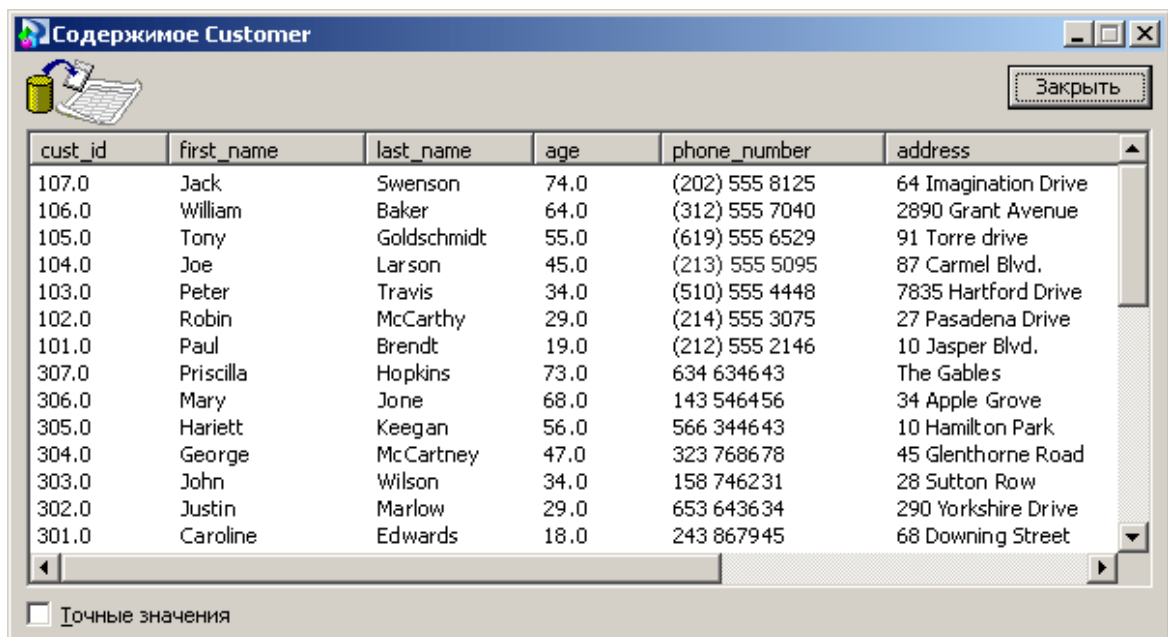
3.13.2 Просмотр значений таблиц и столбцов

В отдельной таблице или столбце можно просмотреть значения данных. По умолчанию количество строк, которые можно просмотреть в любой таблице, равно 100. Если требуется возвращать большее или меньшее количество строк, это значение можно изменить.

3.13.2.1 Просмотр значений в таблице

Чтобы просмотреть значения в таблице, выполните следующие действия.

1. Выберите таблицу на панели *Структура*.
2. Выберите пункт меню ► Вид ► *Значения таблицы* ▾.
Откроется диалоговое окно с содержимым таблицы, в котором будут выведены значения каждого столбца.



3. Если необходимо выводить только неповторяющиеся значения, установите флажок *Различные значения*.
4. Нажмите *Заккрыть*.

3.13.2.2 Просмотр значений столбца

При просмотре значений столбцов можно увеличить изображение столбцов, выбрав пункт меню ► Вид ► Увеличить масштаб ▢. Это облегчает выбор столбца.

Значения отдельного столбца можно просмотреть следующим образом.

1. Поместите указатель мыши над столбцом таблицы на панели *Структура*.
Указатель примет форму руки.
2. Щелкните столбец правой кнопкой мыши и выберите в контекстном меню команду *Просмотреть значения столбца*.
Откроется окно с содержимым столбца, в котором будут выведены его значения.



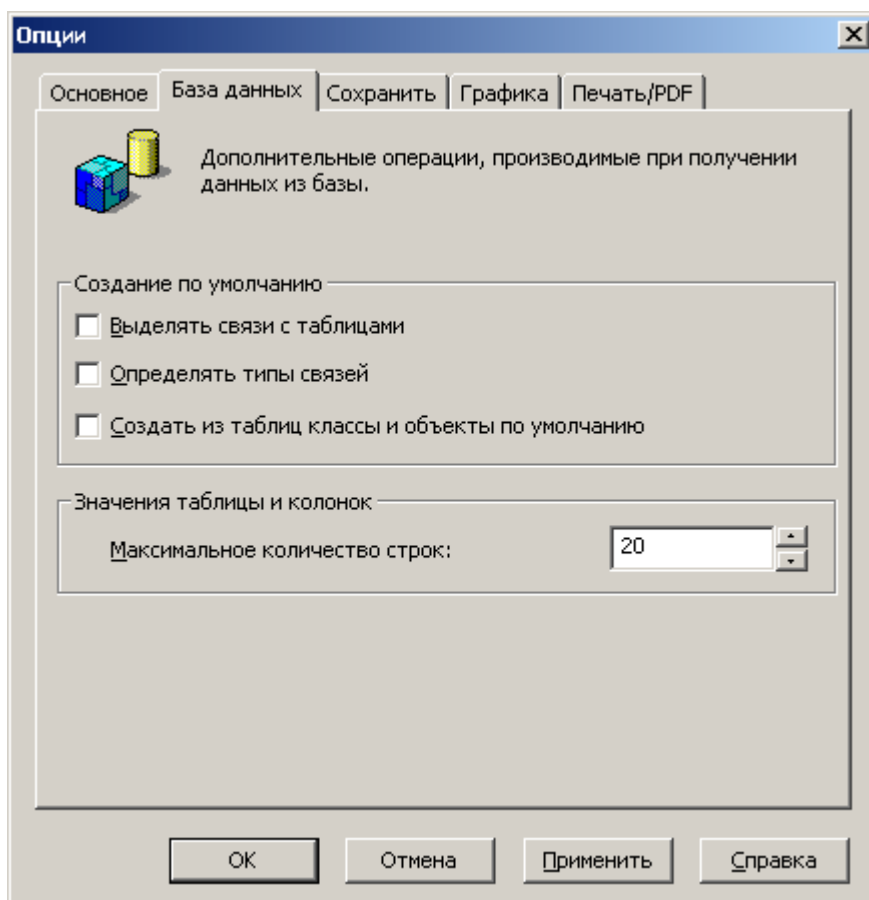
3. Если необходимо выводить только неповторяющиеся значения, установите флажок [Различные значения](#).
4. Нажмите [Заккрыть](#).

3.13.2.3 Изменение значения по умолчанию для количества возвращаемых строк

Во время просмотра значений таблиц или столбцов можно изменять значение по умолчанию для количества возвращаемых строк. Это может оказаться полезным, если требуется просмотреть небольшую выборку значений таблицы, благодаря чему можно уменьшить количество возвращаемых значений.

Чтобы изменить количество строк в выборке из таблицы, выполните следующие действия.

1. Выберите ► [Инструменты](#) ► [Параметры](#) ►
Откроется диалоговое окно [Параметры](#).
2. Щелкните вкладку [База данных](#).
Откроется страница [База данных](#).
3. Введите или выберите количество с помощью стрелок "вверх" и "вниз" в поле списка [Значения таблиц и столбцов](#).
Ниже на странице [База данных](#) установлено, что при просмотре таблицы или столбца возвращается 20 строк.



4. Нажмите кнопку *OK*.

3.13.3 Просмотр количества строк в таблицах базы данных

В каждой таблице можно выводить количество строк. Это делается в два этапа.

- Установите параметр графического отображения *Показывать количество строк* (► *Инструменты* ► *Параметры* ► *Графическое отображение* ►),
- Обновите количество строк для всех таблиц, выбрав пункт меню ► *Вид* ► *Количество строк в таблице* ►.

В каждой таблице базы данных можно выводить количество строк или установить фиксированное количество строк для выбранной таблицы с целью оптимизации производительности запросов. Это дает возможность управлять порядком таблиц в утверждении FROM на основе веса таблицы. Это описывается в разделе [Изменение показаний счетчика строк таблицы \[страница 68\]](#).

❗ Примечание

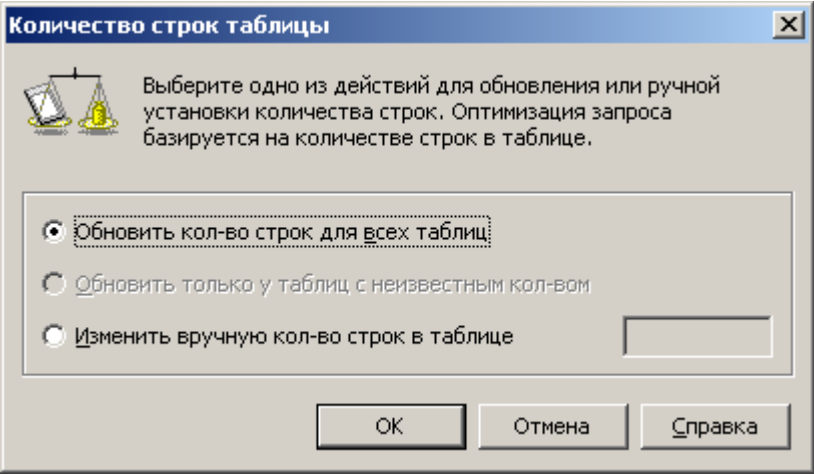
Вывод количества строк в таблице – это не то же самое, что установка количества строк, возвращаемых для просмотра значений столбцов или таблицы.

3.13.3.1 Отображения количества строк в таблицах

Чтобы отобразить количество строк в каждой таблице,

1. Выберите ► *Инструменты* ► *Параметры* ►
Откроется диалоговое окно *Параметры*.
 2. Выберите вкладку *Графическое отображение*.
Откроется страница *Графическое отображение*.
 3. Установите флажок *Отобразить счетчик строк*.
 4. Нажмите кнопку *ОК*.
 5. Выберите одну или несколько таблиц.
Или
Щелкните панель *Структура* и выберите ► *Правка* ► *Выделить все* ►, чтобы выбрать все таблицы на панели структуры.
- 📌 **Примечание**

Щелкнув панель *Структура*, можно активировать элементы меню, которые относятся к компонентам на панели *Структура*. Если не щелкнуть панель *Структура* перед выбором элемента меню, доступными будут только те элементы меню, которые применяются к панели *Юниверс*.
6. Выберите ► *Вид* ► *Количество строк в таблице* ►.
Отобразится диалоговое окно *Счетчик строк таблиц*.



Ниже описаны параметры данного диалогового окна.

Действие	Описание
Обновить счетчик строк для всех таблиц	Обновляет индикатор счетчика строк для выбранных таблиц или для всех таблиц на панели структуры.

Действие	Описание
Обновить только у таблиц с неопределенным количеством строк	Отображает счетчик строк ранее невыбранных таблиц. В результате счетчик строк отображается у всех таблиц на панели Структура .
Изменить вручную показания счетчика строк таблицы	Позволяет изменить показания счетчика строк выбранных таблиц или всех таблиц на панели структуры. Введите новое значение в текстовом поле рядом с параметром. Данный параметр не используется для оптимизации запросов. Эта тема описана в следующем разделе.

- Нажмите переключатель [Обновить показания счетчика во всех таблицах](#).
- Нажмите кнопку [ОК](#).
Счетчик строк каждой выбранной таблицы отобразится в нижнем левом углу символа таблицы на панели [Структура](#).

3.13.3.2 Изменение показаний счетчика строк таблицы

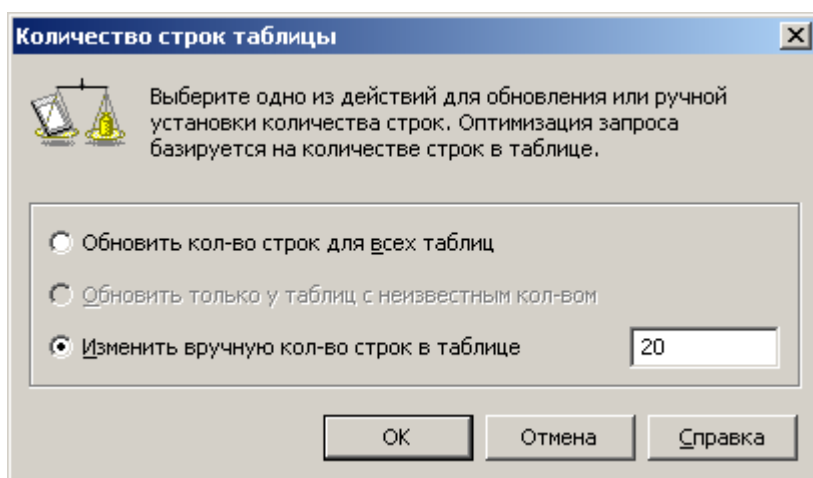
Можно изменять показания счетчиков строк таблиц. Для этого существует две причины.

Изменить показания счетчика строк на...	Описание
Оптимизировать запросы	<p>Оптимизация запросов основана на порядке таблиц в утверждении FROM сгенерированного SQL-выражения. Таблицы с большим количеством строк отображаются перед таблицами с меньшим количеством строк. Данный порядок важен для RDBMS, которой не хватает функции средства оптимизации.</p> <p>Путем изменения количества строк в таблицах можно изменять их порядок в утверждении FROM.</p>
Настройте счетчик строк для последующих изменений в пропускной способности данных.	Можно изменять показания счетчика строк таблицы, если он не влияет на количество строк, которые должны храниться в таблице. Например, можно работать с текстовой таблицей, у которой показания счетчика строк равны 100, даже если в таблице будет содержаться 50 000 строк.

Чтобы изменить показания счетчика строк одной или нескольких таблиц,

- Выберите [Инструменты](#) [Параметры](#).
Откроется диалоговое окно [Параметры](#).
- Выберите вкладку [Графическое отображение](#).
Откроется страница [Графическое отображение](#).

3. Установите флажок *Отобразить счетчик строк*.
4. Нажмите кнопку *ОК*.
5. Выберите одну или несколько таблиц.
Или
Щелкните панель *Структура* и выберите ► *Правка* ► *Выделить все* ►, чтобы выбрать все таблицы на панели структуры.
6. Выберите ► *Вид* ► *Количество строк в таблице* ►.
Отобразится диалоговое окно *Счетчик строк таблиц*.
7. Нажмите переключатель *Изменить показания счетчика таблиц вручную*.
8. Введите количество строк, которое необходимо отобразить в таблице.



9. Нажмите кнопку *ОК*.
Счетчик строк каждой выбранной таблицы отобразится в нижнем левом углу символа таблицы на панели *Структура*.





3.14 Печать юниверса

Средство создания юниверсов поддерживает все стандартные функции печати Windows. На панели *Структура* можно распечатать схему, списки таблиц, столбцы и объединения. Можно управлять путем компонентов и сведениями на распечатанной странице.

📌 Примечание




Можно распечатать версию PDF определения юниверса и схему с помощью сохранения юниверса в качестве файла PDF и печати данного файла. Дополнительные сведения см. в разделе [Сохранение определения юниверса в формате PDF \[страница 47\]](#).

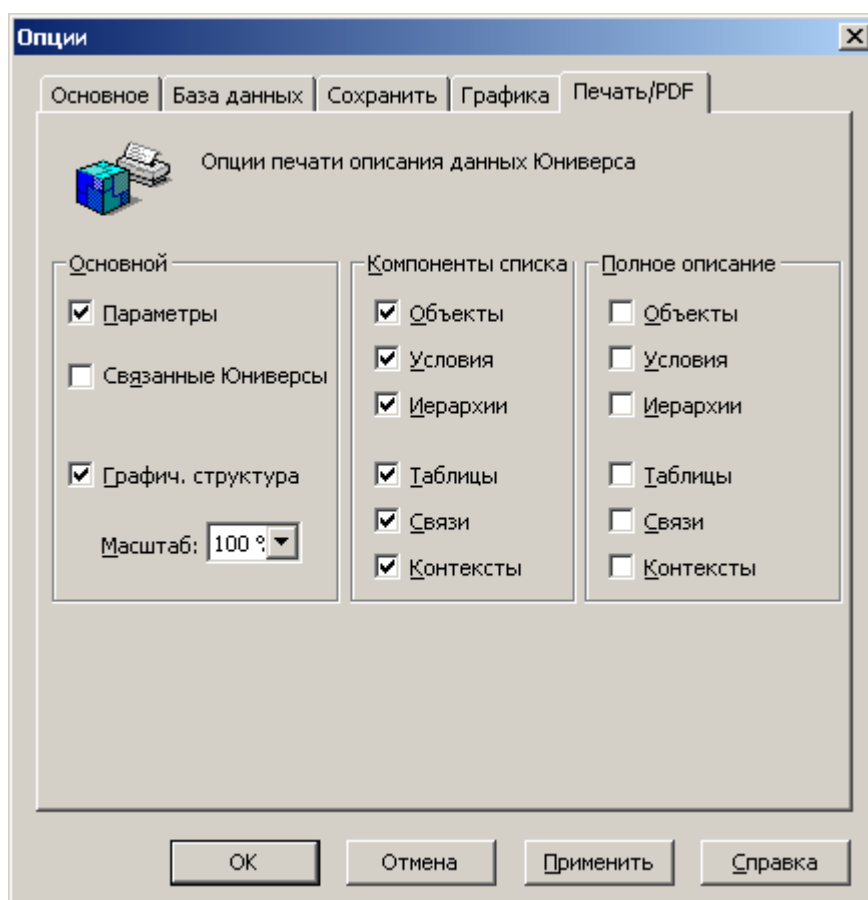
3.14.1 Установка параметров печати

Можно выбрать параметры печати на странице [Печать](#) диалогового окна [Параметры](#)  [Сервис](#)  [Параметры](#)  [Печать](#) . В установленных параметрах печати также применяются параметры, которые сохранены в файле PDF во время сохранения определения юниверса в качестве PDF. Можно задать следующие параметры печати и PDF.

Параметр "Печать"	Выводит на печать...
Общая информация	Сведения о: <ul style="list-style-type: none">• Параметрах юниверса• Связанных юниверсах Графическая структура схемы на панели Структура . Выбор масштаба графического отображения.
Списки компонентов	Списки компонентов юниверса, сгруппированных по одному или по нескольким типам: объекты, условия, иерархии, таблицы, объединения и контексты.
Описания компонентов	Описания компонентов: описания объектов, условий, иерархий, таблиц, объединений и контекстов в юниверсе. Описание включает подробные сведения о свойствах компонента. Сведения об объекте могут включать определение SQL, квалификацию и уровень защиты доступа.

Чтобы установить параметры печати для юниверса,

1. Выберите  [Инструменты](#)  [Параметры](#) .
Откроется диалоговое окно [Параметры](#).
2. Выберите вкладку [Печать/PDF](#).
Отобразится страница [Печать](#).



3. Установите флажки параметров печати, где необходимо.
4. Нажмите кнопку **OK**.

3.14.1.1 Указание параметров настройки страницы

Чтобы указать параметры настройки страницы,

1. Выберите ► **Файл** ► **Настройка страниц** .
Откроется лист **Настройка страниц**.
2. Выберите или введите параметры настройки страницы.
3. Нажмите кнопку **OK**.

3.14.1.2 Использование предварительного просмотра печати

Можно просмотреть юниверс перед печатью двумя способами.

- Выберите ► **Файл** ► **Предварительный просмотр печати** .

- Нажмите кнопку *Предварительный просмотр печати*.

3.14.1.3 Печать юниверса

Можно напечатать юниверс двумя способами.

- Выберите ► *Файл* ► *Печать* ▾.
- Нажмите кнопку *Печать*.

4 Создание юниверса и настройка его параметров




Перед построением юниверса необходимо создать новый файл юниверса.

При создании файла юниверса необходимо указать параметры соединения, которые используются для обеспечения доступа юниверса к ПО среднего яруса базы данных. Также можно указать другие параметры, определяющие способ, которым средство создания юниверсов создает объекты, связи от текущего юниверса к другим юниверсам и ограничения запроса.

Новый юниверс сохраняется как файл .unv. Новый юниверс не содержит классы и объекты. Они создаются в процессе разработки юниверса путем разработки схемы таблиц и последующего создания объектов, отображающих структуру баз данных.

4.1 Что такое параметры юниверса?

Параметры юниверса – это определения и ограничения, которые определяются для юниверса, идентифицируют сам юниверс и его соединения с базами данных, указывают типы запросов, выполняемых с помощью данного юниверса, а также устанавливают элементы управления для использования системных ресурсов.

Параметры юниверса устанавливаются в диалоговом окне [Параметры юниверса](#)  [Файл](#)  [Параметры](#)  во время создания юниверса. Соединение с базами данных является единственным параметром, который необходимо устанавливать вручную или создавать при создании нового юниверса.

Эти параметры можно изменить в любое время. Можно задать следующие параметры юниверса.

Параметр	Описание
Определение	Имя юниверса, описание, параметры соединения и информация. Это параметры, которые определяют юниверс. Информацию об определении и изменении данных параметров см. в разделе Определение юниверса [страница 77] .
Сводная информация	Сведения о версии и редакции, комментарии дизайнеров и статистики юниверса. Информацию об определении и изменении данных параметров см. в разделе Просмотр и ввод сводной информации [страница 76] .

Параметр	Описание
Стратегии	Определяет стратегии, используемые юниверсом. Стратегия – это скрипт, используемый для извлечения структурной информации из базы данных. Информацию об определении и изменении данных параметров см. в разделе Выбор стратегий [страница 88] .
Средства управления	Определяет набор ограничений на использование системных ресурсов. Информацию об определении и изменении данных параметров см. в разделе Определение средств управления ресурсами [страница 92] .
SQL	Определяет типы запросов, которые конечный пользователь может использовать из панели запросов. Информацию об определении и изменении данных параметров см. в разделе Определение средств управления ресурсами [страница 92] .
Связи	Определяет настройки, определенные для соединенных юниверсов. Информацию об определении и изменении данных параметров см. в разделе Ввод параметров ограничения SQL [страница 96] .

4.2 Создание нового юниверса

Ниже описывается процедура создания нового юниверса из рабочей области путем определения его параметров и последующего его сохранения. Данная процедура содержит обзор всех страниц, доступных в диалоговом окне параметров.

Подробную информацию по каждому шагу см. в разделе о соответствующем параметре данной главы.

Определение всех параметров в процессе создания юниверса может оказаться ненужным. Необходимо выбрать какое-либо соединение, но можно принять значения по умолчанию для других параметров, а затем при необходимости изменить их надлежащим образом.

4.2.1 Создание нового юниверса из рабочей области

Для создания нового юниверса из рабочей области выполните следующие действия.

1. Выберите **Файл** > **Создать**.

Откроется диалоговое окно [Параметры юниверса](#) на странице [Определения](#). Дополнительную информацию о данной странице см. в разделе [Определение юниверса \[страница 77\]](#).

Примечание

При выборе вариантов параметров юниверса параметр [Нажмите здесь для выбора юниверса хранимых процедур](#) не активен. Его нельзя выбрать или отменить. Чтобы изменить тип создаваемого юниверса, нажмите кнопку [Отмена](#) и начните сначала.

- Введите имя и описание для юниверса.

Предупреждение

Не используйте в имени юниверса следующие специальные символы: \ / : * ? " < > | &

- Выберите соединение в раскрывающемся списке [Соединение](#).

Или

- Нажмите кнопку [Создать](#), если нужно определить новое соединение, которое отсутствует в раскрывающемся списке. Информацию об определении нового соединения см. в разделе [Изменение параметров идентификации юниверсов](#) [страница 78].

- Выберите вкладку [Сводка](#).

Появится страница [Сводка](#). Дополнительную информацию о данной странице см. в разделе [Просмотр и ввод сводной информации](#) [страница 76].

- Введите сведения о юниверсе в поле [Комментарии](#).

- Выберите вкладку [Стратегии](#).

Откроется страница [Стратегии](#). На ней выводятся стратегии, доступные для конкретного источника связанных данных. Дополнительную информацию о данной странице см. в разделе [Выбор стратегий](#) [страница 88].

- Выберите стратегию из каждого выпадающего списка "Объекты", "Объединения" и "Таблицы". В зависимости от версии реляционной СУБД, используемой для установления соединения, в каждом выпадающем списке может иметься более одной стратегии.

- Выберите вкладку [Элементы управления](#).

Откроется страница [Элементы управления](#). Дополнительную информацию о данной странице см. в разделе [Определение средств управления ресурсами](#) [страница 92].

- Установите или снимите флажки в окне группы [Ограничения запросов](#).
- Введите значения для установленных флажков.

- Выберите вкладку [SQL](#).

Появится страница SQL. Дополнительную информацию о данной странице см. в разделе [Определение ограничений SQL](#) [страница 94].

- Установите или снимите флажки надлежащим образом.

- Выберите вкладку [Связи](#), если нужно связать новый юниверс с уже существующим.

Откроется страница "Связи". Дополнительную информацию о данной странице см. в разделе [Ввод параметров ограничения SQL](#) [страница 96].

- Чтобы выбрать юниверс, который нужно связать с новым юниверсом, нажмите кнопку [Добавить связь](#).

- Выберите вкладку [Параметры](#).

Отобразится страница [Параметры](#). Она содержит список параметров SQL, которые можно установить для оптимизации создания SQL. Дополнительную информацию о данной странице см. в разделе [Установка параметров генерации SQL](#) [страница 96].

- Нажмите кнопку [ОК](#).

В средстве создания юниверсов открываются панели юниверса и структуры.

- Выберите [Файл](#) > [Сохранить](#).

- Введите имя для файла юниверса.
- Нажмите кнопку [Сохранить](#).

4.3 Просмотр и ввод сводной информации

На странице [Сводка](#) представлена административная информация о юниверсе. Она может быть полезна для отслеживания развития активного юниверса.


На странице [Сводка](#) отображаются следующие сведения.

Информация	Описание
Создан	Дата создания юниверса и имя создателя.
Изменен	Дата последнего изменения и имя пользователя, который произвел изменение.
Редакция	Номер редакции, который показывает, сколько раз данный юниверс экспортировался в CMS.
Комментарии	Сведения о юниверсе, внесенные пользователем для собственных целей или для других дизайнеров. Эта информация доступна только в средстве создания юниверсов. Информацию о юниверсе необходимо вносить в поле Описание на странице Идентификация .
Статистика	Количество классов, объектов, таблиц, псевдонимов, объединений, контекстов и иерархий, которые содержатся в данном юниверсе.

4.4 Настройка параметров юниверса

Параметры юниверса можно настроить в следующих целях:

- [Определение юниверса \[страница 77\]](#)
- [Изменение параметров идентификации юниверсов \[страница 78\]](#)
- [Просмотр и ввод сводной информации \[страница 76\]](#)
- [Выбор стратегий \[страница 88\]](#)
- [Определение ограничений SQL \[страница 94\]](#)
- [Ввод параметров ограничения SQL \[страница 96\]](#)
- [Установка параметров генерации SQL \[страница 96\]](#)

Каждый тип параметра содержится на странице диалогового окна [параметров](#)  [Файл](#)

 [Параметры](#) . Все группы параметров описаны в соответствующих разделах ниже.

4.4.1 Определение юниверса

Каждый юниверс определяется следующими параметрами.

Идентификатор	Используется
Имя файла (8 символов)	Файловая система и Web Intelligence для ссылки на юниверс.
Полное имя (35 символов)	Пользователи Web Intelligence.
Описание	Пользователи Web Intelligence.
Уникальный числовой идентификатор	CMS для определения юниверса. Этот номер приписывается юниверсу при первом экспорте в CMS.

Имя и параметры описания определяются при создании юниверса на странице [Определения](#) диалогового окна [Параметры юниверса](#). Параметры идентификации юниверса можно изменить в любой момент.

Соединение базы данных также определяется на этой странице.

Дополнительные сведения об определении нового соединения см. в разделе [Изменение параметров идентификации юниверсов](#) [страница 78].

Для юниверса можно определить следующие параметры идентификации.

Параметры идентификации	Описание
Имя	Имя юниверса Определяет юниверс пользователям Web Intelligence. Символы имени, поддерживаемые реестром определяются администратором. Поддержка символа зависит от RDBMS. <div>⚠ Предупреждение Не используйте в имени юниверса следующие специальные символы: \ / : * ? " < > &</div>
Описание	Описание цели и содержимого юниверса. Дополнительное поле. Это описание доступно для просмотра пользователям Web Intelligence. Поэтому сведения в этом поле предоставляют полезную информацию о роли юниверса.
Соединение	Именованное множество параметров определяет, как Web Intelligence получает доступ к данным в файле базы данных. В окне раскрывающегося списка соединения отображаются все доступные соединения. Также можно создать новые соединения.

4.4.1.1 Изменение параметров идентификации универсов

Для изменения параметров идентификации универса

1. Выберите ► [Файл](#) ► [Параметры](#) .
Или
Выберите [Параметры универса](#) на панели инструмента.
На странице [Определения](#) открывается диалоговое окно [Параметры универса](#).
2. Введите имя и описание.

⚠ Предупреждение

Не используйте в имени универса следующие специальные символы: \ / : * ? " < > | &

3. Выберите соединение в раскрывающемся списке [Соединение](#).
4. Для проверки доступности соединения нажмите кнопку [Проверить](#).
Если получено сообщение о том, что сервер не отвечает, соединение является недопустимым.
Параметры соединения можно изменить, нажав [Редактировать](#) и изменив свойства соединения.
Если ошибка не устраняется, см. раздел документации RDBMS, связанный с сообщениями об ошибках.
5. Нажмите кнопку [ОК](#).

4.4.2 Определение и редактирование соединений

Соединение – это именованное множество параметров, который определяет, как приложение BusinessObjects получает доступ к данным в файле базы данных. Ссылки для соединения с Web Intelligence для ПО промежуточного слоя. Необходимо соединение для доступа к данным.

Для создания универса необходимо выбрать или создать соединение. Соединение можно изменять, удалять или заменять в любое время.

📘 Примечание

Объекты соединения имеют дополнительные права "Локальная загрузка соединения", определяемые администратором. Администратору необходимо определить связанный с соединением уровень защиты, т. е. пользователей, которые могут локально загружать значимую информацию о соединении.

📘 Примечание

Для получения полной информации по созданию, изменению и оптимизации соединений смотрите руководство по доступу к данным.

Управлять соединениями можно следующим образом:

Как управлять соединением	Описание
Воспользуйтесь панелью Панель соединений .	Посредством команды меню Инструменты Соединения . На этой панели отображается список доступных соединений (персональных, совместных и защищенных) в CMS, на который выполнен вход. Если средство создания юниверсов запущено в автономном режиме, отображаются только персональные и совместные соединения. Соединения могут отображаться в виде одноуровневого списка или иерархии, в которой соединения показываются в подпапках. На этой странице можно удалять, редактировать и создавать новые соединения.
Перейдите в диалоговое окно Параметры юниверса .	На странице Определение диалогового окна Параметры юниверса (Файл Параметры Определение). При отсутствии соединения, подходящего для текущего юниверса, можно создать новое соединение. Соединения также можно изменять.

Соединение содержит три составляющие:

- Драйвер доступа к данным
- Параметры соединения и входа
- Тип соединения

Каждая составляющая описывается в следующих разделах:

4.4.2.1 Управление папками соединений

В средстве создания юниверсов можно управлять папками соединений. Папки соединений отображаются во всех расположениях, в которых отображается или администрируется защищенное соединение. Управлять соединениями можно следующим образом:

Как управлять соединением	Описание
Панель соединений	Посредством команды меню Инструменты Соединения . На этой панели отображается список доступных соединений (персональных, совместных и защищенных) в CMS, на который выполнен вход. Если средство создания юниверсов запущено в автономном режиме, отображаются только персональные и совместные соединения. Соединения могут отображаться в виде одноуровневого списка или иерархии, в которой соединения показываются в подпапках. На этой странице можно удалять, изменять и создавать новые соединения. После проверки любые действия, выполняемые на Панели соединений , автоматически фиксируются на сервере CMS.
Диалоговое окно Параметры юниверса	На странице Определение диалогового окна Параметры юниверса (Файл Параметры Определение).

Как управлять соединением	Описание
	При отсутствии соединения, подходящего для текущего юниверса, можно создать новое соединение. Соединения также можно изменять.

Возможные действия зависят от прав пользователя. При отсутствии ограничений доступны следующие действия:

- Создать новое соединение
- Создать папку нового соединения
- Изменить соединение или папку
- Удалить соединение или пустую папку
- Переименовать соединение или папку
- Изменить описание соединения
- Просмотреть свойства соединения
- Вырезать, скопировать и вставить в режиме изменения.

4.4.2.2 Драйвер доступа к данным

Драйвер доступа к данным – это уровень программного обеспечения, соединяющий юниверс с промежуточным ПО.

Драйверы доступа к данным входят в комплект поставки продукции Business Objects. Существует свой драйвер доступа к данным для каждого поддерживаемого программного обеспечения среднего яруса. В процессе установки средства создания юниверсов ключ доступа к данным определяет, какие драйверы для доступа к данным установлены.

При создании нового соединения необходимо выбрать подходящий драйвер Доступа к данным для программного обеспечения среднего яруса RDBMS, необходимый для соединения с целевым RDBMS.

4.4.2.3 Параметры соединения и входа

Необходимо настроить драйвер доступа к данным, указав следующие параметры соединения и входа.

Параметр	Описание
Тип	Тип соединения: персональное, общедоступное или без-опасное.
Имя	Определение имени для соединения.
Имя пользователя	Имя пользователя базы данных. Оно обычно присваивается пользователю администратором базы данных.

Параметр	Описание
Пароль	Пароль пользователя для базы данных. Оно обычно присваивается пользователю администратором базы данных.
Для обновления отчетов во время просмотра используйте единый вход	Будучи выбраны, имя пользователя и пароль, используемые для доступа к репозиторию, автоматически используются в качестве параметров входа в базу данных. Для получения дополнительных сведений о настройке единого входа см. Руководство администратора BusinessObjects Enterprise
Используйте учетные данные базы данных, связанных с учетной записью BusinessObjects.	После выбора, необходимо ввести пароль пользователя, связанный с учетной записью BusinessObjects для обновления отчета. Выполнение этих условий установлено на уровне Центральной консоли управления. Дополнительные сведения о настройке данного параметра см. в руководстве администратора Business Objects Enterprise.
Источник данных/Служба	Источник данных или имя базы данных. При использовании драйвера ODBC имя источника данных определяет целевую базу данных. Если используется собственный драйвер, имя базы данных определяет целевую базу данных.

❗ Примечание

Можно создать соединение с помощью ODBC с файлами Excel и текстовыми файлами в формате .csv. Чтобы в приложении Web Intelligence использовать юниверсы, созданные на основе текстового файла или файла Excel, полученные с помощью ODCB, необходимо отредактировать файл msjet.prm для соединения.

Расположение файла: \$INSTALLDIR\$/BusinessObjects Enterprise 12.0/win32_x86/dataAccess/connectionserver/odbc, где \$INSTALLDIR\$ – это каталог, в который устанавливаются приложения Business Objects. Измените параметр DB_TYPE в файле msjet.prm следующим образом:

C: <Parameter Name='DB_TYPE'>MS Jet Engine</Parameter>

Ha: <Parameter Name='DB_TYPE'>MS Jet</Parameter>

По завершении изменения необходимо перезагрузить сервер Business Objects Enterprise.

Примечание. Если средство создания юниверсов используется на одном компьютере с сервером Web Intelligence, и необходимо создать дополнительные юниверсы на основе текстовых файлов или файлов Excel после изменения данного значения, то необходимо восстановить следующее значение: <Parameter Name='DB_TYPE'>MS Jet Engine</Parameter>

4.4.2.4 Тип соединения

Тип соединения определяет, кто может использовать это соединение для получения доступа к данным. Средство создания юниверсов автоматически сохраняет все соединения, созданные во время сеанса. При следующем запуске сеанса эти соединения будут доступны.

С помощью средства проектирования можно создать три типа соединений.

- Персональное
- Общее
- Защищенное

Каждый тип соединения описывается следующим образом.

Персональные соединения

Доступ к данным только для создателя юниверса на компьютере, где было создано соединение.

Параметры соединения хранятся в файле PDAC.LSI, расположенном в папке LSI Business Objects 12.0 в каталоге профиля пользователя, например:

```
C:\Documents and Settings\<имя пользователя>\Application Data\Business  
Objects\Business Objects 12.0\lsi
```

Данные параметры постоянны и не обновляются.

Персональные соединения являются незащищенными с позиции безопасности продуктов Business Objects.

Для распределения юниверсов нельзя использовать персональные соединения. Необходимо использовать персональные соединения для получения доступа к личным данным локальной машины.

Общие соединения

Разрешают доступ к данным для всех пользователей. Эти соединения являются незащищенными на основе безопасности продуктов Business Objects.

Параметры соединения хранятся в файле SDAC.LSI, расположенном в папке lsi Business Objects 12.0 в каталоге профиля пользователя, например:

```
C:\Documents and Settings\<имя пользователя>\Application Data\Business  
Objects\Business Objects 12.0\lsi
```

Защищенные соединения

- Сосредотачивают и контролируют доступ к данным. Это самый безопасный тип соединения, который используется для защиты от доступа к конфиденциальным данным.

- С помощью средства создания юниверсов можно создавать защищенные соединения.
- При распределении юниверсов через CMS необходимо использовать защищенные соединения.
- Защищенные соединения можно использовать и обновлять в любой момент.

4.4.2.5 Установка паролей для персональных и общедоступных соединений

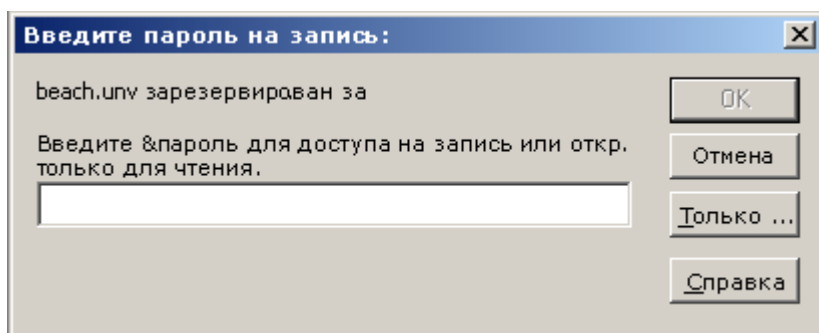
Можно установить пароль для доступа к юниверсу, обладающим типом персонального или общедоступного соединения. Пароли защищают юниверс от несанкционированных пользователей в среде, где нет репозитория.

❗ Примечание

При утрате пароля невозможно восстановить файл юниверса. Необходимо сохранить резервный файл паролей юниверса.

Можно назначить два разных параметра для пароля:

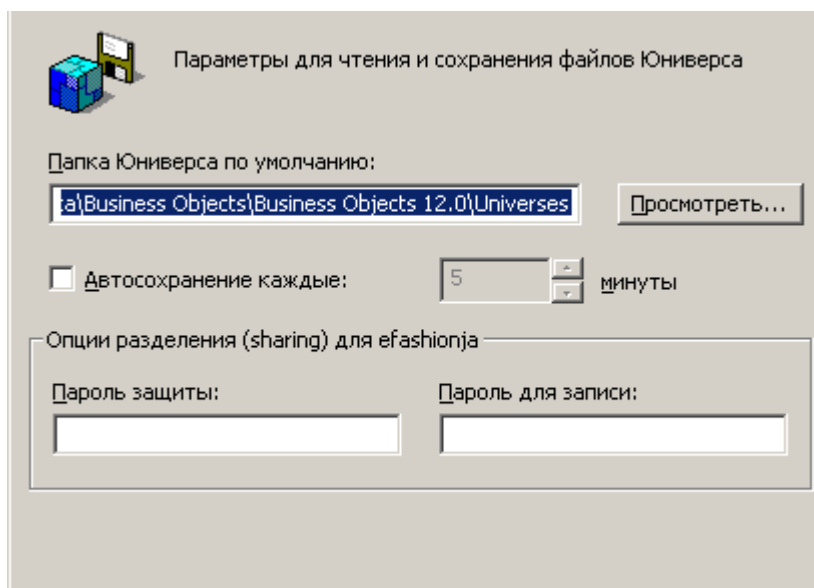
- Пароль защиты открывает диалоговое окно; он просто предлагает пользователю ввести пароль. Если пароль верный, юниверс открывается.
- Пароль для записи открывает диалоговое окно, в котором



в котором пользователь может открыть юниверс только в режиме чтения или в режиме чтения и записи после введения правильного пароля.

Для установки пароля с использованием персональных и общедоступных соединений

1. Выберите ► [Инструменты](#) ► [Параметры](#) ►
Откроется диалоговое окно [Параметры](#).
2. Выберите вкладку [Сохранить](#).
Откроется страница [Сохранить](#).



3. Введите пароль в текстовом окне *Пароль защиты* или *Пароль для записи*. Пароль может включать до 40 буквенно-цифровых символов.
4. Нажмите кнопку *OK*.

4.4.2.6 Получение доступа к базе данных после изменения пользовательского параметра DBPass

Администратор BusinessObjects может позволить продолжать получать доступ к данным под именем и паролем пользователя BusinessObjects после того, как пароль пользователя базы данных был изменен.

Если установлены следующие параметры, пользователь BusinessObjects может продолжать получать доступ к данным, не обращаясь к администратору BusinessObjects, после того как пароль базы данных был изменен.

- В *СМС* необходимо установить флажок *Активировать и обновлять учетные данные пользователя для источника данных при входе в систему*.
- В средстве создания юниверсов на странице *Задать новое соединение мастера создания соединений* необходимо установить флажки *Использовать реквизиты БД, соответствующие учетной записи пользователя Business Objects* и *Использовать службы единого входа при обновлении отчетов во время просмотра*.

При установке флажков, описанных выше, обновленные параметры DBUser и DBPass автоматически связываются с учетной записью пользователя BusinessObjects.

❗ Примечание

DBUser и DBPass являются статическими параметрами, которые необходимо обновлять в СМС. При изменении пароля базы данных рекомендуется обновить пароль для каждой учетной записи пользователя в СМС.

4.4.2.7 Определение нового соединения

Новое соединение можно определить с помощью [Мастера новых соединений](#). Запустить мастер можно следующим образом:

- Диалоговое окно [Параметры юниверса, Страница определения](#) (▢ [Файл](#) ▸ [Параметры](#) ▸ [Определение](#) ▸). Новое соединение обычно создается, когда нет существующего соединения, доступного для данных, к которым юниверсу необходимо осуществить доступ.
- Список [Список соединений](#) (▢ [Инструменты](#) ▸ [Соединения](#) ▸).

Мастер соединений можно использовать для установления расширенных и пользовательских параметров соединения. Для получения полной информации по созданию, изменению и оптимизации соединений смотрите Руководство по доступу к данным.

При создании соединения с помощью диалогового окна [Параметры юниверса](#), появляется окно [Параметры юниверса](#) с новым соединением, внесенным в поле [Подключение](#).

При создании соединения с помощью диалогового окна Подключения, соединение появится в списке.

Связанные сведения

[Запуск мастера нового соединения \[страница 438\]](#)

4.4.2.8 Просмотр доступных соединений

Все доступные соединения можно просмотреть в списке соединений. Существующие соединения можно изменять и создавать новые.

Для просмотра доступных соединений выполните следующее.

1. Нажмите ▢ [Инструменты](#) ▸ [Соединения](#). ▸
Откроется список [Соединения](#). В нем отображены все доступные соединения текущего юниверса.
2. Нажмите [Отмена](#), чтобы закрыть диалоговое окно.

Соединения можно редактировать в диалоговом окне [Соединения](#).

Защищенное соединение можно редактировать только при работе в интерактивном режиме. Персональные и общие соединения возможно редактировать в любом режиме.

Имя существующего соединения изменить невозможно.

4.4.2.9 Редактирование соединения

Чтобы отредактировать соединение, выполните следующие действия.

1. Нажмите ▢ [Инструменты](#) ▸ [Соединения](#). ▸

Откроется список [Соединения](#).

2. В списке доступных соединений выберите имя соединения.
3. Нажмите [Редактировать](#).
Откроется страница [Вход](#) для соединения.
4. При необходимости введите новый источник данных или имя базы данных в поле "Источник данных" или "Служба".
5. Внесите изменения в параметры входа в систему.
6. Нажмите кнопку [Далее](#).
Откроется страница [Выполнить проверку](#).
7. Нажмите кнопку [Проверить источник](#) для проверки измененного соединения.
8. Нажмите кнопку [Далее](#) для перехода к страницам [Дополнительно](#) и [Пользовательский](#). Измените необходимые параметры. Также можно принять существующие значения или установленные по умолчанию.
9. На странице [Пользовательский](#) нажмите кнопку [Готово](#), чтобы применить изменения к соединению.

4.4.2.10 Удаление соединения

Соединения могут быть удалены из списка соединений. Защищенное соединение можно удалить только при работе в интерактивном режиме. Персональные и общие соединения возможно удалять в любом режиме.

Чтобы удалить соединение, выполните следующие действия.

1. Нажмите ► [Инструменты](#) ► [Соединения](#).
Откроется список [Соединения](#).
2. Выберите имя соединения в списке.
3. Щелкните [Удалить](#).
Откроется окно с запросом на подтверждение.
4. Щелкните [Да](#).
Соединение будет удалено из списка.

4.4.2.11 Добавление нового соединения

Чтобы добавить новое соединение, откройте страницу [Соединения](#), выбрав ► [Инструменты](#) ► [Соединения](#) ►, нажмите кнопку [Добавить](#) и следуйте указаниям мастера [определения нового соединения](#). Полные инструкции по работе с мастером соединений приведены в разделе [Редактирование соединения \[страница 85\]](#).

Использование соединений DSN

Если в средстве дизайна информации или средстве создания юниверсов для источников данных юниверсов используются имена источников данных (DSN) ODBC, источники данных с такими же

именами должны быть созданы на физическом компьютере, где установлены серверные компоненты для приложений отчетности SAP BusinessObjects, в которых используются опубликованные юниверсы.

Использование соединений без DSN

Драйверы сервера соединений платформы SAP BusinessObjects Business Intelligence 4.3 сейчас поддерживают только 64-разрядные приложения. Чтобы использовать UDT, пользователям необходимо установить клиенты Business Objects в 64-разрядной системе вместо 32-разрядной.

Если средство создания юниверсов использует соединение без DSN, когда пользователь выбирает подключение с использованием строки соединения, он должен объявить драйвер и строку соединения в мастере соединений.

Для получения дополнительной информации о соединениях DSN и драйверах ODBC см. раздел [Установка драйверов ODBC](#) в руководстве по доступу к данным.

4.4.3 Установка параметров сводки юниверса

На странице "Сводка" представлена административная информация о юниверсе. Она может быть полезна для отслеживания развития активного юниверса.

На странице "Сводка" отображаются следующие сведения.

Информация	Описание
Создан	Дата создания юниверса и имя создателя.
Изменен	Дата последнего изменения и имя пользователя, который произвел изменение.
Редакция	Номер редакции, который показывает, сколько раз данный юниверс экспортировался в репозиторий.
Комментарии	Сведения о юниверсе, внесенные пользователем для собственных целей или для других дизайнеров.
Статистика	Количество классов, объектов, таблиц, псевдонимов, объединений, контекстов и иерархий, которые содержатся в данном юниверсе.

4.4.3.1 Для просмотра и добавления сводной информации

1. Выберите "Файл" > "Параметры".
или
Выберите средство "Параметры".

Отобразится диалоговое окно параметров юниверса.

2. Выберите вкладку "Сводка".

Появится страница "Сводка".

3. Введите комментарий в текстовое поле комментария.
4. Нажмите "ОК".

4.4.4 Выбор стратегий

Стратегия представляет собой скрипт, который автоматически извлекает структурную информацию из файла базы данных или простого файла. Стратегии выполняют две основные роли.

- Автоматическое определение объединения и мощности (Стратегии объединений)
- Автоматическое создание классов, объектов и объединений (Стратегии объектов и объединений)

Стратегии используются для автоматического создания или определения структур юниверса, основанного на структурах SQL базы данных.

📘 Примечание

Стратегии, автоматизирующие создание структур юниверсов не являются обязательным элементом при создании и дизайне юниверса. Они используются для ускорения процесса создания юниверса, позволяя использовать информацию о метаданных, уже существующих в базе данных или средстве создания баз данных. Тем не менее, при построении юниверса путем создания объектов и объединений, основанных на взаимосвязях, полученных непосредственно из анализа потребностей пользователя можно не использовать стратегию автоматического создания.

В средстве создания юниверсов можно указать два типа стратегий:

Стратегия	Описание
Встроенная стратегия	Стратегия, которая по умолчанию поставляется в комплекте со средством проектирования. Встроенные стратегии нельзя настраивать.
Внешняя стратегия	Скрипт, создаваемый пользователем и содержащий тот же тип информации, что и встроенная стратегия, но настроенный для оптимального извлечения информации из базы данных.

4.4.4.1 Выбор стратегии

Для выбора стратегии выполните следующие действия.

1. Выберите ► *Файл* ► *Параметры* .
Или
Выберите средство *Параметры*.

Откроется диалоговое окно [параметров юниверса](#).

2. Выберите вкладку [Стратегии](#).

Откроется страница [Стратегии](#).

3. Выберите стратегию из каждого раскрывающегося списка [Объекты](#), [Объединения](#) и [Таблицы](#).

4. Нажмите кнопку [ОК](#).

4.4.4.2 Использование встроенных стратегий

Встроенные стратегии – это стратегии по умолчанию, которые поставляются в комплекте со средством создания юниверсов. Существуют встроенные стратегии для всех поддерживаемых баз данных. Их нельзя изменять. В раскрывающемся списке стратегий встроенные стратегии по умолчанию отображаются выше внешних стратегий.

Встроенные стратегии используются в следующих целях.

Стратегия	Используется для...
Объекты	Автоматическое создание классов и объектов по умолчанию при создании таблиц в схеме таблиц.*
Объединения	<ul style="list-style-type: none">• Автоматическое извлечение объединений по умолчанию в случае, когда таблицы создаются в схеме таблиц.*• Автоматическая вставка мощностей множества в создание объединения.*• Автоматическое определение объединений в схеме таблиц. Если выбрать Инструменты Автоматическое определение Определять объединения, в средстве создания юниверсов используется стратегия для автоматического определения вероятных объединений. Реализацию объединения можно подтвердить или отклонить.• Автоматическое определение и вставка мощностей множества для существующих объединений в схеме таблиц. Если выбрать Инструменты Автоматическое определение Определять свойства связей, в средстве создания юниверсов используется стратегия определения свойств связей для объединений, выбранных в схеме таблицы.
Таблицы	Данные фильтрации доступны для таблиц в средстве просмотра таблиц.

* Функция автоматического создания для стратегий должна быть активирована на странице [баз данных](#) в диалоговом окне [Параметры](#).

Использование стратегии объектов

Стратегии объектов используются только для автоматического создания классов и объектов при добавлении таблицы в схему таблиц. Чтобы воспользоваться этой стратегией, активируйте ее на странице [База данных](#) диалогового окна [Параметры](#). Дополнительную информацию см. в разделе [Использование функций автоматического создания стратегии](#) [страница 90].

Использование стратегии объединений

Выбранная стратегия задает способ автоматического определения связей и объединений в схеме таблицы, который используется средством создания юниверсов.

В зависимости от базы данных в списке может находиться одна и более стратегий объединений. Например, при использовании базы данных Oracle можно указать стратегию объединения для автоматического определения объединений, основанных на соответствующих именах столбцов или соответствующих номерах столбцов.





Если стратегия не выбрана, средство создания юниверсов будет использовать стратегию объединений по умолчанию, которая предусматривает сопоставление названий столбцов для определения объединений. Нет необходимости активировать выбранную стратегию объединения для определения объединений. Она всегда используется для определения объединений или мощностей множества в схеме таблиц.

Стратегия объединения также используется для автоматического создания объединений и применения мощностей множества при создании объединений. Чтобы использовать функции автоматического создания данной стратегии по умолчанию, активируйте ее на странице [База данных](#) диалогового окна [Параметры](#). Дополнительную информацию см. в разделе [Использование функций автоматического создания стратегии](#) [страница 90].

Использование стратегии таблиц

Выбранная стратегия таблиц считывает структуру таблиц базы данных. В зависимости от стратегии можно определить тип информации, отображенной в обозревателе таблиц. Например, тип данных столбца и описания.

4.4.4.3 Использование функций автоматического создания стратегии

Функции автоматического создания и вставки для стратегий не активны по умолчанию. Для использования этих функций необходимо установить флажок [Создание по умолчанию](#) напротив соответствующей стратегии, которая будет применяться к созданию объекта или объединения. Они перечислены на странице [База данных](#) диалогового окна [Параметры](#) ( [Инструменты](#)  [Параметры](#)  [База данных](#) ).

Каждый параметр создания по умолчанию на странице [базы данных](#) имеет следующее описание.

Параметр	При сбросе	При выборе
Извлечение объединений с таблицами	Объединения необходимо создавать вручную. Если выбрать ► Инструменты ► Автоматическое определение ► Определять объединения ▾, в средстве создания юниверсов используется стратегия для определения объединений и предлагаются возможные объединения. Реализацию вероятного объединения можно подтвердить или отклонить	Получение таблиц с объединениями, связывающими их в соответствии с выбранной стратегией объединения.
Определение мощностей множества в объединениях	Мощности множества необходимо устанавливать вручную. Если выбрать ► Инструменты ► Автоматическое определение ► Определять свойства связей ▾, в средстве создания юниверсов используется стратегия для определения и применения свойств связей для выбранных объединений.	Определяет и реализует мощности множества, интегрированные в объединения при их создании.
Создание классов и объектов по умолчанию из таблиц.	Классы и объекты необходимо создавать вручную непосредственно на панели юниверса или путем перетаскивания таблицы или столбцов с панели структуры на панель юниверса .	Классы и объекты по умолчанию автоматически создаются на панели юниверса при добавлении таблицы на панель структуры . Класс соответствует имени таблицы, а объекты соответствуют именам столбцов. Все символы подчеркивания (_) заменяются пробелами.

Чтобы выбрать параметр создания по умолчанию для стратегий, выполните следующие действия.

1. Выберите ► [Инструменты](#) ► [Параметры](#) ▾.
Откроется диалоговое окно [Параметры](#).
2. Щелкните вкладку [База данных](#).
Откроется страница [База данных](#).
3. Установите флажок, соответствующий функции создания по умолчанию, для которой требуется использовать стратегию.
4. Нажмите [ОК](#).

4.4.4.4 Установка количества строк для просмотра

В диалоговом окне [Параметры базы данных](#) можно также указать максимальное количество запрашиваемых строк в каждой таблице базы данных. Нельзя ограничить физическое количество строк, извлекаемых из базы данных, поскольку это может изменить количество строк просмотра по умолчанию при просмотре значений таблицы или столбца. Данное действие применимо к строкам, возвращаемым в средстве создания юниверсов, но не применяется к запросам в Web Intelligence.

Чтобы задать количество строк для просмотра, выполните следующие действия.

- Введите значение в текстовом поле параметра [Максимальное количество полученных строк](#). Можно также нажать один или несколько раз стрелки вверх и вниз, чтобы увеличить или уменьшить значение по умолчанию (100).

4.4.4.5 Использование внешних стратегий

Внешняя стратегия представляет собой скрипт SQL, который определяется пользователем, следует за определяемой структурой вывода данных и используется для выполнения настраиваемых задач автоматического создания юниверса. Внешние стратегии хранятся в файле внешних стратегий XML (<RDBMS>.STG). Сценарии SQL в этом файле отображаются в раскрывающемся списке на странице стратегий вместе с другими стратегиями.

Внешние стратегии содержат тот же тип сведений, что и встроенные стратегии, но с некоторыми изменениями, позволяющими средству создания юниверсов получать определенный тип сведений базы данных или оптимизировать способ получения сведений из базы данных.

Дополнительную информацию об определении внешних стратегий см. в разделе [Использование внешних стратегий для настройки процесса создания юниверса](#) [страница 399].

4.4.5 Определение средств управления ресурсами

Средство создания юниверсов предоставляет ряд параметров, позволяющих контролировать использование системных ресурсов.

❗ Примечание

При просмотре данной вкладки в диалоговом окне "Предварительный просмотр ограничений" измененные параметры, применяемые к ограничению, отображаются красным.

4.4.6 Какие параметры системных ресурсов доступны?

Для системных ресурсов можно указать следующие ограничения.

Ограничения запроса	Описание
Ограничение размера результата в соответствии с указанным значением	Количество строк, возвращающихся в запросе, ограничивается указанным количеством. Таким образом, будет ограничено количество возвращающихся строк, но обработка RDBMS всех строк в запросе останется прежней. Количество строк будет ограничено, как только RDBMS начнет передавать строки.
Ограничение по времени выполнения в соответствии с указанным значением	Время выполнения запроса ограничивается количеством указанных минут. Тем самым ограничивается время передачи данных в WebIntelligence, но процесс работы над базой данных не прекращается.
Ограничение размера объектов длинного текста в соответствии с указанным значением	<p>Укажите максимальное количество символов для объектов длинного текста.</p> <p>Если флажок не установлен, параметр не активен. По умолчанию автоматически устанавливается максимальное значение (1000). Чтобы убедиться, что указываемые результаты превышают заданные по умолчанию, необходимо установить флажок и ввести значение.</p>

4.4.7 Чтобы ввести информацию для управления ресурсом, выполните следующие действия.

1. Выберите "Файл" > "Параметры".
Или
Выберите средство "Параметры".
Отобразится диалоговое окно параметров юниверса.
2. Выберите вкладку [Элементы управления](#).
Откроется страница [элементов управления](#).
3. Установите флажок в окне группы [Ограничения запроса](#).
Введите значение в текстовое поле, соответствующее выбранному параметру ограничения запроса. С помощью стрелок вверх и вниз в нижней части текстовых окон можно увеличить или уменьшить введенное значение.
4. Нажмите кнопку [OK](#).

4.4.8 Ограничение времени выполнения запросов, создающих более одного SQL-оператора

Ограничение времени, установленное для выполнения запроса – это общее время выполнения запроса. Если запрос содержит несколько SQL-инструкций, каждой инструкции отводится время выполнения,

равное общему времени выполнения запроса, деленному на количество инструкций. Таким образом, каждой инструкции отводится одинаковое время выполнения.

Если какой-либо инструкции требуется больше времени выполнения, чем другим, может произойти сбой выполнения запроса вследствие несовпадения временных пределов выполнения запроса.

При указании ограничения времени выполнения для нескольких SQL-инструкций необходимо взять стандартное время выполнения самой долгой инструкции и умножить это значение на количество инструкций в запросе.

4.4.9 Определение ограничений SQL

Можно указать элементы управления для типов запросов, которые конечные пользователи могут формулировать с помощью средства [Панель запросов](#) в приложении Web Intelligence.

Можно определить элементы управления для следующих областей создания запроса.

- Использование подзапросов, операторов и комплексных операндов в отдельных запросах.
- Создание нескольких SQL-инструкций.
- Предотвращение или предупреждение возникновения декартова произведения.

В следующих разделах описывается каждый из этих наборов управлений.

4.4.9.1 Управления запроса

Можно установить следующие управления для индивидуальных запросов.

Параметр	Описание
Разрешить использование операторов объединения, пересечения и разности	Позволяет конечным пользователям объединять запросы, используя операторов набора данных (объединение, пересечение и разность) для получения одного набора результатов.

4.4.9.2 Элементы управления несколькими SQL-инструкциями

Можно установить следующие элементы управления, чтобы определить, как обрабатываются несколько SQL-инструкций.

Параметр	Описание
Несколько SQL-инструкций для каждого контекста	Позволяет конечным пользователям создавать запросы, которые при использовании контекста содержат несколько SQL-инструкций. Выберите этот параметр при наличии любого контекста в универсе.
Несколько SQL-инструкций для каждой меры	<p>Разделяет SQL-выражение на несколько инструкций, если запрос включает объекты меры, взятые из столбцов в разных таблицах. Дополнительную информацию о данном параметре см. в разделе Использование нескольких SQL-инструкций для каждой меры [страница 246].</p> <p>Если объекты меры основаны на столбцах в той же таблице, SQL не разделен, даже если этот параметр проверен.</p>
Разрешить выбор нескольких контекстов	<p>Позволяет конечным пользователям создавать запросы для объектов в более, чем одном контексте и генерировать один набор результатов из нескольких контекстов.</p> <p>При использовании контекстов для устранения циклов, ловушек разногласий, ловушек раскрытия или любых других проблем объединения необходимо убрать флажок.</p>

4.4.9.3 Элементы управления декартовым произведением

Декартово произведение – это результирующее множество, содержащее все возможные комбинации каждой строки таблицы запроса. Декартово произведение почти всегда вычислено неправильно.

Можно установить следующие управления для производства декартового произведения.

Параметр	Описание
Предотвратить	При выборе данного параметра запрос, приводящий к декартову произведению не выполняется.
Предупредить	При выборе данного параметра появится сообщение с предупреждением, что запрос приведет к декартову произведению.

4.4.9.4 Ввод параметров ограничения SQL

Для введения параметров ограничения SQL выполните следующие действия.

1. Выберите **Файл** > **Параметры**.
Или
Выберите средство **Параметры**.
Откроется диалоговое окно **Параметры юниверса**.
2. Выберите вкладку **SQL**.
Откроется страница **SQL**.
3. Выберите или отмените параметры в групповых окнах **Запрос** и **Несколько путей**.
4. Нажмите переключатель в групповом окне **Декартово произведение**.
5. Нажмите кнопку **OK**.

4.4.10 Определение параметров для связанных юниверсов

Вкладка **Связи** используется для динамически связанных юниверсов, которые описаны в главе [Развертывание юниверсов \[страница 507\]](#).

4.4.11 Установка параметров генерации SQL

В средстве создания юниверсов можно динамически настроить определенные параметры SQL, которые являются общими для большинства реляционных СУБД, чтобы оптимизировать SQL-код, созданный в продуктах Web Intelligence, использующих юниверс.

4.4.11.1 Использование файлов параметров (PRM) в предыдущих версиях средства создания юниверсов

В версиях программы Designer, параметры генерации SQL, используемые юниверсом, поддерживались и редактировались в отдельном файле, называемом файлом параметров (PRM). Значения, заданные в файле PRM применялись ко всем юниверсам, использующим драйвер доступа к связанным данным, который был определен для соединения.

Многие из параметров SQL, которые использовались для оптимизации создания запроса, теперь управляются в отдельном файле юниверса. PRM-файл больше не используется для установки параметров создания запроса, которые можно задать в средстве создания юниверсов. Тем не менее, PRM-файлы по-прежнему используются для установки параметров, относящихся к базам данных.

❗ Примечание

Дополнительную информацию о файле PRM для драйвера доступа к данным см. в *Руководстве по доступу к данным*. Данное руководство можно открыть, выбрав ► [Справка](#) ► [Руководство по доступу к данным](#) ►.

4.4.11.2 Динамическая установка параметров SQL в средстве создания юниверсов

Многие параметры, общие для большинства поддерживаемых ПО среднего яруса RDBMS доступны для редактирования на вкладке [Параметры](#) диалогового окна параметров юниверса (► [Файл](#) ► [Параметры](#) ► [Параметр](#) ►).

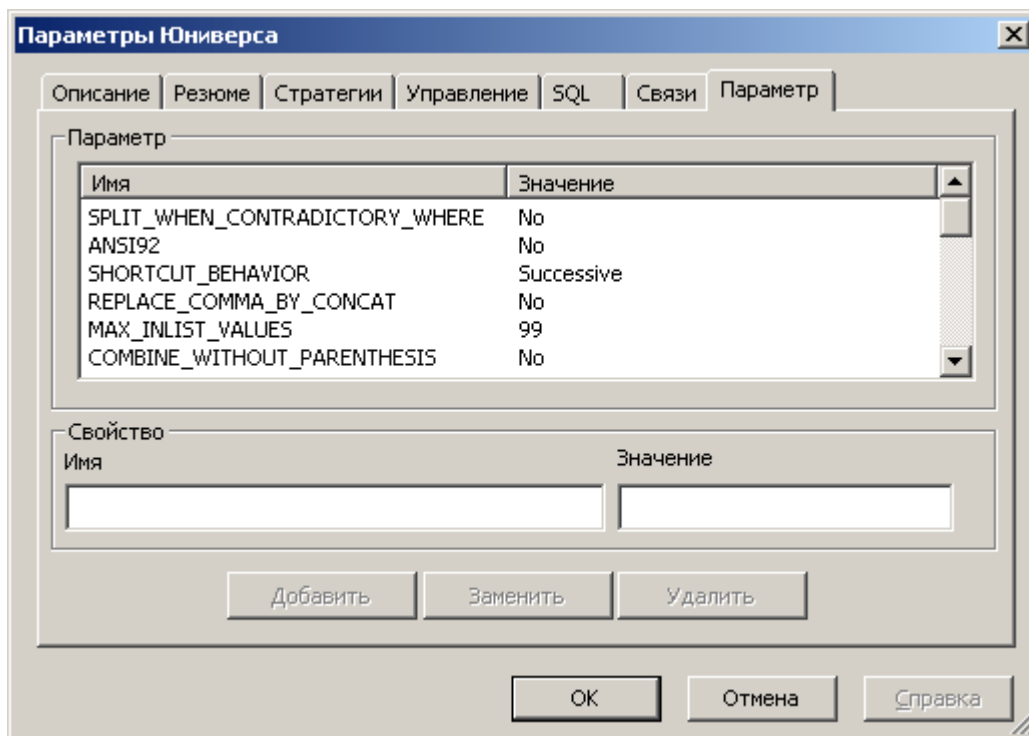
Эти параметры применяются только к активным юниверсам и сохраняются в UNV-файле. При изменении параметра SQL для юниверса в средстве создания юниверсов будет использоваться значение, определенное в этом средстве, а не в PRM-файле, связанном с драйвером доступа к данным для соединения.

4.4.11.3 Изменение параметров генерации программного кода SQL

Значения параметров SQL, определяющих генерацию SQL в продуктах, использующих юниверс можно изменить.

Чтобы редактировать параметры генерации SQL, выполните следующие действия.

1. Выберите ► [Файл](#) ► [Параметры](#) ►.
Откроется диалоговое окно [параметров](#).
2. Выберите вкладку [Параметр](#).
Откроется страница [параметров](#).



3. Измените, добавьте или удалите параметры, как описано ниже.

Выполните...	затем выполните следующее:
Добавление нового параметра	<ul style="list-style-type: none"> Выберите в списке необходимый параметр. Введите имя в поле "Имя" Введите значение в поле "Значение". Нажмите кнопку Добавить. В нижней части списка отобразится новое значение.
Измените имя значения.	<ul style="list-style-type: none"> Выберите в списке необходимый параметр. Введите новое имя в поле "Имя" Введите новое значение в поле Значение. Нажмите Заменить. <p>Определено новое значение.</p>
Удаление параметра	<ul style="list-style-type: none"> Выберите в списке параметр, который необходимо удалить. Щелкните Удалить.

4. Нажмите кнопку [OK](#).

❗ Примечание

Значения параметров генерации SQL, установленных для юниверса, доступны только для тех продуктов, которые используются данным юниверсом.

4.4.12 О параметрах генерации SQL

Ниже описываются параметры, влияющие на генерацию скрипта запроса. Параметры приводятся в алфавитном порядке в двух группах:

- Параметры SQL, задаваемые в интерфейсе пользователя средства создания юниверсов или средства дизайна информации. Это параметры SQL, которые являются общими для большинства драйверов доступа к данным. Каждый параметр действителен для юниверсов в которых он установлен.
- Параметры SQL, задаваемые в PRM-файлах. Это параметры, зависящие от соединения, которые указываются в файле параметров доступа к данным (PRM) для целевого драйвера доступа к данным.

Связанные сведения

[Параметры SQL, задаваемые в пользовательском интерфейсе \[страница 100\]](#)

[Параметры SQL, задаваемые в PRM-файлах \[страница 117\]](#)

4.4.13 Изменение параметров динамической генерации SQL

1. Выберите "Файл" > "Параметры".

Откроется диалоговое окно параметров.

2. Выберите вкладку "Параметры".

Появится страница "Параметры".

3. Измените, добавьте или удалите параметры, как описано ниже.

Выполните...	выполните следующее действие...
Чтобы добавить новый параметр,	Выберите в списке необходимый параметр. Введите имя в поле "Имя" Введите значение в поле "Значение". Нажмите "Добавить" В нижней части списка отобразится новое значение.

Выполните...	выполните следующее действие...
Чтобы изменить имя или значение, выделите параметр в списке.	Введите новое имя в поле "Имя"
	Введите новое значение в поле "Значение".
	Нажмите "Заменить".
	Определено новое значение.
Удаление параметра	выделите в списке параметр, который требуется удалить.
	Нажмите "Удалить".

4. Нажмите "ОК".

Значения параметров генерации SQL, установленных для юниверса, доступны только для тех продуктов, которые используются данным юниверсом.

Дополнительные сведения о динамических параметрах SQL и редактировании PRM-файла для драйвера доступа к данным см. Руководство доступа к данным. Чтобы ознакомиться с данным руководством, выберите "Справка" > "Руководство по доступу к данным".

4.4.14 Параметры SQL, задаваемые в пользовательском интерфейсе

4.4.14.1 ANSI92

ANSI92 = Yes|No

Значения	Yes/No
По умолчанию	No
Описание	<p>Указывает соответствует ли сгенерированный SQL ANSI92 стандарту.</p> <p>Yes: Позволяет выполнить генерацию SQL в соответствии со ANSI92 стандартом.</p> <p>No: Генерация SQL происходит в соответствии с PRM параметром OUTER_JOIN_GENERATION.</p>

4.4.14.2 AUTO_UPDATE_QUERY

AUTO_UPDATE_QUERY = Yes|No

Значения	Yes / No
По умолчанию	No
Описание	<p>Определяет причины недоступности объекта в запросе профилю пользователя.</p> <p>Yes: Запрос обновлен. Объекты из запроса удалены.</p> <p>No: Объект, хранящийся в запросе.</p>

4.4.14.3 Параметр BACK_QUOTE_SUPPORTED

BACK_QUOTE_SUPPORTED = Yes|No

Значения	<p>YES: в SQL-коде используются символы обратной кавычки.</p> <p>NO: в SQL-коде не используются символы обратной кавычки.</p>
По умолчанию	<p>YES</p> <p>No для базы данных OpenAccess.</p>
Описание	Задаёт использование в SQL-коде символов обратной кавычки для выделения имен таблиц и столбцов, содержащих пробелы и специальные символы.
Результат	Table name= `My Table`

4.4.14.4 BEGIN_SQL

BEGIN_SQL = <Строка>

Значения	Строка
По умолчанию	Пустая строка

Описание	<p>BEGIN_SQL используется в качестве префикса SQL-инструкций для учета ресурсов, назначения приоритетов и распределения нагрузки. Этот параметр применяется во всех случаях создания SQL, включая создание документов и запросы LOV.</p> <p>Он поддерживается в Web Intelligence, LiveOffice и QaaWS. Этот параметр игнорируется в отчетах Crystal.</p> <p>Пример для Teradata:</p> <pre>BEGIN_SQL=SET QUERY_BAND='string' for transaction;</pre> <p>Для данного параметра необходима строка, содержащая одно или несколько пар "имя-значение", разделенных точкой с запятой и заключенных в одиночные кавычки. Параметр, указанный после BEGIN_SQL, будет использоваться в качестве префикса для всех SQL-инструкций. Пары "имя-значение", введенные в данный параметр, записываются в системной таблице GetQueryBandPairs.</p> <p>Пример с тремя парами "имя-значение":</p> <pre>BEGIN_SQL=SET QUERY_BAND= 'UserID=Jones;JobID=980;AppID=TRM' for transaction;</pre> <p>Также можно использовать функцию @Variable в качестве значения в паре "имя-значение", возвращенное значение заключается в одиночные кавычки:</p> <pre>BEGIN_SQL=SET QUERY_BAND= 'USER= '@Variable('BOUSER');Document= '@Variable('DPNAME');' for transaction;</pre>
----------	---

4.4.14.5 BLOB_COMPARISON

BLOB_COMPARISON = Yes|No

Значения	Yes/No
По умолчанию	No
Возможно ли редактирование?	Нет
Описание	<p>Указывает, можно ли сгенерировать запрос с инструкцией DISTINCT, когда файл BLOB используется в инструкции SELECT. Связан с параметром No Duplicate Row в свойствах запроса.</p> <p>Yes: инструкция DISTINCT может использоваться в запросе.</p> <p>No: инструкция DISTINCT не может использоваться в запросе, даже если параметр запроса No Duplicate Row включен.</p>

4.4.14.6 BOUNDARY_WEIGHT_TABLE

BOUNDARY_WEIGHT_TABLE = Integer 32bits [0-9]

Значения	Целое число длиной 32 бита [0-9 или отрицательное целое число]
По умолчанию	-1
Описание	<p>Позволяет оптимизировать утверждение FROM, когда в таблице слишком много строк.</p> <p>Если размер таблицы (количество строк) превышает введенное значение, таблица объявляется подзапросом:</p> <pre>FROM (SELECT col1, col2,....., coln, ,....., FROM Table_Name WHERE простое условие).</pre> <p>Простое условие – это условие, не имеющее подзапроса.</p> <p>-1, 0 или любое отрицательное число означает, что данная оптимизация не используется.</p>
Ограничения	<p>Оптимизация не реализована в следующих случаях:</p> <ul style="list-style-type: none">• Оператор OR содержится в условии запроса• В SQL-запросе участвует только одна таблица• Запрос содержит внешнее объединение• В оптимизируемой таблице не выбрано ни одного условия• Оптимизируемая таблица является производной.

4.4.14.7 COLUMNS_SORT

COLUMNS_SORT = Yes|No

Значения	Yes/No
По умолчанию	No
Описание	<p>Определяет порядок отображения столбцов в таблице на панели структур.</p> <p>Yes: Столбцы отображаются в алфавитном порядке</p> <p>No: Столбцы отображаются в том порядке, в котором они были извлечены из базы данных</p>

4.4.14.8 COMBINE_WITHOUT_PARENTHESES

COMBINE_WITHOUT_PARENTHESES= Yes|No

Значения	Yes / No
По умолчанию	No
Описание	<p>Определяет необходимость инкапсулирования запроса со скобками, если он содержит UNION, INTERSECT или MINUS операторы. Используется с помощью RedBrick.</p> <p>Yes Удаляет скобки.</p> <p>No Оставляет скобки.</p>

4.4.14.9 COMBINED_WITH_SYNCRO

COMBINED_WITH_SYNCRO = Yes|No

Значения	Yes No
По умолчанию	No
Описание	<p>Указывает, разрешать ли выполнение запроса, который содержит UNION , INTERSECTION или EXCEPT операторы , и какие объекты в каждом подзапросе несовместимы.</p> <p>Yes: Указывает, разрешать ли выполнение запроса, который содержит UNION , INTERSECTION и EXCEPT операторы , и какие объекты в каждом подзапросе несовместимы. Этот тип запроса генерирует синхронизацию (два блока в отчете).</p> <p>No: Указывает, разрешать ли выполнение запроса, который содержит UNION , INTERSECTION и EXCEPT операторы , и какие объекты в каждом подзапросе несовместимы. При выполнении запроса отобразится следующее сообщение об ошибке: "Данный запрос слишком сложный" Один из подзапросов содержит несовместимые объекты. Это значение по умолчанию.</p>

4.4.14.10 COMPARE_CONTEXTS_WITH_JOINS

COMPARE_CONTEXTS_WITH_JOINS = Yes|No

Значения	Yes No
По умолчанию	Yes
Описание	<p>Указывает, как происходит сопоставление контекстов.</p> <p>Yes: Система подтверждает, что контексты выдают одинаковые объединения..</p> <p>No: Система подтверждает, что контексты выдают одинаковые наборы таблиц. Это значение по умолчанию.</p>

4.4.14.11 CORE_ORDER_PRIORITY

CORE_ORDER_PRIORITY = Yes|No

Значения	Yes No
По умолчанию	Нет
Описание	<p>Данный параметр применяется к классам или объектам, которые добавляются к связанному производному юниверсу. Этот параметр не применяется к классам и объектам главного юниверса и исходного производного юниверса. Этот параметр указывает способ организации новых классов и объектов в средстве создания юниверсов.</p> <p>См. также параметр FIRST_LOCAL_CLASS_PRIORITY.</p> <p>Yes: указывает, что классы и объекты организованы следующим образом:</p> <ul style="list-style-type: none"> Первый класс главного юниверса Объекты главного юниверса Все объекты производного юниверса, принадлежащие к первому классу главного юниверса Второй класс главного юниверса Объекты главного юниверса Все объекты производного юниверса, принадлежащие второму классу главного юниверса Другие классы главного юниверса... Классы и объекты производного юниверса <p>No: указывает, что классы и объекты следуют исходному порядку, указанному в производном юниверсе. Это значение по умолчанию.</p>

4.4.14.12 CORRECT_AGGREGATED_CONDITIONS_IF_DRILL

⚠ Предупреждение

Этот параметр устарел.

CORRECT_AGGREGATED_CONDITIONS_IF_DRILL = Yes|No

Значения	Yes No
По умолчанию	No
Описание	<p>Применяется только к Desktop Intelligence. Указывает, может ли Desktop Intelligence агрегировать меры в запросах и условиях.</p> <p>Yes: Desktop Intelligence может агрегировать меры по отдельности, в главном окне и в условиях, если запрос поддерживает детализацию.</p> <p>No: Desktop Intelligence не может агрегировать меры по отдельности, в главном окне и в условиях, если запрос не поддерживает детализацию.</p>

4.4.14.13 CUMULATIVE_OBJECT_WHERE

CUMULATIVE_OBJECT_WHERE = Yes|No

Значения	Yes No
По умолчанию	Нет

Описание	<p>Этот параметр применяется только к отфильтрованным объектам. Определяет способ объединения утверждения WHERE указанных объектов с условием запроса этих объектов.</p> <p>Yes: указывает на то, что утверждения WHERE объединяются с условием основного запроса с помощью оператора AND.</p> <p>Нет: указывает на то, что утверждение WHERE объекта объединяется с условием для данного объекта.</p> <p>Пример:</p> <p>Если задано условие найти всех клиентов из Франции, чьи имена отличаются от имени "Джон", или американские города, отличные от Нью-Йорка, то SQL-запрос будет выглядеть следующим образом:</p> <p>Да:</p> <pre>(customer.first_name <> 'John') OR (city.city <> 'New York' AND customer_country.country = 'France' AND city_country.country = 'USA')</pre> <p>Нет:</p> <pre>(customer.first_name <> 'John' AND customer_country.country = 'France') OR (city.city <> 'New York' AND city_country.country = 'USA')</pre>
----------	--

4.4.14.14 DECIMAL_COMMA

DECIMAL_COMMA = Да|Нет

Значения	Yes No
По умолчанию	No
Описание	<p>Указывает, что продукты Business Objects при необходимости вставляют запятую, в качестве разделителя десятков.</p> <p>Yes: Продукты Business Objects при необходимости вставляют запятую, в качестве разделителя десятков.</p> <p>No: Продукты Business Objects не вставляют запятую, в качестве разделителя десятков. Это значение по умолчанию.</p>

4.4.14.15 DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION

DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION = Yes|No

Значения	Yes/No
По умолчанию	No
Описание	<p>Алгоритм оптимизации можно использовать для того, чтобы оптимизировать размер возвращенных массивов вместо использования значений по умолчанию.</p> <p>No: Во всех запросах к юниверсу используется оптимизация.</p> <p>Yes: В запросах используется набор значений по умолчанию.</p> <p>Этот параметр также применяется к соединениям OLAP.</p>

4.4.14.16 DISTINCT_VALUES

DISTINCT_VALUES = GROUPBY|DISTINCT

Значения	GROUPBY DISTINCT
По умолчанию	DISTINCT
Описание	<p>Указывает, создается ли код SQL с использованием утверждения DISTINCT или GROUP BY в списке значений и панели запросов, когда параметр "Не извлекать повторяющиеся строки" активен.</p> <p>DISTINCT: код SQL сгенерирован с помощью утверждения DISTINCT, например;</p> <pre>SELECT DISTINCT cust_name FROM Customers</pre> <p>GROUPBY: код SQL сгенерирован с помощью утверждения GROUP BY, например;</p> <pre>SELECT cust_name FROM Клиенты GROUP BY cust_name</pre>

4.4.14.17 END_SQL

END_SQL = Строка

Значения	Строка
----------	--------

По умолчанию	<пустая строка>
Описание	Инструкция, указанная в данном параметре, добавляется в конец каждой SQL-инструкции.
Пример	<ul style="list-style-type: none"> Для источника данных SAP HANA можно передать имя пользователя, выполняющего запрос, из юниверса в SAP HANA, вставив параметр @Variable('BOUSER') следующим образом: END_SQL=-- @Variable(' BOUSER ') Для баз данных IBM DB2 можно использовать следующее: END_SQL=FOR SELECT ONLY. Сервер будет считывать блоки данных намного быстрее.

4.4.14.18 EVAL_WITHOUT_PARENTHESIS

EVAL_WITHOUT_PARENTHESIS = Yes|No

Значения	Yes No
По умолчанию	No
Описание	<p>По умолчанию функция @Выбрать(Класс\объект) заменяется инструкцией SELECT для объекта <Класс\объект>, заключенного в скобки.</p> <p>Например, при объединении двух инструкций @Select, @Select(объект1) *@Select(объект2).</p> <p>Если SQL(объект1) = A-B а SQL(объект2) =C,</p> <p>тогда будет выполнена операция (A-B) * (C).</p> <p>Чтобы избежать добавление скобок по умолчанию, установите EVAL_WITHOUT_PARENTHESIS = Yes. Тогда будет выполнена операция A-B * C.</p> <p>Да: Скобки удалены из инструкции SELECT для функции @Select(Класс\объект)</p> <p>Но: инструкция Select берется в скобки для функции@Select(Класс\объект).</p>

4.4.14.19 FILTER_IN_FROM

FILTER_IN_FROM = Yes|No

Значения	Yes No
----------	----------

По умолчанию	No
Описание	<p>Определяет, включены ли условия запроса в утверждение FROM. Эта настройка применима только, если другая настройка параметра юниверса ANSI 9.2 установлена на Yes.</p> <p>Yes: при редактировании внешнего объединения свойствам поведения по умолчанию, выбранным в раскрывающемся списке диалогового окна "Дополнительные свойства объединения" средства создания юниверсов, присваивается значение "Все объекты в FROM".</p> <p>No: при редактировании внешнего объединения свойствам поведения по умолчанию, выбранным в раскрывающемся списке диалогового окна "Дополнительные свойства объединения" средства создания юниверсов, присваивается значение "Нет объектов в FROM".</p>

4.4.14.20 FIRST_LOCAL_CLASS_PRIORITY

FIRST_LOCAL_CLASS_PRIORITY = Yes|No

Значения	Yes No
По умолчанию	No
Описание	<p>Этот параметр применяется только к Web Intelligence и Crystal Reports.</p> <p>Учитывается только при CORE_ORDER_PRIORITY=Yes.</p> <p>Yes: Классы производного юниверса указываются в верхней части списка.</p> <p>No: Объекты и подклассы из производного юниверса отображаются после объектов и подклассов главного юниверса.</p>

4.4.14.21 FORCE_SORTED_LOV

FORCE_SORTED_LOV = Yes|No

Значения	Yes No
По умолчанию	No

Описание	Получает список значений, который был сортирован.
	Yes: Указывает на то, что список значений сортирован.
	No: Указывает на то, что список значений не сортирован.

4.4.14.22 INNERJOIN_IN_WHERE

INNERJOIN_IN_WHERE = Yes|No

Значения	Yes No
По умолчанию	No. Чтобы активировать параметр, необходимо добавить его вручную.
Описание	<p>Позволяет дать системе указание на необходимость генерации синтаксической конструкции SQL со всеми внутренними объединениями в операторе WHERE, когда для параметра ANSI 92 установлено значение yes. Это возможно, если запрос содержит только внутренние объединения (не содержит объединений FULL OUTER, RIGHT OUTER или LEFT OUTER).</p> <p>Yes. Если для параметра ANSI 92 установлено значение yes, создается синтаксис объединений ANSI 92 в утверждении FROM, за исключением случая, когда запрос содержит только внутренние объединения. В этом случае внутренние объединения переходят в утверждение WHERE.</p> <p>No. Если для параметра ANSI 92 установлено значение Yes, создается синтаксис объединений ANSI 92 в утверждении FROM.</p>

4.4.14.23 JOIN_BY_SQL

JOIN_BY_SQL = Yes|No

Значения	Yes No
По умолчанию	Нет
Описание	<p>Указывает, как обрабатываются несколько инструкций SQL. Несколько инструкций можно комбинировать (если это разрешено базой данных).</p> <p>Yes: указывает на то, что несколько инструкций SQL комбинируются.</p> <p>No: указывает на то, что несколько инструкций SQL не комбинируются. Это значение по умолчанию.</p>

4.4.14.24 MAX_INLIST_VALUES

MAX_INLIST_VALUES = [0-99]

Значения	Целое число: минимальное – 1, максимальное зависит от DB
По умолчанию	-1
Описание	<p>Позволяет устанавливать максимальное количество значений, которые можно внести в условие, при использовании IN LIST оператора.</p> <p>99: Указывает, что можно внести до 99 значений, при создании условия с помощью оператора IN LIST .</p> <p>Максимальное разрешенное значение, которое можно ввести, зависит от базы данных.</p> <p>Значение -1 означает, что ограничения на количество возвращаемых значений отсутствуют, действуют только ограничения, налагаемые базой данных.</p>

4.4.14.25 OLAP_UNIVERSE

OLAP_UNIVERSE = Yes|No

Значения	Yes No
По умолчанию	Нет значений по умолчанию.
Описание	<p>Указывает, используется ли юниверс OLAP. Если в средстве создания юниверсов используется юниверс OLAP, это значение равняется Yes, а параметр является видимым в списке параметров SQL. Если юниверс не является юниверсом OLAP, то параметр не отображается в списке параметров SQL.</p> <p>Yes: Данный юниверс является юниверсом OLAP.</p> <p>No: Данный юниверс не является юниверсом OLAP.</p>

4.4.14.26 PATH_FINDER_OFF

По умолчанию параметр не вносится в список. Необходимо добавить параметр в список вручную и установить значение.

PATH_FINDER_OFF= Да|Нет

Значения	Yes No
По умолчанию	Не по умолчанию. Параметр необходимо ввести вручную.
Описание	Используется для HPIW, так как генерирование объединения выполнено базой данных. Yes: объединения не генерируется в запросе. No: объединения генерируется в запросе. Это значение по умолчанию.

4.4.14.27 REPLACE_COMMA_BY_CONCAT

REPLACE_COMMA_BY_CONCAT= Yes|No

Значения	Yes No
По умолчанию	No
Описание	В предыдущих версиях средства создания юниверсов запятая могла использоваться для разделения нескольких полей в инструкции Select для объекта. Запятая рассматривалась как оператор конкатенации. Чтобы сохранить эти параметры для юниверсов, которые используют запятую таким образом, необходимо установить REPLACE_COMMA_BY_CONCAT на No. В текущей версии средства создания юниверсов этот параметр по умолчанию имеет значение Yes таким образом, что любые выражения, использующие запятую, автоматически переходят на использование синтаксиса конкатенации. Yes: Запятая заменяется выражением конкатенации, если обнаружен объект с многочисленными полями. No: Оставьте запятую как есть.

4.4.14.28 SELFJOINS_IN_WHERE

SELFJOINS_IN_WHERE = Yes|No

Значения	Yes No
По умолчанию	No

Описание	<p>Самостоятельные объединения обычно включаются в утверждение FROM. Этот параметр позволяет принудительно создать синтаксис SQL со всеми условиями самостоятельных объединений в утверждении WHERE. Чтобы параметр ANSI 92 учитывался, для него необходимо установить значение Yes.</p> <p>Чтобы активировать параметр, необходимо добавить его в список вручную.</p> <p>Yes: Условия самостоятельных объединений находятся в утверждении WHERE запроса SQL.</p> <p>No: Синтаксис самостоятельных объединений создается в соответствии с соглашением ANSI 92, а условия самостоятельных объединений находятся в утверждении ON определения объединений таблиц из утверждения FROM запроса SQL.</p>
----------	---

4.4.14.29 SHORTCUT_BEHAVIOR

SHORTCUT_BEHAVIOR = Global|Successive

Значения	Global Successive
По умолчанию	Successive
Описание	<p>Указывает применение объединений быстрого доступа. Данный параметр был внесен в список ранее как GLOBAL_SHORTCUTS в файлы PRM. Значения были изменены с Global на Yes и с Successive на No.</p> <p>Global: Указывает, что объединения быстрого доступа считаются поочередными. Объединение быстрого доступа применяется только в том случае, если он действительно пропускает одну или несколько таблиц и если он не удаляет таблицу из пути объединения, используемом следующим объединением быстрого доступа.</p> <p>Successive: Указывает на применение объединений быстрого доступа. Примечание: Если генерируется декартово произведение, объединения быстрого доступа не могут быть применены.</p>

4.4.14.30 SMART_AGGREGATE

SMART_AGGREGATE = Yes|No

Значения	Yes No
По умолчанию	No

Описание	<p>Определяет использование агрегатных таблиц для интеллектуальных мер на основе агрегатных таблиц. Это обеспечивает правильное агрегирование объектов юниверса, опирающихся на соотношения. По умолчанию системой используются предварительно вычисленные значения из агрегатных таблиц. Если эти таблицы не согласованы по времени (разным периодам времени), этот параметр применяется для обеспечения использования наиболее подробных агрегатных таблиц.</p> <p>Этот параметр не виден в списке параметров юниверса (по умолчанию он не активен). Дизайнер юниверса должен вручную вставить его в список параметров, прежде чем активировать его (значение Yes).</p> <p>Yes: любой дополнительный запрос из группирующего набора должен базироваться на таблице агрегирования начального запроса для интеллектуальной меры на основе такой таблицы.</p> <p>No: система берет наиболее подходящую таблицу агрегирования.</p>
----------	--

4.4.14.31 STORED_PROC_UNIVERSE

STORED_PROC_UNIVERSE = Yes|No

Значения	Yes No
По умолчанию	No
Описание	<p>Для данного значения автоматически задается значение Yes при создании юниверса, содержащего хранимые процедуры. Не изменяйте это значение вручную.</p> <p>Yes: создаваемый/редактируемый юниверс содержит хранимые процедуры.</p> <p>No: юниверс не содержит хранимых процедур.</p>

4.4.14.32 THOROUGH_PARSE

THOROUGH_PARSE = Yes|No

Значения	Yes No
По умолчанию	No

Описание	<p>Указывает на методологию, используемую для разбора по умолчанию в панели запросов и разбора отдельного объекта.</p> <p>Да: инструкции <code>PREPARE</code>, <code>DESCRIBE</code> и <code>EXECUTE</code> используются при разборе SQL для объектов.</p> <p><code>Prepare+DescribeCol+Execute</code></p> <p>Инструкции No: <code>PREPARE</code> и <code>DESCRIBE</code> используются при разборе SQL для объектов.</p>
----------	--

4.4.14.33 TRUST_CARDINALITIES

TRUST_CARDINALITIES = Yes|No

Значения	Yes No
По умолчанию	No
Описание	<p>Позволяет оптимизировать SQL в случае завышенных результатов.</p> <p>Yes: В запросах, которые включают в себя меру, все условия, завышающие меру и не отображающиеся в объектах результата, преобразуются в подзапросы, чтобы гарантировать, что таблицы которые возвращают ложные результаты, не будут включены в запрос.</p> <p>No: Оптимизация не применима.</p>

4.4.14.34 UNICODE_STRINGS

UNICODE_STRINGS = Yes|No

Значения	Yes No
По умолчанию	No

Описание	<p>Указывает, может ли текущий юниверс, управлять строкой Unicode. Применим только к серверу Microsoft SQL и Oracle 9. Если знак базы данных, установленный в файле SBO, настроен на Unicode, тогда необходимо изменить генерацию SQL, чтобы обработать определенные типы столбцов Unicode, таких как NCHAR и NVARCHAR.</p> <p>Yes: Условия, основанные на строках, отформатированных в SQL, в соответствии со значением параметра UNICODE_PATTERN в файле PRM, например Microsoft SQL Server (sqlsrv.prm) : UNICODE_PATTERN=N\$</p> <p>Условие Customer_name='Arai ' становится</p> <p>Customer_name=N'Arai'.</p> <p>Примечание. При создании запроса с синтаксисом @Prompt , основанном на значении Unicode, тип данных должен быть 'U', а не 'C'</p> <p>Но: Все условия, основанные на строках, форматируются в стандартном SQL. Например, условие Customer_name='Arai ' остается Customer_name='Arai'</p>
----------	---

4.4.15 Параметры SQL, задаваемые в PRM-файлах

4.4.15.1 Параметр CASE_SENSITIVE

```
<Parameter Name="CASE_SENSITIVE">NO</Parameter>
```

Описание	Определяет необходимость учета регистра для базы данных. Этот параметр используется в Oracle.
Значения	<p>YES: в базе данных учитывается регистр.</p> <p>NO: в базе данных не учитывается регистр.</p>
По умолчанию	NO

4.4.15.2 Параметр CHECK_OWNER_STATE

```
<Parameter Name="CHECK_OWNER_STATE">NO</Parameter>
```

Описание	Указывает, проверяет ли SQL следующее: поддерживает ли база данных классификацию таблиц по имени владельца.
----------	---

Значения	<p>YES: SQL проверяет, поддерживает ли база данных классификацию таблиц по имени владельца.</p> <p>NO: SQL не проверяет, поддерживает ли база данных классификацию таблиц по имени владельца.</p>
По умолчанию	YES

4.4.15.3 Параметр CHECK_QUALIFIER_STATE

```
<Parameter Name="CHECK_QUALIFIER_STATE">NO</Parameter>
```

Описание	Указывает, проверяет ли SQL следующее: поддерживает ли база данных классификацию таблицы по классификатору.
Значения	<p>YES: SQL проверяет, поддерживает ли база данных классификацию таблиц по квалификатору.</p> <p>NO: SQL не проверяет, поддерживает ли база данных классификацию таблиц по квалификатору.</p>
По умолчанию	YES

4.4.15.4 Параметр COMMA

```
<Parameter Name="COMMA">|| ' ' ||</Parameter>
```

Описание	<p>Указывает, какой символ оператору конкатенации базы данных следует использовать для замены запятой в случае объектов, которые имеют следующий синтаксис:</p> <p>Tab.Col1, Tab.Col2.</p> <p>Этот параметр используется со всеми драйверами доступа к данным.</p>
Значения	<p> ' ' </p> <p>+ ' '+</p>
По умолчанию	' '
Результат	Tab.Col1 ' ' Tab.Col2

4.4.15.5 Параметр CONCAT

```
<Parameter Name="CONCAT">||</Parameter>
```

Описание	Указывает оператора конкатенации. Этот параметр используется со всеми драйверами доступа к данным.
Значения	двойная вертикальная черта () или знак плюса (+)
По умолчанию	

4.4.15.6 Параметр DATE_WITHOUT_QUOTE

```
<Parameter Name="DATE_WITHOUT_QUOTE">YES</Parameter>
```

Описание	Определяет, заключаются ли даты в синтаксисе SQL в одинарные кавычки. Этот параметр используется в MS Access.
Значения	YES: даты не заключаются в одинарные кавычки. NO: даты заключаются в одинарные кавычки.
По умолчанию	YES

4.4.15.7 DELIMIT_LOWERCASE

```
<Parameter Name="DELIMIT_LOWERCASE"></Parameter>
```

Описание	Определяет, требуется ли заключать в кавычки идентификаторы в нижнем регистре.
Значения	YES: идентификаторы в нижнем регистре заключаются в кавычки. NO: идентификаторы в нижнем регистре не заключаются в кавычки.

4.4.15.8 Параметр EXTERN_SORT_EXCLUDE_DISTINCT

```
<Parameter Name="EXTERN_SORT_EXCLUDE_DISTINCT">YES</Parameter>
```

Описание	Определяет, создается ли в приложении выражение SELECT DISTINCT, если запрос содержит утверждение ORDER BY.
Значения	<p>YES: выражение SELECT DISTINCT не создается, если запрос содержит оператор ORDER BY.</p> <p>NO: выражение SELECT DISTINCT создается, если запрос содержит оператор ORDER BY.</p>
По умолчанию	YES

4.4.15.9 GROUPBY_WITH_ALIAS

```
<Parameter Name="GROUPBY_WITH_ALIAS">YES</Parameter>
```

Описание	Указывает, может ли база данных создавать утверждение GROUP BY, содержащее псевдонимы, в инструкции SELECT.
Значения	<p>YES: позволяет создавать оператор GROUP BY с псевдонимами в выражении SELECT.</p> <p>NO: не позволяет создавать оператор GROUP BY с псевдонимами в выражении SELECT.</p>
По умолчанию	YES

4.4.15.10 Параметр IDENTIFIER_DELIMITER

```
<Parameter Name="IDENTIFIER_DELIMITER">"</Parameter>
```

Описание	<p>Определяет следующие функции:</p> <ul style="list-style-type: none"> Имена таблиц и столбцов, в которых содержатся пробелы или специальные символы, заключаются в кавычки, если включен параметр BACK_QUOTE_SUPPORTED. Имена таблиц или столбцов, независимо от содержащихся в них символов, заключаются в кавычки, если включен параметр DELIMIT_IDENTIFIERS. <p>Для использования данного параметра необходимо присвоить параметру BACK_QUOTE_SUPPORTED или DELIMIT_IDENTIFIERS значение YES. Данное значение является значением по умолчанию для обоих параметров.</p>
----------	--

Значения	" (двойные кавычки): имена таблиц или столбцов, содержащие пробелы или специальные символы, заключаются в двойные кавычки. " (одинарные кавычки): имена таблиц или столбцов, содержащие пробелы или специальные символы, заключаются в одинарные кавычки. Это значение можно использовать только в Microsoft Access.
По умолчанию	"
Результат	Table name="My Table"

4.4.15.11 Параметр IF_NULL

```
<Parameter Name="IF_NULL">NO</Parameter>
```

Описание	Указывает функцию, которая принимает два параметра. Если первый параметр возвращает значение NULL, используется значение второго параметра.
Значения	Зависят от базы данных.
По умолчанию	Зависят от базы данных.

4.4.15.12 OUTERJOINS_COMPLEX

```
<Parameter Name="OUTERJOINS_COMPLEX"></Parameter>
```

Описание	Этот параметр вместе с параметром OUTERJOINS_GENERATION управляет генерацией запросов внешних объединений.
Значения	YES NO

4.4.15.13 Параметр OUTERJOINS_GENERATION

```
<Parameter Name="OUTERJOINS_GENERATION">ANSI92</Parameter>
```

Этот параметр управляет поведением генерации внешнего объединения по умолчанию.

- Генерация внешнего объединения соответствует спецификации ANSI92.

- Генерация внешнего объединения остается такой же, что и в предыдущих версиях приложения средства создания юниверсов.

📌 Примечание

Параметр `OUTERJOINS_GENERATION` файла PRM связан с настройкой юниверса `ANSI92` следующим образом:

- Если для параметра `OUTERJOINS_GENERATION` файла PRM задано значение `ANSI92`, а для настройки юниверса `ANSI92` задано значение `NO`, то параметр PRM заменяет собой настройку юниверса, и внешние объединения соответствуют поведению `ANSI92`.
- Если для параметра `OUTERJOINS_GENERATION` PRM-файла задано значение `USUAL`, то настройка `ANSI92` имеет более высокий приоритет, и внешние связи соответствуют `ANSI92` в зависимости от значения настройки юниверса `ANSI92`: `YES` или `NO`.

→ Напоминание

Значение `ANSI92` позволяет не использовать параметр `REVERSE_TABLE_WEIGHT` для оптимизации создания SQL-кода. Внешние связи, соответствующие требованиям `ANSI92`, имеют наивысший приоритет среди таблиц в выражении SQL.

Описание	<p>Указывает синтаксис SQL для внешних объединений.</p> <p>Значение <code>ANSI 92</code> генерирует внешнее объединение в утверждении <code>FROM</code>. Другие значения генерируют внешнее объединение в утверждении <code>WHERE</code>.</p> <p>При изменении этой настройки следует проверить свойства объединения, чтобы убедиться в допустимости выражения внешнего объединения и правильности количества элементов. <code>ANSI92</code> не поддерживает настройку вручную в синтаксисе объединения.</p>
Значения	<p>Основные значения параметра <code>OUTERJOINS_GENERATION</code>:</p> <ul style="list-style-type: none"> • <code>ANSI92</code>: поведение по умолчанию внешнего объединения соответствует стандарту <code>ANSI92</code>, независимо от значения настройки <code>ANSI92</code> на уровне средства создания юниверсов. • <code>NO</code>: внешние объединения не поддерживаются. • <code>USUAL</code>: поведение по умолчанию внешнего объединения то же, что и в предыдущих версиях средства создания юниверсов. Это поведение переопределяется, если параметру <code>ANSI92</code> средства создания юниверсов присвоено значение <code>YES</code>. <p>Другие настройки доступны в зависимости от базы данных. См. значения по умолчанию ниже.</p>

По умолчанию

ANSI_92: значение по умолчанию для Oracle, MS SQL Server 2005 и Sybase.

DB2: значение по умолчанию для IBM DB2.

FULL_ODBC: значение по умолчанию для Microsoft SQL Server.

INFORMIX: значение по умолчанию для IBM Informix.

INGRES: значение по умолчанию для Teradata.

NO: значение по умолчанию для ODBC.

USUAL: значение по умолчанию для HP Neoview, Netezza, IBM Red Brick и MS SQL Server 2000.

Примеры настроек параметра OUTERJOINS_GENERATION

Настройка = USUAL:

```
FROM T1, T2
WHERE T1.col1(+) = T2.col2
```

Настройка = DB2:

```
FROM T2 LEFT OUTER JOIN T1
ON T1.col1 = T2.col2
```

Настройка = ODBC:

```
FROM {oj T1 LEFT OUTER JOIN T2 ON T1.col1=T2.col2}
Where (T2.col3 = T3.col1)
```

Настройка = INFORMIX

```
FROM T2
OUTER T1
WHERE T1.col1=T2.col2
```

Настройка = FULL-ODBC

```
FROM {oj T1 RIGHT OUTER JOIN T2 ON T2.col2=T1.col1
T2 INNER JOIN 3 on T2.col3 = T3.col1}
```

Настройка = ANSI_92:

```
SELECT DISTINCT
  t1.col1,
  t2.col2
FROM
  (t1 RIGHT OUTER JOIN t2 ON (t1.col1=t2.col2) )
```

Использование OUTERJOINS в Oracle

Настройка по умолчанию OUTERJOINS_GENERATION может влиять на поведение имеющихся юниверсов независимо от настройки параметра ANSI92 на уровне юниверса.

Чтобы задать для имеющихся юниверсов Oracle поведение как в предыдущих версиях средства создания юниверсов, выполните следующие действия:

1. В файле PRM задайте для параметра OUTERJOINS_GENERATION значение USUAL.
2. В файле PRM задайте для параметров LEFT_OUTER и RIGHT_OUTER значение \$ (+)

Дополнительные сведения о параметрах SQL для юниверса, а также о PRM-файлах предыдущих версий средства создания юниверсов см. в *Руководстве разработчика*.

4.4.15.14 Параметр OVER_CLAUSE

```
<Parameter Name="OVER_CLAUSE">YES</Parameter>
```

Описание	Позволяет приложениям SAP BusinessObjects включать функции RSQL при генерации SQL-кода. Поддерживаемые функции RSQL для базы данных перечислены в параметре ANALYTIC_FUNCTIONS.
Значения	YES: приложения могут включать функции RSQL при генерации SQL. NO: приложения не могут включать функции RSQL при генерации SQL.
По умолчанию	YES

4.4.15.15 Параметр OWNER

```
<Parameter Name="OWNER">YES</Parameter>
```

Описание	Определяет, поддерживает ли база данных использование имени владельца в качестве префикса для таблиц.
Значения	YES: база данных поддерживает использование имени владельца в качестве префикса для таблиц. NO: база данных не поддерживает использование имени владельца в качестве префикса для таблиц.
По умолчанию	YES

4.4.15.16 Параметр PREFIX_SYS_TABLE

```
<Parameter Name="PREFIX_SYS_TABLE">RBW_</Parameter>
```

```
<Parameter Name="PREFIX_SYS_TABLE">MSys</Parameter>
```

Описание	Определяет необходимость отображения системных таблиц в средстве создания универсов.
Значения	<p>MSys: системные таблицы MS Access скрыты в браузере таблиц средства создания универсов.</p> <p>RBW_ : системные таблицы Red Brick скрыты в браузере таблиц средства создания универсов.</p> <p>Нет значения: системные таблицы отображаются в браузере таблиц средства создания универсов.</p>
По умолчанию	<p>MSys: значение по умолчанию для MS Access</p> <p>RBW_ : значение по умолчанию для IBM Red Brick</p>

4.4.15.17 QUALIFIER

```
<Parameter Name="QUALIFIER">NO</Parameter>
```

Описание	Определяет, поддерживает ли база данных использование квалификатора в качестве префикса для таблиц.
Значения	<p>YES: база данных поддерживает использование имени квалификатора в качестве префикса для таблиц.</p> <p>NO: база данных не поддерживает использование имени квалификатора в качестве префикса для таблиц.</p>
По умолчанию	Зависит от реляционной СУБД.

4.4.15.18 Параметр QUOTE_OWNER

```
<Parameter Name="QUOTE_OWNER">YES</Parameter>
```

Описание	Указывает, следует ли заключать в одиночные кавычки имя владельца. Используется только в базах данных IBM Informix.
Значения	<p>YES: имена таблиц получают префикс в виде имени владельца, заключенного в одиночные кавычки. Это обязательное требование для базы данных IBM Informix, соответствующей стандарту ANSI. В противном случае IBM Informix преобразует имя владельца в символы верхнего регистра.</p> <p>NO: имена таблиц не получают префикс в виде имени владельца, заключенного в одиночные кавычки.</p>
По умолчанию	YES
Результат	<p>SELECT Alias.col (<Alias> – локальный псевдоним)</p> <p>FROM 'Owner'.table.col Alias</p>

4.4.15.19 Параметр REFRESH_COLUMNS_TYPE

```
<Parameter Name="REFRESH_COLUMNS_TYPE">O</Parameter>
```

Описание	Указывает порядок обновления столбцов.
Значения	<p>O: столбцы обновляются по имени владельца.</p> <p>Q: столбцы обновляются по имени квалификатора.</p> <p>T: столбцы обновляются по имени таблицы.</p>

По умолчанию

O: значение по умолчанию для
Oracle

Q: значение по умолчанию для
IBM Red Brick, Sybase, MS SQL
Server и MS Access

4.4.15.20 REMOVE_SEMICOLONS

```
<Parameter Name="REMOVE_SEMICOLONS"></Parameter>
```

Описание

Определяет необходимость удаления точек с запятой в пользовательском коде SQL для панели запросов SAP BusinessObjects.

Значения

YES: в панели запросов удаляются точки с запятой.

NO: в панели запросов не удаляются точки с запятой.

4.4.15.21 Параметр REVERSE_TABLE_WEIGHT

```
<Parameter Name="REVERSE_TABLE_WEIGHT">YES</Parameter>
```

Описание	<p>Указывает, в каком порядке должны генерироваться таблицы. Этот параметр используется в Oracle. Этот параметр также может использоваться в ряде других баз данных, причем значения YES и NO могут иметь обратный смысл.</p> <div> <div> Примечание <p>Этот параметр не поддерживается базой данных Teradata.</p> </div> <div> → Напоминание <p>Если параметру OUTERJOINS_GENERATION присвоено значение ANSI92 или параметру юниверса ANSI92 присвоено значение YES, значение параметра REVERSE_TABLE_WEIGHT не влияет на оптимизацию процесса создания SQL-кода.</p> </div> </div>
Значения	<p>YES: создание таблиц осуществляется от наименьших к наибольшим.</p> <p>NO: создание таблиц осуществляется от наибольших к наименьшим.</p>
По умолчанию	YES

4.4.15.22 Параметр UNICODE_PATTERN

```
<Parameter Name="UNICODE_PATTERN">UNISTR($)</Parameter>
```

Описание	<p>Применяется только в том случае, если параметру генерации SQL-кода юниверса UNICODE_STRINGS присвоено значение YES. Все основанные на строках условия форматируются с использованием этого строкового значения. Этот параметр используется только для Microsoft SQL Server и Oracle.</p>
----------	---

Значения	N\$: для Microsoft SQL Server
	UNISTR (\$): для Oracle

4.4.15.23 Параметр USER_INPUT_DATE_FORMAT

```
<Parameter Name="USER_INPUT_DATE_FORMAT">'dd-MM-yyyy HH:mm:ss'</Parameter>
```

Описание	Указывает форматы даты и времени по умолчанию, генерируемые в утверждении WHERE из SQL-инструкции.
----------	--

Значения	<p>{\d 'yyyy-mm-dd'}: формат даты по умолчанию для ODBC.</p> <p>'DD-MM-YYYY HH:MM:SS': формат даты и времени по умолчанию для Oracle.</p> <p>'MM/DD/YYYY': формат даты по умолчанию для IBM Informix.</p> <p>'yyyy-mm-dd HH:mm:ss': формат даты и времени по умолчанию для Microsoft SQL Server и большинства серверов IBM DB2.</p> <p>'mm/dd/yyyy hh:m:s am/pm': формат даты и времени по умолчанию для Sybase.</p> <p>'yyyy-mm-dd': формат даты по умолчанию для шлюза Sybase.</p>
----------	--

Примечание

Если в ODBC требуется использовать переменные времени или метки времени, необходимо заменить значение формата даты по умолчанию на {\t 'hh:mm:ss'} или {\t\s 'yyyy-mm-dd hh:mm:ss'} в файле `odbc.sbo`.

По умолчанию	См. значения выше.
--------------	--------------------

4.4.15.24 USER_INPUT_NUMERIC_SEPARATOR

```
<Parameter Name="USER_INPUT_NUMERIC_SEPARATOR">.</Parameter>
```

Описание	Указывает десятичный знак по умолчанию, используемый в генерируемом скрипте SQL.
Значения	'.' (точка)
По умолчанию	'.'

4.4.15.25 DELIMIT_IDENTIFIERS

```
<Parameter Name="DELIMIT_IDENTIFIERS">YES</Parameter>
```

Описание	Определяет, заключаются ли идентификаторы базы данных в кавычки. Идентификаторы заключаются в кавычки с использованием разделителя, заданного в параметре IDENTIFIER_DELIMITER.
Значения	YES: идентификаторы могут заключаться в кавычки. NO: идентификаторы не могут заключаться в кавычки.
По умолчанию	YES
Результат	Table name="my_table"

4.4.15.26 Параметр EXT_JOIN_INVERT

```
<Parameter Name="EXT_JOIN_INVERT">YES</Parameter>
```

Описание	Указывает, как отображать символ внешнего объединения в выражении объединения. Этот параметр используется в базах данных IBM DB2, IBM Informix, Oracle и Teradata.
Значения	YES: при установке флажка <i>Внешнее соединение</i> в диалоговом окне <i>Редактировать соединение</i> приложения Universe Designer символ внешнего соединения отображается в противоположном положении в выражении объединения. NO: при установке флажка <i>Внешнее соединение</i> в диалоговом окне <i>Редактировать соединение</i> приложения Universe Designer символ внешнего соединения отображается на той же стороне, на которой было создано внешнее соединение.
По умолчанию	YES

4.4.15.27 Параметр KEY_INFO_SUPPORTED

```
<Parameter Name="KEY_INFO_SUPPORTED">YES</Parameter>
```

Описание	Указывает, можно ли извлекать определения первичного и вторичного ключа из базы данных.
Значения	<p>YES: извлечение определений первичного и вторичного ключа из базы данных поддерживается. Этот параметр позволяет приложению Universe Designer отображать ключи в окне Структура.</p> <p>NO: извлечение определений первичного и вторичного ключа из базы данных не поддерживается.</p>
По умолчанию	YES

4.4.15.28 ORDER_BY_STRINGS

```
<Parameter Name="ORDER_BY_STRINGS">YES</Parameter>
```

Описание	Определяет, поддерживает ли база данных корректную обработку утверждения ORDER BY на основе столбца строк. Этот параметр соответствует возможности ORDERBYSTRINGS приложения SAP BusinessObjects Data Federator. Если база данных не поддерживает такую обработку, сервер запросов Data Federator выполняет сортировку.
Значения	<p>YES: база данных поддерживает обработку сортировки.</p> <p>NO: база данных не поддерживает обработку сортировки.</p>

5 Создание схемы с помощью таблиц и объединений

5.1 Общие сведения

В данной главе описывается способ создания схемы, содержащей все структуры SQL, необходимые для создания объектов, которые пользователи Web Intelligence используют для создания отчетов. Данные структуры SQL включают таблицы, столбцы, объединения и функции баз данных. Построение правильной схемы является основой для построения юниверса, который соответствует всем требованиям отчета конечного пользователя.

5.2 Что такое схема?

Схема – это графическое представление структуры базы данных. В средстве создания юниверсов можно создать схему для части базы данных, которую представляет юниверс.

Схема содержит таблицы и объединения. Таблицы содержат столбцы, которые в итоге соответствуют объектам используемых для создания отчетов конечными пользователями. Связанные объединения таблиц, для возвращения верных данных для запросов, которые запускаются более одной таблицей.

Чтобы создать схему на панели [Структура](#), выберите таблицы в целевой базе данных с помощью [обозревателя таблиц](#). Создание объединений связанных таблиц. При наличии дизайнера схемы для юниверса, можно проверить схему используя автоматическую проверку целостности.

Схема в примере юниверс Beach отображается следующим образом:



5.2.1 Дизайнер схемы является основным для успешного юниверса

Хороший дизайнер схемы является основной частью для хорошего дизайнера юниверса. Заполнить схему с помощью таблиц основанных на столбцах, которые относятся к объектам и необходимые для создания отчетов конечных пользователей. Данные объекты должны быть определены в необходимом анализе пользователя. Необходимо просмотреть базу данных для таблиц, которые позволяют создавать данные необходимых объектов.

5.2.2 Процесс разработки схемы и создания юниверса

Создание схемы является первой фазой стадии реализации цикла развития юниверса. Анализы пользователя и стадии планирования выполняются без помощи средства создания юниверсов; однако создание схемы – это первый шаг для создания юниверса с помощью средства проектирования.

В следующем списке показано, какое место занимает фаза разработки схемы в цикле разработки обычного юниверса (реализация, шаг 1):

- Подготовка
 1. Анализ потребностей пользователя
 2. Планирование
- Реализация с помощью средства создания юниверсов
 1. Создание и проверка схемы
 2. Создание и проверка объектов юниверса
 3. Развертывание юниверса с помощью репозитория
- Поддержка
 1. Обновление и поддержка юниверса, основанные на изменениях требований пользователя или источника данных

5.2.3 Каковы стадии проектирования схемы?

В данной главе описываются следующие стадии разработки схемы.

- Вставка и объединение таблиц
- Создание объединений и установка мощностей множеств
- Устранений проблем объединения, например циклы, ловушки разногласия и раскрытия.
- Проверка целостности схемы.

5.3 Вставка таблиц

Начните разработку схемы с помощью выбора таблиц в целевой базе данных и вставки символов, которые представляют таблицы на панели [Структура](#). В средстве создания юниверсов символы таблиц называются просто таблицами.

Используйте [обозреватель таблиц](#), чтобы выбрать таблицы для вставки в схему. [Обозреватель таблиц](#) – это независимое окно, в котором отображается дерево таблиц и столбцов целевой базы данных.

📘 Примечание

Перед выбором таблиц можно обозначить стратегии, которые необходимо использовать для создания юниверса. Дополнительные сведения по данной теме см. в разделе [Выбор стратегий](#) [страница 88].

5.3.1 Использование обозревателя таблиц

[Обозреватель таблиц](#) – это независимое окно, в котором отображается дерево таблиц и столбцов целевой базы данных. [Браузер таблицы](#) используется для выбора таблиц в базе данных для добавления в схему. Разверните узел рядом с именем таблицы, чтобы отобразить столбцы для таблицы.

5.3.1.1 Активирование обозревателя таблиц

[Обозреватель таблиц](#) невидим по умолчанию. Необходимо активировать [Обозреватель таблиц](#) при добавлении таблиц на панели [Структура](#). Можно активировать [Обозреватель таблиц](#) с помощью любого способа, указанного ниже.

Чтобы активировать [обозреватель таблиц](#), выполните следующие действия.

- Выберите ► [Вставить](#) ► [Таблицы](#) .
Или
- Дважды щелкните пустое пространство на панели [Структура](#).
Или
- Нажмите кнопку [Обозреватель таблиц](#).
На панели [Структура](#) откроется диалоговое окно [Обозреватель таблиц](#).

5.3.1.2 Вставка таблиц из обозревателя таблиц

Можно использовать один из следующих способов, чтобы вставить одну или несколько таблиц с помощью [браузера таблицы](#):

Вставка одной таблицы

Чтобы вставить одну таблицу,

- Выберите таблицу и нажмите кнопку [Вставить](#).
Или
- Щелкнув правой кнопкой мыши таблицу, выберите [Вставить](#) в контекстном меню.
Или
- Дважды щелкните таблицу.
Или
- Щелкните таблицу и перетащите ее мышью на панель [Структура](#).
Таблица отобразится на панели [Структура](#).

Вставка нескольких таблиц

Для вставки нескольких таблиц выполните следующие действия.

1. Удерживая клавишу [\[CTRL\]](#), щелкните необходимые таблицы.
Или
2. Удерживая клавишу [\[SHIFT\]](#), щелкните первую и последнюю таблицы в блоке таблиц.
После этого будут выбраны несколько таблиц.
3. Нажмите кнопку [Вставить](#).
Или
Перетащите таблицы на панель [Структура](#).
Или
Щелкнув правой кнопкой мыши выбранные таблицы, выберите "Вставить" в контекстном меню.
Каждая таблица со всеми столбцами отобразится на панели [структуры](#). В [Обозревателе таблиц](#) рядом с именем каждой таблицы, вставленной в юниверс, отображается метка.

5.3.1.3 Просмотр данных в обозревателе таблиц

С помощью [Обозревателя таблиц](#) можно просматривать данные, содержащиеся в таблице или отдельном столбце.

Чтобы просмотреть данные в [Обозревателе таблиц](#), выполните следующие действия.

1. Щелкните правой кнопкой мыши таблицу в [Обозревателе таблиц](#)
Или
Разверните узел таблицы в [Обозревателе таблиц](#), а затем щелкните правой кнопкой мыши столбец с таблицей.
2. Выберите в контекстном меню [Просмотр значений таблицы](#).
Или
Выберите в контекстном меню [Просмотр значений столбцов](#).
Данные таблицы или столбца будут отображены в открывшемся окне.

Содержимое Customer

Закрыть

cust_id	first_name	last_name	age	phone_number	address	city_id
107.0	Jack	Swenson	74.0	(202) 555 8125	64 Imagin...	19.0
106.0	William	Baker	64.0	(312) 555 7040	2890 Gra...	15.0
105.0	Tony	Goldschmidt	55.0	(619) 555 6529	91 Torre ...	14.0
104.0	Joe	Larson	45.0	(213) 555 5095	87 Carmel...	13.0
103.0	Peter	Travis	34.0	(510) 555 4448	7835 Hart...	12.0
102.0	Robin	McCarthy	29.0	(214) 555 3075	27 Pasad...	11.0
101.0	Paul	Brendt	19.0	(212) 555 2146	10 Jasper...	17.0
307.0	Priscilla	Hopkins	73.0	634 634643	The Gables	38.0
306.0	Mary	Jone	68.0	143 546456	34 Apple...	36.0
305.0	Hariett	Keegan	56.0	566 344643	10 Hamilt...	35.0
304.0	George	McCartney	47.0	323 768678	45 Glenth...	34.0
303.0	John	Wilson	34.0	158 746231	28 Sutton...	33.0
302.0	Justin	Marlow	29.0	653 643634	290 York...	32.0

☐ Точные значения

→ Совет

Если ширина столбцов недостаточна для просмотра целой строки значений, ее можно увеличить, нажав комбинацию клавиш **CTRL-SHIFT** и **+**.

5.3.1.4 Оптимизация производительности обозревателя таблиц

Время, затраченное на вставку таблицы на панели *Структура* из *обозревателя таблиц* может меняться в зависимости от следующих факторов.

Процесс вставки таблицы проходит медленно, так как...

В базе данных находится большое количество таблиц. Средство создания юниверсов запрашивает системный каталог. Если каталог имеет достаточно большой размер, процесс извлечения таблиц происходит медленно.

Оптимизируйте вставку таблицы с помощью...

Создание хранилища данных с помощью таблиц, которые необходимо вставить в учетную запись отдельной базы данных. Создание соединения с новым хранилищем.

Процесс вставки таблицы проходит медленно, так как...

Автоматически вставляйте объединения и проверяйте мощности множеств с таблицами, которые вставляете.

Оптимизируйте вставку таблицы с помощью...

Вставка только таблиц. Необходимо выполнить следующее.

1. Выберите ► [Сервис](#) ► [Параметры](#) ►
Откроется диалоговое окно "Параметры".
2. Щелкните вкладку базы данных.
Откроется страница "База данных".
3. Снимите флажки: [Извлечь объединения с таблицами](#) и [Определить мощности множеств в объединениях](#).
4. Нажмите кнопку [OK](#).

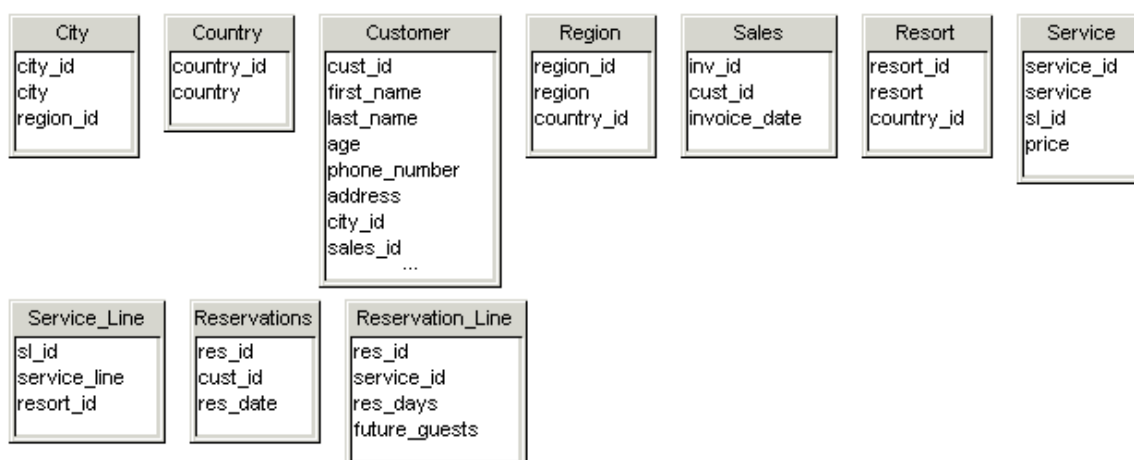
5.3.2 Упорядочивание таблиц на панели "Структура"

Можно автоматически упорядочить таблицы на панели [Структура](#), чтобы привести в порядок исходную схему перед началом ручного упорядочивания таблиц для создания объединений.

5.3.2.1 Автоматическое упорядочивание таблиц на панели структуры

Чтобы автоматически упорядочить таблицы,

- Выберите ► [Вид](#) ► [Упорядочить таблицы](#) ►.
Отображение таблиц будет упорядочено.



5.4 Использование производных таблиц

Производные таблицы являются таблицами, которые определяют в схеме юниверса. На их основе можно создавать объекты. Производная таблица определяется с помощью запроса SQL на уровне юниверса, который может использоваться в качестве логической таблицы в средстве создания юниверсов.

У производных таблиц есть следующие преимущества.

- Сокращенный объем данных, возвращенных в документ для анализа.
В производную таблицу можно включать сложные вычисления и функции. Данные операции выполняются до возврата набора результатов в документ, что сохраняет время и уменьшает потребность в сложном анализе большого объема данных на уровне отчета.
- Уменьшенные таблицы итогов обслуживания базы данных.
Производные таблицы могут, в некоторых случаях, перемещать статистические таблицы, в которых содержатся результаты сложных вычислений, объединенных в юниверсе с помощью поддержки агрегирования. Данные таблицы необходимо поддерживать и часто обновлять. Производные таблицы могут возвращать одинаковые данные и предоставлять анализ данных в режиме реального времени.

Производные таблицы аналогичны представлениям базы данных с преимуществом в том, что SQL данной таблицы может включать подсказки.

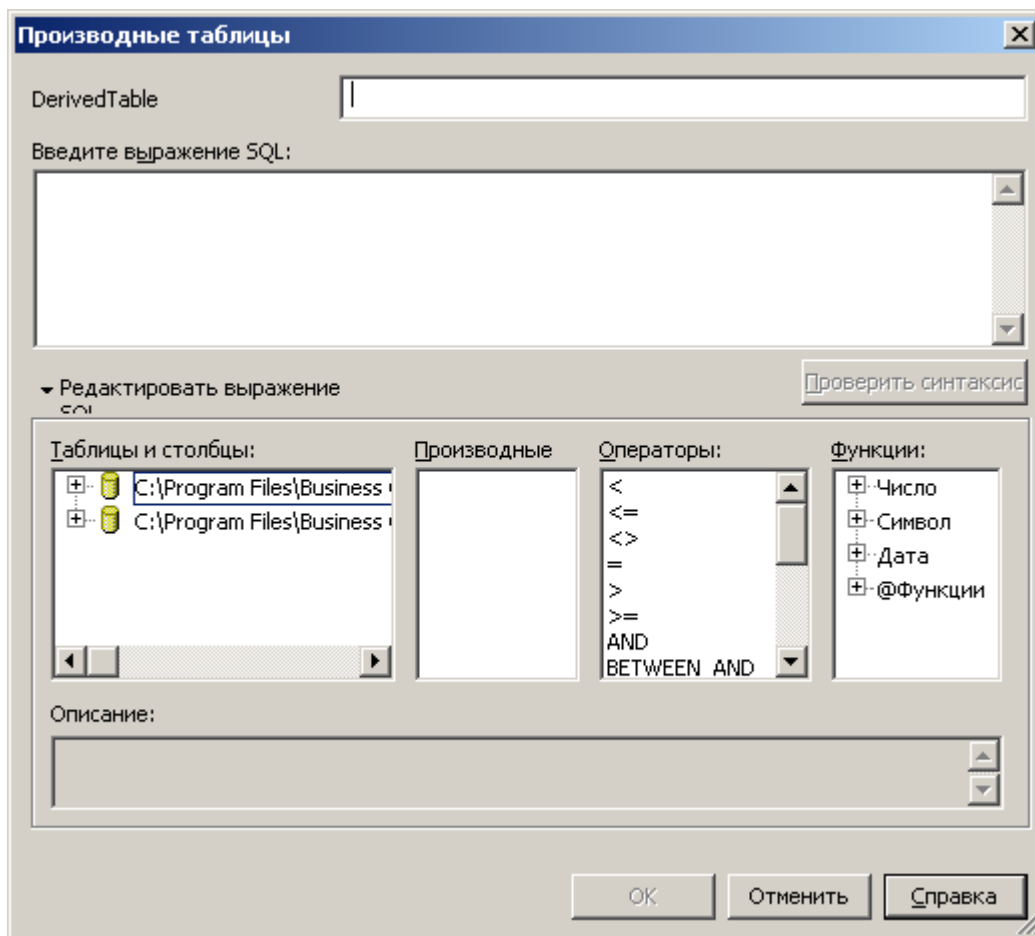
5.4.1 Добавление, редактирование и удаление производных таблиц

Производные таблицы отображаются в схеме средства создания юниверсов таким же образом, что и таблицы обычной базы данных, но рабочий процесс их создания отличается. Добавление, редактирование и удаление производных таблиц описано в следующих разделах.

5.4.1.1 Добавление производной таблицы

Чтобы добавить производную таблицу,

1. Выберите [Производные таблицы](#) в меню [Вставка](#).
Откроется диалоговое окно [Производные таблицы](#).



2. Введите имя таблицы в поле *Имя таблицы*.
3. Создайте таблицу SQL в поле, расположенном ниже поля *Имя таблицы*.
Можно сразу ввести SQL или использовать поля "Таблицы", "Столбцы", "Операторы" и "Функции", чтобы создать ее.
4. Нажмите кнопку *ОК*.
Производная таблица отобразится в схеме с физическими таблицами базы данных.
5. Создайте объекты на основе столбцов производной таблицы таким же образом, как и на основе обычной таблицы.

5.4.1.2 Редактирование производной таблицы

Чтобы редактировать производную таблицу,

1. Щелкните правой кнопкой мыши таблицу в схеме средства создания юниверсов и выберите команду *Редактировать производную таблицу* в контекстном меню.
2. Отредактируйте производную таблицу и нажмите *ОК*.

5.4.1.3 Удаление производной таблицы

1. В панели схемы выберите производную таблицу, которую требуется удалить.
2. Нажмите Delete.

5.4.1.4 Пример. Создание производной таблицы

Пример: Создание произвольной таблицы для возврата информации сервера

В этом примере необходимо создать объекты для добавления пользователем информации о сервере баз данных для их отчетов. Создайте два объекта, `servername` и `version`, которые вернут значения внутренних переменных `@@SERVERNAME` и `@@VERSION` в юниверсе, запущенном в базе данных SQL Server.

Выполните следующие действия:

1. Выберите *Производные таблицы* в меню *Вставить*.
Откроется диалоговое окно *Производные таблицы*.
2. Введите `<serverinfo>` в поле *Имя таблицы*.
3. Введите SQL `Select @@SERVERNAME` в качестве имени сервера, `@@VERSION` в качестве версии в поле SQL.

ⓘ Примечание

Необходимо предоставить псевдонимы в SQL для всех производных столбцов. Средство создания юниверсов использует эти псевдонимы для присвоения имен столбцам производных таблиц.

4. Нажмите кнопку *OK*.
Производная таблица `serverinfo` отображается в схеме средства создания юниверсов.
5. Создайте класс, называемый информацией сервера и добавьте два объекта измерений под классом на основе столбцов имени сервера и версии производного `serverinfo` и столбцов таблицы. Убедитесь, что таблица `serverinfo` появится в списке таблиц как любая таблица обычной базы данных, а ее столбцы появятся в списке столбцов как обычные столбцы таблицы.

Теперь пользователь может размещать объекты `servername` и `version` в отчете.

Пример: Отображение числа регионов в каждой стране

В этом примере создайте таблицу, которая показывает количество регионов в каждой стране. SQL выглядит следующим образом.

выбрать страну,

счет (`r.region_id`) как `number_of_regions`

из страны *c*,

регион *r*

где *r.country_id = c.country_id*

сгруппировать по странам

В этом случае важно дать псевдоним столбцу с вычислением. Средство создания юниверсов использует эти псевдонимы в качестве имен столбцов в производной таблице. В этом случае таблица содержит два столбца: *country* и *number_of_regions*.

5.5 Вложенные производные таблицы

Вложенная производная таблица (также 'производная таблица в производной таблице') – это таблица, выведенная из минимум одной существующей производной таблицы. Вложенная производная таблица также содержит отсылки к таблицам базы данных.

Используйте редактор [Производные таблицы](#), чтобы ввести выражение SQL и выбрать производные таблицы (и физические таблицы в базе данных, если необходимо), чтобы создать вложенные производные таблицы. При создании отчета выражение SQL для производной таблицы вставляется в выражение SQL для вложенной производной таблицы.

5.5.1 Использование редактора "Производные таблицы"

Используйте редактор [Производные таблицы](#) для создания производных таблиц или вложенных производных таблиц. Для создания в редакторе SQL выражений для производной таблицы или вложенной производной таблицы введите SQL выражения и дважды щелкните объект (таблицы, производные таблицы, столбцы, функции).

Для привязки производной таблицы к вложенной производной таблице используйте функцию `@DerivedTable`.

- Функция `@DerivedTable (Derived_table_name)` находится в каталоге редактора "Производные таблицы".
- Центральная панель в нижней части редактора "Производные таблицы" отображает существующие производные таблицы и вложенные производные таблицы. Данная панель отображается только когда производные таблицы существуют в юниверсе.

При нажатии [Проверка целостности](#) выполняются следующие проверки для производных и вложенных производных таблиц:

- При удалении зависимой производной таблицы изменения сказываются на производных таблицах.
- Проверки на наличие циклических ссылок
- Проверка запрещенных `@DerivedTable ()` в определениях объектов (`SELECT` и `WHERE`).

5.5.2 Как создать вложенную производную таблицу

Вложенная производная таблица создается тем же способом, что и производная таблица. Вложенная производная таблица добавляется и переименовывается тем же способом, что и производная таблица.

Для создания вложенной производной таблицы выполните следующие действия.

1. Откройте нужный юниверс (*.unv) в каталоге образцов (Business Objects\BusinessObjects Enterprise 12\Samples\en\UniverseSamples).
2. Щелкните правой кнопкой мыши панель [Структура юниверса](#) и выберите пункт "Производная таблица" в контекстном меню.
Откроется редактор [Производные таблицы](#), и на центральной панели в нижней части окна редактора [Производные таблицы](#) будут перечислены доступные производные таблицы.
3. Наберите имя нужной вложенной производной таблицы.
4. Наберите выражение SQL. Можно набрать весь текст или воспользоваться вспомогательными средствами редактора.
5. Дважды щелкните нужные объекты (таблицы, производные таблицы, столбцы, функции).
6. Используйте функцию @DerivedTable по следующим синтаксическим правилам:
@DerivedTable(Derived_table_name) для выбора производной таблицы.
7. Чтобы проверить синтаксис данной производной таблицы и исправить возможные ошибки, а затем подтвердить правильность вложенной производной таблицы, нажмите кнопку [Проверить синтаксис](#).
Вложенная производная таблица теперь добавлена в юниверс.
8. Нажмите [OK](#) для проверки правильности данной вложенной производной таблицы.
На панели [Структура](#) отобразится вложенная производная таблица. Производные и вложенные производные таблицы выделены более светлым цветом по сравнению с таблицами, которые представляют реальные таблицы базы данных.

❗ Примечание

Чтобы вывести значения таблиц, щелкайте на них правой кнопкой мыши.

Такова процедура создания вложенной производной таблицы в юниверсе.

5.5.3 Переименование вложенных производных таблиц

При переименовании производной таблицы новое имя распространяется и обновляется во всех во всех производных таблицах, на которые оно ссылается.

5.6 Использование таблиц, имеющих столбцы ввода данных

Если вставить в юниверс таблицу со столбцом ввода данных, пользователю Web Intelligence или Query as a Web Service необходимо выбрать или ввести значения, чтобы вычислить столбцы ввода данных.

Столбцы ввода связаны со значениями. Данные для столбцов ввода данных не всегда доступны в исходной базе данных и могут быть следующими:

- Жестко запрограммированными значениями при создании юниверса
- Значениями, введенными конечным пользователем (после подсказки) или выбранными из списка
- Значениями, полученными после объединения с другой таблицей

Таблицы со столбцами ввода данных поддерживаются, только если соединение с базой данных является сервером Business Objects Data Federator.

В процессе анализа объединений, которые отвечают за вычисление столбца ввода, обратите внимание, что:

- Только простые объединения будут рассматриваться в алгоритме решения.
- Не разрешается использование сложных объединений, например, объединений с несколькими столбцами слева или справа
- Рассматриваются только объединения с равными операндами или IN-операндами (INLIST).
Невозможно использовать операторов, например Between, чтобы вычислить столбец ввода данных.

В таблицах со столбцами ввода данных есть указатель рядом со столбцом ввода на панели [Структура](#) и на панели [Обозреватель таблиц](#). Столбцы ввода данных обозначаются определенным значком.

При вставке таблицы со столбцом ввода данных, используйте редактор [Столбцы ввода данных](#), чтобы ввести параметры.

❗ Примечание

При добавлении таблицы в юниверс необходимо присваивать столбцам ввода данных значения по умолчанию.

Данная функция используется в следующих продуктах и компонентах:

- Web Intelligence
- Query as a Web Service

Связанные сведения

[Определение жестко запрограммированного списка значений \[страница 143\]](#)

[Определение списка значений для последующего их выбора или ввода пользователем \[страница 144\]](#)

5.6.1 Определение жестко запрограммированного списка значений

База данных содержит как минимум одну таблицу, содержащую один или несколько столбцов ввода данных.

Жестко запрограммированный список значений используется в качестве входных данных для таблицы, чтобы определить значение(я) столбца(ов) входных данных. Конечный пользователь не вводит никаких значений. Выполните следующие действия, чтобы определить список значений.

1. Выберите таблицу в базе данных и добавьте ее на панель [Структура](#) средства создания юниверсов. Отобразится редактор [Столбцы ввода данных](#).
2. В редакторе [Столбцы ввода данных](#) выберите параметр.
3. В поле [Значение](#) введите значение или список значений. Заключите каждое значение в двойные кавычки, а затем разделите значения в кавычках точкой с запятой (;). Значения появятся в столбце [Значение](#).
4. В списке [Следующий запуск](#) выберите [Использовать данное значение](#). [Использовать данное значение](#) отобразится в столбце [Следующий запуск](#).
5. Нажмите [ОК](#).

Таблица отображается на панели [Структура](#) средства создания юниверсов. Столбец ввода будет отмечен стрелкой.

5.6.2 Определение списка значений для последующего их выбора или ввода пользователем

База данных содержит как минимум одну таблицу, содержащую один или несколько столбцов ввода данных.

Пользователь может ввести или выбрать значение в списке значений, которое будет использоваться в таблице, чтобы определить значение(я) для столбца(ов) ввода. Выполните следующие действия, чтобы определить значения для таблицы столбца ввода в схеме.

1. Выберите таблицу в базе данных и добавьте ее на панель [Структура](#) средства создания юниверсов.
2. В редакторе [Столбцы ввода данных](#) выберите параметр.
3. В списке [Следующий запуск](#) выберите [Запрашивать значение](#).
Выполнив подсказку Web Intelligence или Query as a Web Service, пользователь получит запрос на выбор значения из связанного списка значений.
4. В окне [Редактирование наименований подсказок](#) отредактируйте подсказку по умолчанию для конечного пользователя.
5. Нажмите [Обзор объектов юниверса](#), чтобы выбрать список значений в юниверсе.
Если необходимо удалить объект в списке значений, выбранных для установки, на панели [Выбранные объекты](#) выберите объект и нажмите [Удалить](#).
6. Нажмите кнопку [ОК](#).

Таблица отображается на панели [Структура](#) средства создания юниверсов. Столбец ввода будет отмечен стрелкой. В [обозревателе таблиц](#) столбец ввода отмечается определенным значком.

5.7 Определение объединения

После вставки нескольких таблиц в схему необходимо создать объединения между связанными таблицами. Объединения также важны, как и таблицы в схеме, так как позволяют комбинировать данные из нескольких таблиц смысловым способом.

5.7.1 Что такое объединение?

Объединение – это условие, которое связывает данные в отдельных, но связанных таблицах. Таблицы обычно имеют отношение типа "родительская-дочерняя". Если запрос не содержит объединение, база данных возвращает результирующее множество, которое содержит все возможные комбинации строк из таблиц, указанных в запросе. Такой набор результатов известен как декартово произведение и редко используется.

Например, декартово произведение запроса, ссылаясь на две таблицы с 100 и 50 строками соответственно, содержит 5000 строк. В большой базе данных или запросах с большим количеством таблиц декартовы произведения быстро становятся неуправляемыми. В средстве создания юниверсов объединения представлены в виде линий, связывающих таблицы в схеме.

5.7.2 Почему объединения используются в схеме?

Объединения используются в схеме, чтобы обеспечить возврат правильных результатов из нескольких таблиц. Объединение между двумя таблицами определяет способ возврата данных, если обе таблицы включены в запрос.

Каждая таблица в схеме содержит данные в одной или нескольких столбцах, соответствующих требованиям пользователя. В производстве юниверса пользователи Web Intelligence, возможно, захотят выполнить запросы, сочетающие количество различных объектов (каждый подразумевает столбец) возврата данных из любой комбинации таблиц.

Связывание всех таблиц в схеме с объединениями обеспечивает ограниченное количество способов сочетания данных из столбцов различных таблиц в запросе. Объединения ограничивают сочетания столбцов между таблицами с совпадающими или общими столбцами. Это не допускает возвращения данных, содержащих информацию из столбцов, которые нет смысла сопоставлять.

📌 Примечание

Всегда создавайте объединения на панели структур. Объединения, созданные не на панели "Структура", например объединение, определенное вручную для объекта утверждения Where, создаются во время выполнения, поэтому они не учитываются средством создания юниверсов при проверке целостности и обнаружении контекста. Информация для этих процессов требуется во время разработки. Контексты и целостность юниверса будут рассмотрены далее в этой главе..





5.7.3 Объединение SQL

По умолчанию средство создания юниверсов неявно указывает объединение в утверждении WHERE путем ссылки на совпадающие или общие столбцы таблиц.

Как правило, для каждой пары объединяемых таблиц существует одно утверждение WHERE. Следовательно, для объединения четырех таблиц необходимо три оператора WHERE.

Результат запроса выполняется с учетом двух таблиц, объединенных в одну со всеми представленными в них столбцами. В каждой строке такой таблицы отображаются данные строк различных вложенных таблиц с соотношением значений общих столбцов.

5.7.3.1 Поддержка стандарта ANSI 92

Если целевая реляционная СУБД поддерживает стандарт ANSI 92, можно указать для параметра юниверса ANSI92 ( [Файл](#)  [Параметры](#)  [Параметр](#) ) значение "Да", чтобы включить поддержку ANSI 92 для объединений, созданных в схеме. При поддержке юниверсом стандарта для объединения ANSI 92 новые объединения указываются в утверждении FROM. Также можно выбрать выведенные столбцами объекты для включения в утверждение FROM. Поддержка ANSI 92 описывается в разделе [Поддержка ANSI 92 для объединений в юниверсе](#). [\[страница 159\]](#).

5.7.4 Что отсутствует в таблицах для объединения?

Необходимо объединить все таблицы в схеме, которые включены в SQL, сгенерированной объектами в запросах Web Intelligence выполняемых по данным юниверса. Следующие типы таблиц являются исключениями.

- Базовые таблицы из схемы, отображающиеся для каждого использования. Это оригинальные таблицы, для которых были созданы псевдонимы для переименования или для решения проблем объединения. Это базовые таблицы, которые обычно не используются в любом определении объекта.
- Таблицы, которые являются целью синтаксиса поддержки агрегирования (в зависимости от случая). Например, агрегатные таблицы в примере юниверса efashion (их имена начинаются с "Agg_") не объединяются ни с одной таблицей в схеме.

5.7.5 Объединение первичного и внешнего ключей

Как правило, объединение создается между первичным ключом одной таблицы и внешним ключом другой таблицы. Объединение также можно создать между двумя первичными ключами. Необычно, если хотя бы в одну часть объединения не включен первичный ключ таблицы.

Необходимо понимать конструкцию каждого ключа в базе данных. Ключи с несколькими столбцами могут влиять на способ установки мощности множеств для объединений, а это может влиять на способ установки контекста в схеме.

Определение и использование контекста описано в [Определение и устранение проблем объединений](#) [\[страница 199\]](#)

5.7.5.1 Отображение ключей

Можно отображать первичные и внешние ключи во всех таблицах на панели структуры. Столбцы ключей отображаются с подчеркиванием в каждой таблице к ключами. При выборе параметра для отображения ключей необходимо обновить структуру до отображения ключей с подчеркиванием.

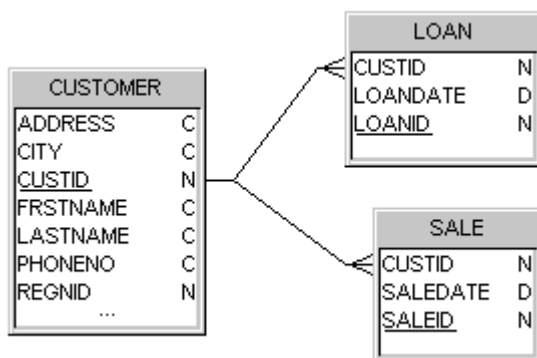
Возможность отображать столбцы ключей с подчеркиванием зависит от первичных ключей, определенных в целевой базе данных.

❗ Примечание

При отображении столбцов ключей с подчеркиванием сведения хранятся в файле .UNV. Эта информация утеряна при экспорте юниверса в репозиторий сервера центрального управления. Необходимо повторно отображать ключи юниверса при каждом импорте.

Отображение ключей.

1. Выберите Инструменты > Параметры.
На главной странице откроется диалоговое окно "Параметры".
2. Выберите вкладку "Графическое отображение".
Откроется страница "Графическое отображение".
3. Установите флажок "Подчеркнуть ключи" в групповом окне "Столбцы".
4. Нажмите кнопку ОК.
Необходимо обновить структуру до отображения столбцов ключей с подчеркиванием.
5. Выберите "Вид" > "Обновить структуру".
Обновляется структура базы данных. Столбцы ключей в схеме подчеркнуты как показано ниже.



5.7.6 Понимание элементов объединения

Мощность множества описывает объединение 2 таблиц с помощью определения количества совпадающих строк в таблицах. Это очень важно для определения проблем объединения и для создания контекстов, чтобы исправить ограничения целевой структуры RDBMS.

Необходимо установить мощности множества для каждого объединения в схеме. Средство создания юниверсов может автоматически определять и устанавливать свойства связей, но необходимо всегда вручную проверять эти свойства, принимая во внимание сущность объединяемых ключей.

Установки и использование мощности множества описаны в разделе [Использование мощности множеств \[страница 180\]](#).

5.7.7 Создание объединений

Существует несколько подходов к созданию объединений с помощью средства создания юниверсов:

- Ручное вычерчивание объединений в схеме
- Непосредственное определение свойств объединений
- Выбирать автоматически определяемые объединения.
- Автоматически создавать объединения на вставке таблиц.

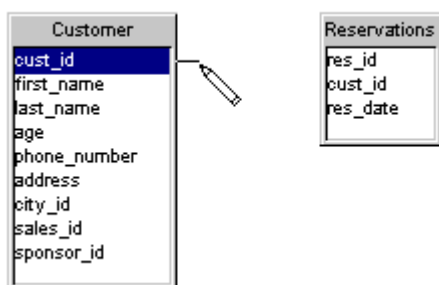
Подробное описание этих подходов см. ниже.

5.7.7.1 Ручное вычерчивание объединений в схеме

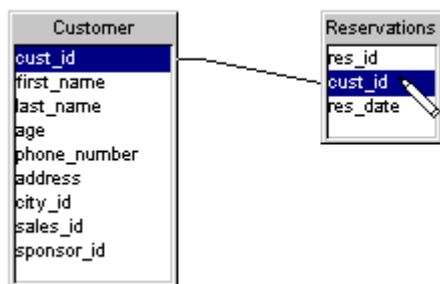
Существует возможность графического создания отдельных объединений между таблицами путем вычерчивания с помощью мыши линии от столбца одной таблицы до соответствующего столбца в другой таблице.

Создание объединений ручным вычерчиванием.

1. Поместите указатель мыши над столбцом, который будет одним краем объединения.
Указатель мыши примет форму руки.
2. Нажмите и удерживайте нажатой левую кнопку мыши.
Выбранный столбец будет подсвечен.
3. Переместите указатель мыши к столбцу, который будет другим краем объединения.
По завершении этой операции указатель примет вид карандаша.



4. Поместите символ карандаша над конечным столбцом.
Этот столбец будет подсвечен.



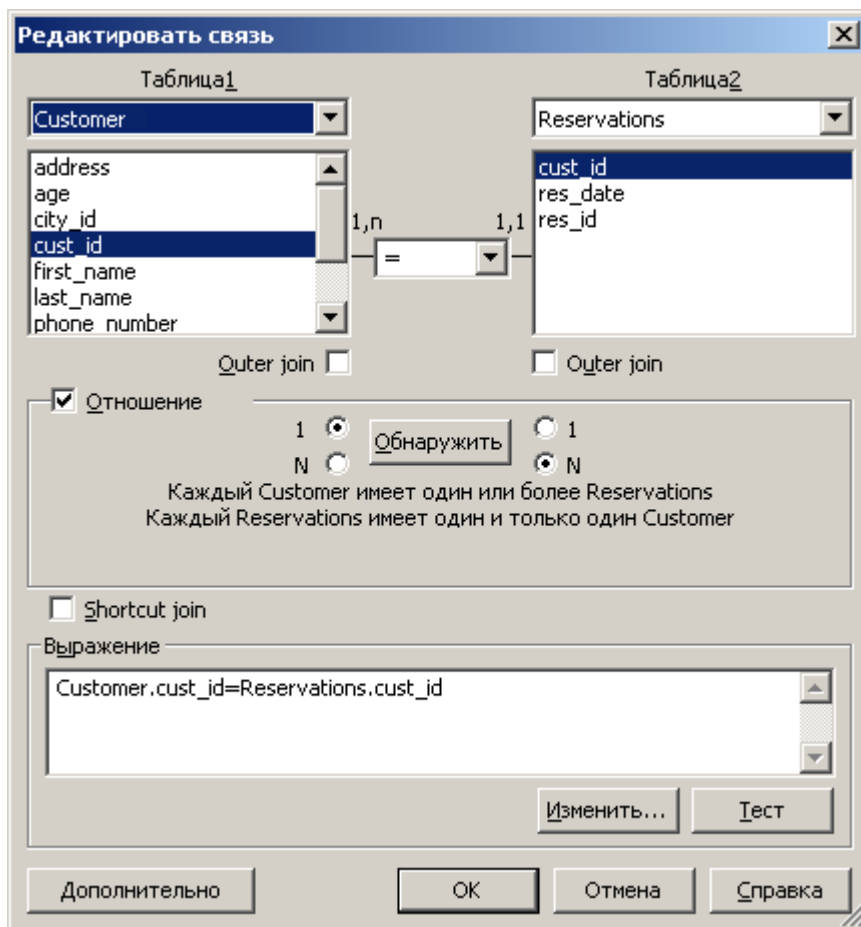
5. Отпустите кнопку мыши.
Объединение двух таблиц теперь создано.
6. Дважды щелкните мышью новое объединение.
Откроется диалоговое окно "Редактировать объединение". В нем выводятся свойства объединения. Свойства, которые можно установить для объединения, включающего мощность множества и тип объединения, описаны в разделе [Свойства объединения \[страница 153\]](#).
7. Введите и выберите свойства объединения.
8. Нажмите кнопку [OK](#).

5.7.7.2 Непосредственное определение свойств объединений

Можно создать объединение, непосредственно определив его свойства в диалоговом окне "Редактировать объединение".

Непосредственное создание объединения.

1. Выберите пункт меню "Вставка" > "Объединение".
Или
Нажмите кнопку "Вставить объединение".
Откроется диалоговое окно "Редактировать объединение".



2. Выберите таблицу из выпадающего списка "Таблица1".
В поле списка под именем таблицы появятся столбцы для выбранной таблицы.
3. Выберите имя столбца, который будет одним краем объединения.
4. Выберите таблицу из выпадающего списка "Таблица2".
В поле списка под именем таблицы появятся столбцы для выбранной таблицы.
5. Выберите имя столбца, который будет другим краем создаваемого объединения.
Свойства, которые можно установить для объединения, включающего оператора объединений, мощность множества и тип объединения, описаны в разделе [Свойства объединения \[страница 153\]](#).
6. Введите и выберите свойства объединения.
7. Нажмите кнопку [OK](#).
Новое объединение появится в схеме, связывающей эти две таблицы, определенные в диалоговом окне "Редактировать объединение".

5.7.7.3 Выбор автоматически определяемых объединений

Можно использовать функцию средства создания юниверсов "Определять объединения" для автоматического определения выбранных объединений в схеме. Средство проектирования определяет

имена столбцов в целевой базе данных и предлагает вероятные объединения для таблиц в схеме. Можно выбрать или принять все предлагаемые объединения для создания.

Как осуществляется автоматическое определение объединений?

Определение объединений основано на стратегии объединений, отображенной на странице стратегий в диалоговом окне "Параметры" ("Файл" > "Параметры" > вкладка "Стратегии").

Стратегия представляет собой файл скрипта, который автоматически извлекает структурную информацию из базы данных. Существует определенное количество встроенных стратегий, которые поставляются в комплекте со средством создания юниверсов. Они перечислены в полях раскрывающихся списков на странице "Стратегии" в диалоговом окне "Параметры".

Автоматическое определение объединения по умолчанию определяет объединения на основании соответствия имен столбцов, информация ключей при этом не используется. Здесь можно выбрать ту стратегию объединения, которая будет использована при осуществлении автоматического определения объединения.

❗ Примечание

Дополнительные сведения об использовании стратегий см. [Выбор стратегий \[страница 88\]](#).

Использование автоматического определения объединения должным образом

Автоматическое определение объединений используется для быстрого создания объединений в схеме. Тем не менее, необходимо помнить об ограничениях автоматического определения объединения при составлении схемы.

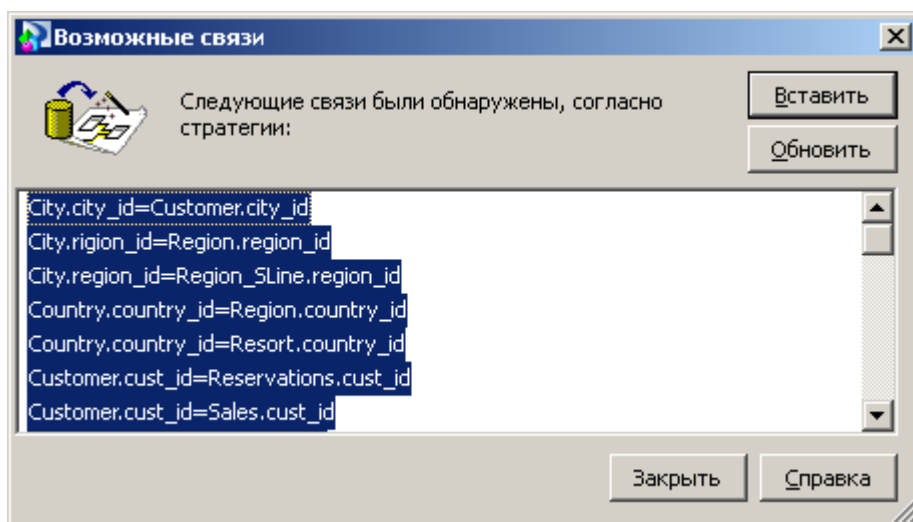
Стратегии объединения используются для определения возможных объединений соответствующих имен столбцов из базы данных. В целевой базе данных могут находиться экземпляры, в которых имена первичного и внешнего ключей и других столбцов объединений различаются в зависимости от таблицы. Эти столбцы не будут выбираться средством создания юниверсов. Необходимо всегда проверять вручную каждое автоматически определенное объединение, создание которого подтверждается. Следует помнить о том, что какие-то объединения могут не быть определены.

Создание объединения с помощью автоматического определения.

1. Убедитесь, что в окне раскрывающегося списка объединений в диалоговом окне "Параметры" выбрана стратегия объединения, которую предполагается использовать для определения объединений. Это можно проверить следующим способом.
 - Выберите "Файл" > "Параметры" и нажмите на вкладку "Стратегии".
 - Выберите стратегию, которую предполагается использовать для определения объединений в окне раскрывающегося списка объединений, и нажмите "ОК".
2. Выберите несколько таблиц на панели структуры.

Можно выбрать нужные таблицы, щелкнув каждую из них и удерживая клавишу SHIFT, или выбрать все таблицы в зоне, щелкнув пустое пространство и растянув прямоугольную зону выделения на нужное количество таблиц.

3. Выберите "Инструменты" > "Автоматическое определение" > "Определить объединения".
Или
Нажмите кнопку "Определить объединения".
Отобразится диалоговое окно вероятных объединений. В нем отображается список вероятных или предлагаемых объединений для выбранных таблиц. Вероятные объединения также отображаются синими линиями между выбранными таблицами в панели структуры.



4. Для создания всех вероятных объединений нажмите "Вставить".
5. Или
Выберите одно или несколько объединений и нажмите "Вставить".
Чтобы выбрать одно или несколько объединений, щелкните отдельные таблицы, удерживая клавишу CTRL, или щелкните первое и последнее объединение в непрерывном блоке, удерживая клавишу SHIFT.
Объединения добавляются в схему.
6. Нажмите кнопку "Закреть".

5.7.7.4 Автоматическая вставка объединений с помощью связанных таблиц

Существует возможность автоматически вставлять объединения в схему одновременно со вставкой таблиц, использующих эти объединения, в панель структур. Автоматическое создание объединения определяется двумя процессами.

- Активная стратегия объединения определяет данные столбца, используемые для определения объединения.
- Для автоматического создания объединений из связанных ими таблиц необходимо установить параметр по умолчанию "Извлекать объединения из таблиц". Этот параметр находится на странице "База данных" в диалоговом окне "Параметры".

Ограничения при вставке объединений автоматически

Автоматическая вставка объединений в схему со связанными таблицами – это быстрый способ поместить объединения в схему, но это может привести к серьезным ошибкам в ее проектировании. Вставляемые объединения основываются на структуре базы данных, поэтому столбцы, являющиеся общими для нескольких таблиц, которые были переименованы в базе данных, не будут учтены.

Этот прием не следует использовать для создания объединений в порождающем юниверсе. Вместо этого применяйте его для демонстрационных целей или как быстрый способ построения юниверса, в котором затем будет проверяться каждое объединение после вставки.

Автоматическое создание объединений со связанной таблицей.

1. Убедитесь, что на странице "Стратегии" диалогового окна "Параметры" выбрана стратегия объединения, которую предполагается использовать для определения объединений.
2. Выберите "Инструменты" > "Параметры".
Откроется диалоговое окно "Параметры".
3. Выберите вкладку "База данных".
Откроется страница "База данных".
4. Установите флажок "Извлекать объединения из таблиц".
5. Нажмите кнопку ОК.
Теперь, когда вставлена таблица, в которой есть столбцы, ссылающиеся на другие столбцы в таблицах, уже вставленных в панели "Структура", ссылки между таблицами автоматически вставляются как объединения между соответствующими таблицами.

5.7.8 Свойства объединения

В диалоговом окне "Правка объединений" можно задать параметры объединений. Для объединения могут быть определены следующие свойства.

Свойство	Описание
Таблица 1	Таблица у левого края объединения. В раскрывающемся списке представлены столбцы для выбранной таблицы.
Таблица 2	Таблица у правого края объединения. В раскрывающемся списке представлены столбцы для выбранной таблицы.
Оператор	Оператор, устанавливающий способ объединения таблиц. Операторы доступные для объединения описаны в разделе Операторы объединений [страница 154] .
Внешнее объединение	Определяет, какая из таблиц содержит несоответствующие данные в отношениях внешних объединений. Внешние объединения полностью описаны в разделе Создание тета-объединений [страница 169] .

Свойство	Описание
Мощность множества	Позволяет задать мощность множества объединения. Определение и использование мощности множеств описано в разделе Использование мощности множеств [страница 180].
Объединение быстрого доступа	Устанавливает объединение в качестве объединения быстрого доступа. Сведения об объединениях быстрого доступа содержатся в разделе Ограничения для использования внешних объединений [страница 174].
Выражение	Утверждение WHERE используется для ограничения данных, которые возвращаются, когда две объединенные таблицы включены в запрос.
Дополнительно	<p>Доступно, если в юниверсе активирована поддержка ANSI 92. При нажатии отображается второе окно свойств объединения, в котором перечислены объекты, надстроенные над столбцами двух таблиц объединения. Здесь можно выбрать объекты для включения в утверждение FROM.</p> <p>Дополнительные сведения об активации поддержки ANSI 92 для синтаксиса объединения см. в разделе Поддержка ANSI 92 для объединений в юниверсе. [страница 159].</p>

5.7.8.1 Операторы объединений

В раскрывающемся списке между окнами Таблицы 1 и Таблицы 2 может быть выбран оператор объединения. Оператор позволяет задать ограничение, которое используется объединением для приведения в соответствие данных между объединенными столбцами.

Для объединения могут быть заданы следующие операторы.

Оператор	Описание
=	равно
!=	не равно
>	больше
<	меньше
=	больше или равно

Оператор	Описание
=	меньше или равно
В промежутке	– между (тета-объединения)
Комплекс –	комплекс отношения

5.7.8.2 Редактирование и разбор

В диалоговом окне "Редактирование объединения" доступны две функции, позволяющие редактировать и проверять синтаксическую конструкцию объединения.

Редактировать

Кнопка "Редактировать" открывает редактор SQL. Данный графический редактор используется для изменения синтаксиса в таблицах, столбцах, операторах и функциях объединения. Дополнительные сведения об использовании редактора см. в разделе [Использование редактора SQL-кода объединения \[страница 157\]](#).

Разбор

Кнопка "Разбор" начинает разбор функции, которая подтверждает синтаксис SQL выражения объединения. Если разбор проведен успешно, выводится сообщение "ОК". При обнаружении ошибки средство создания юниверсов выводит сообщение об ошибке с указанием источника проблемы.

5.7.9 Редактирование объединения

Для редактирования объединения можно использовать следующие способы.

- Изменить свойства объединения в диалоговом окне "Редактировать объединение".
- Изменить синтаксис SQL объединения непосредственно в редакторе SQL объединения.
- Изменить синтаксис SQL объединения непосредственно в строке формул.

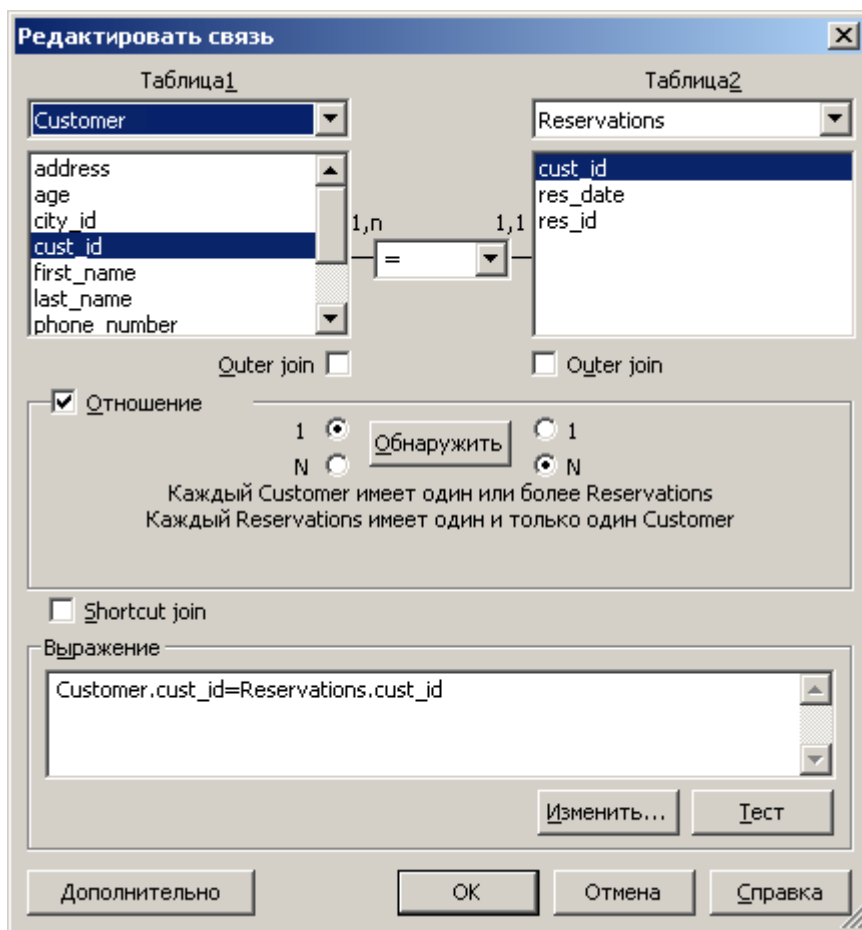
Каждый из этих способов описывается в данном разделе.

5.7.9.1 Использование диалогового окна "Редактировать объединение"

Для определения и редактирования свойств объединения можно использовать диалоговое окно "Редактировать объединение". Для редактирования синтаксиса непосредственно из этого диалогового окна можно также использовать редактор SQL объединения. Свойства объединения описаны в разделе [Свойства объединения \[страница 153\]](#).

Редактирование объединения с помощью диалогового окна "Редактировать объединение".

1. Выберите двойным щелчком мыши объединение на панели структуры.
Или
Щелкните объединение и выберите "Правка" > "Объединение".
Откроется диалоговое окно "Редактировать объединение".



2. Выберите оператора из раскрывающегося списка между таблицами.
3. Выберите другие необходимые свойства.
4. Если объединение определено с помощью синтаксиса ANSI 92, нажмите кнопку "Дополнительно".
5. Нажмите кнопку OK.

→ Совет

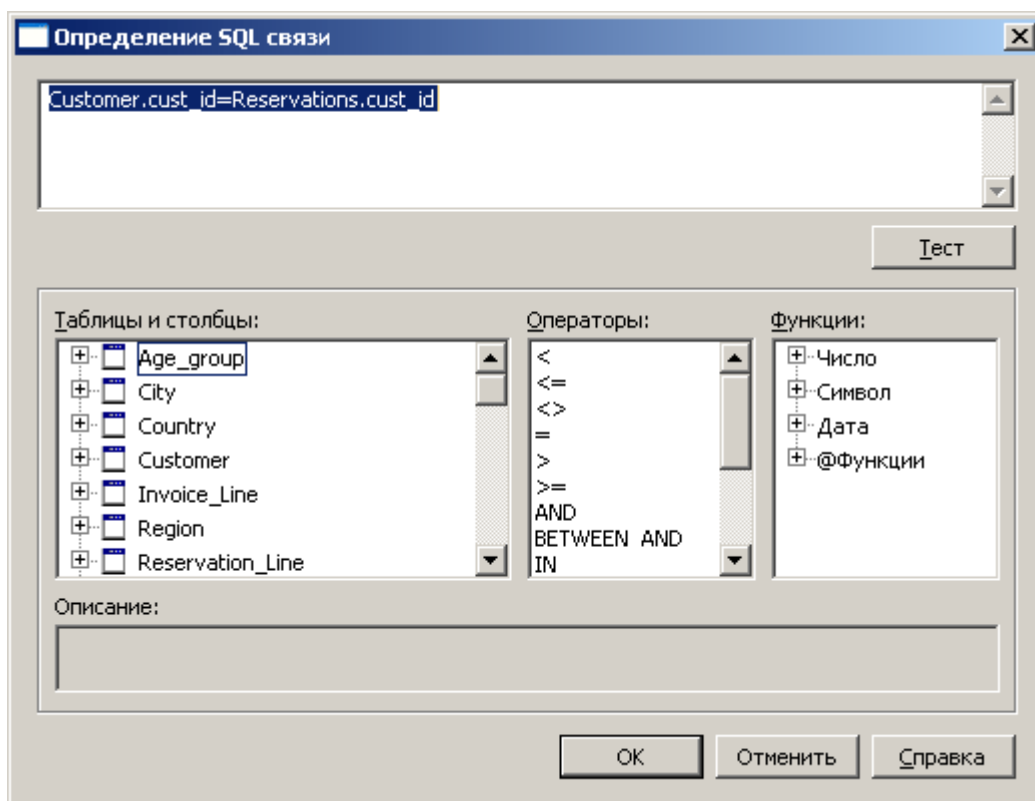
Для редактирования SQL непосредственно для объединения нажмите кнопку "Редактировать" и "Использование редактора SQL объединения". См. раздел [Использование редактора SQL-кода объединения \[страница 157\]](#) для получения дополнительной информации.

5.7.9.2 Использование редактора SQL-кода объединения

Данный графический редактор используется для непосредственного изменения синтаксиса SQL объединения. Вызов редактора производится в диалоговом окне "Редактировать объединение".

Изменение объединения с помощью редактора SQL объединения.

1. Выберите двойным щелчком мыши объединение на панели структуры.
Или
Щелкните объединение и выберите "Правка" > "Объединение".
Откроется диалоговое окно "Редактировать объединение".
2. Нажмите кнопку "Обработать".
Отобразится окно определение SQL объединения. Выражение SQL объединения отобразится в текстовом окне.



3. В поле редактирования щелкните выражение объединения в том месте, где требуется добавить или изменить синтаксис SQL.

Для изменения или добавления синтаксиса SQL можно использовать следующие возможности редактирования.

Вы хотите...	Затем выполните следующее...
Внесите изменение в столбец любого конца объединения	<ul style="list-style-type: none"> Разверните узел таблицы в поле "Таблицы и столбцы". Дважды щелкните имя столбца.
Изменение оператора с помощью объединения	Дважды щелкните оператор в поле "Операторы".
Использование функции в объединении	<ul style="list-style-type: none"> Разверните узел семейства функций. Дважды щелкните функцию.

В определении объединения будут отображены столбец, оператор или функция.

4. Нажмите кнопку "ОК".

❗ Примечание



Функции @Select и @Where невозможно использовать для ограничений доступа к строкам.



Связанные сведения

[Создание ограничения \[страница 512\]](#)

5.7.9.3 Использование строки формул

Строка *Формула* – это текстовое поле, расположенное под окном *Юниверс*, в котором отображается формула или выражение любого выбранного объединения на панели *Структура* или выбранный объект на панели *Юниверса*. Используйте три кнопки редактирования, расположенные слева от строки формул.

Кнопка "Редактировать"	Описание
	Отмена последнего невыполненного изменения. После выполнения нескольких изменений для выражения объединения без проверки изменений нажмите кнопку <i>Отмена</i> для возврата выражения к оригинальному состоянию. Чтобы отменить какие-либо отдельные изменения, необходимо использовать параметр  или нажать кнопку <i>Отмена</i> .

Кнопка "Редактировать"	Описание
	Подтвердите выражение. Применяются все изменения для выражения объединения. Можно отменить изменения после подтверждения, выбрав ► Редактировать ► Отмена ► или нажав кнопку Отмена .
	Откройте диалоговое окно "Редактировать объединение" для выбранного объединения.

Отображение строки формул.

- Выберите ► [Вид](#) ► [Панель формул](#) ►. [Панель формул](#) отобразится ниже окна [юниверса](#).

Изменение объединения с помощью [панели формул](#).

1. Выберите объединение для редактирования. Формула для объединения появится в [панели формул](#).
2. Нажмите выражение объединения в [панели формул](#) в том месте, где необходимо изменить синтаксис.
3. Измените выражение при необходимости.
4. Нажмите [Подтвердить](#), чтобы применить изменения.
5. Нажмите клавишу Возврат для выхода из [панели формул](#) .
Или
Нажмите в любом месте за пределами [панели формул](#).

5.7.10 Поддержка ANSI 92 для объединений в юниверсе.

Средство создания юниверсов поддерживает синтаксис ANSI 92 для объединений. По умолчанию ANSI 92 не поддерживается. Необходимо активировать поддержку путем установки значения YES для параметра ANSI92 юниверса SQL. Этот параметр находится в списке на странице параметров диалогового окна параметров юниверса ("Файл" > "Параметры" > "Параметр"). После активации можно использовать синтаксис ANSI 92 для объединений в юниверсе.

📘 Примечание

Установка ANSI 92 также объявлена в файлах .prg. Если в PRM-файле установлен параметр "usual", приоритет имеют параметры средства создания юниверсов. Если в PRM-файле установлен параметр "ANSI92", параметры средства создания юниверсов переопределяются. Дополнительные сведения о файлах .prg и установке ANSI 92 см. в руководстве [Доступ к данным](#). Характеристика может зависеть от версии базы данных. Дополнительную информацию см. в технических сведениях баз данных.

Перед использованием синтаксиса в объединениях проверьте, чтобы целевая RDBMS поддерживала ANSI 92.

Активация поддержки ANSI 92 в юниверсе и определение объединения с использованием синтаксиса ANSI 92 описано ниже.

5.7.10.1 Сравнение синтаксиса объединения по умолчанию и синтаксиса ANSI 92

Ниже представлен пример синтаксиса для двух объединений. В первом случае рассматривается поведение по умолчанию, при котором объединение определяется в утверждении WHERE. Во втором случае это же объединение определяется в утверждении FROM с помощью стандарта ANSI 92.

Синтаксис объединения по умолчанию

```
SELECT
  Resort.resort,
  'FY'+Format(Sales.invoice_date,'YYYY'),
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
  Resort,
  Sales,
  Invoice_Line,
  Service,
  Service_Line
WHERE
  ( Sales.inv_id=Invoice_Line.inv_id )
  AND ( Invoice_Line.service_id=Service.service_id )
  AND ( Resort.resort_id=Service_Line.resort_id )
  AND ( Service.sl_id=Service_Line.sl_id )
GROUP BY
  Resort.resort,
  'FY'+Format(Sales.invoice_date,'YYYY')
```

Это же объединение с помощью стандарта ANSI 92

```
SELECT
  Resort.resort,
  'FY'+Format(Sales.invoice_date,'YYYY'),
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
  Resort INNER JOIN Service_Line ON (Resort.resort_id=Service_Line.resort_id)
  INNER JOIN Service ON (Service.sl_id=Service_Line.sl_id)
  INNER JOIN Invoice_Line ON (Invoice_Line.service_id=Service.service_id)
  INNER JOIN Sales ON (Sales.inv_id=Invoice_Line.inv_id)
GROUP BY
  Resort.resort,
  'FY'+Format(Sales.invoice_date,'YYYY')
```

5.7.10.2 Активация поддержки ANSI 92 в юниверсе

Чтобы активировать поддержку ANSI 92 для объединений, выполните следующие действия.

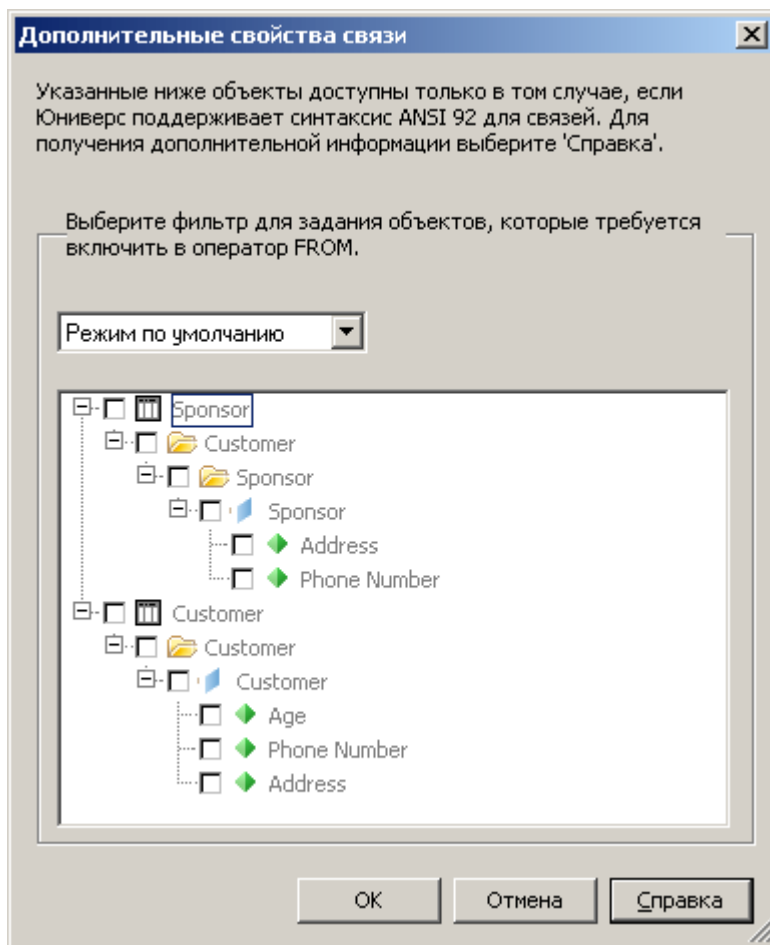
1. Выберите "Файл" > "Параметры".
Откроется диалоговое окно "Параметры юниверса".
2. Выберите вкладку "Параметры".
Откроется страница "Параметры". В ней содержатся определенные параметры генерации SQL, которые можно устанавливать на уровне юниверса, чтобы оптимизировать генерацию SQL для текущего юниверса. В предыдущих версиях продуктов Business Objects эти параметры были включены в PRM-файл для конечного объекта RDBMS. Некоторые параметры RDBMS все еще хранятся в PRM-файлах, но большинство стандартных параметров SQL отображаются на странице "Параметры". Полный список доступных параметров см. в главе [Установка параметров генерации SQL \[страница 96\]](#).
3. Выберите в списке параметр ANSI92.
4. В окне значений введите "Да".
5. Нажмите "Заменить".
6. Нажмите кнопку ОК.
Теперь стандарт ANSI 92 можно применять для объединения определений в юниверсе. Если в диалоговом окне "Редактирование объединения" нажать кнопку "Дополнительно", отобразится окно "Дополнительное объединение". Можно задать фильтр, чтобы определить, какие измерения необходимо включить в утверждение FROM для объединения.

5.7.10.3 Определение объединения с помощью синтаксиса ANSI 92

Синтаксис ANSI 92 можно использовать для определения объединения в свойствах диалогового окна "Редактировать объединение". Это можно сделать, используя окно дополнительного редактирования, позволяющее выбирать объекты, которые необходимо включить в определение объединения.

Чтобы определить объединение с помощью синтаксиса ANSI 92, выполните следующие действия.

1. Активируйте поддержку ANSI 92 для юниверса. Дополнительные сведения см. в разделе [Активация поддержки ANSI 92 в юниверсе \[страница 160\]](#).
2. Двойным щелчком мыши выберите объединение в схеме.
Отобразится окно "Редактировать объединение".
3. Нажмите кнопку "Дополнительно".
Откроется диалоговое окно "Дополнительные свойства объединения".



4. В раскрывающемся списке выберите один из следующих фильтров утверждения FROM.

Параметр FROM	Описание
Режим по умолчанию	Для объединений применяется синтаксис по умолчанию. Объединения заданы в утверждении WHERE.
Все объекты в операторе FROM	Все объекты, определенные столбцами таблицы с правой и левой стороны объединения, включаются в утверждение FROM.
Ни одного объекта в операторе FROM	В утверждение FROM не включается ни один объект.
Выбранные объекты в операторе FROM	В утверждение FROM включаются только объекты, выбранные в древовидном представлении "Дополнительные свойства объединения" таблиц объединений.

5. Выберите объекты, которые необходимо включить в утверждение FROM, если выбран фильтр "Выбранные объекты в FROM".
6. Нажмите кнопку OK.
7. Введите любые другие параметры объединения в окне "Редактировать объединение".
8. Нажмите OK.

5.7.11 Удаление объединений

Чтобы удалить объединение,

1. Щелчком мыши выберите объединение.
Объединение выбрано
2. Выполните одно из следующих действий.
 - Нажмите клавишу Backspace на клавиатуре
 - Нажмите клавишу Delete на клавиатуре
 - Щелкните правой кнопкой мыши объединение и выберите "Удалить" в контекстном меню.
Откроется окно с запросом подтверждения удаления объединения.
3. Нажмите "Да"
Объединение удалено.

❗ Примечание

Убедитесь в осведомленности всех последствий в схеме и юниверсе при удалении объединения. Проверьте, что удаление объединения не влияет на контекст. При попытке удалить объединение средство создания юниверсов предупреждает, если объединение используется в одном или нескольких контекстах. Необходимо вручную проверить эти контексты и оценить степень воздействия на юниверс со стороны возможного изменения контекста в результате удаления объединения.

5.8 Определение отдельных типов объединений

В средстве создания юниверсов можно определить следующие объединения:

Тип объединения	Описание
Равные объединения (включая сложные равные объединения)	Соединяют таблицы, основанные на равенстве значений в столбце одной таблицы и значений в столбце другой. Так как один и тот же столбец представлен в обеих таблицах, это объединение синхронизирует две таблицы. Также можно создать сложные равные объединения, где одно объединение связывает несколько столбцов между двумя таблицами.
Тета-объединения (условные объединения)	Связывают таблицы, основанные скорее на взаимосвязи, чем на равенстве между двумя столбцами.
Внешние объединения	Связывают таблицы, в одной из которых присутствуют строки, несоответствующие строкам в общем столбце другой таблицы.

Тип объединения	Описание
Объединение быстрого доступа	Объединение, предоставляющее альтернативный путь между двумя таблицами, пропускающее промежуточные таблицы, приводящее к такому же результату, независимое от направления. Оптимизирует время запроса, сокращая путь до минимума.
Самоограниченные объединения	Объединение одиночной таблицы используется для установки защиты для таблицы.

Каждый тип объединения полностью описан в соответствующем разделе каждой главы. Используйте одинаковый способ для создания каждого типа объединения. Однако во время создания объединения необходимо определять различные свойства для каждого объединения в поле "Редактировать объединение".

5.8.1 Создание равных объединений

Равное объединение связывает две таблицы по общим значениям в столбце таблицы 1 со столбцом таблицы 2. Задаваемое ограничение соответствует следующим синтаксическим правилам:

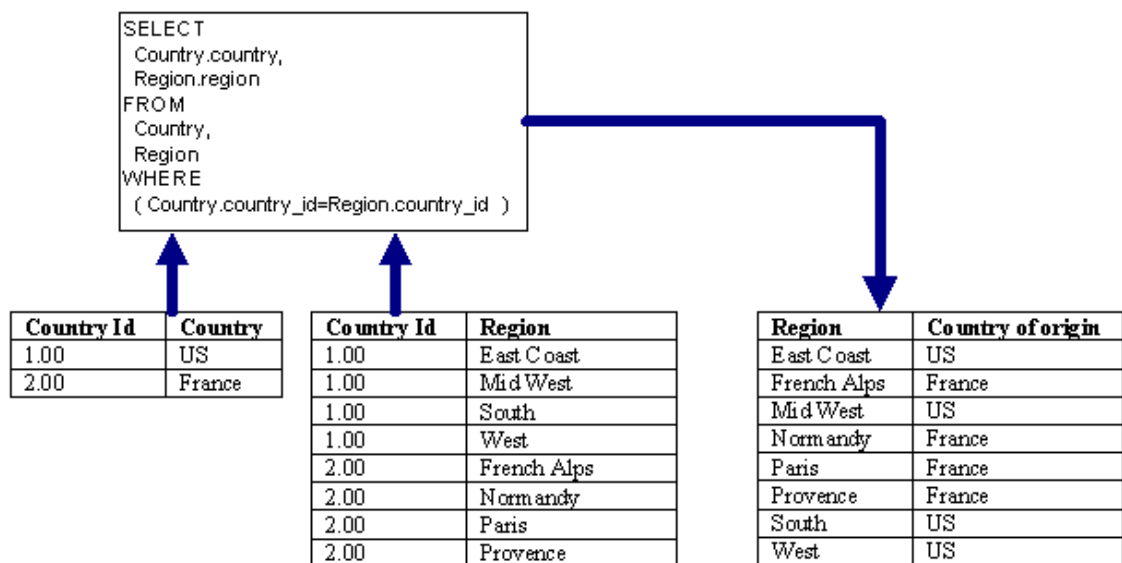
Таблица1.столбец_a = Таблица2.столбец_a

В стандартной базе данных, используемые столбцы в равном объединении, обычно являются первичным ключом из одной таблицы и внешним ключом из другой. Для получения дополнительных сведений о ключах, см. в разделе [Что отсутствует в таблицах для объединения? \[страница 146\]](#).

Это равное объединение создаваемое по умолчанию при создании нового объединения. Большинство объединений в схеме должны быть равными.

Пример: Равное объединение ограничивает данные

Если в примере ниже запускается инструкция Select, то утверждения Select и From создают декартов продукт. Однако, прежде чем будут возвращены какие-либо данные, утверждение Where применяет ограничение, чтобы возвращались только строки, у которых совпадают данные в столбце идентификатора страны в обеих таблицах.



5.8.1.1 Создание нового равного объединения

Чтобы создать новое равное объединение выполните следующие действия:

- Создайте объединение двух таблиц.
Новое объединение по умолчанию является равным объединением.

→ Совет

Можно использовать различные способы для создания объединений. Дополнительные сведения см. в разделе [Отображение ключей \[страница 146\]](#).

5.8.1.2 Создание равного объединения в существующем объединении

Чтобы создать равное объединение в существующем объединении выполните следующие действия:

1. Дважды щелкните мышью существующее объединение.
Откроется окно "Редактировать объединение".
2. Выберите столбец в поле списка Таблицы1.
3. Выберите соответствующие столбцы в поле списка Таблицы2
4. Выберите = в поле выпадающего списка "Оператор".
Поле "Редактирование объединений" отображает равное объединение между таблицами клиента и блокировок.

❗ Примечание

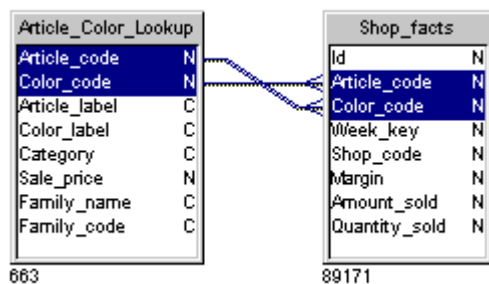
Общие столбцы не всегда с одинаковыми именами. Необходимо проверить имена столбцов первичного и внешнего ключа в базе данных. Различные таблицы можно использовать одинаковыми столбцами ключа, переименование каждого столбца зависит от роли таблицы в базе данных.

5. Нажмите кнопку "Разбор" для проверки синтаксиса объединения.
При получении сообщения об ошибке проверьте, что данный столбец является общим для обеих таблиц.
6. Нажмите кнопку "OK".

5.8.1.3 Создание сложных равных объединений

Также существует возможность создания сложных равных объединений. Это простое объединение, которое связывает несколько столбцов между таблицами. Можно создавать сложные равные объединения, используя сложный оператор для объединения на листе "Редактирование свойств".

Юниверс выборки eFashion содержит сложное объединение, как показано ниже.



Использование сложного равного объединения вместо нескольких простых равных объединений между объединенными столбцами имеет следующие преимущества.

- Необходимо определять только одну мощность множества. Это может сэкономить время при определении мощностей множеств, а также предохраняет схему базы данных от ошибок и облегчает ее чтение.
- Выражения SQL для всех объединений между двумя таблицами можно увидеть в текстовом поле "Выражение" в блоке "Изменить свойства" для объединения. При использовании нескольких простых равных объединений между двумя таблицами для каждого объединения будет использоваться единственное выражение.

Чтобы создать равное сложное объединение выполните следующие действия:

1. Дважды щелкните мышью существующее объединение. Откроется окно "Редактировать объединение".
2. Выберите несколько столбцов в окне списка "Таблица1".
3. Выберите соответствующие столбцы в окне списка "Таблица2".
4. Выберите элемент "Сложный" из выпадающего списка "Оператор". Поле "Редактирование объединения" отображает сложное объединение между таблицами Article_Color_Lookup и Shop_facts.

5. Нажмите кнопку "Разбор" для проверки синтаксиса объединения.
При получении сообщения об ошибке проверьте, что данный столбец является общим для обеих таблиц.
6. Нажмите кнопку "OK".

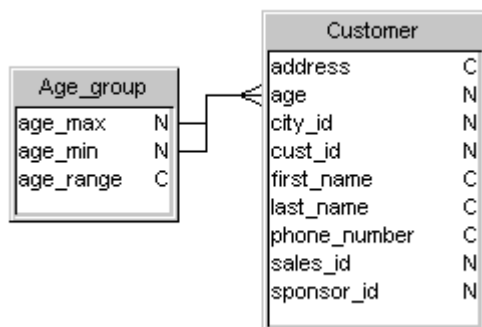
5.8.2 Тета-объединения

Тета-объединение – объединение, которое соединяет таблицы, основанные на отношении, а не на равенстве между двумя столбцами. Тета-объединение может использовать любой оператор, кроме оператора равенства.

Следующий пример покажет как создать тета-объединение которое используется оператором "Между".

Пример: Тета-объединение

Таблица ниже Age_Group содержит сведения о возрасте диапазона которые могут быть использованы для анализа данных о возрасте клиентов.

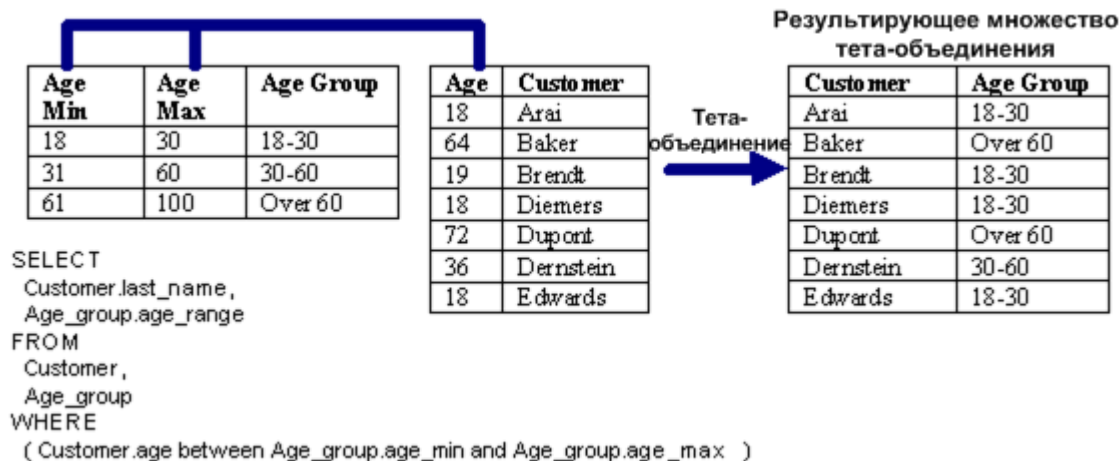


Необходимо включить данную таблицу в юниверс, но нет общих столбцов между таблицей клиента и таблицей Age_Group, поэтому не удалось использовать равные объединения.

Созданное тета-объединения использует оператора "Между" для максимального и минимального возраста диапазона. Используя тета-объединение, вычислите какие объединения существуют, где находится значение в строке столбца возраста в таблице клиента между значениями в строке для столбцов Age_Min и Age_Max и таблицы Age_Group. Объединение определяется с помощью следующих выражений:

Возраст заказчика между Age_group.age_min и Age_group.age_max

Диаграмма ниже отображает объединения между максимальным возрастом, минимальным возрастом и просто возрастом, и результирующее множество, которое возвращается когда используется тета-объединение в выполнении запроса таблиц Age_Group и Клиента.



5.8.2.1 Создание тета-объединений

Для создания тета-объединений используйте столбцы диапазона:

1. Создайте объединение двух таблиц.
Равное объединение создано по умолчанию.
2. Дважды щелкните объединение.
Откроется диалоговое окно "Редактировать объединение".
3. Нажмите столбец в поле списка столбца Таблицы 1
4. Нажмите и удерживайте клавишу CTRL и нажмите два столбца из поля списка столбца Таблицы 2.
В примерах ниже показаны выбранные два столбца age_min и age_max. Оператор "Между" автоматически появится в раскрывающемся списке оператора.

Редактировать связь

Таблица1: Customer

Таблица2: Age_group

Столбцы Таблицы 1: address, age, city_id, cust_id, first_name, last_name, phone number

Столбцы Таблицы 2: age_max, age_min, age_range

Оператор: Between

Кардинальность: 1,1 (Таблица 1) / 1,n (Таблица 2)

☒ Отношение

1 ☐ Обнаружить ☒ 1

N ☒ N

Каждый Customer имеет один и только один Age_group

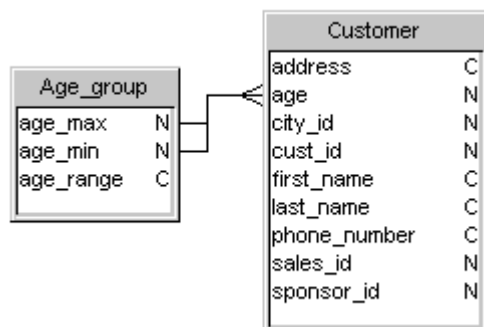
Каждый Age_group имеет один или более Customer

☐ Shortcut join

Выражение: Customer.age between Age_group.age_min and Age_group.age_max

Изменить... Тест

5. Нажмите кнопку "Разбор", чтобы осуществить проверку объединения.
При получении сообщения об ошибке, проверьте правильность выбора столбцов.
6. Нажмите кнопку OK.
Объединение создано в панели структуры.



5.8.3 Внешние объединения

Внешнее объединение – это объединение, связывающее две таблицы, в одной из которых присутствуют строки, не соответствующие строкам в общем столбце другой таблицы.

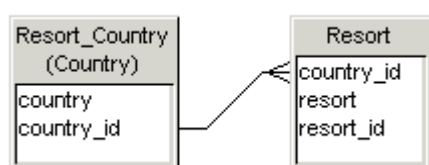
Необходимо определить внешнее объединение, указав, какая таблица является внешней в исходном равном объединении. Внешняя таблица содержит столбец, необходимый для возврата всех значений, даже несоответствующих. Необходимо указать внешнюю таблицу для выбранного объединения в диалоговом окне "Редактировать объединение".

5.8.3.1 Полное внешнее объединение

По умолчанию можно создать либо левое либо правое внешнее объединение, зависящие от построенной поверхности объединения внешней таблицы. Можно также создать полное внешнее объединение путем активации поддержки ANSI 92 для объединений в юниверсе. Достигнуто с помощью установки юниверса SQL параметра ANSI 92 для YES (Файл > Параметры > Параметр). Это позволяет юниверсу поддерживать синтаксис ANSI 92 для объединений, поэтому можно выбрать таблицы другой части объединений внешней таблицы. Дополнительные сведения по созданию полного внешнего объединения [Определение полного внешнего объединения \[страница 174\]](#) см. в разделе.

Пример: Внешнее объединение

Таблицы Resort_Country и Курорт связываются с помощью равного объединения.



Курорт относится к стране, но не каждая страна является курортной. При использовании равных объединений, набор результатов запроса покажет сведения о курортных странах; Австралия, Франция, и США.

Country	Resort
Australia	Australian Reef
France	French Riviera
US	Bahamas Beach
US	Hawaiian Club
US	Royal Caribbean

Однако, можно отобразить все страны независимо от равенства значений во внешнем ключе таблицы курорта. Для этого определите внешнее объединение так, что все страны вернутся, не смотря на соответствующие столбцы курорта.

Синтаксис (Microsoft Access) для внешнего объединения следующий:

```
SELECT
Resort_Country.country,
Resort.resort
FROM
Country Resort_Country,
Resort,
{ oj Resort_Country LEFT OUTER JOIN Resort ON
Resort_Country.country_id=Resort.country_id }
```

📌 Примечание

В примере выше используется Microsoft Access, так что любое одно ко множеству объединений следующих таблиц курорта, следует так же использовать внешнее объединение. Если нет, то значение NULL, которое было возвращено исходным равным объединением, не учитывается, если последующие объединения не вернули совпадающее значение NULL. Обработка внешнего объединения выполняется каждой РСУБД по-своему, для получения дополнительных сведений см. документацию РСУБД. Дополнительные сведения об ограничениях использования внешнего объединения [Ограничения для использования внешних объединений \[страница 174\]](#) см. в разделе.

5.8.3.2 Создание внешнего объединения

Чтобы создать внешнее объединение,

1. Дважды щелкните мышью существующее равное объединение.
Откроется диалоговое окно "Редактировать объединение".
2. Установите флажок "Внешнее объединение" для таблицы, возвращающей все значения в запросе.
В примере ниже верните все значения для Resort_Country.

Редактировать связь

Таблица1

Resort_Country

country

country_id

Таблица2

Resort

country_id

resort

resort_id

0,n 1,1

=

☒ Outer join ☐ Outer join

☒ Отношение

1 ☒ N ☐ ☐ 1 ☒ N

Каждый Resort_Country имеет ноль или более Resort
Каждый Resort имеет один и только один Resort_Country

☐ Shortcut join

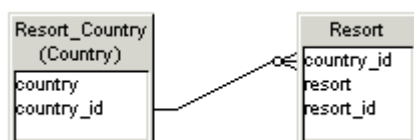
Выражение

Resort_Country.country_id=Resort.country_id

- Нажмите кнопку "Анализ" для проверки синтаксиса объединения.
При получении сообщения об ошибке, проверьте правильность выбора столбцов.
- Нажмите кнопку "ОК".
В средстве создания юниверсов объединение отображается в панели "Структура". Если внешнее объединение создано с использованием параметра графического отображения *Вилка*, оно обозначается маленьким кружком на противоположной стороне объединения в таблице, которая возвращает несоответствующие значения.

Примечание

Если внешнее объединение создается с параметром графического отображения, отличным от параметра *Вилка*, маленький кружок не появляется.



5.8.3.3 Определение полного внешнего объединения

Можно определить внешнее объединение, используя стандарт ANSI 92 для определения внешних объединений. Это позволяет указывать полное внешнее объединение. Чтобы использовать стандарт ANSI 92 для внешних объединений, необходимо установить параметр ANSI 92 на "ДА". Данный параметр доступен на странице "Параметр" ("Файл" > "Параметры" > "Параметр").

📌 Примечание

Сведения об установке данного параметра и других параметров генерации SQL для юниверса см. в разделе [Установка параметров генерации SQL \[страница 96\]](#).

Если параметр ANSI 92 установлен на "ДА", можно выбирать таблицы с двух сторон объединения и назначать их качестве внешних таблиц. До установки параметра необходимо убедиться, что целевая RDBMS поддерживает синтаксис ANSI 92 для внешних объединений.

Определите полное внешнее объединение в две фазы.

- Активируйте поддержку ANSI 92 для внешних объединений юниверса. Дополнительные сведения см. в разделе [Активация поддержки ANSI 92 в юниверсе \[страница 160\]](#).
- Используйте диалоговое окно "Редактировать объединение", чтобы определить полное внешнее объединение.

Чтобы определить полное внешнее объединение,

1. Активируйте поддержку ANSI 92 для юниверса.
2. Двойным щелчком мыши выберите объединение в схеме.
Откроется диалоговое окно "Редактировать объединение".
3. Установите флажок "Внешнее объединение" для обеих таблиц, включенных в объединение.
4. Нажмите кнопку "ОК".
В средстве создания юниверсов объединение отображается в панели "Структура". Полное внешнее объединение обозначается двумя кружками на ссылке связи между двумя таблицами.

5.8.3.4 Ограничения для использования внешних объединений

Использование внешних объединений очень полезно. Но прежде необходимо ознакомиться с проблемами производительности и реализации.

- Производительность может снижаться. Возвращается большее количество строк, а некоторые базы данных не используют индексы, если содержатся внешние объединения. Поэтому большое количество данных может замедлять производительность запроса.
- Ограничения базы данных на использование внешних объединений. Не все базы данных позволяют управлять внешними объединениями в утверждении WHERE. Это необходимо при использовании самоограниченных объединений. Например, самоограниченное объединение TYPE_CODE=10 может вернуть все строки, где ТИП=10, или тип имеет неопределенное значение, так как "ТИП=10" никогда не будет иметь значение "истина", если код типа имеет неопределенное значение, в то время как неопределенные значения генерируются с помощью внешнего объединения.

- Необходимо проверить, как целевая RDBMS обрабатывает внешние объединения, чтобы избежать неполные пути к запросу после исходного внешнего объединения. Например, в образце Club.mdb базы данных Microsoft Access все объединения типа "один ко многим", которые следуют за внешним объединением в пути объединения, должны быть определены в качестве внешних объединений. Если это не так, исходное внешнее объединение будет игнорироваться исходным запросом. В приведенном выше примере объединение между Resort и Service_Line игнорирует пустые значения, возвращенные с помощью внешнего объединения между Resort_Country и Resort. При выполнении запроса с помощью трех таблиц возвращается ошибка базы данных. При этом пользователю предлагается создать отдельный запрос, который выполняет первое объединение, а затем включить этот запрос в SQL-инструкцию. Данный тип ошибки может привести в замешательство многих пользователей. Поэтому в таких случаях предпочтительно не использовать внешние объединения и не завершать путь к ним.



5.8.4 Объединения быстрого доступа

Объединение быстрого доступа – это объединение, которое предоставляет альтернативный путь между двумя таблицами. Объединения ярлыков увеличивают скорость запроса, так как не учитывают промежуточные таблицы, таким образом укорачивая путь объединения.

Распространенное использование объединений быстрого доступа – объединение совместно используемой таблицы соответствий с таблицей, расположенной дальше на пути объединения. Путь объединения включает в себя несколько различных таблиц в одном контексте.

В таком случае объединение быстрого доступа эффективно только тогда, когда найденное значение было смещено на более низкий уровень в иерархии таблиц. Таким образом, одно и то же значение существует на всех объединенных уровнях.

Прямое объединение будет игнорироваться при отсутствии любого прямого пути объединения для данного контекста. SQL, созданный для связанного запроса Web Intelligence, не будет учитывать неэффективное прямое объединение.

📌 Примечание

Средство создания юниверсов не рассматривает ярлыки объединения во время автоматического определения контекста и цикла. Однако при назначении мощности множества для прямого объединения можно избежать появления сообщения "Не все мощности множества назначены" во время определения контекстов.

5.8.4.1 Создание объединения быстрого доступа

Чтобы создать объединение быстрого доступа:

1. Установите две таблицы в один путь объединения, который может осуществлять прямое объединение.
2. Создайте объединение двух таблиц.
3. Дважды щелкните мышью новое объединение.
Откроется диалоговое окно "Редактировать объединение".
4. Выберите поле проверки объединения быстрого доступа.
5. Выберите или введите другие требуемые свойства объединения.
6. Нажмите "ОК".
Объединение быстрого доступа отображается, объединяя две таблицы. Объединение быстрого доступа отображается в виде пунктирной линии в панели структуры.

❗ Примечание

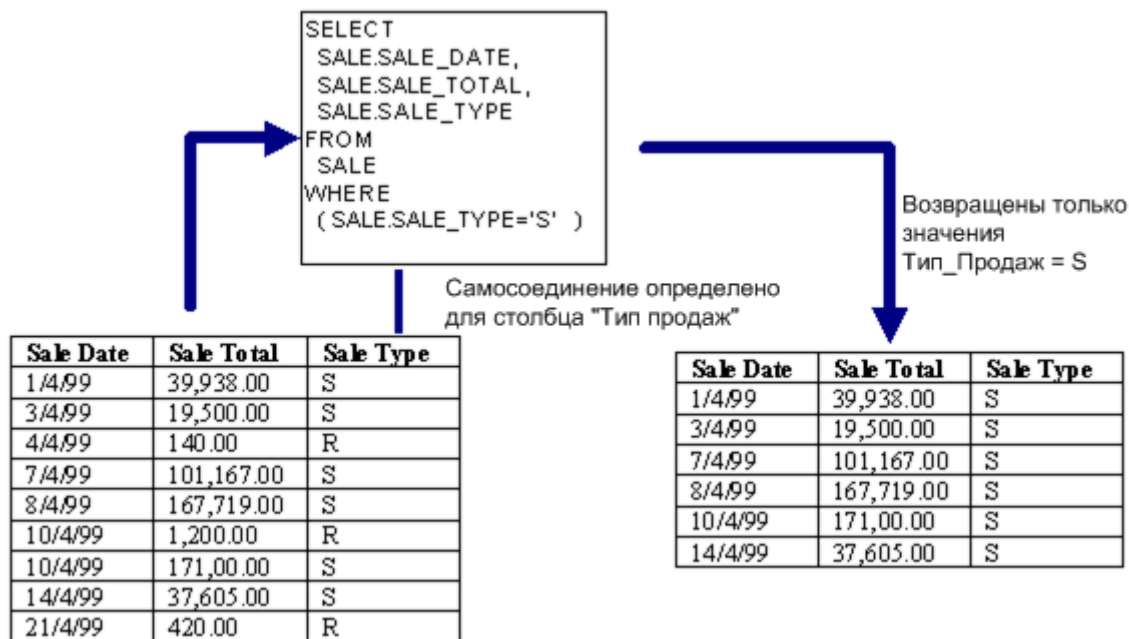
Следует настроить то же значение числа элементов объединения быстрого доступа, что и для пути объединения, который оно заменяет.

5.8.5 Самоограниченные объединения

Самоограниченное объединение на самом деле не является объединением, а самоограничением в конкретной таблице. Можно использовать самоограниченное объединение, чтобы ограничить результаты, возвращенные значениями таблицы, используя фиксированные значения.

Пример: Самоограниченное объединение

В таблице "Продажи", расположенной ниже, содержатся строки данных о проданных и арендуемых автомобилях. Столбец `Sale_Type` используется в качестве метки, чтобы определить тип транзакции (`S` = автомобиль для продажи, `R` = автомобиль для аренды). Самоограниченное объединение ограничивает данные, возвращенные из таблицы "Продажи" в `Sale_Type = S`. Это гарантирует, что любой объект на основе таблицы "Продажи" или объединений, проходящих через данную таблицу, получит результаты запроса только по проданным автомобилям.



Без самоограниченного объединения набор результатов запроса получит строки, где столбец Sale_Type будет равным S или R.

→ Совет

Настройка мощности множества для самоограниченного объединения помогает предотвратить появление сообщения "Не все мощности множеств указаны" при выявлении контекстов. Следует настроить мощность множества в точном соответствии один другому, несмотря на то что фактические настройки являются неважными, пока они установлены.

5.8.5.1 Создание самоограниченного объединения

Чтобы создать самоограниченное объединение,

1. Выберите пункт меню "Вставка" > "Объединение".
Откроется диалоговое окно "Редактировать объединение".
2. Выберите таблицу, для которой необходимо настроить самоограниченное объединение, в списке таблиц в раскрывающемся поле.
Столбцы для выбранной таблицы отображаются в списке столбцов таблицы.
3. Нажмите столбец, который необходимо использовать для определения ограничения, в списке столбцов в раскрывающемся поле.
4. Выберите ту же таблицу, которую выбрали в списке таблиц в раскрывающемся поле.
5. Нажмите тот же столбец, который выбрали в списке столбцов таблицы.
Выражение объединения отобразится в текстовом окне "Выражения".

Редактировать связь

Таблица1

Article_lookup

Article_id
Article_label
Category
Sale_price
Family_name
Family_code

=

Таблица2

Article_lookup

Article_id
Article_label
Category
Sale_price
Family_name
Family_code

Outer join ☐ Outer join ☐

☒ Отношение

1 ☐
N ☐

Обнаружить

1 ☐
N ☐

☐ Shortcut join

Выражение

Article_lookup.Family_code=Article_lookup.Family_code

Изменить... Тест

Дополнительно OK Отмена Справка

6. Замените значение операнда в выражении объединения с ограничением значений, которое необходимо установить в столбце объединения.
Например, если необходимо ограничить возвращенное значение из столбца Family_code в F3, установите Article_lookup.Family_code после знака "=" с F3, как показано ниже.

Редактировать связь

Таблица1

Article_lookup

Article_id
Article_label
Category
Sale_price
Family_name
Family_code

=

Таблица2

Article_lookup

Article_id
Article_label
Category
Sale_price
Family_name
Family_code

Outer join ☐ Outer join ☐

☒ **Отношение**

1 ☐ N ☐ **Обнаружить** 1 ☐ N ☐

☐ **Shortcut join**

Выражение

Article_lookup.Family_code='F3'

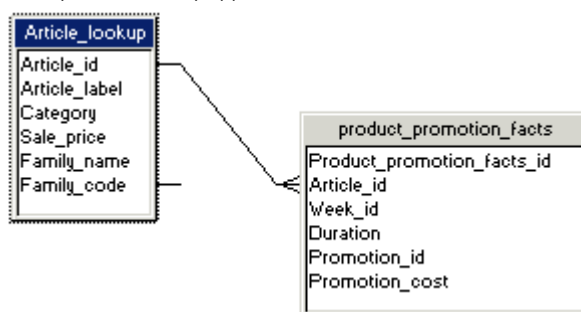
Изменить... Тест

Дополнительно OK Отмена Справка

7. Нажмите кнопку "Анализ" для проверки синтаксиса.

8. Нажмите кнопку OK.

Самоограниченное объединение отобразится в виде короткой строки напротив столбца, для которого оно определено.



5.9 Использование мощности множеств

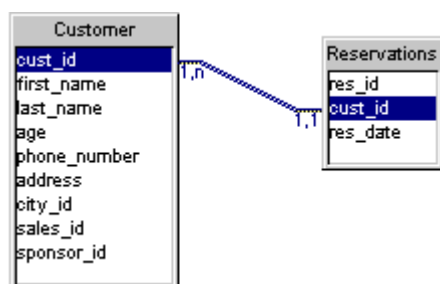
Мощность множества – это свойство объединения, которое описывает, сколько строк в одной таблице совпадает со строками в другой таблице.

Мощность множества выражена минимальным и максимальным числом строк в столбце в одном конце объединения, которые имеют совпадающие строки в столбце на другом конце объединения.

Минимальное и максимальное количество совпадающих строк может быть равно 0, 1 или объединение N. А представляет собой реверсивную связь, таким образом необходимо иметь две мощности множеств, по одной на каждый конец объединения.

Пример: Мощность множеств объединения

Две таблицы "Клиент" и "Предварительные заказы" связаны с помощью объединения.



Мощность множества в объединении выше может быть выражена следующим образом.

Описание	Представление
Каждый клиент может сделать один или несколько предварительных заказов.	(1,N)
Для каждого предварительного заказа может быть только один клиент	(1,1)

5.9.1 Использование свойств связей в средстве создания юниверсов

Свойства связи для объединения не играют роли в SQL, сгенерированном при выполнении запроса. Однако средство создания юниверсов использует свойства связей, чтобы определять контексты и допустимые пути запросов.

Контекст – это набор объединений, которые обеспечивают путь допустимого запроса. Контексты используются для решения проблем объединения, которые могут возвращать слишком много или

слишком мало строк, из-за способа объединения таблиц в целевой базе данных. Контексты описаны в разделе [Определение и устранение проблем объединений \[страница 199\]](#).

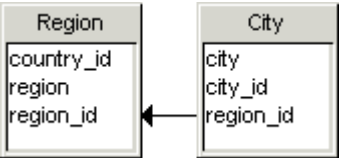
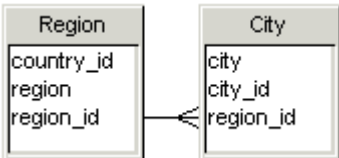
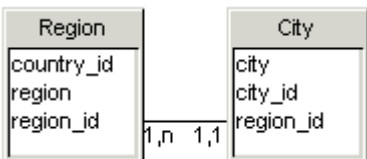
Контексты влияют на SQL, сгенерированный для запроса, так как они направляют конечного пользователя выбирать определенный путь объединения или решать проблему пути объединения.

Необходимо проверить, чтобы мощности множеств были правильно установлены для всех объединений в схеме, чтобы убедиться, что контексты правильные, а пути объединения допустимы.

Установка мощности множеств помогает понять, как таблицы взаимодействуют в базе данных, и графически определить потенциальные проблемы пути объединения в схеме.

5.9.1.1 Отображение мощностей множеств

Можно отображать мощности множеств в панели структуры, используя следующие символы.

Символ мощности множества	Пример	Описание
Указатель		Указатель отображает направлений объединения. Если мощность множества равна 1,1, то указатель отображается на конце каждого объединения.
Вилка		"Птичья лапка" обозначает "главный" конец объединения. Если мощность множества равна 1,1, то отображается прямая линия.
1,N		Мощность множества отображается в виде отношения на каждом конце объединения.

Чтобы отобразить мощности множеств:

1. Выберите "Инструменты" > "Параметры".
На главной странице откроется диалоговое окно "Параметры".
2. Выберите вкладку "Графическое отображение".
Откроется страница "Графическое отображение".
3. Щелкните указатель, вилку или переключатель 1,n.

4. Нажмите кнопку "OK".

5.9.2 Установка мощностей множеств вручную

Можно указать мощности множеств для объединений вручную, с помощью диалогового окна [Редактировать объединение](#).

Зачем устанавливать мощности множеств вручную?

При настройке мощностей множеств вручную необходимо рассматривать отдельно каждое объединение. Это помогает получить сведения о возможных ошибках пути объединения в схеме. Данные проблемы можно не обнаружить, если выбирать автоматически определенные мощности множеств; например изолированные объединения "один-к-одному" в конце пути объединения или избыточные первичные ключи, где все столбцы необходимы для подтверждения уникальности.

Основные сведения о ключах

Определение мощностей множеств для большинства объединений производится с помощью оценки первичного и внешнего ключей в каждой таблице. Первичные и внешние ключи описываются следующим образом:

Ключ	Описание
Первичный	Один столбец или их комбинация в таблице, значения которой определяют каждую строку в таблице. Первичный ключ гарантирует уникальность строк в таблице. Каждая таблица обладает только одним первичным ключом.
Внешний	Столбец или их комбинация в таблице, значения которой должны совпадать с первичным или другим уникальным ключом в другой таблице. Внешние ключи создают ограничения. Например, невозможность продажи продукта пользователю, если пользователь еще не был создан. Каждая таблица может иметь несколько внешних ключей.

Каковы критерии для настройки мощностей множеств?

Оценка отношений между первичными и внешними ключами необходима для определения мощности множеств объединения следующим образом:

Если объединение связывается...	Мощность множеств может быть...
Полный первичный ключ таблицы 1 с полным первичным ключом таблицы 2. Например:	Один к одному (1,1). Только одной строке из каждой таблицы будет возвращено значение первичного ключа.
Полный первичный ключ таблицы 1 с соответствующим внешним ключом таблицы 2. Например:	Один ко многим (1,N). Значения внешних ключей таблицы не гарантируют уникальности и, таким образом, могут дать много совпадающих значений для одного значения первичного ключа исходной таблицы.
Полный первичный ключ таблицы 1 с частью первичного ключа таблицы 2. Например:	Один ко многим (1,N). Неполное совпадение первичного ключа может установить много совпадающих значений для одного первичного ключа исходной таблицы.

5.9.2.1 Установка мощностей множеств вручную

Можно указать мощности множеств для объединений вручную, с помощью диалогового окна [Редактировать объединение](#).

Зачем устанавливать мощности множеств вручную?

При настройке мощностей множеств вручную необходимо рассматривать отдельно каждое объединение. Это помогает получить сведения о возможных ошибках пути объединения в схеме. Данные проблемы можно не обнаружить, если выбирать автоматически определенные мощности множеств; например изолированные объединения "один-к-одному" в конце пути объединения или избыточные первичные ключи, где все столбцы необходимы для подтверждения уникальности.

Основные сведения о ключах

Определение мощностей множеств для большинства объединений производится с помощью оценки первичного и внешнего ключей в каждой таблице. Первичные и внешние ключи описываются следующим образом:

Ключ	Описание
Первичный	Один столбец или их комбинация в таблице, значения которой определяют каждую строку в таблице. Первичный ключ гарантирует уникальность строк в таблице. Каждая таблица обладает только одним первичным ключом.
Внешний	<p>Столбец или их комбинация в таблице, значения которой должны совпадать с первичным или другим уникальным ключом в другой таблице.</p> <p>Внешние ключи создают ограничения. Например, невозможность продажи продукта пользователю, если пользователь еще не был создан. Каждая таблица может иметь несколько внешних ключей.</p>

Каковы критерии для настройки мощностей множеств?

Оценка отношений между первичными и внешними ключами необходима для определения мощности множеств объединения следующим образом:

Если объединение связывается...	Мощность множеств может быть...
Полный первичный ключ таблицы 1 с полным первичным ключом таблицы 2. Например:	<p>Один к одному (1,1).</p> <p>Только одной строке из каждой таблицы будет возвращено значение первичного ключа.</p>
Полный первичный ключ таблицы 1 с соответствующим внешним ключом таблицы 2. Например:	<p>Один ко многим (1,N).</p> <p>Значения внешних ключей таблицы не гарантируют уникальности и, таким образом, могут дать много совпадающих значений для одного значения первичного ключа исходной таблицы.</p>

Если объединение связывается...

Мощность множеств может быть...

Полный первичный ключ таблицы 1 с частью первичного ключа таблицы 2. Например:

Один ко многим (1,N). Неполное совпадение первичного ключа может установить много совпадающих значений для одного первичного ключа исходной таблицы.

5.9.2.2 Автоматическое определение мощностей множеств

Можно использовать функцию средства создания юниверсов "Определять свойства связей" для автоматического определения свойств связей в следующих случаях:

- Выбранные объединения
- Все объединения
- При создании объединения
- В диалоговом окне "Редактировать объединение"

При использовании автоматического определения мощностей множеств определение происходит автоматически.

❗ Примечание

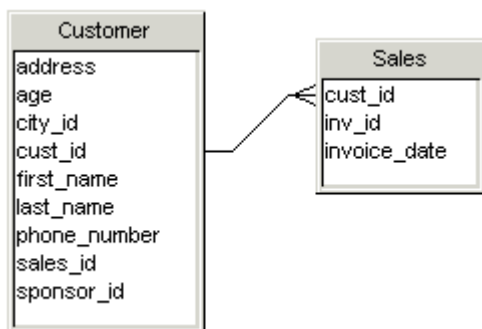
Следует использовать автоматическое определение мощностей множеств должным образом. Может быть очень полезно быстро определять мощности множеств в схеме, однако существуют определенные структурные проблемы, присущие реляционным базам данных, которые могут привести к некорректному определению мощностей множеств. В них входят неполные первичные объединения и предварительно разработанные первичные ключи. Это обсуждается в разделе [Оптимизация автоматического определения мощности множества \[страница 189\]](#).

Автоматическое определение мощностей множеств для выбранных объединений

Чтобы автоматически определить мощности множеств для выбранных объединений,

- Нажмите объединение и выберите "Инструменты" > "Определить мощности множеств"
- Щелкните правой кнопкой мыши объединение и выберите "Определить мощности множеств" в контекстном меню.

Мощность множества отображается "птичьей лапкой" на главном конце объединения.

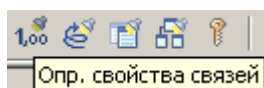


Если выбрать "Инструменты" > "Определить мощности множеств" напрямую, не выбирая объединение, будет получено сообщение, о том, что объединение не было выбрано, и уточняющее, нужно ли определить мощности множеств для всех объединений.

Автоматическое определение мощностей множеств для всех объединений

Чтобы автоматически определить мощности множеств для всех объединений,

1. Нажмите "Инструменты" > "Автоматическое определение" > "Определить мощности множеств". Или



Нажмите кнопку "Определить мощности множеств".

Окно сообщения отображается, спрашивая, нужно ли определить мощности множеств для всех объединений.

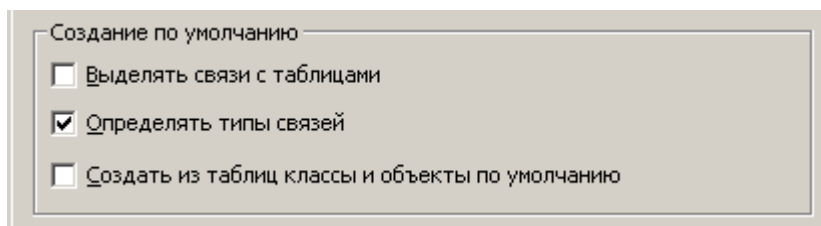
2. Нажмите "Да"

Все объединения на панели структуры отображаются с мощностями множеств.

Автоматическое определение мощностей множеств на основе средства создания объединения

Чтобы автоматически определить мощности множеств на основе средства создания объединения,

1. Выберите Инструменты > Параметры.
На главной странице откроется диалоговое окно "Параметры".
2. Выберите вкладку "База данных".
Откроется страница "База данных".
3. Выберите "Определить мощности множеств" в окне проверки объединений.

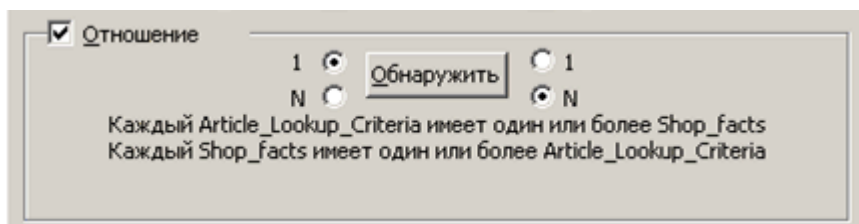


4. Нажмите кнопку ОК.
5. Во время создания нового объединения мощность множества автоматически определяется и отображается в объединении.

Автоматическое определение мощности множества в поле "Редактировать объединение"

Чтобы автоматически определить мощность множества в поле "Редактировать объединение",

1. Дважды щелкните объединение.
Откроется диалоговое окно "Редактировать объединение".
2. Установите флажок "Мощность множества".
3. Нажмите кнопку "Определить".
Переключатель мощности множества будет автоматически выбран для определенной мощности множества. Две мощности множеств выражены в форме предложения.



4. Нажмите кнопку "ОК".

5.9.2.3 Оптимизация автоматического определения мощности множества

Можно улучшить время ответа определения мощности множества с помощью изменения параметра в файле PRM целевой RDBMS. Это дает указание алгоритму определения прочесть две, а не три SQL-инструкции, что улучшает производительность алгоритма.

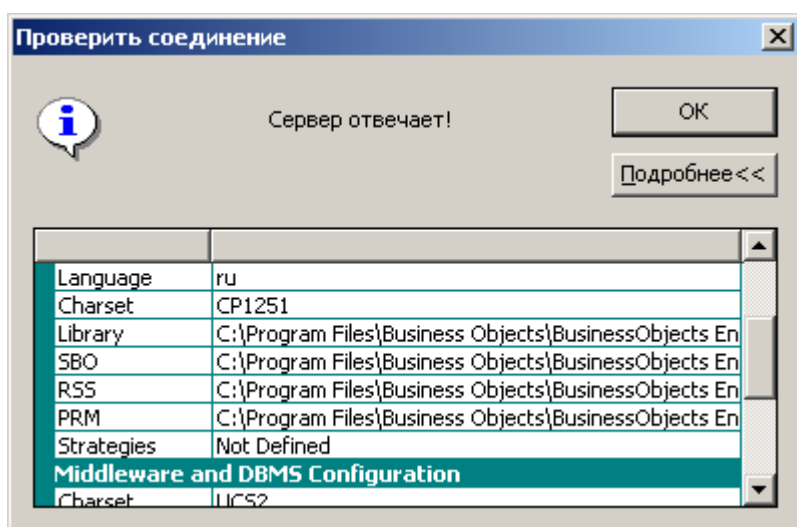
Файл PRM – это текстовый файл, в котором перечислены параметры, используемые для настройки средства создания юниверса и генерации запроса SQL в Web Intelligence. Для каждой поддерживаемой RDBMS существует файл PRM.

Файлы PRM находятся в папках базы данных: <INSTALLDIR>\win32_x86\dataAccess\ConnectionServer\

Проверка, какой файл PRM используется данным соединением

Чтобы проверить, какой файл PRM используется данным соединением юниверса,

1. Выберите "Файл" > "Параметры".
Откроется диалоговое окно параметров.
2. Нажмите кнопку "Проверить".
Откроется окно с сообщением "Проверить соединение".
3. Нажмите кнопку "Подробные сведения".
Подробные сведения о соединении отобразятся в раскрывающемся окне сообщений.
4. Прокрутите вниз окно сообщения к строке, которая начинается с файла PRM.
Данная строка обозначает путь файла и имя файла PRM, которое использует активный юниверс в данный момент.



5. Нажмите кнопку OK.
Вернитесь в диалоговое окно "Параметры".
6. Нажмите "Отмена".

Оптимизация определения мощности множества с помощью файла PRM

Чтобы оптимизировать определение мощности множества с помощью файла PRM,

1. Откройте файл PRM для целевой базы данных в текстовом редакторе.
Файлы PRM хранятся в папке "Доступ к данным" в пути Business Objects.
2. Установите параметр LIGHT_DETECT_CARDINALITY на "ДА".
3. Сохраните и закройте файл PRM.
Открыв юниверс в следующий раз, автоматическое определение мощности множества будет оптимизировано.

5.9.2.4 Оптимизация автоматического определения мощности множества

Можно улучшить время ответа определения мощности множества с помощью изменения параметра в файле PRM целевой RDBMS. Это дает указание алгоритму определения прочесть две, а не три SQL-инструкции, что улучшает производительность алгоритма.

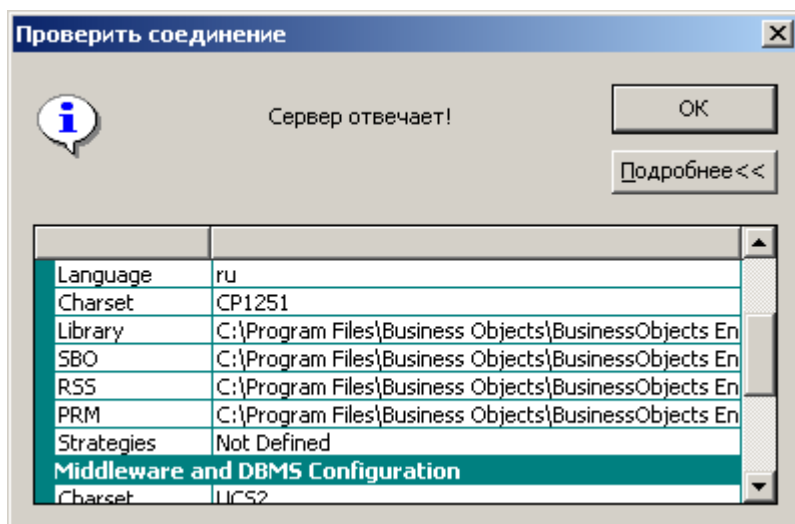
Файл PRM – это текстовый файл, в котором перечислены параметры, используемые для настройки средства создания юниверса и генерации запроса SQL в Web Intelligence. Для каждой поддерживаемой RDBMS существует файл PRM.

Файлы PRM находятся в папках базы данных: <INSTALLDIR>\win32_x86\dataAccess\ConnectionServer\

Проверка, какой файл PRM используется данным соединением

Чтобы проверить, какой файл PRM используется данным соединением юниверса,

1. Выберите "Файл" > "Параметры".
Откроется диалоговое окно параметров.
2. Нажмите кнопку "Проверить".
Откроется окно с сообщением "Проверить соединение".
3. Нажмите кнопку "Подробные сведения".
Подробные сведения о соединении отобразятся в раскрывающемся окне сообщений.
4. Прокрутите вниз окно сообщения к строке, которая начинается с файла PRM.
Данная строка обозначает путь файла и имя файла PRM, которое использует активный юниверс в данный момент.



5. Нажмите кнопку OK.
Вернитесь в диалоговое окно "Параметры".
6. Нажмите "Отмена".

Оптимизация определения мощности множества с помощью файла PRM

Чтобы оптимизировать определение мощности множества с помощью файла PRM,

1. Откройте файл PRM для целевой базы данных в текстовом редакторе.
Файлы PRM хранятся в папке "Доступ к данным" в пути Business Objects.
2. Установите параметр LIGHT_DETECT_CARDINALITY на "ДА".
3. Сохраните и закройте файл PRM.
Открыв юниверс в следующий раз, автоматическое определение мощности множества будет оптимизировано.

5.10 Проверка юниверса

Во время разработки юниверса, необходимо периодически проверять его целостность. Можно проверить целостность юниверса следующим образом.

Проверить юниверс	Описание
Автоматически	Можно установить параметры средства создания юниверсов таким образом, чтобы проверять синтаксис SQL-структур юниверса во время создания, экспорта юниверса или при его открытии.
Вручную	Запустите "Проверить целостность", чтобы проверить целостность выбранных структур юниверса.

5.10.1 Автоматическая Проверка целостности юниверса

В средстве создания юниверсов можно установить следующие параметры проверки целостности юниверса, чтобы анализировать SQL-структуры во время создания, экспорта юниверса или при его открытии:

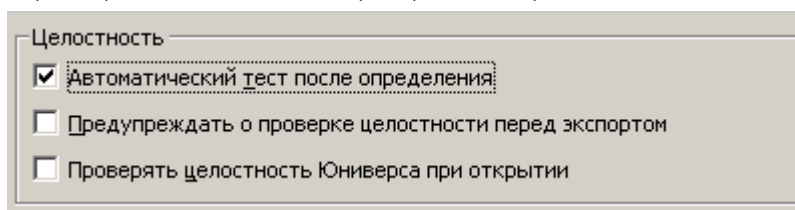
Параметр автоматической проверки	Описание
Автоматическая проверка определения	Средство создания юниверсов автоматически проверяет определения SQL всех объектов, условий и объединений при создании. Это произойдет, если нажать кнопку "ОК" для проверки создания структуры.

Параметр автоматической проверки	Описание
Отправить проверку целостности	Средство создания юниверсов будет отображать предупреждение каждый раз при попытке экспорта непроверенного юниверса.
Проверка целостности юниверса при открытии	При открытии все юниверсы проверяются автоматически.

5.10.1.1 Установка параметров автоматической проверки юниверса

Чтобы установить параметры автоматической проверки юниверса,

1. Выберите Инструменты > Параметры.
На главной странице откроется диалоговое окно "Параметры".
2. В групповом окне "Целостность" установите или удалите флажки для соответствующих параметров автоматической проверки юниверса.



3. Нажмите кнопку "OK".

5.10.1.2 Проверка целостности юниверса вручную

Можно использовать параметр "Проверить целостность", если дизайн активного юниверса точный и отвечает современным требованиям.

Данный параметр определяет:

- Ошибки в объектах, объединениях, условиях и мощностях множеств юниверса.
- Циклы в путях объединения.
- Любые необходимые контексты.
- Изменения в целевой базе данных.

Перед сравнением элементов юниверса и элементов базы данных этот параметр проверяет, допустимо ли соединение с базой данных. Если соединение недопустимо, функция прекращает работу и возвращает ошибку.

5.10.1.3 Типы ошибок, определенные с помощью параметра "Проверить целостность"

Данный параметр может определить:

- Недопустимый синтаксис в определении SQL объекта или объединения.
- Циклы
- Изолированные таблицы
- Изолированные объединения
- Циклы в контекстах
- Отсутствующие и неверные мощности множеств

Как параметр "Проверить целостность" определяет изменения в подключенной базе данных?

С помощью функции "Проверить целостность" в базу данных отправляется запрос о списке таблиц. Затем список сравнивается с таблицами в юниверсе. Можно выполнить ту же операцию для столбцов.

На панели структуры с помощью данного параметра отмечаются таблицы и столбцы, несовпадающие с таблицами и столбцами в списке, так как они недоступны. В базе данных можно определять и переименовывать таблицы и столбцы. См. раздел [Проверка целостности юниверса с помощью параметра "Проверить целостность" \[страница 192\]](#).

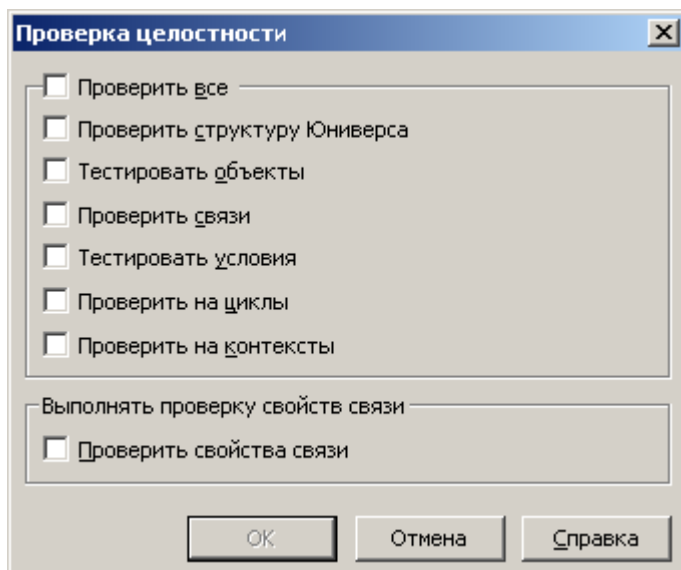
📘 Примечание

Параметр "Проверить мощность множества" медленно обрабатывает большое количество данных. При нахождении неоднозначных или отсутствующих данных, результаты могут быть неточные. Если база данных большая и содержит записи неточных данных, необходимо выбрать параметр "Проверить мощность множества". Если использовать данный параметр, можно оптимизировать определение мощности множества с помощью изменения файла PRM. Дополнительные сведения см. в разделе [Оптимизация автоматического определения мощности множества \[страница 189\]](#).

5.10.1.4 Проверка целостности юниверса с помощью параметра "Проверить целостность"

Чтобы проверить целостность юниверса,

1. Выберите "Инструменты" > "Проверить целостность".
Или
Нажмите кнопку "Проверить целостность".
2. Откроется диалоговое окно "Проверить целостность".

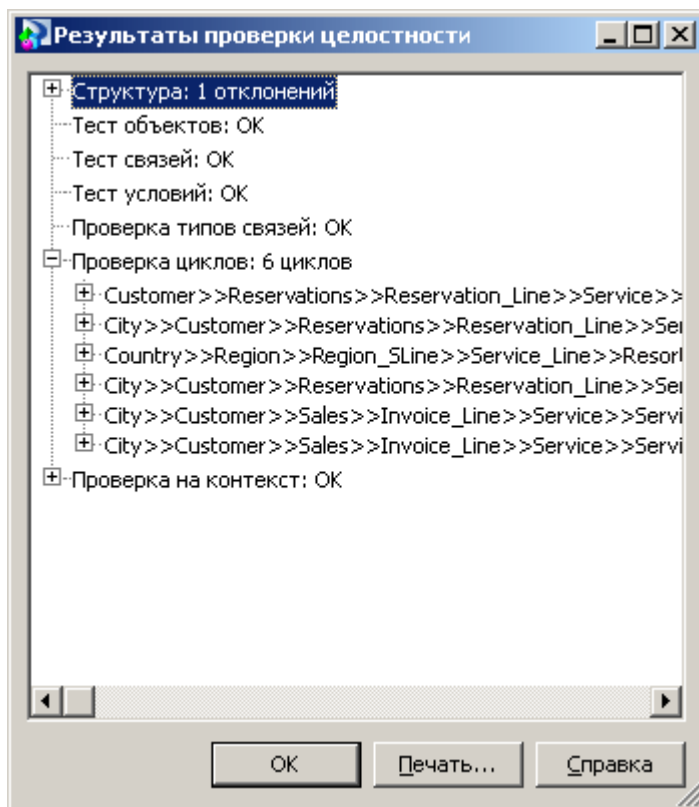


3. Установите флажки на компоненты для проверки.

❗ Примечание

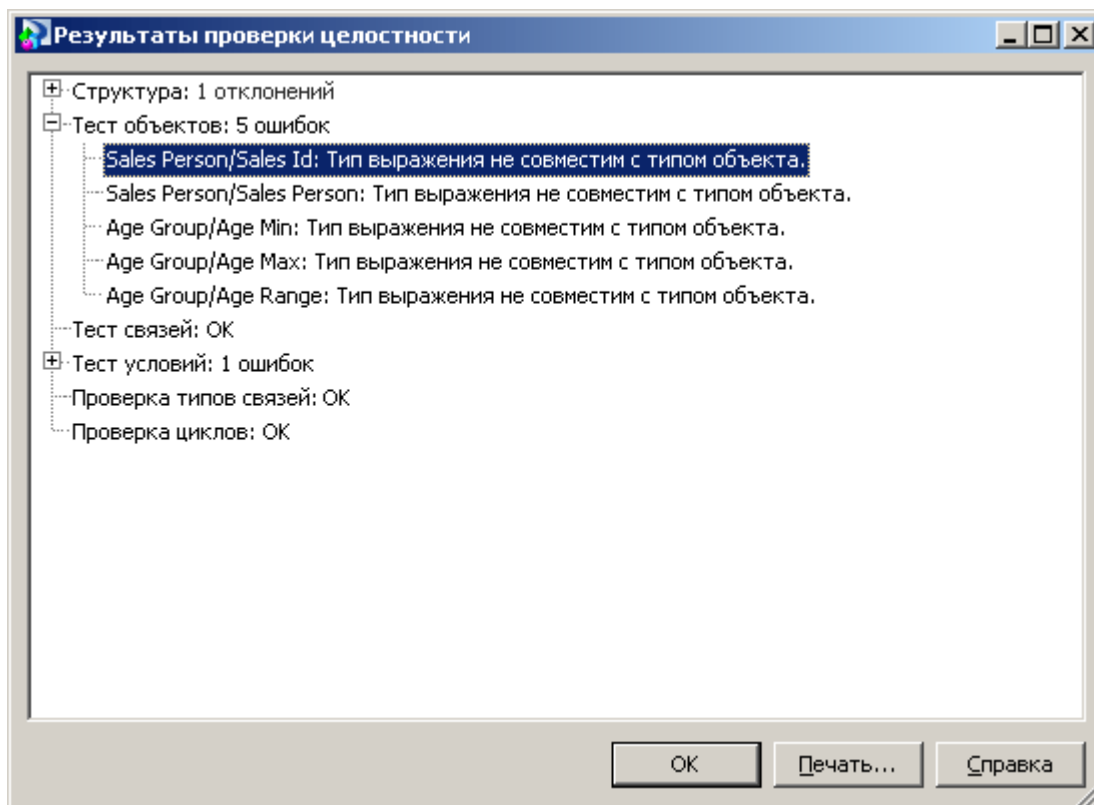
Можно выбрать "Проверить мощности множеств" независимо от параметра "Проверить все". Это позволяет проверять структуру юниверса без проверки мощностей множеств, которая отнимает много времени в зависимости от базы данных.

4. Установите флажки на компоненты для отмены проверки.
5. Установите флажок "Быстрый анализ" для проверки синтаксиса компонентов.
Или
Установите флажок "Тщательный анализ" для проверки синтаксиса и семантики компонентов.
6. Нажмите кнопку "ОК".
В окне сообщения отобразится процесс проверки юниверса.



Если параметр "Проверить целостность" не обнаружит ошибок, рядом с каждым типом ошибки отобразится "OK".

7. Щелкните значок плюс (+) рядом с типом ошибки для просмотра списка компонентов, в которых эти ошибки произошли.



Дважды щелкните мышью элемент в списке для выделения соответствующих компонентов в панели структуры.

8. Нажмите кнопку "Печать", чтобы напечатать содержимое окна.
9. Нажмите кнопку "ОК".

📌 Примечание

Перед открытием диалогового окна "Проверить циклы" убедитесь, что мощности множества элементов объединений уже обнаружены. Иначе функция ошибочно определит циклы в объединениях.

5.10.1.5 Обновление структуры юниверса

Если параметр "Проверить целостность" определит, что база данных соединения с юниверсом была изменена, можно использовать параметр "Обновить структуру", чтобы обновить содержимое панели структуры.

Данный параметр может изменить структуру юниверса, чтобы соответствовать изменениям в базе данных следующим образом.

Если	В средстве проектирования выполняются следующие действия
Столбцы добавлены в таблицы	Добавляет столбцы в соответствующие таблицы в юни-версе.
Столбцы удалены из таблиц	Отображает сообщение с предупреждением, что столбцы и соответствующие объединения необходимо удалить.
Таблицы удалены из базы данных	Отображает сообщение с предупреждением, что та-блицы и соответствующие объединения необходимо удалить.
Таблицы переименованы в базе данных	Отображает сообщение, что соответствующие таблицы больше не распознаются в юниверсе. Необходимо пере-именовать данные таблицы, чтобы они соответствовали таблицам в базе данных. Если имена все еще не соот-ветствуют друг другу, средство создания юниверсов воз-вращает сообщение, в котором говорится, что переиме-нованные таблицы не существуют в базе данных.
Изменения в базе данных не производились	Отображает сообщение, что обновление не требуется.

Чтобы обновить структуру юниверса,

- Выберите "Вид" > "Обновить структуру".
- Отобразится окно с сообщением об изменении в базе данных или о том, что обновления не требуется, если изменения не были произведены.

6 Устранение проблем объединений в схеме

6.1 Обзор

В данной главе описаны типы проблем, которые могут возникнуть во время создания объединений между таблицами в схеме. В ней объясняется, как можно определить и устранить данные проблемы объединений, чтобы убедиться, что пути объединений, используемые запросами, выполняемыми в юниверсе, возвращают верные результаты. Проблемы объединений необходимо устранить до построения юниверса.

6.2 Что собой представляет проблема пути объединения?

Путь объединения – это серии объединений, которые запрос может использовать для получения доступа к данным в таблицах, связанных с помощью объединений.

Проблемы пути объединения могут возникать из ограниченного способа, которым связаны таблицы соответствия и реальные таблицы в реляционной базе данных. Существует три главные проблемы пути объединения, которые могут возникать во время разработки схемы.

- циклы
- ловушки разногласия
- ловушки раскрытия

Эти проблемы можно решить с помощью создания псевдонимов (копий основной таблицы), контекстов (определенных путей объединений) и использования функций, доступных в средстве создания юниверсов, которые позволяют разделить запросы на меры и контексты.

В этом разделе кратко определены таблицы соответствия и фактов, а также описаны типы проблем пути объединения, которые могут возникать во время использования данных таблиц. В этом разделе объясняется, как можно использовать псевдонимы, контексты и другие функции средства создания юниверсов, чтобы устранить проблемы пути объединения в схеме юниверса.

В средстве создания юниверсов, как правило, создаются объединения между таблицами соответствия и фактов.

6.2.1 Что собой представляет таблица соответствия

В таблице соответствия (или измерении) содержатся сведения, связанные с сущностью или темой. Например, в таблице соответствия содержатся географические сведения о клиентах: имена, номера телефонов, а также городах и странах, в которых они проживают.

В средстве создания юниверсов измерение и объекты описания, как правило, выводятся из таблиц соответствия.

6.2.2 Что собой представляет таблица фактов

В реальной таблице содержатся статистические сведения о транзакциях. Например, в ней могут содержаться цифры: доход с продаж или прибыль.

Большинство мер в юниверсе определены на основе реальных таблиц.

6.2.3 Какие типы путей объединения возвращают неверные результаты?

Запросы могут возвращать неверные результаты из-за ограничений в способе выполнения объединений в реляционных базах данных. В зависимости от того, как связаны таблицы соответствия и реальные таблицы в схеме, пути объединения могут получать экземпляры, где слишком мало или слишком много строк.

Следующие типы путей объединения могут получать неверные результаты.

Тип пути объединения	Возвращает	Описание
Цикл	Слишком мало строк	Объединения создают несколько путей между таблицами.
Схождение объединений "многие-к-одному"	Слишком много строк	Объединений "многие-к-одному" из двух таблиц фактов сходятся в одной основной таблице. Данный тип схождения объединений может вызвать проблему пути объединения под названием ловушка разногласий.

Тип пути объединения	Возвращает	Описание
Серийные объединения "многие-к-одному"	Слишком много строк	Объединение "многие-к-одному" связывает таблицу, которая, в свою очередь, связана с помощью объединения "один-ко-многим". Данный тип развертывания объединений "один-ко-многим" может вызвать проблему пути объединения под названием ловушка раскрытия.

6.2.4 Определение и устранение проблем объединений

Средство создания юниверсов предоставляет несколько способов определения и устранения проблем объединений. Каждый из данных способов описан в соответствующем разделе.

Чтобы определить и устранить проблемы объединений, можно использовать следующие способы.

Проблема объединений	Определенная с помощью	Устраненная с помощью
Цикл	<ul style="list-style-type: none"> Определить псевдонимы Определить контексты Определить циклы Проверить целостность Визуальный анализ схемы 	Создание псевдонимов и контекстов, чтобы прервать циклы.
Ловушка разногласия (преобразование объединений "многие к одному")	Визуальный анализ схемы таблиц.	<ul style="list-style-type: none"> Создание контекста. Использование функции "Несколько инструкций SQL" для каждой меры Создание нескольких юниверсов (только Web Intelligence).
Ловушка раскрытия (серийные объединения "многие к одному")	Визуальный анализ схемы таблиц.	<ul style="list-style-type: none"> Создание псевдонима, контекста с помощью псевдонима и влияющих объектов мер на основе данных псевдонимов. Использование нескольких инструкций SQL для каждой меры

Большинство проблем пути объединения устраняется с помощью создания псевдонима и реализации контекста. В средстве создания юниверсов можно использовать инструменты автоматического определения циклов, чтобы определять циклы в схеме, и автоматическое определение контекстов, чтобы определять, где находятся ловушки разногласия. Однако чтобы устранить ловушки раскрытия, необходимо провести анализ схемы визуально и создать псевдонимы, а если необходимо, и контексты вручную.

6.3 Определение псевдонимов

Псевдонимы – это ссылки на существующие объекты в схеме. Таблица "Псевдоним" точно дублирует исходную таблицу (основная таблица) с другим именем. Данные таблицы "Псевдоним" и исходной таблицы совпадают, но другое имя "вводит в заблуждение" SQL запроса о том, что пользователь использует две различные таблицы.

Схема юниверса "Пляж" содержит две таблицы псевдонимов: "Страна_Курорт" (псевдоним для таблицы "Страна") и "Спонсор" (псевдоним для таблицы "Заказчик"). Для каждой таблицы псевдонимов в скобках выводится имя исходной таблицы.

6.3.1 Как использовать псевдонимы в схеме?

Использование псевдонимов происходит по двум причинам.

- Чтобы использовать таблицу в запросе несколько раз. Это главная причина использования псевдонимов. Сюда также входит использование псевдонимов, чтобы прерывать циклы и устранять ловушки раскрытия. В примере юниверса Beach содержится два псевдонима: Resort_Country для страны и спонсор для клиента.
- Чтобы сократить имя таблицы для сохранения набора при создании freehand SQL.

→ Совет

Другая причина использования псевдонимов – создание псевдонима для каждой таблицы, так как она вставляется в схему. Потом необходимо создать схему с помощью использования таблиц псевдонимов, а не исходных базовых таблиц. Переместите основные таблицы из структуры основного юниверса. Это позволит дать значимые имена таблицам и устранил необходимость снова создавать основные разделы структуры юниверса. Необходимо ли давать псевдоним основной таблице на более позднем этапе?

6.3.1.1 Использование псевдонимов для прерывания циклов

Наиболее распространенное использование псевдонимов в среде юниверса – прерывание потенциальных циклов в использовании общих таблиц. Цикл – это набор объединений, который определяет закрытый путь между таблицами в схеме. Циклы встречаются, если объединения создают несколько путей между таблицами соответствия.

Можно использовать псевдоним, чтобы прервать цикл с помощью предоставления альтернативной таблицы для исходной таблицы соответствия, которая используется для нескольких путей запросов. Данное использование псевдонимов описано в разделе [Устранение циклов \[страница 215\]](#).

6.3.1.2 Использование псевдонимов для устранения ловушек раскрытия

Псевдонимы также используют, чтобы устранить потенциальные ловушки раскрытия. Это может произойти в пути объединения "один ко многим", что может вернуть завышенные результаты, если совокупности суммируются на конце объединений "многие". Данное использование псевдонимов описано в разделе [Устранение ловушек разногласия \[страница 241\]](#).

6.3.2 Создание псевдонимов

Псевдонимы можно создавать вручную или с помощью средства создания юниверсов, которое поддерживает автоматическое определение возможных псевдонимов, позволяющих устранить цикл в пути объединения.

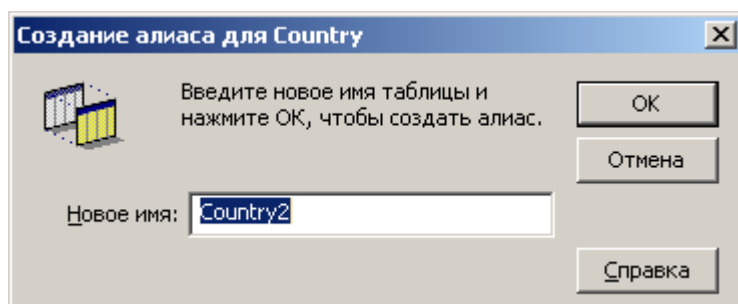
Необходимо создать псевдоним вручную для устранения ловушки раскрытия. Можно также создавать псевдонимы вручную при создании схемы, которая использует только псевдонимы, а не базовые таблицы.

Автоматическое определение и создание псевдонимов для прерывания циклов описано в разделе [Определение и создание псевдонима \[страница 226\]](#).

6.3.2.1 Создание псевдонима вручную

Чтобы создать псевдоним вручную,

1. Выберите таблицу для создания псевдонима.
2. Выберите "Вставка" > "Псевдоним"
Или
Нажмите кнопку "Вставить псевдоним".
Откроется окно "Псевдоним". Предлагается ввести имя для нового псевдонима.



3. Введите новое имя или оставьте предложенное.

Примечание

Имя псевдонима должно соответствовать его роли выделения в базовой таблицы. Например, Resort country – это псевдоним для Country. Resort Country используется для запросов, а базовая таблица Country используется в запросах.

4. Нажмите кнопку "OK".
Таблица отобразится на панели структуры.
5. Создайте любое объединение между псевдонимами и другими таблицами в схеме.

→ Совет

Чтобы избежать смешивания базовых таблиц с псевдонимами, необходимо отобразить этот псевдоним с именем базовой таблицы. Это представлено в заголовке таблицы следующим образом. Выберите "Инструменты" > "Параметры" > "Графическое отображение", а потом установите флажок "Псевдоним".

6.3.2.2 Переименование псевдонима

Можно переименовать псевдоним в любое время. Допущения имен и псевдонимов таблиц зависят от RDBMS. Можно переименовать псевдоним непосредственно с помощью переименования таблицы или в списке псевдонимов юниверса.

Непосредственное переименование псевдонима

Чтобы переименовать псевдоним,

1. Выделите таблицу и нажмите "Редактировать" > "Переименовать таблицу".
Или
Щелкнув правой кнопкой мыши таблицу, выберите "Переименовать таблицу" в контекстном меню.
Откроется диалоговое окно "Переименовать таблицу".

The screenshot shows a dialog box titled "Переименовать таблицу 'Sponsor'". It contains the following fields and controls:

- Имя таблицы:** A text box containing the word "Sponsor".
- Владелец:** An empty text box.
- Квалификатор:** An empty text box.
- Установить регистр в:** A group box containing two checkboxes:
 - ☐ Верхний регистр
 - ☐ Нижний регистр
- Buttons:** "OK", "Отмена", and "Справка" are located on the right side of the dialog.

2. Введите в поле "Имя таблицы" новое имя.
Доступность полей "Владелец" и "Квалификация" определяется конкретной базой данных. Если они активны, то можно внести необходимые изменения.
3. Установите флажок "Верхний регистр", если имя состоит из прописных букв.
Или
Установите флажок "Нижний регистр", если имя состоит из строчных букв.
4. Нажмите кнопку "ОК".

Переименование псевдонима в списке

Чтобы переименовать псевдоним в списке,

1. Выберите "Инструменты" > "Список псевдонимов".
2. Отобразится список псевдонимов. В нем перечислены все псевдонимы активного юниверса.
3. Щелкните имя псевдонима в списке.
4. В текстовом поле "Создать имя" введите новое имя выбранного псевдонима.
5. Нажмите "Применить".
6. Нажмите кнопку "ОК".

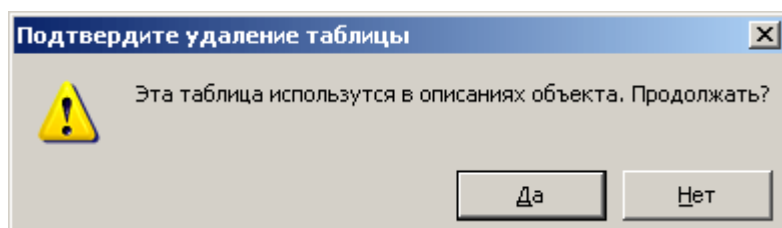
6.3.2.3 Удаление псевдонима

Удалите псевдоним таким же способом, которым удаляете таблицу. Определив объекты с помощью псевдонима, необходимо изменить данные объекты до удаления псевдонима, чтобы они использовали другую таблицу или удалили объекты, если они больше не нужны.

Если не изменить или удалить объекты с помощью удаленных псевдонимов, запросы с помощью данных объектов сгенерируют ошибки в Web Intelligence.

Чтобы удалить псевдоним,

1. Щелкните псевдоним и выберите "Редактировать" > "Удалить".
Или
Правой кнопкой щелкните псевдоним и выберите "Удалить" в контекстном меню.
Или
Щелкните псевдоним и нажмите клавишу DELETE.
Если объекты используют псевдонимы, отобразится следующее сообщение.



Если объекты не используют псевдонимы, окно с подтверждением не отобразится. Псевдоним удаляется мгновенно.

2. Нажмите "Да"

Псевдоним удален на панели структуры.

6.4 Определение контекстов

Контексты – это совокупности объединений, которые обеспечивают путь допустимого запроса для Web Intelligence для генерации SQL.

6.4.1 Использование контекстов в схеме

Контексты можно использовать в юниверсе для следующих целей.

- Устранение циклов.
- Устранение ловушек разногласия.
- Содействие в некоторых устранениях ловушек раскрытия.
- Содействие в выявлении несовместимости для объектов поддержкой агрегирования.

6.4.1.1 Использование контекстов для устранения циклов

Чаще всего контексты используются для разделения путей запросов так, чтобы один запрос возвратил данные для одной реальной таблицы, а другой – для другой реальной таблицы. Используйте контексты, чтобы направить путь объединения в схему, которая содержит несколько реальных таблиц. Псевдонимы не соответствуют данной схеме. Данное использование контекста описано в разделе [Устранение циклов \[страница 215\]](#).

6.4.1.2 Использование контекста для устранения ловушек разногласия и раскрытия

Контексты также используются для устранения потенциальных ловушек разногласия. Они могут появляться, если два пути объединения "многие к одному" сходятся в одной таблице. Несколько строк могут быть возвращены для одного измерения, являясь причиной завышенных результатов. Контексты могут разделять запрос так, что правильное количество строк возвращается для измерения. Контексты также используются с псевдонимами для устранения ловушек раскрытия. Данное использование контекстов описано в разделе [Устранение ловушек разногласия \[страница 241\]](#).

6.4.1.3 Использование контекстов для определения несовместимости AggregateAwareness

Можно использовать контексты для отстранения контекстов, несовместимых с объектом, который использует функцию @AggregateAware в своем определении, от использования в запросе с агрегированным объектом.

6.4.2 Создание контекста

Можно разрешить средству создания юниверсов автоматически определять контексты. Также можно создавать контексты вручную.

При использовании контекста для устранения цикла или ловушки разногласий необходимо всегда разрешать определение контекстов средству создания юниверсов. Однако для устранения ловушке разногласий (другая проблема пути объединения) необходимо вручную строить контекст.

Автоматическое определение контекстов для устранения цикла описано в разделе [Устранение циклов \[страница 215\]](#).

📘 Примечание

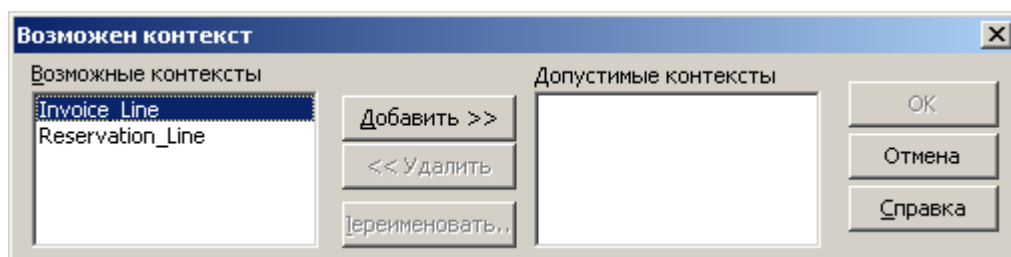
При создании одного или более контекстов все объединения должны быть включены в один или несколько контекстов. Если таблица связана объединением, не включенным в контекст, объединение не будет учитываться при выполнении запроса.

Следующие процедуры описывают, как можно создавать контекст автоматически и вручную.

6.4.2.1 Автоматическое создание контекста

Чтобы автоматически создать контекст

1. Нажмите "Инструменты" > "Автоматическое определение" > "Определить контексты".
Откроется окно вероятных контекстов. В нем предлагаются вероятные контексты для данной схемы. Данные вероятные контексты необходимы для устранения циклов или ловушек разногласия, так как ловушки разногласия возникают в ветви, где сходятся два контекста.



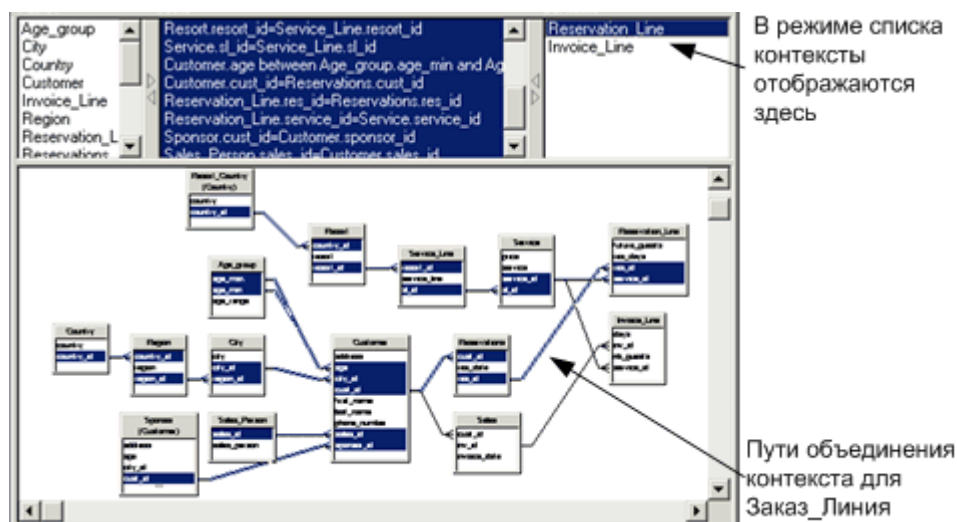
2. Выберите контекст в списке вероятных контекстов и нажмите кнопку "Добавить".

3. Выполните повторно шаг 2 для каждого вероятного контекста в списке.

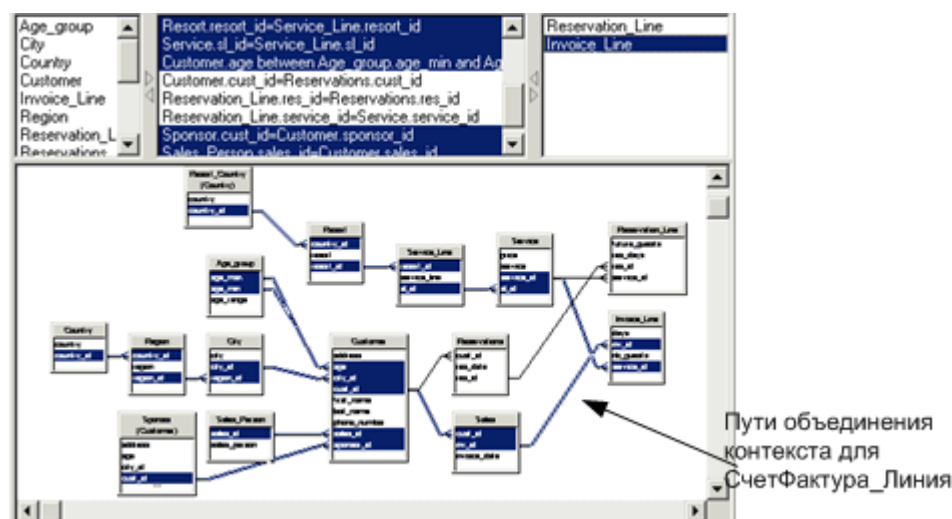
Примечание

После добавления вероятного контекста в список принятых контекстов, переименуйте контекст следующим образом. Выберите контекст и нажмите кнопку "Переименовать". Откроется окно "Редактировать". Введите новое имя и нажмите "ОК".

4. Нажмите кнопку ОК.
Контексты перечислены на панели контекстов, если режим списка активен ("Вид" > "Режим списка"). Контекст строки счета показан ниже.



5. Контекст для Invoice_Line показан ниже.



6.4.2.2 Создание контекста вручную

Чтобы создать контекст вручную,

1. Выберите "Вставить" > "Контекст".
Или
Нажмите кнопку "Вставить контекст".
Откроется окно "Создать контекст".

2. Введите имя контекста в текстовом окне "Имя контекста".
3. Выберите все объединения, определяющие контекст в списке текущих объединений контекстов.
При создании контекста необходимо иметь следующие параметры.
4. Нажмите кнопку "Определить" для отображения объединений, создающих предлагаемый контекст с именем контекста.
5. Установите флажок "Показывать только выбранные" для отображения выбранных объединений.
6. Нажмите кнопку "Проверить".
Средство создания юниверсов проверяет выбранные объединения на наличие циклов.
7. Введите описание данных, возвращенных контекстом. При выполнении запросов, использующих путь контекста, все пользователи Web Intelligence смогут прочесть данный вспомогательный текст.
Этот текст может быть полезен для пользователей.
8. Нажмите кнопку "ОК".
Контекст создан.

6.4.3 Редактирование контекста

Можно использовать редактора контекста для изменения следующих свойств контекста.

- Имя
- Объединения, включенные в контекст
- Описание

Также можно проверить контекст на наличие неустраненных циклов.

6.4.3.1 Редактирование свойств контекста

Чтобы отредактировать свойства контекста,

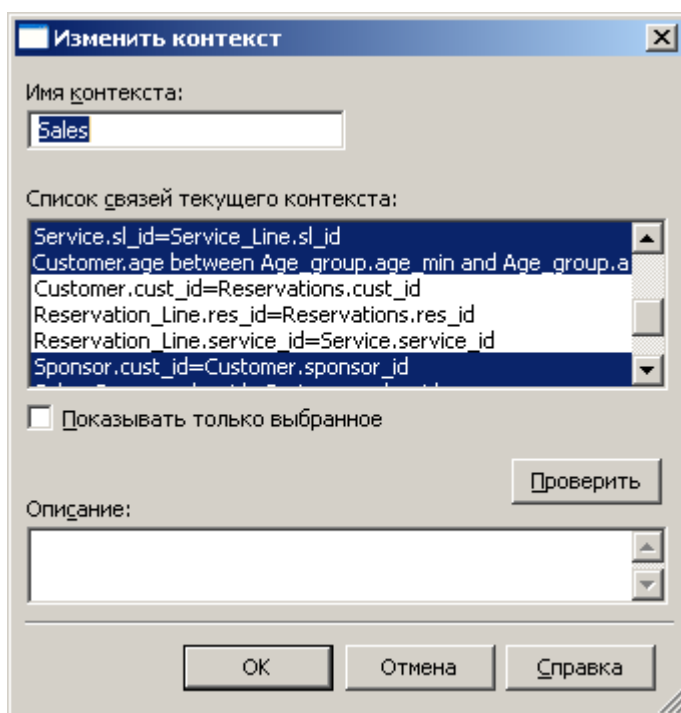
1. Выберите "Просмотр" > "Режим списка".

Над панелью структуры отобразится панель списка. В ней содержатся окна списка для таблиц, объединений и контекстов на панели структуры.



2. Дважды щелкните имя контекста на панели списка "Контексты".

Откроется окно "Редактировать контекст".



3. Если необходимо изменить имя контекста, введите новое имя в окне "Имя контекста".
4. Выберите выделенное объединение для удаления его из контекста.
Или
Щелкните мышью невыделенное объединение, чтобы добавить его к контексту.
5. Введите описания контекста.
6. Нажмите кнопку ОК.
В контексте отобразятся изменения.

6.4.4 Удаление контекста

В списке "Контекст" на панели списка можно удалить контекст в любое время. Добавив или удалив таблицу или объединение в контексте, необходимо удалить контекст до выполнения изменений с таблицей или объединением.

Выполнив изменение, можно вручную создать контекст, если он используется для устранения ловушки разногласий, или использовать параметр "Удалить контексты", чтобы автоматически удалить новый контекст, если он используется для устранения цикла. Дополнительные сведения об удалении контекстов см. в разделе [Определение и создание контекста \[страница 228\]](#).

6.4.4.1 Удаление контекста в списке контекстов

Чтобы удалить контекст в списке контекстов,

1. Убедитесь, что режим списка активен (выберите "Вид" > "Режим списка").
2. В окне списка "Контексты" правой кнопкой щелкните имя контекста и выберите "Удалить" в контекстном меню.
Или
В окне списка "Контексты" щелкните имя контекста и выберите "Редактировать" > "Удалить".
Контекст удален из списка.

6.4.5 Обновление контекстов

Если структура юниверса изменена, контексты не удаляются автоматически. Если необходимо добавить или удалить таблицы или объединения в структуре, сначала необходимо обновить все контексты.

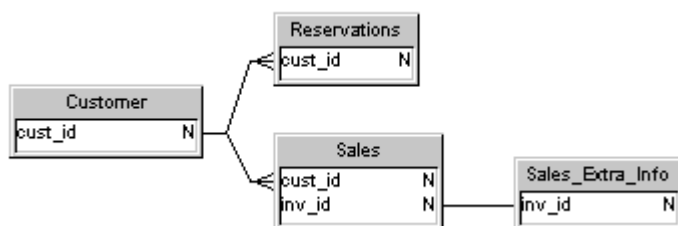
Если в структуре было выполнено простое изменение, можно обновить объединение, которое включено в контекст вручную, с помощью поля "Редактировать контекст" или панели списков. Однако если в структуре юниверса были выполнены значительные изменения, необходимо удалить текущие контексты и создать их снова.

6.4.6 Пути объединения, которые предотвращают обнаружение контекста

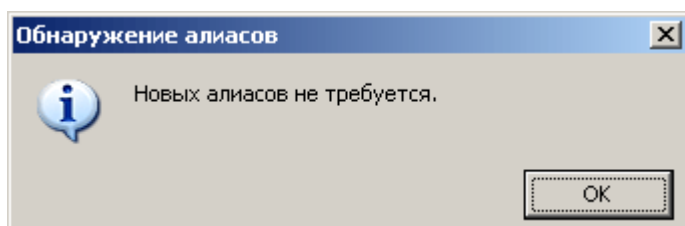
Свойство связи "один к одному", расположенное в конце пути объединения, может помешать выполнению функции определения контекста в средстве создания юниверсов. Можно устранить данную проблему, изменив мощность множества таблицы в конце пути объединения на "один ко многим".

Пример: Мощность множества "один к одному", предотвращающая определение контекста

В схеме ниже отображена таблица Sales_Extra_Info, в которой содержатся подробные сведения о каждой продаже. Она объединена с помощью объединения "один к одному" в таблице "Продажи".



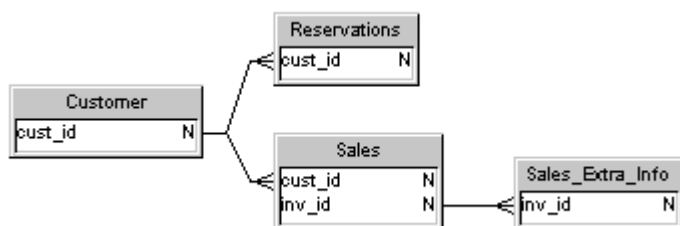
Если визуально проверить пути объединения, в схеме существует два контекста: контекст предварительных заказов и контекст продаж. Однако если автоматически определить контексты на данном типе пути объединения ("Инструменты" > "Автоматическое определение" > "Определить контексты"), отобразится следующее сообщение.



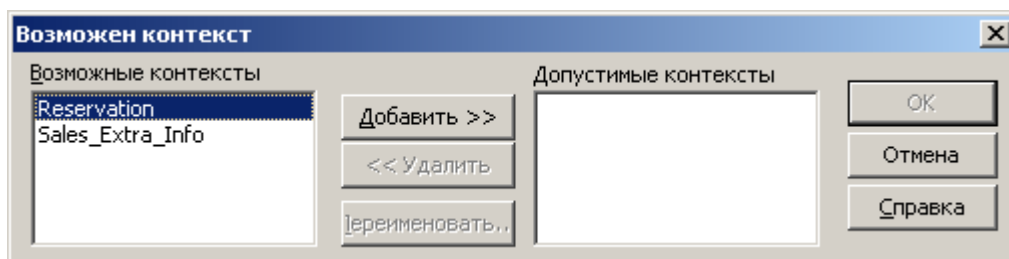
Средство создания юниверсов не учитывает объединение "один к одному" в конце пути объединения в определении контекста, поэтому не учитывается существование двух контекста.

6.4.6.1 Изменение мощности множества, чтобы позволить определение контекста

Устраните данную проблему, установив мощность множества объединения, связывающую Sale_Extra_Info в параметром "Продажи" и объединением "один ко многим". Оно также может быть "многие к одному". Важный фактор – не получить объединение "один ко многим" в конце пути объединения. В схеме ниже показано объединение "один ко многим" в конце пути объединения.



Выполнив запрос "Определить контексты", определяются два контекста, как показано ниже.



6.4.7 Как контексты влияют на запросы?

В зависимости от того, как пользователям Web Intelligence разрешается использовать объекты, определенные в структурах схемы, контексты могут привести к выполнению трех типов запросов.

- Неоднозначные запросы
- Подразумеваемые запросы
- Несовместимые запросы

Можно выполнить три типа запросов в Web Intelligence, чтобы проверить SQL, сгенерированный с помощью контекстов. Если один из данных типов запроса выдает ошибку или возвращает неверные данные, необходимо проанализировать связанные пути объединения.

6.4.7.1 Неоднозначные запросы

Конечному пользователю предлагается выбирать между одним путем запроса и другим. Это происходит, если в запрос включены объекты, которые, когда используются вместе, не предоставляют достаточно сведений для определения между контекстами.

Если запрос неоднозначный, Web Intelligence отображает диалоговое окно, которое предлагает пользователю выбрать один или другой контекст. Если пользователь выбирает контекст, соответствующие таблицы и объединения вставляются в запрос SQL.

Пример: Выполнение неоднозначного запроса

Пользователь Web Intelligence выполняет следующий запрос.

Предоставить услуги, которые используются каждой возрастной группой посетителей для каждого курорта.

 Service  Age group  Resort

Если запрос выполнен, отобразится диалоговое окно с просьбой выбрать контекст, в данном случае контексты "Предварительные заказы" и "Продажи".

Пользователю необходимо выбрать, услуги зарезервированы ил оплачены. Если выбрать контекст "Предварительные заказы", сгенерируется следующее выражение SQL.

```
SELECT Service.service, Age_group.age_range, Resort.resort FROM
Service, Age_group, Resort, Customer, Reservations, Reservation_Line,
Service_Line WHERE ( Resort.resort_id=Service_Line.resort_id )
AND ( Service.sl_id=Service_Line.sl_id ) AND
( Customer.age between Age_group.age_min and
Age_group.age_max ) AND ( Customer.cust_id=Reservations.cust_id )
AND ( Reservation_Line.res_id=Reservations.res_id ) AND
( Reservation_Line.service_id=Service.service_id )
```

Объединения, на которые ссылается другой контекст (Продажи) не отображаются в SQL.

6.4.7.2 Подразумевающиеся запросы

Запрос Web Intelligence выполнен без уведомления конечного пользователя о выборе контекста. В запросе содержится достаточно сведений, чтобы контекст был подразумевающимся. Например, пользователь выполняет следующий запрос.

Предоставить число будущих гостей по возрастной группе для каждой услуги.

 Service  Age group  Future guests

Если запрос выполнен, данные возвращаются без уведомления пользователя о выборе контекста. Объект Будущие гости – это сумма в таблице Reservation_Line, которая является частью контекста "Предварительные заказы". Web Intelligence подразумевает, что контекст "Предварительные заказы" является единственным, чтобы использовать для запроса.




6.4.7.3 Несовместимые запросы

Объекты из двух различных контекстов объединены в запросе. Две инструкции Select синхронизированы, чтобы выводить возвращаемые данные в отдельных таблицах.

Пример: Выполнение несовместимого запроса

Пользователь Web Intelligence выполняет следующий запрос.

Предоставить общее число гостей компании по возрастной группе и месяцы, когда были сделаны предварительные заказы.

 Number of guests  Age group  Reservation Month

Если запрос выполнен, уведомление не отображается, так как Web Intelligence влияет на использование контекстов "Предварительные заказы" и "Продажи". Инструкции Select для обоих контекстов синхронизированы следующим образом.

```
SELECT
  Age_group.age_range,
  sum(Invoice_Line.nb_guests)
FROM
  Age_group,
  Invoice_Line,
  Service_Line,
  Sales,
  Customer,
  Service
WHERE
  ( Customer.cust_id=Sales.cust_id )
  AND ( Invoice_Line.inv_id=Sales.inv_id )
  AND ( Invoice_Line.service_id=Service.service_id )
  AND ( Service.sl_id=Service_Line.sl_id )
  AND ( Customer.age between Age_group.age_min and Age_group.age_max )
  AND ( Service_Line.service_line = 'Accommodation' )
GROUP BY
  Age_group.age_range
```

Запрос разделен на две части.

- Возрастная группа и количество гостей

- Месяц предварительного заказа

При получении результатов двух запросов Web Intelligence объединяет результаты (с помощью запроса Возрастная группа). Затем результаты отображаются в двух таблицах в одном отчете, как показано ниже.

18-30

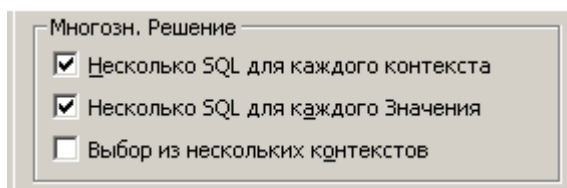
Number of guests	Reservation Month
451.00	Apr
	Aug
	Feb
	Jan
	Jun
	May
	Nov
	Oct
	Sep

Чтобы несовместимые запросы выполнялись в Web Intelligence, необходимо выбрать несколько SQL-инструкций в средстве создания юниверсов для каждого параметра контекста. Это описано в следующем разделе.

6.4.7.4 Выбор нескольких SQL-инструкций для каждого контекста

Чтобы выбрать несколько SQL-инструкций для каждого контекста,

1. Выберите "Файл" > "Параметры".
Откроется диалоговое окно "Параметры юниверса".
2. Выберите вкладку "SQL".
Откроется страница SQL.
3. Установите флажки на несколько SQL-инструкций для каждого контекста.



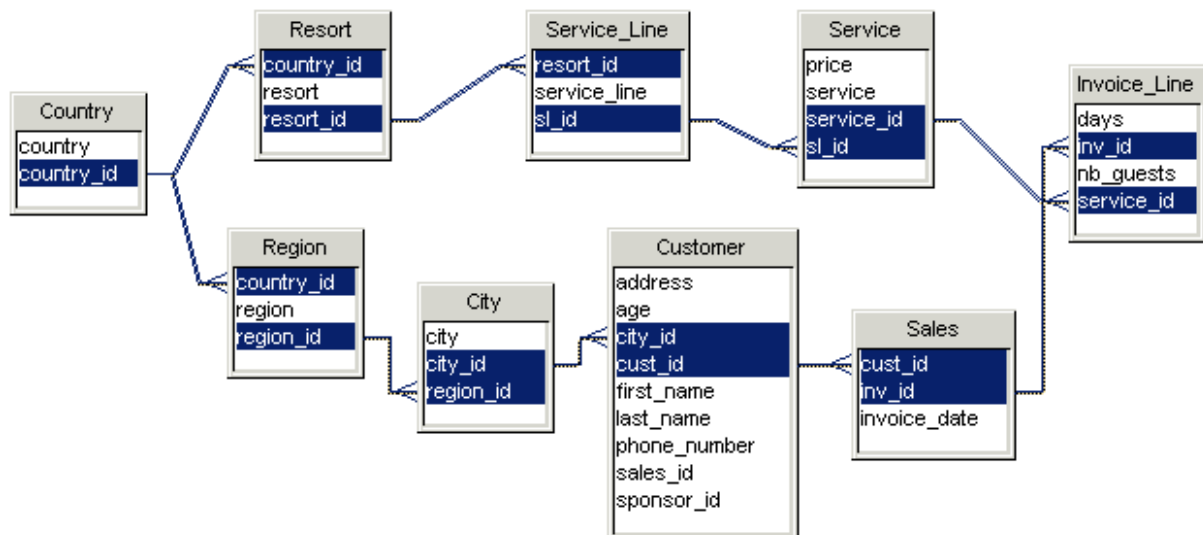
4. Нажмите кнопку ОК.

6.5 Устранение циклов

В схеме реляционной базы данных общий тип пути объединения, который возвращает слишком мало строк, называется циклом.

6.5.1 Что такое цикл?

Цикл – это набор объединений, который определяет закрытый путь через набор таблиц в схеме. Циклы возникают, если объединения создают несколько путей между таблицами соответствия. Пример цикла приведен ниже.



В схеме содержатся два связанных набора сведений.

Для каждого...	следующие сведения связаны
Курорт	Доступные сервисные линии, услуги для каждой сервисной линии, сведения о счетах для каждой услуги и страна, где расположен курорт.
Клиент	Город, регион и страна, где живет клиент, продажи для клиента и сведения счета по каждой продаже.

Данные два набора сведений связаны в общий путь объединения, который создает цикл. Таблица соответствия "Страна" может представлять страну, где находится курорт, или страну, где живет клиент.

6.5.1.1 Почему циклы возникают в схеме юниверса, а не в базе данных?

В базе данных несколько путей между таблицами могут быть допустимыми и реализованными, чтобы отвечать потребностям особых пользователей. Если каждый путь включен в запрос автономно, он возвращает уникальный набор результатов.

Однако для схемы, разрабатываемой в средстве создания юниверсов, часто требуется разрешить запросы с несколькими путями, которые могут не разрабатываться в реляционной базе данных. Потому возвращаемые сведения могут быть неверными.

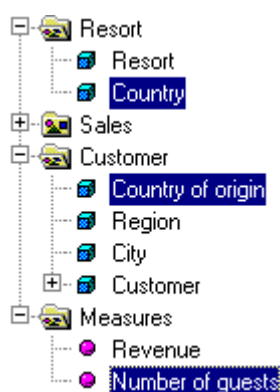
Возвращаемые строки являются пересечением результатов каждого пути. Поэтому возвращается меньше строк, чем ожидалось. Часто трудно определить проблему при проверке результатов.

6.5.2 Как цикл влияет на запросы?

При создании юниверса на основе вышеприведенной структуры любой запрос, выполненный в таблицах в цикле, возвращает результаты, где значения страны для курортов и значения страны для происхождения клиента являются эквивалентами. Данное двойное ограничение на основе общедоступной таблицы соответствия "Страна" возвращает меньше строк, чем ожидалось.

Пример: Цикл возвращает неверные результаты

Создайте следующие объекты с помощью схемы, в которой содержится вышеуказанный цикл.



Запустите следующий запрос в Web Intelligence:

Для каждой курортной страны предоставьте список гостей из каждой страны, которые останавливаются на курорте.

Country Country of origin Number of guests

Результат будет следующим.

France

Country of origin	Number of guests
Germany	141.00
Japan	154.00
US	151.00

US

Country of origin	Number of guests
Germany	329.00
Japan	345.00
US	431.00

Для курортов во Франции и США есть определенное количество немецких, японских и американских туристов, которые останавливаются на курортах данных стран.

Однако при выполнении запроса с помощью юниверса с циклом результаты будут следующие.

Country	Country of origin	Number of guests
US	US	431.00

Это подразумевает, что только туристы из США останавливаются на курортах США. Из других стран нет туристов.

6.5.2.1 Что представляет собой цикл, применяемый к запросу?

Объединения в структуре используются для создания утверждения Where в выводимом SQL-коде запроса. Цель данных объединений – ограничить данные, возвращенные с помощью запроса. В цикле объединения применяют больше ограничений, чем ожидалось, а возвращенные данные являются неверными.

Утверждение Where, созданное с помощью цикла, показано ниже.

```
WHERE ( Country.country_id=Resort.country_id )
AND ( Resort.resort_id=Service_Line.resort_id )
AND ( Service_Line.sl_id=Service.sl_id )
AND ( Service.service_id=Invoice_Line.service_id )
AND ( Sales.inv_id=Invoice_Line.inv_id ) AND
( Customer.cust_id=Sales.cust_id ) AND ( City.city_id=Customer.city_id )
AND ( Region.region_id=City.region_id ) AND
( Country.country_id=Region.country_id ) AND ( Service_Line.service_line =
'Accommodation' )
```

Оба следующих объединения применяют ограничение к таблице "Страна".

- Country.country_id=Resort.country_id
- Country.country_id=Region.country_id

Параметр "Страна" служит для двух целей.

- Поиск курортной страны.
- Поиск страны происхождения клиента.

Это создает ограничение, поэтому данные возвращаются, только если курортная страна и страна клиента совпадают. В конечном отчете отображается только количество туристов из США, кто посетил курорты США.

В зависимости от характера цикла можно устранять цикл в средстве создания юниверсов с помощью псевдонима, чтобы разорвать путь объединения, или контекста, чтобы разделить два пути объединения и чтобы в запросе мог использоваться только один путь.

6.5.2.2 Как псевдоним прерывает цикл?

Псевдоним прерывает цикл с помощью использования одной и той же таблицы дважды в том же запросе для другой цели. Псевдоним идентичен основной таблице с другим именем. Данные в псевдониме идентичны исходной таблице, но другое имя "сбивает с толку" SQL и заставляет принять, что в использовании находятся две различные таблицы.

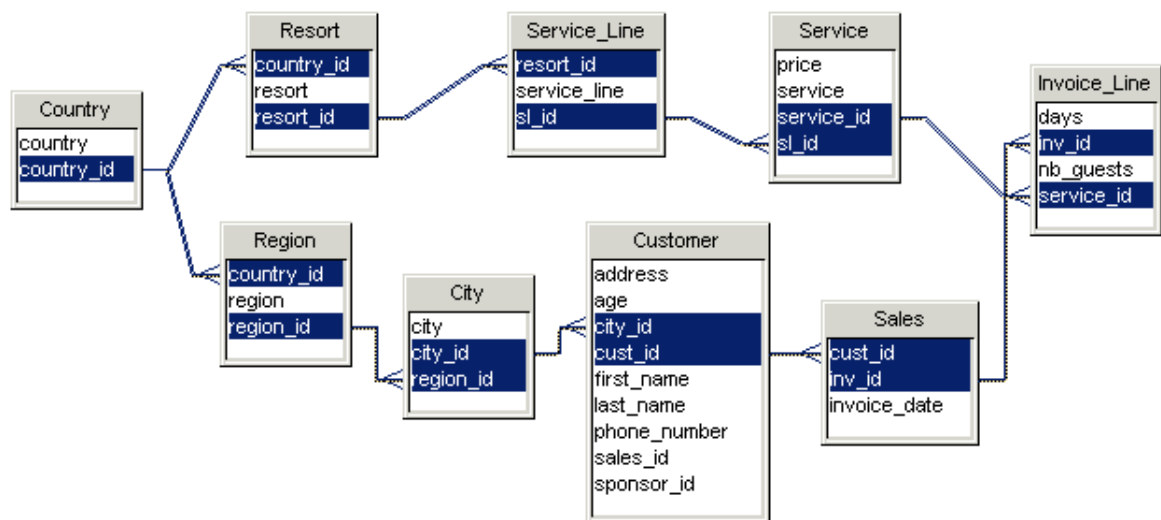
📌 Примечание

Можно устранить данный цикл с помощью создания только одной таблицы псевдонимов в использованном примере. Объединение "Регион" использует только одну таблицу "Страна", пока объединение "Выставочный зал" использует таблицу псевдонимов. Однако можно создавать отдельную таблицу псевдонимов для каждого объединения в исходной таблице. В некоторых системах реляционных баз данных это необходимо.

Пример: Прерывание цикла с помощью псевдонима

Схема, указанная ниже, и схема с циклом предыдущего раздела являются идентичными. В ней отображен путь объединения, в котором таблица соответствия "Страна" получает концы "один" двух объединений. Поэтому она может быть использована для следующих двух целей в пути объединения.

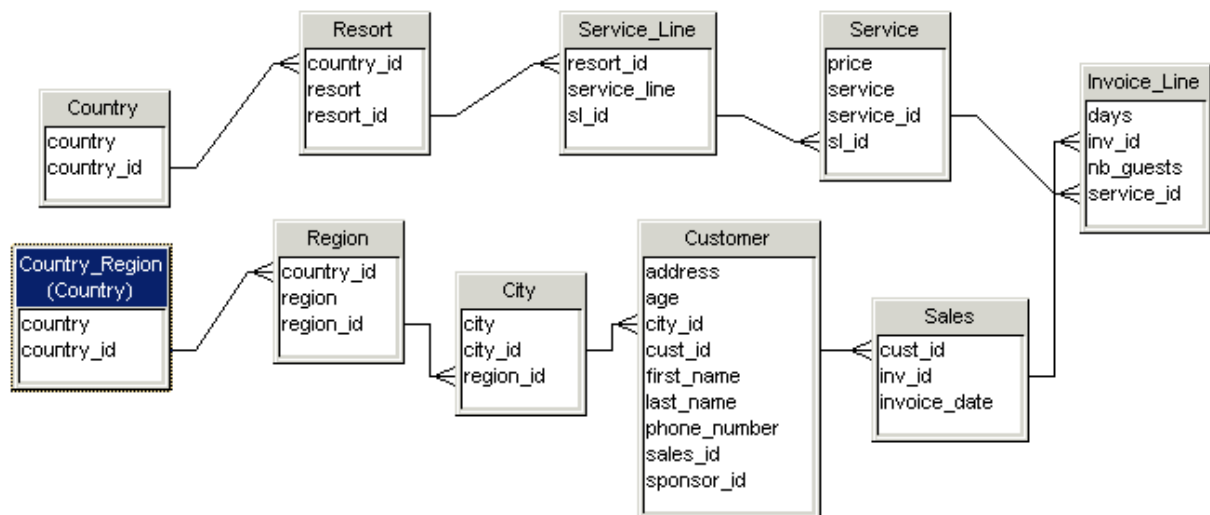
- Страны для курортов
- Страны для клиентов



Создайте псевдоним "Страна" и переименуйте его в Country_Region. Два объединения с концами "один" разделены следующим образом.

- Страна сохраняет объединение в таблице "Курорт".
- Country_Region объединен с таблицей "Регион".

Данная схема отображена ниже.



При выполнении идентичного запроса, который произвел слишком мало строк в предыдущем примере.

Для каждой курортной страны предоставьте список гостей из каждой страны, которые останавливаются на курорте.

Country
 Country of origin
 Number of guests

Утверждение Where для данного запроса выглядит следующим образом.

```
WHERE ( City.city_id=Customer.city_id ) AND  
( City.region_id=Region.region_id ) AND  
( Country.country_id=Region.country_id ) AND  
( Resort_Country.country_id=Resort.country_id ) AND  
( Customer.cust_id=Sales.cust_id ) AND ( Invoice_Line.inv_id=Sales.inv_id )  
AND ( Invoice_Line.service_id=Service.service_id )  
AND ( Resort.resort_id=Service_Line.resort_id ) AND  
( Service.sl_id=Service_Line.sl_id ) AND ( Service_Line.service_line =  
'Accommodation' )
```

Одно объединение применяет ограничение в таблице "Страна", а другое – в таблице Resort_Country. Цикл прерван.

При выполнении запроса возвращается следующая таблица.

Country	Country of origin	Number of guests
France	Germany	141,00
France	Japan	154,00
France	US	151,00
US	Germany	329,00
US	Japan	345,00
US	US	431,00

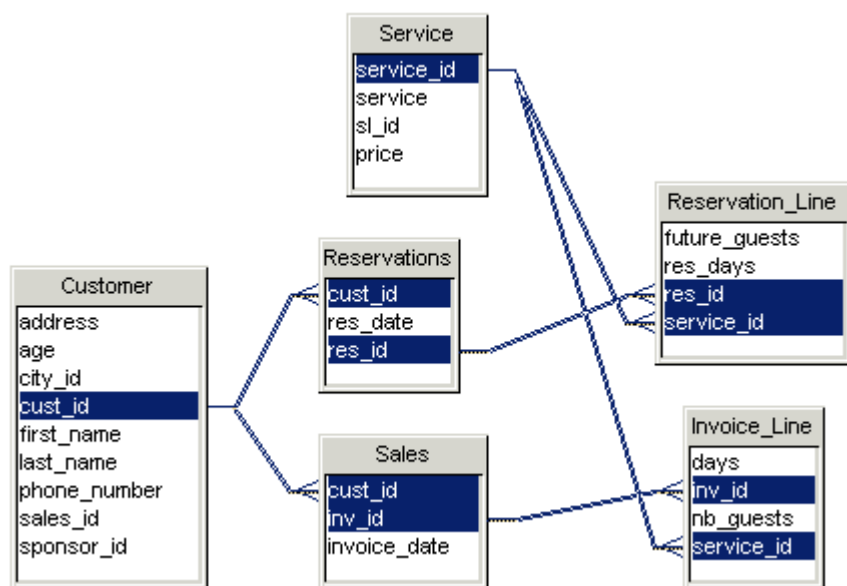
6.5.2.3 Как контекст устраняет цикл?

Контекст устраняет цикл с помощью определения набора объединений, которые указывают один определенный путь между таблицами в цикле. Это гарантирует, что объединения не будут объединены из различных путей в одном запросе SQL.

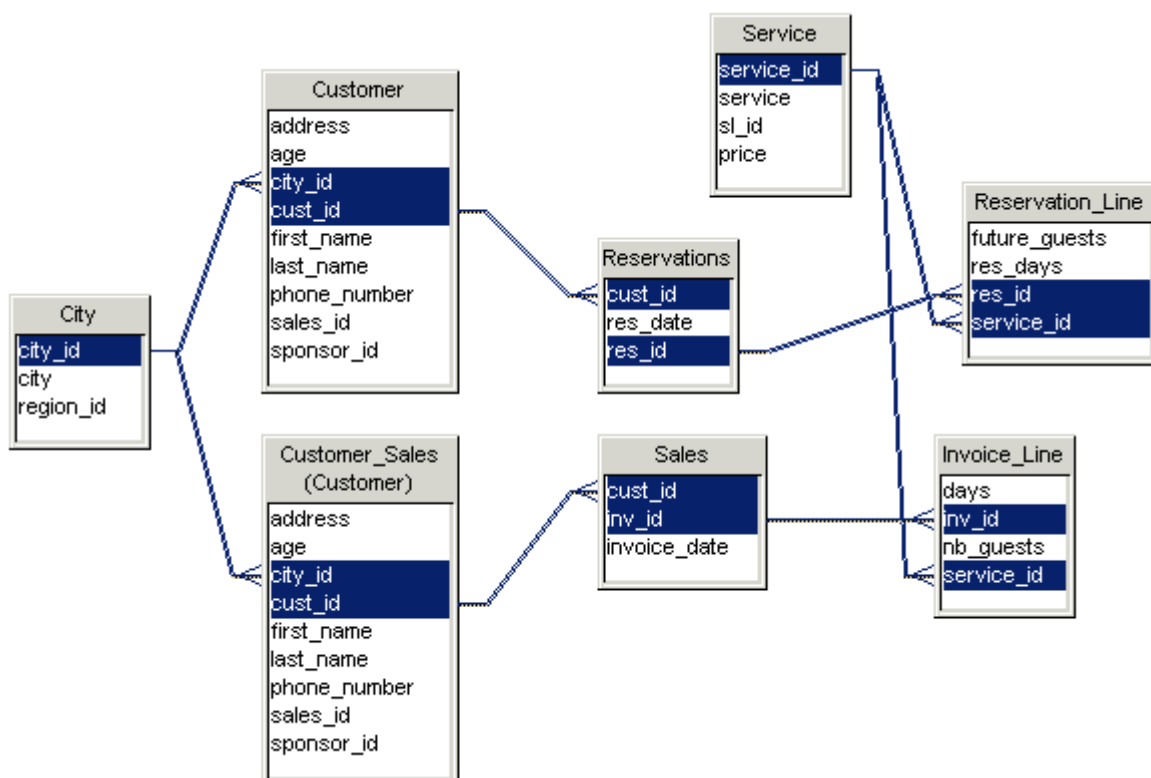
Часто в схеме используются контексты, в которых содержится несколько реальных таблиц ("несколько звездочек"), которые делают таблицы соответствия общедоступными.

Пример: Устранение цикла с помощью контекста

В схеме, указанной ниже, содержатся статистические сведения о продажах и предварительных заказах. Статистические данные, относящиеся к каждому типу транзакции, хранятся в реальных таблицах "Продажи" и "Предварительные заказы". В схеме содержится цикл, так как путь объединения может следовать за путем продаж и предварительных заказов, чтобы получить сведения об услуге.



Создав псевдоним для клиента, можно получить объединение таблиц "Клиент" и Предварительные заказы" и объединение таблиц "Customer_Sales" и "Продажи". Но если необходимо добавить таблицу "Город" в схему, можно "покончить" с циклом снова, как показано ниже.



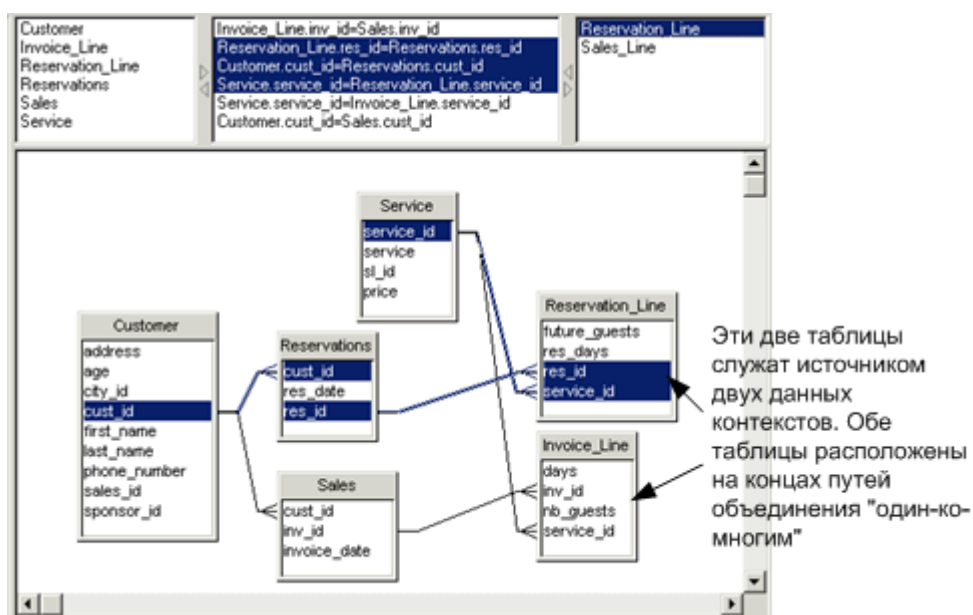
Необходимо продолжить создание псевдонимов для каждой новой таблицы при добавлении в схему. Это трудно поддерживать, а также "покончить" с распространением одинаковых объектов с помощью каждой таблицы в юниверсе.

Единственный способ устранить данный цикл – оставить его на том же месте и создать контекст, который укажет путь в обход схемы. Это гарантирует возврат запросами ответов на вопросы о для или иной транзакции, такие как: необходимы ли сведения о клиенте из ракурса продаж или предварительных заказов?

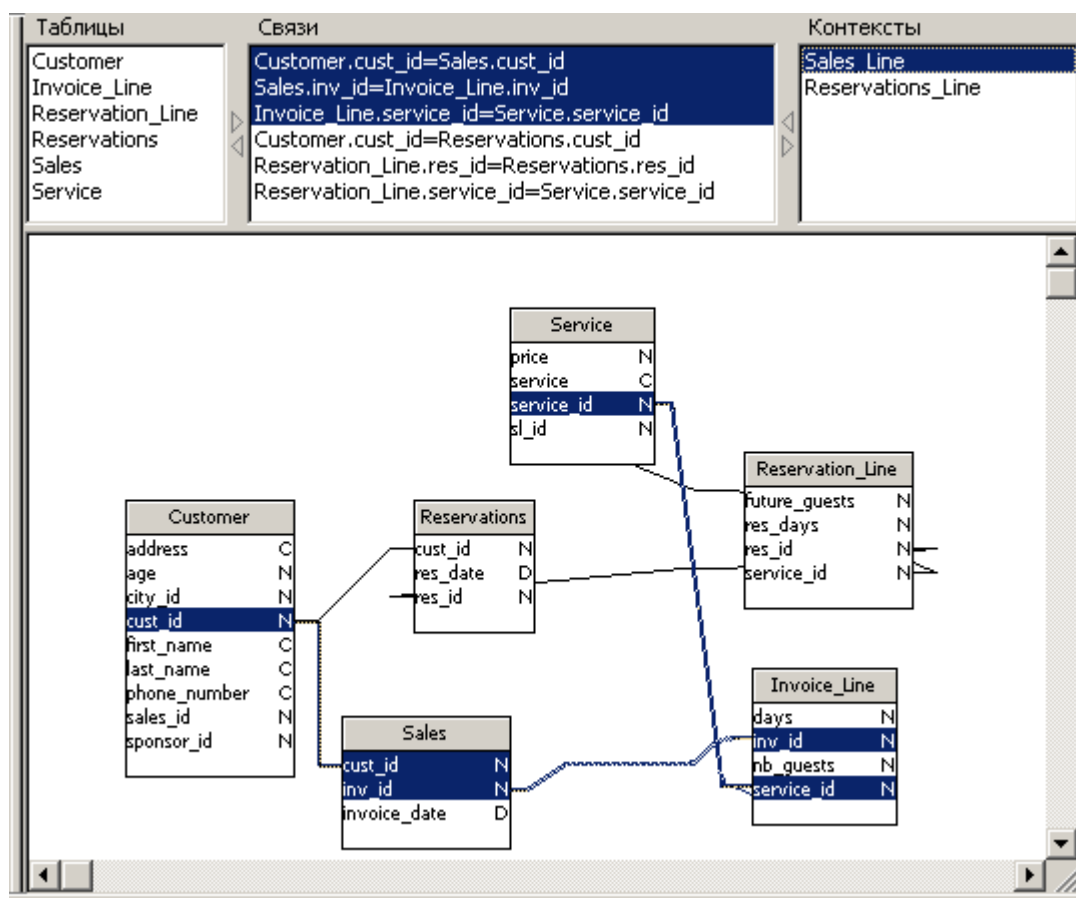
В данном примере можно проследить два различных пути от таблицы "Клиент" до таблицы "Услуга".

Для данного пути...	Средство определяет следующие контексты...
Предварительные заказы и Reservation_Line	Reservation_Line
Продажи и Invoice_Line	Sales_Line

Контекст Reservation_Line отобразится ниже.



Контекст Sales_Line отобразится ниже.



Можно создавать различные наборы объектов из таблиц в различных контекстах. Пользователи могут выполнять запросы "Предварительные заказы" или "Продажи" в зависимости от выбранных объектов.

6.5.3 Визуальное определение циклов

Можно использовать следующие рекомендации для анализа схемы, чтобы определить, что подходит для устранения циклов: псевдоним или контекст. Данные сведения могут быть полезны для понимания схемы. Но необходимо использовать параметры "Определить псевдонимы" и "Определить контексты", чтобы определить и устранить циклы. Дополнительные сведения см. в разделах [Определение и создание псевдонима \[страница 226\]](#) и [Определение и создание контекста \[страница 228\]](#).

Если в цикле содержится...	тогда цикл может быть устранен с помощью...
Только одна таблица соответствия	Псевдоним
Таблица соответствия, которая получает только концы "один" объединений.	Псевдоним
Две или несколько реальных таблицы	Контекст

6.5.4 Автоматическое определение и устранение циклов

Можно использовать средство создания юниверсов, чтобы автоматически определять циклы и предлагать вероятные псевдонимы и контексты, которые можно вставлять в схему для устранения циклов.

6.5.4.1 Мощности множеств должны быть установлены до определения циклов

Перед использованием функций автоматического определения и устранения цикла все мощности множеств должны быть установлены для объединений в схеме.

Определение или применение мощностей множеств вручную является хорошей практикой по разработке, которую предлагает средство создания юниверсов с помощью автоматизации.

Можно установить мощности множеств двумя способами.

- Вручную. Дополнительные сведения см. в разделе [Использование мощности множеств \[страница 180\]](#).
- Используйте параметр "Определить мощности множеств". Дополнительные сведения см. в разделе [Использование мощности множеств \[страница 180\]](#).

6.5.5 Функции средства проектирования для определения и устранения циклов

Для определения и устранения циклов можно использовать следующие функции средства создания юниверсов:

Определяйте и устраняйте циклы с помощью...	Описание
Определить псевдонимы	<p>Определяет таблицы с псевдонимами, чтобы устранять циклы в структуре и предлагать вероятные псевдонимы для каждой таблицы. Можно вставлять и переименовывать псевдонимы в поле.</p> <p>Необходимо выполнить запрос "Определить псевдонимы" перед "Определить контексты", чтобы убедиться, что созданные псевдонимы включены в подтвержденные контексты.</p> <p>Это не определяет необходимость для псевдонима в устранении ловушки раскрытия.</p>

Определяйте и устраняйте циклы с помощью...	Описание
Определить контексты	<p>Определяет контексты, которые могут быть использованы для устранения цикла в структуре и предлагать вероятные контексты. Можно подтвердить или переименовать каждый контекст в поле.</p> <p>Выполните запрос "Определить контексты" после "Определить псевдонимы", чтобы убедиться, что подтвержденные контексты содержат новые псевдонимы.</p> <p>Это не всегда определяет необходимость для контекста в устранении ловушки разногласия. Если нет, то необходимо определить контекст вручную.</p>
Определить циклы	<p>Определяет и выделяет циклы в структуре, а также вставляет псевдоним или контекст для устранения цикла. В поле "Определить циклы" можно подтвердить псевдоним или контекст.</p> <p>Используйте данное поле, чтобы выполнить быструю проверку в схеме, или визуализировать цикл. Не используйте его для определения и устранения циклов, так как невозможно редактировать или увидеть вероятные псевдонимы перед вставкой.</p>

6.5.5.1 Основной способ определения и устранения циклов

Процедура определения и устранения циклов представлена ниже. В разделах подробно описаны шаги данного процесса.

1. Проверьте, все ли мощности множеств установлены.
См. раздел [Использование мощности множеств \[страница 180\]](#).
2. Выполните запрос "Определить псевдонимы", чтобы определить, необходим ли схеме псевдоним для устранения циклов.
Дополнительные сведения см. в разделе [Определение и создание псевдонима \[страница 226\]](#).
3. Вставьте вероятные псевдонимы, предложенные в запросе "Определить псевдонимы".
4. Выполните запрос "Определить контексты", чтобы определить, необходим ли схеме контекст для устранения цикла, который невозможно устранить только с помощью псевдонима.
Дополнительные сведения см. в разделе [Определение и создание контекста \[страница 228\]](#).
5. Подтвердите вероятные контексты, предложенные в запросе "Определить контексты"
6. Протестируйте устраненный цикл с помощью создания объектов и выполнения запросов.
Сведения о создании объектов и тестировании структур юниверса см. в главе [Создание юниверсов \[страница 263\]](#).

❗ Примечание

Устранив циклы в схеме, в таблицах которой уже определены объекты, необходимо определить объекты снова с помощью псевдонима, но без основной таблицы.

6.5.5.2 Определение и создание псевдонима

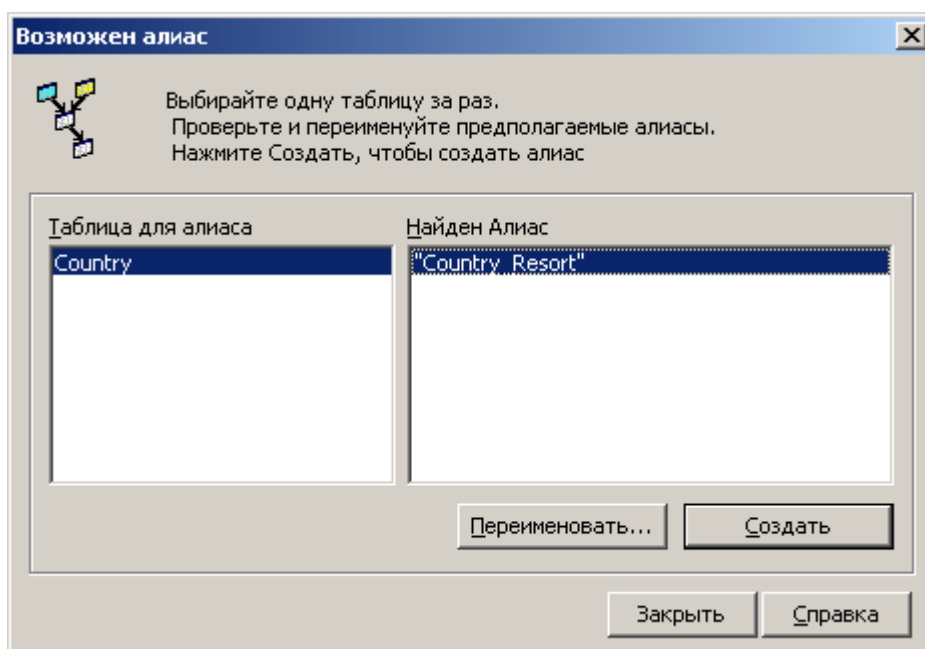
Можно использовать запрос "Определить псевдоним" для автоматического определения и указания таблиц, вызывающих циклы в активном юниверсе. Запрос "Определить псевдонимы" предлагает вероятные таблицы для редактирования и вставки в схему.

❗ Примечание

Перед использованием запроса "Определить псевдонимы", проверьте, все ли таблицы в схеме связаны объединениями, а мощности множеств установлены.

Чтобы определить и создать псевдоним,

1. Выберите ► *Инструменты* ► *Автоматическое определение* ► *Найти псевдонимы* ►.
Или
Нажмите кнопку *Определить псевдонимы*.
Откроется диалоговое окно *Вероятные псевдонимы*. На левой панели отобразится таблица или таблицы для псевдонима. На правой панели отобразится список предложенных псевдонимов для прерывания цикла.

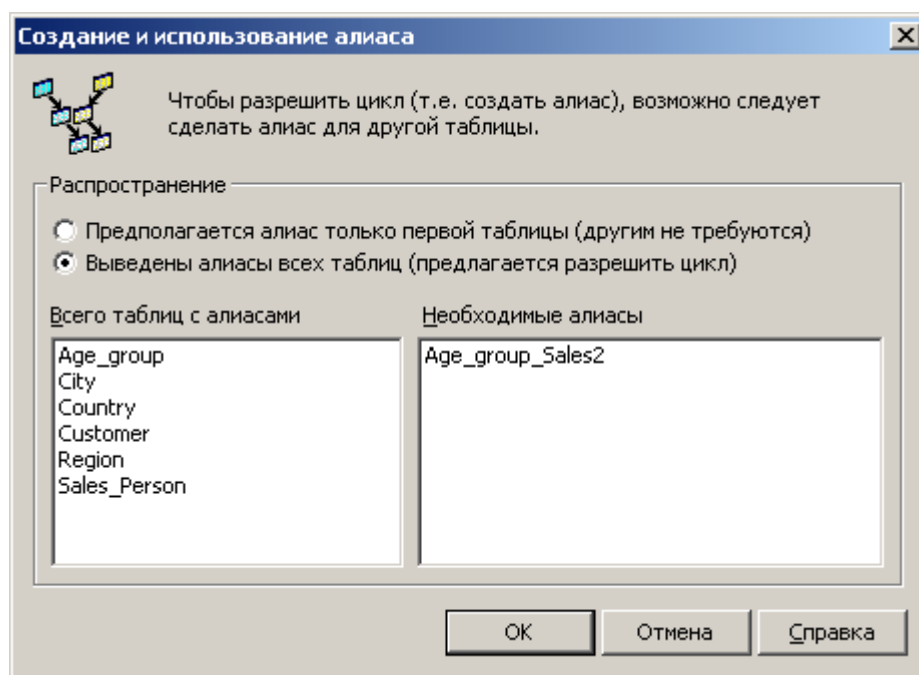


2. Выберите таблицу в левой панели.
В правой панели содержится предложенное имя для вероятного псевдонима.
3. Для переименования предлагаемых псевдонимов нажмите кнопку *Переименовать* и введите новое имя в поле *Переименовать*.

4. Нажмите кнопку [Создать](#).
В окне сообщения предлагается подтвердить создание псевдонима.
5. Нажмите [OK](#).
В панели структуры отобразится псевдоним.
6. Повторите шаги 2 и 5 для оставшихся таблиц.
7. Нажмите кнопку [Заккрыть](#).

6.5.5.3 Определение и создание нескольких псевдонимов

Иногда во время создания псевдонима необходимо создавать дополнительные псевдонимы, чтобы соответствовать новым путям объединения. С помощью запроса "Определить псевдоним" средство создания юниверсов определяет необходимость использования дополнительных псевдонимов. При нажатии кнопки "Создать", отобразится следующее диалоговое окно.



В данной ситуации доступны два параметра.

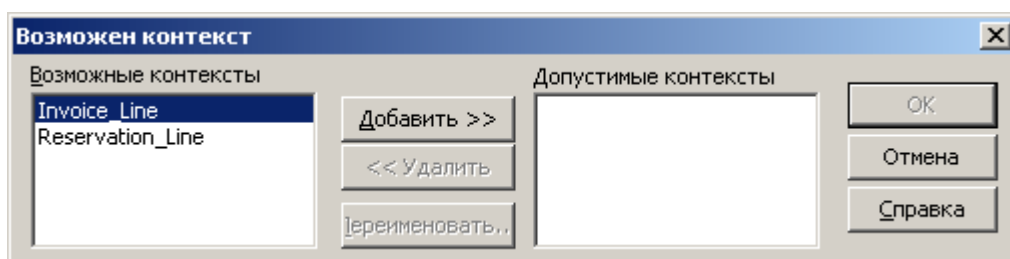
- Можно выбрать псевдоним только для предложенной таблицы.
- Можно выбрать псевдонимы для всех указанных таблиц.

6.5.5.4 Определение и создание контекста

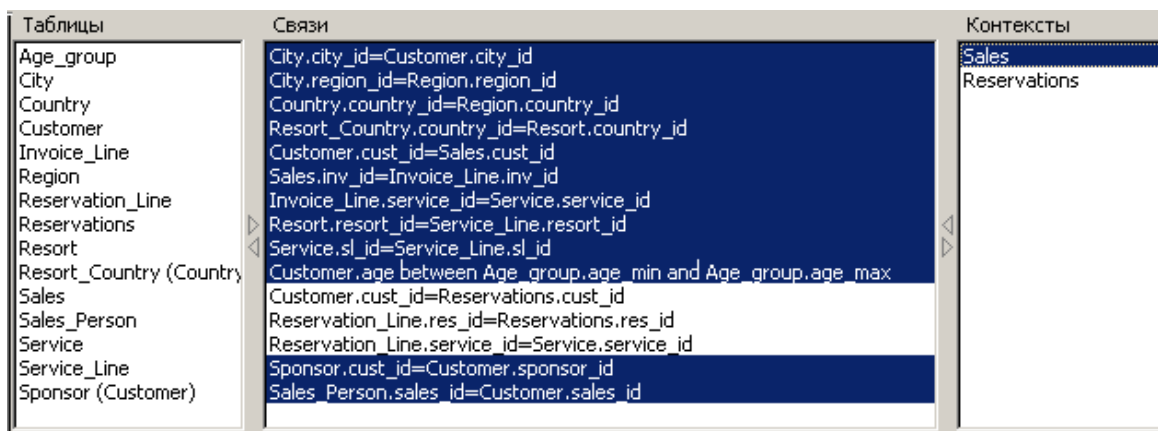
Можно использовать запрос "Определить контексты" для автоматического определения необходимости в контексте. Данный запрос предлагает вероятный контекст. Можно редактировать вероятный контекст до подтверждения.

Чтобы определить и создать контекст,

1. Нажмите "Инструменты" > "Автоматическое определение" > "Определить контексты".
Или
Нажмите кнопку "Определить контексты".
Откроется диалоговое окно "Вероятные контексты". На левой панели отобразятся вероятные контексты.



2. Выберите имя контекста.
Таблицы, включенные в вероятный контекст, выделены в схеме.
3. Нажмите кнопку "Добавить".
Имя контекста отобразится на панели допустимых контекстов. В правой панели можно удалить контекст, щелкнув его мышью и нажав кнопку "Удалить".
4. Повторите шаги 3 и 4, если необходимо, чтобы добавить другие контексты.
5. Если необходимо переименовать контекст, щелкните его в правой панели и нажмите кнопку "Переименовать".
Откроется диалоговое окно "Переименовать контекст". Введите новое имя.
6. Нажмите "ОК".
Контексты отобразятся в поле "Контексты" окна юниверса.



📌 Примечание

Если в юниверсе содержится цикл, неоднозначный для пользователя, всегда необходимо давать имя контексту, с помощью которого устраняется цикл, чтобы пользователям было легче понять. Пользователю Web Intelligence должно быть ясно, что путь к сведениям предоставляет контекст.

6.5.5.5 Автоматическое определение циклов

Можно определить циклы в юниверсе с помощью функции "Определить циклы". Данная функция автоматически определяет циклы в схеме и предлагает псевдоним или контекст для устранения цикла.

Определение циклов полезно, когда необходимо выполнить быструю проверку на наличие циклов в схеме. Данная функция также предлагает псевдонимы и контексты для устранения определенных циклов. Однако у нее меньше возможностей в управлении порядком создания псевдонимов и контекстов, чем при использовании функций "Определить псевдонимы" и "Определить контексты" для решения проблем, связанных с циклами.

Рекомендованный процесс устранения циклов описан в разделе [Основной способ определения и устранения циклов \[страница 225\]](#).

📌 Примечание

Также можно использовать функцию "Проверить целостность", чтобы автоматически находить ошибки в структурах юниверса, включая объединения, мощности множеств и циклы. Данная функция предлагает решения для любых ошибок, которые она определяет. Дополнительные сведения см. в разделе [Проверка целостности юниверса вручную \[страница 258\]](#).

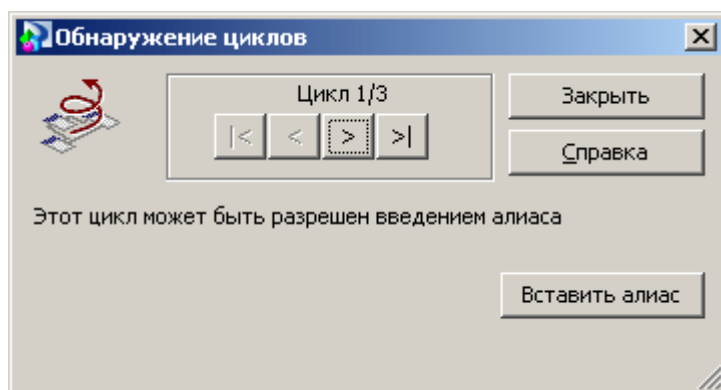
Чтобы определить циклы в схеме,

1. Убедитесь, что мощности множества установлены для всех объединений в схеме.
2. Нажмите "Инструменты" > "Автоматическое определение" > "Определить циклы".

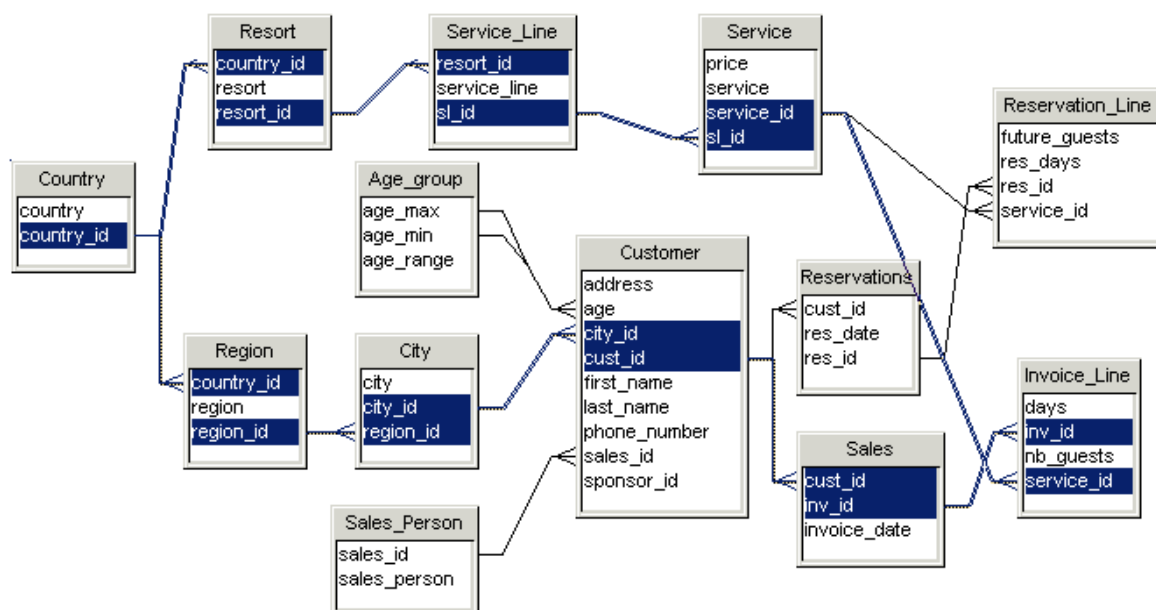
Или

Нажмите кнопку "Определить циклы".

Откроется окно поиска циклов. В нем показано количество обнаруженных циклов и предложены способы их устранения.



Определенный путь объединения, который создает цикл, одновременно выделяется на панели структуры следующим образом.



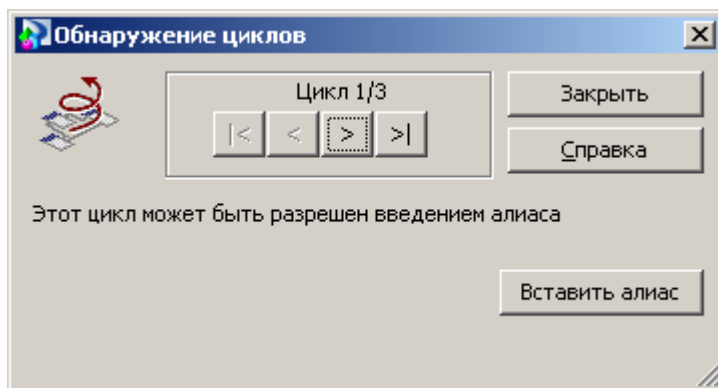
3. Чтобы перейти к следующему циклу и предлагаемому решению, нажмите кнопку перехода вперед. Для каждого цикла, обнаруживаемого средством создания юниверсов, путь объединения выделяется на панели структуры.
4. Нажмите "Закреть".

6.5.5.6 Автоматическое создание псевдонимов и контекстов

Средство создания юниверсов предлагает возможный псевдоним или контекст для устранения цикла во время выполнения запроса "Обнаружить цикл". В поле "Определить циклы" можно вставлять вероятный псевдоним или подтверждать вероятный контекст.

Чтобы создать псевдоним с помощью функции "Определить цикл".

1. Нажмите "Инструменты" > "Автоматическое определение" > "Определить циклы".
Откроется диалоговое окно "Определить циклы". В нем отобразятся один или несколько циклов в схеме, а для каждого цикла будет предложен вероятный псевдоним или контекст.
2. Нажимайте кнопку перехода вперед, пока для определенного цикла не отобразится следующее сообщение.
Не удалось устранить данный цикл с помощью псевдонима.

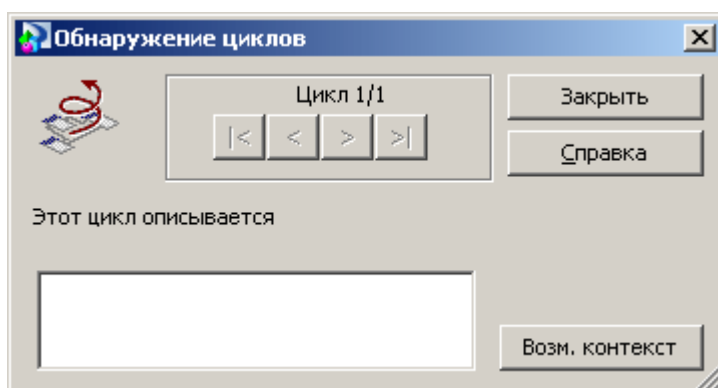


3. Нажмите кнопку "Вставить псевдоним".
Псевдоним автоматически вставится на панели структуры. Он объединится с таблицей, которая вызвала цикл в схеме.

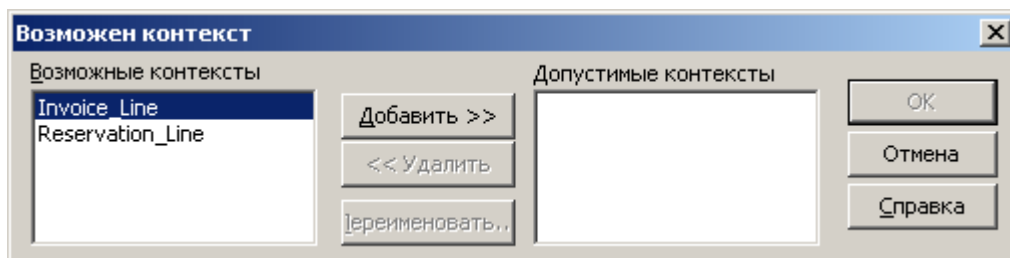
6.5.5.7 Создание контекста с помощью функции "Определить цикл"

Чтобы создать контекст с помощью функции "Определить цикл":

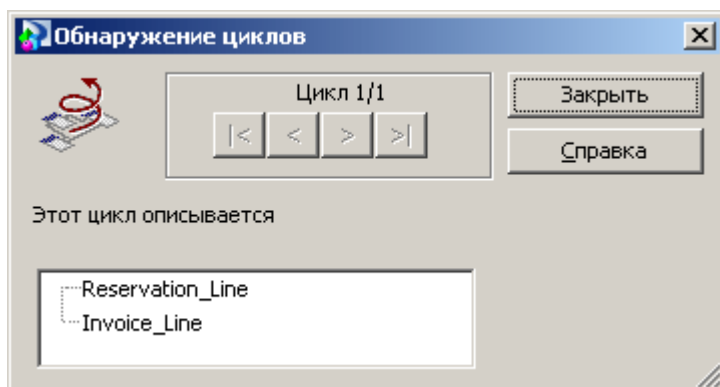
1. Нажмите "Инструменты" > "Автоматическое определение" > "Определить циклы".
Откроется диалоговое окно "Определить циклы". В нем отобразятся один или несколько циклов в схеме, а для каждого цикла будет предложен вероятный псевдоним или контекст.
2. Нажимайте кнопку перехода вперед, пока для определенного цикла не отобразится следующее сообщение.
Данный цикл не описан ни в одном контексте.



3. Нажмите кнопку "Вероятный контекст".
Откроется диалоговое окно "Вероятные контексты".



4. Выберите имя контекста.
Таблицы, включенные в вероятный контекст, выделены в схеме.
5. Нажмите кнопку "Добавить".
Имя контекста отобразится на панели допустимых контекстов. В правой панели можно удалить контекст, щелкнув его мышью и нажав кнопку "Удалить".
6. Повторите шаги 3 и 4, если необходимо, чтобы добавить другие контексты.
7. Нажмите кнопку "ОК".
Откроется диалоговое окно подтверждения контекста.



8. Нажмите "Закреть".
Контексты отобразятся в поле "Контексты" окна юниверса.

6.5.6 Примеры устранения циклов

В следующих примерах показано, как выполнять следующие действия.

- Создание псевдонима для прерывания цикла, вызванного общедоступными таблицами соответствия
- Создание псевдонима для прерывания цикла, вызванного общедоступными таблицами соответствия
- Определение несоответствия псевдонима прерыванию цикла
- Создание контекста для устранения цикла
- Совместное использование псевдонима и контекста для устранения цикла

Данные схемы не основываются на юниверсе "Пляж". Они используют схему на основе транспортной компании и предлагают другую перспективу примеров решения определенных циклов, показанную в данной главе с помощью юниверса "Пляж".

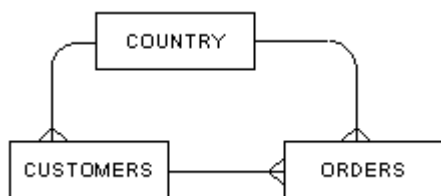
6.5.6.1 Создание псевдонима для прерывания цикла, вызванного общедоступными таблицами соответствия

В базе данных продаж содержатся сведения о продуктах, проданных клиентам по всему миру. Клиенты могут:

- Проживать в любой точке мира
- Заказывать продукты у компании
- Делать запрос о доставке продуктов в любую страну

Например, клиент, проживающий в Великобритании, может заказать автомобиль и попросить доставить его в Бразилию.

Схема данного типа базы данных выглядит следующим образом.



Данную схему можно интерпретировать следующим образом.

- Каждый клиент находится в определенной стране.
- Каждый клиент может поместить один или несколько заказов для одного продукта.
- Компания доставляет любой заказанный продукт в страну, которая является пунктом назначения, и необязательно должна совпадать со страной проживания клиента.

Таблицы и их столбцы показаны ниже.

country_id	country
1	USA
2	UK
3	France
4	Germany
5	Spain

cust_id	last_name	loc_country
100	COLTRANE	1
101	MULLIGAN	1
102	WALDRON	3
103	HANCOCK	4
104	DAVIS	2
105	BARBIERI	5
106	STREATS	5

order_id	cust_id	order_date	ship_country
12345	100	1/1/95	2
12346	101	1/6/95	1
12347	101	2/6/95	3
12348	102	8/4/95	5
12349	103	10/3/95	4
12350	104	15/8/95	2
12351	105	6/2/95	5
12352	106	7/3/95	4

Выполните запрос, чтобы получить следующие сведения.

- Имена клиентов
- Страна проживания клиента
- Даты каждого заказа
- Пункт назначения для осуществления доставки

SQL для извлечения данных выглядит следующим образом.

```
SELECT      CUSTOMERS.LAST_NAME,   COUNTRY.COUNTRY,   ORDERS.ORDER_ID,
ORDERS.ORDER_DATE,   COUNTRY.COUNTRY FROM   CUSTOMERS,   ORDERS,   COUNTRY WHERE
(CUSTOMERS.CUST_ID=ORDERS.CUST_ID) AND
(ORDERS.SHIP_COUNTRY=COUNTRY.COUNTRY_ID) AND
(CUSTOMER.LOC_COUNTRY=COUNTRY.COUNTRY_ID)
```

После извлечения SQL возвращает неполные результаты; возвращаются только те клиенты, которые заказали доставку до страны проживания. Клиенты, выбравшие другую страну для доставки, не возвращаются.

Возвращенные строки являются пересечением страны проживания клиента и пункта назначения для осуществления доставки. Вместо генерации подробных результатов, показанных ниже

last_name	country	order_id	order_date	country
COLTRANE	USA	12345	1/1/95	UK
MULLIGAN	USA	12346	1/6/95	USA
MULLIGAN	USA	12347	2/6/95	France
WALDRON	France	12348	8/4/95	Spain
HANCOCK	Germany	12349	10/3/95	Germany
DAVIS	UK	12350	15/8/95	UK
BARBIERI	Spain	12351	6/2/95	Spain
STREATS	Spain	12352	7/3/95	Germany

SQL возвращает только данные результаты.

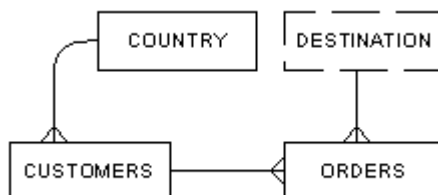
last_name	country	order_id	order_date	country
MULLIGAN	USA	12346	1/6/95	USA
HANCOCK	Germany	12349	10/3/95	Germany
DAVIS	UK	12350	15/8/95	UK
BARBIERI	Spain	12351	6/2/95	Spain

Можно прервать цикл с помощью вставки псевдонима. Первым шагом в создании псевдонима является определение таблицы соответствия, у которой есть несколько целей в структуре базы данных. Это описано в следующем разделе.

6.5.6.2 Определение многоцелевых таблиц соответствия

Таблица "СТРАНА" используется для одновременного поиска страны проживания клиента и пункта назначения для осуществления доставки. Эта таблица называется общедоступной таблицей соответствия.

Создайте псевдоним в схеме под названием "ПУНКТ НАЗНАЧЕНИЯ".



Три исходных объединения еще существуют, но цикл был прерван с помощью псевдонима "ПУНКТ НАЗНАЧЕНИЯ", поэтому больше не существует закрытого пути объединения.

6.5.6.3 Ссылка на общедоступную таблицу соответствия и псевдоним в утверждении FROM

Теперь в утверждении From необходимо ссылаться на имя таблицы дважды: первый раз – на обычное имя, а второй раз – на псевдоним. Таким образом, к исходному имени добавляется альтернативное имя.

Конечный SQL выглядит следующим образом:

```
SELECT  CUSTOMER.NAME,  COUNTRY.NAME,  ORDERS.ORDER_DATE  DESTINATION.NAME
FROM    CUSTOMER,  ORDERS,  COUNTRY,  COUNTRY DESTINATION
```

```
WHERE (CUSTOMER.CUST_ID=ORDERS.CUST_ID) AND (ORDERS.SHIP_DEST_ID=
DESTINATION.COUNTRY_ID) AND (CUSTOMER.CUST_LOC_ID=COUNTRY.COUNTRY_ID)
```

6.5.6.4 Создание псевдонима для прерывания цикла, вызванного общедоступными таблицами соответствия

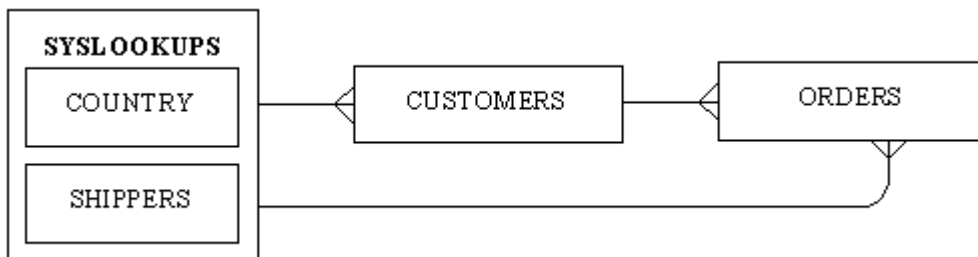
В базе данных продаж содержатся сведения о клиентах из разных стран. Данные клиенты могут заказывать товары, которые доставляются курьерами или транспортными компаниями.

В базе данных названия стран и грузоотправителей упорядочены в таблицах соответствия. Упорядочивание – это процесс, который совершенствует отношения таблиц за счет удаления избыточностей.

Для структурных причин с помощью кода, описания и типового поля была создана только одна, а не две таблицы соответствия (SYSLOOKUPS). Типовое поле означает определенный тип сведений, которые содержатся в записи: например страна или грузоотправитель.

См. также "гибкий поиск". Данный тип таблицы часто отображается в схеме автоматически генерированным с помощью инструментов CASE.

Макет схемы и таблицы изображен ниже.



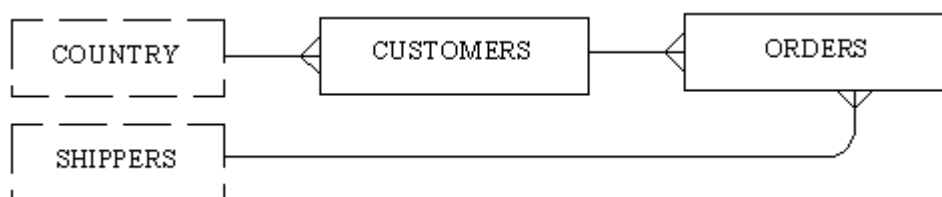
cust_id	last_name	loc_country
100	COLTRANE	1
101	MULLIGAN	1
102	WALDRON	3
103	HANCOCK	4
104	DAVIS	2
105	BARBIERI	5
106	STREATS	5

order_id	cust_id	order_date	ship_id
12345	100	1/1/95	2
12346	101	1/6/95	1
12347	101	2/6/95	3
12348	102	8/4/95	5
12349	103	10/3/95	4
12350	104	15/8/95	2
12351	105	6/2/95	5
12352	106	7/3/95	4

type	code	description
CTRY	1	USA
CTRY	2	UK
CTRY	3	France
CTRY	4	Germany
CTRY	5	Spain
SHIP	1	Man With A Van
SHIP	2	'Cut You Up' Couriers
SHIP	3	Parcel Fun
SHIP	4	Boggit & Leggit Couriers
SHIP	5	Deliveries 'R Us
SHIP	6	Sky Nut

Таблица SYSLOOKUPS служит для нескольких целей. Поэтому необходимо создать такое же количество псевдонимов, как доменов у таблицы (определенные значения для типового поля). На основе двух целей, представленных в таблице SYSLOOKUPS, можно создать два псевдонима: "СТРАНА" и "ГРУЗОТПРАВИТЕЛИ".

Окончательная схема изображена ниже.



В средстве создания юниверсов создайте объект Страна клиента, определенный как COUNTRY.DESCRPTION, и объект Грузоотправитель, определенный как SHIPPERS.DESCRPTION.

Соответствующими объединениями будут.

CUSTOMERS.LOC_COUNTRY=COUNTRY.CODE

ORDERS.SHIP_ID=SHIPPERS.CODE

Использование самоограниченных объединений для ограничения результатов

Определив объекты, необходимо ограничить каждый псевдоним, чтобы он вернул только сведения о собственном домене, а не о других. Дополнительные сведения о создании самоограниченных объединений см. в разделе [Самоограниченные объединения \[страница 176\]](#).

Например, если необходимо выяснить названия грузоотправителей, которые отправили два заказа клиенту 101, ожидайте, что вернуться две строки.

Однако следующий SQL

```
SELECT  ORDERS.ORDER_ID,    ORDERS.CUST_ID,    ORDERS.ORDER_DATE,
        SHIPPERS.DESRIPTION SHIPPER FROM    ORDERS,    SYSLOOKUPS SHIPPERS WHERE
        (ORDERS.SHIP_ID=SHIPPERS.CODE)
```

предоставит результаты, как показано ниже.

order_id	cust_id	order_date	shipper
12346	101	1/6/95	Man With A Van
12346	101	1/6/95	USA
12347	101	2/6/95	Parcel Fun
12347	101	2/6/95	France

Запрос вернул названия стран и грузоотправителей. У "США" и "Man With a Van" общий код 1, а у "Франции" и "Parcel Fun" – 3.

Можно исправить ошибки следующим образом.

- Примените новое самоограниченное объединение к псевдониму "Грузоотправители". В диалоговом окне "Редактировать объединение" назначьте таблицу 1 и таблицу 2 для "ГРУЗОТПРАВИТЕЛИ" и введите выражение SQL `SHIPPERS.TYPE= 'SHIP'`.
- Примените новое самоограниченное объединение к псевдониму "Грузоотправители". В диалоговом окне "Редактировать объединение" назначьте таблицу 1 и таблицу 2 для "СТРАНА" и введите выражение SQL `COUNTRY.TYPE= 'STRY'`.

Проблемы в использовании ограничений

Добавление ограничения к утверждению Where объекта или к существующему объединению между псевдонимом и таблицей "КЛИЕНТЫ И ЗАКАЗЫ" может вызвать следующие проблемы.

- Добавив ограничение к утверждению Where объекта, необходимо добавить такое же ограничение к каждому объекту, созданному из псевдонима. Создание ряда объектов на основе псевдонима со множеством столбцов может вызвать проблемы с подтверждением юниверса.
- Данное ограничение к объединению между псевдонимом и другой таблицей вступает в силу, если активировано объединение. При выполнении запроса только с объектом **Грузоотправитель** возвращается каждая строка псевдонима "ГРУЗОТПРАВИТЕЛИ" (включая лишние строки таблицы "Страна"), так как нет причины включать таблицу "ЗАКАЗЫ". Так как объединение не видно должным образом, ограничение не применяется.

Сводка

В данном примере указана схема с общедоступной таблицей соответствия. Итог выполненных действий выглядит следующим образом.

1. Создайте псевдонимы "СТРАНА" и "ГРУЗОТПРАВТЕЛИ" для общедоступной таблицы соответствия.
2. Создайте самоограниченные объединения для псевдонимов в качестве ограничений.

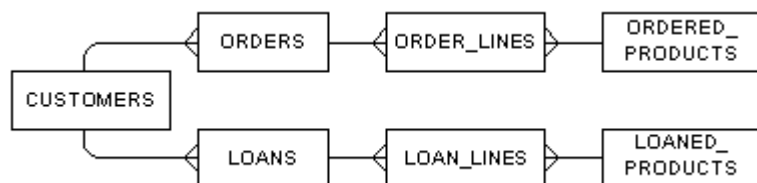
Псевдонимы в данном примере устраняют цикл с помощью одной объединенной таблицы соответствия в качестве двух различных таблиц. Для данных псевдонимов также необходима установка ограничений (самостоятельные объединения). Таким образом, в некоторых структурах псевдонимы вызывают потребность в дополнительных установках или ограничениях.

6.5.6.5 Определение несоответствия псевдонима прерыванию цикла

Создание псевдонима для устранения цикла описано выше и не является оптимальным решением. В данном случае использование контекста является лучшим решением. В следующем примере объясняется, почему псевдонимы являются неподходящими, и почему контексты являются лучшим решением в данном случае.

При попытке определить таблицу соответствия, которая используется для нескольких целей, становится неясно, какая это таблица: "ПРОДУКТЫ" или "КЛИЕНТЫ".

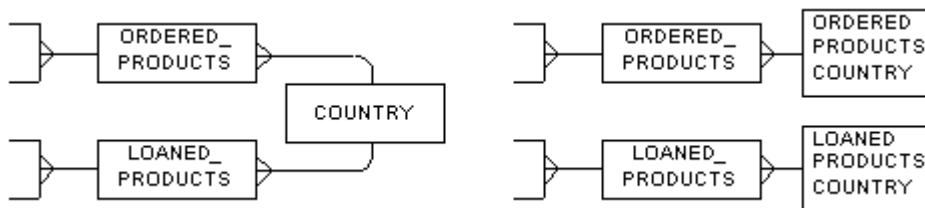
Если необходимо создать два псевдонима для таблицы "ПРОДУКТЫ", как показано ниже.



Псевдонимы будут следующие: ORDERED_PRODUCTS и LOANED_PRODUCTS. Это может привести в замешательство пользователей, так как они, вероятнее всего, освоят продукты за исключением заказанных или взятых в кредит.

Чтобы добавить таблицу "СТРАНА" для указания того, что продукты произведены в разных странах, необходимо объединить ее напрямую с таблицей "ПРОДУКТЫ".

Окончательная схема будет выглядеть следующим образом.



В схеме, указанной выше, необходимо создать два новых псевдонима, ORDERED_PRODUCTS_COUNTRY и LOANED_PRODUCTS_COUNTRY. Использование псевдонимов является явно неудовлетворительным и сложным для определенной схемы.

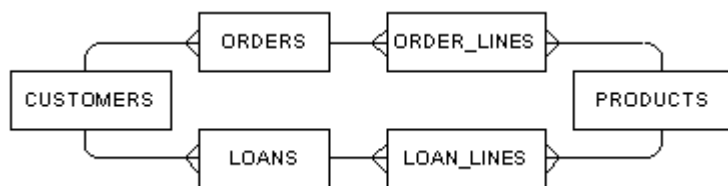
В данном случае, необходимо создавать контексты.

6.5.6.6 Создание контекста для устранения цикла

В базе данных содержатся сведения о клиентах, которые покупают или берут в кредит продукты. В базе данных есть два различных способа для представления отношений между клиентами и продуктами.

- По продуктам, заказанным клиентом (или проданным ему).
- По продуктам, взятым в кредит клиентом.

Тип схемы базы данных выглядит следующим образом.



Если необходимо выполнить запрос, который возвращает только список имен клиентов и список названий продуктов, можно использовать таблицу "ЗАКАЗ И ORDER_LINES". Результатом запроса будет являться список продуктов, заказанный каждым клиентом.

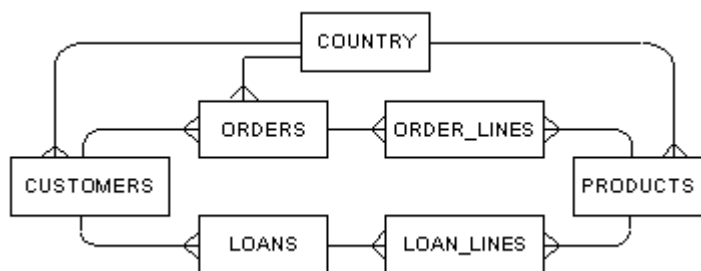
С помощью таблиц "КРЕДИТЫ" и "LOAN_LINES" можно получить список продуктов, взятых в кредит каждым пользователем.

В данной схеме содержится цикл, который вызывает запрос, включающий шесть объединений одновременно, чтобы получить единый список проданных и взятых в кредит продуктов. Если продукт был продан и никогда не брался в кредит или наоборот, он не отобразится в списке результатов.

Совместное использование псевдонима и контекста для устранения цикла

Можно использовать контексты и псевдонимы, чтобы устранить циклы в юниверсе. В следующем примере показано, как можно совместно использовать псевдонимы и контексты в устранении циклов.

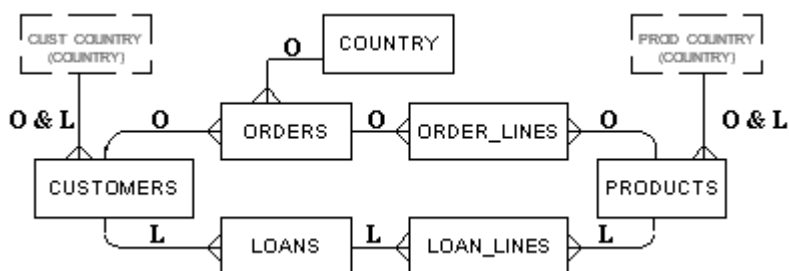
Схема юниверса выглядит следующим образом.



Можно использовать псевдонимы и контексты, чтобы устранить циклы в юниверсе следующим образом.

- Создайте два псевдонима для таблицы "СТРАНА": CUST_COUNTRY и PROD_COUNTRY
- Определите два контекста для устранения циклов от таблицы "КЛИЕНТЫ" до таблицы "ПРОДУКТЫ" (заказы и кредиты)
- Убедитесь, что два объединения "КЛИЕНТЫ и CUST_COUNTRY" и "ПРОДУКТЫ и PROD_COUNTRY" отображаются в обоих контекстах.

Окончательная схема выглядит следующим образом.



6.6 Устранение ловушек разногласия

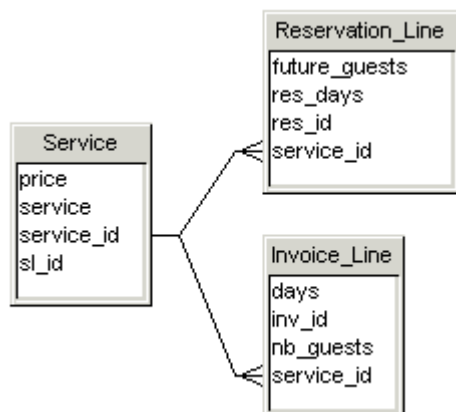
Ловушка разногласия – это общая проблема в схеме реляционной базы данных, в которой путь объединения возвращает больше данных, чем предполагалось.

6.6.1 Что такое ловушка разногласия?

Ловушка разногласия – это тип пути объединения между тремя таблицами, когда два объединения "многие к одному" сходятся в одну таблицу, а в месте, где разделяются пути объединения схождения, нет контекста.

В примере ниже отображена часть схемы юниверса "Пляж". Данные три таблицы были отделены от остальной схемы, чтобы проиллюстрировать ловушку разногласия. Она использует то же соединение

"Клуб" для данных. Таблица "Услуга" получает единственные концы двух объединений "один ко многим".



Результаты будут неверными, если только все следующие условия существуют.

- Отношение "многие-к-одному-ко-многим" существует в трех таблицах в структуре юниверса.
- В запрос включены объекты на основе двух таблиц на концах "многие" соответствующих объединений.
- Существует несколько строк, возвращенных для одного измерения.

В следующем примере показано, как запросы, выполненные, когда все вышеуказанные условия существуют, возвращают декартово произведение.

Пример: Ловушка разногласия возвышает результаты без предупреждения

С помощью схемы, указанной выше, пользователь Web Intelligence выполняет следующие отдельные запросы:

Запрос	Возвращенные результаты				
<div>Service</div> <div>Number of guests</div> <div>Service Равно 'Sports'</div>	<table> <tr> <th>Service</th><th>Number of guests</th></tr> <tr> <td>Sports</td><td>145.00</td></tr> </table>	Service	Number of guests	Sports	145.00
Service	Number of guests				
Sports	145.00				
<div>Service</div> <div>Number of future guests</div> <div>Service Равно 'Sports'</div>	<table> <tr> <th>Service</th><th>Number of future guests</th></tr> <tr> <td>Sports</td><td>8.00</td></tr> </table>	Service	Number of future guests	Sports	8.00
Service	Number of future guests				
Sports	8.00				

Пользователь выполняет запрос, в который включены арендаторы и будущие клиенты.

Service	Number of guests	Number of future guests
Service Равно 'Sports'		

Отобразятся следующие результаты:

Service	Number of guests	Number of future guests
Sports	188.00	96.00

Значительно возросло количество клиентов, которые уже пользовались услугой "Спорт", и будущих клиентов, которые заказали ее. Декартово произведение возвращено. Результаты неверные. Если их не определить, может возникнуть ряд проблем. Вышеуказанный пример заставит менеджера островных курортов задуматься, что спортивная деятельность на курортах – это более привлекательная услуга для клиентов, чем показывают реальные цифры.

6.6.2 Как ловушка разногласия завышает результаты?

Ловушка разногласия вынуждает запрос возвращать каждую возможную совокупность строк для одной меры с каждой возможной совокупностью строк для другой меры. Это произошло в примере, указанном выше.

- Транзакции "Количество клиентов" *Транзакции "Количество клиентов"
- Транзакции "Количество будущих клиентов" *Транзакции "Количество будущих клиентов"

В данном примере подробно проверяется, как ловушка разногласия возвращает декартово произведение.

Пример: Проверка декартового произведения ловушки разногласия

Необходимо проверить строки, возвращенные с помощью запросов для получения агрегированных цифр. Это можно осуществить с помощью добавления измерений "Дни выставления счета" и "Дни заказов" к запросам, чтобы вернуть сведения об автономной транзакции.

Отчет "Количество клиентов" отображается следующим образом.

Service	Days billed	Number of guests
Sports	3.00	4.00
Sports	4.00	133.00
Sports	6.00	8.00

Отчет "Количество будущих клиентов" отображается следующим образом.

Service	Days reserved	Number of future guests
Sports	1.00	7.00
Sports	2.00	1.00

В двух отчетах отображено следующее количество транзакций.

- Количество клиентов = 3 транзакции
- Количество будущих клиентов = 2 транзакции

Если в запрос добавляются два измерения, возвращаются следующие результаты.

Service	Days billed	Number of guests	Days reserved	Number of future guests
Sports	3.00	4.00	1.00	3.00
Sports	3.00	4.00	2.00	1.00
Sports	4.00	129.00	1.00	75.00
Sports	4.00	35.00	2.00	9.00
Sports	6.00	8.00	1.00	6.00
Sports	6.00	8.00	2.00	2.00
	Сумма:	188.00	Сумма:	96.00

Запрос возвращает каждую возможную совокупность строк "Количество клиентов" с каждой возможной совокупностью строк "Количество будущих клиентов": транзакции "Количество клиентов" каждая отображается дважды, а транзакции "Количество будущих клиентов" – трижды.

Если сумма составлена на основе возвращенных данных, суммарные результаты являются неверными.

В отличие от циклов, ловушки разногласия не удаляются автоматически с помощью средства создания юниверсов. Однако можно использовать функцию "Определить контексты" ("Инструменты" > "Определить контексты"), чтобы автоматически определить и предложить возможные контексты в схеме.

Функция "Определить контексты" проверяет объединения "многие к одному" в схеме. Она создает таблицу, которая получает сходящиеся объединения "многие к одному", и предлагает контексты для разделения запросов, выполненных на основе таблицы. Это самый эффективный способ убедиться, что в схеме нет ловушек разногласия.

Также можно определять ловушки разногласия графически с помощью анализа путей объединений "один ко многим" в схеме.

Если не выполнить запрос "Определить контексты" и не определить ловушку разногласия в схеме, единственным способом увидеть проблему является просмотреть строки сведений. Иначе не будет другого способа предупредить ситуацию.

6.6.3 Определение ловушки разногласия

Определить ловушку разногласия можно с помощью функции "Определить контексты", чтобы определить и предложить вероятные контексты, а потом проверить таблицу, куда отклоняются два контекста. Точка, в которой пересекаются два контекста, является источником ловушки разногласия.

Если две реальные таблицы с объединениями "многие к одному" сходятся в одной таблице соответствия – это потенциальная ловушка разногласия.

→ Совет

Сведения об упорядочивании схемы таблицы для определения проблем объединения см. в разделе [Графическая демонстрация выявления проблем объединения \[страница 254\]](#).

6.6.4 Устранение ловушки разногласия

Чтобы устранить ловушку безопасности, необходимо выполнить два отдельных запроса и объединить результаты. В зависимости от типа объектов, определенных для реальных таблиц, и типа среды конечного пользователя можно использовать следующие способы для устранения ловушек разногласия.

- Создать контекст для каждой реальной таблицы. Данное решение работает во всех случаях.
- Изменить параметры SQL для юниверса, чтобы генерировать отдельные запросы SQL для каждого показателя. Данное решение работает для объектов показателей. Оно не генерирует отдельные запросы для измерения или объектов-описаний.

Все способы описаны в следующих разделах.

6.6.4.1 Использование контекстов для устранения ловушек разногласия

Можно определить контекст для каждой таблицы на конце "многие" объединений. В данном примере можно определить контекст из таблицы "УСЛУГА" в "RESERVATION_LINE" и из таблицы "УСЛУГА" в "INVOICE_LINE".

Выполнение запроса, в который включены объекты из двух контекстов, создает инструкции Select, которые синхронизированы для получения двух отдельных таблиц в Web Intelligence, избегая создания декартова произведения.

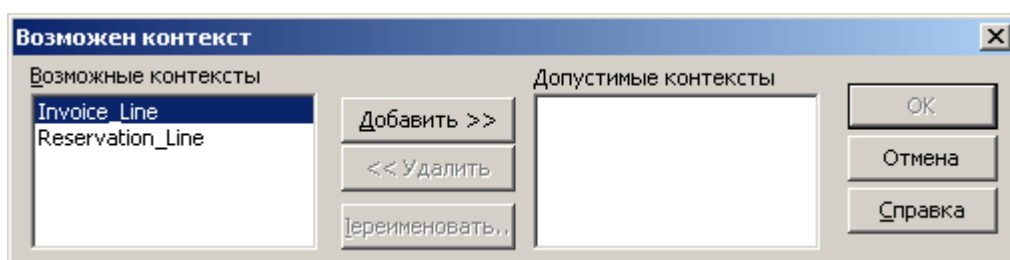
6.6.4.2 Когда использовать контексты?

Создание контекстов всегда устраняет ловушку разногласия в юниверсе. При наличии объектов измерения в одной или нескольких реальных таблицах всегда необходимо использовать контекст.

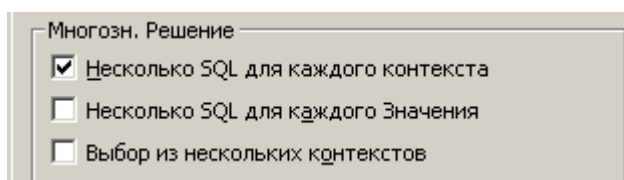
6.6.4.3 Использование контекстов для устранения ловушки разногласия

Чтобы использовать контексты для устранения ловушки разногласия,

1. Определите потенциальную ловушку разногласия с помощью анализа отношений пути объединения "один-ко-многим-ко-одному" в схеме.
2. Выберите "Инструменты" > "Определить контексты".
Откроется окно вероятных контекстов.



3. В окне списка "Вероятные контексты" выберите предложенный контекст и нажмите кнопку "Добавить", чтобы добавить его в список "Допустимые контексты".
4. Повторите данное действие для других контекстов.
Новые контексты отобразятся на панели "Контексты" панели представления списков.
5. Выберите "Файл" > "Параметры".
Откроется диалоговое окно "Параметры юниверса".
6. Выберите вкладку "SQL".
Откроется страница SQL.
7. Установите флажки на несколько SQL-инструкций для каждого контекста.



8. Нажмите кнопку "ОК".
При выполнении запросов на основе таблиц в ловушках разногласия запрос разделяется для мер и измерений, определенных на основе задействованных таблиц.

6.6.4.4 Использование нескольких SQL-инструкций для каждой меры

Если имеются только объекты мер, определенные для двух реальных таблиц, можно использовать параметр "Несколько SQL-инструкций для каждой меры" в диалоговом окне "Параметры юниверса". Это вызывает процесс генерации отдельных запросов SQL для каждой меры, которая отображается на панели запросов.

Данное решение не работает для измерения и объектов-описаний.

В следующей таблице описано, когда можно использовать несколько SQL-инструкций для каждой меры и когда следует избегать данного использования.

Используйте...	В данных ситуациях...
Несколько SQL-инструкций для каждой меры	В юниверсах, где содержатся только объекты мер, определенных для двух реальных таблиц. Преимуществом в использовании нескольких SQL-инструкций является то, что можно избежать использования контекстов, которые необходимо подтвердить позже.
Не используйте несколько SQL-инструкций для каждой меры	<p>Если измерение или объекты-описания определены для одной или двух реальных таблиц. Если измерение или объект-описание включены в запрос на основе юниверса с помощью данного решения, декартово произведение будет возвращено.</p> <p>Так как данное решение может замедлить время ответа на запрос и предоставить неверные результаты, необходимо рассматривать создание контекста в качестве процесса устранения ловушки разногласия.</p>

Чтобы активировать несколько SQL-инструкций для каждой меры,

1. На панели меню выберите "Файл" > "Параметры".
Отобразится диалоговое окно параметров юниверса.
2. Выберите вкладку "SQL".
3. В диалоговом окне "Множественные пути" установите флажки на несколько SQL-инструкций для каждой меры.
4. Нажмите кнопку "OK".

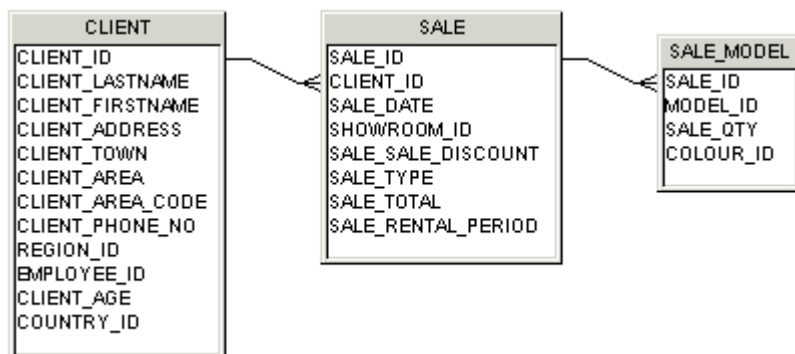
6.7 Устранение ловушки раскрытия

Ловушка раскрытия – это менее распространенная проблема, чем ловушки разногласия в схеме реляционной базы данных. Она также возвращает больше данных, чем ожидается.

6.7.1 Что такое ловушка раскрытия?

Ловушка раскрытия – это тип пути объединения трех таблиц, когда объединение "один-к-нескольким" связывает таблицы, каждая из которых в свою очередь связана другим объединением "один-к-нескольким". Эффект раскрытия объединений типа "один-к-нескольким" может вызвать вывод неверных результатов в случае, когда запрос включает в себя данные обеих таблиц.

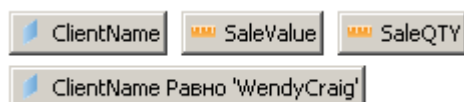
Простой пример ловушки раскрытия приведен ниже:



При введении запроса о полном количестве проданных в каждом модельном ряду машин определенной категории клиентов, будут выданы неверные результаты при применении в таблице функции агрегирования на в таблице, которая находится на одном конце объединения и связана с несколькими – на другом.

Пример: Ловушка раскрытия завышает результаты без предупреждения

Используя описанную выше схему, пользователь Web Intelligence вводит следующий запрос:



Отобразятся следующие результаты:

ClientName	SaleQTY	SaleValue
WendyCraig	2.00	57,092.00

Это верный результат. Однако, конечный пользователь добавляет идентификатор модели измерения к запросу следующим образом:



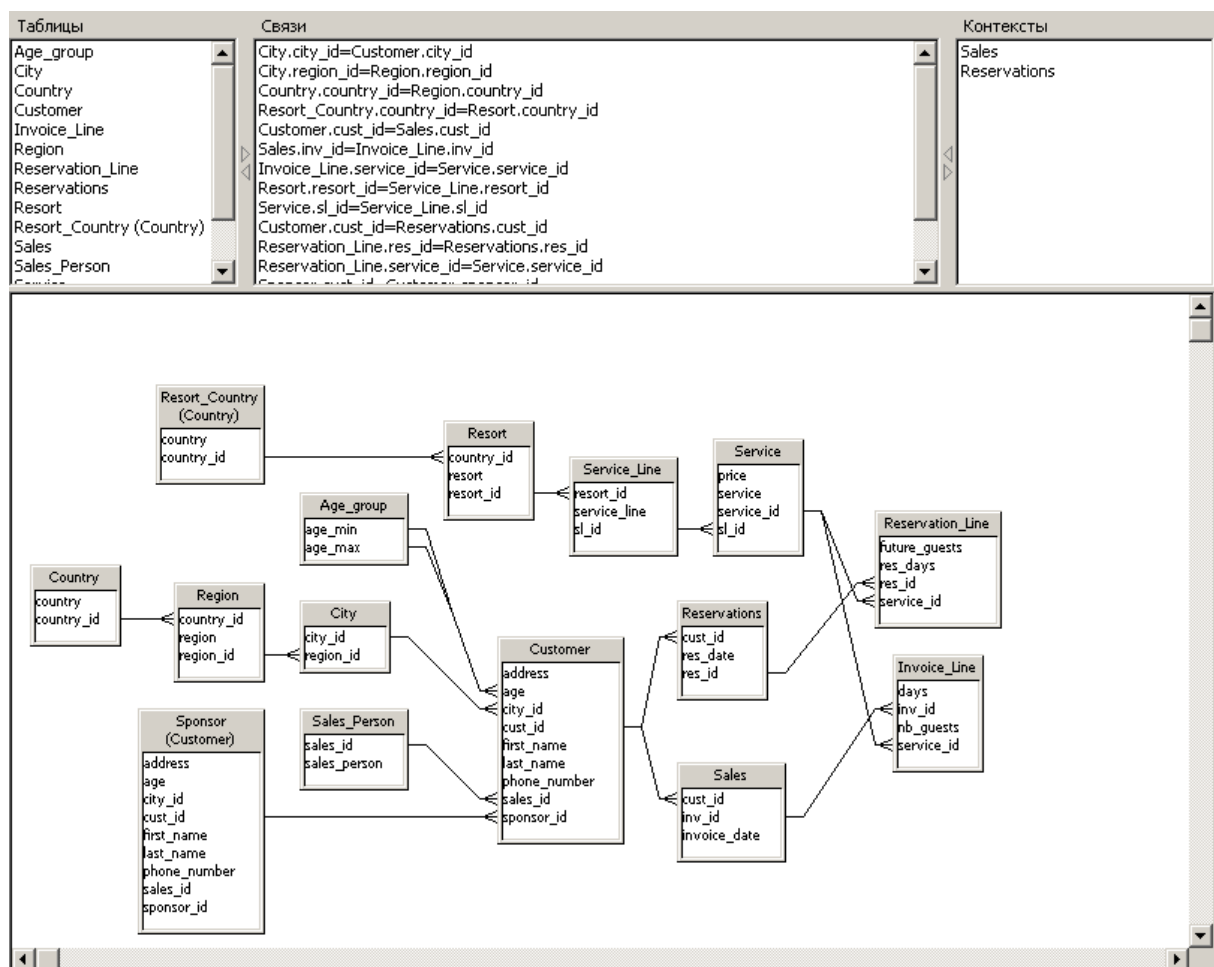
В отчете отображаются следующие результаты:

WendyCraig

Model Id	SaleValue	SaleQTY
1,034.00	57,092.00	1.00
1,081.00	57,092.00	1.00
Сумма:	114,184.00	2.00

Агрегирование общей стоимости продаж осуществляется дважды. По разу на каждый идентификатор модели Model_ID. При агрегировании данных результатов в отчете будет выведена неверная сумма. Ловушка раскрытия отобразила декартово произведение. Ольга купила две машины на общую сумму 57 092 долл. США, а не 114 184, как показано в отчета. Включение идентификатора модели Model_ID в запрос привело к агрегированию стоимости продаж. Сколько строк – столько идентификаторов модели Model_ID.

Чтобы устранить явления ловушек раскрытия при использовании объектов измерений, используются псевдонимы и контексты. Ниже приведена схема, являющаяся решением для схемы ловушки раскрытия:



Исходный запрос, отобразивший декартово произведение для Ольги Белой, теперь отображает следующую таблицу при введении запроса со следующим решением:

WendyCraig

Sale Qty	Model Id	Sale Total
1.00	1,034.00	57,092.00
1.00	1,081.00	

6.7.2 Как обнаружить ловушку раскрытия?

Ловушки раскрытия невозможно определить автоматически. Нужно самостоятельно просмотреть направление координат, отображенных в схеме таблицы.

Если имеются две таблицы, связанные объектами показателей и объединенные в серию объединений принципа от нескольких к одной, в этом случае есть возможность наличия ловушки раскрытия.

Описание построения схемы таблицы таким образом, чтобы избежать проблем при объединении см. в разделе [Графическая демонстрация выявления проблем объединения \[страница 254\]](#).

6.7.3 Устранение ловушки раскрытия

Есть два пути устранения ловушек раскрытия.

- Создать псевдоним для таблицы, содержащей исходное агрегирование, затем использовать "Обнаружение контекстов" (Инструменты > Обнаружить контексты) для выявления и предложения контекстов для всех таблицы под псевдонимом и контекстов для исходной таблицы. Это наиболее эффективный способ устранения ловушек раскрытия.
- Изменение параметров SQL для юниверса. Это работает только для объектов показателей.

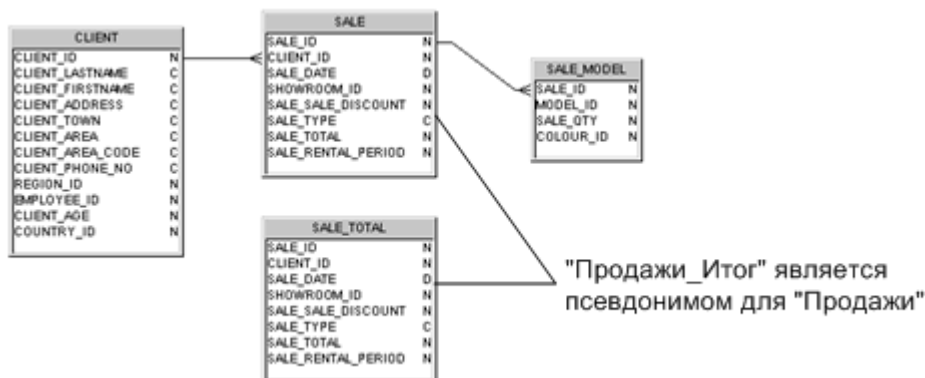
Оба метода описаны ниже.

6.7.3.1 Использование псевдонимов и контекстов для устранения ловушек раскрытия

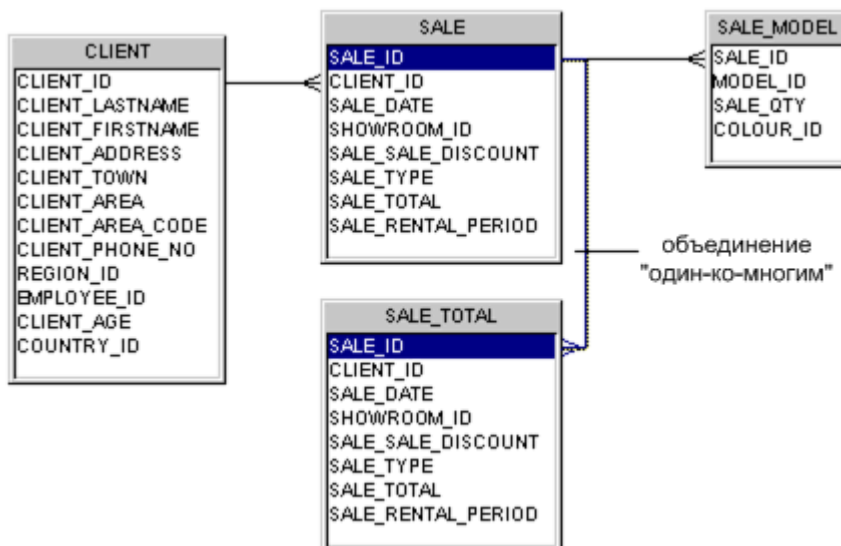
Создается таблица под псевдонимом для таблицы синтезирующей агрегирование и затем обнаруживаются контексты объединения с целью разделения запроса. Это можно сделать следующим образом:

Для использования псевдонимов и контекстов в решении проблем ловушек раскрытия сделайте следующее:

1. Определите потенциальные ловушки раскрытия путем анализа "один-к-нескольким-к-одному-к-нескольким" отношений путей объединения в схеме.
2. Создайте псевдоним для таблицы, создающей несколько агрегирований.
Например, общая стоимость продаж в предыдущем примере – это агрегирование столбца Sale_Total в таблице "Продажи". Создайте псевдоним Sale_Total для "Продажи".



3. Создайте объединение для исходной таблицы и таблицы под псевдонимом.
Если создать объединение типа "один-к-одному", средство создания юниверсов не определяет контекст – его нужно строить вручную. Во многих случаях можно использовать объединение типа "один-к-нескольким", которое позволяет автоматически определять и вводить контексты.
Например, создается объединение типа "один-к-нескольким" между "Продажи" и Sale_Total.



4. Создайте объект, оперирующий агрегированием в таблице под псевдонимом.
Например, исходный объект Общая стоимость продаж был определен следующим образом:
`sum(SALE.SALE_TOTAL)`. Новое определение для общей стоимости продаж:
`sum(Sale_Total.SALE_TOTAL)`.
5. Выберите "Инструменты" > "Определить контексты".

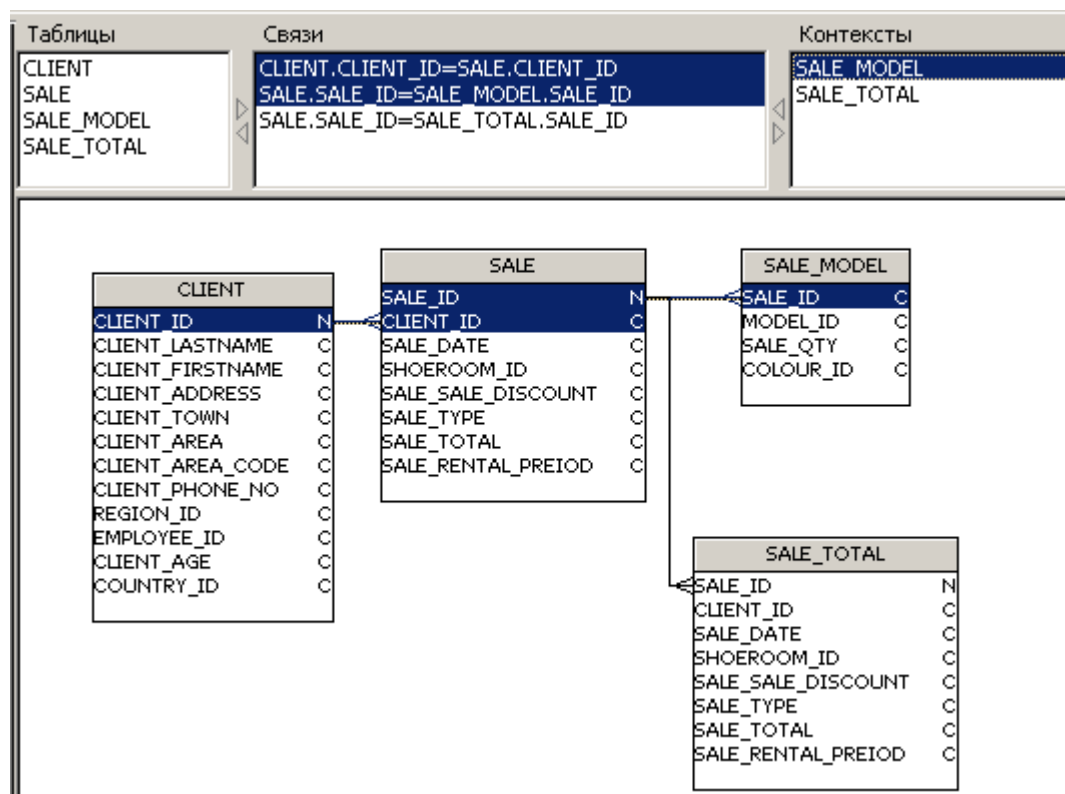
Откроется окно вероятных контекстов. В нем предлагаются вероятные контексты пути объединения для основной таблицы и новый путь объединения для таблицы под псевдонимом.

Примечание

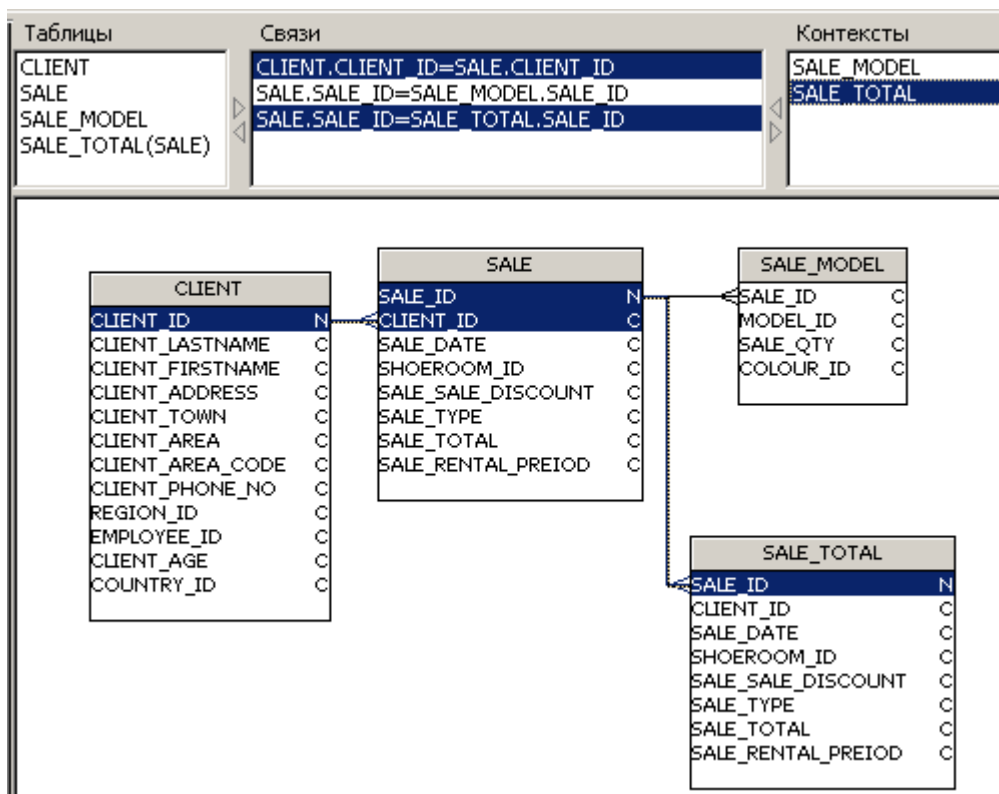
Если между основной таблицей и таблицей под псевдонимом установлено объединение типа "один-к-одному", необходимо вручную создать контекст.

6. Выберите подходящий контекст и нажмите "Добавить".
7. Повторите то же самое для других контекстов.
8. Нажмите кнопку "OK".

Контексты созданы в схеме. Их можно просмотреть на панели контекстов, если активен Режим списка (Вид > Режим списка). Контекст для пути объединения КЛИЕНТ>ПРОДАЖА>SALE_MODEL появляется следующим образом:



Добавьте второй контекст для пути объединения КЛИЕНТ>ПРОДАЖА>SALE_TOTAL:



9. Выберите "Файл" > "Параметры".
Появляется диалоговое окно "Параметры".
10. Выберите вкладку SQL.страница SQL.
Откроется страница SQL.
11. Установите флажок "Несколько SQL-инструкций" для каждого контекста.

Многозн. Решение

☒ Несколько SQL для каждого контекста

☐ Несколько SQL для каждого Значения

☐ Выбор из нескольких контекстов

12. Нажмите кнопку "OK".
13. Введите запрос для проверки решения проблемы ловушки раскрытия.

6.7.3.2 Использование нескольких SQL-инструкций для каждой меры

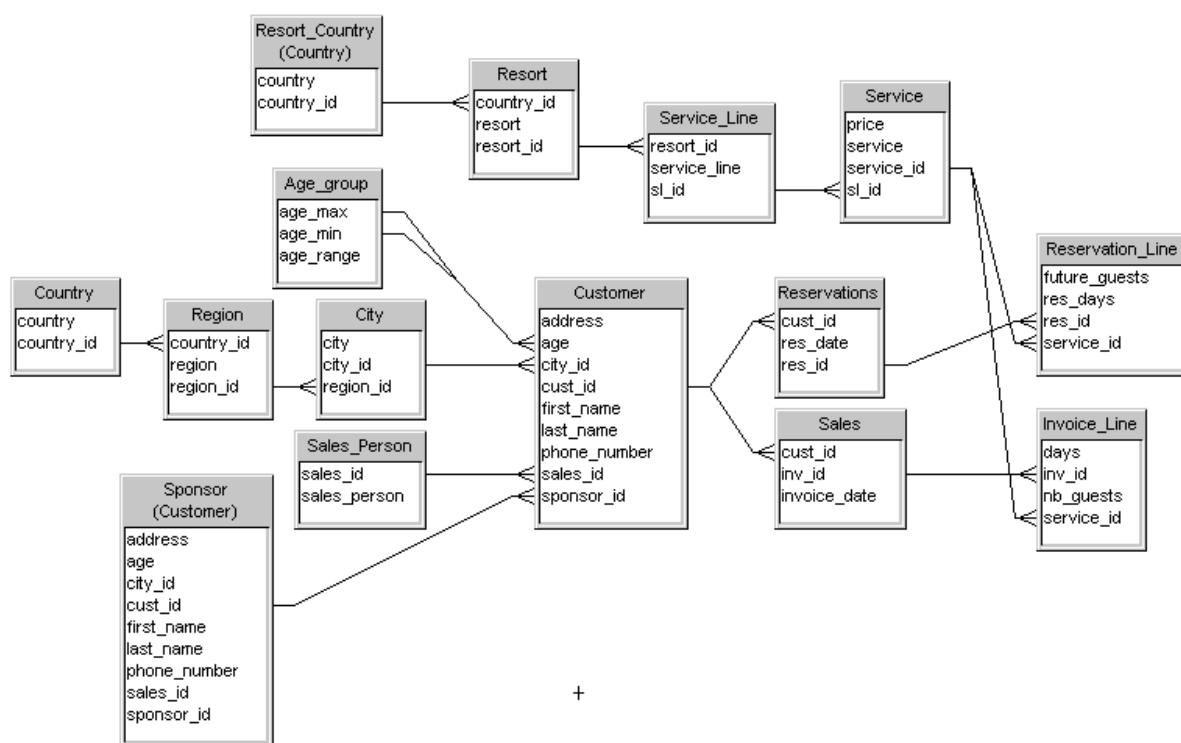
При использовании только объектов мер, определенных для обеих таблиц на многих концах серийных объединений типа "один-к-нескольким" можно использовать функцию Параметров юниверса "Несколько SQL-инструкций для каждой меры". Это вызывает процесс генерации отдельных запросов SQL для каждой меры, которая отображается на панели запросов.

Нельзя использовать данный метод для создания нескольких запросов для измерений. Если конечный пользователь может включить измерения из любой таблицы, связывающей объекты мер в запросе, в таком случае необходимо псевдоним и контекст для решения проблемы ловушки раскрытия.

Дополнительные сведения и процедуры активации данной опции содержатся в разделе [Использование нескольких SQL-инструкций для каждой меры \[страница 253\]](#).

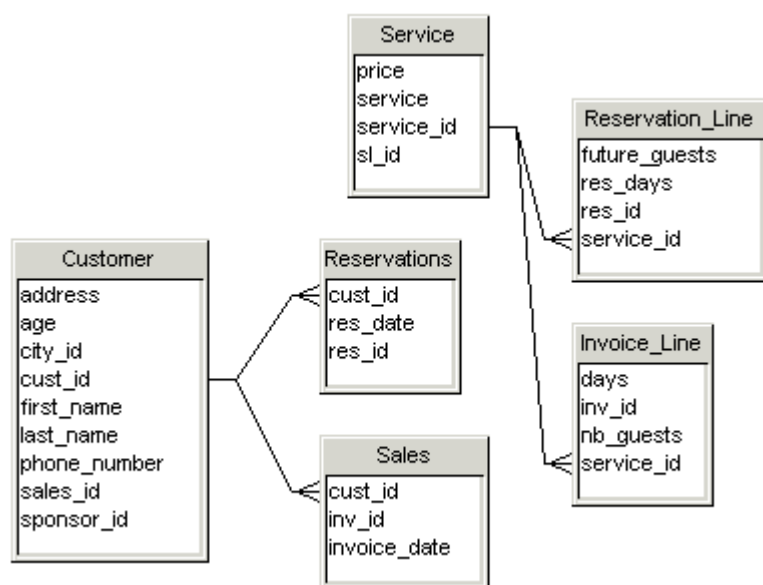
6.8 Графическая демонстрация выявления проблем объединения

Ловушки разногласия и раскрытия можно определить визуально в схеме базы данных путем упорядочивания таблиц в Панели структуры. Таким образом "несколько" концы объединений будут находиться по одну сторону панели, а концы "одной" – по другую. Пример, приведенный внизу демонстрирует схему юниверса Пляж, упорядоченную по направлению: одна – слева, многие – справа.



6.8.1 Вероятная ошибка разногласий

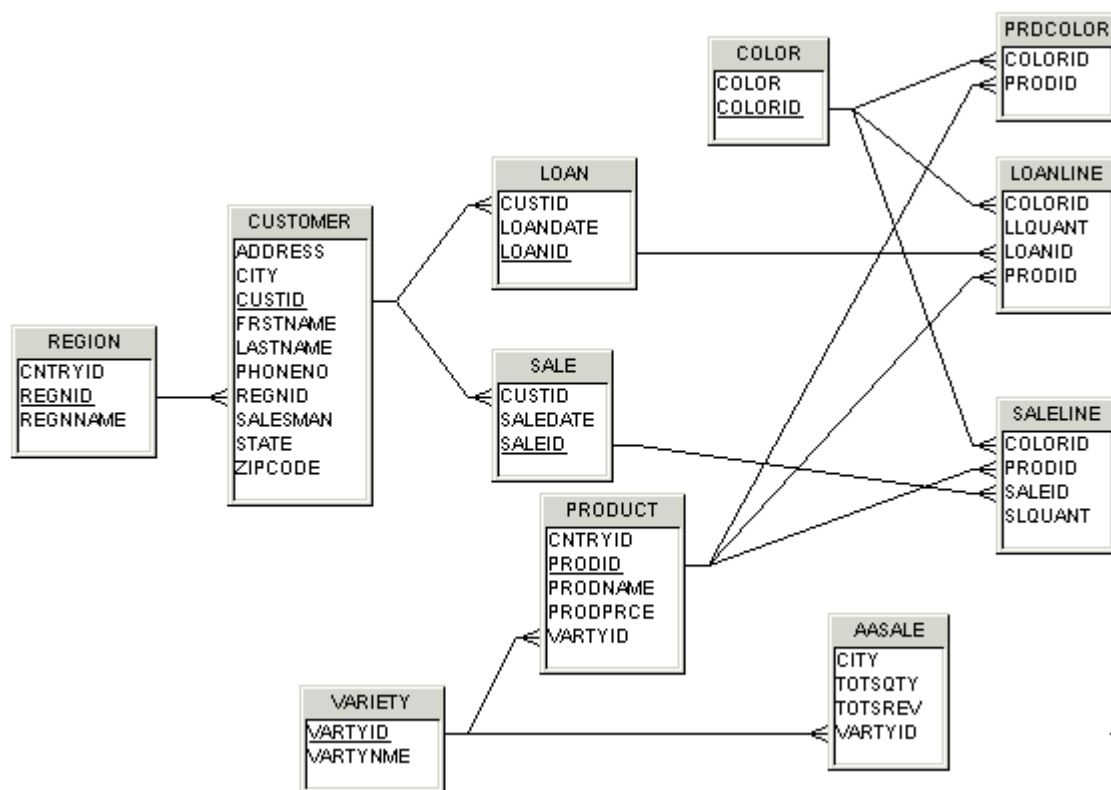
Вероятные ошибки разногласий показаны ниже:



И первый и второй путь объединения были разделены с использованием контекстов Продажи и Бронирование.

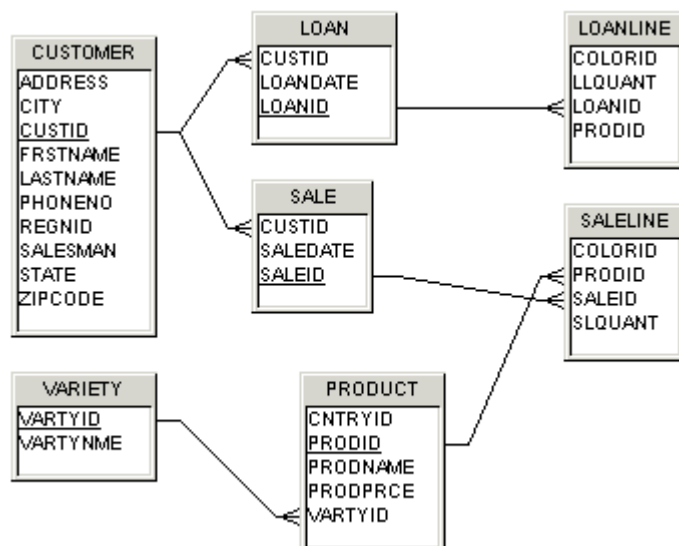
6.8.2 Вероятная ошибка раскрытия

Внизу показана схема юниверса для базы данных продажи автомобилей:



Следующие таблицы содержат вероятные ошибки раскрытия

- КЛИЕНТ, КРЕДИТ и КРЕДИТНАЯ ЛИНИЯ
- КЛИЕНТ, ПРОДАЖИ и ЛИНИЯ ПРОДАЖ
- АССОРТИМЕНТ, ПРОДУКТ и ЛИНИЯ ПРОДАЖ



→ Совет

После заполнения схемы необходимыми таблицами не преступайте к определению объектов сразу же. Поперемещайте таблицы таким образом, чтобы все объединения типа "один-к-нескольким" были расположены в одном направлении. Средство создания юниверсов – это графический инструмент, позволяющий использовать визуальные возможности для создания юниверсов. Потратьте час или чуть больше на упорядочение таблиц – это поможет в будущем сэкономить много времени при проектировании.

6.9 Проверка юниверса

Во время разработки юниверса, необходимо периодически проверять его целостность. Можно проверить целостность юниверса следующим образом.

Проверить юниверс	Описание
Автоматически	Можно установить параметры средства создания юниверсов таким образом, чтобы проверять синтаксис SQL-структур юниверса во время создания, экспорта юниверса или при его открытии.
Вручную	Запустите "Проверить целостность", чтобы проверить целостность выбранных структур юниверса.

6.9.1 Типы ошибок, определенные с помощью параметра "Проверить целостность"

Данный параметр может определить:

- Недопустимый синтаксис в определении SQL объекта или объединения.
- Циклы
- Изолированные таблицы
- Изолированные объединения
- Циклы в контекстах
- Отсутствующие и неверные мощности множеств

Как параметр "Проверить целостность" определяет изменения в подключенной базе данных?

С помощью функции "Проверить целостность" в базу данных отправляется запрос о списке таблиц. Затем список сравнивается с таблицами в юниверсе. Можно выполнить ту же операцию для столбцов.

На панели структуры с помощью данного параметра отмечаются таблицы и столбцы, несовпадающие с таблицами и столбцами в списке, так как они недоступны. В базе данных можно определять и переименовывать таблицы и столбцы. См. раздел [Проверка целостности юниверса с помощью параметра "Проверить целостность" \[страница 192\]](#).

📌 Примечание

Параметр "Проверить мощность множества" медленно обрабатывает большое количество данных. При нахождении неоднозначных или отсутствующих данных, результаты могут быть неточные. Если база данных большая и содержит записи неточных данных, необходимо выбрать параметр "Проверить мощность множества". Если использовать данный параметр, можно оптимизировать определение мощности множества с помощью изменения файла PRM. Дополнительные сведения см. в разделе [Оптимизация автоматического определения мощности множества \[страница 189\]](#).

6.9.1.1 Типы ошибок, определенные с помощью параметра "Проверить целостность"

Данный параметр может определить:

- Недопустимый синтаксис в определении SQL объекта или объединения.
- Циклы
- Изолированные таблицы
- Изолированные объединения
- Циклы в контекстах
- Отсутствующие и неверные мощности множеств

Как параметр "Проверить целостность" определяет изменения в подключенной базе данных?

С помощью функции "Проверить целостность" в базу данных отправляется запрос о списке таблиц. Затем список сравнивается с таблицами в юниверсе. Можно выполнить ту же операцию для столбцов.

На панели структуры с помощью данного параметра отмечаются таблицы и столбцы, несовпадающие с таблицами и столбцами в списке, так как они недоступны. В базе данных можно определять и переименовывать таблицы и столбцы. См. раздел [Проверка целостности юниверса с помощью параметра "Проверить целостность" \[страница 192\]](#).

❗ Примечание

Параметр "Проверить мощность множества" медленно обрабатывает большое количество данных. При нахождении неоднозначных или отсутствующих данных, результаты могут быть неточные. Если база данных большая и содержит записи неточных данных, необходимо выбрать параметр "Проверить мощность множества". Если использовать данный параметр, можно оптимизировать определение мощности множества с помощью изменения файла PRM. Дополнительные сведения см. в разделе [Оптимизация автоматического определения мощности множества \[страница 189\]](#).

6.9.2 Проверка целостности юниверса вручную

Функцию "Проверить целостность" можно использовать для проверки активного юниверса на предмет его точности и актуальности.

Данный параметр определяет:

- Ошибки в объектах, объединениях, условиях и мощностях множеств юниверса.
- Циклы в путях объединения.
- Любые необходимые контексты.
- Изменения в целевой базе данных.

Перед сравнением элементов юниверса и элементов базы данных эта функция проверяет, допустимо ли соединение с базой данных. Если соединение недопустимо, функция прекращает работу и возвращает ошибку.

6.9.2.1 Типы ошибок, определенные с помощью параметра "Проверить целостность"

Данный параметр может определить:

- Недопустимый синтаксис в определении SQL объекта или объединения.
- Циклы
- Изолированные таблицы
- Изолированные объединения

- Циклы в контекстах
- Отсутствующие и неверные мощности множеств

Как параметр "Проверить целостность" определяет изменения в подключенной базе данных?

С помощью функции "Проверить целостность" в базу данных отправляется запрос о списке таблиц. Затем список сравнивается с таблицами в юниверсе. Можно выполнить ту же операцию для столбцов.

На панели структуры с помощью данного параметра отмечаются таблицы и столбцы, несовпадающие с таблицами и столбцами в списке, так как они недоступны. В базе данных можно определять и переименовывать таблицы и столбцы. См. раздел [Проверка целостности юниверса с помощью параметра "Проверить целостность" \[страница 192\]](#).

📘 Примечание

Параметр "Проверить мощность множества" медленно обрабатывает большое количество данных. При нахождении неоднозначных или отсутствующих данных, результаты могут быть неточные. Если база данных большая и содержит записи неточных данных, необходимо выбрать параметр "Проверить мощность множества". Если использовать данный параметр, можно оптимизировать определение мощности множества с помощью изменения файла PRM. Дополнительные сведения см. в разделе [Оптимизация автоматического определения мощности множества \[страница 189\]](#).

6.9.2.2 Как параметр "Проверить целостность" определяет изменения в подключенной базе данных?

С помощью функции "Проверить целостность" в базу данных отправляется запрос о списке таблиц. Затем список сравнивается с таблицами в юниверсе. Можно выполнить ту же операцию для столбцов.

На панели структуры с помощью данного параметра отмечаются таблицы и столбцы, несовпадающие с таблицами и столбцами в списке, так как они недоступны. В базе данных можно определять и переименовывать таблицы и столбцы. См. раздел [Обновление структуры юниверса \[страница 261\]](#).

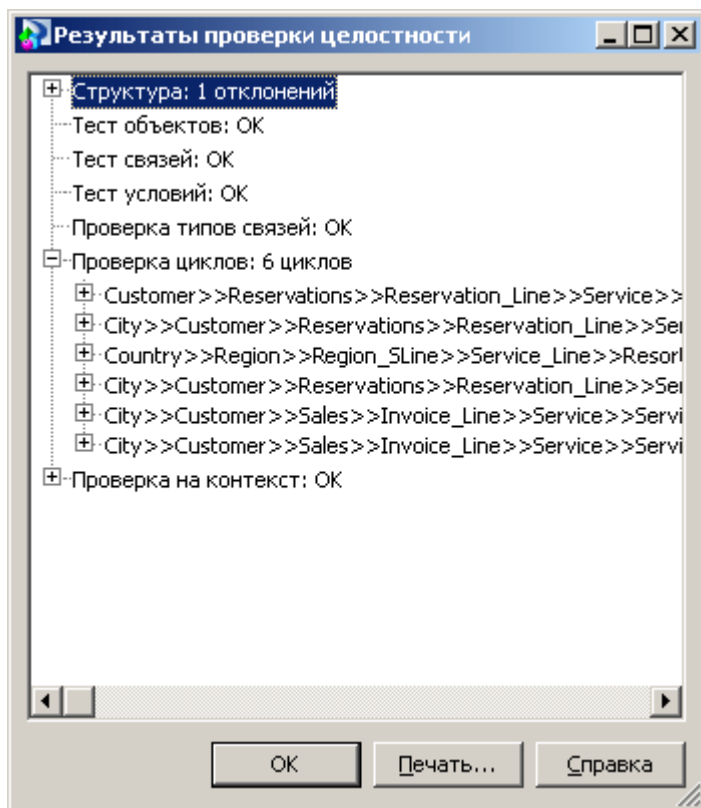
📘 Примечание

Параметр "Проверить мощность множества" медленно обрабатывает большое количество данных. При нахождении неоднозначных или отсутствующих данных, результаты могут быть неточные. Если база данных большая и содержит записи неточных данных, необходимо выбрать параметр "Проверить мощность множества". Если использовать данный параметр, можно оптимизировать определение мощности множества с помощью изменения файла PRM. Дополнительные сведения см. в разделе [Оптимизация автоматического определения мощности множества \[страница 189\]](#).

6.9.2.3 Проверка целостности юниверса с помощью параметра "Проверить целостность"

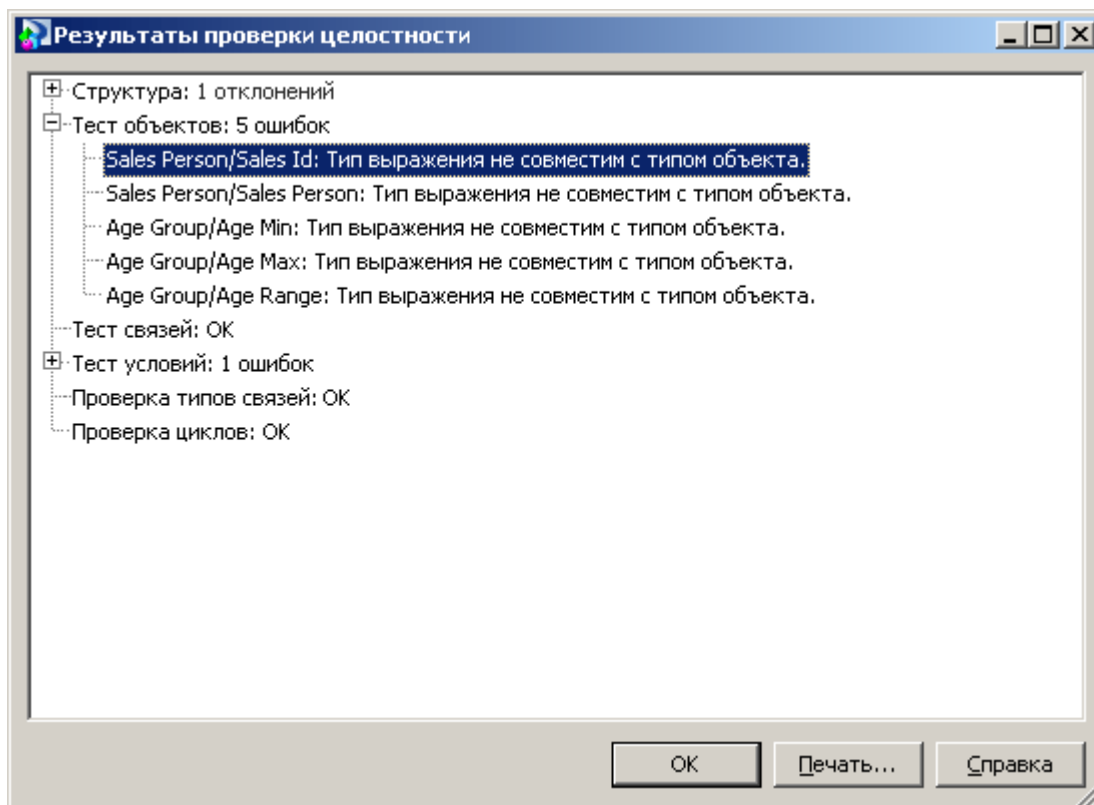
Чтобы проверить целостность юниверса,

1. Выберите "Инструменты" > "Проверить целостность".
Или
Нажмите кнопку "Проверить целостность".
Откроется диалоговое окно "Проверить целостность".
2. Установите флажки на компоненты для проверки
3. Установите флажки на компоненты для отмены проверки.
4. Установите флажок "Быстрый анализ" для проверки синтаксиса компонентов.
Или
Установите флажок "Тщательный анализ" для проверки синтаксиса и семантики компонентов.
5. Нажмите кнопку "ОК".
В окне сообщения отобразится процесс проверки юниверса.



Если параметр "Проверить целостность" не обнаружит ошибок, рядом с каждым типом ошибки отобразится "ОК".

6. Щелкните значок плюс (+) рядом с типом ошибки для просмотра списка компонентов, в которых эти ошибки произошли.



Дважды щелкните мышью элемент в списке для выделения соответствующих компонентов в панели структуры.

7. Нажмите кнопку "Печать", чтобы напечатать содержимое окна.
8. Нажмите кнопку "ОК".

📌 Примечание

Перед открытием диалогового окна "Проверить циклы" убедитесь, что мощности множества элементов объединений уже обнаружены. Иначе функция ошибочно определит циклы в объединениях.

6.9.3 Обновление структуры универса

Если параметр "Проверить целостность" определит, что база данных соединения с юниверсом была изменена, можно использовать параметр "Обновить структуру", чтобы обновить содержимое панели структуры.

Данный параметр может изменить структуру юниверса, чтобы соответствовать изменениям в базе данных следующим образом.

Если	В средстве проектирования выполняются следующие действия
Столбцы добавлены в таблицы	Добавляет столбцы в соответствующие таблицы в юни-версе.
Столбцы удалены из таблиц	Отображает сообщение с предупреждением, что столбцы и соответствующие объединения необходимо удалить.
Таблицы удалены из базы данных	Отображает сообщение с предупреждением, что та-блицы и соответствующие объединения необходимо удалить.
Таблицы переименованы в базе данных	Отображает сообщение, что соответствующие таблицы больше не распознаются в юниверсе. Необходимо пере-именовать данные таблицы, чтобы они соответствовали таблицам в базе данных. Если имена все еще не соот-ветствуют друг другу, средство создания юниверсов воз-вращает сообщение, в котором говорится, что переиме-нованные таблицы не существуют в базе данных.
Изменения в базе данных не производились	Отображает сообщение, в котором говорится о том, что обновления не требуется.

6.9.3.1 Обновление юниверса

Чтобы обновить структуру юниверса,

- Выберите "Вид" > "Обновить структуру".
Отобразится окно с сообщением об изменении в базе данных или о том, что обновления не требуется, если изменения не были произведены.

7 Создание юниверсов

После создания схемы, выполнения проверки целостности и устранения проблем циклов можно создать юниверс, который будет использоваться средствами создания отчетов.

7.1 Обзор

В данной главе описывается методика создания классов и объектов, используемых пользователями Web Intelligence для выполнения запросов и создания отчетов. Также в этой главе приведены сведения об оптимизации определений объекта, направленной на улучшение процессов генерирования отчетов конечными пользователями, а также на оптимизацию юниверсов.

В предыдущих разделах описывался процесс планирования юниверса, создание схемы таблиц, содержащей структуру базы данных юниверса (таблицы, столбцы и объединения), а также способы устранения циклов на путях объединения.

Созданная схема недоступна для просмотра пользователям Web Intelligence. После создания структуры базы данных можно создать классы и объекты, которые будут отображаться на [панели юниверса](#), и использоваться пользователями для выполнения запросов на основе структуры баз данных для создания документов и отчетов.

7.2 Введение в создание юниверса

Создание юниверса – это фаза процесса создания объекта цикла среды юниверса. Созданные объекты должны основываться изучении потребностей пользователя и использовать дизайн звуковой схемы, протестированной на предмет проблем пути объединения.

В следующем списке показано, какое место занимает фаза разработки (и проверки) в цикле разработки обычного юниверса (реализация, шаг 2):

- Подготовка
 1. Анализ потребностей пользователя
 2. Планирование
- Реализация
 1. Создание и проверка схемы
 2. Создание и проверка объектов юниверса
 3. Развертывание юниверса с помощью репозитория
- Поддержка
 1. Обновление и поддержка юниверса, основанные на изменениях требований пользователя или источника данных

7.2.1 Что такое объект?

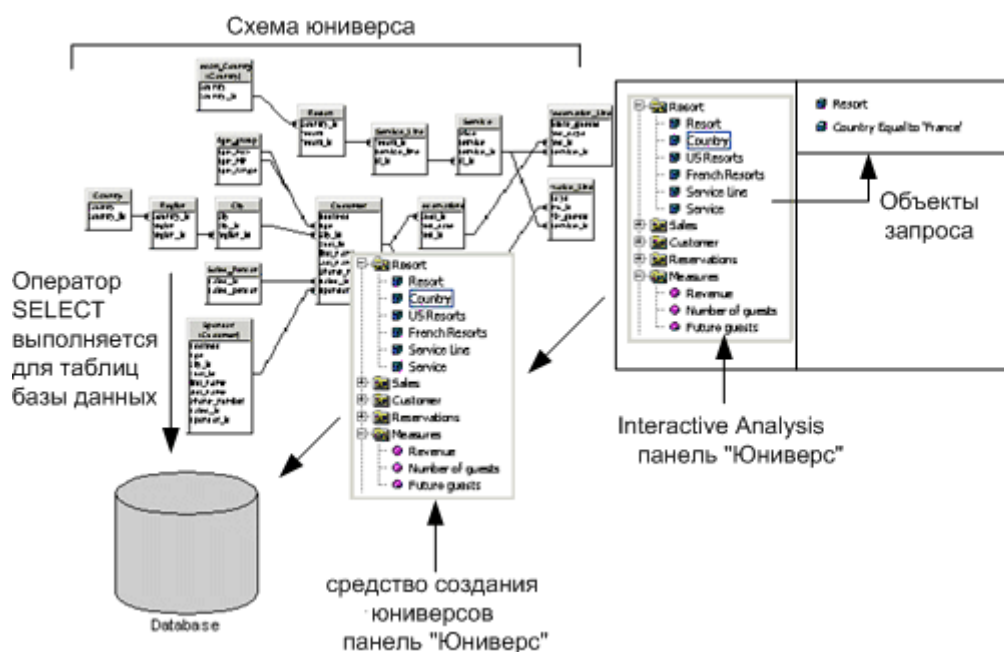
Продукты объекта Business Objects – это названный компонент в юниверсе представляющий столбцы или функции в базе данных.

Объекты отображаются в виде значка на [Панель юниверса](#). Каждый объект представляется значимой сущностью, фактом, или вычислением использованным в бизнес-среде конечного пользователя. Объекты, созданные на панели [Юниверс](#) в средстве создания юниверсов, являются объектами, которые конечные пользователи видят и используют в инструментах создания отчетов. Также можно создать объекты, которые будут использоваться только в средстве создания юниверсов. Эти объекты можно скрыть на панели [Юниверс](#), просматриваемой пользователями Web Intelligence.

Пользователи Web Intelligence перетаскивают объекты с панели [Юниверс](#) на панель [Запрос](#) для запросов и создания отчетов с помощью возвращенных данных.

Каждый объект соответствует столбцу или функции в целевой базе данных и при использовании на панели инструментов [Запрос](#) вычисляет инструкцию SELECT. Когда совмещаются несколько объектов, в базе данных выполняется инструкция SELECT, включая SQL-выражение, выведенное каждым объектом. При этом применяется утверждение по умолчанию WHERE.








На представленном далее рисунке показаны объекты в Web Intelligence на панели [Юниверс](#) и те же объекты в средстве создания юниверсов на панели [Юниверс](#). Каждый объект в средстве создания юниверсов на панели [Юниверс](#) сопоставлен со столбцом в схеме юниверса и выводит инструкцию SELECT при использовании в запросе.



Дизайнер юниверса использует средство создания юниверсов для создания объектов, которые пользователи Web Intelligence включают на панель [Запросы](#) для выполнения запросов.

7.2.2 Какие типы объектов используются в юниверсе?

Объект может быть квалифицирован как один из трех типов.

Классификация объекта	Примеры	Описание
Измерение	 Resort  Country  Service Line	Фокус анализа в запросе. Измерение сопоставляется одному или нескольким столбцам или функциям в базе данных, которые являются ключевыми для запроса.
Объект-описание	 Сведения	Содержит описательные данные об измерении. Объект-описание всегда прикрепляется к измерению. Он сопоставляется одному или нескольким столбцам или функциям в базе данных, которые содержат подробные сведения, связанные с измерением.
Мера	 revenue  Number of guests  Future guest	Содержит агрегированные функции, которые сопоставляются статистическим показателям в базе данных.

При создании объекта ему присваивается классификация на основе роли, которую он будет иметь в запросе. В зависимости от этой роли определяется инструкция Select, которая выводится объектом при его использовании на панели [Запрос](#).

7.2.3 Использование классов и объектов

Классы и объекты организуются на панели юниверсов таким образом, чтобы пользователям Web Intelligence было удобнее работать с информацией, предоставляемой объектами.

7.2.4 Что такое класс?

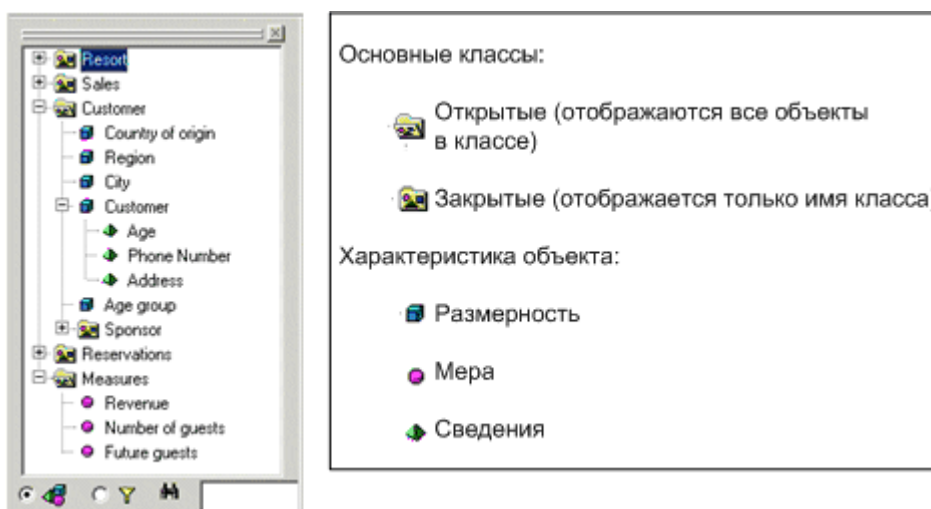
Класс представляет собой контейнер объектов. Понятие класса эквивалентно понятию папки в среде Windows. Классы создаются для размещения объектов, имеющих общее назначение в юниверсе.

7.3 Использование области "Юниверс"

Создание классов и объектов в юниверсе с помощью *области юниверса*.

Панель Юниверса представляет иерархический вид классов и объектов в активном юниверсе. Использование *Панель Юниверса* для просмотра, создания, редактирования, и упорядочивания классов и объектов

Панель Юниверса представлена ниже. Имена классов отображаются рядом с значком папки, а имена объектов рядом с их символами.



7.3.1 Отображение классов и объектов или условий

Можно использовать два переключателя расположенные в нижней части окна для отображения классов и объектов, или объектов условия на панели юниверса. Объекты условия являются predetermined утверждениями Where, которые могут использоваться в одной или нескольких инструкциях Select.

Можно отобразить два окна на панели юниверса:

Окно	Чтобы отобразить окно...	Что отображает
Классы/Объекты	Выберите левый переключатель	Все классы и объекты
Классы /Условия	Выберите правый переключатель	Все классы и условия применены к объектам, которые содержат каждый класс

Связанные сведения

[Определение ограничений для объекта \[страница 297\]](#)

7.4 Основные операции над классами, объектами и условиями

Можно выполнить следующие операции на панели *Юниверс*, которые являются основными для классов, объектов и условий:

7.4.1 Вырезать, копировать, вставить

Можно вырезать, копировать и вставить выбранный компонент с помощью обычных стандартных команд используемых в среде Windows.

7.4.2 Перемещение классов, объектов, или условий

Можно переместить компонент в другое положение в окне путем перетаскивания его в необходимое место.

7.4.3 Скрыть или отобразить классы, объекты и условия

Можно скрыть один или несколько компонентов на панели юниверса. Такие компоненты скрываются в приложении Web Intelligence, однако остаются видимыми для пользователей средства создания юниверсов.

Скрытые объекты от конечных пользователей могут быть полезны для любых из следующих причин:

- Компоненты из связанных юниверсов не являются необходимыми в активном юниверсе.
- Объекты используются только для оптимизации синтаксиса SQL и их следует скрыть от конечных пользователей.
- Идет процесс разработки компонента, который не должен быть доступен для просмотра для конечных пользователей на панели запроса *Запроса*.
- При необходимости отключить компоненты на время без удаления.

7.4.3.1 Скрыть классы, объекты или условия

Чтобы классы, объекты или условия:

1. Нажмите компонент на панели юниверса.
2. Выберите ► [Редактировать](#) ► [Скрыть элементы](#) ▾.
Или
Нажмите кнопку [Показать/Скрыть](#) на панели инструментов [Редактирование](#).
Имя компонента отобразится курсивом на панели [Юниверс](#)

7.4.3.2 Отображение скрытых классов, объектов или условий

Имя скрытых компонентов отобразится курсивом.

Чтобы скрыть класс, объект или условие:

1. Нажмите скрытый компонент на панели [Юниверс](#).
2. Выберите ► [Редактировать](#) ► [Скрыть элементы](#) ▾.
Имя компонентов отобразится не курсивом. Видимый для конечного пользователя.

7.5 Определение классов

Класс содержит один или более объектов. Каждый объект в юниверсе должен содержать класс. Используйте классы для группировки связанных объектов. Классы облегчают поиск определенных объектов для конечного пользователя. Можно создать новые классы и редактировать свойства существующих классов. Классы представляются в качестве папок иерархического дерева на панели юниверса.

→ Совет

Удобный способ для создания классов требует группировку связанных объектов измерения и объектов атрибута в классы, и размещение объектов мер в отдельный класс. Группировка связанных объектов может быть далее упорядочен с помощью использования подклассов чтобы разорвать объекты на подмножества. Подклассы описаны в разделе [Использование подклассов](#) [страница 271]

7.5.1 Создание класса

Существует два способа создания класса на панели юниверса:

- Определение класса вручную.

- С помощью перетаскивания таблицы со схемы таблицы на панель юниверса автоматически.

Оба способа описаны следующим образом:

7.5.1.1 Создание класса вручную

Можно создать классы вручную на панели юниверса. После анализа потребностей пользователей и составления классификационного списка групп объектов можно приступить к созданию классов вручную на основе составленных списков, что максимально позволит организовать структуру юниверса в соответствии с потребностями конечных пользователей

Чтобы создать класс на пустой панели юниверса, выполните следующие действия:

1. Выберите "Вставить" > "Класс".
Или
Нажмите кнопку "Вставить класс".
Откроется окно "Свойства класса".
2. Введите имя в текстовом поле "Имя файла"
3. Введите описание для класса в текстовом поле "Описание".
4. Нажмите кнопку "ОК".
Новое имя папки класса отобразится на панели юниверса.

→ Совет

Если щелкнуть значок "Применить" вместо "ОК", то имя и описание для класса будут применены, но окно свойств останутся открытыми. Если создать другой класс, то можно ввести свойства для нового класса в то же поле. Это позволит создать серии классов, которые используют единые свойства поля. Можно не тратить время на ненужные нажатия и избежать отображения новых свойств окна созданных классов.

7.5.1.2 Создание класса на панели юниверса с существующими классами

Чтобы создать класс с существующими классами, выполните следующие действия:

1. Нажмите на класс, который должен быть предшествующим нового класса в древовидное представление и выберите "Вставить" > "Класс".
Или
Нажмите на класс, который должен быть предшествующим нового класса в древовидное представление и нажмите кнопку "Вставить класс".
Откроется окно "Свойства класса".
2. Введите имя и описание для нового класса.
3. Нажмите кнопку "ОК".
Новое имя папки класса отобразится на панели юниверса.

7.5.1.3 Автоматическое создание класса из схемы таблиц

Классы можно создавать автоматически путем перетаскивания таблицы их схемы таблиц в панель юниверса. Имя таблицы является именем класса по умолчанию. Новые объекты создаются в классе также автоматически. Каждый новый объект соответствует столбцу таблицы.

Свойства новых классов и объектов необходимо редактировать, чтобы убедиться, что им присвоены соответствующие имена и они соответствуют потребностям конечных пользователей. Редактирование свойств объектов описано в разделе [Определение объектов \[страница 272\]](#).

Стратегия объектов, выбранная на странице стратегий диалогового окна "Параметры юниверса" определяет способ автоматического создания классов и объектов ("Файл">"Параметры">вкладка "Стратегии"). Эту стратегию можно изменить. Стратегии также можно создавать для настройки процесса создания классов и объектов. Дополнительную информацию о стратегиях см. в разделах [Использование внешних стратегий для настройки процесса создания юниверса \[страница 399\]](#) и [Выбор стратегий \[страница 88\]](#).

📘 Примечание

При автоматическом создании класса и объектов компоненты юниверса создаются непосредственно из структуры базы данных. Создаваемые класс и объекты должны быть результатом анализа потребностей пользователя, а не направляться столбцами и таблицами базы данных. Создание юниверса из потребностей пользователей описано в разделе [Методология создания юниверса \[страница 22\]](#).

Чтобы автоматически создать класс из схемы таблиц, выполните следующие действия.

1. Выберите таблицу в схеме таблиц.
2. Перетащите таблицу в панель юниверса и расположите в необходимом местоположении в иерархии класса.
В иерархии появится новый класс. Он будет содержать объект для каждого столбца таблицы, перемещенной в панель юниверса. По умолчанию имя класса совпадает с именем таблицы, а имя каждого объекта совпадает с именем соответствующего столбца.

7.5.2 Свойства класса

Для класса можно установить следующие свойства.

Свойство	Описание
Имя	Может содержать специальные символы. Должно быть уникальным для данного юниверса. Имя класса необходимо указать с учетом регистра. Данное имя можно изменить в любой момент.

Свойство	Описание
Описание	Комментарий, описывающий класс. Пользователь может просмотреть данное описание на панели запросов. Информация в этом поле должна быть сформулирована в деловом стиле и соответствовать требованиям запроса. Чтобы вставить конец строки, нажмите "CTRL+Возврат".

7.5.3 Изменение класса

Имя и описание класса можно изменить в диалоговом окне свойств класса в любое время. Существует несколько способов вызова диалогового окна свойств класса.

- Щелкните дважды папку класса.
- Щелкните правой кнопкой мыши папку класса, затем выберите "Редактировать" > "Свойства класса".
- Щелкните папку класса, затем выберите "Редактировать" > "Свойства класса".

📌 Примечание

Действия с помощью мыши можно выполнять с папкой класса или именем класса для вызова диалогового окна свойств класса.

7.5.4 Использование подклассов

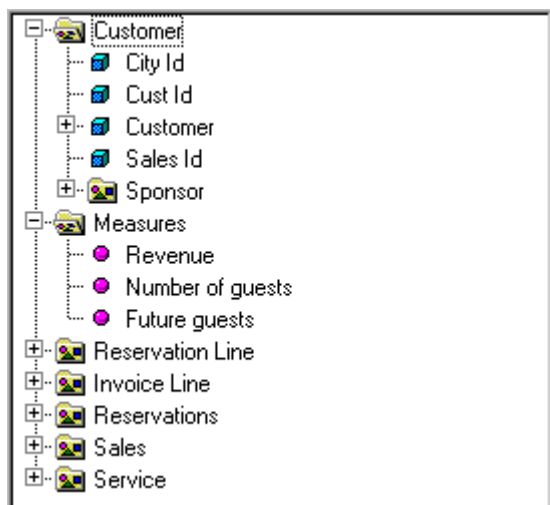
Подклассом является класс внутри класса. Подклассы используются для организации групп связанных объектов. Подкласс может также содержать подклассы или объекты.

7.5.4.1 Создание подкласса

Чтобы создать подкласс, выполните следующие действия.

- Нажмите папку или имя класса или имя, затем выберите "Вставить" > "Подкласс".
- Нажмите правой кнопкой мыши папку или имя класса, затем в контекстуальном меню выберите "Вставить подкласс".

На панели юниверса ниже представлен подкласс "Спонсор" в классе "Клиент".



7.6 Определение объектов

Объект – это компонент юниверса, который преобразуется в один или более столбцов одной или нескольких таблиц в схеме базы данных юниверса. Объект также может преобразовываться в функцию, указанную в одной или нескольких столбцах.

Каждый объект выводит инструкцию Select для столбца или функции, в которую он преобразуется. Когда пользователь Web Intelligence строит запрос с использованием одного или нескольких объектов на панели запросов, содержимое строки утверждения Select в инструкции Select выводится с помощью столбцов или функции, представленных каждым объектом.

7.6.1 Создание объекта

Объекты создаются на панели юниверса. Пользователи Web Intelligence находят объект по его имени и классификации. Объекты можно создавать вручную на панели "Юниверс" или автоматически, путем перетаскивания необходимой структуры базы данных с панели "Структура" на панель "Юниверс".

7.6.1.1 Создание объекта вручную

Объекты можно создавать вручную, путем вставления их в панель юниверса и последующего определения свойств объекта. Объект должен принадлежать к классу.

Чтобы создать объект вручную, выполните следующие действия.

1. Нажмите правой кнопкой мыши класс на панели юниверса, в контекстном меню выберите "Вставить объект".

Или

Выберите класс и нажмите средство "Вставить объект".

Объект вставлен в выбранный класс и затем открывается окно редактирования свойств объекта.

2. Введите имя в поле "Имя"
Имена объектов должны всегда соответствовать деловому словарю конечного пользователя. Это имя может отличаться от фактических имен столбца, с которым связан объект в схеме базы данных.
3. Нажмите вкладку "Свойства" и выберите свойства объекта.
4. Введите инструкцию Select в поле "Select" или нажмите кнопку "Select", чтобы воспользоваться редактором SQL.
5. Нажмите кнопку "ОК".

Связанные сведения

[Использование редактора SQL \[страница 288\]](#)

[Свойства объекта \[страница 274\]](#)

7.6.1.2 Автоматическое создание объекта

Объекты можно создавать автоматически, выбрав столбец таблицы на панели структуры и перетаскив его на панель юниверса. Объект создается в классе, ближайшем к указанному местоположению столбца. По умолчанию именем объекта является имя столбца. Все символы нижнего подчеркивания заменяются пробелами. Тип данных объекта по умолчанию выводится из типа данных столбца. Это значение можно изменить, выбрав новый тип данных в раскрывающемся списке на листе "Редактирование свойств" объекта.

Свойства нового объекта необходимо редактировать, чтобы убедиться, что ему присвоено соответствующее имя и он соответствует потребностям конечных пользователей. Редактирование свойств объектов описано в разделе [Определение объектов \[страница 272\]](#).

Стратегия объектов, выбранная на странице стратегий диалогового окна "Параметры юниверса" определяет способ автоматического создания классов и объектов ("Файл">"Параметры">вкладка "Стратегии"). Эту стратегию можно изменить. Стратегии также можно создавать для настройки процесса создания классов и объектов.

Дополнительные сведения об использовании стратегий см. в [Использование внешних стратегий для настройки процесса создания юниверса \[страница 399\]](#) и [Выбор стратегий \[страница 88\]](#).

📘 Примечание

При автоматическом создании класса и объектов компоненты юниверса создаются непосредственно из структуры базы данных. Создаваемые классы и объекты должны быть результатом анализа потребностей пользователя, а не направляться столбцами и таблицами базы данных. Создание юниверса из потребностей пользователей описано в разделе [Методология создания юниверса \[страница 22\]](#).

Чтобы создать объект автоматически, выполните следующие действия.

1. Выберите столбец таблицы на панели структуры.
2. Перетащите столбец на панель юниверса и разместите таблицу в необходимом месте иерархии класса. Столбец необходимо разместить в существующем классе.
В иерархии появится новый объект. По умолчанию имя объекта соответствует имени столбца.

Имена объектов должны всегда соответствовать деловому словарю конечного пользователя. Это имя может отличаться от фактических имен столбца, с которым связан объект в схеме базы данных.

7.6.2 Свойства объекта

В диалоговом окне "Редактировать свойства" для выбранного объекта можно определить следующие свойства.

Страница "Редактировать свойства"	Свойства
<p>Описание</p> <p>Подробные сведения о доступных свойствах определения объекта см. в разделе Определение объекта [страница 275].</p>	<ul style="list-style-type: none"> • Имя • Тип данных • Описание • Инструкция Select • Утверждение Where Можно получить доступ к редактору SQL на данной странице, чтобы определить синтаксис операторов SELECT и WHERE.
<p>Свойства</p> <p>Подробные сведения о доступных свойствах объекта см. в разделе Свойства [страница 278].</p>	<ul style="list-style-type: none"> • Классификация объекта • Объединенный список значений
<p>Дополнительно</p> <p>Подробные сведения о доступных дополнительных свойствах объекта см. в разделе Дополнительно [страница 279].</p>	<ul style="list-style-type: none"> • Защита • Права пользователя на объект • Форматы дат
<p>Ключи</p> <p>Сведения об определении поддержки индекса для объекта см. в разделе Определение поддержки индекса [страница 281].</p>	<ul style="list-style-type: none"> • Тип ключа • Выбрать • Where • Включить
<p>Исходные сведения</p> <p>Сведения об использовании данной вкладки см. в разделе Исходные сведения [страница 285].</p>	<ul style="list-style-type: none"> • Технические сведения • Сопоставление • Происхождение

Можно изменять свойства объекта в любое время. Подробные сведения о каждом объекте, указанном выше, см. на странице "Редактировать свойства" в разделе [Изменение объекта \[страница 275\]](#).

7.6.3 Изменение объекта

Свойства объекта можно определить в процессе создания объекта или изменять их в любое время. Можно определять свойства объекта в диалоговом окне "Редактировать свойства" (щелкните правой кнопкой объект > "Свойства объекта"). Свойства, определенные на каждой странице диалогового окна "Редактировать свойства", описаны следующим образом.

7.6.4 Определение объекта

Страница определения отображена ниже.

На странице определений в диалоговом окне "Редактировать свойства" можно определить следующие свойства.

Свойство	Описание	Необходимое/дополнительное
Имя	Имя объекта Оно может состоять из буквенно-цифровых символов, включая специальные символы и пробелы. Имя должно быть задано с учетом регистра. Имена объектов должны быть уникальны в пределах класса. Объекты в различных классах могут иметь одинаковое имя.	Требуется

Свойство	Описание	Необходимое/дополнительное
Тип	<p>Тип данных объекта. Объект может принадлежать к одному из четырех типов.</p> <ul style="list-style-type: none"> Символ Дата Длинный текст Число <p>В текущей версии средства создания юниверсов большие двоичные объекты не поддерживаются.</p>	Требуется
Описание	Комментариев объекта. Можно просматривать данное поле на панели запросов. Сюда также можно включать сведения об объекте, которые могут быть полезны конечному пользователю. Нажмите Ctrl+Return, чтобы переместить указатель на следующую строку.	Дополнительно
Выбрать	Инструкция Select, выводимая объектом. Для создания инструкции Select используется редактор SQL. См. раздел Свойства [страница 278] .*	Требуется
Where	Утверждение Where инструкции Select подразумевается объектом. Утверждение Where ограничивает количество строк, возвращаемых запросом. Для создания утверждения Where используется редактор SQL.*	Дополнительно

* Можно вставить или изменить @Prompt в инструкции Select или утверждении Where. Щелкните правой кнопкой мыши в инструкции Select или утверждении Where, и в контекстном меню отобразится элемент [Новый @Prompt](#), если @Prompt отсутствует в инструкции, или [Редактировать @Prompt](#), если щелкнуть внутри существующего @Prompt. Откроется окно редактора @Prompt.

Кнопка "Таблицы"

Для отображения списка таблиц, используемых в схеме, нажмите кнопку "Таблицы". В данном списке можно выбрать другие столбцы других таблиц для включения их в определение объекта. Это позволит объекту вывести в инструкции Select столбцы из нескольких таблиц. Дополнительные сведения см. в разделе [Применение ограничения путем логического вывода нескольких таблиц \[страница 308\]](#).

Кнопка "Анализ"

Для проведения анализа инструкции Select для объекта нажмите кнопку "Проверка". При обнаружении ошибок в синтаксисе выводится сообщение с описанием ошибки.

Связанные сведения

[Использование редактора SQL \[страница 288\]](#)

[Вычисляемые показатели в юниверсах OLAP \[страница 454\]](#)

[Редактор @Prompt \[страница 375\]](#)

7.6.4.1 Изменение определения объекта

Чтобы отредактировать определение объекта,

1. Дважды щелкните объект.
На странице определений откроется диалоговое окно "Изменение свойств".
2. Введите или выберите необходимое определение объекта и свойства.
3. Нажмите "ОК".

7.6.4.2 Определение объекта как динамической гиперссылки

Можно определить текст в ячейке как гиперссылку. Этот метод полезен для динамических гиперссылок в отчете, где текст в ячейках столбца становится гиперссылкой к конкретному ресурсу (зависящему от объекта результата).

Измените инструкцию SELECT объекта, чтобы включить объявление гиперссылки, и выберите свойство формата "Считывать как гиперссылку".

7.6.4.3 Определение объекта как динамической гиперссылки

Используя этот подход, объект будет создавать динамическую гиперссылку в результирующем отчете.

1. Щелкните правой кнопкой мыши объект и выберите [Свойства объекта](#).
Будет открыто диалоговое окно [Изменение свойств объекта](#).
2. Введите инструкцию SELECT и включите соответствующую гиперссылку.
3. Сохраните новые свойства.

- Щелкните правой кнопкой мыши объект и выберите [Формат объекта](#).
Отобразится панель [Формат объекта](#).
- Выберите параметр [Считывать как гиперссылку](#).
- Нажмите кнопку [OK](#), чтобы сохранить настройки формата.
- Это средство составления отчетов используется для создания отчета и тестирования ссылки.

Если объект используется в отчете, результирующий столбец будет включать гиперссылки.

Пример: Использование гиперссылки для календарных данных

Следующая инструкция SELECT получает данные годового календаря из timeanddate.com в зависимости от значения года в ячейке столбца. Обратите внимание, что объявление берет четыре самые правые символа (год) и удаляет обозначение FY (финансовый год) из строки, которое не распознается целевым URL-адресом.

```
'<a href=http://www.timeanddate.com/calendar/?year=>'+right(@Select(Reservations\Reservation Year),4) +'</a>'
```

7.6.5 Свойства

Можно указывать квалификации и свойства текущего объекта для списка значений на странице "Свойства" диалогового окна "Редактировать свойства".

Свойство	Описание
Квалификация	<p>Определенная роль объекта во время использования на панели запросов. Объект может быть квалифицирован как один из трех типов.</p> <ul style="list-style-type: none"> Измерение Объект-атрибут Мера <p>Более подробное описание квалификаций объекта см. в разделе Какие типы объектов используются в юниверсе? [страница 265].</p>
Связать со списком значений	<p>При выборе данного параметра файл, содержащий значения данных, связывается с объектом. Активирован по умолчанию. Дополнительные сведения см. в разделе Использование списков значений [страница 321].</p>

7.6.5.1 Указание квалификации и свойств списка значений для объекта

Чтобы указать квалификацию и свойства списка значений для объекта,

1. Дважды щелкните объект.
Откроется окно изменения свойств объекта.
2. Выберите вкладку "Свойства".
Откроется страница свойств.
3. Нажмите переключатель квалификации, чтобы определить, является данный объект измерением, объектом-описанием или мерой.
Чтобы связать список возвращенных значений с объектом, установите флажок "Связать список значений".
Дополнительные сведения о создании и использовании списков значений см. в разделе [Использование списков значений \[страница 321\]](#).
4. Нажмите кнопку "ОК".

7.6.6 Дополнительно

Ниже отображена страница "Дополнительно".

The screenshot shows a dialog box with three sections:

- Уровень безопасности**
Объект будет доступен пользователям с уровнем привилегий равным или большим, чем:
A dropdown menu is set to "Public".
- М.б. использовано в**
Three checkboxes are checked:
☒ Результат
☒ Условие
☒ Сортировка
- Формат базы данных**
По умолчанию формат определяет национальные установки. Вы можете определить другой формат.
An empty text input field is provided below the text.

На странице "Дополнительно" диалогового окна "Редактировать свойства" можно определить следующие свойства.

Свойство	Описание
Уровень защиты доступа	<p>Определяет уровень защиты доступа объекта. Можно выбирать уровень защиты, который ограничивает использование объекта для пользователей с соответствующим уровнем защиты.</p> <p>Можно установить следующие уровни защиты доступа.</p> <ul style="list-style-type: none"> • Общедоступный • Управляемый • С ограничениями • Конфиденциальный • Частный <p>При установке общедоступного уровня все пользователи могут видеть и использовать объект. Уровень защиты доступа с ограничениями позволяет видеть и использовать объект только пользователям, чей профиль соответствует данному уровню или превосходит его.</p>
Можно использовать в "Результат"	Выбранный объект можно использовать в запросе.
Можно использовать в "Условие"	Выбранный объект можно использовать и устанавливать в условии.
Можно использовать в "Сортировка"	Выбранные, возвращенные значения можно сортировать.
Формат базы данных	<p>Данный параметр доступен только для объектов дат.</p> <p>По умолчанию формат дат объекта установлен в диалоговом окне "Свойства региональных параметров" на панели управления MS-Windows. Его можно изменить в случае, когда для хранения дат используется формат целевой базы данных. Например, может использоваться формат даты, принятый в США, или европейский формат. Сведения об изменении данного значения см. в разделе Определение формата объекта [страница 289].</p>

7.6.6.1 Определение защиты объекта и прав пользователя

Чтобы определить защиту и права пользователя для объекта,

1. Дважды щелкните объект.
Откроется окно изменения свойств объекта.
2. Перейдите на вкладку "Дополнительно".
Появится страница "Дополнительные параметры".

3. Выберите уровень защиты доступа в окне раскрывающегося списка уровней защиты доступа.
4. Установите один или несколько флажков в групповом окне "Используется в".
5. Введите формат даты в текстовое окно формата базы данных, если необходимо изменить формат даты, установленный по умолчанию.
6. Нажмите кнопку "ОК".

7.6.7 Определение поддержки индекса

На вкладке "Ключи" можно определять поддержку индекса для объекта. Поддержка индекса – это возможность воспользоваться индексами на основе ключевых столбцов, чтобы ускорить извлечение данных.

Объекты, созданные в средстве создания юниверсов, основываются на столбцах базы данных, содержащих значимые данные для конечного пользователя. Например, объект "Клиент" извлекает поле с именем клиента. В данной ситуации в таблице клиентов обычно используется первичный ключ (например целое число), который не является значительным для конечного пользователя, но важен для производительности базы данных. При настройке поддержки индексирования в средстве создания юниверсов следует указать основные столбцы базы данных и внешние ключи. Это производит сильный эффект на производительность запросов следующими способами.

- Средство создания юниверсов поддерживает использование индексов ключевых столбцов для быстрого извлечения данных.
- Средство создания юниверсов поддерживает создание SQL-кода, обеспечивающего эффективную фильтрацию. Это чрезвычайно важно в базе данных со звездобразной схемой. Если создать запрос с фильтрацией на основе значения таблицы измерений, в средстве создания юниверсов можно применять фильтр на основе таблицы фактов с помощью внешнего ключа таблицы измерений. Таким образом можно устранить излишние и дорогостоящие объединения в таблицах измерений.

Средство создания юниверсов не пропускает повторяющиеся объекты с поддержкой индексирования. При совпадении имен двух клиентов средство создания юниверсов извлечет только одно, если не известно, что каждый клиент имеет отдельный первичный ключ.

Пример: Поиск клиентов в списке городов

В данном примере можно создать отчет на основе юниверса Island Resorts Marketing, который возвращает доход от клиента к клиенту в Хьюстоне, Далласе, Сан-Франциско, Сан-Диего или Лос-Анджелесе. Для этого необходимо перетащить объекты "Клиент" и "Доход от продаж" на панель "Объекты результатов" в панели запросов, а затем перетащить объект "Город" на панель "Условия" и ограничить город списком выше.

Если поддержка индексирования не включена, средство создания юниверсов создает следующий SQL-код:

```
SELECT
  Customer.last_name,
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
```

```

Customer,
Invoice_Line,
Service,
City,
Sales
WHERE
( City.city_id=Customer.city_id )
AND ( Customer.cust_id=Sales.cust_id )
AND ( Sales.inv_id=Invoice_Line.inv_id )
AND ( Invoice_Line.service_id=Service.service_id )
AND (
City.city IN ( 'Houston', 'Dallas', 'San Francisco', 'Los Angeles', 'San
Diego' )
)
GROUP BY
Customer.last_name

```

В данном случае в средстве проектирования юниверсов создается объединение с таблицей "Город", чтобы ограничить набор извлекаемых городов.

Если поддержка индексирования включена, в средстве создания юниверсов можно установить параметр `city_id` в качестве первичного ключа для таблицы "Город", который также будет отображаться в таблице "Клиент" в качестве внешнего ключа. С помощью этих сведений в средстве проектирования можно ограничивать города без объединения с таблицей "Город". SQL выглядит следующим образом.

```

SELECT
Customer.last_name,
sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
Customer,
Invoice_Line,
Service,
Sales
WHERE
( Customer.cust_id=Sales.cust_id )
AND ( Sales.inv_id=Invoice_Line.inv_id )
AND ( Invoice_Line.service_id=Service.service_id )
AND (
Customer.city_id IN (10, 11, 12, 13, 14)
)
GROUP BY
Customer.last_name

```

В данном случае в средстве проектирования можно генерировать SQL, в котором ограничиваются города с помощью фильтрации значений внешнего ключа `city_id`.

7.6.7.1 Установка поддержки индекса по первичному ключу

Чтобы установить поддержку индекса первичного ключа,

1. Щелкните правой кнопкой объект, для которого необходимо установить поддержку индекса, выберите [Свойства объектов](#) в меню. Откроется диалоговое окно "Редактировать свойства".
2. Щелкните вкладку [Ключи](#).
3. Нажмите [Вставить](#).

Строка "Первичный ключ" вставляется, как показано ниже на странице "Ключи".

4. Выполните следующие действия, чтобы создать поддержку ключей для первичного ключа.
 - Выберите "Первичный" в списке "Типы ключей".
Нажмите кнопку "..." в поле Select, чтобы открыть диалоговое окно редактирования SQL.

Отобразится редактор SQL.

- Воспользуйтесь редактором SQL, чтобы создать SQL-утверждение SELECT для первичного ключа, или введите его напрямую. Например, для объекта "Город" первичным ключом является city.city_id

Дополнительные сведения о редакторе SQL см. в разделе [Использование редактора SQL \[страница 288\]](#).

- Выберите тип данных первичного ключа в раскрывающемся списке типов ключей.
5. Если необходимо добавить утверждение WHERE, выполните следующие действия.
 - Щелкните строку под столбцом Where, как показано ниже.

- Нажмите кнопку "..." в поле Where, чтобы открыть диалоговое окно редактирования SQL.

Отобразится редактор SQL.

- Воспользуйтесь редактором SQL, чтобы создать SQL-утверждение WHERE для первичного ключа, или введите его напрямую. В примере, указанном выше, нет утверждения Where.
- Выберите номер в раскрывающемся списке типов ключей.

6. Нажмите [Активировано](#).
7. Нажмите кнопку "ОК".

❗ Примечание

Чтобы добавить несколько столбцов для первичного ключа, их можно объединить, чтобы определить первичный ключ. Данные столбцы должны быть в одной таблице и иметь одинаковый тип данных.

напр., в базе данных "club.mdb" в таблице "Курорт" есть первичный ключ с несколькими столбцами на основе Country_id и Resort_id.

Таким образом, чтобы определить поддержку индекса на основе объекта <курорт>, пользователь может объединить "Country_id" и "Resort_id", чтобы определить первичный ключ.

Resort.country_id & Resort.resort_id

& – это оператор конкатенации MS Access.

7.6.7.2 Установка поддержки внешнего ключа

Чтобы установить поддержку внешнего ключа,

1. Правой кнопкой щелкните объект, для которого необходимо установить поддержку индекса. В меню выберите "Свойства объектов". Откроется диалоговое окно "Редактировать свойства".
2. Щелкните вкладку *Ключи*.
3. Нажмите *Вставить*. Строке ключей вставиться на странице "Ключи".
4. Выполните следующие действия, чтобы создать поддержку ключей для внешнего ключа.
 - Выберите "Внешний ключ" в списке "Типы ключей".
 - Нажмите кнопку "..." в поле Select, чтобы открыть диалоговое окно редактирования SQL. Отобразится редактор SQL.
 - Воспользуйтесь редактором SQL, чтобы создать SQL-утверждение SELECT для внешнего ключа, или введите его напрямую.
 - Выберите тип данных внешнего ключа в раскрывающемся списке типов ключей.
5. Повторите шаги 3 и 4 для всех столбцов, которые составляют внешний ключ.
6. Если необходимо добавить утверждение WHERE, выполните следующие действия.
 - Щелкните выделенную строку под столбцом Where.
 - Нажмите кнопку "..." в поле Where, чтобы открыть диалоговое окно редактирования SQL. Отобразится редактор SQL.
 - Воспользуйтесь редактором SQL, чтобы создать SQL-утверждение WHERE для внешнего ключа, или введите его напрямую.
 - Выберите номер в раскрывающемся списке типов ключей.
7. Нажмите *Активировано*.
8. Повторите шаги, указанные выше, для всех столбцов во внешнем ключе.

Например, в разделе *Определение поддержки индекса* вкладка *Ключи* должна выглядеть следующим образом:

Задать для объекта первичный и внешний ключ. Номер

Тип ключа	Выбор	Where	Включить
Основной ключ	City.city_id		<input checked="" type="checkbox"/>
Внешний ключ	▼ Customer.city_id		<input checked="" type="checkbox"/>

Вставить Удалить Поиск... Тест

7.6.8 Исходные сведения

Страница исходных сведений используется юниверсами, сгенерированными с помощью приложения Data Integrator. Ниже отображена вкладка "Исходные сведения".

Исходные сведения

Технические сведения

Отображение

Происхождение

Для юниверсов, сгенерированных с помощью приложения Data Integrator, техническое описание и формулы используются, чтобы вычислить целевые таблицы на основе исходных таблиц, отображаемых на этой вкладке. Данные сведения доступны пользователям Web Intelligence.


На вкладке "Исходные сведения" можно указать следующие типа сведений.

- Техническое описание. Техническое описание доступно в юниверсах, сгенерированных с помощью приложения Data Integrator.
- Сопоставление. Сопоставление между исходной и целевой таблицей, применяемое в приложении Data Integrator. В данном случае, цель состоит не в том, чтобы предоставить выражение сопоставления, а в том, чтобы отобразить его в качестве описательного комментария для информирования пользователя исходных столбцов, используемых в определении объекта.
- Сведения о происхождении данных. Список исходных столбцов содержится в целевом столбце. Эти сведения упрощают анализ факторов влияния при составлении отчетов Data Integrator и Web Intelligence.

7.6.9 Использование редактора SQL для определения объекта

Редактор SQL можно использовать для определения инструкции Select или утверждения Where для объекта, а также для вставки операторов и функций MDX для объектов юниверса OLAP. Редактор SQL является графическим редактором, который выводит таблицы, столбцы, объекты, операторы и функции в трех видах. Двойное нажатие любой из перечисленных структур позволяет поместить ее в поле Select или Where.

В редакторе SQL есть следующие параметры редактирования.

Параметры редактирования	Описание
Таблицы и столбцы	Все таблицы и соответствующие им столбцы, которые отображаются в панели структур. <div> Примечание Этот параметр доступен только для реляционных юниверсов и не доступен для юниверсов OLAP.</div>
Классы и объекты	Все классы и соответствующие им объекты, которые отображаются на панели юниверса.
Операторы	Операторы, с помощью которых комбинируются структуры SQL в инструкции Select или задаются условия в утверждении Where.

Параметры редактирования	Описание
Функции	<ul style="list-style-type: none"> • Функции баз данных, например, число, символ и функции даты. • Функции @Function, специфические для продуктов Business Objects. <p>Доступные функции перечислены в разделе функций (Functions) в файле параметров (.PRM) для целевой базы данных. Для каждой поддерживаемой базы данных существует файл .PRM. Они хранятся в папке доступа к данным (Data Access) в каталоге BusinessObjects. Можно добавлять или изменять доступные функции, редактируя файл .PRM.</p> <p>Редактирование файлов .PRM описано в "Руководстве по доступу к данным".</p>
Показывать SQL объекта	При выборе этой команды выводится синтаксическая конструкция на языке SQL для объектов, которые появляются в полях Select и Where.
Разбор	Эта команда выполняет синтаксический разбор. Если синтаксис является недопустимым, появляется окно сообщения с описанием проблемы.
Описание	Отображает описание выбранного объекта или функцию.

Связанные сведения

[Функции MDX для запросов к кубу \[страница 456\]](#)

[Использование редактора SQL \[страница 288\]](#)

7.6.9.1 Установка проекции агрегирования для меры

Чтобы создать меру, необходимо указать путь, с помощью которого функция агрегирования будет спроецирована в отчете.

Возвращенные значения меры объекта агрегируются на двух уровнях в процессе запроса.

- Уровень запроса. Данные агрегируются с помощью унаследованной инструкции SELECT.
- Из микрокуба на уровень блока. Если данные проецируются из микрокуба в блок отчета. Функция проекции мер позволяет локальное агрегирование в микрокубе.

📘 Примечание

Микрокуб – это концептуальный путь, чтобы представить данные, возвращенные с помощью запроса, перед проецированием в отчет. Он представляет возвращенные значения, хранящиеся в памяти, с помощью продукта отчета Business Objects. Уровень блока – это отчет на основе двух измерений, который пользователь создает с помощью возвращенных данных. Пользователь может использовать все или только некоторые данные, хранящиеся в микрокубе, чтобы создать отчет.

Пользователь может также выполнить функции агрегирования на основе возвращенных значений в микрокубе (локальное агрегирование), чтобы создать новые значения в отчете.

Два уровня агрегирования соответствуют процессу запроса, как показано ниже.

- Пользователь создает запрос в Web Intelligence.
- Web Intelligence выводит SQL из запроса и отправляет инструкцию SELECT в целевую базу данных.
- Данные возвращаются в микрокуб. Это первый уровень агрегирования.
- Микрокуб проецирует агрегированные данные в отчет. На панели запросов данные разделяются для агрегирования на низших уровнях. Это второй уровень агрегирования.

После выполнения запроса набор результатов инструкции Select хранится в микрокубе, а все данные микрокуба проецируются в блок. Так как данные проецируются из низшего уровня микрокуба, агрегирование проекции не производится.

Однако когда панель запросов используется для проецирования лишь части данных из микрокуба требуется агрегирование, чтобы отобразить значения мер на более высоком уровне.

Если в предыдущем примере не спроецировать данные за год в блок, три строки, связанные с годом, необходимо сократить до одной, чтобы отобразить общий доход от продаж для курорта. В данном случае используется агрегирование суммы.

Агрегирование проекции указывается на странице [Свойства](#) листа [Изменить свойства](#) меры (правой кнопкой щелкните "Объект" > "Свойства объекта" > "Свойства").

Агрегирование проекции отличается от агрегирования оператора SELECT.

7.6.9.2 Использование редактора SQL

Редактор SQL можно использовать для вставки выражений SQL и MDX в определение объектов. Также можно щелкнуть правой кнопкой мыши в SQL-инструкции и выбрать [Новый @Prompt](#) для вставки выражения @Prompt в SQL или выбрать [Редактировать @Prompt](#) для изменения существующего выражения @Prompt. Откроется окно редактора @Prompt.

Чтобы использовать редактор SQL, выполните следующие действия.

1. Дважды нажмите объект.
Появится диалоговое окно изменения свойств объекта.
2. Нажмите кнопку ">>" рядом с полем Select или Where.
Появится окно редактирования инструкции Select или утверждения Where.
3. Щелкните в том месте инструкции Select или утверждения Where, куда необходимо добавить синтаксис для структуры. Если поле пусто, щелкните в любом его месте. Курсор автоматически появится в левом верхнем углу поля.
4. Разверните узлы таблицы, чтобы отобразить столбцы.
5. Щелкните столбец два раза, чтобы вставить определение столбца в инструкцию Select или утверждение Where.

→ Совет

Чтобы выбрать в списке одно или больше значений для выбранного столбца, щелкните правой кнопкой мыши столбец и выберите в контекстном меню "Список значений".

6. Разверните узлы класса, чтобы отобразить объекты.
7. Щелкните объект два раза, чтобы вставить функцию @Select или @Where в инструкцию Select или утверждение Where. Эти функции предписывают текущему объекту использовать инструкцию Select или утверждение Where выбранного объекта. Дополнительные сведения об использовании функций @Function см. в разделе [Использование функций @Function в SQL-коде для объекта \[страница 370\]](#).
8. Два раза нажмите оператор, чтобы поместить его в поле редактирования.
9. Разверните узлы функций, чтобы отобразить доступные функции.
10. Дважды щелкните функцию, чтобы поместить ее в поле редактирования.
11. Нажмите кнопку "Разбор" для проверки достоверности синтаксиса.
12. Нажмите "ОК".

7.6.10 Определение формата объекта

Для значений данных выбранного объекта можно определить формат. Он будет применяться к значениям связанных данных, выводимым в ячейках отчетов Web Intelligence.

На вкладке диалогового окна "Формат объекта" осуществляются настройки для чисел, выравнивания, шрифта, границы и затенения.

Например, можно выводить целое число в формате 1000 долл. США вместо формата по умолчанию 1,000 00, или применять цвет, например красный, к значениям критически важных данных.

Категории "Числовой", "Денежный", "Экспоненциальный" и "Процентный" применяются к объектам и переменным только числового типа, а категория "Дата" или "Время" применяется только к данным соответствующего типа.

Сведения о форматах экспортируются и импортируются вместе с универсом.

Команда "Удалить формат объекта" используется для удаления определенного формата.

Связанные сведения

[Функции MDX для запросов к кубу \[страница 456\]](#)

7.6.10.1 Изменение формата объекта

Чтобы изменить формат объекта, выполните следующие действия.

1. Щелкните объект правой кнопкой мыши.
2. Выберите "Формат объекта" в контекстном меню.
Откроется лист "Формат объекта".
3. Перейдите на вкладку формата и выберите или введите формат объекта.
4. Нажмите кнопку "ОК".

7.6.10.2 Удаление формата объекта

Формат объекта можно удалить в любое время.

Чтобы удалить формат объекта, выполните следующие действия.

- Выберите объект и перейдите в меню "Файл" > "Удалить формат".
Или
- Щелкните объект правой кнопкой мыши и в контекстном меню выберите команду "Удалить формат".

7.6.11 Просмотр таблицы, используемой в определении объекта

Таблицу можно просмотреть на панели "Структура", которая применяется в определении объекта из панели "Юниверс". Это позволяет быстро определить таблицу, используемую объектом, когда имена объектов не указывают явным образом на конкретную таблицу.

7.6.11.1 Просмотр таблицы, используемой объектом

Чтобы просмотреть таблицу, используемую объектом, выполните следующие действия.

1. Щелкните правой кнопкой мыши объект на панели "Юниверс".
Появится контекстное меню.
2. Выберите в нем команду "Просмотреть связанную таблицу".
Связанная таблица будет подсвечена на панели "Структура".

7.6.12 Определение измерения

Измерение – это объект, который является фокусом анализа в запросе. Измерение сопоставляется одному или нескольким столбцам или функциям в базе данных, которые являются ключевыми для запроса. Например, "Страна", "Продавец", "Продукты" или "Линия продаж".

Измерение является классификацией (квалификацией) по умолчанию при создании объекта. Классификацию для измерения можно изменить в любое время.

Чтобы определить объект измерения, выполните следующие действия.

1. Дважды щелкните объект.
Появится диалоговое окно объекта Редактировать свойства.
2. Выберите вкладку "Свойства".
Откроется страница свойств.
3. Выберите переключатель "Измерение" в групповом блоке "Классификация".
4. Нажмите кнопку "ОК".

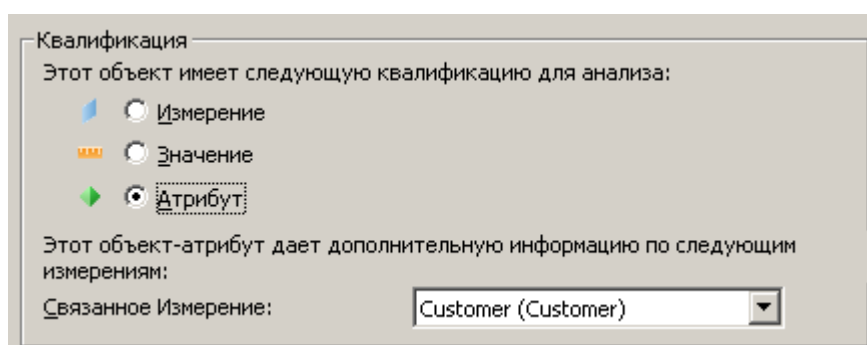
7.6.13 Определение объекта-описания

Объект-описание содержит описательные данные об измерении. Объект-описание всегда прикрепляется к измерению. Он сопоставляется одному или нескольким столбцам или функциям в базе данных, которые содержат подробные сведения, связанные с измерением.

Объект-описание определяется путем выбора "Объект-описание" как классификации для объекта и указания измерения, прикрепленного к объекту-описанию.

Чтобы определить объект-описание, выполните следующие действия.

1. Дважды щелкните объект.
Появится диалоговое окно объекта Редактировать свойства.
2. Выберите вкладку "Свойства".
Откроется страница свойств.
3. Выберите переключатель "Объект-описание" в групповом блоке "Классификация".
Откроется выпадающий список "Связанное измерение", в котором содержатся все объекты измерений в универсе.
4. Выберите в нем измерение. Объект-описание описывает качество или свойство этого измерения.



5. Нажмите кнопку "OK".

7.6.14 Определение меры

Объект меры определяется путем выбора классификации объекта "Мера". Меры являются очень гибкими объектами благодаря своей динамичности. Возвращаемые значения для объекта меры меняются в зависимости от объектов измерений и описаний, используемых вместе с ним в запросе. Например, мера "Доход от продаж" возвращает различные значения, когда используется с объектом "Страна" в одном запросе, а затем с объектами "Регион" и "Страна" в отдельном запросе.

Поскольку объекты мер являются более сложными и мощными по сравнению с измерениями и объектами-описаниями, они обсуждаются более подробно в следующих разделах.

7.6.14.1 Тип сведений, возвращаемый показателем

Объект показателя возвращает числовые данные. Показатель создается с помощью агрегированных функций. Следующие пять агрегированных функций применяются наиболее часто:

- Сумма
- Количество
- Среднее
- Минимум
- Максимум

7.6.14.2 Отличия показателей от измерений и объектов-описаний

Показатели отличаются от измерений и объектов-описаний по следующим параметрам.

- Показатели являются динамическими.
- Показатели могут проектировать агрегаты.

Оба этих свойства описываются следующим образом.

7.6.14.3 В чем проявляется динамичность поведения мер?

Возвращаемые значения для объекта меры меняются в зависимости от объектов измерений и описаний, используемых вместе с ним в запросе.

В следующем примере показан тот же объект меры "Доход", используемый в двух отдельных запросах с различными измерениями, которые дают результат в виде меры, возвращающей различные значения.

Объекты результата

Resort Year Revenue

Условия

Resort Equal to 'Bahamas Beach'

Year	Revenue
FY00	307,400.00
FY01	376,115.00
FY99	287,929.00

Одна и та же мера возвращает разные результаты

Объекты результата

Service Year Revenue

Условия

Resort Equal to 'Bahamas Beach'
And Year Equal to 'FY01'

Service	Revenue
Activities	22,400.00
Bungalow	59,040.00
Excursion	15,300.00
Fast Food	5,840.00
Hotel Room	60,288.00
Hotel Suite	115,632.00
Poolside Bar	14,280.00
Restaurant	75,335.00
Sports	8,000.00

7.6.14.4 Меры логически выводят утверждение Group By

Когда выполняется запрос, который включает объект меры с другими типами объектов, утверждение Group By автоматически включается в инструкцию Select.

Включение утверждения Group By зависит от следующего правила языка SQL.

Если строка утверждения Select содержит агрегированный показатель, все, что есть за его пределами, должно быть отображено в утверждении Group By.

Согласно этому правилу, любое измерение или объект-описание, используемые в том же запросе, что и объект меры, всегда будут включаться в автоматически генерируемое утверждение Group By. Чтобы запрос всегда возвращал корректные результаты, объекты измерений и описаний НЕ должны содержать агрегированные показатели.

В следующем примере показано, что все объекты измерений "Курорт", "Линия обслуживания" и "Год" автоматически вставляются в утверждения Select и Group By.

Result Objects

Resort Service Line Year Revenue

Conditions

Resort Equal to 'Bahamas Beach'

Measurements

Revenue

Group By

Resort.resort, Service_Line.service_line, 'FY'+Format(Sales.invoice_date, 'YY')

SELECT

```
SELECT
  Resort.resort,
  Service_Line.service_line,
  'FY'+Format(Sales.invoice_date, 'YY'),
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
  Resort,
  Service_Line,
  Sales,
  Invoice_Line,
  Service
WHERE
  ( Invoice_Line.inv_id=Sales.inv_id )
  AND ( Invoice_Line.service_id=Service.service_id )
  AND ( Resort.resort_id=Service_Line.resort_id )
  AND ( Service.sl_id=Service_Line.sl_id )
  AND (
    Resort.resort = 'Bahamas Beach'
  )
GROUP BY
  Resort.resort,
  Service_Line.service_line,
  'FY'+Format(Sales.invoice_date, 'YY')
```

Bahamas Beach

Year	Service Line	Revenue
FY00	Accommodation	225,240.00
FY00	Food & Drinks	38,360.00
FY00	Recreation	43,800.00
FY01	Accommodation	234,960.00
FY01	Food & Drinks	95,455.00
FY01	Recreation	45,700.00
FY99	Accommodation	213,464.00
FY99	Food & Drinks	35,865.00
FY99	Recreation	38,600.00

Измерения, полученные в результате работы оператора GROUP BY

Результаты, агрегированные по худшим курортам, затем по линии обслуживания и году

Примечание

Если запрос содержит только объекты мер, утверждение Group By не генерируется.

7.6.14.5 Установка проекции агрегирования для меры

Чтобы создать меру, необходимо указать путь, с помощью которого функция агрегирования будет спроецирована в отчете.

Возвращенные значения меры объекта агрегируются на двух уровнях в процессе запроса.

- Уровень запроса. Данные агрегируются с помощью унаследованной инструкции SELECT.
- Из микрокуба на уровень блока. Если данные проецируются из микрокуба в блок отчета. Функция проекции мер позволяет локальное агрегирование в микрокубе.

❗ Примечание

Микрокуб – это концептуальный путь, чтобы представить данные, возвращенные с помощью запроса, перед проецированием в отчет. Он представляет возвращенные значения, хранящиеся в памяти, с помощью продукта отчета Business Objects. Уровень блока – это отчет на основе двух измерений, который пользователь создает с помощью возвращенных данных. Пользователь может использовать все или только некоторые данные, хранящиеся в микрокубе, чтобы создать отчет. Пользователь может также выполнить функции агрегирования на основе возвращенных значений в микрокубе (локальное агрегирование), чтобы создать новые значения в отчете.

Два уровня агрегирования соответствуют процессу запроса, как показано ниже.

- Пользователь создает запрос в Web Intelligence.
- Web Intelligence выводит SQL из запроса и отправляет инструкцию SELECT в целевую базу данных.
- Данные возвращаются в микрокуб. Это первый уровень агрегирования.
- Микрокуб проецирует агрегированные данные в отчет. На панели запросов данные разделяются для агрегирования на низших уровнях. Это второй уровень агрегирования.

После выполнения запроса набор результатов инструкции Select хранится в микрокубе, а все данные микрокуба проецируются в блок. Так как данные проецируются из низшего уровня микрокуба, агрегирование проекции не производится.

Однако когда панель запросов используется для проецирования лишь части данных из микрокуба требуется агрегирование, чтобы отобразить значения мер на более высоком уровне.

Если в предыдущем примере не спроецировать данные за год в блок, три строки, связанные с годом, необходимо сократить до одной, чтобы отобразить общий доход от продаж для курорта. В данном случае используется агрегирование суммы.

Агрегирование проекции указывается на странице [Свойства](#) листа [Изменить свойства](#) меры (правой кнопкой щелкните "Объект" > "Свойства объекта" > "Свойства").

Агрегирование проекции отличается от агрегирования оператора SELECT.

7.6.14.5.1 Функция проекции "Делегированная база данных"

Любая мера юниверса может обладать функцией проекции ([Sum](#), [Min](#), [Max](#), [Count](#) и [Avg](#)). Функция проекции используется для локального агрегирования меры в приложении Web Intelligence, когда

количество отображаемых в отчете измерений меньше, чем количество измерений в наборе результатов запроса.

Неаддитивные меры, такие как отношение, среднее значение и вес могут быть отражены только на том же уровне агрегирования, что и набор результатов запроса. Поэтому функция проекции неаддитивных мер в юниверсе обычно устанавливается на значение *Нет*.

Функция проекции *Делегированная база данных* позволяет делегировать агрегирование неаддитивных мер на сервер баз данных. В приложении Web Intelligence они называются интеллектуальными мерами. На странице свойств объекта функция проекции интеллектуальной меры устанавливается на значение *Делегированная база данных*. Дополнительные сведения об использовании этих и других функций в Web Intelligence см. в разделе *Расчет значений с интеллектуальными мерами* в документе *Использование функций, формул и вычислений в Web Intelligence*.

❗ Примечание

Для юниверсов OLAP, основанных на источниках данных MSAS и Essbase, все меры юниверса создаются с функцией проекции, установленной по умолчанию на значение *Делегированная база данных*.

❗ Примечание

При использовании интеллектуальной меры, основанной на мере с поддержкой агрегирования, необходимо учесть следующие ограничения: настоятельно рекомендуем убедиться, что таблицы агрегирования, используемые при определении меры, содержат непротиворечивые данные (значения агрегирования соответствуют значениям объектов-атрибутов). В противном случае интеллектуальные меры могут выдать неточные данные. Например, если таблицы агрегирования по году и по дню используются для интеллектуальной меры, годовая таблица агрегирования будет соответствовать дневной таблице по завершенным годам, но по текущему году годовая таблица может быть пустой, в то время как дневная таблица будет содержать точные данные по дням. В этом случае отчет, использующий интеллектуальные меры на основе текущих таблиц года и дня, может выдать неточные результаты.

Пример: Интеллектуальная мера

В данном примере запрос содержит два измерения: "Страна" и "Регион" и три меры: "Сумма заказа", "Сумма доставки" и "% доставки".

Регион L01	Сумма доставки	Сумма заказа	% доставки
Reg1	497318880	497332680	99,997
Reg2	199463776	199466536	99,998
Reg3	198927552	198933072	99,997
Сумма:			299,992

Сумма меры "% доставки" неверна, поскольку представляет собой сумму данных столбца "% доставки".

Если в юниверсе функция проекции этой меры имеет значение *База данных делегирована*, то при обновлении отчета Web Intelligence соединяется с базой данных для расчета верного значения.

Регион LO1	Сумма доставки	Сумма заказа	% доставки
Reg1	497318880	497332680	99,997
Reg2	199463776	199466536	99,998
Reg3	198927552	198933072	99,997
		Сумма:	299,992
		Всего:	99,997

📘 Примечание

Некоторые функции, например функцию соотношения (Average), необходимо использовать с осторожностью. При расчете среднего значения столбца поведение этой функции может оказаться непредвиденным, если она была неправильно сконфигурирована.

Например, функция SQL `sum(Shop_facts.Margin)/sum(Shop_facts.Quantity_sold)` может выдать непредвиденные результаты. Если она сконфигурирована неправильно, то она произведет расчет среднего значения каждой ячейки и выдаст сумму средних значений. Чтобы исправить такое поведение, необходимо задать параметры функции следующим образом.

1. Перейдите к параметру функции *Изменить свойства*.
2. Для параметра *Выберите способ проектирования меры при агрегировании* в раскрывающемся списке "Функция" выберите функцию *Db delegated*.
3. Сохраните изменения.

Связанные сведения

[Установка проекции агрегирования для меры \[страница 453\]](#)

7.6.14.6 Создание меры

Как создать меру

1. Дважды нажмите объект.
Появится диалоговое окно объекта Редактировать свойства.
2. Выберите вкладку "Свойства".
Откроется страница свойств.
3. Выберите переключатель меры в поле группы квалификации.
Поле выпадающего списка функции отображает функции агрегирования.
4. Укажите функцию.

5. Нажмите кнопку ОК.

Связанные сведения

[Установка проекции агрегирования для меры \[страница 453\]](#)

[Функция проекции "Делегированная база данных" \[страница 294\]](#)

7.6.15 Определение ограничений для объекта

Ограничение – это условие в SQL, устанавливающее критерии по ограничению вывода данных по запросу.

Ограничения определяются для объекта с целью ограничения доступа к данным для других пользователей. Основой для ограничения доступа к данным для пользователей служат запросы данных целевых пользователей. Пользователю могут быть необходимы конкретные данные без отображения других. Ограничить доступ к данным для пользователей можно также, исходя из соображений безопасности

В юниверсе используются два типа ограничения

Тип ограничения	Описание
Принудительный	Ограничение, определенное для объекта в утверждении Where. Данные недоступны для пользователей и таким образом не могут быть переадресованы в Web Intelligence.
Дополнительный	Ограничение, определенное для специальных объектов условия, которые могут быть выбраны для пользования или других целей в запросах. Объект условия – предопределенное утверждение Where, которое может быть добавлено в инструкцию Select, выводимую объектами на панели запросов.

❗ Примечание

В Web Intelligence пользователи могут применять условия на панели запросов. После создания юниверса не стоит создавать дополнительные ограничения, которые легко применить на пользовательском уровне. При необходимости пользователи сами могут создать необходимые условия.

7.6.15.1 Определение утверждения Where для объекта

Необходимо применить дальнейшее ограничение для объекта с помощью добавления условия в поле *Where* на странице [Определение](#) диалогового окна [Редактировать свойства](#).

Условие можно определить в процессе создания объекта или добавить его в определение объекта в любой момент.

В юниверсе утверждение Where в SQL-инструкции можно использовать двумя способами, чтобы ограничить количество строк, возвращаемых запросом.

- Утверждение WHERE автоматически выводится в инструкции SELECT для объекта с помощью объединений таблиц в схеме. Объединения основаны на равенстве таблиц. Они препятствуют созданию декартова произведения с помощью ограничения данных, возвращаемых из объединенных таблиц.
- Добавьте условие в утверждение WHERE для объекта. Это дополнительное условие для существующего утверждения WHERE, выведенного с помощью объединений. Утверждение WHERE определяется, чтобы еще больше ограничить данные, которые будут возвращаться в запросе. Например, оно используется, если нужно ограничить пользователей запросами к подмножеству данных.

Пример: Изменение утверждения Where, используемого по умолчанию (только объединение), для объекта

Отчет, указанный ниже, является неограниченным блоком с данными для продавцов всех стран.

Sales Person	Country of origin
Barrot	France
Carlin	France
Edwood	UK
Fischer	Germany
Galagers	US
Ishimoto	Japan
Nagata	Japan

Ниже отобразится SQL для данного запроса. В утверждении Where содержатся только ограничения, выведенные с помощью объединений таблиц "Клиент", "Город", "Регион" и Sales_Person.

```
SELECT
  Sales_Person.sales_person, Country.country
FROM
  Sales_Person,
  Country,
  Region,
  City,
  Customer
WHERE
  ( City.city_id=Customer.city_id )
```

```

AND ( City.region_id=Region.region_id )
AND ( Country.country_id=Region.country_id )
AND ( Sales_Person.sales_id=Customer.sales_id )

```

Если необходимо, чтобы пользователи видели возвращаемые значения, относящиеся только к Франции, можно добавить условие в утверждение Where объекта "Страна". В следующем отчете отображены сведения только о продавцах Франции.

Sales Person	Country of origin
Barrot	France
Carlin	France

SQL для запроса выглядит следующим образом.

```

SELECT
    Sales_Person.sales_person,
    Country.country
FROM
    Sales_Person,
    Country,
    Region,
    City,
    Customer
WHERE
    ( City.city_id=Customer.city_id )
    AND ( City.region_id=Region.region_id )
    AND ( Country.country_id=Region.country_id )
    AND ( Sales_Person.sales_id=Customer.sales_id )
    AND ( Country.country = 'France' )

```

Утверждение WHERE – это дополнительная строка. Это ограничение, добавленное в утверждение WHERE объекта "Страна".

📌 Примечание

В утверждениях WHERE не следует создавать объединений, кроме самоограниченных. Объединение в утверждении WHERE не рассматривается функцией обнаружения контекстов (автоматическое определение контекста) или обнаружения несовместимости поддержки агрегирования. Убедитесь, что все объединения являются видимыми на панели [Структура](#). Это гарантирует, что все объединения доступны для инструментов автоматического определения средства создания юниверсов.

7.6.15.2 Определение утверждения Where

Как определить утверждение Where:

1. Дважды щелкните объект.
На странице определений откроется диалоговое окно "Редактировать свойства".
2. Введите синтаксис напрямую в текстовое окно утверждения Where.
Или
Нажмите кнопку ">>" рядом с окном утверждения Where, чтобы открыть редактор утверждения Where.

3. Дважды щелкните мышью столбцы, объекты, операторов или функции в структурах SQL и списках свойств.

→ Совет

Можно выбрать значения для утверждения Where следующим образом: щелкните правой кнопкой мыши столбец в списке таблиц и столбцов. Выберите "Значения просмотра". Отобразится список всех значений столбца. Можно выбрать одно или несколько значений для вставки в утверждение Where, например при использовании оператора In.

4. Нажмите OK для закрытия редактора.
Утверждение Where для объекта страны показано ниже. Это ограничивает значения для страны, но только для Франции.

The screenshot shows a SQL editor window with the following fields:

- Имя:** Country of origin
- Тип:** Символ
- Описание:** Customer's country of origin
- Select:** Country.country
- Where:** Country.country='France'

At the bottom, there are two buttons: "Таблицы..." and "Тест".

5. Нажмите кнопку "OK".

7.6.15.3 Проблемы при использовании утверждений Where

Утверждения Where полезны для ограничения данных, но в юниверсе их необходимо использовать с осторожностью, чтобы избежать следующих проблем:

Проблема	Описание	Решение
Распространение похожих объектов.	При ограничении данных для объекта путем создания нескольких объектов, каждый из которых выводит утверждение Where для одной части данных, можно получить множество объектов с похожими именами. Например, французские клиенты, клиенты США и японские клиенты. Многочисленные объекты, кажущиеся похожими, могут привести в замешательство пользователей.	Создайте объекты условия для каждого ограничения.
Иерархии, вызывающие трудности.	При наличии нескольких объектов, которые выводят утверждения Where для одних и тех же данных, пользователям будет сложно создать логическую иерархию по умолчанию для развертки.	Создайте объекты условия для каждого ограничения.
Путаница между именем объекта и примененным ограничением.	До момента, пока объекты не получат точного имени, ограничение не будет видимо пользователям по одному только имени объекта. Пользователь может видеть утверждение Where, просматривая SQL для запроса, но не все пользователи смогут просматривать SQL, не запустив запрос.	<ul style="list-style-type: none"> Создайте объекты условия для каждого ограничения. Укажите имя для каждого объекта соответственно.
Конфликт между утверждениями Where.	Если два или более аналогично ограниченных объекта включены в один запрос, возникнет конфликт между утверждениями Where, и данные не будут возвращены.	Создайте объекты условия для каждого ограничения и убедитесь, что пользователи выполняют объединение или синхронизацию запросов на уровне отчета.

Создание объектов условия разрешит многочисленные проблемы, связанные с объектами, иерархией и путаницей имен объектов.

Конфликт между утверждениями Where может быть разрешен путем создания объектов условия и обеспечением того, чтобы пользователи знали, что необходимо объединять запросы, используя оператор UNION или SYNCHRONIZE на уровне отчетов.

При наличии проблем с утверждениями Where, указанными в определении объекта, необходимо избегать их использования и по возможности создавать объекты условия, которые, когда используются правильно, могут избежать проблем с утверждениями Where с жестко заданным кодом.

❗ Примечание

Вне самоограниченных объединений не следует создавать объединения в объекте условия. Объединение в объекте условия приравнивается к созданию объединения в повторно

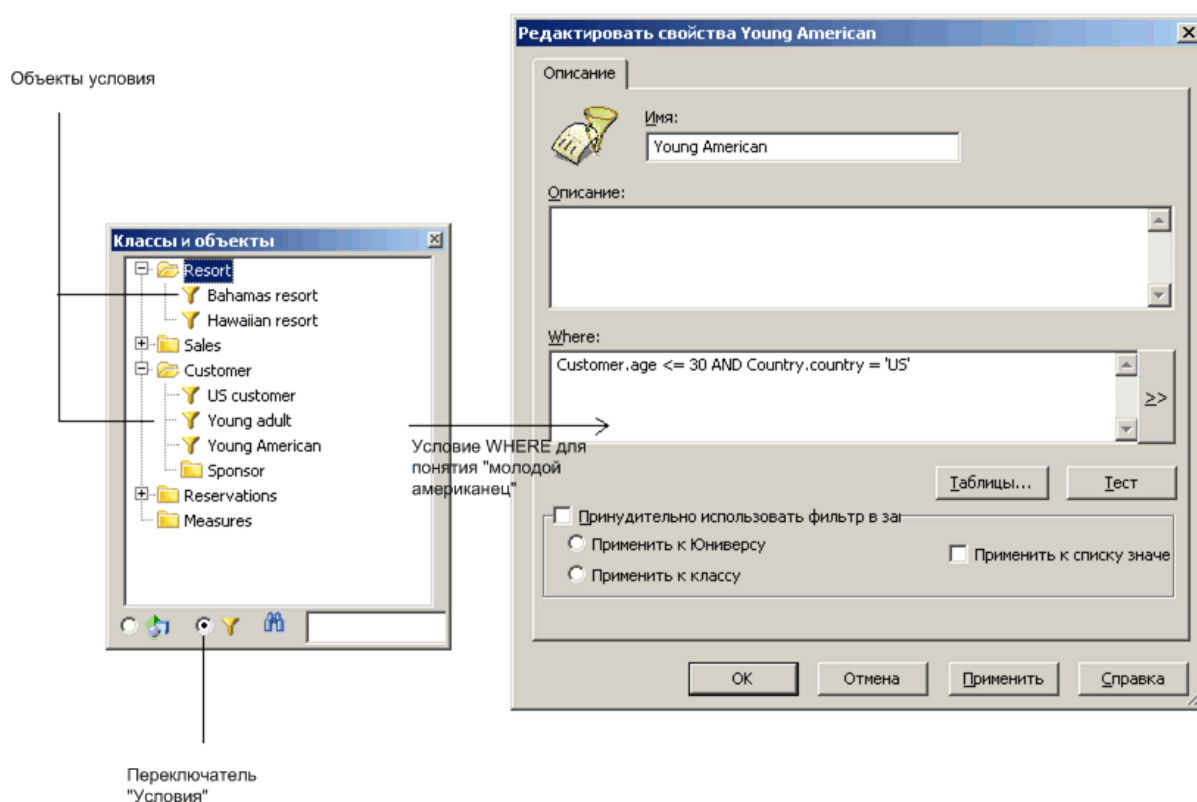
используемом утверждении Where, а значит не рассматривается функцией обнаружения контекстов (автоматическое определение контекста) или обнаружения несовместимости поддержки агрегирования. Следует убедиться, что все объединения видимы в панели структуры. Это обеспечивает доступность всех объединений средствам автоматического определения.

7.6.16 Определение условных объектов

Объект условия – это предопределенное утверждение Where, которое может быть вставлено в инструкцию Select, обозначенную объектами в панели запросов.

Объекты условия хранятся в поле "Условия" панели юниверса. Можно получить доступ к просмотру условий, выбрав переключатель "Условия" в правом нижнем углу панели юниверса.

Условные объекты для юниверса Beach и утверждения Where, вычисляемого условием Young American, показаны ниже.



7.6.16.1 Преимущества и ограничения использования объектов условия

Преимущества использования объектов условия следующие.

- Полезно для сложных или часто используемых условий.
- Предоставляет пользователям выбор в применении условия.
- Нет необходимости в нескольких объектах.
- Объекты условия не изменяют просмотр классов и объектов в панели юниверса.

❗ Примечание

Может понадобиться привести пользователей к использованию просмотра объектов условия панели Юниверса.

Единственным недостатком использования объектов условия является то, что может возникнуть необходимость ужесточить условия для пользователей, чтобы ограничить им доступ к части данных. В таком случае необходимо определить утверждение Where в определении объекта.

7.6.16.2 Объекты условия не устраняют конфликты между утверждениями Where

Использование объектов условия не решает проблему конфликтующих утверждений Where, возвращающих пустой набор данных. Если пользователь использует запрос, включающий два объекта условия, работающих с одними данными, два условия комбинируются с помощью оператора AND, следовательно ни одно условие не удовлетворяется, и данные не будут возвращены. Эта проблема решается пользователями на уровне отчета с помощью создания двух запросов, по одному для каждого объекта условия, и объединения созданных запросов.

7.6.16.3 Обязательные фильтры

Имеется два типа обязательных фильтров:

- **Юниверс:** обязательный фильтр юниверса не зависит от класса, к которому он принадлежит. Обязательный фильтр юниверса включается в запрос, независимо от объектов, включенных в запрос (измерений, мер и сведений).
Большинство переменных SAP Business Warehouse (BW) создаются как обязательные фильтры юниверса при создании юниверсов OLAP в SAP BW.
- **Класс:** обязательные фильтры класса появляются только в том случае, если в запросе используется элемент класса объекта.
Обязательные фильтры класса включаются, если пользователь выполняет следующие действия:
 - Добавляет объект (измерение, мера или сведения) в область [результатов панели запросов](#) в Web Intelligence.
 - Добавляет предустановленный [фильтр](#) юниверса в область фильтров [панели запросов](#), даже если в области результатов не было выбрано ни одного объекта, принадлежащего к этому классу.
 - Создает фильтр при помощи объекта (измерение, мера или сведения), принадлежащего к классу с обязательным фильтром.

Обязательный фильтр может иметь значения по умолчанию или ассоциироваться со списком значений.

Обязательный фильтр скрыт и не может быть выбран на [панели запросов](#) в Web Intelligence. Если в средстве создания юниверсов фильтр устанавливается в качестве обязательного в запросе, то он автоматически скрывается, и команда [Показать элементы](#) отключается. Если отменить обязательность фильтра, он перестает быть скрытым. Команда [Показать детали](#) включается.

Запрос конечного пользователя может содержать несколько обязательных фильтров. По умолчанию все обязательные фильтры объединяются в запросе при помощи оператора AND.

Все подклассы наследуют обязательные фильтры вышестоящего класса. Примечание:

- Объект (измерение, мера, сведения), ссылающийся на другой объект с использованием функции @SELECT, не наследует обязательный фильтр класса того объекта, на который он ссылается.
- Утверждение WHERE объекта, которое ссылается на утверждение WHERE другого объекта с использованием функции @WHERE, не наследует обязательный фильтр класса того объекта, на который ссылается.
- Предустановленный фильтр, ссылающийся на другой предустановленный фильтр или на утверждение WHERE объекта с использованием функции @WHERE, не наследует обязательный фильтр класса того объекта, на который ссылается.

Пример: Обязательный фильтр в юниверсе OLAP

Следующий фильтр (отображенный в коде XML) устанавливает подлинность кода, введенного пользователем в подсказке.

```
<FILTER KEY="[BCOMUSI]">
  <CONDITION OPERATORCONDITION="InList">
    <CONSTANT TECH_NAME=
      "@Prompt('CO_CODE Char User MultiSingle Man Def',
        'A', 'Company code\Lov[BCOMUSI]Base',
        multi,primary_key)"/>
    </CONDITION>
  </FILTER>
```

Связанные сведения

[Примеры обязательных фильтров \[страница 304\]](#)

[Обязательные фильтры и список значений \[страница 305\]](#)

7.6.16.3.1 Примеры обязательных фильтров

Следующие примеры демонстрируют, как можно использовать обязательные фильтры юниверсов:

Для проверки имени пользователя, указанного пользователем с помощью имени пользователя, содержащегося в таблице необходимо следующее:

```
1 = (Select 1 from Club.dbo.Login
```

```
where Login = @Variable('BOUSER')
AND Password = @Prompt('Password?', 'A', ,mono,free) )
```

Чтобы ограничить время использования юниверса с 9.00 до 18.00 необходимо следующее:

```
1 = (select 1
where datepart(HH,getdate()) between 9 and 18)
```

Далее следует пример обязательного фильтра класса:

Определенный в классе, содержащем Страну/Регион/Город/Потребителя, ограничьте запросы данных о продажах на определенный период. Запросите у пользователя данные об этом периоде.

```
Club.dbo.Customer.cust_id in
(Select cust_id from Club.dbo.Sales
where @Select(Sales\Year) in
@Prompt('Sales Periods?', 'A',
'Sales\Year', multi, constrained))
```

Связанные сведения

[Обязательные фильтры \[страница 303\]](#)

7.6.16.4 Обязательные фильтры и список значений

Обязательные фильтры могут быть ассоциированы со списком значений. Для присоединения списка значений следует однозначно указать параметры списка значений на странице свойств объекта, к которому будет применен фильтр.

Обязательный фильтр юниверса может быть ассоциирован с многоуровневым списком значений.

Обязательный фильтр класса может быть ассоциирован с многоуровневым списком значений, если хотя бы один объект соответствующего класса является составной частью многоуровневого списка значений. Это действительно даже в том случае, когда в многоуровневом списке значений сгруппированы объекты различных классов.

Рекомендации

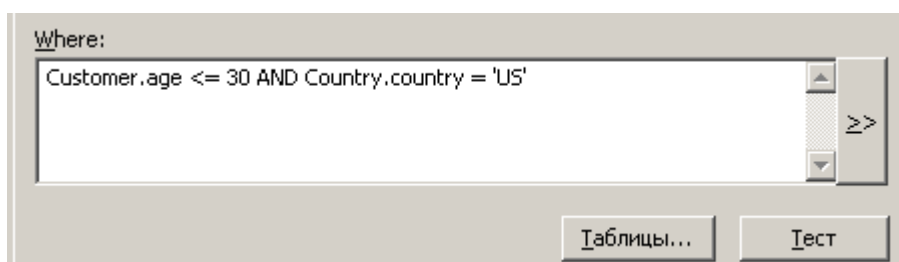
Обязательные фильтры следует создавать только на верхнем уровне многоуровневого списка значений.

Не следует ассоциировать многоуровневый список значений с обязательным фильтром, содержащим подсказку. Web Intelligence не поддерживает подсказки в многоуровневом списке значений.

7.6.16.5 Создание объекта условия

Как создать объект условия

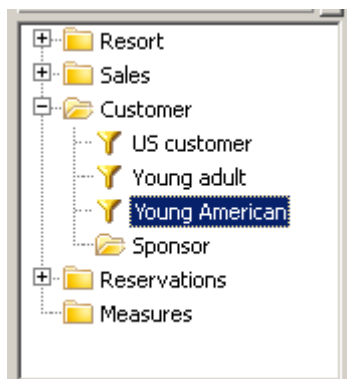
1. Выберите переключатель "[Условие](#)" в правом нижнем углу панели [Юниверса](#).
Отобразится представление [условий](#) панели [Юниверс](#). В нем содержится дерево обзора всех классов юниверса.
2. Щелкните правой кнопкой мыши класс и выберите [Вставить условие](#) в контекстном меню.
Или
Выберите класс и нажмите кнопку [Вставить условие](#).
Откроется диалоговое окно [Редактировать свойства](#). В диалоговом окне [Имя](#) отобразится имя по умолчанию. Окно [Where](#) пустое.
3. Введите имя условия.
4. Введите синтаксис утверждения Where в диалоговое окно утверждения [Where](#).
Или
Нажмите кнопку ">>" рядом с окном утверждения [Where](#) для открытия редактора утверждений [Where](#).
5. Дважды щелкните мышью столбцы, объекты, операторов или функции в [структурах SQL](#) и списках [свойств](#).
6. Нажмите [OK](#) для закрытия редактора.
Определение для условия Young American отображено ниже. Это ограничивает возвращенные американским потребителям значения до возраста 30 лет или меньше.



7. Чтобы проверить синтаксис, нажмите [Разбор](#).
8. Чтобы определить фильтр как обязательный, установите флажок "[Вспомогательный фильтр](#)".
По умолчанию обязательный фильтр применяется к классу и не применяется к списку значений.
9. Выберите переключатель, чтобы определить обязательный фильтр как применяемый к классу или юниверсу.
10. Чтобы применить обязательный фильтр к списку значений, установите флажок "[Применить к списку значений](#)".
11. Нажмите кнопку ["OK"](#).
Отобразится новый объект условия в поле [условий](#) панели [Юниверса](#).

❗ Примечание

Чтобы отредактировать объект условия, используя редактор предварительно определенных фильтров, нажмите кнопку >>.



Связанные сведения

[Обязательные фильтры \[страница 303\]](#)

[Обязательные фильтры и список значений \[страница 305\]](#)

[Предопределенные условия в юниверсах OLAP \[страница 458\]](#)

[Дополнительные подсказки в юниверсах OLAP \[страница 463\]](#)

7.6.16.6 Использование объектов условия в одном запросе

При наличии двух объектов условия, определенных для одного объекта и используемых в одном запросе, данные не будут возвращены, так как оба утверждения WHERE создают неверное условие. Когда это возможно, следует избегать утверждений WHERE с жестко заданным кодом в определении объекта, но также, когда используются объекты условия, пользователи должны помнить о потенциальных проблемах.

Пользователи могут решить проблемы возврата пустых наборов данных посредством объединения двух запросов, один запрос для каждого объекта условия.

📌 Примечание

Чтобы избежать пользователей Web Intelligence, использующих два объекта условия в одном запросе, можно включить описание для объекта условия "X", где указано, что он должен использоваться с объектом "Y".

7.6.16.7 Почему несколько утверждений Where возвращают пустой набор данных?

При добавлении утверждения Where к определению объекта, ограничение добавляется к набору ограничений, установленному объединениями с помощью оператора AND. При комбинировании в

запросе двух объектов, применяющих ограничение к одному набору данных, оба утверждения Where комбинируются в последовательных утверждениях AND. Результатом подобного запроса является то, что данные не удовлетворят ни одно из условий и не будут возвращены.

Например, пользователь хочет узнать, какие услуги доступны на гостиничных курортах багамского и гавайского клубов. Следующий запрос используется вместе с объектами условия для курорта на Багамах и Гавайях:

SQL для этого запроса является следующим:

```
SELECT Service.service, Resort.resort FROM Service, Resort,  
Service_Line WHERE ( Resort.resort_id=Service_Line.resort_id ) AND  
( Service.sl_id=Service_Line.sl_id ) AND ( ( Resort.resort = 'Bahamas  
Beach' ) AND ( Resort.resort = 'Hawaiian Club' ) )
```

Ограничения двух утверждений Where комбинируются в утверждениях AND в конце утверждения Where.

Когда запрос запущен, два ограничения на страну не могут быть выполнены в одном запросе, следовательно данные не будут возвращены. Окно сообщения отображается, сообщая об отсутствии данных, которые могут быть получены.

Создание двух запросов для комбинирования ограничений

Проблему использования двух объектов условия в одном запросе пользователи могут решить с помощью двух запросов, по одному для каждого утверждения Where, и оператора UNION для объединения результатов.

7.6.17 Использование самоограниченных объединений для применения ограничений.

Можно использовать самоограниченные объединения, чтобы ограничивать данные тем или иным столбцом в таблице на основе метки, которая служит для переключения между столбцами. Флаг является третьим столбцом, значения которого определяют, какой из двух столбцов используется в запросе.

Дополнительные сведения о создании и использовании самоограниченных объединений см. в разделе [Самоограниченные объединения \[страница 176\]](#).

7.6.18 Применение ограничения путем логического вывода нескольких таблиц

Можно ограничить данные, возвращенные для объекта данными из таблицы, вычисленной объектом, которые согласуются с данными в другой таблице.

Например, объект "Страна происхождения" вычисляет таблицу стран. Объект "Страна происхождения" выдает следующие данные:

Country of origin
Australia
France
Germany
Holland
Japan
UK
US

Если требуется использовать объект "Страна происхождения" в классе "Продавец", то будут возвращены только страны, в которых находятся продавцы, в этом случае можно переименовать объект в "Страны продавцов" и создать ограничение в таблице "Страны" для возвращения только значений стран продавцов из таблицы "Продавцы".

Объект стран продавцов обладает следующим SQL:

```
SELECT Country.country FROM Country,
Sales_Person, Customer, City, Region WHERE
( City.city_id=Customer.city_id ) AND ( City.region_id=Region.region_id )
AND ( Country.country_id=Region.country_id ) AND
( Sales_Person.sales_id=Customer.sales_id )
```

Объект стран продавцов выдает следующие данные:

Sales people countries
France
Germany
Japan
UK
US

Если применяется ограничение через указание объекта "Страна", используемого в запросе, то таблица "Продавцы" также должна быть обозначена в утверждении From инструкции Select.

Тогда Страна в классе "Продавцы" возвращает те страны, в которых находятся продавцы. Чтобы применить ограничение, нажмите кнопку "Таблицы" в таблице определений объекта.

Таблица "Страны" должна объединяться с таблицей "Продавцы" через посредничество только равных объединений.

📘 Примечание

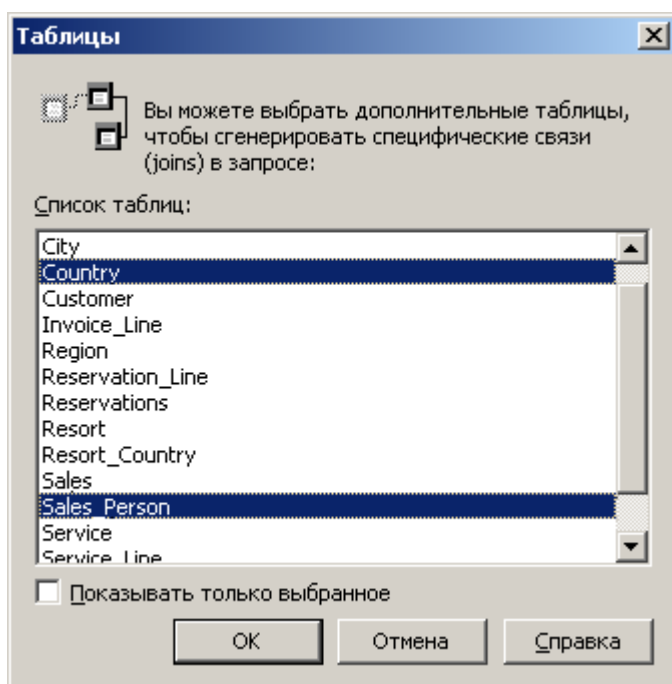
Внося любые изменения в SQL для объекта, который имеет ограничение таблицы, определенное в инструкции Select, средство создания юниверсов автоматически переопределяет, какие таблицы

требуются инструкции Select и утверждению Where объекта. Не будет получено уведомление, если ограничение таблицы аннулировано в таблице, вычисленной объектом.

7.6.18.1 Предположение нескольких таблиц для применения к условию

Для вычисления нескольких таблиц, устанавливающих условие для объекта

1. Дважды щелкните объект.
Появится диалоговое окно изменения свойств объекта.
2. Нажмите кнопку "Таблицы".
Откроется список таблиц универса.
3. Выберите одну или несколько таблиц, которые требуется вычислить с помощью объекта, в дополнение к текущей таблице. Можно выбрать несколько таблиц, удерживая клавишу CTRL и щелкая имена таблиц в списке. Таблицы "Страна" и "Sales_Person" выбраны ниже:



4. Нажмите "OK" в каждом диалоговом окне.
5. Выполните запросы в Web Intelligence, чтобы протестировать ограничение таблиц.

7.6.18.2 Когда используется каждый метод для применения ограничения?

Для установления ограничений в универсе можно использовать следующие руководства:

- Избегайте использования утверждений Where в определениях объектов. Если необходимо использовать утверждение Where, следует помнить о возможных проблемах при использовании нескольких объектов и конфликтующих утверждениях Where.
- Используйте объекты условия, при необходимости оказания пользователям поддержки предоставляя дополнительные предопределенные условия, избегая многочисленных объектов и изменений в просмотре объектов и классов панели Юниверса.
- Используйте самоограниченные объединения, чтобы применить ограничения к таблицам, когда необходимо применить ограничение независимо от того, где таблица используется в SQL. Данный метод идеален, когда таблица использует метку для переключения между двумя и более доменами.
- Используйте дополнительные объединения, в случае когда таблица поиска служит более чем одной цели в юниверсе.

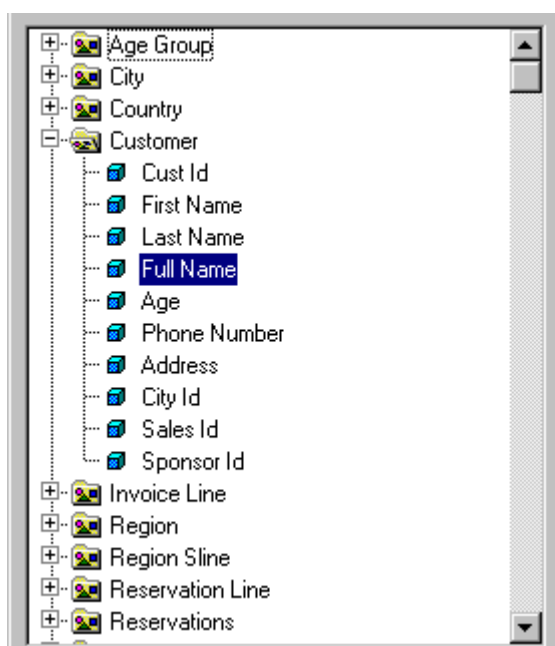
7.6.19 Объединенные объекты

Объединенный объект является комбинацией двух существующих объектов. Например, можно создать объект "Полное имя", который будет являться объединением объектов "Фамилия" и "Имя" в класса "Потребитель".

7.6.19.1 Создание объединенного объекта

Чтобы создать объединенный объект, выполните следующие действия:

1. Создайте объект.
Например, создайте новый объект "Полное имя" в классе "Клиенты". Следует так же ввести описание для объектов таких как "Данный объект является объединением имени и фамилии клиента."



2. Дважды щелкните объект.
Откроется диалоговое окно "Редактировать свойства"..
3. Введите синтаксис для объединенного объекта в поле "Выбрать".
Например, введите следующий синтаксис для объекта "Полное имя"(синтаксис MS Access):
`rtrim (Customer.first_name + ' ' + Customer.last_name)`
Where rtrim является функцией, которая удаляет пробелы в конце символьной строки и использует две кавычки для вставки пробела между именем и фамилией

Описание | Свойства | Дополнительно | Ключи | Исходные сведения

Имя: Тип:

Описание:

Select:

Where:

Таблицы... Тест

📌 Примечание

Чтобы открыть "Редактор SQL", можно также нажать кнопку "Редактировать". Можно использовать в редакторе графические инструменты, для помощи в указании синтаксиса SQL для объекта. Дополнительные сведения о данном редакторе, см. раздел "Разработка схемы".

4. В каждом из диалоговых окон нажмите кнопку ОК.
При выполнении запроса для объекта "Полное имя" возвращаются полные имена, перечисленные в алфавитном порядке имен.

7.7 Определение иерархий

Создание иерархий объектов позволяет пользователям выполнить многомерный анализ.

7.7.1 Что такое многомерный анализ?

Многомерный анализ – это анализ объектов измерения установленный в значимой иерархии.

Многомерный анализ позволяет пользователям следить за данными различными точками зрения. Это позволяет им распознавать тенденции и исключения в данных.

Иерархия является упорядоченной серией соответствующих измерений. Примером иерархии является География, которая может группировать измерения таких как Страна, Регион и Город.

Web Intelligence может использовать свертывание и развертку иерархии, чтобы произвести многомерный анализ.

7.7.1.1 Детализация

Пользователи могут использовать переход по иерархии, чтобы перемещаться по уровням иерархии детализации. Пользователи могут вернуть или выполнить развертку иерархии.

Например, руководителю со временем необходимо вычислить дату записи. Поскольку это дизайнер юниверса, то можно установить иерархию времени Бронирование, включая измерения Год бронирования, Квартал бронирования, Месяц бронирования и Дата бронирования.

От высокого уровня агрегирования, например Квартал бронирования, руководитель может выполнить развертку до более подробного уровня например, Месяц бронирования или Бронирование Даты. Он также может выполнить свертывание иерархии от Квартал бронирования до Год бронирования, чтобы просмотреть итоговый просмотр данных.

7.7.2 Определение иерархий

Иерархии могут принимать различные формы. Примеры классических иерархий включают.

- География. Континент Страна Регион Город
- Продукты. Категория Бренд Продукт
- Время. Год Квартал Месяц Неделя День

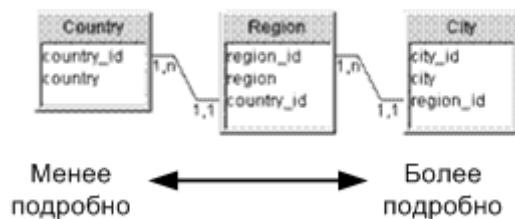
Иерархия может быть "смешанной".

География/продукты. Континент Страна Категория Бренд Продукт

Иерархии, подразумеваемые в данных, зависят от характера данных и способа хранения в базе данных. Необходимо осторожно анализировать данные, чтобы найти иерархии в системе, которая лучше всего отвечает требованиям группы пользователей.

Если нет точных правил для определения, где хранятся иерархии в данных, отношения "один ко многим" (1-N), встроенные в данные, могут означать существование иерархий.

В схеме, указанной ниже, отношения "один ко многим" между таблицами подразумевают географическую иерархию.



7.7.3 Установка иерархий

По умолчанию в средстве создания юниверсов представлен набор стандартных иерархий для многомерного анализа. В нем присутствуют классы и объекты, расположенные в соответствии с тем порядком, в каком они отображаются на панели юниверса. При создании объектов их необходимо упорядочить в иерархическом порядке, чтобы убедиться, что иерархии по умолчанию важны для пользователя.

Во многих случаях требуется создание настроенных иерархий, которые включают в себя объекты различных классов. Для этого создается новая иерархия.

В [редакторе иерархий](#) можно просматривать и создавать новые иерархии по умолчанию. Данный графический редактор позволяет управлять иерархиями в юниверсе.

❗ Примечание

Если определяется пользовательская иерархия, иерархии по умолчанию становятся неактивными и недоступными конечному пользователю. Если нужно сделать их активными, необходимо явно выбрать их в [редакторе иерархий](#) и добавить в список пользовательских иерархий.

7.7.3.1 Просмотр иерархий

Можно просмотреть иерархии следующим способом.

7.7.3.1.1 Чтобы просмотреть иерархии в юниверсе, выполните следующие действия.

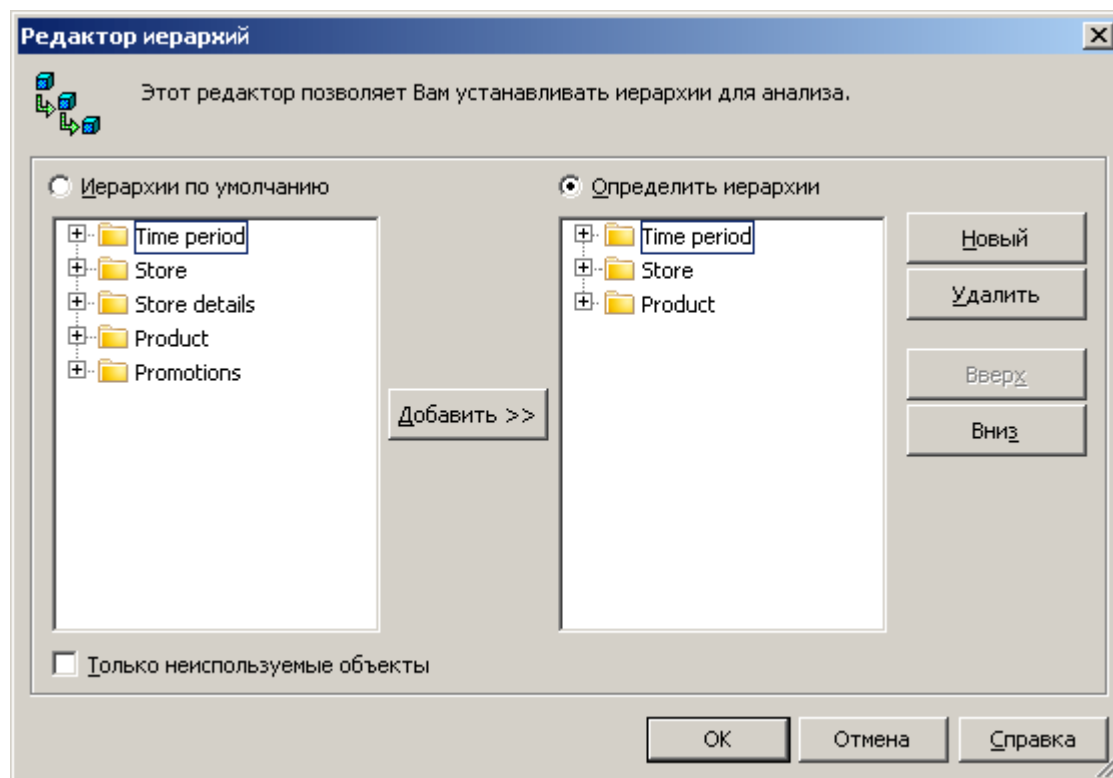
1. Выберите "Инструменты" > "Иерархии".

Или

Нажмите кнопку "Иерархии".

Откроется редактор иерархий. В средстве создания юниверсов иерархии представляются с помощью символа папки, а измерения – с помощью символа куба.

На левой панели выводится список всех классов, содержащих объекты измерений в активном юниверсе. На правой панели отображаются настроенные иерархии, созданные пользователем.



2. Щелкните узел иерархии (значок "+"), чтобы открыть иерархически упорядоченные измерения.
3. Нажмите "Отмена".

7.7.3.2 Установка иерархий

Для создания новой иерархии необходимо создать на панели пользовательских иерархий новую папку, затем добавить соответствующие измерения в иерархическом порядке.

Для удаления иерархии или измерения выберите их и нажмите кнопку "Удалить".

7.7.3.2.1 Чтобы создать новую иерархию,

1. Нажмите кнопку "Создать" в редакторе иерархий.

Или

В редакторе иерархий выберите класс на левой панели и перетащите его на правую панель.

Папка, обозначающая иерархию, отобразится на правой панели.

2. Введите имя иерархии.
3. Нажмите "RETURN" для подтверждения имени.
4. Выберите новую иерархию.

Иерархия выделена.

5. Разверните узел иерархии по умолчанию на левой панели.

Данная иерархия содержит измерения, которые необходимо добавить в новую пользовательскую иерархию.

6. Щелкните измерение. Чтобы выбрать серии измерений, щелкните каждое из них, удерживая клавишу CTRL.

Одно или несколько измерений выделены.

7. Нажмите кнопку "Добавить".

На левой панели под выбранной иерархией отобразится одно или несколько измерений.

Примечание

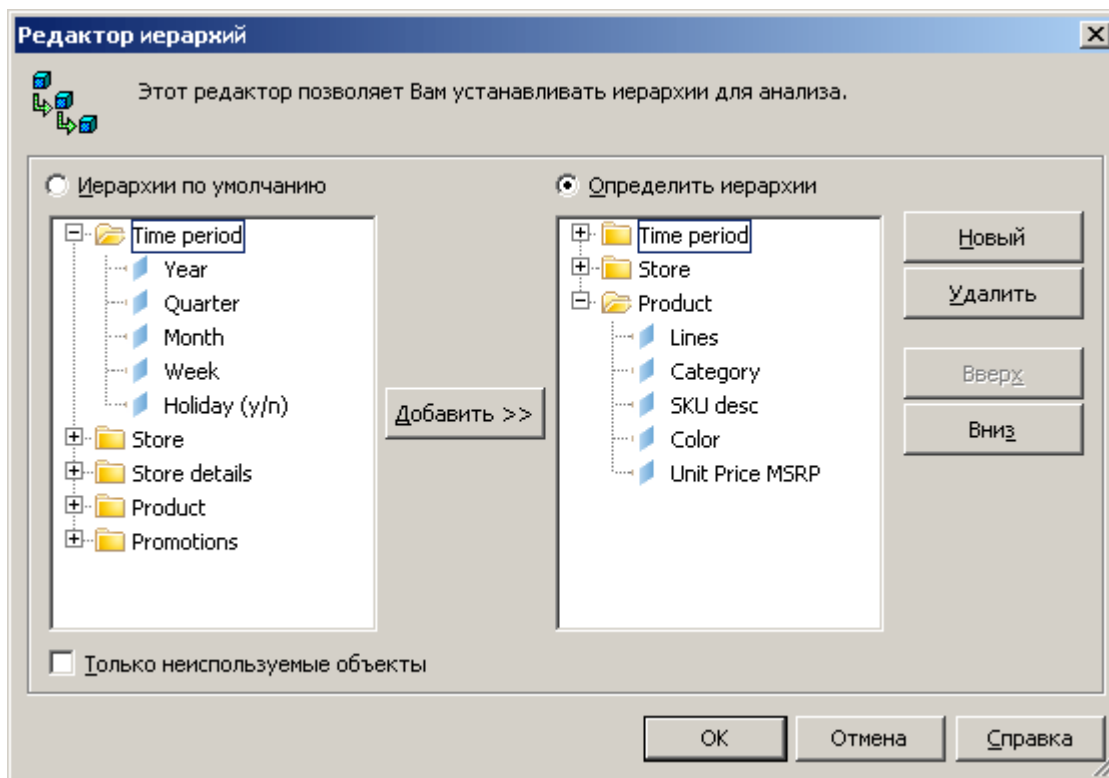
Чтобы просмотреть измерения, которые еще не выбраны для включения в иерархию, установите флажок "Только неиспользуемые объекты".

7.7.3.3 Изменение порядка измерений и иерархий

Порядок отображения объектов измерений внутри иерархии можно изменить. Чтобы передвинуть объект, щелкните его и затем нажимайте кнопки "Вверх" и "Вниз". Также можно изменить порядок иерархий.

Объект измерения или иерархию также можно переместить, перетаскив его мышью.

Ниже приведены примеры иерархий и объектов измерений.



В редакторе иерархий, указанном выше, установлены три настроенные иерархии: период времени, магазин и продукты. В иерархии продуктов содержатся следующие измерения: строки, категория, нисходящая SKU, цвет и цена MSRP за единицу.

7.8 Использование каскадных списков значений для иерархий

Пользовательскую иерархию или иерархию по умолчанию можно связать с каскадными списками значений.

Примечание

Список значений (LOV) содержит значения данных, связанные с объектом. Списки значений подробно описаны в разделе [Использование списков значений \[страница 321\]](#).

Каскадный список значений – это последовательность списков значений, связанных с иерархией в юниверсе. Для каждого уровня иерархии установлены подсказки для ввода списков значения данного уровня.

При обновлении отчета с иерархией, связанной с каскадным списком значений, отобразится иерархия и будет предложено выбрать уровень и одно или несколько значений в соответствующем списке перед выполнением запроса.

Например, квартал предварительных заказов, связан с иерархией года. При использовании резервирования квартала по месяцу в запросе отобразится годовая иерархия и будет предложено выбрать год соответствующего квартала перед выполнением запроса.

7.8.1 Создание каскадного списка значений

Можно создать каскадный список значений иерархии по умолчанию или пользовательской иерархии. Для каждого уровня создается файл .LOV. При выполнении запроса возвращается только список значений для запрашиваемого уровня иерархии.

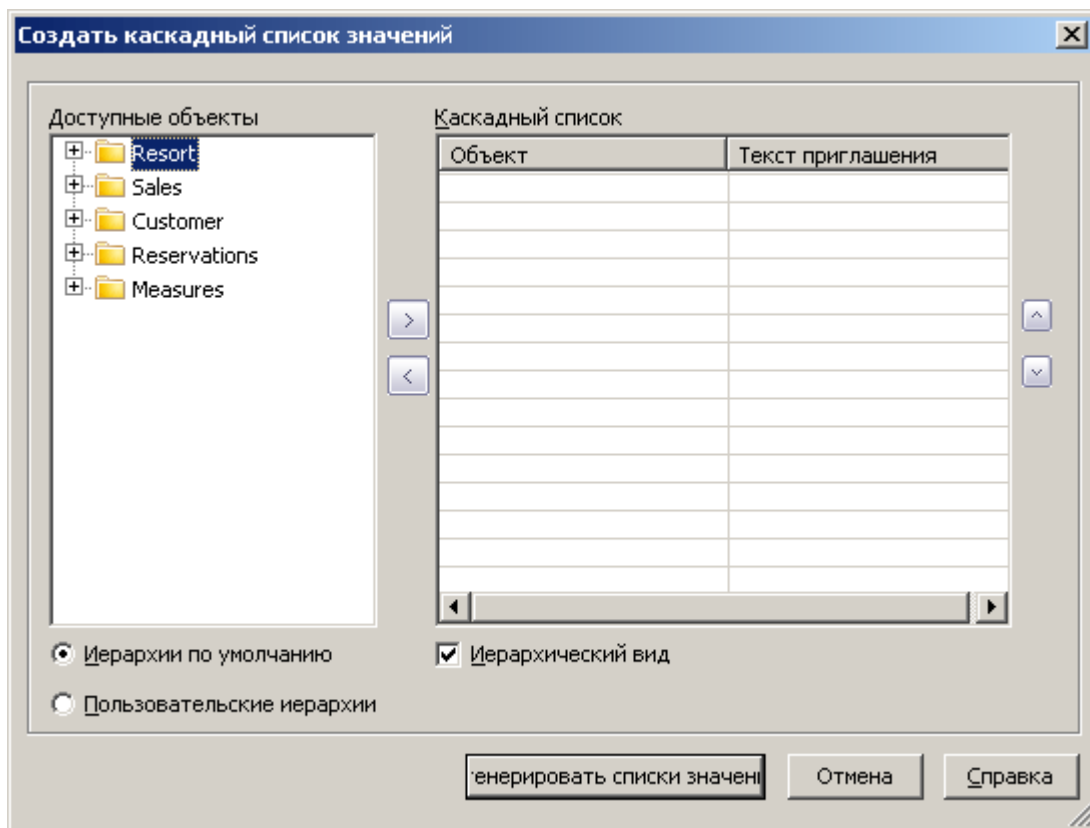
📘 Примечание

Данное итеративное использование списка значений для иерархии отличается от создания иерархии для списка значений, в котором все списки значений для всех уровней иерархии возвращаются в микрокубе. При использовании каскадных списков значений файл LOV не будет возвращен в микрокубе, пока не будет выполнена подсказка для уровня. При этом для уровня возвращается только LOV.

7.8.1.1 Чтобы создать каскадный список значений,

1. Выберите ► [Инструменты](#) ► [Списки значений](#) ► [Создать каскадные списки значений](#) ►.

Откроется диалоговое окно [Создать каскадные списки значений](#).



Доступны следующие параметры.

Параметр каскадного списка значений	Описание
Иерархии по умолчанию	Если одна из иерархий выбрана, соответствующая иерархия по умолчанию или пользовательская иерархия, определенная в юниверсе, отобразится на панели Доступные . Дополнительные сведения о данных типах иерархии см. в разделе Установка иерархий [страница 314].
Пользовательские иерархии	
Иерархический вид	Выбранная иерархия отобразится в виде дерева на Панели запросов . Это облегчит навигацию по иерархии. Щелкнув уровень, список значений отобразится на панели, справа от Панели запросов .
Объект	Уровень иерархии для измерения.
Текст подсказки	Текст, который отображается в запросе для списка значений уровня.

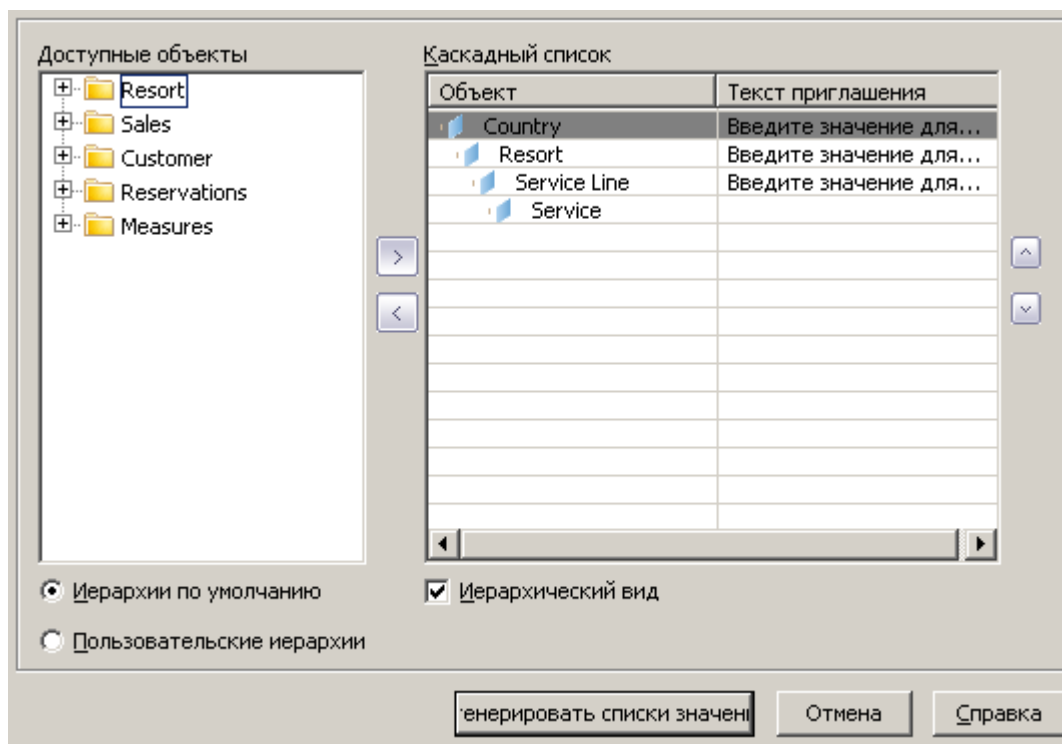
- Нажмите переключатель [Иерархии по умолчанию](#) или [Пользовательские иерархии](#).
Отобразится соответствующий список иерархий в юниверсе.
- Выберите или разверните класс и выделите один или несколько объектов.

4. Нажмите правую стрелку.

Все объекты данного класса отобразятся в списке объектов.

Или

Выбранные объекты отобразятся в списке [Объект](#).



5. Введите текст запроса для каждого объекта.

6. Если необходимо отобразить положение объекта в [Каскадном списке значений](#), щелкните объект и используйте стрелки "вверх" и "вниз", чтобы передвигаться по списку.

Чтобы удалить объект, выберите его и нажмите левую стрелку.

7. Установите или снимите флажок [Иерархический вид](#).

8. Выберите [Сгенерировать списки значений](#).

Откроется диалоговое окно [Создать каскадный список значений](#). Список значений создается для каждого уровня каскадных списков значений. Каждый файл .LOV сохранен во вложенной папке юниверса в файловой системе, например C:\Documents and Settings\<user>\Application Data\Business Objects\Business Objects 12.0\Universes\<CMS name>\beachXI3.0\.

Примечание

Сведения о редактировании, экспорте в CMS и создании списков значений для автономных объектов см. в разделе [Использование списков значений \[страница 321\]](#).

7.9 Использование списков значений

Список значений – это список, в котором содержатся значения данных, связанные с объектом. В списке значений могут быть данные из двух типов источника данных.

Источник данных списка значений	Описание
Файл базы данных	<p>При создании объекта в средстве создания юниверсов список значений автоматически сопоставляется с объектом. Список значений не создается, пока пользователь или дизайнер не выберет отобразить список значений для объекта на панели запросов. Запрос "ВЫБРАТЬ ОТДЕЛЬНО" выполняется в столбце или столбцах, выведенных с помощью объекта.</p> <p>Возвращенные данные хранятся в файле с расширением .LOV во вложенной папке юниверса, созданной на основе той же папки, которая хранит файл юниверса. Файл .LOV используется в качестве источника значений списка.</p>
Внешний файл	<p>Персональные данные, например текстовый файл или файл Excel, могут быть связаны со списком значений.</p> <p>Список значений на основе внешнего файла, является фиксированным. Динамическую связь с внешним файлом получить невозможно. Необходимо обновить файл .LOV, если внешний файл изменен.</p>

7.9.1 Как использовать список значений?

В Web Intelligence пользователь может создавать запросы на [Панели запросов](#) с помощью операнда [Отобразить список значений](#), чтобы применить к объекту во время применения условия.

❗ Примечание

Файл .LOV создается вне зависимости от применения условия к объекту на [Панели запросов](#), который требует ограничения для значений столбцов, выведенных с помощью объекта.

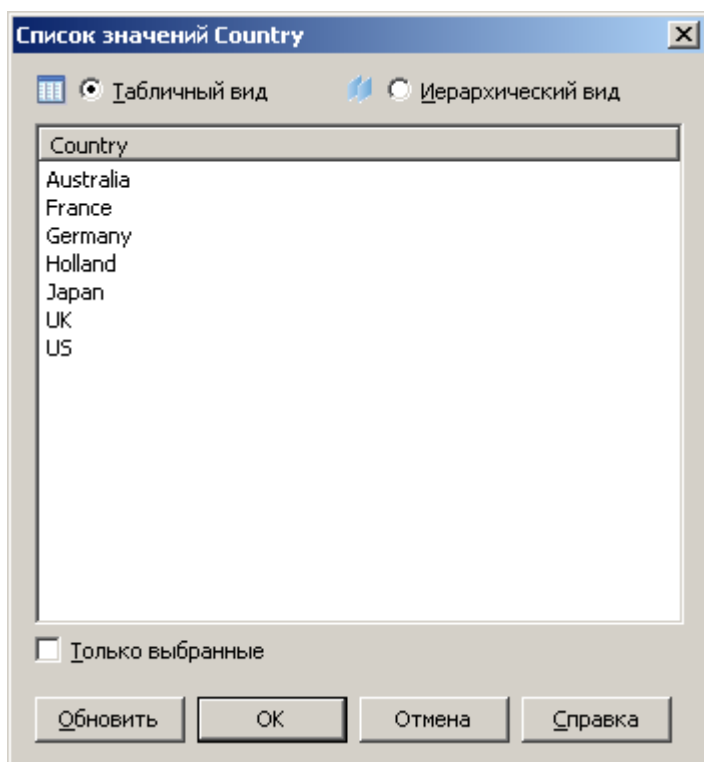
Список значений для объекта отображается со значениями, доступными для объекта, что позволяет пользователю выбирать сроки для условия. При первом использовании списка значений он сохраняется в качестве файла .LOV во вложенной папке юниверса в файловой системе. Для данного объекта запрос "ВЫБРАТЬ ОТДЕЛЬНО" выполняется только один раз.

В этой папке также хранятся файлы с расширением LOV, созданные в средстве создания юниверсов и используемые для ограничения списка значений, возвращаемых для объектов, доступом которых к данным хочет управлять разработчик.

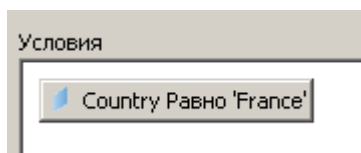
Пример: Использование списка значений для объекта "Страна"

Определение объекта "Страна" в утверждении Select выглядит следующим образом: COUNTRY.COUNTRY_NAME. В списке значений по умолчанию, связанным с объектом, содержатся все названия отдаленных стран в столбце COUNTRY_NAME. Данный список возвращается, если объект "Страна" используется в условии запроса.

Если пользователю необходимо ограничить значения в запросе только для объекта "Франция", он может выбрать данный объект в списке, в котором отображены все значения страны в таблице "Страна" для условия.



Выбрав объект "Франция", условие на панели *Условия Панели запросов* выглядит следующим образом.



Запрос возвращает значения только для объекта "Франция".

7.9.2 Определение способа использования списка значений по отношению к объекту

Когда в средстве создания юниверсов создается объект измерения или описания, ему автоматически присваивается связанный список значений (LOV). При создании объекта этот список физически не существует, но по умолчанию у объекта есть возможность осуществлять запрос в базу данных для получения списка его значений, когда он используется на панели запросов.

📌 Примечание

Объектам мер список значений по умолчанию не присваивается.

Когда условие в первый раз помещается в объект на панели запросов, для которого требуется выводить список значений в средстве создания юниверсов, по отношению к соответствующим столбцам выполняется инструкция `SELECT DISTINCT`, выведенная объектом, и возвращается список значений.

Чтобы сохранить список значений, во вложенной папке юниверса создается файл `.LOV`. В следующий раз, когда объекту в средстве создания юниверсов потребуется список значений, значения будут возвращаться из файла с расширением `LOV`, а не из базы данных.

7.9.2.1 Роль дизайнера (разработчика) в управлении списками значений

Дизайнер юниверса определяет, как представляются данные в списке, и устанавливает ограничения на количество и тип данных, возвращаемых в список.

Можно задать свойства объекта для того, чтобы определить следующие действия для списка значений.

- Если список значений связывается с объектом.
- Когда обновляется список.
- Определите запрос, который устанавливает условия для запроса `SELECT DISTINCT`, который используется объектом при возвращении списка значений. Этот запрос сохраняется в свойствах объекта.
- Выведите значения списка в виде простого списка или иерархии объектов.
- Если список основан на значениях столбца или значениях из внешнего файла, например, электронной таблицы Excel.

Можно также создать постоянный список значений объекта и экспортировать его в репозиторий. Этот файл `.LOV` всегда будет использоваться в качестве списка значений этого объекта. Он не обновляется.

7.9.3 Свойства и параметры списка значений

Можно определить следующие свойства объекта, которые позволяют контролировать использование списка значений для объекта в Web Intelligence.

Свойство	Описание
Связать со списком значений	<ul style="list-style-type: none"> • Когда оно выбрано, позволяет списку значений быть связанным с объектом. Оно является выбранным по умолчанию. • Когда удалено, списки значений не могут связываться с объектом. • Выбрано по умолчанию для измерений и подробных сведений. Не выбрано для измерений.
Имя списка	Имя файла .LOV, содержащего возвращенные данные списка. Ограничено до 8 символов.
Позволить пользователям редактировать этот список значений	<ul style="list-style-type: none"> • Если выбран этот параметр, пользователи могут редактировать файл списка значений в Web Intelligence. • Когда удалено, пользователь не может редактировать список. <div> <p>Примечание</p> <p>Это не применимо к файлам персональных данных, например таблицам Excel. Они не могут экспортироваться в репозиторий. Они остаются на локальном ПК. Пользователь может редактировать локальные файлы или изменять целевой список значений для другого локального файла данных.</p> </div> <p>Целью списка значений обычно является ограничение набора доступных пользователю значений. Если невозможно редактировать список, пропадает возможность контролировать выбор пользователями значений. Обычно, если файл с личными данными не используется как источник списка значений, этот параметр отменяется, для уверенности в том, что пользователи не редактируют списки значений.</p>
Автоматическое объединение перед использованием (только BusinessObjects)	<ul style="list-style-type: none"> • Если выбрано, данные списка обновляются каждый раз, когда список значений объекта отображается на панели запросов. Можно повлиять на производительность при обновлении .LOV. Данный параметр не применяется для отчетов Web Intelligence. • Если создано, список обновляется только один раз в начале сессии пользователя при входе в систему. Если в списке содержатся значения, которые регулярно изменяются, необходимо выбрать данный параметр, но необходимо принять во внимание влияние на производительность. Если содержимое списка не изменяется, необходимо удалить данный параметр.
Иерархический вид	Выберите свойство "Показ иерархии", чтобы отобразить каскадный список значений в качестве иерархии в Web Intelligence.

Свойство	Описание						
Экспорт с помощью юниверса	<ul style="list-style-type: none"> Если выбрано, файл .LOV, связанный с объектом, экспортируется с помощью юниверса в репозиторий. Необходимо создать список значений для экспорта, связанный с объектом. Список сохраняется в виде файла .LOV. Если удалено, файл .LOV для объекта не экспортируется в репозиторий. Выберите данный параметр, если необходимо настраивать данный список регулярно. Это позволяет регулярно осуществлять экспорт и импорт изменений с помощью юниверса. 						
Делегировать поиск	<p>Выберите свойство "Делегировать поиск", чтобы позволить пользователям Web Intelligence ограничивать количество значений, возвращенных в список значений. Если свойство "Делегировать поиск" выбрано, в Web Intelligence представлено пустое поле списка значений для пользователя во время выполнения запроса. Пользователь вводит значения, чтобы определить критерии поиска для фильтрации списка значений.</p> <p>Большинство источников данных поддерживают одиночные символы, чтобы упростить поиск в базе данных. В Web Intelligence поддерживаются следующие подстановочные символы:</p> <table> <tr> <td>*</td><td>Сопоставляет количество символов, даже символы 0</td></tr> <tr> <td>?</td><td>Сопоставляет только один символ</td></tr> <tr> <td>\</td><td>Переходит на другой символ, позволяя поиск одиночного символа.</td></tr> </table> <p>В параметре "Делегированный поиск" есть следующие ограничения.</p> <p>Делегированный поиск не поддерживается каскадным списком значений.</p> <p>Делегированный поиск можно активировать только для списка значений на объекты, которые являются символами типа.</p> <p>Делегированный поиск невозможно активировать, если пользовательский SQL введен в список значений.</p> <p>Делегированный поиск невозможно активировать, если параметр "Иерархический вид" используется для списка значений.</p>	*	Сопоставляет количество символов, даже символы 0	?	Сопоставляет только один символ	\	Переходит на другой символ, позволяя поиск одиночного символа.
*	Сопоставляет количество символов, даже символы 0						
?	Сопоставляет только один символ						
\	Переходит на другой символ, позволяя поиск одиночного символа.						

Можно редактировать, отображать или назначать имя по умолчанию для списка значений, щелкнув мышью следующие кнопки.

Параметр	Описание
Восстановить значения по умолчанию	Восстанавливает имя по умолчанию, назначенное для файла .LOV во время создания объекта.

Параметр	Описание
Редактировать	Позволяет редактировать значения в списке. Можно использовать редактор, чтобы ограничивать значения в списке во время использования на панели запросов.
Отобразить	Отображает список значений объекта. Если необходимо создать постоянный список для экспорта в юниверс репозитория, нажмите "Отобразить", чтобы создать файл .LOV. Потом можно редактировать файл.

7.9.3.1 Определение свойств и параметров для списка значений

Чтобы определить свойства и параметры для файла (.LOV) списка значений,

1. Дважды щелкните объект.
На странице определений откроется диалоговое окно "Изменить свойства".
2. Выберите вкладку "Свойства".
Откроется страница свойств.
3. Установите или снимите флажки в групповом окне списка значений внизу страницы.
4. Введите имя соответствующего файла .LOV в поле "Имя списка".
5. Нажмите кнопку "Редактировать", если необходимо определить ограничения в списке значений.
6. Используйте панель запросов, чтобы создать запрос на основе данных списка.
7. Нажмите "Отобразить", чтобы просмотреть список значений.
Нажав данную кнопку, начинается процесс выполнения запроса "ВЫБРАТЬ ОТДЕЛЬНО" в столбцах, выведенных с помощью объекта в базе данных. Идентичный способ используется в продуктах отчета для создания файла .LOV объекта.
8. Нажмите кнопку "ОК".

7.9.3.2 Просмотр списка значений, связанного с объектом

В средстве создания юниверсов можно просматривать список значений, связанный с объектом. При просмотре списка значений в каталоге "User Docs" автоматически создается стандартный файл .LOV, в котором хранятся возвращаемые данные. По умолчанию при просмотре списка значений автоматически создается файл .LOV.

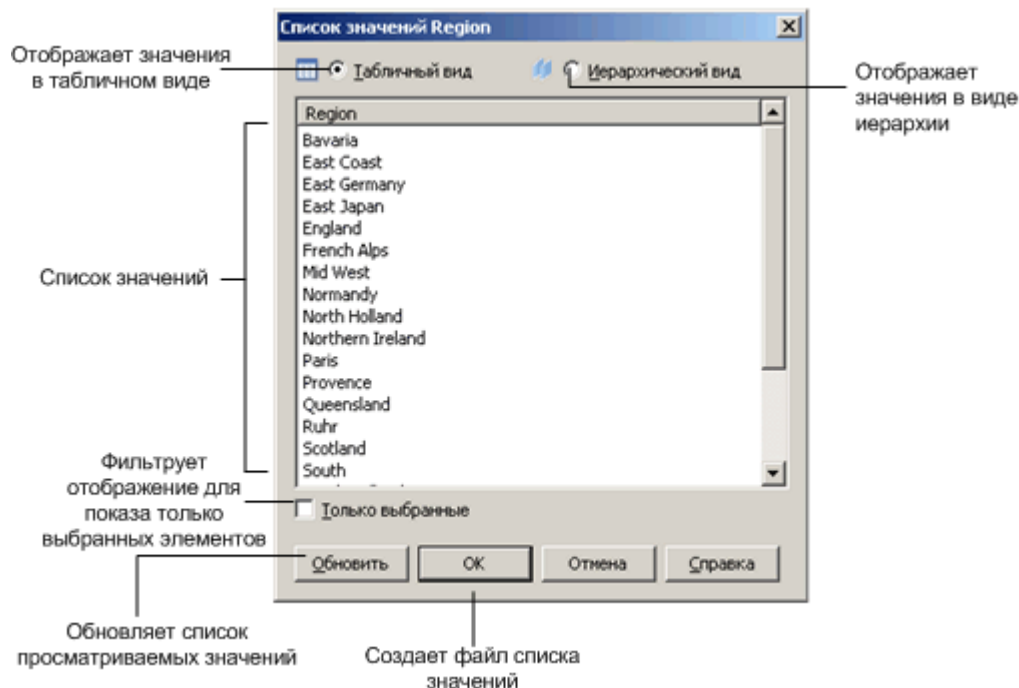
Список значений можно просматривать в формате списка или в виде иерархии объектов.

Чтобы просмотреть список значений, выполните следующие действия.

1. Дважды щелкните объект.
На странице определений откроется диалоговое окно "Изменить свойства".
2. Выберите вкладку "Свойства".
Откроется страница свойств.

3. Нажмите кнопку "Отобразить"

В диалоговом окне "Список значений" выводятся все возможные значения данных, связанные с объектом.



4. Нажмите "Отмена".

7.9.3.3 Создание списка значений

Список значений создается следующим образом.

1. Просмотрите список значений объекта.

2. Нажмите кнопку "OK".

В средстве создания юниверсов файлы со списками значений (LOV) хранятся в папке юниверса, вложенной в ту же папку, которая содержит файл юниверса. Имя вложенной папки совпадает с юниверсом, который содержит объект, использованный для создания файла .LOV.

После создания файла .LOV можно отредактировать список, чтобы ограничить данные, возвращаемые в файл .LOV, или изменить представление данных в списке.

7.9.4 Редактирование списка значений

Содержимое списка значений можно изменять двумя способами.

- Применить условие к запросу SELECT DISTINCT, который генерирует этот список. Например, курорты в списке значений объекта "Курорт" ограничиваются до тех курортов, которые имеют количество зарезервированных гостей больше заданного минимума.

- Создать иерархию для упрощения выбора пользователями значения в списке. Это бывает полезным, когда в списке содержится большое количество значений.

7.9.4.1 Применение условия к списку значений

Чтобы применить условия к списку значений, выполните следующие действия.

1. Дважды щелкните объект.
Откроется страница "Редактировать свойства".
2. Выберите вкладку "Свойства".
Откроется страница свойств.
3. Установите флажок "Связать со списком значений".
4. Если необходимо переименовать список, введите имя файла .LOV в поле "Имя списка".

☒ Связать со списком значений

☐ Позволять редактировать этот список

☐ Обновлять перед использованием

☐ Показ иерархии

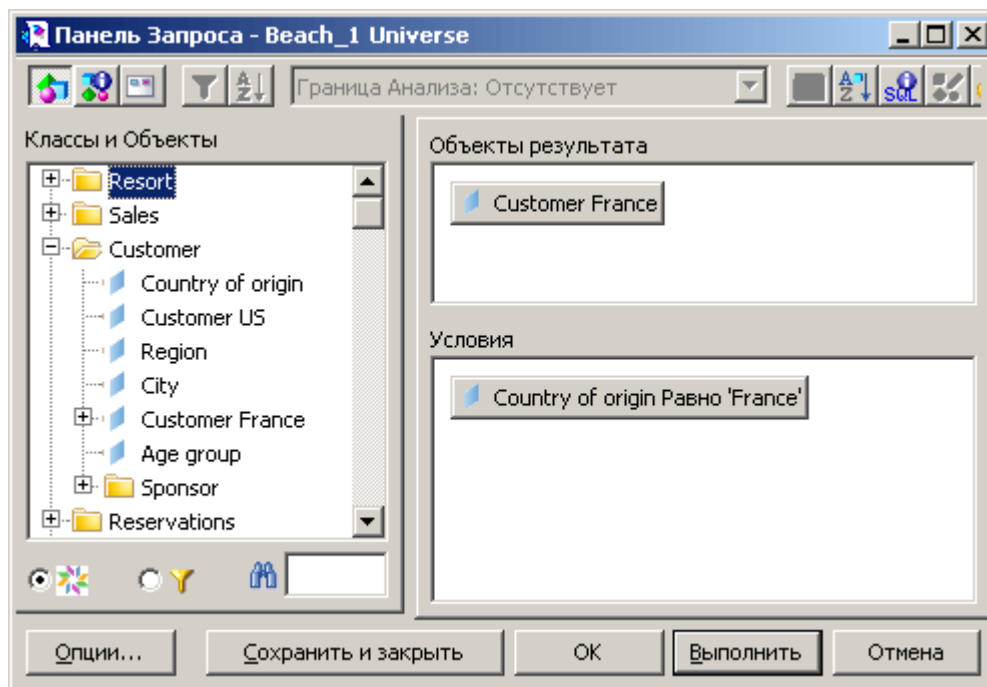
☒ Экспорт с Универсом

☐ Делегировать поиск

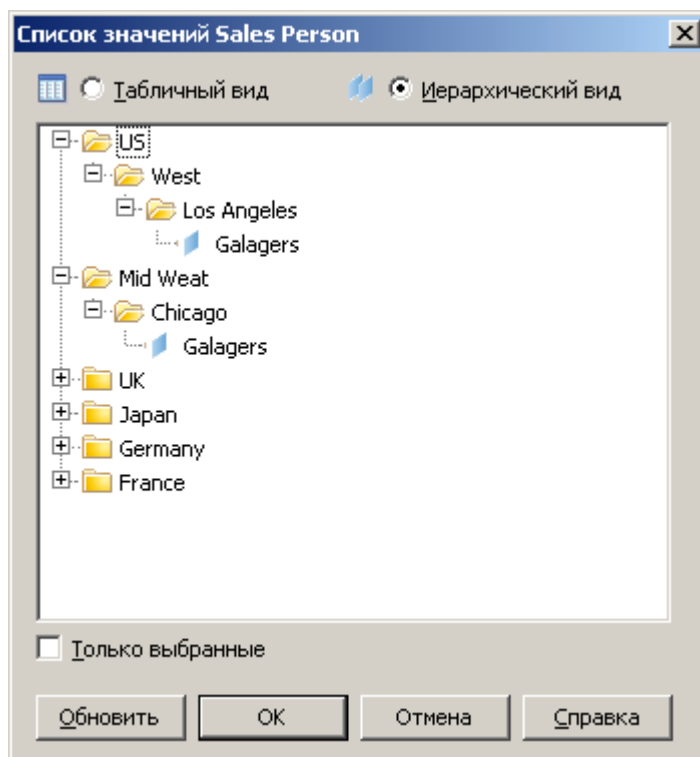
Имя списка:
Cust_FR

По умолчанию Изменить... Показать...

5. Нажмите кнопку "Редактировать".
Отобразится панель запросов. Активный объект находится на панели "Объекты результатов".
6. Перетащите на панель "Условия" объект, который будет служить условием на список значений для активного объекта.
7. Дважды щелкните оператор на панели "Операторы".
8. Дважды щелкните операнд на панели "Операнд".
9. При необходимости выберите или введите значения.
Например, следующий запрос возвращает клиентов только из Франции.



10. Нажмите кнопку "OK".
11. Чтобы просмотреть ограниченный список значений, нажмите "Отобразить".
Отобразится пустой список.
12. Нажмите "Обновить".
13. Значения отобразятся в списке.

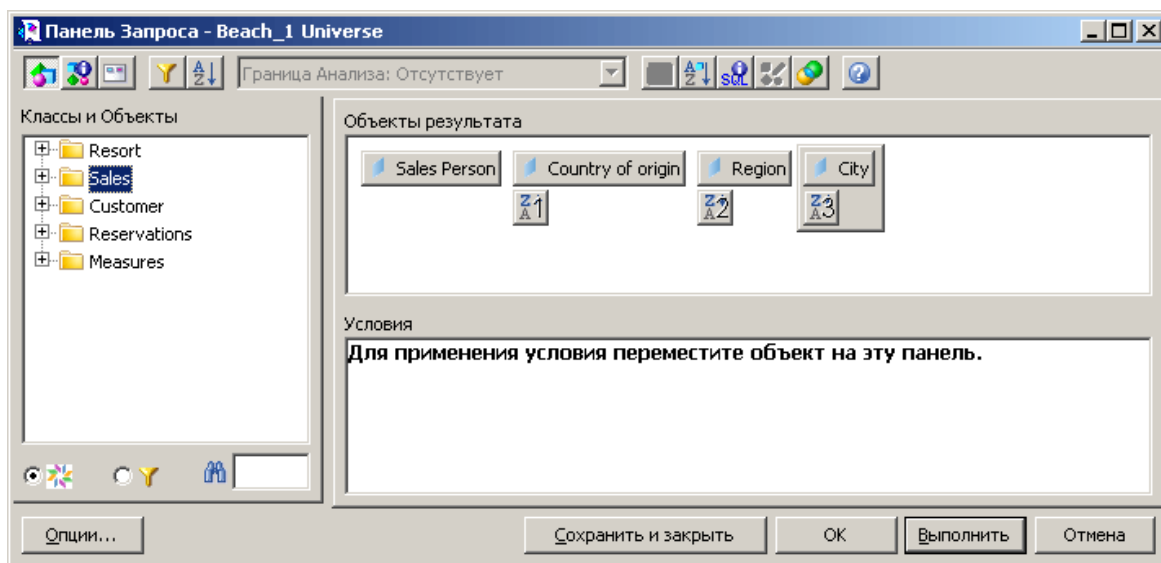


14. В каждом из диалоговых окон нажмите кнопку "OK".

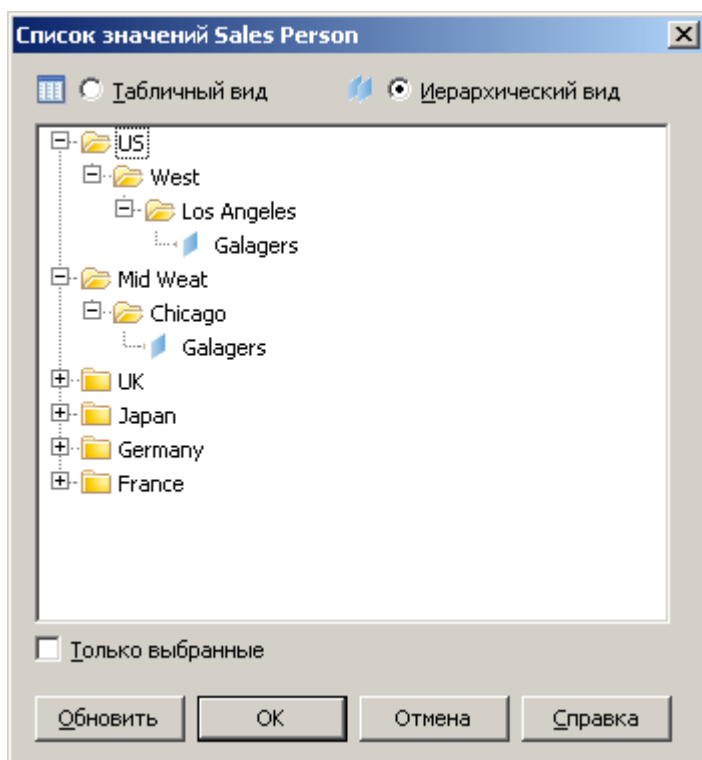
7.9.4.2 Создание иерархии для списка значений

Чтобы создать иерархию для списка значений,

1. Дважды щелкните объект.
Откроется страница "Редактировать свойства".
2. Выберите вкладку "Свойства".
Откроется страница свойств.
3. Установите флажок "Связать со списком значений".
4. Если необходимо переименовать список, введите имя файла .LOV в поле "Имя списка".
5. Нажмите кнопку "Редактировать".
Отобразится панель запросов. Активный объект находится на панели "Объекты результатов".
6. Перетащите объект, который необходимо переместить в иерархию, в окно "Объекты результатов" справа от существующего объекта, как показано ниже.



7. Нажмите кнопку "OK".
8. Чтобы просмотреть ограниченный список значений, нажмите "Отобразить".
Отобразится пустой список.
9. Нажмите "Обновить".
Значения отобразятся в списке.



10. В каждом из диалоговых окон нажмите кнопку "OK".

7.9.5 Экспорт списка значений

Можно экспортировать список значений с помощью юниверса в CMS. В файловой системе связанный файл .LOV копируется в подкаталог юниверса в ту же папку, где хранится файл юниверса.

7.9.5.1 Использование экспортированного файла .LOV в Web Intelligence

Если пользователь выполняет запрос в Web Intelligence с помощью объекта, который связан с LOV-файлом, экспортированным из средства создания юниверсов, список значений, который возвращается для объекта, определяется одним из следующих вариантов:

- Данные содержатся в файле .LOV.
- SQL для запроса "ВЫБРАТЬ ОТДЕЛЬНО" определен в файле .LOV.

Если в средстве создания юниверсов создано условие, ограничивающее значения данных, возвращаемые для объекта, вместо списка всех значений данных по умолчанию отображается ограниченный список. Этот список содержит все условия и параметры форматирования, примененные в средстве создания юниверсов.

Если не экспортировать файл .LOV с помощью юниверса, объект вернет список по умолчанию без условий и форматирования. В данном случае создается файл .LOV, чтобы хранить данные.

7.9.5.2 Экспорт списка с или без данных

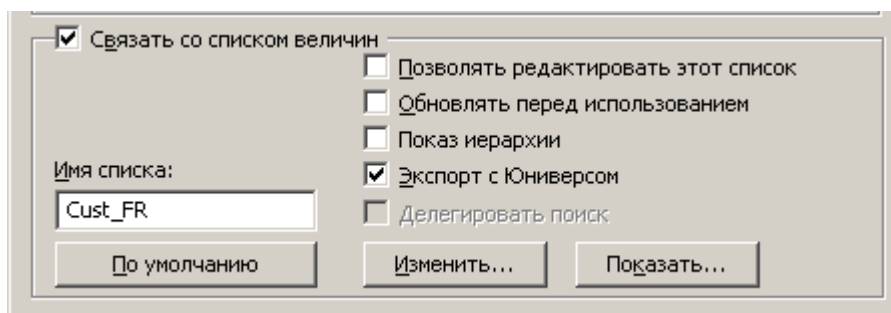
Можно экспортировать список значений в репозиторий сервера центрального управления (CMS) двумя способами.

Экспортировать .LOV...	Описание
Только с помощью определения запроса (нет данных)	Файл .LOV экспортируется с помощью определения запроса "ВЫБРАТЬ ОТДЕЛЬНО", чтобы вернуть значения в список. Сохраняются все условия, установленные для LOV-файла на панели запросов средства создания юниверсов. В файле .LOV не содержится данных, он заполняется при первом использовании объекта, чтобы вернуть значения на панели запросов. Необходимо использовать данный способ для регулярно обновляемых данных или в случае, если список значений очень большой.
С данными	LOV-файл экспортируется или импортируется с данными, которые возвращаются, если отобразить или редактировать список значений в средстве создания юниверсов. Это может быть полезно, если данные в файле .LOV не меняются. Однако если данные регулярно обновляются, или список значений переполнен, не следует экспортировать данные с файлом .LOV, так как это отобразит процесс экспорта.

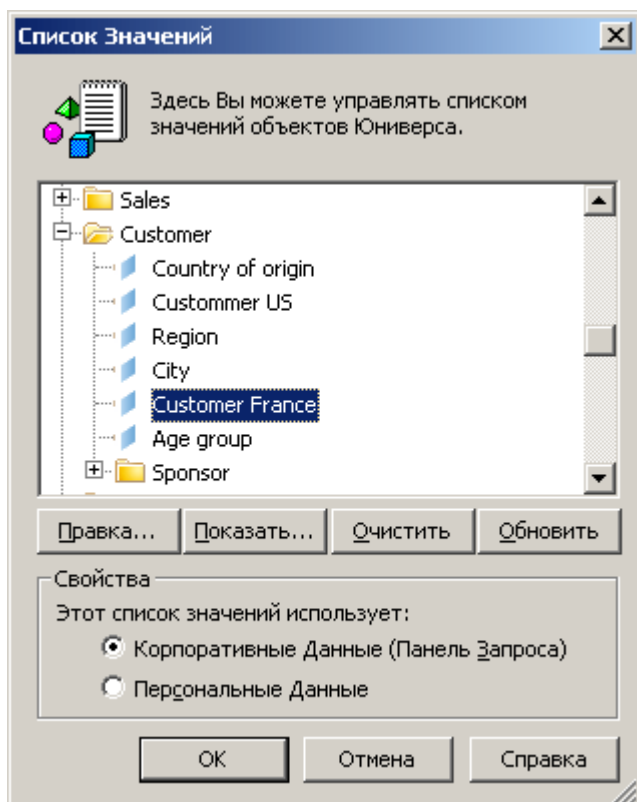
Экспорт определения списка значений

Чтобы экспортировать определение списка значений (нет данных),

1. Создайте список значений для объекта
2. Установите флажок "Экспорт с помощью юниверса" на странице "Свойства" для объекта.
Как показано ниже, список значений Cust_FR связан с объектом "Клиент", чтобы возвращать только значения для клиентов во Франции.



3. Выберите "Инструменты" > "Списки значений".
Откроется диалоговое окно "Список значений". В нем перечислены классы и объекты текущего юниверса, а также содержатся параметры, чтобы управлять списком значений каждого объекта.
4. Разверните класс и выберите объект со связанным файлом .LOV, который необходимо экспортировать в репозиторий.



5. Нажмите кнопку "Удалить".
В файле .LOV данные для объекта удалены. Сейчас в файле .LOV содержится только определение запроса для списка значений.
6. Нажмите кнопку "ОК".
7. Выберите "Файл" > "Экспорт".
Откроется диалоговое окно "Экспортировать юниверс".
8. В списке юниверсов выберите имя файла юниверса.
9. Нажмите кнопку "ОК".

Откроется окно с сообщением о том, что юниверс успешно экспортирован.

Экспорт списка значений с данными

Чтобы экспортировать список значений с данными,

1. Создайте список значений для объекта
2. Установите флажок "Экспорт с помощью юниверса" на странице "Свойства" для объекта.
3. Нажмите кнопку "Отобразить"
Отобразится список значений.
4. Если список пуст, нажмите кнопку "Обновить", чтобы заполнить список.
5. В каждом из диалоговых окон нажмите кнопку ОК.
6. Выберите "Файл" > "Экспорт".
Откроется диалоговое окно "Экспортировать юниверс".
7. В списке юниверсов выберите имя файла юниверса.
8. Нажмите кнопку "ОК".
Откроется окно с сообщением о том, что юниверс успешно экспортирован.

7.9.6 Обновление значений в списке значений

В средстве создания юниверсов можно обновлять данные в списке значений двумя способами:

- Отобразить список значений для объекта и нажать кнопку "Обновить".
- Выбрать "Инструменты" > "Списки значений", чтобы отобразить окно управления списком значений, потом выберите объект и нажмите кнопку "Обновить".

7.9.7 Использование данных в файле персональных данных

Можно назначить список значений для объекта, в котором содержатся не корпоративные, а персональные данные, извлеченные в сервере базы данных.

Персональные данные – это данные, которые хранятся в простом файле, например в текстовом файле, или данные в одном из следующих приложений: Microsoft Excel, Lotus 1-2-3 или dBASE.

Использование файла персональных данных, так как в списке значений есть следующие преимущества.

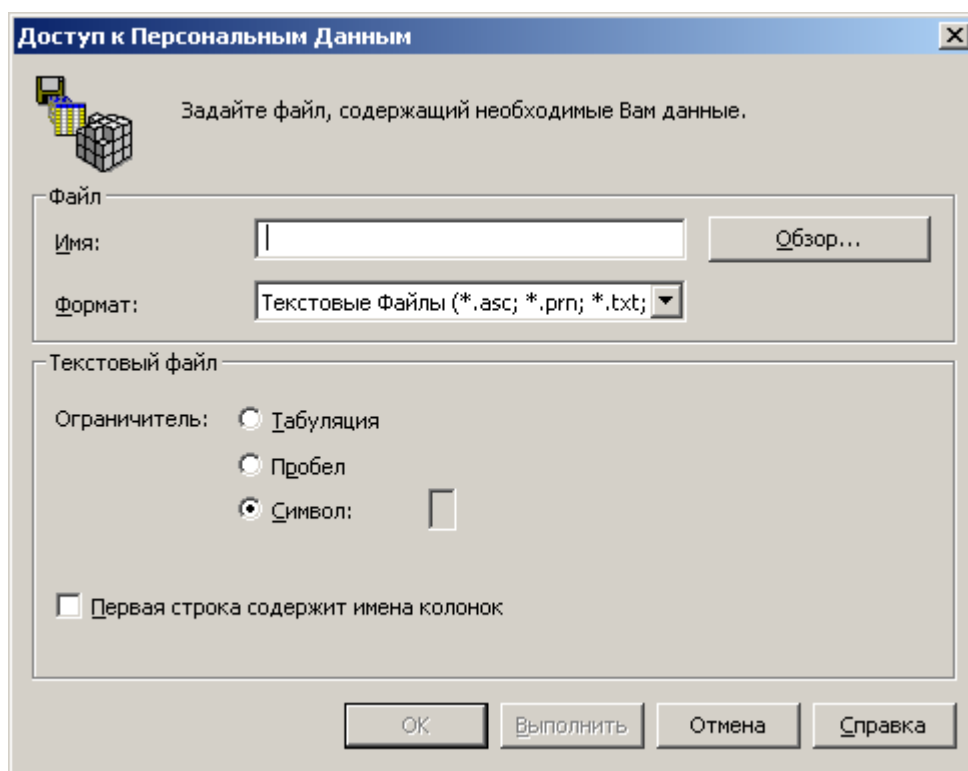
- Процесс извлечения файла персональных данных проходит быстрее, чем получение доступа к корпоративной базе данных.
- Пользователям необходимы значения, которых нет в базе данных.
- Можно управлять значениями, которые видят пользователи во время работы со списком значений.

Недостаток использования файла персональных данных заключается в том, что данные в нем являются фиксированными. Необходимо обновлять данные вручную, если значения необходимо изменить.

7.9.7.1 Создание списка значений в файле персональных данных

Чтобы создать список значений в файле персональных данных,

1. Выберите "Инструменты" > "Списки значений".
Откроется диалоговое окно "Список значений".
2. Разверните класс и выберите объект.
3. В групповом окне "Свойства" нажмите переключатель "Персональные данные".
В окне сообщения отобразится информация о том, что список значений будет изменен с корпоративного на персональный.
4. Нажмите кнопку "ОК".
Откроется окно "Доступ к персональным данным". Доступные параметры зависят от выбранного типа файла.



5. Нажмите кнопку "Просмотр" и выберите файл, который необходимо использовать в качестве списка значений.
Или
В текстовом поле "Имя" введите имя файла.
6. В окне списка "Формат" выберите файл.
7. Можно выбрать один из следующих форматов файла.
 - Текстовые файлы (*.asc; *.prn; *.txt; *.csv)
 - Файлы Microsoft Excel
 - dBASE

- Microsoft Excel 97.

ⓘ Примечание

Если файл создан в Excel 97, необходимо использовать параметр Microsoft Excel 97 вместо параметра "Файлы Microsoft Excel".

8. Если необходимо, укажите оставшиеся параметры.
В текстовом файле строки эквивалентны. Определите тип разделителя столбцов для текстового файла: табуляция, пробел или символ. Выбрав символ, введите его в текстовое поле.
9. Нажмите кнопку "ОК".

7.9.8 Администрирование списков значений в юниверсе

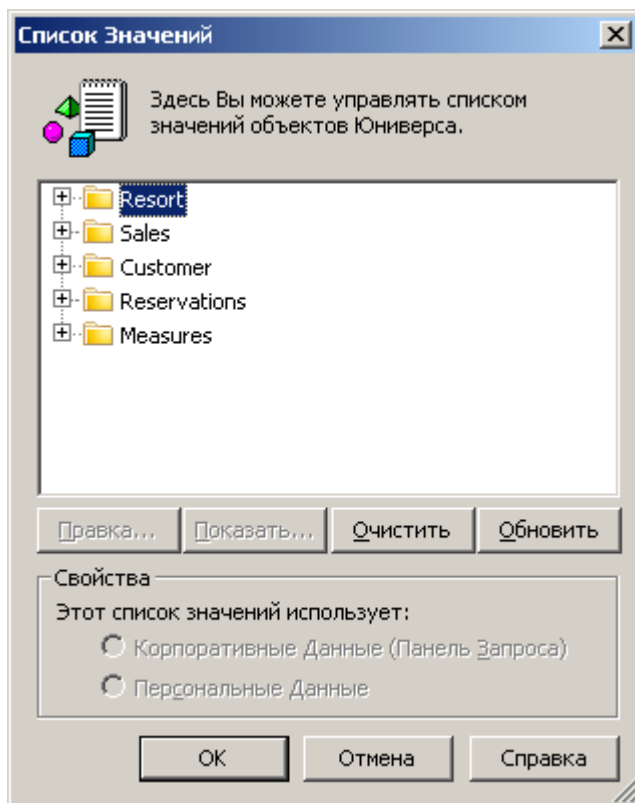
С помощью диалогового окна "Списки значений" можно управлять списками значений в активном юниверсе ("Инструменты" > "Списки значений"). Все классы и объекты представлены в виде дерева. Можно выбрать объект и получить доступ к его списку значений. В диалоговом окне "Списки значений" можно выполнить следующие действия.

Действие	Описание
Редактировать	Отображает панель запросов, которая используется, чтобы определить запрос для выбранного объекта. Для списка значений можно определять и редактировать существующие запросы.
Отобразить	Отображает текущий список значений выбранного объекта.
Очистить	Удаляет содержимое списка значений, назначенное для выбранного объекта.
Обновить	Обновляет отображение списка значений.

7.9.8.1 Получение доступа к инструменту администрирования списков значений

Чтобы получить доступ к инструменту администрирования списков значений,

1. Выберите "Инструменты" > "Списки значений" > "редактировать список значений".
Откроется диалоговое окно "Списки значений".



2. Разверните класс и выберите объект.
3. Нажмите кнопку и выберите параметр, чтобы выполнить административную задачу.
4. Нажмите кнопку "ОК".

7.9.9 Оптимизация и настройка файлов списков значений

Существует несколько общих способов, как оптимизировать и настраивать списки значений.

Метод	Описание
Укажите список значений для более маленькой таблицы	По умолчанию списки значений указывают на тот же объект, к которому прикреплены. Но если объект указывает на большую таблицу (количество строк), процесс обновления списка значений длится долго. Если существует альтернативная более маленькая и быстрая таблица, которая возвращает одинаковые значения, тогда список значений необходимо отредактировать, чтобы он указывал на альтернативную таблицу.

Метод	Описание
Объединение кода и описания	Обычная настройка файла .LOV – это объединить код и описание. Объект возвращает "код типа продаж", который не представляет весомое значений для некоторых пользователей. Редактирование списка значений, чтобы отобразить "описание типа продаж", поможет при просмотре списка значений. Можно выполнить обратное действие "описание типа продаж", чтобы отобразить код вместе с описанием.

7.10 Связывание юниверсов

Можно динамически связать один или несколько юниверсов.

7.10.1 Что представляют собой связанные юниверсы?

Связанные юниверсы – это юниверсы, которые совместно используют общие компоненты, например параметры, классы, объекты или объединения.

В связывании двух юниверсов один юниверс играет роль главного юниверса, а другой – производного. Изменения, выполненные в главном юниверсе, автоматически распространяются по всем производным юниверсам.

❗ Примечание

Сведения о развертывании связанных юниверсов см. в разделе [Производные юниверсы и списки значений \[страница 347\]](#)

7.10.1.1 Что представляет собой главный юниверс?

Главный юниверс – это юниверс, с которым связаны другие юниверсы. В нем содержатся компоненты, которые являются общими для других юниверсов, связанных с ним. Данные юниверсы называются производными юниверсами. Главный юниверс представляет библиотеку компонентов многократного использования.

Главный юниверс может быть базовым или основным юниверсом в зависимости от способа использования компонентов главного юниверса с производных юниверсах. Базовые и основные юниверсы описаны в разделе [Создание связи между двумя юниверсами \[страница 342\]](#).

7.10.1.2 Что представляет собой производный юниверс?

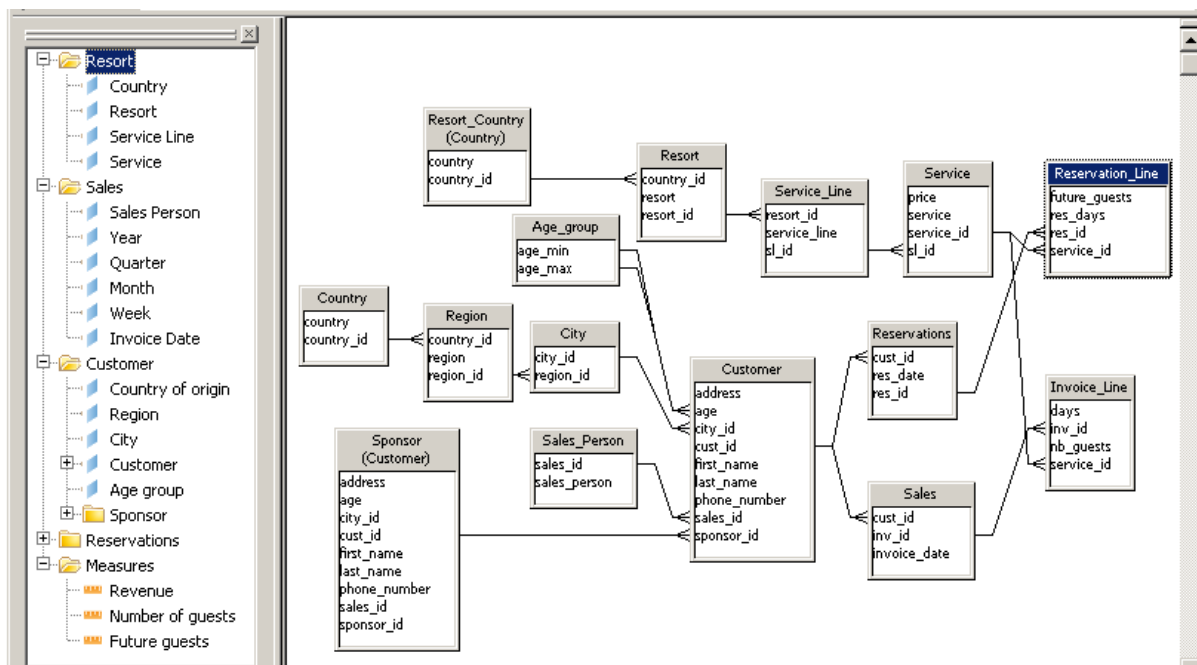
Производный юниверс – это юниверс, который содержит ссылку на главный юниверс. Такая связь позволяет производному юниверсу совместно использовать общие компоненты главного юниверса.

- Если связанный главный юниверс является базовым, то в производный юниверс можно добавлять компоненты.
- Если связанный главный юниверс является основным, то производный юниверс будет содержать все компоненты главного юниверса. Классы и объекты в производный юниверс не добавляются. Они могут быть скрыты в производном юниверсе в зависимости от потребностей пользователей из целевой аудитории.

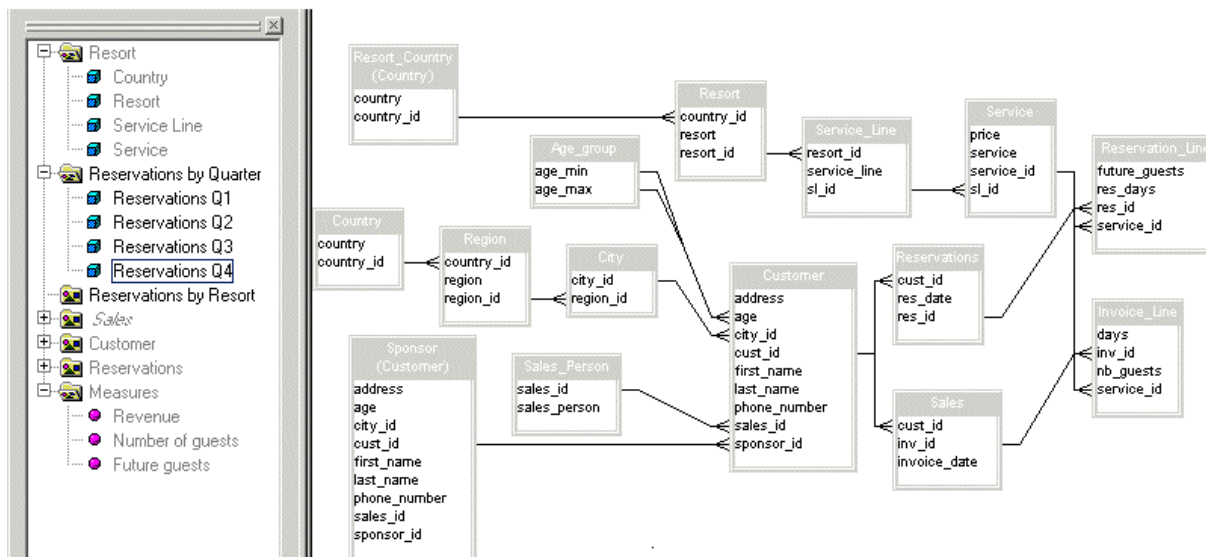
Пример: Связанные главный и производный юниверсы.

В этом примере показаны два юниверса: один – главный юниверс, содержащий общие компоненты, другой – производный юниверс, который использует структуры главного, но в нем также есть классы и объекты, являющиеся для него специфическими.

Beach.unv – это главный юниверс. Он используется менеджером по продажам по направлению островных курортов (Island Resorts) для проведения маркетингового анализа. Этот юниверс является одним из демонстрационных вариантов, поставляемых с этой версией. Содержимое юниверса приведено ниже.



Используя этот юниверс, менеджер создает производный юниверс, предназначенный для предварительных заказов.



Компоненты, производные от главного юниверса, будут неактивными. Менеджер создал два новых класса: "Предварительные заказы по кварталам" (Reservations by Quarter) и "Предварительные заказы по курортам" (Reservations by Resort). Эти классы и их объекты обычно отображаются. Менеджер также решил скрыть класс "Продажи" (Sales), который не нужен в юнивере "Предварительные заказы" (Reservations). Все изменения в компонентах главного юниверса автоматически распространяются на производный юниверс.

7.10.2 Различные способы связывания юниверсов

Для установления связи между юниверсами можно использовать любой из следующих подходов:

- Подход с базовым юниверсом
- Подход с основным юниверсом
- Компонентный подход

Любой из этих подходов можно применять по отдельности или в их сочетании.

7.10.2.1 Подход с базовым юниверсом

При использовании подхода с базовым юниверсом один юниверс содержит компоненты главного юниверса. Это компоненты, которые являются общими для всех юниверсов. Производные юниверсы, созданные из этого базового юниверса, содержат эти главные компоненты наряду со своими собственными специфическими компонентами.

Все изменения, произведенные в базовом юниверсе, автоматически отражаются в главных компонентах всех производных юниверсов.

7.10.2.2 Подход с основным юниверсом

Это другой способ организации общих компонентов связанных юниверсов.

Основной юниверс содержит все возможные компоненты. В юниверсах, производных от основного, некоторые компоненты являются скрытыми в зависимости от их соответствия задачам целевых пользователей производного юниверса.

Компоненты, видимые в производных юниверсах, всегда являются подмножеством основного юниверса. Непосредственно в производный юниверс новые компоненты не добавляются. Ниже в примере показано, как юниверсы "Трудовые ресурсы" и "Продажи" выводятся из основного юниверса. Они содержат компоненты из основного юниверса, некоторые из которых могут быть скрыты.

Все изменения, произведенные в основном юниверсе, автоматически отражаются в главных компонентах всех производных юниверсов.

7.10.2.3 Компонентный подход

В данном подходе два или более юниверса объединяются в один юниверс. Ниже показано, как юниверс "Продажи" (Sales) был создан путем объединения двух юниверсов: "Часть 1" (Part 1) и "Часть 2" (Part 2).



7.10.3 Преимущества связывания юниверсов

Связывание юниверсов дает следующие преимущества.

- Уменьшение времени разработки и эксплуатации. Изменение компонента в главном юниверсе распространяется средством создания юниверсов на идентичный компонент во всех производных юниверсах.
- Часто используемые компоненты можно централизовать в главном юниверсе, а затем включать их во все новые юниверсы. Нет необходимости заново создавать общие компоненты каждый раз при создании нового юниверса.
- Повышение эффективности специализации. Разработку юниверсов можно распределить между администраторами баз данных, которые будут устанавливать базисный главный юниверс, и

более специализированными дизайнерами (разработчиками), которые будут создавать более функциональные юниверсы для определенных областей.

7.10.4 Требования по связыванию юниверсов

Активный юниверс можно связывать с главным юниверсом, только если выполняются следующие требования.

- Главный юниверс и производный юниверс используют одни и те же данные учетной записи и одну и ту же реляционную СУБД. Использование одного и того же соединения для главного и производного юниверсов облегчает управление юниверсами, но это можно изменить в любое время.
- Главный и производный юниверсы должны располагаться в одном репозитории.
- Главный юниверс был экспортирован и повторно импортирован по крайней мере один раз. Производный юниверс перед созданием связи экспортировать не нужно.
- Экспортированные производные юниверсы располагаются в том же домене юниверса, что и главный юниверс.
- Есть права на связывание данного юниверса.

7.10.5 Ограничения при связывании юниверсов

При установлении связи между юниверсами необходимо иметь в виду следующие ограничения.

- Нельзя установить связь с юниверсом, который использует хранимые процедуры.
- Можно использовать только один уровень связывания. Нельзя создавать производные юниверсы на основе юниверса, который является производным сам по себе.
- Все классы и объекты являются уникальными как в главном, так и в производных юниверсах. Если не возникнут противоречия.
- Структуры этих двух юниверсов должны позволять создавать объединения между таблицами из разных юниверсов. Если нет, при выполнении запроса могут быть получены декартовы произведения с объектами из обеих структур.
- В производном юниверсе доступны только схема таблицы, классы и объекты из главного юниверса. Контексты в производном юниверсе необходимо повторно определить.
- Когда экспортируется производный юниверс со структурами главного юниверса, список значений, связанных с главным юниверсом, не сохраняется.

7.10.6 Создание связи между двумя юниверсами

Активный юниверс можно связать с другим юниверсом. В этом случае активный юниверс становится производным, а связанный с ним юниверс – главным. Компоненты главного юниверса наследуются производным юниверсом.

Для того чтобы связать какой-либо юниверс с главным юниверсом, необходимо экспортировать главный юниверс из репозитория.

❗ Примечание

При связывании юниверсов можно перемещать главный юниверс в пределах того же репозитория, не разрывая связь. Это позволяет экспортировать главный юниверс в другую папку репозитория, сохраняя в то же время действующие связи с производными юниверсами.

7.10.6.1 Создание связи между производным юниверсом

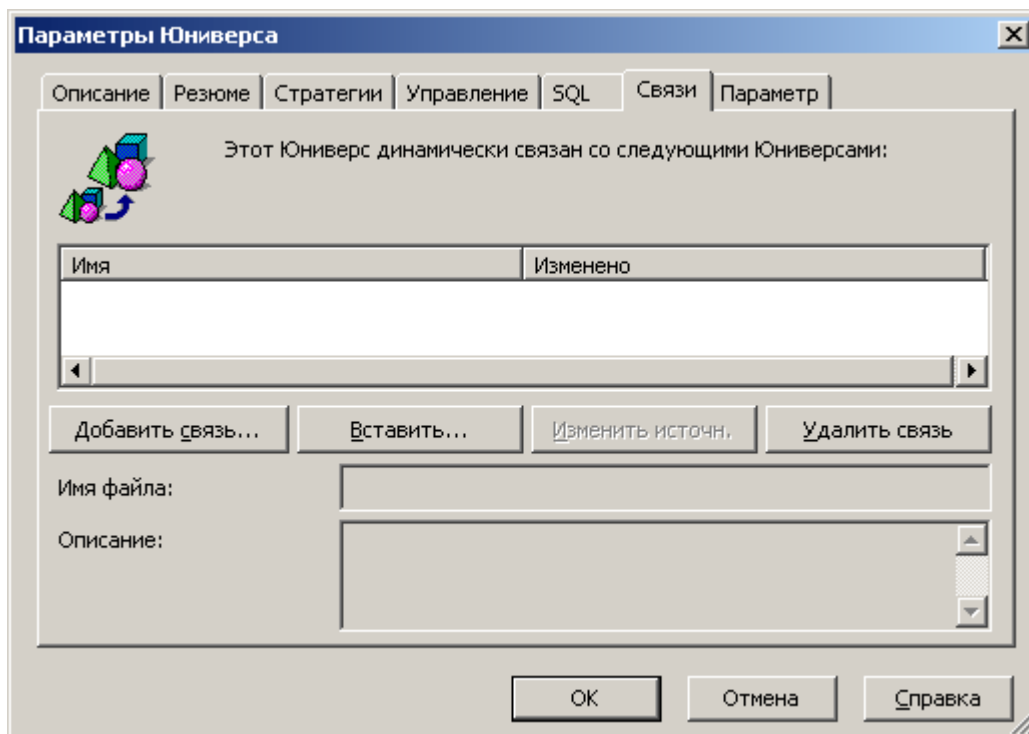
1. Убедитесь, что активный юниверс для создания связи с главным юниверсом выбран правильно.

Например, приведенный ниже юниверс является версией юниверса "Пляж" (Beach), который содержит сведения только по странам, но не по курортам. Предположим, требуется связать этот юниверс продаж с юниверсом курортов, который содержит данные по курортам. Ниже юниверс продаж "Пляж" (Beach) – производный юниверс, а юниверс "Курорт" (Resort) – это главный юниверс.



2. Выберите "Правка" > "Связи".

Откроется диалоговое окно "Параметры юниверса" на странице "Связи".

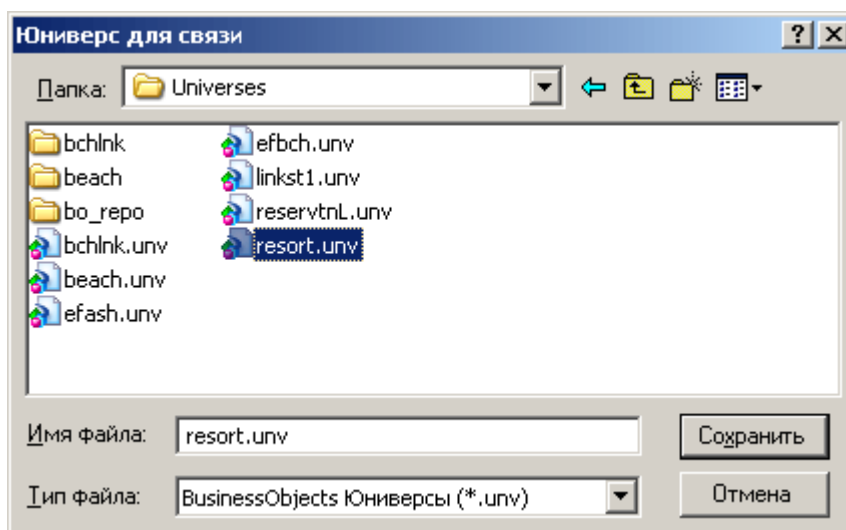


3. Нажмите кнопку "Добавить связь".

Откроется диалоговое окно "Связать универс". В нем перечислены универсы доступных доменов.

4. Переместитесь к универсу, с которым требуется создать связь. Компоненты данного универса, который теперь является главным, будут использоваться в активных универсах.

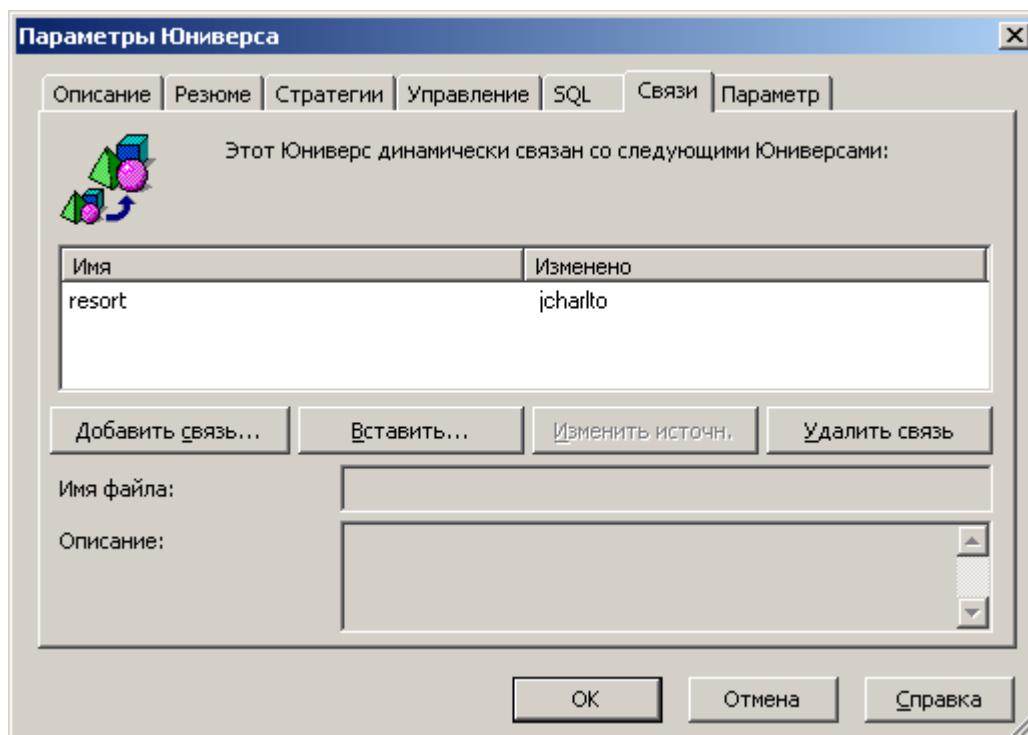
В примере выбран универс курортов.



Если данный универс никогда не экспортировался, то появится сообщение об ошибке. Перед созданием связи универс необходимо экспортировать.

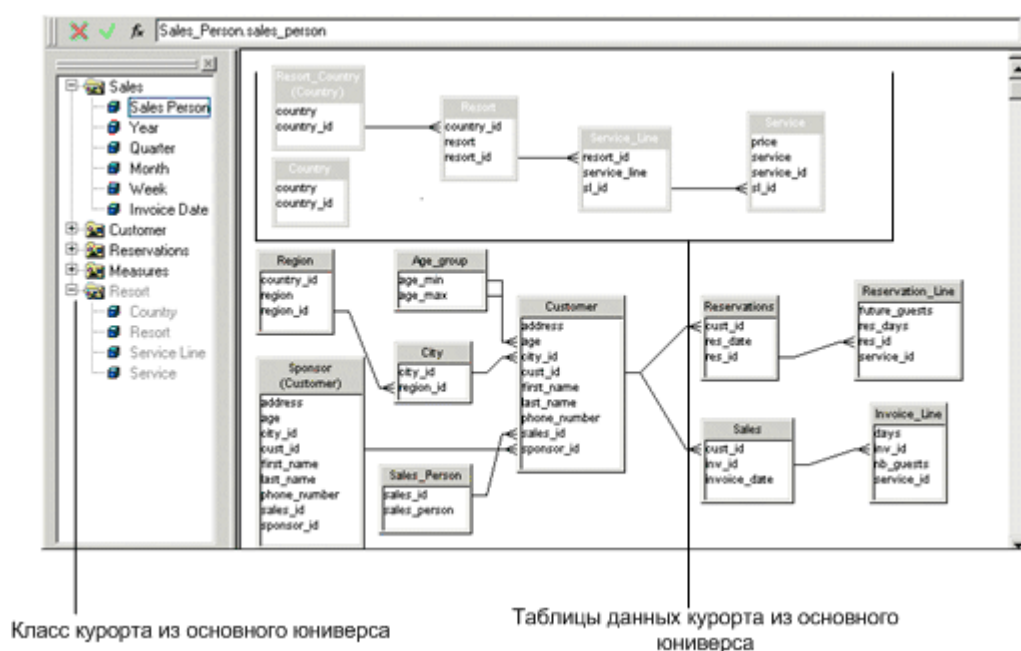
5. Нажмите кнопку "Открыть".

Выбранный юниверс отобразится в списке.



6. Нажмите кнопку "OK".

Связь создана. В активном юниверсе основные компоненты отображаются как неактивные.



7.10.7 Редактирование производного юниверса

Процесс связывания завершается созданием объединений между главными таблицами и таблицами производного юниверса. Для этого необходимо удалить все существующие контексты и повторно определить контексты для новой структуры.

📘 Примечание

Нельзя изменять какую-либо структуру, класс или объект из связанного юниверса (главного юниверса) внутри производного юниверса.

7.10.7.1 Редактирование производного юниверса

Чтобы отредактировать производный юниверс, выполните следующие действия.

1. Создайте объединения структур главного и производного юниверсов.
Создание объединений обеспечивает, что для объектов, включаемых в запрос из обеих структур, не будут возвращаться декартовы произведения.
2. Удалите существующие контексты.
3. Определите псевдонимы.
4. Определите контексты.
5. При необходимости скройте или создайте новые объекты.

📘 Примечание

Дополнительные сведения о скрытии компонента см. в разделе [Скрыть или отобразить классы, объекты и условия \[страница 267\]](#).

7.10.8 Удаление связи

Удаление связи с главным юниверсом возможно только в том случае, если производный юниверс не содержит объектов, основанных на главных компонентах или на объединениях, включающих в себя главные компоненты.

7.10.8.1 Удаление связи в производном юниверсе

Чтобы удалить связь в производном юниверсе, выполните следующие действия.

1. Откройте производный юниверс.

2. Выберите "Правка" > "Связи".
Откроется страница связей в диалоговом окне "Параметры юниверса".
3. Выберите в списке имя главного юниверса.
4. Нажмите кнопку "Удалить связь".
5. Нажмите "ОК".
Теперь компоненты главного юниверса удалены из активного юниверса.

7.10.9 Перемещение главного юниверса

Если положение главного юниверса изменилось, то для сохранения связи необходимо указать это новое положение.

7.10.9.1 Обновление связи с перемещенным главным юниверсом

Чтобы обновить связь с перемещенным главным юниверсом, выполните следующие действия.

1. Откройте производный юниверс.
2. Выберите "Правка" > "Связи".
3. Выберите в списке связанный главный юниверс.
4. Нажмите кнопку "Изменить источник".
Откроется диалоговое окно "Связать юниверс".
5. Перейдите к новому положению главного юниверса.
6. Нажмите кнопку "Открыть".
Новый главный юниверс появится в списке связей.

7.10.10 Производные юниверсы и списки значений

Списки значений, связанных с главными объектами, не сохраняются вместе с производным юниверсом при его экспорте в репозиторий.

Существует следующий способ сохранения списков значений, связанных с главными объектами.

1. Новые объекты следует создавать, используя то же определение, что и для объектов, содержащих списки значений, которые необходимо экспортировать в репозиторий вместе с главным юниверсом.
2. Присвойте новым объектам списки значений, идентичные главным объектам.
3. Скройте эти новые объекты.
Скрытые объекты выполняют функцию сохранения списков значений, благодаря чему их можно экспортировать и импортировать вместе с производным юниверсом.

7.10.11 Представление объектов в порядке их следования в главном юниверсе

По умолчанию порядком, в котором упорядочиваются объекты производного юниверса, является тот, который будет виден пользователям юниверса, даже если в главном юниверсе позже изменится порядок. Если необходимо, чтобы производный юниверс всегда представлял объекты в том же порядке, в каком они представлены в главном юниверсе, то для это следует установить параметр согласно файлу *.PRM используемой базы данных.

Настройка параметра производится следующим образом: CORE_ORDER_PRIORITY = Y.

Дополнительные сведения о том, как устанавливать параметры в соответствующем файле *.PRM, см. в "Руководстве по доступу к данным (Справка > Руководство по доступу к данным).

7.11 Включение одного юниверса в другой

Компоненты главного юниверса можно копировать в производный юниверс. Полученные в результате компоненты производного юниверса будут независимы от исходных компонентов главного юниверса. Эти компоненты не будут связаны с главным юниверсом. Никакие изменения, вносимые в главный юниверс, не наследуются производным юниверсом.

7.11.1 Копирование главного юниверса в производный юниверс

При копировании главного юниверса в производный полученные в результате этого компоненты производного юниверса будут независимы от исходных компонентов главного юниверса. Эти компоненты не будут связаны с главным юниверсом. Никакие изменения, вносимые в главный юниверс, не наследуются производным юниверсом.

Копирование главного юниверса в производный может быть необходимо в следующих случаях:

- чтобы скопировать содержимое данного юниверса в активный юниверс;
- если больше не нужно поддерживать динамическую связь между двумя юниверсами.

❗ Примечание

Если до выполнения этой операции данные два юниверса были связаны, процедура удалит эту динамическую связь, так что компоненты в активном юниверсе больше не будут динамически связаны с внешним юниверсом.

7.11.1.1 Копирование главного юниверса в производный юниверс

Чтобы скопировать главный юниверс в производный, выполните следующие действия.

1. Откройте юниверс.
2. Выберите "Правка" > "Связи".
Откроется страница связей в диалоговом окне "Параметры юниверса".
3. Нажмите кнопку "Добавить связь".
Откроется диалоговое окно "Связать юниверс". В нем перечислены юниверсы доступных доменов.
4. Перейдите к юниверсу, который требуется скопировать, и выберите его. Компоненты данного юниверса, который теперь является главным, будут использоваться в активных юниверсах.
5. Нажмите кнопку "Включить".
6. Нажмите кнопку "ОК".
Теперь компоненты главного юниверса отображаются внутри активного юниверса.

7.12 Создание юниверсов хранимых процедур

Юниверс хранимых процедур – это специальный юниверс, позволяющий пользователям Web Intelligence получить доступ к хранимым процедурам в базе данных. Это единственный способ для пользователей Web Intelligence получить доступ к хранимым процедурам. Пользователи Web Intelligence не могут использовать юниверсы хранимых процедур для создания отчетов, основанных на хранимых процедурах в базе данных.

Хранимая процедура – это скомпилированная программа SQL, которая состоит из одной или нескольких SQL-инструкций, хранится в целевой базе данных и там же выполняется.

Хранимые процедуры предлагают следующие преимущества.

- Они инкапсулируют код. Действие базы данных отобразится только один раз в хранимой процедуре и ни разу в источнике приложения. Это улучшает отладку также как и удобство эксплуатации.
- Изменения в схеме базы данных влияет на код источника только в одном месте, в хранимой процедуре. Любые изменения схемы – задача администрирования базы данных, а не исправление кода.
- Так как хранимые процедуры находятся на сервере, можно установить более строгие ограничения безопасности. Это сохраняет большее количество доверительных полномочий для хорошо защищенных хранимых процедур.
- Так как хранимые процедуры скомпилированы и хранятся за пределами приложения клиента, они могут использовать более конфиденциальные переменные в синтаксисе SQL, такие как пароль или личные данные.
- Использование хранимых процедур снижает сетевой трафик.

Следующие ограничения применяются к юниверсам хранимых процедур:

- Объединения между объектами в юниверсах хранимых процедур запрещены.
- В юниверсах хранимых процедур не используются фильтры.
- Невозможно установить связь между юниверсом хранимой процедуры и стандартным юниверсом.

- Администратор Web Intelligence предоставляет доступ к базе данных или учетной записи, содержащей хранимые процедуры.
- Не все RDBMS поддерживают хранимые процедуры. В этом случае обратитесь к руководству базы данных.
- Инструкции COMPUTE, PRINT, OUTPUT или STATUS, содержащиеся в хранимых процедурах, не выполняются.

7.12.1 Хранимые процедуры в юниверсах Java bean

BusinessObjects XI версии 3.0 поддерживает создание юниверса на базе Java beans. Юниверс на основе Java beans возвращает результирующее множество, с помощью которого создаются отношения сущностей юниверса.

Юниверс с основой Java beans использует те же потоки работ и обладает теми же преимуществами, что и юниверс, созданный на базе хранимых процедур. Ограничения совпадают.

- Объединения не допустимы
- Использование фильтров в юниверсах не допустимо.

Дополнительные сведения о доступе к Java beans см. в [Руководстве по доступу к данным](#).

7.12.2 Создание юниверса на основе хранимых процедур

Система Business Objects поддерживает хранимые процедуры:

- без параметров
- с параметрами (IN)
- с многорезультатными множествами
- с множественными инструкциями (с предложениями SQL, отличными от SELECT)

Чтобы создать хранимые процедуры с параметрами, запустите мастер Quick Design и выполните шаги, описанные в разделе [Создание юниверса на основе хранимых процедур с параметрами](#).

Связанные сведения

[Использование хранимых процедур \[страница 351\]](#)

[Создание юниверса на основе хранимых процедур с входными параметрами. \[страница 352\]](#)

[Хранимые процедуры с большим количеством результатов \[страница 353\]](#)

7.12.2.1 Классы и объекты в хранимых процедурах

- Средство создания юниверсов создает одну таблицу для выбранной хранимой процедуры (или несколько таблиц для многочисленных результирующих множеств) и один объект на столбец, возвращаемый хранимой процедурой.
- Структура результирующего множества определяется при описании функции.

7.12.2.2 Использование хранимых процедур

Можно создать юниверс, основанный на одной или нескольких хранимых процедурах без параметров. Используйте мастер быстрого дизайна на панели инструментов.

1. Нажмите кнопку *Мастер Quick Design* на панели инструментов.
Отобразится начальная панель окна.
2. Установите флажок *Нажмите здесь для выбора юниверса хранимых процедур* внизу панели.
3. Нажмите *Начать*.
Отобразится панель *Определить параметр юниверса*.
4. Введите имя юниверса в поле *Ввести имя юниверса*.
5. Выберите подключение базы данных из раскрывающегося списка в списке *Выбрать соединение базы данных*.
6. Нажмите *Далее*.
Отобразится панель *Создать начальные классы и объекты*.
7. Щелкните мышью хранимую процедуру.
8. Нажмите *Добавить*.
Хранимая процедура создается в панели *Классы и объекты юниверса*.
9. Нажмите *Далее*.
10. Нажмите кнопку *Завершить*. Отобразится панель поздравления.

7.12.2.2.1 Параметр хранимой процедуры в юниверсе

Для увеличения производительности необходимо иметь несколько параметров хранимых процедур, основанных на одном юниверсе для одного источника данных (выберите "Мастер быстрого проектирования" или "Вставить"> "Хранимые процедуры"> "Обновить")

В параметрах Юниверса убедитесь, что для параметра `STORED_PROC_UNIVERSE` установлено значение `YES`. Это обозначает, что текущий юниверс основан на хранимой процедуре.

Чтобы избежать ошибки при синтаксическом анализе в столбцах хранимых процедур, рекомендуется, чтобы столбцы результата псевдонима основывались на сложном SQL, например, используя функции агрегирования `sum` и `count`. Создание объектов с псевдонимами не может быть ограничено.

⚠ Ограничение

Хранимые процедуры не поддерживают `OUT` или параметры наборов динамических результатов.

7.12.2.3 Создание юниверса на основе хранимых процедур с входными параметрами.

Если данная процедура приглашает пользователя выбрать какое-либо значение из списка в базе данных, то декларированные объекты уже имеются.

Можно создать юниверс, основанный на хранимых процедурах, которые требуют один или несколько входных параметров. В зависимости от введенного значения процедура возвращает данные факта на основе значения из соответствующей таблицы фактов.

1. Нажмите кнопку *Мастер Quick Design* на панели инструментов.
Отобразится начальная панель окна.
2. Установите флажок *Нажмите здесь для выбора юниверса хранимых процедур* внизу панели.
3. Нажмите *Начать*.
Отобразится панель *Определить параметр юниверса*.
4. Введите имя юниверса в поле *Ввести имя юниверса*.
5. Выберите подключение базы данных из раскрывающегося списка в списке *Выбрать соединение базы данных*.
6. Нажмите *Далее*.
Отобразится панель *Создать начальные классы и объекты*.
7. Щелкните мышью хранимую процедуру.
8. Нажмите *Добавить*.
Когда хранимая процедура запрашивает входные параметры, открывается *Редактор хранимых процедур*.
9. Выберите в списке нужный параметр.
10. Введите значение для параметра в поле *Значение* или наберите его в поле подсказки.
11. Выберите параметр *Использовать указанное значение* или *Запрашивать искомое значение*.
При вводе значения во время выполнения процедуры ей передается это значение, и она извлекает столбцы и структуру результирующего множества.
12. Если выбран параметр *Запрашивать искомое значение*, введите его в поле подсказки.
Можно ввести сообщение в поле подсказки или просмотреть папку и выбрать существующий объект (например, список идентификаторов абонента из таблицы).
13. Нажмите кнопку *ОК*.
14. Нажмите *Далее*.
15. Нажмите кнопку *Завершить*. Отобразится панель поздравления.

7.12.2.3.1 Подсказки значений

Использование подсказки для определения значения для параметра при выполнении хранимой процедуры

По умолчанию имя параметра хранимой процедуры происходит от структуры хранимой процедуры и отображает подсказку с именем хранимой процедуры.

Чтобы добавить больше значений из списка, необходимо настроить семантику и связать список значений с подсказкой.

Перед каждым параметром, в диалоговом окне параметра хранимой процедуры, имеется кнопка, которая открывает дополнительное диалоговое окно.

7.12.2.4 Использование списка значений в подсказке для хранимых процедур

↔ Синтаксис

Затем с помощью динамического параметра задайте хранимую процедуру. Можно выбрать объект юниверса на основе стандартной таблицы путем сопоставления списка значений (lov) формулировкам подсказки (при чем список значений должен быть из стандартной таблицы). Это удобный способ предложения пользователю списка значений объектов юниверса.

Список значений также может содержать и простые значения. Невозможно редактировать или создавать пользовательский список значений.

ⓘ Примечание

Если вложенные классы или объекты не описаны в хранимой процедуре, они скрыты. Статус их отображения изменить невозможно.

7.12.2.5 Хранимые процедуры с большим количеством результатов

Пример: хранимая процедура, возвращающая несколько наборов результатов. Во время разработки некоторые таблицы создаются в структуре юниверса, основанной на той же хранимой процедуре.

```
CREATE PROCEDURE qaputel.sp_getcustomer_2results
@location varchar(10)
AS
SELECT customer_key as KEYID, CUST_LNAME as Lname
FROM CUSTOMER
WHERE ADDRESS_LINE1 like @location
SELECT PREFIX as PREFIX, GENDER as GENDER, BIRTH_DT as BirthDATE
FROM CUSTOMER
```

Идея обработки набора с большим количеством результатов (RS):

Хранимая процедура RS1: a, b + RS2: b, d, e

Таблица A1: A, B

Таблица A2: B, D, E

Этот образец синтаксиса создает две таблицы, основанные на одном идентификаторе. Во время редактирования таблицы A1 в средстве создания юниверсов можно также редактировать таблицу A2.

Столбцы распределяются в соответствии со структурой набора результатов хранимой процедуры. Две таблицы созданы на основании одной хранимой процедуры. Структура набора результатов в

этом примере одинакова для обоих потоков. Средство создания юниверсов изменяет имя таблицы, полученное из второго набора результатов в хранимой процедуре. Средство создания юниверсов позволяет производить точную настройку бизнес-элементов.

Средство создания юниверсов создает в юниверсе одну таблицу для каждого набора результатов. Для каждой таблицы создается несколько независимых друг от друга соответствующих объектов. Можно изменять юниверс, как если бы это было нормальной хранимой процедурой.

7.13 Проверка юниверса

Можно проверить целостность объектов и классов в юниверсе, путем регулярной проверки с помощью команды "Проверить целостность" ("Инструменты" > "Проверить целостность"), и путем проверки объектов в Web Intelligence. Также можно просматривать SQL, нажав кнопку "Просмотр SQL", объекты которых создаются в запросе используя "Панель запросов" для создания запроса с объектами юниверса.

7.13.1 Проверка объектов в панели запросов

Можно просмотреть SQL для запроса с использованием панели запросов следующим образом:

1. Выберите "Инструменты" > "Панель запросов".
Откроется "Панель запросов".
2. Перетащите объекты на панель "Результаты" в правой части.
3. Нажмите кнопку "Параметры SQL".



4. Появится SQL для запроса.
5. Чтобы закрыть панель запросов нажмите кнопку "ОК", затем "Отмена".

7.13.2 Проверка целостности юниверса

Создание и изменение классов и объектов, требует регулярного использования функции "Проверка целостности", чтобы проверить целостность юниверса, которая использует проверку целостности. Дополнительные сведения об использовании функции "Проверка целостности" [Автоматическая Проверка целостности юниверса \[страница 190\]](#) см. в.

7.13.3 Проверка юниверса с помощью Web Intelligence

Можно проверить объекты с помощью проверки запросов в Web Intelligence. При проверке объектов можно задавать следующий тип вопросов:

- Объекты существуют? Если нет, сохранили ли вы юниверс после создания?
- SQL правильный?
- Результаты запросов правильны?

Можно также проверить объединения, с помощью оценки, если возвращенные значения правильны, и с помощью проверки компонентов схемы "Проверка целостности".

8 Оптимизация юниверсов

Время запроса может быть сокращено с помощью оптимизации юниверса. Существует несколько способов оптимизации юниверса:

- Оптимизация параметра "Выборка массива" в параметрах юниверса.
- Присвоение размера для каждой таблицы.
- Использование объединений быстрого доступа.
- Создание и использование агрегированных таблиц в базе данных.

Каждый из данных способов описан следующим образом.

8.1 Обзор

Повысить эффективность работы юниверсов можно с помощью следующих методик:

- [Использование таблиц агрегирования \[страница 356\]](#)
- [Использование функций @Function в SQL-коде для объекта \[страница 370\]](#)
- [Использование внешних стратегий для настройки процесса создания юниверса \[страница 399\]](#)
- [Использование аналитических функций \[страница 413\]](#)

8.2 Использование таблиц агрегирования

В свойствах средства создания юниверсов можно определить инструкцию Select для объекта с целью применения запроса к таблицам агрегирования, а не к базовым таблицам, в базе данных. Можно задать условия, которые позволят при оптимизации запроса применять его к таблицам агрегирования, а вне режима оптимизации – к базовым таблицам. Такая способность объекта использовать таблицы агрегирования для оптимизации запроса называется поддержкой агрегирования.

Данная глава содержит сведения о настройке поддержки агрегирования для юниверса.

8.2.1 Что такое поддержка агрегирования?

Поддержка агрегирования – это термин, который описывает способность юниверса использовать таблицы в базе данных. Это таблицы с заранее вычисленными данными. Можно использовать функцию @Aggregate_Aware в инструкции Select для объекта, который указывает на выполнение запроса в таблицах агрегирования, а не в таблицах с неагрегированными данными.

Использование таблиц агрегирования ускоряет выполнение запросов, повышая эффективность транзакций SQL.

Надежность и полезность поддержки агрегирования в юниверсе зависит от точности таблиц агрегирования. Они должны обновляться в одно время с реальными таблицами.

Юниверс, у которого один или несколько объектов с альтернативными определениями основан на таблицах агрегирования, является "поддержанным агрегированием". Данные определения относятся к уровням агрегирования. Например, объект "Доход" может быть агрегирован с помощью месяца, квартала или года. Данные объекты называются объектами агрегирования.

Запросы, созданные в юниверсе с помощью объектов агрегирования, возвращают сведения, агрегированные на соответствующем уровне при оптимальной скорости.

8.2.2 Применение поддержки агрегирования к хранилищам данных

Поддержка агрегирования обычно используется во время работы с хранилищами данных. Рассмотрим хранилище данных, состоящее из трех измерений: время, география и продукт..

На низшем уровне хранилище данных хранит сведения о клиентах и продуктах за день. Существует строка для каждых ежедневных покупок клиента. Она выглядит следующим образом.

365 дней x 100 городов x 10 продуктов = 365 000 строк.

При запросе сведений о продажах за год процессор базы данных добавляет большое количество строк. Однако годовые продажи компаний могут включать меньше строк, как показано ниже.

3 года x 3 страны x 3 компании = 27 строк

В данном примере 27 строк в таблице достаточно для ответа на вопрос. На основе данной информации будет более эффективно заранее суммировать данные строки в таблицах агрегирования.

8.2.3 Установка поддержки агрегирования

Настройка поддержки агрегирования в юниверсе выполняется в четыре шага. Главные шаги данной методологии изложены ниже.

- Создание объектов:
 1. Найдите все возможные определения (таблица/комбинация столбцов) объектов.
 2. Упорядочьте объекты по уровню агрегирования.
 3. Создайте объекты с помощью функции `Aggregate_Awareness`.
- Определение несовместимости объектов:
 1. Создайте матрицу объектов/таблиц агрегирования.
 2. Для первой таблицы агрегирования необходимо определить, будут ли ее объекты совместимы (идентичный уровень агрегации или более высокий) или нет (более низкий уровень агрегации).
 3. Установите флажки для объектов, которые будут несовместимы с данной таблицей.

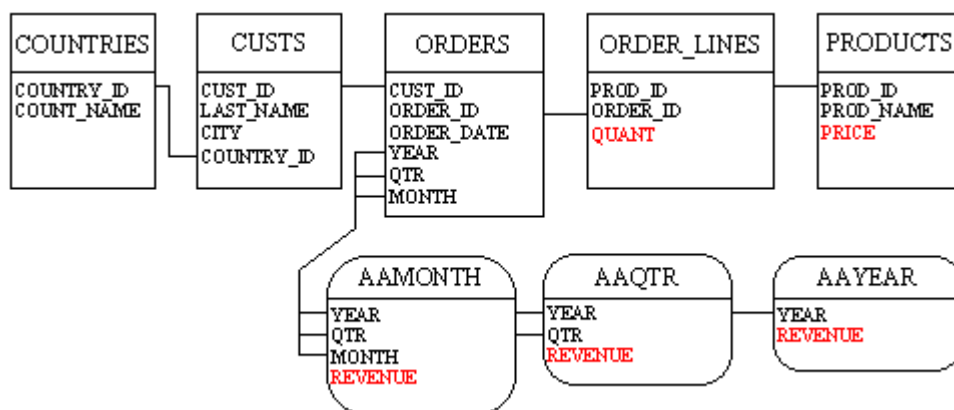
4. Повторите шаги 1 и 3 для оставшихся таблиц агрегирования.

- Создайте необходимые контексты

Создайте один контекст на уровне агрегирования.

- Проверьте результаты
1. Выполните несколько запросов.
 2. Сравните результаты.

Каждый этап вышеуказанного процесса описан в следующих разделах. Схема-пример, указанная ниже, используется для иллюстрации каждого этапа.



В схеме-примере содержатся три ранее определенных таблицы агрегирования: AAMONTH, AAQTR и AAYEAR.

❗ Примечание

Схема-пример не является образцом типичной схемы. Используйте ее для установки поддержки агрегирования. В схеме производительности таблица агрегирования объединяет несколько измерений, а не одно измерение на основе времени. Хронологическое измерение (год, квартал и месяц) определяется в основной таблице, а не в таблице агрегирования.

8.2.4 Создание объектов

Первым шагом в установке поддержки агрегирования в юниверсе является определение, для каких объектов необходима поддержка агрегирования. Можно использовать объекты мер и объекты измерений.

Объект "Доход от продаж" имеет следующее определение на основе вышеуказанной схемы.

`PRODUCTS.PRICE*ORDER_LINES.QUANT`

Необходимо заново определить Sales_Revenue, чтобы использовать таблицы агрегирования, где возможно, вместо выполнения агрегирования с помощью неагрегированных таблиц.

На каждом этапе, который выполняется для повторного определения функции "Доход от продаж" в качестве функции, подходящей для поддержки агрегирования, необходимо выполнить данные этапы для других объектов, в определении которых используются таблицы агрегирования.

8.2.5 Определение всех комбинаций объектов агрегирования

Необходимо определить все возможные совокупности объектов в различных таблицах. Объект "Доход от продаж" можно определить следующим образом.

- AAMONTH.REVENUE
- AAYEAR.REVENUE
- AAQTR.REVENUE
- PRODUCTS.PRICE*ORDER_LINES.QUANT

8.2.6 Расположение объектов в порядке агрегирования

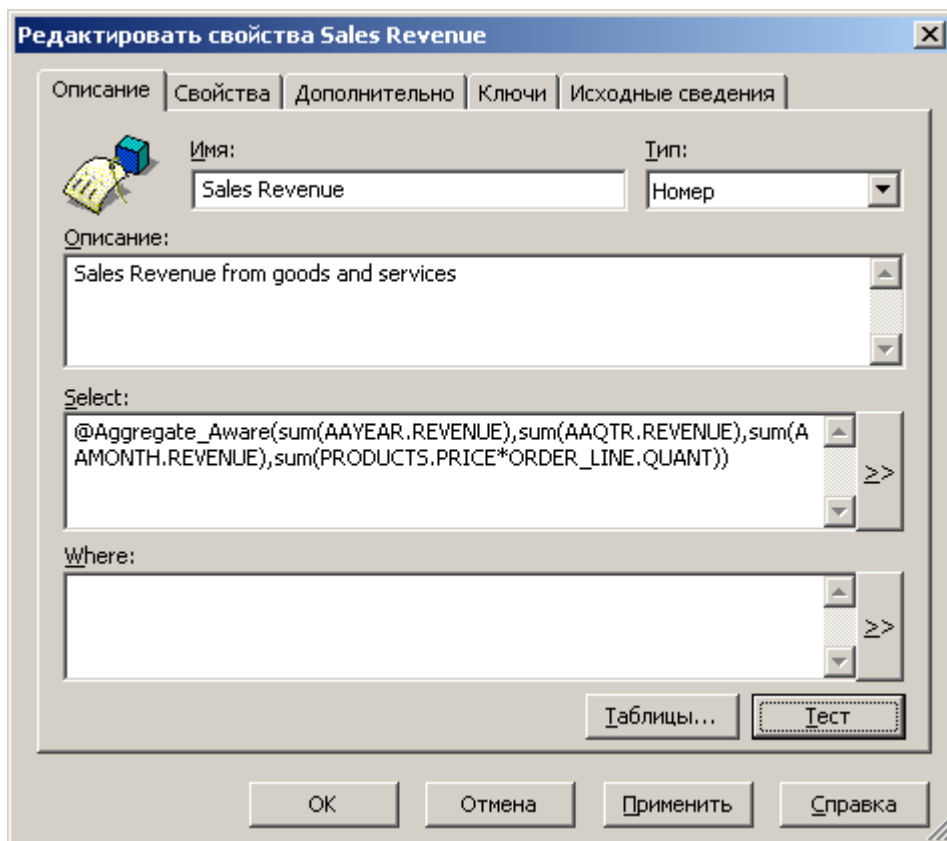
Определив все совокупности объектов, расположите их согласно уровням агрегирования следующим образом.

- AAYEAR.REVENUE – наивысший уровень агрегирования.
- AAQTR.REVENUE – следующий уровень.
- AAMONTH.REVENUE – следующий уровень.
- PRODUCTS.PRICE*ORDER_LINES.QUANT – это самый низкий уровень агрегирования.

8.2.7 Определение объектов агрегирования с помощью функции @Aggregate_Aware

Необходимо заново определить инструкцию Select с помощью функции @Aggregate_Aware для всех объектов с поддержкой агрегирования. Функция @Aggregate_Aware сначала направляет объект для выполнения запроса всех таблиц агрегирования в качестве параметров объекта. Если таблицы агрегирования не соответствуют, тогда запрос выполняется с помощью исходного агрегирования на основе неагрегированных таблиц. Дополнительные сведения о функциях @Function см. в разделе [Использование функций @Function в SQL-коде для объекта \[страница 370\]](#).

Инструкция Select для "Доход от продаж" с использованием функции @Aggregate_Aware отображается ниже.



Синтаксис функции @Aggregate_Aware выглядит следующим образом.

```
@Aggregate_Aware(sum(agg_table_1), ... sum(agg_table_n))
```

где `agg_table_1` – это таблица с самым высоким уровнем агрегирования, а `agg_table_n` – с самым низким.

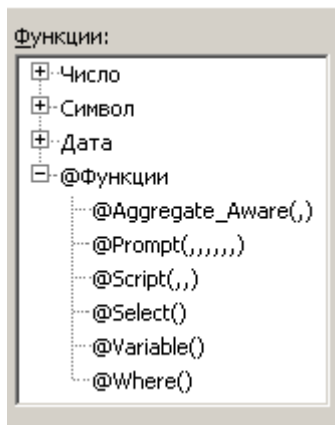
Необходимо ввести имена таблиц агрегирования в качестве аргументов. Можно перемещать имена таблиц слева направо в порядке агрегирования по убыванию.

8.2.7.1 Определение объекта с помощью функции @Aggregate_Aware

Чтобы повторно определить объект с помощью функции @Aggregate_Aware,

1. Дважды щелкните объект.
Откроется диалоговое окно объекта *Редактировать свойства*.
2. Нажмите кнопку >> рядом с полем *Select*.
Откроется диалоговое окно *Редактировать инструкцию Select*.
3. Щелкните начало инструкции SELECT.
Или
Щелкните поле выбора, если у объекта все еще нет инструкции SELECT.
Курсор автоматически появится в левом верхнем углу поля.

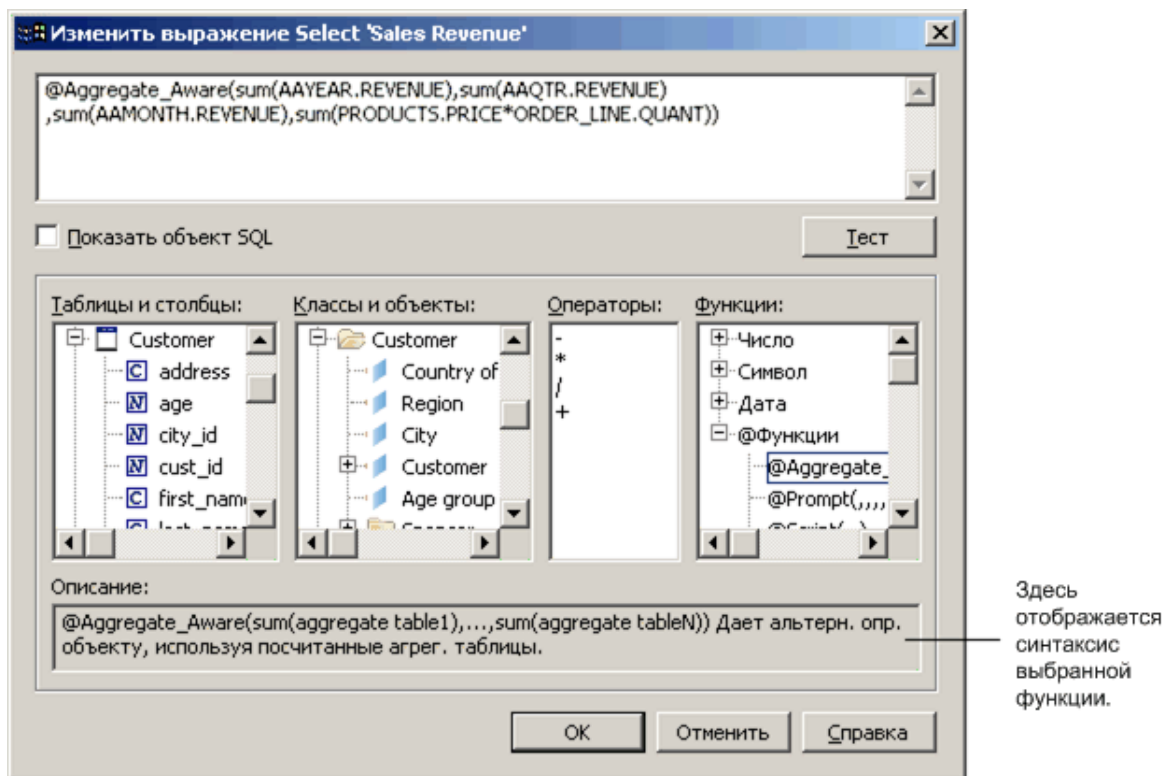
- Щелкните узел функций @Function на панели \Функции.
Отобразится список доступных функций @function.



- Дважды щелкните Aggregate_Aware.
Синтаксис функции @Aggregate_Aware будет вставлен в инструкцию Select. Описание синтаксиса появится в поле [Описание](#) в нижней части диалогового окна. Это может помочь указать параметры для @function.
- Расположите агрегации функции @AggregateAware в порядке в скобках (от самого высокого до самого низкого уровня данных агрегирования).
- Отделите каждую агрегацию запятой. Например, синтаксис "Доход от продаж" выглядит следующим образом.

```
@Aggregate_Aware(сумма (AAYEAR.REVENUE), сумма (AAQTR.REVENUE), сумма (AAMONTH.REVENUE), сумма (PRODUCTS.PRICE*ORDER_LINES.QUANT))
```

- Чтобы проверить синтаксис, нажмите [Разбор](#).
Страница [Редактировать выражение Select редактора SQL](#) для дохода от продаж отображена ниже.



9. В каждом из диалоговых окон нажмите кнопку **OK**.
В данном примере необходимо повторно определить объекты измерений года и квартала с помощью функции @Aggregate_Aware.

8.2.8 Указание несовместимых объектов

Для каждой таблицы агрегирования в юниверсе необходимо указать несовместимые объекты. Набор указанных несовместимых объектов определяет, какие таблицы агрегирования игнорируются в процессе генерации SQL.

С учетом таблицы агрегирования объект может быть совместимым и несовместимым. Правила совместимости выглядят следующим образом.

- Если объект находится на том же уровне агрегирования, как и таблица, или выше, он совместим с таблицей.
- Если объект находится на уровне агрегирования ниже, чем таблица (или совсем не относится к таблице), он несовместим с таблицей.

8.2.8.1 Использование матрицы в анализе объектов

Может оказаться полезным создать матрицу, чтобы анализировать совместимость объектов и таблиц агрегирования. В первых двух столбцах матрицы можно перечислить имена классов и объектов. Можно

создать столбец с заголовком для каждой таблицы агрегирования в юниверсе. Пустая матрица на основе схемы-примера выглядит следующим образом.

Класс	Объект	AAYEAR	AAQTR	AAMONTH
Клиенты	Код клиента			
	(CUSTOMER.CUST_ID)			
	Имя клиента			
	(CUSTOMER.LAST_NAME)			
Продукты	Город клиента			
	(CUSTOMER.CITY)			
	Национальность клиента			
	(COUNTRIES.COUNT_NAME)			
Заказы	Код продукта			
	(PRODUCT.PROD_ID)			
	Название продукта			
	(PRODUCT.PROD_NAME)			
Продажи	Год заказа			
	(AAYEAR.PROD_NAME)			
	Квартал заказа			
	(AAQTR.QTR)			
Показатель	Месяц заказа			
	(AAMONTH.MONTH)			
	Дата заказа			
	(ORDERS.ORDER_DATE)			
Показатель	Доход с продаж			
	(@Aggregate_Aware(...))			

Введите X для каждой таблицы, если объект является несовместимым.

Завершенная матрица на основе примера отображена ниже.

Класс	Объект	AAYEAR	AAQTR	AAMONTH
Клиенты	Код клиента (CUSTOMER.CUST_ID)	X (n)	X (n)	X (n)
	Имя клиента (CUSTOMER.LAST_NAME)	X (n)	X (n)	X (n)
	Город клиента (CUSTOMER.CITY)	X (n)	X (n)	X (n)
	Национальность клиента (COUNTRIES.COUNTRY_NAME)	X (n)	X (n)	X (n)
Продукты	Код продукта (PRODUCT.PROD_ID)	X (n)	X (n)	X (n)
	Название продукта (PRODUCT.PROD_NAME)	X (n)	X (n)	X (n)
Заказы	Год заказа (AAYEAR.PROD_NAME)	- (s)	- (h)	- (h)
	Квартал заказа (AAQTR.QTR)	X (l)	- (s)	- (h)
	Месяц заказа (AAMONTH.MONTH)	X (l)	3 (l)	- (s)
	Дата заказа (ORDERS.ORDER_DATE)	X (l)	X (l)	X (l)
Продажи	Доход с продаж	-	-	-
Показатель	(@Aggregate_Aware(...))			

X (n): данный объект не связан с таблицей агрегирования. Таким образом, он является совместимым.

X (l): данный объект имеет уровень агрегирования ниже, чем данная таблица агрегирования; он не может использоваться для извлечения информации. Таким образом, он не является совместимым.

- (s): данный объект имеет уровень агрегирования, идентичный уровню агрегирования данной таблицы агрегирования; он может использоваться для извлечения информации. Таким образом, он является совместимым.

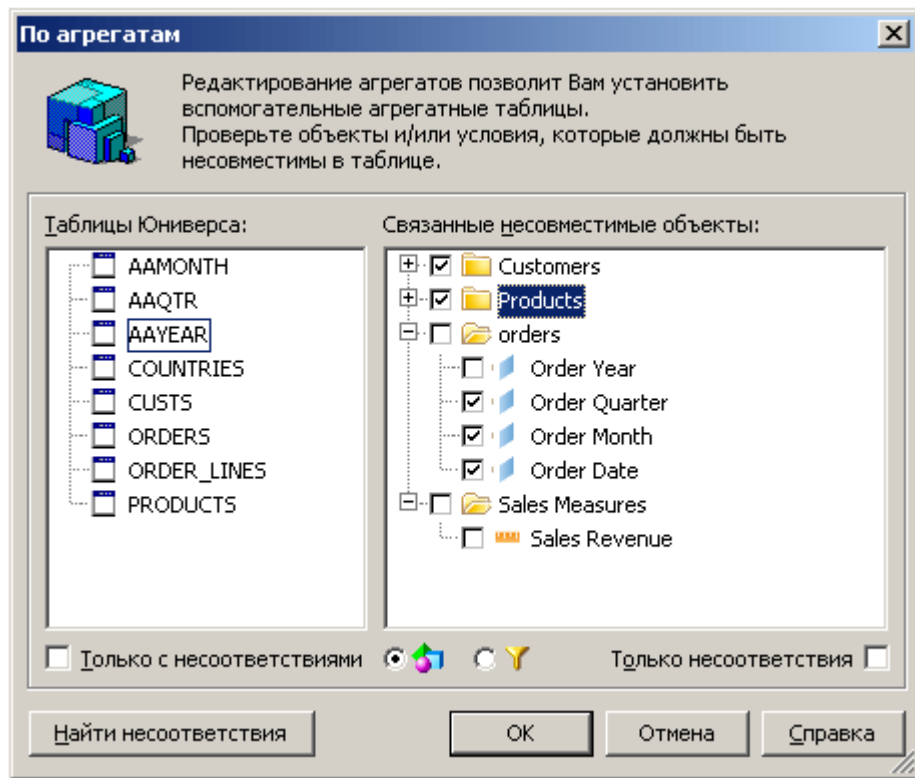
- (s): данный объект имеет уровень агрегирования, превышающий уровень агрегирования данной таблицы агрегирования; он может использоваться для извлечения информации. Таким образом, он является совместимым.

8.2.9 Указание несовместимых объектов

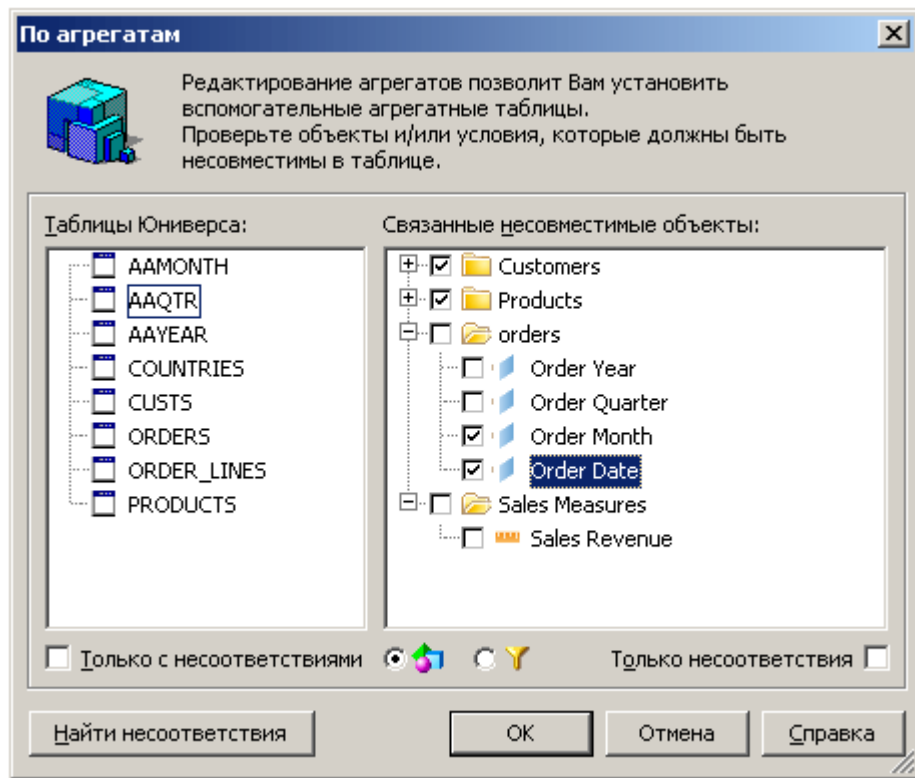
Теперь несовместимые объекты можно указывать. С помощью диалогового окна [Навигация агрегирования](#) (► [Инструменты](#) ► [Навигация агрегирования](#) ►) можно указать несовместимые объекты.

Для этого в диалоговом окне [Навигация агрегирования](#) выполните следующие действия.

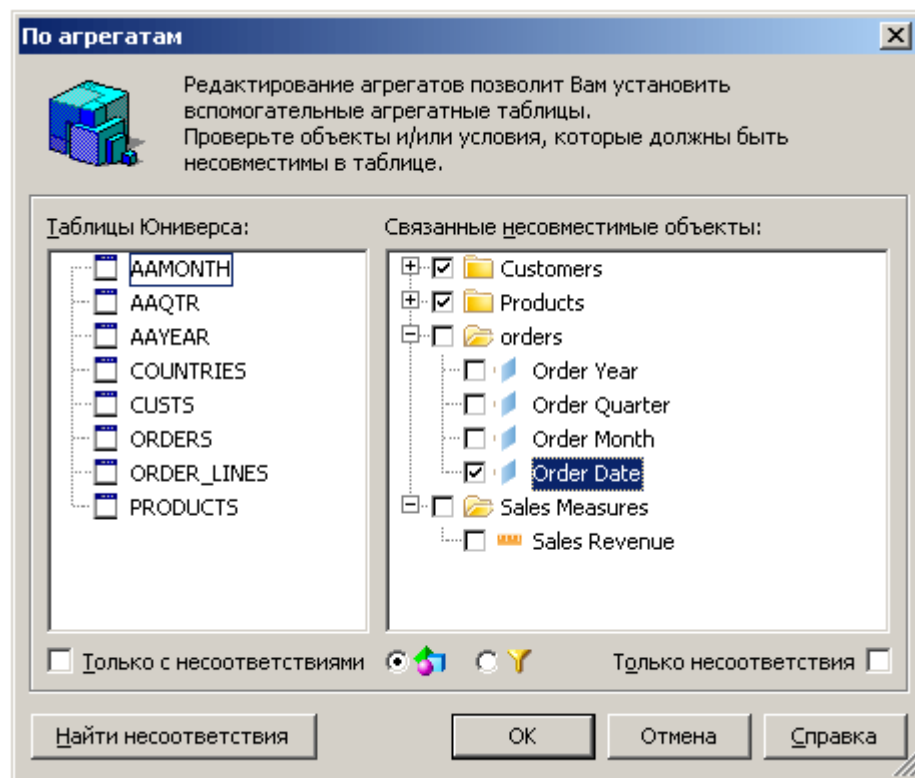
1. Выберите ► [Инструменты](#) ► [Навигация агрегирования](#) ►.
Откроется диалоговое окно [Навигация агрегирования](#). Оно состоит из двух панелей.
 - [Таблицы юниверса](#), которая отображает все таблицы юниверса.
 - [Связанные несовместимые объекты](#), которая отображает все объекты юниверса.
2. Выберите таблицу агрегирования в левой панели.
3. Установите флажок для каждого несовместимого объекта в правой панели.
Например, на основе матрицы, для таблицы AAYEAR все объекты в классе "Клиенты" являются несовместимыми. Установите флажок рядом с именем класса, как показано ниже.



4. Выполните повторно эти действия для каждой таблицы агрегирования в универсе. Несовместимые объекты для таблицы AAQTR отображены ниже.



Для таблицы AAMONTH только один объект является несовместимым.



5. Нажмите **OK** после указания несовместимых объектов для всех таблиц.

❗ Примечание

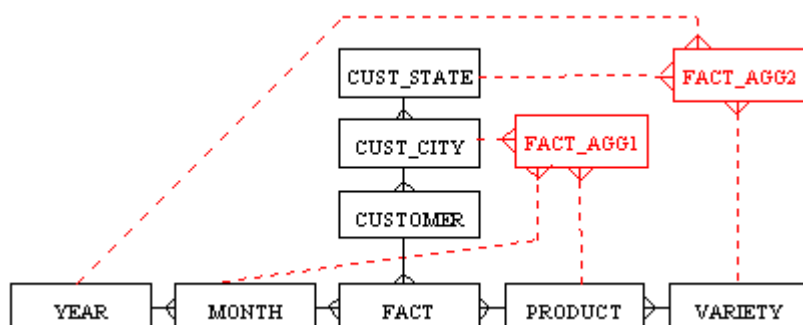
Помощником в процессе указания несовместимых объектов в диалоговом окне также является кнопка **Найти несовместимость**. Если щелкнуть таблицу, а затем нажать эту кнопку, средство создания юниверсов автоматически проверит объекты, которые были определены как несовместимые. Необходимо просматривать предварительно выбранные несовместимые объекты, предложенные инструментом **Найти несовместимость**, а не итоговый список объектов.

8.2.10 Устранение циклов с помощью таблиц агрегирования

Если в базе данных содержится одна или несколько таблиц агрегирования, необходимо устранять циклы с помощью контекстов.

Пример: Устранение цикла с таблицей агрегирования

Ниже изображена простая схема с таблицей агрегирования.



Отметьте следующие пункты в схеме.

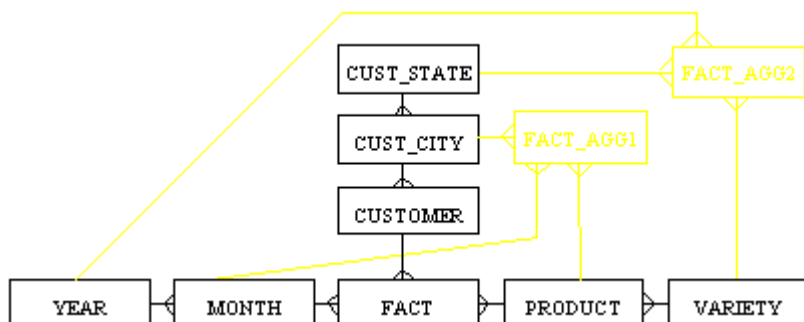
- FACT_AGG1 – это таблица агрегирования, идентичная РЕАЛЬНОЙ таблице. В ней содержится ключ поля "Город" (Клиент), "Продукт" и "Месяц" в дополнение к количеству мер агрегированных для "Состояние клиента", "Продукт" и "Месяц".
- FACT_AGG2 – это также таблица агрегирования, идентичная РЕАЛЬНОЙ таблице. Ее меры агрегированы для "Состояние клиента", "Продукт" и "Год".
- Меры (ключевые индикаторы производительности) хранятся во всех реальных таблицах. Сведения о доходе от продаж хранятся в таблицах FACT_AGG1, FACT_AGG2 и FACT, но агрегированы на соответствующих уровнях каждой таблицы.

Для запроса с продажами "Доход" и "Состояние клиента" необходимо использовать объединение между CUST_STATE и FACT_AGG2 вместо объединения между CUST_STATE и CUST_CITY.

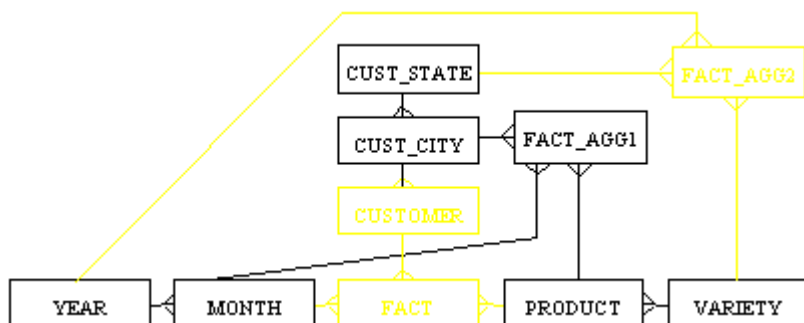
Однако перед выполнением запроса необходимо определить три контекста, например FACT, FACT_AGG1 и FACT_AGG2. Переименовывать контекст с помощью более значимых меток не требуется, так как они прозрачны для пользователей.

На следующей странице отображено объединение, включенное в три контекста. В каждой схеме более темный набор объединений представляет данный контекст.

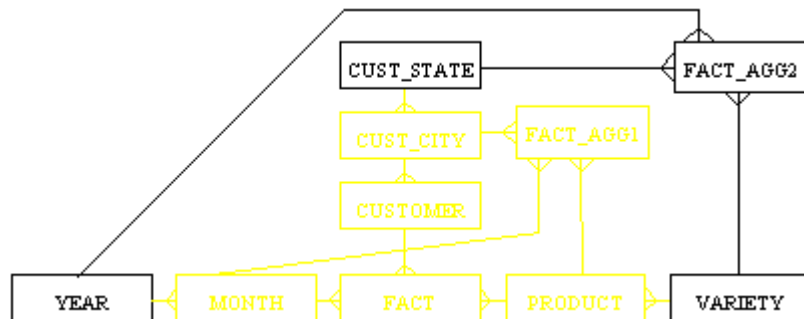
Контекст FACT



Контекст FACT_AGG1



Контекст FACT_AGG2



8.2.11 Тестирование поддержки агрегирования

Последний шаг в установке поддержки агрегирования – проверка результатов в Web Intelligence.

Опираясь на первый пример, можно выполнить следующие запросы и сравнить различные результаты.

- Годовой заказ и доход от продаж.
- Квартальный заказ и доход от продаж.
- Месячный заказ и доход от продаж.
- Заказчик и доход от продаж.
- Товар и доход от продаж.

8.3 Использование функций @Function в SQL-коде для объекта

@Function – это особые функции, которые предоставляют удобные способы указания SQL для объекта. @Function доступны на панели [Функции](#) диалогового окна [Редактировать выражение Select](#) для объекта.

Можно объединить одну или несколько функций @Function в инструкции SELECT или утверждении WHERE объекта. Доступны следующие функции @Function.

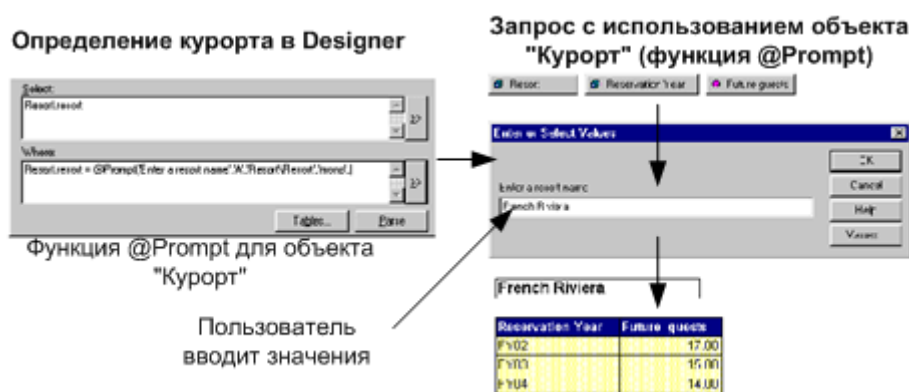
@Function	Описание	Обычно используется в объекте
@Aggregate_Aware	Объединяет столбцы, содержащие агрегированные и данные измерений в объектах.	Инструкция SELECT
@Prompt	Вставляет подсказку в код SQL. Когда пользователь выполняет запрос, ему выдается подсказка значения для ограничения каждый раз, когда объект, использующий функцию @Prompt, включается в запрос.	Инструкция SELECT Утверждение WHERE
@Script	Запускает скрипт каждый раз, когда объект, использующий функцию @Script, включен в запрос.	Утверждение WHERE
@Select	Позволяет использовать инструкцию SELECT другого объекта.	Инструкция SELECT
@Variable	Вызывает значение переменной, хранящееся в памяти. Например во влияющем текстовом файле.	Утверждение WHERE

@Function	Описание	Обычно используется в объекте
@Where	Позволяет использовать утверждение WHERE другого объекта.	Утверждение WHERE

Пример: Использование функции @Prompt, чтобы ограничить возвращенные значения для введенного значения подсказки.

Функция @Prompt – это одна из функций @Functions, доступных в средстве создания юниверсов. Можно использовать функцию @Prompt, чтобы отобразить окно сообщения, если объект используется в запросе Web Intelligence.

Окно сообщения предлагает пользователю ввести значение объекта. Запрос возвращает значения введенного значения подсказки, как показано ниже.

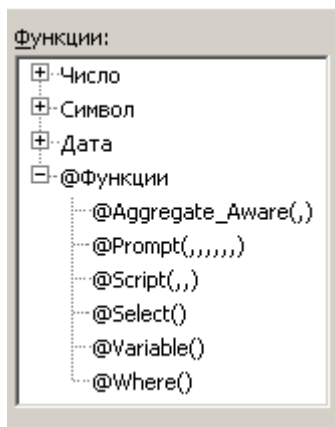


8.3.1 Вставка функции @Function в объект

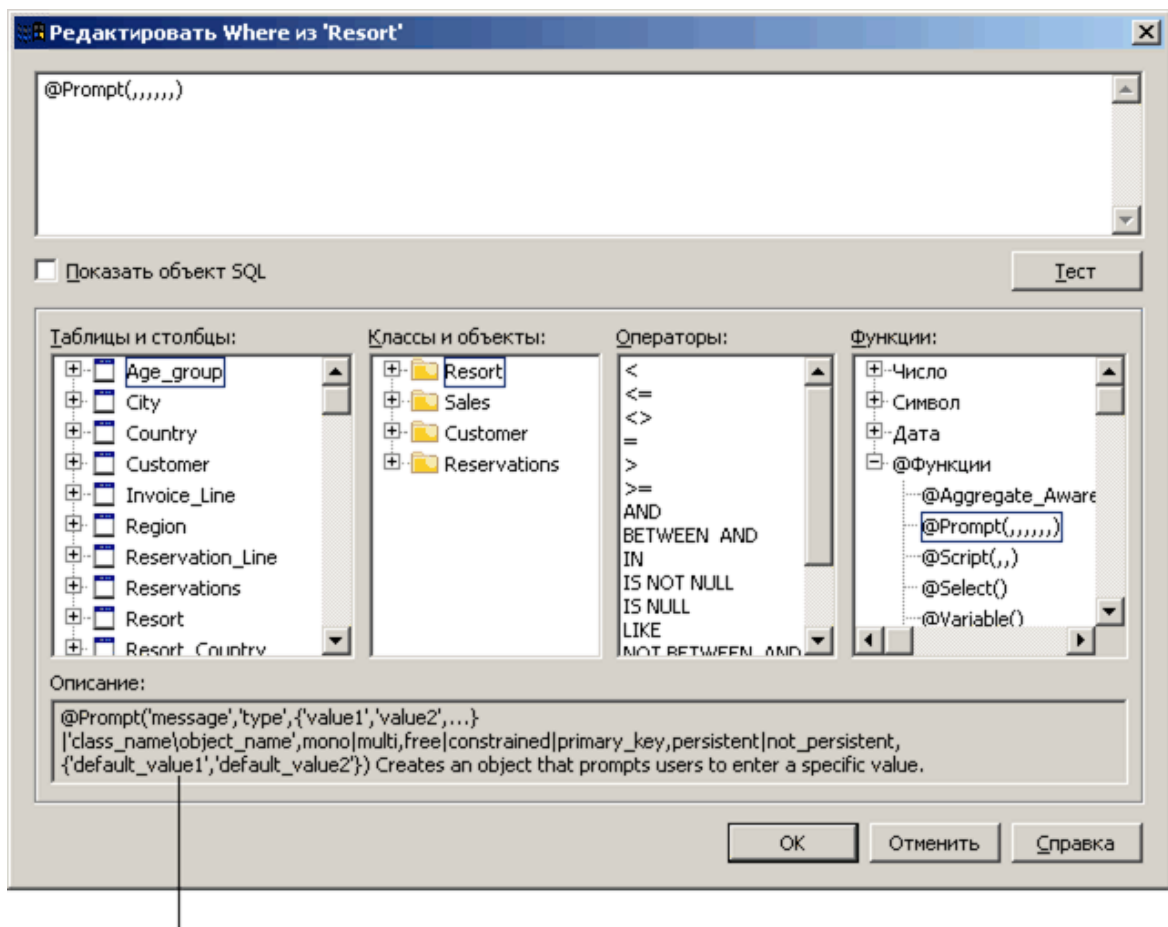
Чтобы вставить функцию @Function в определение SQL для объекта, выполните следующие действия.

- На [панели юниверса](#) дважды нажмите объект. Появится диалоговое окно объекта [Редактировать свойства](#).
- Нажмите кнопку >> рядом с полем [Select](#).
Или
Нажмите кнопку >> рядом с полем [Where](#).
Появится диалоговое окно [Редактировать инструкцию Select](#) или [Редактировать утверждение Where](#).
- Щелкните в том месте инструкции select или утверждения where, где необходимо добавить функцию @Function. Если поле пусто, как показано выше, нажмите любое место в поле. Курсор автоматически появится в левом верхнем углу поля.

4. Click the [@Functions](#) node in the [Functions](#) pane.
Отобразится список доступных функций [@Function](#).



5. Дважды щелкните функцию [@Function](#).
Синтаксис функции [@Function](#) будет добавлен в инструкцию SELECT или утверждение WHERE объекта. Описание синтаксиса появится в поле [Описание](#) в нижней части диалогового окна. Оно может помочь при вводе параметров функции [@Function](#).



6. Введите необходимые параметры в верхней панели диалогового окна.
7. Чтобы проверить синтаксис, нажмите [Разбор](#).
8. В каждом из диалоговых окон нажмите кнопку [OK](#).

8.3.2 Функция @Aggregate_Aware

Функция `@Aggregate_Aware` позволяет объекту использовать преимущества таблиц содержащих сводку данных в базе данных. Если база данных содержит сводку таблиц, а работа ведется с запросами, возвращающими агрегированные данные, то более быстрым решением является применение инструкции `SELECT` к столбцам, содержащим сводку данных, а не к столбцам, содержащим данные о фактах или событиях. В объектах, которые несовместимы с агрегированными таблицами, невозможно использовать агрегированные таблицы, вместо этого для запроса можно использовать основные таблицы.

Можно использовать функцию `@Aggregate_Aware` чтобы настроить поддержку агрегирования в юниверсе. Данный процесс включает несколько других шагов, связанных с использованием функции `@Aggregate_Aware`.

- Укажите несовместимые объекты для каждой агрегированной таблицы.
- Устраните все циклы в агрегированных таблицах.
- Проверьте агрегированные таблицы, чтобы убедиться, что они возвращают правильные результаты.

8.3.2.1 Синтаксис функции @Aggregate_Aware

Синтаксис функции @Aggregate_Aware является следующим:

```
@Aggregate_Aware(sum(agg_table_1), ...
                 sum(agg_table_n))
```

Необходимо указать имена всех таблиц агрегирования в виде аргументов. Разместите имена таблиц слева направо по убыванию относительно порядка агрегирования.

Синтаксис	Описание
agg_table_1	является таблицей агрегирования с наиболее высоким уровнем агрегирования.
agg_table_n	является таблицей агрегирования с наиболее низким уровнем агрегирования.

Пример

```
@Aggregate_Aware(    R_Country.Revenue,
                    R_Region.Revenue,
                    R_City.Revenue,
                    R_Customer.Revenue,
                    R_Age_Range.Revenue,
                    sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
                    )
```

В данном примере, когда объект, использующий эту функцию, объявляется несовместимым со всеми прочими таблицами, эти таблицы игнорируются. Для объекта, который несовместим с таблицами R_Country и R_Region, используемый код SQL принимает следующий вид:

```
@Aggregate_Aware(    R_City.Revenue,
                    R_Customer.Revenue,
                    R_Age_Range.Revenue,
                    sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
                    )
```

8.3.3 @Prompt

Функция @Prompt используется для вставки подсказки в запрос. Подсказки применяются для ограничения данных или облегчения использования объектов с большими значениями, когда пользователь создает отчет. Функция @Prompt используется в инструкции SELECT или утверждении

WHERE для объекта. Данная функция обязывает пользователя вводить одно или несколько значений для ограничения всякий раз, когда объект используется в запросе, либо выбирать значение или список значений. При выполнении запроса отображается окно подсказки, запрашивающее пользователя ввести значение.

@Promptsледует использовать, когда необходимо установить ограничение в предполагаемой таблице, не задавая значения условия.

При необходимости можно определить значения для подсказки по умолчанию. Подсказка, содержащая значения по умолчанию, выполняется так же, как и подсказка Web Intelligence, содержащая значения по умолчанию.

Определение @Prompt можно редактировать следующими способами:

- с помощью редактора @Prompt;
- путем ввода определения для условия в поле *SELECT* или *WHERE* на панели *Определение* диалогового окна *Изменить свойства*;
- с помощью ввода определения на панели редактирования *Дополнительно* диалогового окна *Изменить свойства*.

❗ Примечание

Если используются две похожих подсказки, например, с различными первичными ключами, не используйте одинаковые вопросы (текст подсказки). Это связано с тем, что в таком случае системе не удастся различить эти подсказки, что приведет к выдаче несоответствующего ответа.

❗ Примечание

В связи с потенциальной сложностью определения @Prompt, особенно при заполнении списка значений, рекомендуется использовать редактор @Prompt.

❗ Примечание

Можно объединять функцию @Variable с функцией @Prompt в одном запросе, когда функция @Prompt возвращает одно значение.

Связанные сведения

[Редактор @Prompt \[страница 375\]](#)

[Редактирование существующего выражения @Prompt \[страница 381\]](#)

[Синтаксис функции @Prompt \[страница 388\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

8.3.3.1 Редактор @Prompt

С помощью редактора @Prompt можно определить или отредактировать подсказки, которые отображаются при использовании инструментов клиента BI, таких как Web Intelligence, для выполнения

запроса к реляционным юниверсам или юниверсам OLAP. Редактор упрощает процесс определения и изменения подсказок. Определение подсказки отображается в нижней части панели и автоматически обновляется при определении для подсказки других значений. Если щелкнуть правой кнопкой мыши существующее выражение @Prompt, элемент меню *Редактировать @Prompt* активен, только если синтаксис строки @Prompt является правильным.

Также можно ввести определение @Prompt непосредственно на панели оператора *SELECT* или *WHERE* диалогового окна *Изменить свойства*.

Связанные сведения

[Синтаксис функции @Prompt \[страница 388\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

[Определение статического списка значений для подсказки \[страница 379\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Выбор пользовательского списка значений для подсказки \[страница 380\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Редактирование существующего выражения @Prompt \[страница 381\]](#)

8.3.3.2 Свойства выражения @Prompt для редактора @Prompt

Можно изменить следующие свойства для подсказки.

Свойство	Описание
Сообщение	Сообщение подсказки, которое видит пользователь. Например, можно ввести "Выберите страну". Текст подсказки (вопрос). По умолчанию = Ввести значения
Тип значения	Тип данных, который вводит и выбирает пользователь. Таким образом, тип данных, вводимых или выбираемых пользователем, будет точно правильным. Выберите из: <ul style="list-style-type: none">• Буквенно-цифровой (A)• Числовой (N)• Дата (D) По умолчанию = Буквенно-цифровой

Свойство	Описание
Тип ключа	<p>Если в качестве режима выбора установлено значение "Primary_key", задайте тип ключа, вводимого или выбираемого пользователем. Выберите из:</p> <ul style="list-style-type: none"> • Буквенно-цифровой (A) • Числовой (N) • Дата (D) <p>По умолчанию = Нет</p>
Разрешить множественный выбор	<p>Если этот параметр установлен, пользователь может ввести или выбрать более одного значения.</p> <p>По умолчанию = Не выбран – пользователь может ввести или выбрать только одно значение</p>
Режим выбора	<p>Определите, как пользователь выбирает значения, необходимые для подсказки. Выберите:</p> <ul style="list-style-type: none"> • Свободно: пользователь может ввести любое значение • Ограничено: пользователю необходимо выбрать из предложенных значений • Primary_key*. Использование значения первичного ключа в запросе существенно увеличивает время отклика. Можно выбрать или ввести первичные ключи для объектов в юниверсе. Во время подсказки пользователь выбирает имя объекта, но на самом деле запрос использует соответствующее значение первичного ключа. *Можно использовать установку первичного ключа, если "Поддержка индекса" установлена в таблицах базы данных. <p>По умолчанию = Свободно</p>
Сохранить последнее выбранное значение	<p>Если этот параметр установлен, при следующем выполнении запроса предлагаются значения, использовавшиеся в предыдущий раз. Если этот параметр не установлен, всегда используются значения по умолчанию.</p> <p>По умолчанию = Не выбран</p>
Отображаемые значения	<p>Пользователю необходимо выбрать один или несколько элементов в списке значений. Список значений можно определить следующими способами:</p> <ul style="list-style-type: none"> • Введите список значений в окне "Список значений" (это статический список) • Выберите объект в юниверсе • Импортируйте файл с помощью мастера ввода файлов <p>По умолчанию = Статический</p>
Значения по умолчанию	Можно определить значения по умолчанию для подсказки.

Связанные сведения

[Синтаксис функции @Prompt \[страница 388\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

[Определение статического списка значений для подсказки \[страница 379\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Выбор пользовательского списка значений для подсказки \[страница 380\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Редактирование существующего выражения @Prompt \[страница 381\]](#)

8.3.3.3 Использование редактора подсказок для создания выражения @Prompt

Юниверс открыт в средстве создания юниверсов, и выполняется создание или редактирование объекта.

1. В поле выражения *SELECT* или *WHERE* условия щелкните правой кнопкой мыши там, куда необходимо добавить подсказку, и выберите *Редактор @Prompt* в меню быстрого вызова.
2. В окне *Сообщение* введите сообщение, которое должен увидеть пользователь.
3. Выберите *Сохранить последнее выбранное значение*, если необходимо, чтобы подсказка предлагала значения, использованные для предыдущего отчета.
При первом запуске отчета предлагаются значения по умолчанию (если такие имеются).
4. Выберите *Разрешить множественный выбор*, если пользователь может ввести или выбрать несколько значений.
5. Установите *Режим выбора*. Для выбора *Свободно* пользователь может ввести любое разрешенное значение. Для выбора *Ограничено* пользователь должен выбрать значение из списка. Для *первичного ключа* пользователь выбирает имя объекта, но первичный ключ объекта используется запросом. Параметр *Первичный ключ* можно использовать, только если установлена поддержка индекса.

❗ Примечание

Если выбран параметр *Ограничено* и не указан список значений, цвет соответствующей вкладки изменяется на красный, кнопка "ОК" отключается и при наведении курсора на выделенную вкладку отображается контекстная подсказка о том, как решить эту проблему.

6. Установите для режима выбора *Тип значения*: *Буквенно-цифровой*, *Числовой* или *Дата*.
7. Если для режима выбора указан *Первичный ключ*, задайте следующий *Тип ключа*: *Буквенно-цифровой*, *Числовой* или *Дата*.
8. При выборе нескольких значений определите их список. Для этого можно ввести или импортировать список значений либо выбрать объект юниверса.
9. Выберите вкладку *Значения по умолчанию* и определите значения по умолчанию. Если допускается выбор нескольких значений, можно указать более одного значения по умолчанию.
10. Нажмите кнопку *ОК*.
Редактор *@Prompt* закроется, а выражение подсказки отобразится в инструкции условия на панели *Изменить свойства*.
11. Проверьте подсказку и нажмите *Синтаксический разбор*.
Если синтаксис является неверным, отображается сообщение *Сбой при синтаксическом разборе* с фрагментом SQL, в котором содержится ошибка.

Связанные сведения

[Синтаксис функции @Prompt \[страница 388\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

[Определение статического списка значений для подсказки \[страница 379\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Выбор пользовательского списка значений для подсказки \[страница 380\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Редактирование существующего выражения @Prompt \[страница 381\]](#)

8.3.3.4 Определение статического списка значений для подсказки

В нижней части редактора *@Prompt* содержится таблица, которая используется для определения статического списка значений. Можно использовать стрелки вверх и вниз, чтобы изменять положение введенных значений. Если текст заголовка (*Надпись*) выделен красным, необходимо ввести или исправить значения.

1. Введите первое значение в поле *Надпись*. При выборе *основного ключа* в качестве *режима выбора* введите значение индекса во второе поле.
2. Нажмите **+**, чтобы вставить значение в таблицу статического списка значений.
3. Введите несколько значений в таблицу, чтобы заполнить список значений.
4. Введите любое значение по умолчанию на панели *Значения по умолчанию*. Значения по умолчанию предлагаются при запуске отчета. Если установлен флажок *Сохранить последнее выбранное значение*, предлагаются значения, используемые при последнем запуске отчета. В противном случае при каждом запуске отчета будут предлагаться значения по умолчанию.

📘 Примечание

Чтобы изменить значение в статическом списке, выберите его, затем измените значение в таблице и нажмите кнопку обновления.

📘 Примечание

Чтобы удалить значение, выберите значение и нажмите **-**.

Связанные сведения

[Синтаксис функции @Prompt \[страница 388\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Выбор пользовательского списка значений для подсказки \[страница 380\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Редактирование существующего выражения @Prompt \[страница 381\]](#)

8.3.3.5 Выбор объекта юниверса в качестве списка значений для подсказки

Можно выбрать объект юниверса в качестве списка значений для подсказки.

📘 Примечание

При создании ограничения на доступ к строке нельзя использовать *объект юниверса* в редакторе *@Prompt* или функции.

1. В редакторе *@Prompt* выберите *Объект юниверса* на панели списка значений.
2. Перейдите к объекту юниверса, который необходимо использовать в подсказке, и выберите его.
3. Дважды щелкните объект.
Редактор *@Prompt* закроется, а выражение подсказки отобразится в инструкции условия на панели *Изменить свойства*.

Связанные сведения

[Синтаксис функции @Prompt \[страница 388\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

[Определение статического списка значений для подсказки \[страница 379\]](#)

[Выбор пользовательского списка значений для подсказки \[страница 380\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Редактирование существующего выражения @Prompt \[страница 381\]](#)

8.3.3.6 Выбор пользовательского списка значений для подсказки

Мастер ввода файла данных позволяет импортировать пользовательский список значений в таблицу надписей редактора *@Prompt* для вставки в функцию *@Prompt*. Поддерживаются только текстовые файлы и файлы Excel.

1. В редакторе *@Prompt* выберите *Ввод файла* на панели списка значений.
Отобразится *мастер ввода файла данных*.
2. Нажмите *Обзор* и перейдите к файлу ввода, который необходимо использовать.
3. Нажмите кнопку *Открыть*.
4. Нажмите *Далее*.

5. Выберите [Первая строка определяет заголовок столбца](#), если первая строка списка значений используется в качестве значений заголовков столбцов списка.
6. Выберите тип [кодировки файла](#)
7. Установите символ [Разделитель](#), который используется для разделения данных в файле ввода.
8. Нажмите [Далее](#).
9. Нажмите [Получить значения уникальных столбцов](#), чтобы убедиться, что предложены только уникальные значения.
10. Нажмите [Получить количество записей TOP](#), чтобы определить количество значений, которое будет предложено пользователю.
11. Используйте панель [Карта столбцов](#), чтобы выбрать столбцы, используемые для надписи и для значения первичного ключа.
12. Нажмите [Сортировка содержимого столбца](#), чтобы отсортировать содержимое выбранного столбца, выберите [По возрастанию](#), чтобы отсортировать содержимое столбца в порядке возрастания. Если [По возрастанию](#) не выбрано, содержимое столбца сортируется в порядке убывания.
13. Щелкните [Готово](#).
Отобразится сообщение для подтверждения [Добавление строк столбца](#). Нажмите [Да](#), чтобы подтвердить выбор.
14. Пользовательский список значений вставлен в таблицу списка значений в редакторе [@Prompt](#), а значения вставлены в определение @Prompt.
15. Нажмите [OK](#), чтобы вставить @Prompt в утверждение [SELECT](#) или [WHERE](#).

Связанные сведения

[Синтаксис функции @Prompt \[страница 388\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

[Определение статического списка значений для подсказки \[страница 379\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Редактирование существующего выражения @Prompt \[страница 381\]](#)

8.3.3.7 Редактирование существующего выражения @Prompt

Юниверс открыт в средстве создания юниверсов. Выбранное выражение объекта содержит функцию @Prompt.

Требуется изменить функцию @Prompt, которая уже существует в определении или в утверждении Where объекта или условия. Изменение выполняется через редактор [@Prompt](#).

1. Щелкните правой кнопкой мыши функцию @Prompt и выберите [Редактировать подсказку](#) в меню быстрого вызова.

2. Измените выражение подсказки с помощью редактора [@Prompt](#).
3. Нажмите [Разбор](#), чтобы проверить синтаксис [@Prompt](#).

Связанные сведения

[Синтаксис функции @Prompt \[страница 388\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

[Определение статического списка значений для подсказки \[страница 379\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

[Выбор пользовательского списка значений для подсказки \[страница 380\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

8.3.3.8 Определение функции @Prompt вручную для SQL-инструкции

❗ Примечание

Если функция [@Prompt](#) не была определена заранее, рекомендуется использовать редактор [@Prompt](#). Функция [@Prompt](#) имеет достаточно сложный синтаксис, поэтому рекомендуется ознакомиться с определением синтаксиса на соответствующей странице.

Чтобы определить функцию [@Prompt](#) для объекта вручную, выполните следующие действия:

1. Откройте окно [Изменить свойства](#), дважды щелкните класс, объект или условие на панели юниверса или дважды щелкните объект и выберите [Свойства объекта](#) или [Правка > Свойства](#).
2. Щелкните в диалоговом окне инструкции SELECT или WHERE панели [Определения](#) и определите значения [@Prompt](#) в соответствии с необходимым синтаксисом (см. ссылку ниже).
3. Для проверки синтаксиса SQL нажмите [Разбор](#).

Связанные сведения

[Синтаксис функции @Prompt \[страница 388\]](#)

[Редактор @Prompt \[страница 375\]](#)

[Редактирование существующего выражения @Prompt \[страница 381\]](#)

[Определение статического списка значений для подсказки \[страница 379\]](#)

[Выбор объекта юниверса в качестве списка значений для подсказки \[страница 380\]](#)

8.3.3.9 Свойства выражения @Prompt для определения подсказки вручную

Функция @Prompt имеет следующий синтаксис:

```
@Prompt('message','type','lov',Mono|Multi,free|constrained|primary_key,persistent|not_persistent,{ 'default value' })
```

Ниже приведен пример:

```
@Prompt('Displayed text ','A','Store\City',Mono,constrained,Persistent,{ 'Paris' })
```

Свойства значений выражения @Prompt описаны в приведенной ниже таблице.

Свойство	Описание
' сообщение '	Обязателен Текст сообщения подсказки. Текст должен быть заключен в одиночные кавычки. Например, 'Выберите регион', 'Выберите период времени', или 'Выберите выставочный зал'. Текст отображается в окне подсказки при запуске запроса пользователем.
' тип '	Обязателен, но может быть пустым (параметр 'A' принимается по умолчанию). Тип данных третьего параметра. Может быть одним из нижеперечисленных: <ul style="list-style-type: none">• 'A' для буквенно-цифровых• 'N' для цифровых• 'D' для даты Указанный тип данных должен быть заключен в одинарные кавычки. При использовании жестко запрограммированного списка пар значений синтаксис выглядит следующим образом 'value_type:key_type', например 'A:N', где первое значение является надписью, которую видит конечный пользователь, а второе значение является значением первичного ключа, которое используется запросом для увеличения скорости запроса. Каждый тип (надпись и первичный ключ) может быть A, N или D, как указано выше. Например, 'A:A' или 'A:N'. В этом случае следующий параметр "список_значений" содержит список пар параметров. Аналогичным образом параметр 'значение_по_умолчанию' будет содержать пары значений. Индекс поддержки должен быть установлен при использовании первичного ключа.

Свойство	Описание
список_значений	<p>Обязателен, но может быть пустым. Если список пуст, то запятая обязательна. Если данный параметр является объектом юниверса, пятый параметр (режим выбора= свободно ограничено первичный ключ) должен быть <code>primary_key</code>, поддержка индекса должна быть установлена в юниверсе.</p> <p>Можно указать два типа списка значений:</p> <ul style="list-style-type: none"> Указатель на Список значений из существующего объекта юниверса. Целевой список значений активизируется, если дважды щелкнуть объект, содержащий список значений, который требуется использовать на панели <i>Классы и объекты</i>. Отобразится имя класса и имя объекта, разделенные обратным слешем. Они должны быть заключены в одиночные кавычки. Например: 'Клиент\Страна'. Когда используется поддержка индекса и необходимо вернуть объекту значения ключа, для параметра <code>первичный_ключ</code> следует установить пятое значение. Жестко заданный список одинарных значений или пар значений. Значения в паре разделяются двоеточиями. Каждое значение заключается в одинарные кавычки. Пары значений разделяются запятыми. Весь список заключается в фигурные скобки. Установите ограничение на параметр <code>первичный_ключ</code>. <p>Синтаксическая конструкция для одного значения по умолчанию:</p> <pre>{ 'значение' }</pre> <p>Синтаксическая конструкция для нескольких одинарных значений по умолчанию:</p> <pre>{ 'значение_1', 'значение_2', . . . , 'значение_n' }</pre> <p>Можно определять пары значений по умолчанию.</p> <p>Синтаксическая конструкция для одной пары значений по умолчанию:</p> <pre>{ 'значение' : 'ключ' }.</pre> <p>Двоеточие (:) является разделителем между значением и ключом.</p> <p>Синтаксическая конструкция для нескольких пар значений по умолчанию:</p> <pre>{ 'значение_1' : 'ключ_1', 'значение_2' : 'ключ_2', . . . , 'значение_n' : 'ключ_n' }</pre> <p>Например: { 'Австралия' : 'А', 'Франция' : 'Ф', 'Германия' : 'Г', 'Япония' : 'Я', 'Испания' : 'И', 'Великобритания' : 'ВБ' }.</p>
Моно Мульти	<p>Обязателен, но может быть пустым (параметр Моно принимается по умолчанию). Использование запятой обязательно.</p> <p>Используйте параметр Моно, если пользователь может выбрать только одно значение из списка значений.</p> <p>Используйте параметр Мульти, если пользователь может выбрать несколько значений из списка значений.</p>

Свойство	Описание
свободно с ограничениями первичный_ключ	<p>Обязателен, но может быть пустым (параметр свободно принимается по умолчанию). Использование запятой обязательно.</p> <p>Используйте параметр свободно, если пользователь может указывать значение или выбирать его из списка значений.</p> <p>Используйте параметр с ограничениями, если пользователь должен выбирать значение из списка.</p> <p>Параметр первичный_ключ применяется в тех случаях, когда в юниверсе установлена поддержка индекса. Вместо введенного или отображаемого значения используется связанное значение ключа для объекта. Когда используется параметр список_значений, параметр первичный_ключ обязателен.</p>
постоянно непостоянно	<p>Необязательно. Когда этот параметр установлен, ввод аргумента следует завершать запятой. Если он не установлен, но есть седьмой установленный параметр (значения_по_умолчанию), необходимо также ставить запятую.</p> <div> <p>Примечание</p> <p>Обратите внимание на то, что этот параметр в приложении Desktop Intelligence не действует.</p> </div> <p>Параметр постоянно следует использовать в тех случаях, когда при обновлении документа по умолчанию отображаются последние значения, использованные в подсказке, даже если определены значения по умолчанию.</p> <p>Применяется параметр непостоянно, если по умолчанию при обновлении документа в подсказке не отображаются никакие ранее использованные значения.</p>

Свойство	Описание
'значение_по_умолчанию'	<p>Необязательно. Параметр значение по умолчанию используется, чтобы определить значения по умолчанию, представленные пользователю. Когда применяется жестко заданный список, введенные здесь значения по умолчанию должны присутствовать в списке [список_значений].</p> <p>Например, для одинарного значения:</p> <pre>{'Франция'}</pre> <p>Для пары значений:</p> <pre>{'Франция':'Ф'}</pre> <p>Для двух пар значений:</p> <pre>{'Франция':'Ф','Германия':'Г'}</pre> <p>При обновлении документа по умолчанию будут отображаться эти значения, но, если установлен параметр ПОСТОЯННО, тогда вместо значений по умолчанию используются последние значения, использованные в подсказке.</p> <p>Это могут быть одинарные значения или пары значений.</p> <p>Если в определении подсказки для данных определен параметр первичный_ключ, тогда необходимо указать значения ключа.</p>

8.3.3.10 Примеры: использование функции @Prompt

Далее приведены примеры синтаксиса @Prompt.

Минимальное использование функции @Prompt:

```
@Prompt('Displayed text ','A',,,)
```

Использование функции @Prompt со списком значений без значений по умолчанию:

```
@Prompt('Displayed text ','A',{'Paris','London','Madrid'},,)
```

Использование функции @Prompt со списком значений и одним значением по умолчанию:

```
@Prompt('Displayed text ','A',{'Paris','London','Madrid'},,,'Paris')
```

Использование функции @Prompt с объектом и значением по умолчанию:

```
@Prompt('Displayed text ','A','Store\City',,,'Paris')
```

Использование функции @Prompt со всеми возможными параметрами:

```
@Prompt('Displayed text ','A','Store\City',Mono,Constrained,Persistent,{'Paris'})
```

Использование функции @Prompt со списком пар значений без значений по умолчанию:

```
@Prompt('Displayed text ','A:N',{ 'Paris':'12','London':'7','Madrid':'15'},,,)
```

Использование функции @Prompt со списком пар значений и одной парой значений по умолчанию:

```
@Prompt('Displayed text ','A:N',{ 'Paris':'12','London':'7','Madrid':'15'},,,,
{ 'Paris':'12'})
```

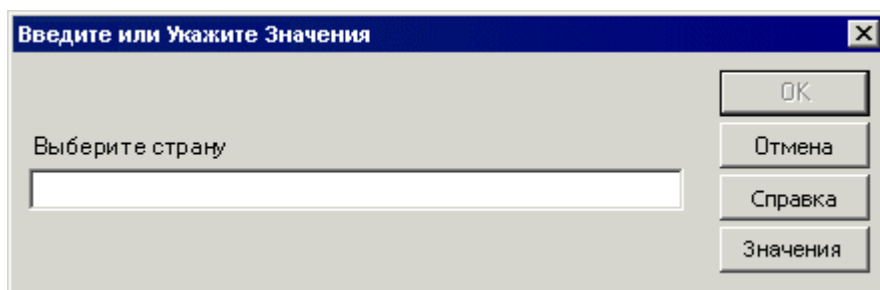
Пример: Использование функции @Prompt для ограничения стран без значений по умолчанию

Объект "Страна" возвращает значения для курортных стран. При необходимости ограничить возвращаемые значения в курорт только для одной страны, требуется отдельные объекты для каждой курортной страны в юниверсе. Однако при использовании @Prompt необходим только один объект:

```
Country.country=@prompt('Choose a country',
'A','Customer\Country of origin',Mono,Constant)
```

Пользователь получает подсказку на ввод названия страны, при этом будут возвращены те значения, которые являются курортами этой отдельной страны.

Если запрос выполняется в приложении Web Intelligence, отобразится окно со следующей подсказкой:



Пример: Синтаксис @Prompt со значениями по умолчанию

```
@prompt('Enter value(s) for Customer with IA:',
'A','Customer\Customer with I A',
Multi,primary_key,,{ 'Baker','Brendt'})
```

Пример: Синтаксис @Prompt с жестко заданным списком значений

Следующий пример демонстрирует, как предоставить список стран со значением по умолчанию. Когда объект поддерживает индекс и установлено ограничение primary_key, то значения по умолчанию могут

быть набором пар (значение, ключ), например {'Англия: 21', 'Шотландия:39'}. Пользователь должен выбрать только один регион, поэтому установлен параметр "Моно". Значение по умолчанию должно появиться в списке значений.

```
SELECT dbo.region.sales_region
FROM dbo.region
WHERE dbo.region.region_id = @Prompt('Choose a region','A:N',
{'England':'21', 'Scotland':'39', 'Wales':'14'},
Mono, primary_key, Persistent, {'Scotland':'39'})
```

Данная функция также позволит выполнять действия утверждения CASE WHEN ELSE в базах данных, которые его не поддерживают, в особенности в базах данных OLAP.

Пример: Синтаксис функции @Prompt для создания предопределенного условия с помощью подсказки с образцом для сравнения

В приведенном ниже примере пользователь выбирает имя клиента, вводя первую букву его имени. Если пользователь Web Intelligence введет H%, то в отчете будут возвращены все клиенты, фамилия которых начинается на H.

```
(@Select(Client\Client Name)
LIKE (@Prompt('enter','A',,,)+%))
```

Чтобы клиент мог использовать или верхний, или нижний регистр, применяется следующая синтаксическая конструкция:

```
(@Select(Client\Client Name)
LIKE lower(@Prompt('enter','A',,,)+%) OR
(@Select(Client\Client Name)
LIKE upper(@Prompt('enter','A',,,)+%))
```

8.3.3.11 Синтаксис функции @Prompt

Синтаксис функции @Prompt сложен из-за разносторонности данной функции. Необходимо написать сообщение подсказки и указать тип данных, являются ли данные единственным значением или несколькими значениями, являются ли данные постоянными, а также можно указать значения по умолчанию. Синтаксис указан ниже.

```
@Prompt('message','type',[lov],Mono|Multi,
free|constrained|primary_key,persistent|not_persistent,
[default_values])
```


Связанные сведения

[@Prompt \[страница 374\]](#)

[Определение функции @Prompt вручную для SQL-инструкции \[страница 382\]](#)

[Свойства выражения @Prompt для определения подсказки вручную \[страница 383\]](#)

8.3.4 @Select

Функция `@Select` применяется для повторного использования инструкции `SELECT` другого объекта. При использовании функции `@Select` в инструкции `SELECT` объекта путь к другому объекту юниверса указывается в качестве параметра функции `@Select` в следующей форме: `Class_Name\Object_Name`. В дальнейшем он действует как указатель на инструкцию `SELECT` для объекта, на который сделана ссылка.

Использование функции `@Select` дает следующие преимущества.

- Необходимо поддерживать только один экземпляр кода SQL.
- Обеспечивается согласованность кода.

📌 Примечание

При использовании функций `@Select` и `@Where` один объект в юниверсе зависит от другого. Создается новая зависимость объектов. При удалении одного объекта другой объект, использующий функцию `@Select` или `@Where` необходимо обновлять вручную.

8.3.4.1 Синтаксис для функции @Select

Функция `@Select` обладает следующим синтаксисом:

```
@Select (Classname\Objectname)
```

Синтаксис	Описание
Имя класса	Имя класса, который содержит соответствующие объекты.
Имя объекта	Имя соответствующего объекта.

8.3.4.2 Пример использования функции @Select

Пример: Использование @Select для повторного использования инструкции Select для объекта Service_line

Создается объект, называемый Линия службы повышения, который используется для возвращения линий служб, использованных в промо-акциях для различных курортов базы данных клуба. Этот объект принадлежит новому классу, который называется Повышения. Можно использовать @Select, чтобы сослаться на существующую инструкцию SELECT для объекта Service_lines.

Инструкция SELECT для линии службы повышения уровня отображается ниже:

Имя: promotional service line Тип: Символ

Описание:

Select: @Select(Resort\Service Line)

Where:

Таблицы... Тест

8.3.5 @Variable

Функция @Variable используется, например, в операторе WHERE для вызова значения, присвоенного одному из следующих типов переменных:

- Системные переменные BusinessObjects
- Переменные отчетов
- Переменные языка (языкового стандарта)
- Переменные операционной системы

В большинстве случаев @Variable вставляется со стороны операнда в условии оператора WHERE для объекта со страницы [Определение](#) листа [Изменить свойства](#). Запрос извлекает значение переменной.

❗ Примечание

Функция `@Variable` возвращает одно значение и не может использоваться с операторами `IN` и `INLIST`.

❗ Примечание

Если одна функция `@Variable` выполняется в запросе несколько раз, подсказка отображается только один раз.

❗ Примечание

Функция `@Variable` эквивалентна функции `@Prompt` с одним значением со следующими настройками:

```
@Prompt('Question','A',,mono,free)
```

Функцию `@Variable` можно объединить с функцией `@Prompt` в одном запросе, когда функция `@Prompt` возвращает одно значение.

Связанные сведения

[Синтаксис для функции @Variable \[страница 391\]](#)

[Описания свойств переменных \[страница 392\]](#)

8.3.5.1 Синтаксис для функции @Variable

Функция `@Variable` обладает следующим синтаксисом:

```
@Variable('<Variablename>')
```

❗ Примечание

Имя переменной должно заключаться в одинарные кавычки.

Пример: Синтаксис @Variable, возвращающий значение BOUSER

```
@Variable('BOUSER')
```

Связанные сведения

[@Variable \[страница 390\]](#)

[Описания свойств переменных \[страница 392\]](#)

8.3.5.2 Описания свойств переменных

Во всех случаях имя переменной необходимо заключить переменную в одинарные кавычки.

Имя переменной	Описание
Системные переменные BusinessObjects <ul style="list-style-type: none">BOUSER – регистрационное имя пользователяDBUSER – имя пользователя базы данныхDBPASS – пароль пользователя базы данных	Значения для переменных системы BusinessObjects. Выходные данные затем ограничиваются на основе регистрационного имени пользователя BusinessObjects. Значения для заявленного пользователя базы данных BusinessObjects.
Переменные отчетов <ul style="list-style-type: none">DOCNAME – имя документаDPNAME – имя поставщика данныхDPTYPE – тип поставщика данныхUNVNAME – имя юниверсаUNVID – идентификатор использованного юниверса	На эти переменные можно ссылаться, например, в параметре Begin_SQL, который будет выполняться перед инструкцией SELECT. Это может применяться в целях аудита использования базы данных (например, для определения наиболее часто используемого запроса отчета или юниверса).
Переменные языка <ul style="list-style-type: none">PREFERRED_VIEWING_LOCALEDOMINANT_PREFERRED_VIEWING_LOCALE	Переменные языка <ul style="list-style-type: none">PREFERRED_VIEWING_LOCALE – предпочитаемый пользователем языковой стандарт для просмотра. Это такой же языковой стандарт, выбранный пользователем, чтобы отобразить метаданные юниверса.DOMINANT_PREFERRED_VIEWING_LOCALE – основной языковой стандарт предпочитаемого пользователем языкового стандарта для просмотра. Это препятствует переводу пользователем данных во всех языковых стандартов (fr_FR, fr_BE, fr_CA, ...). Если переводы доступны в fr_FR, а языковым стандартом пользователя является fr_BE или fr_CA, так как они пользуются одним основным языковым стандартом, они могут повторно использовать переводы в fr_FR.

Имя переменной	Описание
Переменные операционной системы	Переменные среды Windows используются для получения информации об установке.

Связанные сведения

[@Variable \[страница 390\]](#)

[Синтаксис для функции @Variable \[страница 391\]](#)

8.3.5.3 Использование системной переменной BusinessObjects

Можно использовать функцию `@Variable` с системными переменными BusinessObjects™, чтобы ограничить данные в соответствии с идентификацией авторизованного в текущий момент пользователя BusinessObjects™.

❗ Примечание

Параметры входа в BusinessObjects™ должны совпадать с параметрами входа в базу данных.

Имя, назначенное для каждого пользователя BusinessObjects™ содержится в качестве следующей системной переменной BusinessObjects™:

- BOUSER – имя пользователя

Данная переменная отображается в поле идентификации пользователя, когда пользователь выполняет авторизацию для одного из продуктов Business Objects.

Используйте функцию `@Variable` в утверждении WHERE для объекта, чтобы ограничить доступ к данным для пользователя и его профиля базы данных, когда объект используется в запросе.

Вставка `@Variable` производится на стороне операнда условия в утверждении WHERE для объекта со страницы [Определение](#) на листе [Изменить свойства](#).

Пример

Использование `@Variable` для ограничения доступа служащих к служебным данным.

В юниверсе для базы данных человеческих ресурсов существует объект "Имя сотрудника". Желательно ограничить возвращаемые данные для имени сотрудника значениями, разрешенными в базе данных для каждого пользователя. Это позволит контролировать уровень доступа каждого пользователя к информации. Эта информация определяется их профилем базы данных.

Вставка функции `@Variable` в утверждение WHERE осуществляется следующим образом:

Служащие.Имя_служащего = @Переменная('BOUSER')

Когда объект "Имя служащего" используется в запросе, данные выдаются только для значения в таблицах, со значениями BOUSER которых они совпадают.

Связанные сведения

[@Variable \[страница 390\]](#)

[Синтаксис для функции @Variable \[страница 391\]](#)

[Описания свойств переменных \[страница 392\]](#)

8.3.5.4 Использование переменных языкового стандарта

Переменные языкового стандарта функции @Variable необходимо использовать для того, чтобы определить параметры языкового стандарта, чтобы приложение Web Intelligence получало отчеты и отображало сведения в соответствующем языковом стандарте. В таблицах базы данных должен содержаться столбец, в котором перечислены языки для строк с переводом данных. Языковой стандарт описывает язык и географическую область, способ сортировки данных, форматирование дат и прочие специальные форматы. В утверждении WHERE для объекта используйте функцию @Variable. Пользователю необходимо выбрать языковой стандарт, если данный объект используется в запросе. При запуске запроса отображается окно подсказки, запрашивающее пользователя ввести языковой стандарт. Руководство по средству управления переводами содержит список кодов языковых стандартов и основных языковых стандартов.

Ниже представлены параметры, которые можно определить.

- @Variable('PREFERRED_VIEWING_LOCALE')
- @Variable('DOMINANT_PREFERRED_VIEWING_LOCALE')

Пример

Таблицы "ПРОДУКТ", указанная ниже, была переведена на много языков. Пользователь хочет вставить имена продуктов в определенный языковой стандарт.

Таблица "ПРОДУКТ"

Идентификатор продукта	ЯЗЫКОВОЙ СТАНДАРТ	Product_Name
DC1212	en_GB	Digital camera
DC1212	fr_FR	Appareil photo numérique
DC1212	de_DE	Digitalkamera
DC1212	es_ES	Cámaras digitales

Идентификатор продукта	ЯЗЫКОВОЙ СТАНДАРТ	Product_Name
...

```
SELECT Product_Name
FROM PRODUCT
WHERE PRODUCT.LOCALE = @Variable('PREFERRED_VIEWING_LOCALE')
```

Во время запроса пользователь замещает переменную правильной локальной переменной и приложение Web Intelligence получает сведения в языковой стандарт.

Связанные сведения

[@Variable \[страница 390\]](#)

[Синтаксис для функции @Variable \[страница 391\]](#)

[Описания свойств переменных \[страница 392\]](#)

8.3.5.5 Использование переменных отчета

Функция `@Variable` применяется в утверждении `WHERE` объекта для включения переменных отчета в запрос.

На эти переменные можно ссылаться в параметре `Begin_SQL`, который будет выполняться перед инструкцией `SELECT`. Это может применяться в целях аудита использования базы данных (например, для определения наиболее часто используемого запроса отчета или юниверса).

На переменные можно ссылаться в следующих объектах:

- Определение объекта: утверждения `SELECT`, `WHERE` и т.д.
- Фильтры
- Выражение `Join`
- Параметр `Begin_SQL`

Связанные сведения

[@Variable \[страница 390\]](#)

[Синтаксис для функции @Variable \[страница 391\]](#)

[Описания свойств переменных \[страница 392\]](#)

8.3.5.6 Использование переменных операционной системы

Для получения информации об установке можно использовать переменные среды Windows. Например, `NUMBER_OF_PROCESSORS`, `USERNAME`.

Пример

Если в запрос включить выражение `@Variable (NUMBER_OF_PROCESSORS)`, в ответ будет включено количество используемых в компьютере процессоров.

Связанные сведения

[@Variable \[страница 390\]](#)

[Синтаксис для функции @Variable \[страница 391\]](#)

[Описания свойств переменных \[страница 392\]](#)

8.3.5.7 Применение пользовательских переменных в приложении Desktop Intelligence

При работе с приложением Desktop Intelligence можно использовать функцию `@Variable` в утверждении `WHERE` объекта, чтобы ссылаться на переменную в связанном текстовом файле. Это позволяет указать зависящие от пользователя условия для объекта.

Для использования этой переменной, BusinessObjects требуется запустить из командной строки с параметром `-vars`. Будет необходимо изменить командную строку в ярлыках Windows на всех ПК, использующих данную функцию.

❗ Примечание

Обеспечение запуска BusinessObjects с помощью командной строки, делает использование функции `@Variable` сложным для поддержки развертывания юниверса более чем несколькими пользователями. При большом количестве пользователей или широкой географии их расположения, не следует использовать функции `@Function` со связанными текстовыми файлами для применения ограничений.

Использование функции `@Variable` с переменными текстовых файлов имеет следующие преимущества:

Основным преимуществом в использовании функций `@Variable` с переменными текстовых файлов является возможность обновлять значения переменных в текстовом файле, при этом не внося никаких изменений в юниверс.

Использование функции `@Variable` с переменными текстовых файлов имеет следующие недостатки:

- Необходимо изменять параметры командной строки запуска на каждом клиентском компьютере, вписывая в нее аргумент – `vars <textfile.txt>`.
- Могут возникнуть затруднения с обеспечением безопасности, так как текстовый файл на ПК может быть изменен локально.

Принимая во внимание количество возможных проблем при использовании функции `@Variable` с текстовыми переменными, если в среде предприятия используются продукты Business Objects, для контроля за доступом к данным следует использовать параметры безопасности, предлагаемые модулем Supervisor.

Связанные сведения

[Применение пользовательских переменных в приложении Desktop Intelligence \[страница 397\]](#)

[@Variable \[страница 390\]](#)

[Синтаксис для функции @Variable \[страница 391\]](#)

[Описания свойств переменных \[страница 392\]](#)

8.3.5.7.1 Применение пользовательских переменных в приложении Desktop Intelligence

Чтобы использовать одно или несколько предварительно определенных значений переменных, выполните следующие действия.

1. Создать текстовый файл, который содержит список переменных с соответствующими значениями. Используйте следующий формат: имя переменной = значение
2. В командную строку, используемую для запуска системы BusinessObjects, добавьте следующее: `Busobj.exe -vars <textfile.txt>` Например, если имеется текстовый файл `Bovars.txt`, можно ввести следующее: `C:\BusinessObjects\Busobj.exe -vars Bovars.txt` Синтаксическая конструкция `-vars` – это переключатель, который указывает операционной системе, что текстовый файл следует загрузить в память для его использования системой BusinessObjects.
3. Откройте лист [Изменить свойства](#) объекта, для которого необходимо указать текстовую переменную.
4. Вставьте `@Variable` со стороны операнда условия в утверждении WHERE. Например: `COUNTRY.COUNTRY_NAME = @Variable('Страна')`. 'Страна' – это имя переменной в текстовом файле. Термин должен быть заключен в одинарные кавычки.
5. Нажмите [OK](#) и сохраните юниверс.

8.3.6 Оператор Where

Функция `@where` служит для повторного использования утверждения WHERE другого объекта. Когда функция `@where` используется в утверждении WHERE объекта, путь к другому объекту юниверса

указывается в качестве параметра функции @where в следующей форме: Class_Name\Object_Name. В дальнейшем он действует как указатель на утверждение WHERE объекта, на который сделана ссылка.

С помощью утверждения WHERE создается динамическая связь между двумя объектами. При изменении утверждения WHERE исходного объекта утверждение WHERE ссылающегося объекта автоматически обновляется.

Использование функции @where позволяет использовать существующий код. Это дает следующие преимущества:

- Необходимо поддерживать только один экземпляр кода SQL.
- Обеспечивается согласованность кода.

При использовании @Select и @where функций один объект юниверса теперь зависит от другого. Создается новая зависимость объектов. При удалении одного объекта другой объект необходимо обновить вручную с помощью функции @Select или @where.

8.3.6.1 Синтаксис для функции Where

Синтаксис данной функции следующий:

```
@Where ( Classname\Objectname )
```

Синтаксис	Описание
Classname	Название класса.
Objectname	Имя соответствующего объекта.

8.3.6.2 Пример: использование функции @Where для повторного использования утверждения WHERE

Пример: Использование @Where для повторного использования утверждения WHERE курорта

Можно создать объект, называемый Линия обслуживания курортов, который используется для вывода линии обслуживания, доступной на каждом курорте. Необходимо использовать функцию @Prompt , определенную в объекте курорта таким образом, чтобы пользователи получали подсказку указать название курорта при запросе доступных услуг конкретного курорта.

SQL объекта курорта (объект, который необходимо упомянуть) отображается следующим образом:

Описание:
Name of the resort

Select:
Resort.resort

Where:
Resort.resort = @Prompt('Enter a resort name','A','Resort\Resort','mono',)

Таблицы... Тест

Новый объект "Линия обслуживания курортов" использует функцию @Prompt в утверждении WHERE для курорта следующим образом:

Select:
Service_Line.service_line & ' at ' & Resort.resort

Where:
@Where(Resort\Resort)

Таблицы... Тест

При запуске запроса с линией обслуживания курортов отображается предложение указать имя курорта. При изменении утверждения WHERE для курорта, изменение автоматически производится в объекте линии обслуживания курортов.

8.4 Использование внешних стратегий для настройки процесса создания юниверса

В средстве создания юниверсов используются встроенные подпрограммы автоматического создания компонентов юниверса в соответствие со структурой базы данных. Такие подпрограммы, называемые стратегиями, доступны на странице "Стратегии" диалогового окна "Параметры" (Файлы > Параметры > Стратегии). Эти стратегии поставляются в качестве встроенных в средство создания юниверсов. Они доступны к использованию и могут быть изменяемыми. Способы активации и использования стратегий описаны в разделе [Выбор стратегий \[страница 88\]](#).

Можно также создавать сценарии SQL, позволяющие определенной структуре вывода данных выполнять задачи автоматического создания юниверса. Их можно найти на странице "Стратегии" среди других стратегий. Такие сценарии, которые задаются и настраиваются пользователем, называются внешними стратегиями.

В данном разделе описываются внешние стратегии и способ их использования.

8.4.1 Перенос внешних стратегий в средство создания юниверсов

Внешние стратегии в версиях средства создания юниверсов, предшествующих Universe Designer 6.5, определялись как внешние текстовые файлы с именем st<xxxx>.txt. В текущей версии такие файлы не поддерживаются.

❗ Примечание

При переносе из Universe Designer 6.5 внешние стратегии обрабатываются в средстве создания юниверсов аналогичным образом.

Чтобы убедиться, что настроенные и определенные пользователем внешние стратегии предыдущих версий доступны в средстве создания юниверсов, необходимо выполнить следующие действия:

- Отредактируйте новый файл внешней стратегии (<RDBMS>.STG) следующим образом:
 - Откройте файл внешней стратегии для целевой реляционной СУБД в редакторе XML.
 - Создайте отдельную запись для каждой стратегии.
 - Для каждой стратегии скопируйте скрипт SQL непосредственно в STG-файл с помощью тега SQL.

Или

- Введите путь к файлу, чтобы с помощью тега FILE дать ссылку к данным через внешний текстовый файл.

Оба способа подробно описаны в разделе [Создание внешней стратегии \[страница 411\]](#).

- Скопируйте справочный текст во второй файл XML (<RDBMS><language>.STG). Это описано в разделе [Создание справочного текста для внешней стратегии \[страница 403\]](#).
- Убедитесь, что файл внешней стратегии объявлен в файле основных параметров (SBO), а не в файле параметров (PRM), как это было в предыдущих версиях приложения Universe Designer. Это описывается в разделе [Проверка на наличие описания файла внешней стратегии \[страница 404\]](#).

8.4.2 Общие сведения о внешних стратегиях

Ниже приводится таблица, представляющая используемые файлы и их роли в создании и управлении внешними стратегиями.

Роли и файлы, используемые для процесса управления внешними стратегиями

Описание	
<p>Внешние стратегии хранятся и создаются в виде файла внешних стратегий (<RDBMS>.STG).</p>	<p>XML-файл содержит имя, тип внешней стратегии, скрипт SQL или файловую ссылку на внешний текстовый файл, содержащий данные. Файл хранится здесь: \$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS>.stg. По одному файлу для каждой реляционной СУБД. Используйте файл strategy.dtd здесь: \$INSTALLDIR/dataAccess/RDBMS/connectionServer/strategy.dtd Смежные разделы:</p> <ul style="list-style-type: none">• Какую структуру имеет файл стратегии (STG)? [страница 405]• Создание внешней стратегии [страница 411]
<p>Справочный текст для внешних стратегий хранится и создается в виде файла языка внешней стратегии (<RDBMS><language>.STG)</p>	<p>Файл XML содержит справочный текст для каждой внешней стратегии в файле внешней стратегии. Это текст, появляющийся под внешней стратегией при ее выборе на странице "Стратегии". Файл хранится здесь: \$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS><language>.stg. Используйте файл strategy_localization.dtd, расположенный здесь: \$INSTALLDIR/dataAccess/RDBMS/connectionServer/strategy_localization.dtd. Смежный раздел: Создание справочного текста для внешней стратегии [страница 403].</p>
<p>Файл внешней стратегии описан в файле общего доступа к данным (SBO) для целевой реляционной СУБД.</p>	<p>Файл XML содержит общие параметры доступа к данным для целевой реляционной СУБД. Имя файла внешней стратегии рассматривается как значение для параметра "Внешние стратегии" по умолчанию. Смежный раздел: Проверка на наличие описания файла внешней стратегии [страница 404]</p>

8.4.3 Что такое внешняя стратегия?

Внешняя стратегия – это скрипт SQL, хранимый во внешнем файле .UNV, его структура позволяет использовать его в средстве создания юниверсов для автоматического создания объектов или объединений, а также для задач определения таблиц в юниверсе. Внешние стратегии хранятся в файле внешних стратегий с расширением STG. Файлы внешних стратегий имеют формат XML. Для каждой поддерживаемой реляционной СУБД существует свой файл внешней стратегии.

Файлы внешней стратегии хранятся в следующем каталоге:

\$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<rdbms>.stg

📘 Примечание

Для редактирования файла внешней стратегии следует использовать редактор XML.

8.4.3.1 Доступ к внешним стратегиям

Внешние стратегии появляются в раскрывающихся списках, в которых также указаны встроенные стратегии на странице "Стратегии". Каждый раскрывающийся список соответствует категориальному типу стратегии в файл XML. Имени внешней стратегии в списке предшествует словосочетание "Внешняя стратегия", например:

Внешняя стратегия:<имя стратегии>

Например, внешняя стратегия для создания объединения под названием "Ограничения" в файле стратегии отображается в раскрывающемся списке "Объединения" на странице "Стратегии" в следующем виде: Внешняя стратегия:Ограничения.

8.4.4 Создание справочного текста для внешних стратегий

На странице "Стратегии" под каждой выбранной стратегией появляется комментарий. Это справочный текст о стратегии. Доступ к справочному тексту встроенных стратегий невозможен, также как невозможно его редактирование. Однако доступ к справочному тексту внешних стратегий и его редактирование разрешены.

📘 Примечание

В предыдущих версиях средства создания юниверсов текст справки включался в текстовый файл стратегии в разделе [HELP]. Текст этого раздела теперь хранится в отдельном файле. Сведения о файле языка внешней стратегии приводятся ниже.

8.4.4.1 Справочный текст внешней стратегии хранится в отдельном файле

Текст справки по внешним стратегиям хранится в отдельном языковом файле внешней стратегии под названием <тип_реляционной_СУБД><язык>.stg. Например, oacleen.stg является файлом справочного текста для стратегий файла oracle.stg.

Записи справочного текста можно редактировать и настраивать. Чтобы облегчить работу тех дизайнеров, которые впервые используют стратегию, в справочном тексте обычно даются краткие сведения о функциях стратегии.

Следует убедиться, что запись справочного текста, соответствующая каждой внешней стратегии, появляющейся в виде файла внешней стратегии, отображается в файле языка внешней стратегии.

Для каждой установленной языковой версии средства создания юниверсов существует отдельный языковой файл стратегии. Файл языка внешней стратегии находится в одном каталоге с файлом внешней стратегии. Например, для французской версии средства создания юниверсов языковой файл внешней стратегии для Oracle будет носить имя oraclefr.stg. Для английской версии – oracleen.stg.

Когда создается новая внешняя стратегия в виде файла внешней стратегии, также должна создаваться запись для справочного текста в файле языка внешней стратегии. Эта запись передает сведения о внешней стратегии другим дизайнерам, использующим юниверс.

Пример: Запись справочного текста о стратегии поставляется с драйвером доступа к данным Oracle

Справочный текст стратегии "Классы и объекты", находящийся в списке файла oracleen.stg, приведен ниже. Это справочный текст внешней стратегии "Классы и объекты", определенный в файле oracle.stg.

```
<Strategy Name="Classes_and_Objects">
  <Message id="Help">This strategy reads the database structure. It
  associates tables with classes, and columns with objects.</Message>
  <Message id="Name">External Strategy: Classes and Objects</Message>
```

8.4.4.2 Создание справочного текста для внешней стратегии

Чтобы создать справочный текст для внешней стратегии, выполните следующие действия:

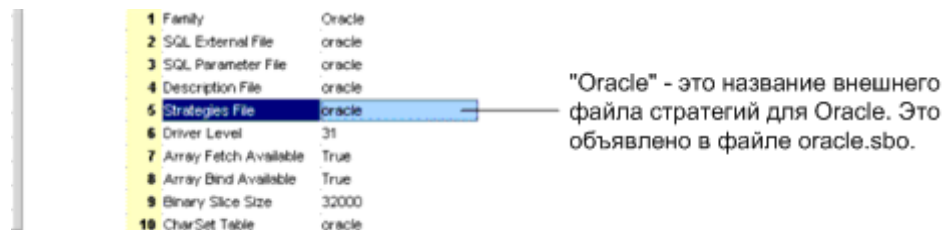
1. Откройте файл языка внешней стратегии для целевой реляционной СУБД в редакторе XML. Файл языка внешней стратегии для целевой реляционной СУБД расположен здесь:
\$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS><language>.stg.
Например,
\$INSTALLDIR/dataAccess/RDBMS/connectionServer/oracle/oracleen.stg.
2. Создайте новый элемент "Имя".
3. Введите имя стратегии. Это имя стратегии, для которой создается справочный текст.
4. Создайте идентификатор сообщения под названием "Справка". Данный тег содержит справочный текст.
5. Введите справочный текст.
6. Создайте идентификатор сообщения под названием "Имя". Данный тег содержит имя, которое будет отображаться в раскрывающемся списке стратегий для выбранной внешней стратегии.
7. Введите имя стратегии.
Подтвердите, сохраните и закройте файл.
При следующем запуске средства создания юниверсов справочный текст будет отображаться под выбранной внешней стратегией.

→ Совет

Простым способом создания элемента "Имя" и настройки его параметров является копирование существующего элемента "Имя" и последующее введение значений для новой стратегии.

8.4.5 Проверка на наличие описания файла внешней стратегии

Файл внешней стратегии описан в разделе "Параметр" общего файла параметра (SBO) для целевой реляционной СУБД. Например, файлом внешней стратегии для Oracle является oracle.stg. Он имеет значение oracle в файле oracle.sbo, как показано ниже:



1	Family	Oracle
2	SQL External File	oracle
3	SQL Parameter File	oracle
4	Description File	oracle
5	Strategies File	oracle
6	Driver Level	31
7	Array Fetch Available	True
8	Array Bind Available	True
9	Binary Slice Size	32000
10	CharSet Table	oracle

"Oracle" - это название внешнего файла стратегий для Oracle. Это объявлено в файле oracle.sbo.

8.4.5.1 Проверка на наличие описания файла стратегии в файле SBO

Чтобы убедиться в правильности описания файла внешней стратегии, выполните следующие действия:

1. Откройте файл SBO для целевой реляционной СУБД.
2. Убедитесь в соответствии установленного параметра "Имя стратегий" имени файла внешних стратегий. Настройка может быть выполнена по умолчанию.
3. Если имя задано неверно, введите правильное имя файла внешних стратегий.
4. Если изменения внесены, сохраните и закройте файл.
Или
5. Если изменения не были внесены, закройте файл без сохранения.

📌 Примечание

Внешние стратегии в предыдущих версиях средства создания универсов определялись с помощью PRM-файла. В версии Universe Designer 6.5 реализовано другое поведение. Параметр "Файл стратегий" в файле SBO настроен по умолчанию на имя файла внешних стратегий для целевой реляционной СУБД. Дополнительные сведения о переносе внешних стратегий в Universe Designer 6.5 см. в разделе [Создание справочного текста для внешних стратегий \[страница 402\]](#).

8.4.6 Использование примера внешних стратегий

Все файлы внешней стратегии содержат несколько существующих стратегий, поставляющихся с продуктами Business Objects. Например, любой файл может содержать одну стратегию объекта, одну стратегию объединения и одну стратегию обозревателя таблиц, или по несколько стратегий каждого типа.

Файл примера можно настраивать или использовать в качестве основы для создания новой внешней стратегии. Можно настроить существующую стратегию или создать собственную.

Прежде чем вносить изменения в файл, сделайте его копию.

8.4.7 Какую структуру имеет файл стратегии (STG)?

Для каждой поддерживаемой реляционной СУБД существует файл внешней стратегии (STG) в формате XML. В этот файл переносят существующие внешние стратегии. В этом файле создают новые внешние стратегии. Все файлы внешней стратегии используют файл стратегии dtd (<RDBMS>.dtd), расположенный в следующем каталоге:

```
$INSTALLDIR/dataAccess/RDBMS/connectionServer
```

Элементы файла внешней стратегии XML определены в файле внешней стратегии DTD. При создании нового элемента стратегии с помощью некоторых редакторов XML, например XML SPY, доступные параметры приводятся в раскрывающемся списке.

Файл внешней стратегии содержит основной раздел под названием "Стратегии". В этом разделе определены все внешние стратегии. Раздел "Стратегии" имеет следующие элементы и параметры:

Элемент файла	Описание
Стратегия	Основной элемент. Все внешние стратегии создаются в пределах данного элемента.
Имя	Имя внешней стратегии. Это имя отображается в раскрывающемся списке на странице "Стратегии". Элемент по умолчанию.
Тип	Список, в котором отображается внешняя стратегия на странице "Стратегии". Существует 3 значения: <ul style="list-style-type: none">ОБЪЕДИНЕНИЕ: стратегия объединения отображается в списке "Объединения".ОБЪЕКТ: стратегия классов и объектов отображается в списке "Классы и объекты".СТРУКТУРА: стратегия обнаружения таблицы отображается в списке "Таблицы".

Элемент файла	Описание
SQL	Код SQL для скрипта. Этот скрипт SQL выполняется средством создания юниверсов при выборе стратегии. Скрипт SQL должен соответствовать специальному формату вывода данных при создании объекта или объединения и обеспечивать правильную работу подпрограмм обнаружения таблиц. Дополнительные сведения об особенностях создания структуры SQL для внешней стратегии см. в разделе Формат вывода стратегий объекта (OBJECT) [страница 407] .
Соединение	Задайте соединение с базой данных. Тип соединения должен быть персональным.
SkipMeasures	При установлении значения Y программа пропускает экран в мастере быстрого дизайна, предназначенный для создания мер:
Файл	Путь к внешнему текстовому файлу, содержащий данные, упорядоченные в специальном формате вывода, автоматически создающего юниверс. Дополнительную информацию см. в разделе Создание текстового файла для данных [страница 412] .

Пример: Внешняя стратегия "Классы и объекты" в файле oracle.stg

Файлом внешней стратегии для Oracle является oracle.stg. Он хранится в каталоге \$INSTALLDIR/dataAccess/RDBMS/connectionServer/oracle/oracle.stg. Этот файл содержит несколько примеров внешних стратегий, входящих в комплект поставки средства создания юниверсов. Эти стратегии можно настраивать или использовать в качестве шаблонов при создании новых стратегий.

Внешняя стратегия из файла oracle.stg, автоматически связывающая таблицы с классам и столбцы с объектами, показана ниже:

```
<Strategy Name="Classes_and_Objects">
  <Type>OBJECT</Type>
  <SQL>SELECT
    U1.table_name,'|',
    U1.column_name,'|',
    translate(initcap(U1.table_name),'_',' '),'|',
    translate(initcap(U1.column_name),'_',' '),'|',
    U1.table_name||'|'||U1.column_name,'|',
    ' ','|',
    decode(SUBSTR(U1.DATA_TYPE,1,1),'N','N','F','N','D','D','C'),'|',
    SUBSTR(U2.comments,1,474),'|',
    'O','|'
  FROM USER_TAB_COLUMNS U1,USER_COL_COMMENTS U2
  WHERE
    U1.table_name=U2.table_name
  and U1.column_name=U2.column_name
  UNION
  SELECT
```

```

        S.SYNONYM_NAME,'|',
        U1.column_name,'|',
        translate(initcap(S.SYNONYM_NAME),'_',' '),'|',
        translate(initcap(U1.column_name),'_',' '),'|',
        S.SYNONYM_NAME||'.'||U1.column_name,'|',
        ' ','|',
        decode(SUBSTR(U1.DATA_TYPE,1,1), 'N','N','F','N','D','D','C'),'|',
        SUBSTR(U2.comments,1,474),'|',
        'O','|'
FROM ALL_TAB_COLUMNS U1, ALL_COL_COMMENTS U2, ALL_OBJECTS O, USER_SYNONYMS S
WHERE
    S.table_owner=O.owner
AND    S.table_name=O.object_name
AND    (O.OBJECT_TYPE='TABLE' OR O.OBJECT_TYPE='VIEW')
AND    O.owner=U1.owner
AND    O.object_name=U1.table_name
AND    U1.owner=U2.owner
AND    U1.table_name=U2.table_name
AND    U1.column_name=U2.column_name</SQL>
</Strategy>

```

8.4.8 Форматы вывода у стратегий

Написание и копирование скрипта SQL выполняется в рамках тега <SQL> в файле внешних стратегий. Порядок и тип информации, возвращаемой скриптом SQL, зависят от того, создается стратегия объекта, объединения или таблицы. В средстве создания юниверсов для различных типов стратегий используются данные различного рода.

При создании скрипта SQL для стратегии, необходимо убедиться, что созданный для скрипта вывод соответствует форматам вывода, описанным ниже.

Вывод для скрипта представлен в виде ряда столбцов. Каждый столбец соответствует информативному элементу, используемому для создания компонентов объекта, объединения или таблицы.

В данном разделе представлены форматы вывода для:

- Стратегий объекта
- Стратегий объединения
- Стратегий обозревателя таблиц.

8.4.8.1 Формат вывода стратегий объекта (OBJECT)

Формат вывода стратегии объекта содержит девять столбцов. Необходимо убедиться, что выход включает все эти столбцы, даже если они содержат нулевые значения. Все возвращенные значения должны быть разделены символом "вертикальная линия" ("|"). "Вертикальную линию" необходимо добавлять в конце возвращенных значений.

Номер столбца	Столбец содержит...	Описание
1	Таблица	Имя формата объекта – [Qualifier.] [Owner.] Таблица, содержащая имена длиной до 35 символов. Если данный столбец пуст, то таблицы получают из Select (пятый столбец) и Where (шестой столбец).
2	Имя столбца	Имя столбца
3	Имя класса	Имя класса. Подклассы записываются следующим образом: формат класса или подкласса.
4	Имя объекта	Имя объекта или условия. Если значение имени объекта является пустым, то создаются класс и описание класса.
5	Select	Инструкция Select.
6	Where:	Если столбец Select оставлен пустым, а утверждение Where включено, то создаются predeterminedное условие и его описание.
7	Тип	C (Символ), N (Число), D (Дата), T (Подробный текст). Если столбец оставить пустым, то его значением по умолчанию будет N.
8	Описание	Описание объекта.
9	Квалификация	D (Измерение), M (Мера) или I (Атрибут). Если столбец оставить пустым, то его значением по умолчанию будет D.

Пример: Стратегия внешнего объекта, копирующая комментарии столбца в описания объекта

Пример, приведенный ниже, не содержит утверждения Where. Столбец вывода для утверждения Where пуст.

```
<Strategies>
```

```
<Strategy Name="Read Column descriptions">
```

```
<Type>OBJECT</Type>
```

<SQL>Select

	Col	Описание
Table_name, ' ',	1	Имя таблицы
Column_name, ' ',	2	Имя столбца
Заменить (Table_name, ' _ '), ' ',	3	Заменить подчеркивания в имени таблицы на пробелы в имени класса
Заменить (Column_name, ' _ '), ' ',	4	Заменить подчеркивание в имени столбца на пробелы в имени объекта.
Table_name ' ' Column_name, ' ',	5	Соединить разведенные периодом имя таблицы и имя столбца. Это инструкция Select.
, ' ',	6	Нет утверждения Where
Column_type, ' ',	7	Получить тип столбца из системных таблиц
Column_Desc, ' ',	8	Получить описание столбца из системных таблиц
' ', ' ',	9	Нулевое значение типа объекта будет значением измерения по умолчанию.

</SQL>

8.4.8.2 Формат вывода стратегий объединения (ОБЪЕДИНЕНИЕ)

Формат вывода стратегии объединения содержит следующие столбцы:

Номер столбца	Столбец содержит...	Описание
1	Таблица 1	Имя первой таблицы в объединении
2	Таблица 2	Имя второй таблицы в объединении
3	Определение объединения	Действительное определение объединения в форме table1.column1=table2.column2

Номер столбца	Столбец содержит...	Описание
4	Outertype	Тип внешнего объединения. L=левое внешнее объединение, R=правое внешнее объединение. Если столбец оставлен пустым, то внешнее объединение отсутствует.
5	Мощность множества (дополнительно)	допустимые значения: 11, 1N, N1.

8.4.8.3 Формат вывода стратегий обозревателя таблиц (СТРУКТУРА)

Формат вывода стратегии обозревателя таблиц содержит следующие столбцы:

Номер столбца	Столбец содержит...	Описание
1	Классификатор	Зависимый элемент реляционной СУБД. Классификатором таблицы является имя базы данных или любой другой идентификатор.
2	Владелец	Зависимый элемент реляционной СУБД
3	Таблица	Имя таблицы, просмотр или синоним.
4	Столбец	Имя столбца.
5	Тип данных	C (Символ), N (Число), D (Дата), T (Подробный текст). Если столбец оставить пустым, то его значением по умолчанию будет C.
6	Свойство необязательной определенности (пустого значения) Y (Да) или N (Нет)	Указывает, могут ли столбцы иметь нулевые значения

8.4.9 Создание внешней стратегии

Внешнюю стратегию можно создать двумя способами:

Создание внешней стратегии с помощью...

Тега в файле XML

Описание

Непосредственная вставка скрипта SQL.

SQL

Скрипт SQL для стратегии вставляется непосредственно в файл внешней стратегии с помощью тега SQL.

Создания ссылки к данным во внешнем файле

ФАЙЛ

Вводятся путь к файлу и имя для внешнего текстового файла, содержащего данные для стратегии.

Оба способа описаны в следующей процедуре.

8.4.9.1 Создание внешней стратегии

Чтобы непосредственно создать внешнюю стратегию, выполните следующие действия:

1. Откройте файл внешней стратегии для целевой реляционной СУБД в редакторе XML. Файл стратегии для целевой реляционной СУБД расположен здесь:
\$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS>.stg.
2. Создайте новый элемент стратегии.
Это новая стратегия. При использовании некоторых редакторов XML, например XML SPY, элементы имени, типа и SQL для стратегии создаются автоматически.
3. Введите имя стратегии.
Имя стратегии отображается на вкладке "Стратегии" диалогового окна "Параметры универса" и в мастере быстрого дизайна.
4. Введите значение для параметра "ТИП": ОБЪЕКТ, ОБЪЕДИНЕНИЕ или СТРУКТУРА.
Например, ТИП=ОБЪЕКТ.
5. Введите SQL-инструкцию стратегии. Сведения о формате SQL см. в разделе [Форматы вывода у стратегий \[страница 407\]](#).
Или
Если необходимо создать ссылку к текстовому файлу, содержащему данные, замените элемент SQL элементом файла. Введите путь к файлу для файла данных, например C:\Path\Filename.txt
6. Добавьте дополнительные элементы и задайте значения при необходимости.
7. Проверьте достоверность файла XML, перед сохранением и закрытием файла.
8. Убедитесь, что файл внешней стратегии описан в файле общего доступа к данным для целевой реляционной СУБД (<RDBMS>.SBO). Для этого выполните следующие шаги:
 - Откройте файл общего доступа к данным (SBO), расположенный в каталоге:
\$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/
 - Убедитесь, что элемент "Файл стратегий" задан в соответствии с именем файла внешних стратегий. Это значение по умолчанию.
 - Если к файлу SBO применены изменения, то необходимо сохранить и закрыть файл.Внешняя стратегия появляется в раскрывающихся списках "Объединение", "Объекты" или "Таблицы" на странице "Стратегии" диалогового окна "Параметры". Чтобы заново созданные внешние стратегии отображались в списках, необходимо перезапустить средство создания универсов.

❗ Примечание

Если требуется добавить справочный текст, появляющийся под названием внешней стратегии, выбранной на странице "Стратегии", необходимо добавить этот текст в отдельный файл, внешний файл <RDBMS><language>.STG, расположенный в одном каталоге с файлом внешней стратегии. Сведения о добавлении справочного текста для внешней стратегии см. в разделе [Создание справочного текста для внешних стратегий \[страница 402\]](#).

8.4.10 Создание текстового файла для данных

Можно создать текстовый файл, содержащий данные для внешней стратегии. При создании внешней стратегии можно ввести путь к файлу и имя для текстового файла вместо прямой вставки SQL. При этом следует вставить элемент "ФАЙЛ" в файл внешней стратегии и задать значения для пути к файлу и имени.

Вывод скрипта SQL должен соответствовать правильному формату типа стратегии, объекта, объединения или таблицы. Сведения о форматах вывода см. в разделе [Форматы вывода у стратегий \[страница 407\]](#).

Все форматы представляют собой столбцы с данными, разделенные табуляцией.

8.4.11 Применение внешних стратегий в средстве создания юниверсов

Применять внешние стратегии следует следующим образом:

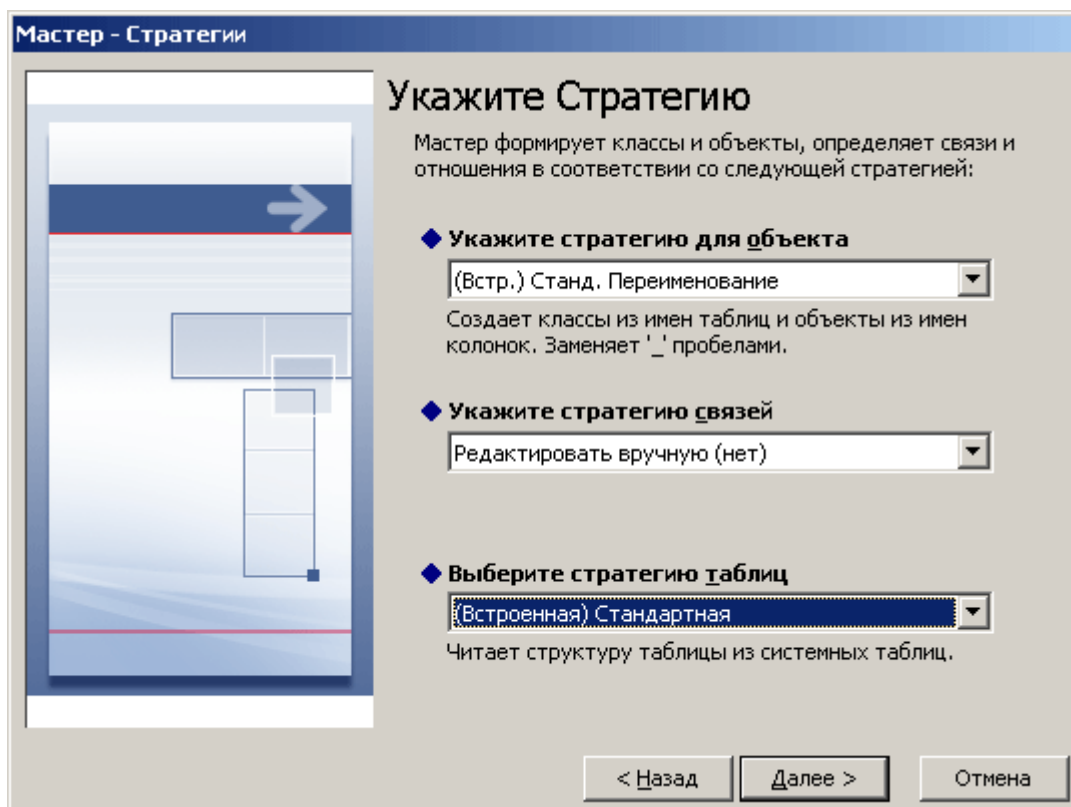
1. Убедитесь, что внешняя стратегия, которую необходимо использовать, выбрана на странице "Стратегии" диалогового окна "Параметры".
Например,
 - Чтобы вставить объекты, извлеченные со стратегией объекта, в меню "Вставка" выберите команду "Вероятные объекты".
 - Чтобы вставить объединения, производные от стратегии объединения, в меню "Инструменты" выберите команду "Обнаружить объединения".
 - Чтобы вставить таблицы, извлеченные со стратегией обозревателя таблиц, в меню "Вставка" выберите команду "Таблицы".

❗ Примечание

При выборе стратегии объединений в средстве создания юниверсов будет использоваться стратегия, чтобы обнаружить возможные объединения и свойства связей. На выбор можно применять или предложенные объединения или мощности множеств. Если требуется автоматически применять вероятное объединение и мощности множеств на базе выбранной стратегии, необходимо выбрать соответствующий параметр создания на странице диалогового окна Параметры (Инструменты > Параметры > база данных). Дополнительную информацию см. в разделе [Использование функций автоматического создания стратегии \[страница 90\]](#).

8.4.11.1 Выбор стратегий в мастере Quick Design

Установленную внешнюю стратегию можно выбрать в мастере быстрого дизайна. Для этого необходимо нажать параметр "Нажмите здесь для выбора стратегий в начальном окне мастера".



8.5 Использование аналитических функций

Средство создания юниверсов поддерживает использование аналитических функций для конкретных реляционных СУБД. Аналитические функции в RedBrick также называют функциями RISQL, а в Teradata – функциями OLAP. С помощью средства создания юниверсов можно определять аналитические функции для объектов в юниверсе.

Пользователи Web Intelligence могут также использовать аналитические функции, чтобы проводить анализ данных, который обычно невозможен с помощью функций создания отчетов в InfoView. Дополнительные сведения об использовании этих и других функций в Web Intelligence см. в разделе *Расчет значений с интеллектуальными мерами* в документе *Использование функций, формул и вычислений в Web Intelligence*.

В данном разделе описан способ определения аналитических, RISQL и OLAP функций для объектов юниверса в следующих реляционных СУБД:

- [IBM DB2 UDB и Oracle \[страница 416\]](#)
- [RedBrick \(функции RSQL\) \[страница 420\]](#)
- [Teradata \(функции OLAP\) \[страница 423\]](#)

8.5.1 Что такое аналитические функции?

Аналитическая функция – это функция, выполняющая аналитическое задание для результирующего множества, которое может подразделяться на упорядоченные группы строк или разделов.

В средстве создания юниверсов можно определять объекты с аналитическими функциями, чтобы рассчитать ранжирование, агрегаты с накоплением и свойства связей в пределах одного или нескольких разделов. В зависимости от реляционной СУБД можно также определять диапазон строк в пределах раздела, к которому необходимо применить анализ.

Полное описание аналитических функций см. в документации по реляционной СУБД.

8.5.2 Какие преимущества дает использование аналитических функций?

Определение объектов с использованием аналитических функций в средстве создания юниверсов дает пользователям Web Intelligence следующие преимущества:

- Сокращение объема работ. Объект определенный с помощью аналитической функции, может проводить анализ данных, для чего на уровне отчетов обычно требуется использование расширенного синтаксиса.
- Дополнительные функциональные возможности. Некоторые задачи анализа данных, такие как вычисление скользящего среднего и применение агрегирования, не доступны в полной мере в InfoView. Объекты, использующие аналитические функции, теперь позволяют пользователям Web Intelligence проводить расширенный анализ данных, что раньше не представлялось возможным.
- Повышенная производительность запроса. Вычисления выполняются на сервере.

8.5.3 Какие семейства аналитических функций поддерживаются?

Можно определить аналитические функции для следующих семейств функций:

- Ранжирование
- Кумулятивное агрегирование
- Мощность множеств, мощность множеств для отчета или агрегирование отчетов

8.5.4 Использование аналитических функций в средстве создания универсов

Аналитическая функция используется при определении аналитической функции для объекта в инструкции SELECT.

В каждом разделе реляционной СУБД в файле параметров (PRM) приводится список аналитических функций, которые могут быть использованы в инструкции SELECT. Этот список может содержать не все доступные функции для каждого семейства и каждой реляционной СУБД, поддерживаемых аналитическими функциями.

8.5.4.1 Что такое PRM-файл?

PRM-файл – это файл параметров, используемый для настройки процессов создания универса и SQL-запроса в Web Intelligence. Для каждой поддерживаемой реляционной СУБД существует свой PRM-файл. PRM-файлы расположены в следующем каталоге:

```
<INSTALLDIR>\dataAccess\RDBMS\connectionServer\<rdbms>\
```

Подробные сведения об изменении файлов параметров см. в документе *Руководство по доступу к данным*.

Прежде чем использовать аналитическую функцию, следует удостовериться в том, что она находится в списке PRM-файла. Если аналитической функции нет в списке PRM-файла, можно добавить имя функции в список. После этого можно использовать эту функцию для объекта в инструкции Select в средстве создания универсов. Дополнительную информацию см. в разделе [Проверка поддержки и добавление аналитической функции в PRM-файл \[страница 417\]](#).

8.5.4.2 Использование аналитических функций каждой реляционной СУБД

Особенности использования аналитических функций будут отмечены для каждой из следующих реляционных СУБД:

- Синтаксис, который можно использовать в инструкции Select для аналитических, RSQL и OLAP функций.
- Как можно проверить и настроить PRM-файл на поддержание аналитических функций, отсутствующих в списке?
- Специальные правила и ограничения использования аналитических функций реляционной СУБД.
- Автоматическая вставка синтаксиса аналитической функции при редактировании инструкций Select.

8.5.5 IBM DB2 UDB и Oracle

Идентичный синтаксис можно использовать для аналитических функций в обеих РСУБД.

8.5.5.1 Определение инструкции SELECT для реляционных СУБД DB2, UDB и Oracle

Можно определять аналитическую функцию в инструкции Select для объекта. Синтаксис необходимо ввести в одном из редакционных полей инструкции Select.

📘 Примечание

Можно автоматизировать запись синтаксиса, добавив аналитические функции в список "Функции" в диалоговом окне "Редактировать инструкцию Select". Чтобы функция была доступна в списке "Функции", необходимо добавить ее в раздел [FUNCTIONS] в PRM- файле. Дополнительные сведения см. в разделе [Автоматическая вставка синтаксической конструкции в инструкции Select \[страница 425\]](#).

Аналитические функции определяются с помощью ключевого слова OVER, например:

```
RANK() OVER (PARTITION BY calender.cal_year ORDER BY SUM(telco_facts.total_billed_rev)DESC)
```

Утверждение, стоящее после ключевого слова OVER, определяет раздел, а также порядок строк в таблице результатов.

Синтаксис для каждого семейства аналитических функций выглядит следующим образом:

Семейство функций	Синтаксис	Описание
Ранжирование	<pre>RANK() OVER(PARTITION BY arg1 ORDER BY arg2 ASC/DESC)</pre>	<ul style="list-style-type: none">arg1 является дополнительным. Если аргумент не включен, то к разделу применяются конечные установки только по умолчанию.arg2 необходим. Ранжирование основывается на значении этого аргумента.Аргументы ASC или DESC определяют, соответственно, будут ли значения сортироваться в восходящем или нисходящем порядке. ASC является значением по умолчанию.

Семейство функций	Синтаксис	Описание
Агрегирование Windows	<pre>SUM(arg1) OVER(PARTITION BY arg2 ORDER BY arg3)</pre>	<ul style="list-style-type: none"> arg1 – аргумент, на котором основывается кумулятивное агрегирование. arg2 – утверждение сброса. Этот аргумент является дополнительным. arg3 – утверждение группы. Этот аргумент является дополнительным.
Агрегирование отчетов	<pre>RATIO_TO_REPORT(arg1) OVER(PARTITION BY arg2)</pre>	<ul style="list-style-type: none"> arg1 – аргумент, на котором основывается мощность множеств. arg2 – утверждение сброса. Этот аргумент является дополнительным.

Использование утверждения Window

Для семейства агрегирования Windows также можно задать утверждение <window clause>, которое определяет диапазон размеров окна после arg3. Например

```
<window frame units> ::= ROW | RANGE <window frame start> ::= UNBOUNDED PRECEDING
| <window frame preceding> | CURRENT ROW <window frame between>
```

Синтаксис утверждения BETWEEN и другие определения размера окна см. в документации по реляционной СУБД.

8.5.5.2 Проверка поддержки и добавление аналитической функции в PRM-файл

PRM-файл для IBM DB2 UDB и Oracle были обновлены с целью поддержки использования аналитических функций.

Однако PRM-файл может содержать не все аналитические функции, доступные в целевой реляционной СУБД. Прежде чем использовать аналитическую функцию, следует убедиться, что она находится в списке раздела реляционной СУБД в PRM-файле. При необходимости, аналитическую функцию можно добавить в список.

Сделать это можно следующим образом:

Чтобы добавить поддержку для аналитической функции в PRM-файл в Oracle или IBM DB2:

1. Просмотрите путь к каталогу Data Access в Business Objects.
2. Откройте PRM-файл для вашей реляционной СУБД в текстовом редакторе.
3. В PRM-файле прокрутите ползунок полосы прокрутки до раздела реляционной СУБД.
4. Убедитесь, что следующие параметры и значения присутствуют:

Параметр и значение в PRM	Описание
OVER_CLAUSE = Y	Создает соответствующий SQL (OVER_CLAUSE).
RISQL_FUNCTIONS = <список используемых функций>	Аналитические функции доступны.

5. Чтобы использовать аналитическую функцию, отсутствующую в списке, необходимо ввести имя функции в конце списка. Например, чтобы использовать функцию RATIO_TO_REPORT, необходимо добавить ее в список в следующем виде:

```
[RDBMS]
(GENERAL)
...
OVER_CLAUSE=Y
RISQL_FUNCTIONS=RANK, SUM, AVG, COUNT, MIN, MAX,
VARIANCE, STDDEV, RATIO_TO_REPORT
```

6. Сохраните изменения и закройте файл.
Чтобы изменения в PRM-файле вступили в силу, необходимо перезапустить средство создания универсов.

8.5.5.3 Правила по использованию аналитических функций реляционных СУБД DB2, UDB и Oracle

Следующие правила применимы при использовании аналитических функций для DB2 UDB и Oracle:

Правило	Описание
Аналитические функции не могут присутствовать в утверждении GROUP BY.	<p>Функции агрегирования (например, SUM), определенные в аналитической функции, используются в утверждении GROUP BY, но аналитическая функция (например, RANK) не используется.</p> <p>Чтобы аналитические функции не использовались в утверждении GROUP BY, они указываются в PRM-файле после параметра RISQL_FUNCTIONS. Для предшествующего ему параметра OVER_CLAUSE необходимо задать значение Y. Это настройка по умолчанию.</p>

Правило	Описание
Аналитические функции не должны создавать утверждение GROUP BY.	Если в раздел "Функции" PRM-файла добавляется аналитическая функция (для заполнения списка функций в диалоговом окне "Правка SQL"), необходимо убедиться, что для GROUP CLAUSE задано значение N. Это предотвратит создание утверждения GROUP BY. Дополнительные сведения см. в разделе Автоматическая вставка синтаксической конструкции в инструкции Select [страница 425] .
Если аналитическая функция использует функцию агрегирования, то все измерения, используемые аналитической функцией, появятся в утверждении GROUP BY.	Например; RANK() OVER (PARTITION BY year ORDER BY SUM(sales)). Утверждение GROUP BY будет содержать год измерения, даже если в запросе используется только функция ранжирования.

8.5.5.4 Ограничения для использования аналитических функций в Oracle и DB2

Существуют следующие ограничения во время использования аналитических функций в IBM DB2 UDB v7.1 и Oracle 8.1.6.

- Нельзя использовать функции @Prompt и @Variable в определении объекта, который также использует аналитические функции.
- Аналитические функции не поддерживаются как объекты пользователя. При добавлении аналитической функции в раздел "Функции" в файл PRM (чтобы заполнить список функций в диалоговом окне "Редактировать SQL") необходимо убедиться, что IN MACRO установлен в N.
- Объекты, которые используют аналитические функции, не могут использоваться в качестве условия или в сортировке. Если конечные пользователи используют данные объекты в качестве условия, они получают сообщение об ошибке SQL. Можно помешать конечному пользователю использовать объекты в условии или в сортировке с помощью редактирования свойств объектов следующим образом.

Предотвращение использования аналитического объекта в условии или сортировке

Чтобы предотвратить использование аналитического объекта в условии или сортировке,

1. Щелкните правой кнопкой мыши объект в средстве создания юниверсов.
2. Выберите в контекстном меню [Свойства объекта](#).
Откроется диалоговое окно [Редактировать свойства](#).
3. В групповом окне [Можно использовать в](#) снимите флажки [Условие](#) и [Сортировка](#).

4. Нажмите **OK**.

8.5.6 RedBrick (функции RISQL)

В следующих разделах описывается, как можно использовать функции RISQL в средстве создания универсов.

8.5.6.1 Определение инструкции SELECT для функций RedBrick RISQL

Можно определять аналитическую функцию в инструкции Select для объекта. Синтаксис необходимо ввести в одно из окон редактирования инструкции Select.

❗ Примечание

Можно автоматизировать запись синтаксиса с помощью добавления функций RISQL в список "Функции" в диалоговом окне "Редактировать инструкцию Select". Чтобы предоставить доступ к функции в списке "Функции", необходимо добавить функцию RISQL в раздел [ФУНКЦИИ] файла PRM. Дополнительные сведения см. в разделе [Автоматическая вставка синтаксической конструкции в инструкции Select \[страница 425\]](#).

Синтаксис каждого семейства функций RISQL описан следующим образом.

Семейство функций	Синтаксис	Описание
Ранжирование (RANK)	RANK(arg1) Например. <code>RANK(SUM(telco_facts.total_billed_rev))</code>	требуется arg1. Ранжирование основано на данном аргументе.
Семейства агрегирования (CUME, MOVINGAVG, MOVINGSUM)	MOVINGSUM(arg1) Например. <code>MOVINGSUM(COUNT(complaints.id), 2)</code>	<ul style="list-style-type: none"> требуется arg1. Накопленное агрегирование на основе данного аргумента. Количество необязательно Это количество предшествующих строк для суммы.
Отношение (RATIOTOREPORT)	RATIOTOREPORT(arg1) Например. <code>RATIOTOREPORT(SUM(telco_facts.total_billed_rev))</code>	требуется arg1. Отношение основано на данном аргументе.

8.5.6.2 Проверка и добавление поддержки функции RISQL в файл PRM

Файл PRM может не содержать все доступные функции RISQL. Перед использованием функции RISQL необходимо проверить, что она указана в разделе RDBMS файла PRM, и, если требуется, добавить в список. Это можно сделать следующим образом.

Чтобы добавить поддержку для аналитической функции в файл Redbrick PRM,

1. Просмотрите каталог "Доступ к данным" в пути Business Objects.
2. Откройте PRM-файл для вашей реляционной СУБД в текстовом редакторе.
3. В PRM-файле прокрутите ползунок полосы прокрутки до раздела реляционной СУБД.
4. Убедитесь, что следующие параметры и значения присутствуют:

Параметр и значение в PRM	Описание
OLAP_CLAUSE = WHEN	Применяет условие.
RISQL_FUNCTIONS = <список использованных функций>	Доступные аналитические функции.

Ниже приводится пример.

```
[RDBMS]
(GENERAL)
...
OVER_CLAUSE=WHEN
RISQL_FUNCTION= RANK, CUME, MOVINGSUM, MOVINGAVG, RATIOTOREPORT, TERTILE
```

5. Если необходимо использовать функцию RISQL, которой нет в списке, введите имя функции в конце списка.
6. Сохраните все изменения и закройте файл.
Чтобы изменения в PRM-файле вступили в силу, необходимо перезапустить средство создания универсов.

8.5.6.3 Правила использования функций RISQL

При использовании функций RISQL применяются следующие правила.

Правило	Описание
Функции RISQL не могут присутствовать в утверждении GROUP BY.	<p>Функции агрегирования (например, SUM), определенные в функции RISQL, используются в утверждении GROUP BY, но аналитическая функция (например, RANK) не используется.</p> <p>Чтобы функции RISQL не использовались в утверждении GROUP BY, они указываются в PRM-файле после параметра RISQL FUNCTIONS. Для предшествующего ему параметра OVER_CLAUSE необходимо задать значение WHEN. Это настройка по умолчанию.</p>
Функции RISQL не должны генерировать утверждение GROUP BY.	<p>Если в раздел "Functions" PRM-файла добавляется функция RISQL (для заполнения списка функций в диалоговом окне "Правка SQL"), необходимо убедиться, что для GROUP CLAUSE задано значение N. Это предотвратит создание утверждения GROUP BY. Дополнительные сведения см. в разделе Автоматическая вставка синтаксической конструкции в инструкции Select [страница 425].</p>
Можно использовать функцию RISQL в условии	Утверждение WHEN сгенерировано.

8.5.6.4 Ограничения использования аналитических функций в RedBrick

Существуют следующие ограничения использования функций RISQL.

- Утверждение RESET BY не поддерживается.
- Утверждение SORT BY не поддерживается. См. в разделе *Ограничения для использования аналитических функций в Oracle и DB2* процедуру для предотвращения использования конечным пользователем объекта в сортировке с помощью редактирования свойств объекта.

8.5.7 Teradata (функции OLAP)

В следующих разделах описывается, как можно использовать функции OLAP в средстве создания юниверсов.

8.5.7.1 Определение инструкции Select для функций Teradata OLAP

Функции отношений недоступны в Teradata V2R3. Можно определять OLAP функцию в инструкции Select для объекта. Синтаксис необходимо ввести в одно из окон редактирования инструкции Select.

Сведения о том, как сделать функцию доступной в списке функций для автоматического ввода синтаксиса, см. в разделе *Ограничения для использования аналитических функций в Oracle и DB2*.

Синтаксис каждого семейства функций OLAP описан следующим образом.

Семейство функций	Синтаксис	Описание
Ранжирование (RANK)	RANK(arg1 DESC/ASC) Например. <pre>RANK(invoice_line.nb_gue sts)</pre>	<ul style="list-style-type: none">требуется arg1. Ранжирование основано на данном аргументе. Аргумент может быть объектом или списком объектов. <div>Примечание Невозможно использовать объект, который использует объект агрегирования (sum, avg, min, count) в качестве arg1.</div> <ul style="list-style-type: none">DESC/ASC указывает порядок ранжирования. ASC – это порядок по умолчанию.
Семейства агрегирования (CSUM, MAVG, MDIFF, MLINREG, MSUM)	CSUM(arg1 DESC/ASC) Например. <pre>CSUM(invoice_line.nb_gue sts)</pre>	<ul style="list-style-type: none">требуется arg1. На данном аргументе основано накопленное агрегирование. Аргумент может быть объектом или списком объектов.DESC/ASC указывает порядок конечных строк. ASC – это порядок по умолчанию.

8.5.7.2 Проверка и добавление поддержки функции OLAP в PRM-файл

Файл PRM для Teradata обновлен для поддержки функций OLAP. Файл PRM может не содержать всех доступных функций OLAP. Перед использованием функции OLAP необходимо проверить, что она указана в разделе RDBMS файла PRM, и, если требуется, добавить в список. Это можно сделать следующим образом:

Чтобы добавить поддержку для аналитической функции в файл Teradata PRM,

1. Просмотрите каталог "Доступ к данным" в пути Business Objects.
2. Откройте PRM-файл для вашей реляционной СУБД в текстовом редакторе.
3. В PRM-файле прокрутите ползунок полосы прокрутки до раздела реляционной СУБД.
4. Убедитесь, что следующие параметры и значения присутствуют:

Параметр и значение в PRM	Описание
OLAP_CLAUSE = QUALIFY	Применяет условие.
RISQL_FUNCTIONS = <список использованных функций>	Доступные аналитические функции.

Ниже приводится пример.

```
[RDBMS]
(GENERAL)
...
OVER_CLAUSE=QUALIFY
RISQL_FUNCTION= RANK, CSUM, MAVG, MDIFF,MLINREG,MSUM,QUANTILE
```

5. Если необходимо использовать функцию RISQL, которой нет в списке, введите имя функции в конце списка.
6. Сохраните все изменения и закройте файл.
Чтобы изменения в PRM-файле вступили в силу, необходимо перезапустить средство создания универсов.

8.5.7.3 Правила использования функций OLAP

В использовании функций OLAP применяются следующие правила.

- Функции OLAP не отображаются в утверждении GROUP BY. Чтобы функции OLAP не использовались в утверждении GROUP BY, они указываются в PRM-файле после параметра RISQL_FUNCTIONS. Для предшествующего ему параметра OVER_CLAUSE необходимо задать значение QUALIFY. Это настройка по умолчанию.
- Невозможно объединить объект, который использует функцию OLAP, с объектом, который использует функцию агрегирования, в одном запросе.
- Можно использовать функции OLAP в условии. Утверждение QUALIFY сгенерировано.
- Можно использовать функции OLAP в утверждении SORT BY.

8.5.7.4 Ограничения использования аналитических функций в Teradata

Существуют следующие ограничения использования функций OLAP.

- Утверждение RESET BY не поддерживается.
- Функции OLAP не используются в подзапросе.
- Функция OLAP не используется в той же инструкции Select в качестве другой функции.
- Функция OLAP не может быть основана на другой функции.
- Функции OLAP не поддерживаются как объекты пользователя.

8.5.8 Автоматическая вставка синтаксической конструкции в инструкции Select

Можно автоматизировать вставку синтаксиса аналитических функций с помощью добавления аналитической функции в поле списка "Функции" в диалоговом окне "Редактировать инструкцию Select".

Заполните поле списка "Функции" с помощью добавления аналитической функции в список функций раздела [ФУНКЦИЯ] в соответствующем разделе PRM целевой RDBMS.

После добавления в файл PRM функция становится доступной в поле списка "Функции" в диалоговом окне "Редактировать инструкцию Select". Дважды щелкнув синтаксис функции, определенный синтаксис вставляется в окно для редактирования.

Добавив аналитическую функцию в файл PRM, необходимо задать следующие значения.

Параметр	Описание
GROUP = N	Аналитические функции и функции RSQL и OLAP не могут генерировать утверждение GROUP BY. Установка значения N предотвращает использование аналитической функции в утверждении GROUP BY.
Только для IBM DB2 UDB v.7.1 и ORACLE 8.1.6: IN_MACRO = N	Это предотвращает аналитическую функцию для DB2 UDB и Oracle от использования в объектах пользователей. Для RedBrick и Teradata данное значение может быть установлено в Y.

Можно добавить аналитическую функцию в раздел [FUNCTION] в файл PRM следующим образом.

Чтобы добавить аналитическую функцию в файл PRM,

1. Просмотрите каталог "Доступ к данным" в пути Business Objects.
2. В текстовом редакторе откройте файл PRM для RDBMS.
3. Прокрутите до раздела FUNCTION файла PRM.
4. Сделайте копию существующей функции и вставьте ее в конец списка.

5. Введите уникальный номер для новой вставленной функции и измените значения в соответствии с аналитической функцией, которую необходимо добавить с список.
6. Назначьте значение GROUP для N.
При использовании IBM DB2 UDB или ORACLE назначьте значение IN_MACRO для N.
Например.

```
(n)
NAME: RANK
TRAD:
HELP: Return the rank of
TYPE=N
IN_MACRO=N
GROUP=N
SQL=
```

7. Сохраните и закройте файл PRM.
Чтобы изменения вступили в силу, необходимо перезапустить средство создания юниверсов.

❗ Примечание

После перезапуска средства создания юниверсов синтаксис добавленной аналитической функции отобразится в соответствующем узле "Тип" (число, символ или дата).

8.6 Использование функции добавления префиксов SQL

Функция добавления префиксов SQL позволяет вставлять SQL-команды перед SQL-инструкциями, созданными в программе BusinessObjects, с помощью параметра юниверса `begin_sql`. Таким образом обеспечивается выполнение этих команд перед всеми созданными инструкциями SQL. Данная функция работает с любой базой данных, поддерживающей добавление параметров перед инструкцией Select. Ниже приведены некоторые примеры.

- Teradata: использование 'QUERY_BAND' для транзакции (относится к документации Teradata)
- Oracle: включение транзакций, доступных только для чтения
- Netezza: изменение параметров оптимизации

Чтобы включить функцию добавления префиксов SQL, укажите параметр `SQL_prefix` для юниверса.

Связанные сведения

[Параметры SQL, задаваемые в пользовательском интерфейсе \[страница 100\]](#)

8.6.1 Добавление префикса перед SQL-инструкцией с помощью параметра юниверса BEGIN_SQL

Предположим, что база данных поддерживает добавление параметров перед инструкцией SELECT.

Параметр BEGIN_SQL позволяет добавлять префикс перед SQL-инструкциями с идентичными параметрами при каждом создании SQL-инструкции.

1. Откройте диалоговое окно [Параметры юниверса](#).
2. Выберите вкладку [Параметр](#).
3. В списке [Параметр](#) выберите параметр BEGIN_SQL и введите необходимые команды префикса.
4. Сохраните настройки.
5. Сохраните юниверс.

Пример

В данном примере используется параметр BEGIN_SQL с базой данных Teradata. Запрос содержит идентификатор пользователя, а идентификатор приложения связан с запросом с целью создания отчетности. На панели [Параметр](#) диалогового окна [Параметры юниверса](#) параметр BEGIN_SQL устанавливается следующим образом:

```
BEGIN_SQL = SET QUERY_BAND = 'UserId=DG12234;AppId=TRM;' FOR TRANSACTION;
```

При выполнении запроса выполняются две SQL-инструкции:

1) Инструкция BEGIN_SQL:

```
SET QUERY_BAND = 'UserId=DG12234;AppId=TRM;' FOR TRANSACTION;
```

2) Результирующая SQL-инструкция основного запроса:

```
SELECT
    RESORT_COUNTRY.COUNTRY, sum(INVOICE_LINE.DAYS * INVOICE_LINE.NB_GUESTS
    * SERVICE.PRICE)
FROM
    COUNTRY RESORT_COUNTRY, INVOICE_LINE, RESORT_COUNTRY.COUNTRY
```

8.7 Оптимизация параметра выборки массива

Параметр "Выборка массива" в файле CS.CFG позволяет устанавливать максимально количество строк, разрешенных в процедуре выборки. Файл CFG – это файл XML, который указывает значения по умолчанию для определенных параметров, которые используются продуктами Business Objects, если запросы выполняются в базе данных.

Параметр "Выборка массива" определяет размер пакета в сети. Например, если установить выборку массива на 20, а планируется извлечь 100 строк, необходимо выполнить пять выборов.

Некоторые источники данных не позволяют изменять размер выборки. В данном случае все строки вернутся в одну выборку. Если необходимо извлечь большие двоичные объекты (BLOB), необходимо задать размер выборки массива на 1.

Если сеть позволяет отправлять большую выборку массива, можно задать новое большее значение (значения могут быть от 1 до 99). Это ускорит процедуру выборки и сократит время выполнения запроса.

8.7.1 Изменение параметра выборки массива

Чтобы изменить параметр "Выборка массива",

1. Откройте файл CS.CFG в редакторе XML.
Файл CFG хранится в следующем каталоге.
<INSTALLDIR>\dataAccess\RDBMS\connectionServer.
2. Поиск параметра "Выборка массива".
3. Задать значение параметра. Сохранить и закрыть файл CFG.
4. Перезапустите средство создания юниверсов.

8.8 Присвоение размера таблицы

Размер таблицы – это мера, отображающая количество строк в таблице. Малые таблицы имеют меньше строк, чем большие. BusinessObjects сортирует малые и большие таблицы по умолчанию (от таблиц с меньшим количеством строк к таблицам с большим количеством). Это определяет порядок таблиц в утверждении FROM из SQL-инструкции.

Порядок, в котором таблицы хранятся на уровне базы данных, зависит от базы данных. Например, Sybase и BusinessObjects используют одинаковый порядок, а Oracle использует противоположный порядок. SQL будет оптимизирована для большинства баз данных, но не для Oracle, где самые маленькая таблица ставится первой в порядке сортировки.

Поэтому с помощью базы данных Oracle можно оптимизировать SQL, изменив порядок, в котором BusinessObjects сортирует таблицы. Для этого необходимо изменить параметр в соответствующем файле PRM базы данных.

8.8.1 Изменение файла PRM для присвоения размеров таблицы

Чтобы изменить файл PRM для присвоения размеров таблицы,

1. Откройте файл PRM базы данных в редакторе XML.
Файл PRM хранится в следующем каталоге.
<INSTALLDIR>\dataAccess\RDBMS\connectionServer\<rdbms>\
Например, файл для Oracle – это oracle.prm:

<INSTALLDIR>\dataAccess\RDBMS\connectionServer\oracle\oracle.prm

2. Найдите параметр REVERSE_TABLE_WEIGHT в разделе файла "Конфигурация".
3. Замените Y на N.
Например параметр отображается в виде REVERSE_TABLE_WEIGHT=N.
Если данной строки нет в файле, значение по умолчанию – Y.
4. Это вынуждает BusinessObjects сортировать таблицы по принципу: от таблиц с большим количеством строк к таблицам с маленьким количеством.
5. Сохранить и закрыть файл .PRM.
6. Чтобы применить изменения в PRM-файле, перезапустите средство создания юниверсов.

8.9 Изменение количества возвращенных строк для таблицы

Также в средстве создания юниверсов можно вручную изменить число строк для любой таблицы. Чтобы просмотреть количество строк в таблице, выберите "Вид" > "Количество строк в таблицах". Количество строк отображается в левом нижнем углу символа каждой таблицы. Можно изменить данное количество следующим образом.

1. Откройте юниверс в средстве создания юниверсов.
2. Правой кнопкой мыши щелкните соответствующую таблицу
3. В контекстном меню выберите параметр "Количество строк в таблице".
Откроется диалоговое окно "Счетчик строк таблицы".
4. Нажмите переключатель "Изменить счетчик строк таблицы вручную".
Слева от диалогового окна отобразится текстовое поле.
5. Введите количество в текстовом поле. Это будет количество строк, необходимое для использования в таблице.
6. Нажмите "ОК", а потом сохраните юниверс.

8.10 Использование объединений быстрого доступа

Объединение быстрого доступа связывает две таблицы, которые уже объединены обычным способом. Можно использовать объединение быстрого доступа, чтобы уменьшить количество таблиц в запросе. Дополнительные сведения см. в разделе [Ограничения для использования внешних объединений \[страница 174\]](#).

❗ Примечание

Объединения быстрого доступа не создают циклы.

9 Работа с юниверсами OLAP

9.1 Юниверсы OLAP

9.1.1 Что такое юниверс OLAP?

Юниверс OLAP является юниверсом Business Objects, созданным из куба OLAP или запроса. Юниверс создается автоматически из выбранного соединения с источником данных OLAP.

Когда юниверс создан, его можно экспортировать на сервер центрального управления, как и любой другой юниверс. В таком случае юниверс будет доступен пользователям Web Intelligence для создания отчетов и выполнения запросов.

Создание и поддержка юниверса OLAP может осуществляться следующими способами:

- Чтобы создать юниверс OLAP, сначала выберите источник данных OLAP.

ⓘ Примечание

В случае защищенного соединения с источником данных OLAP для создания юниверса или обновления его структуры пользователь должен иметь право [Загрузка](#) в отношении соединения. Это право предоставляется в СМС администратором.

- Создайте соединение с источником данных, используя мастер нового соединения, и выберите соединение для нового юниверса.
В средстве создания юниверсов юниверс создается автоматически. Структуры OLAP вносятся напрямую в классы, меры, измерения, подробности и фильтры юниверса. Структура юниверса отображается в панели юниверса.
- Можно сохранить и экспортировать юниверс OLAP на сервер центрального управления.
- Можно изменять все любые компоненты юниверса OLAP.
- Обновление мастера юниверса OLAP позволяет управлять жизненным циклом юниверса OLAP. Мастер автоматически обновляет структуру юниверса, внося изменения, сделанные в источнике данных OLAP. Этот мастер отличает генерируемые объекты от добавленных или измененных вручную объектов, позволяя сохранять изменения, сделанные вручную в средстве создания юниверсов.

Связанные сведения

[Источники данных OLAP, которые можно использовать для создания юниверса \[страница 431\]](#)

[Соединения с источниками данных OLAP \[страница 437\]](#)

[Функции средства создания юниверсов, поддерживаемые для юниверсов OLAP \[страница 450\]](#)

[Управление жизненным циклом юниверса OLAP \[страница 464\]](#)

9.1.2 Источники данных OLAP, которые можно использовать для создания юниверса

Можно автоматически создавать юниверсы из следующих источников данных OLAP.

SAP Business Warehouse (BW)

Службы анализа Microsoft (MSAS) 2000

Службы анализа Microsoft (MSAS) 2005

Hyperion Essbase

❗ Примечание

Чтобы подключиться к источнику данных Essbase OLAP из продукта SAP BusinessObjects OLAP (в том числе из средства создания юниверсов, Web Intelligence Rich Client и Web Intelligence), проверьте правильность установки и настройки промежуточного ПО клиента Essbase на компьютерах, на которых размещаются эти продукты SAP BusinessObjects OLAP. В частности, убедитесь, что переменные среды клиента Essbase ARBORPATH и ESSBASEPATH созданы и установлены в качестве системных переменных среды Windows (в отличие от пользовательских переменных среды Windows).

Один юниверс генерируется автоматически из одного куба или запроса. Юниверсы OLAP поддерживают одиночный куб в юниверсе.

Связанные сведения

[Источники данных SAP Business Warehouse \(BW\) \[страница 431\]](#)

[Сопоставление и использование объектов SAP BW в юниверсе \[страница 495\]](#)

[Функции MSAS, поддерживаемые юниверсами OLAP \[страница 435\]](#)

[Сопоставление кубов MSAS с компонентами юниверса \[страница 505\]](#)

[Функции Essbase, поддерживаемые юниверсами OLAP \[страница 436\]](#)

[Сопоставление кубов Essbase и компонентов юниверса \[страница 504\]](#)

9.1.2.1 Источники данных SAP Business Warehouse (BW)

При создании юниверса OLAP на основе источника данных BW юниверс можно построить непосредственно на основе куба InfoCube/MultiCube или же на основе запроса BEx, использующего любой инфо-провайдер. Инфо-провайдер может быть:

InfoCube

MultiCube или несколько инфо-провайдеров

Оперативное хранилище данных (ODS)

InfoSet

Связанные сведения

Объекты InfoCube приложения SAP Business Warehouse (BW) в качестве источников данных [страница 432]

Запросы SAP BW как источники данных [страница 432]

Запросы как рекомендуемые источники данных [страница 434]

9.1.2.1.1 Объекты InfoCube приложения SAP Business Warehouse (BW) в качестве источников данных

Следующие типы объектов InfoCube поддерживаются в качестве источника данных для построения юниверсов OLAP.

- Стандартные и транзакционные InfoCubes: данные и метаданные хранятся в одной системе SAP Business Warehouse (BW).
- Удаленный InfoCube: данные хранятся в удаленной системе.

ⓘ Примечание

Построение и разворачивание поддерживаемых юниверсов на удаленном InfoCubes не рекомендуется для специальных сценариев запроса, отчета, и анализа. Такая архитектура не отвечает ожиданиям эксплуатации производительности запроса с интерактивными запросами.

- MultiCubes и несколько инфо-провайдеров

ⓘ Примечание

Построение и разворачивание юниверсов бизнес-представлений на верхушке MultiCube или несколько инфо-провайдеров соответствует построению и разворачиванию юниверса на верхушке InfoCube.

Все характеристики, иерархии, контрольные цифры, включая время и единицу, в InfoCube отображаются в юниверсе.

9.1.2.1.2 Запросы SAP BW как источники данных

Клиенты SAP BW используют запросы BEx для доступа к инструментам фронтэнда SAP Business Explorer.

ⓘ Примечание

Для использования в качестве источника данных, доступного через OLAP для юниверсов Business Objects, запрос BEx должен быть доступен для OLE DB для OLAP. Внешний доступ к запросу BEx можно разрешить в SAP BW Query Designer на вкладке *Расширенный режим* в диалоговом окне *Свойства запроса*.

Все объекты InfoObjects в запросе BEx, выбранные в качестве строк, столбцов и свободных характеристик, отображаются в универсе. К ним относятся характеристики, иерархии, контрольные цифры, структуры и переменные.

Объекты InfoSets и оперативные хранилища данных (ODS) могут отображаться в универсах через запросы BEx.

Запросы, основанные на ODS

ODS может отображаться в универсе через запрос BEx.

Объекты ODS часто используются для управления данными уровня детальной транзакции до агрегирования в InfoCubes. Включение объектов ODS в дизайн хранилища данных технологической платформы SAP NetWeaver – это способ уменьшить размер InfoCube и улучшить загрузку и производительность запроса.

❗ Примечание

ODS – это крупная, подробная реляционная структура. Получение доступа к ODS с помощью интерфейса OLAP BAPI не улучшает производительность запроса. Рассматривайте данные альтернативы как возможность конечного пользователя быстро доставить отчеты.

- Создайте прямой доступ к ODS с помощью вызовов BAPI
- Получите доступ к таблицам ODS, используя SQL в Web Intelligence

Запросы, основанные на InfoSet

InfoSet может отображаться в универсе через запрос BEx.

Наборы данных иногда определяются в SAP BW для вывода основных данных в отчет.

❗ Примечание

Основные данные можно выводить в отчет, если универс будет основан на кубе InfoCube, при этом не потребуется использовать наборы InfoSet и запросы BEx. Основная разница между этими двумя подходами заключается в том, что основные данные, получаемый для отчета из кубов InfoCube, ограничивают данные только допустимыми транзакциями.

Связанные сведения

[Запросы как рекомендуемые источники данных \[страница 434\]](#)

9.1.2.1.3 Запросы как рекомендуемые источники данных

Запросы BEx рекомендуется использовать в качестве источников данных для создания юниверсов Business Objects по следующим причинам.

- Не все свойства метаданных SAP BW извлекаются на уровне InfoCube, как представлено в следующей таблице.

Функция метаданных BW	Уровень поддержки интерфейса программирования бизнес-приложений (BAPI) OLAP для SAP
Характеристики (включая время и единицу)	InfoCube/запрос BEx
Иерархии	InfoCube/запрос BEx
Основные контрольные цифры	InfoCube/запрос BEx
Навигационные атрибуты	Только запрос BEx
Атрибуты отображения	InfoCube/запрос BEx
Вычисляемые контрольные цифры / формулы	Только запрос BEx
Ограничиваемые контрольные цифры	Только запрос BEx
Пользовательские структуры	Только запрос BEx
Переменные	Только запрос BEx

- Запросы BEx реализуют гибкое расширение среды моделирования данных. При работе с InfoCubes требуется больше усилий для изменений.
- Запросы BEx реализуют эффективные функции создания настраиваемых источников данных, отвечающих требованиям конечных пользователей.

Хотя у запросов BEx имеются преимущества в качестве источников данных, запросы BEx не обязательно использовать в каждом из отчетов, а для каждого из существующих запросов BEx не обязательно наличие юниверса. Чтобы свести к минимуму затраты на обслуживание, сконцентрируйте стратегию реализации на ограничении конечного числа запросов BEx и юниверсов, необходимых для удовлетворения всех возникающих потребностей в запросах и отчетах. Учитывайте следующие моменты, чтобы сократить количество требуемых юниверсов.

Если Web Intelligence является инструментом интерфейса, пользователь не ограничивается форматом выходных данных в запросе BEx.

Работа с юниверсами OLAP, созданными из больших запросов BEx, напрямую не влияет на производительность. Объекты юниверса OLAP, не вставленные в запрос Web Intelligence, не влияют на производительность запроса.

📌 Примечание

Business Objects рекомендует иметь несколько запросов BEx – от одного до пяти – на каждый куб InfoCube или MultiCube, находящийся в области действия специального запроса или отчета. Юниверс можно построить над каждым из этих запросов BEx.

9.1.2.1.4 Многоязычные юниверсы SAP BW

С помощью Web Intelligence можно управлять многоязыковыми возможностями SAP BW. Для внедрения многоязыковой среды система NetWeaver BW должна содержать многоязыковые данные и метаданные.

Необходимо создать юниверс для каждого языка, поддерживаемого этим решением. Язык, с помощью которого создается соединение юниверса, определяет язык, с помощью которого создается юниверс.

SAP-аутентификация пользователя определяет язык данных, возвращенных в запрос. Пользователю необходимо входить в систему InfoView, используя аутентификацию SAP, и указать необходимый язык для результатов, возвращенных с сервера SAP.

Язык набора результатов зависит от поддержки Unicode системой SAP. Если в системе SAP не содержатся данные на необходимом языке, данные на этом языке являются недоступными в Web Intelligence. Web Intelligence восстанавливает отображение технических имен вместо описаний, когда описания не переводятся в BW.

9.1.2.1.5 Необходимые условия использования SAP BW в средстве создания юниверсов

При создании юниверса с помощью источника данных SAP BW необходимо включить SSO (единый вход) во время просмотра. SSO позволяет пользователям входить в систему SAP BusinessObjects Enterprise со своими учетными данными SAP и управлять аутентификацией SAP.

Чтобы включить SSO для юниверсов OLAP на верхушке SAP необходимо установить интеграцию SAP и настроить программный подключаемый модуль безопасности SAP.

После настройки интеграции безопасности SAP можно запустить средство создания юниверсов с помощью учетных данных SAP. Создайте имя пользователя BusinessObjects Enterprise, объединив идентификатор системы SAP и идентификатор клиента SAP, как определено при настройке интеграции безопасности с помощью идентификатора пользователя SAP.

Дополнительные сведения см. в руководствах пользователя *Интеграция Business Objects XI для установки решений SAP* и *Интеграция Business Objects XI для решений SAP*.

9.1.2.2 Функции MSAS, поддерживаемые юниверсами OLAP

В следующей таблице содержатся сводки по уровню поддержки или функций MSAS для юниверсов, сгенерированных из источника данных MSAS.

Функция метаданных MSAS	Уровень поддержки юниверса OLAP
Куб	Поддерживается
Локальный куб	Поддерживается
Виртуальный куб (MSAS 2000)	Поддерживается

Функция метаданных MSAS	Уровень поддержки юниверса OLAP
Перспектива (MSAS 2005)	Поддерживается
Измерения	Поддерживается
Виртуальные измерения (MSAS 2000)	Поддерживается
Иерархии	Поддерживается
Уровни	Поддерживается
Свойство уровня	Поддерживается
Атрибуты (MSAS 2005)	Поддерживается
Меры	Поддерживается
Группа мер (MSAS 2005)	Поддерживается
Вычисляемые меры	Поддерживается
Отображаемый каталог (MSAS 2005)	Поддерживается
KPI (MSAS 2005)	Не поддерживается
Действие	Не поддерживается
Порядок сортировки базы данных	Необходимо определить порядок пользовательской сортировки в Web Intelligence
Обратная запись	Не поддерживается

Связанные сведения

[Сопоставление кубов MSAS с компонентами юниверса \[страница 505\]](#)

9.1.2.3 Функции Essbase, поддерживаемые юниверсами OLAP

Данная таблица суммирует уровень поддержки функций Essbase для юниверсов, созданных из источника данных Hyperion Essbase.

Функции метаданных Essbase	Уровень поддержки юниверса OLAP
Режим хранения блока	Поддерживается
Режим хранения агрегата	Поддерживается
Гибридный режим	Не поддерживается
Таблицы псевдонимов	Поддерживается
Измерения	Поддерживается
Измерения атрибута	Поддерживается

Функции метаданных Essbase	Уровень поддержки юниверса OLAP
Дублирование элементов	Поддерживается
Поколения	Поддерживается
Уровни	Не поддерживается
Атрибуты определения пользователя (UDA)	Не поддерживается
Динамический временной ряд (DTS)	Не поддерживается
Переход по иерархии через сервис интеграции Essbase (EIS)	Не поддерживается
Переменные замены	Не поддерживается
Связанные разделы	Не поддерживается
Связанные объекты отчета (LRO)	Не поддерживается
Порядок сортировки базы данных	Необходимо определить порядок пользовательской сортировки в Web Intelligence
Обратная запись	Не поддерживается

Связанные сведения

[Сопоставление кубов Essbase и компонентов юниверса \[страница 504\]](#)

9.2 Определение соединений для источника данных OLAP

9.2.1 Соединения с источниками данных OLAP

Для создания юниверса OLAP сначала необходимо определить соединение для источника данных OLAP. Определите одно соединение для каждого куба или запроса, который необходимо использовать для создания юниверса.

Используйте мастер нового соединения для определения соединения. Мастер содержит следующие шаги в создании соединения.

Запустите мастер создания нового соединения в средстве создания юниверсов.

Укажите имя соединения и выберите компонент доступа к базе данных.

Определите параметры входа для соединения. Эти параметры меняются в зависимости от выбранного компонента доступа к базе данных.

Выберите куб или запрос для создания юниверса.

Определение срока службы соединения.

Определение пользовательских параметров. Эти параметры меняются в зависимости от выбранного компонента доступа к базе данных.

Определение соединения – это первый шаг в создании юниверса OLAP. После определения соединения средство создания юниверсов автоматически создает юниверс.

❗ Примечание

Если соединение определяется из списка соединений в меню "Инструменты", необходимо отдельно создать юниверс.

Связанные сведения

[Запуск мастера нового соединения \[страница 438\]](#)

[Выбор программного обеспечения среднего яруса базы данных для соединения OLAP. \[страница 439\]](#)

[Параметры входа для соединений SAP BW OLAP \[страница 442\]](#)

[Параметры входа для соединений MSAS OLAP \[страница 443\]](#)

[Определение параметров входа в систему для соединений Essbase \[страница 444\]](#)

[Выбор куба или запроса источника для соединения OLAP \[страница 445\]](#)

[Определение параметров конфигурации для соединений OLAP \[страница 445\]](#)

[Определение пользовательских параметров для соединений Essbase \[страница 447\]](#)

9.2.2 Запуск мастера нового соединения

Для запуска мастера нового соединения выполните одно из следующих действий.

Начать с...	Выполнить...
Значок Новый юниверс	Нажмите значок <i>Новый юниверс</i> , а затем <i>Создать...</i> на странице <i>Определение</i> окна <i>Параметры юниверса</i> .
Меню "Файл"	В пустом сеансе выберите ► <i>Файл</i> ► <i>Параметры</i> ►, а затем щелкните <i>Создать...</i> на странице <i>Определение</i> окна <i>Параметры юниверса</i> .
Мастер быстрого дизайна	Если включен мастер быстрого дизайна, он запускается автоматически при запуске средства создания юниверсов. Нажмите <i>Создать...</i> в первом шаге мастера Quick Design.

❗ Примечание

Если мастер был отключен, нажмите ► *Файл* ► *Создать* ►. Если мастер не запускается при нажатии ► *Файл* ► *Создать* ►, выберите ► *Инструменты* ► *Параметры* ►. На *главной* странице диалогового окна *Параметры* установите

Начать с...

Выполнить...

флажок *Запуск мастера быстрого дизайна Файл/Создать*. Нажмите *ОК*, затем **Файл** **Создать**.

Меню "Инструменты"

Нажмите **Инструменты** **Соединения**. Нажмите *Добавить...* в диалоговом окне *Мастер соединений*.

9.2.3 Выбор программного обеспечения среднего яруса базы данных для соединения OLAP.

На странице мастера новых соединений *Выбор ПО среднего яруса* для базы данных введите название соединения, выберите тип и программное обеспечение среднего уровня для базы данных для соединения.

Параметр ПО среднего яруса для базы данных	Описание
<i>Тип соединения</i>	<p>Выберите <i>Безопасное</i> для управляемого доступа к соединению (рекомендуется).</p> <p>Выберите <i>Общедоступное</i> для разрешения неуправляемого доступа к соединению для всех пользователей.</p> <p>Выберите <i>Персональное</i> для предотвращения доступа к создателю юниверса. Необходимо использовать только персональные соединения для получения доступа к личным данным локальной машины.</p>
<i>Имя соединения</i>	Введите имя соединения.
<i>Фильтровать сетевые уровни хранимых процедур</i>	Параметр <i>Фильтровать сетевые уровни хранимых процедур</i> не используется для соединений OLAP.
Список доступных драйверов доступа к данным.	<p>На этой странице выводятся базы данных и ПО среднего яруса, которые соответствуют ключу драйвера доступа к данным.</p> <p>Разверните узел целевой базы данных, чтобы отобразить поддерживающее ПО среднего яруса для базы данных.</p> <p>Разверните узел ПО, чтобы отобразить драйвер доступа к данным для ПО среднего яруса OLAP.</p> <p>Выберите драйвер доступа к данным.</p>

9.2.4 Параметры входа для реляционных соединений – UDT


Перечисленные ниже параметры входа применяются для большей части реляционных соединений. Чтобы использовать UDT, необходимо установить клиенты Business Objects в 64-разрядной системе вместо 32-разрядной.

Параметр	Описание
<i>Режим аутентификации</i>	<p>Метод аутентификации учетных данных пользователей при обращении к источнику данных:</p> <ul style="list-style-type: none">• <i>Использовать указанные имя пользователя и пароль</i>: используются параметры <i>Имя пользователя</i> и <i>Пароль</i>, определенные для соединения.• <i>Использовать сопоставление учетных данных BusinessObjects</i>: используются учетные данные для базы данных с учетной записью пользователя, определенной на центральном сервере управления для подключения к источнику данных. Учетные данные для базы данных устанавливаются в свойствах пользователя в консоли Central Management Console. Для получения дополнительных сведений см. <i>Руководство администратора платформы SAP BusinessObjects Business Intelligence</i>.• <i>Использовать единый вход</i>: этот режим аутентификации используется для поддержки сквозного единого входа, определенного на центральном сервере управления (CMS). При использовании внешнего источника аутентификации (например, LDAP) необходимо настроить сервер CMS и источник данных таким образом, чтобы использовать этот внешний источник аутентификации. Для получения дополнительной информации о едином входе см. <i>Руководство администратора платформы SAP BusinessObjects Business Intelligence</i>.• <i>Использовать единый вход при обновлении отчетов во время просмотра</i>: Если сеанс открыт с помощью единого входа (SSO) Windows Active Directory и для некоторых соединений определено использование SSO, выберите этот режим аутентификации при подключении к репозиторию CMS. Дополнительные сведения см. в <i>Руководстве администратора платформы SAP BusinessObjects Business Intelligence</i>.
<i>Имя пользователя</i>	Имя пользователя для доступа к источнику данных в случае, если параметр <i>Режим аутентификации</i> имеет значение <i>Использовать указанные имя пользователя и пароль</i> .
<i>Пароль</i>	Пароль для доступа к источнику данных в случае, если параметр <i>Режим аутентификации</i> имеет значение <i>Использовать указанные имя пользователя и пароль</i> .
<i>Использовать SSL</i>	<p>Этот параметр применяется только к соединениям SAP HANA.</p> <p>Если этот параметр выбран, для подключения к серверу используется протокол SSL.</p>

Параметр	Описание
<i>Один сервер</i>	<p>Этот параметр применяется только к соединениям SAP HANA.</p> <p>Выберите этот параметр, если устанавливается соединение только с одним сервером базы данных SAP HANA.</p> <ul style="list-style-type: none"> <i>Имя хоста</i>: имя сервера, на котором размещается источник данных. Не добавляйте номер порта. <i>Номер экземпляра</i>: номер экземпляра SAP HANA представлен второй и третьей цифрами номера порта. Должен находиться в диапазоне от 00 до 99. Например, если номер порта – 30215, то номер экземпляра – 02.
<i>Несколько серверов</i>	<p>Этот параметр применяется только к соединениям SAP HANA.</p> <p>Выберите этот параметр, чтобы использовать механизм восстановления после отказа SAP HANA.</p> <p><i>Сервер (хост:порт{;хост:порт})</i>: список серверов, разделенных точкой с запятой, например: (хост1:30015;хост2:30015;хост3:30015). JDBC-драйвер выберет один из этих хостов для подключения. Если хост недоступен, драйвер выберет в списке следующий хост.</p> <p>В это поле можно ввести имя хоста и порт только для одного сервера.</p>
<i>Сервер (<хост>:<порт>)</i>	<p>Имя и порт сервера, на котором размещается источник данных.</p> <p>Для соединений Oracle можно ввести список серверов, разделив их запятой: (<хост>:<порт>, <хост>:<порт>).</p>
<i>Сервер</i>	Имя сервера, на котором размещается источник данных.
<i>База данных</i>	Имя базы данных.
<i>Имя источника данных</i>	Для соединений с ODBC это имя источника данных, определенное в диспетчере источников данных используемой операционной системы.
<i>Псевдоним</i>	Для DB2-соединений это псевдоним базы данных, созданной в помощнике по настройке DB2.
<i>Сетевая служба</i>	Для соединений Oracle с использованием микропрограммного обеспечения JDBC – имя сетевой службы Oracle.
<i>Служба</i>	Для соединений Oracle – псевдоним, который содержит сведения об IP-адресе сервера и сетевой службе.
<i>JDBC_URL</i> <i>JDBC_CLASS</i>	Для универсальных JDBC-соединений – URL-адрес и класс JDBC, которые используются для подключения к базе данных.
<i>Сервер Informix</i>	Для соединений с Informix это определенное пользователем имя сервера Informix.
<i>Имя поставщика OLE DB</i>	Для универсальных поставщиков OLE DB – имя поставщика.

9.2.5 Параметры входа для соединений SAP BW OLAP

Диалоговое окно [Параметры входа в систему](#) мастера [нового соединения](#) может содержать следующие параметры:

Параметр входа в систему	Описание
Режим аутентификации	<ul style="list-style-type: none">• Использовать указанные имя пользователя и пароль: используйте учетные данные для входа в качестве аутентификации.• Использовать сопоставление учетных данных Business Objects: для обновления отчета у пользователя запрашивается пароль пользователя для базы данных, связанный с его учетной записью Business Objects. Это множество, использующее параметры dbuser и dbpass. Это множества на уровне администрирования. Дополнительные сведения о настройке данного параметра см. в документе <i>Руководство администратора Business Objects Enterprise</i>.• Использовать единый вход при обновлении отчетов во время просмотра: если выбраны имя пользователя и пароль, используемые для доступа в CMS, то они же автоматически используются в качестве параметров входа в базу данных. Для получения сведений о настройке единого входа (SSO) см. <i>Руководство администратора Business Objects Enterprise</i>.
При наличии использовать SNC	Установите этот флажок, чтобы использовать SNC.
Клиент	Номер, по которому клиент идентифицируется в системе SAP BW (обязательный параметр).
Имя пользователя	Имя пользователя, используемое для доступа к серверу OLAP, когда выбран режим аутентификации Использовать указанные имя пользователя и пароль .
Пароль	Пароль, используемый для доступа к серверу OLAP, когда выбран режим аутентификации Использовать указанные имя пользователя и пароль .
Язык	Язык, который будет использоваться в соединении. <div> Примечание Язык соединения определяет язык, с помощью которого создается юниверс.</div>
Сохранить язык	Язык, который будет использоваться для соединения: <ul style="list-style-type: none">• Если установлен флажок Сохранить язык, будет использоваться значение из поля Язык.• Если флажок Сохранить язык снят, будет использоваться значение из сеанса пользователя.

Параметр входа в систему	Описание
<i>Режим входа</i> или <i>Тип сервера</i>	<p>Выберите <i>Сервер приложений</i> для прямого подключения к серверу SAP без использования распределения нагрузки.</p> <p>Выберите <i>Сервер сообщений</i>, чтобы воспользоваться преимуществами распределения нагрузки SAP.</p>
<i>Сервер приложений</i>	Выберите или введите имя или IP-адрес сервера приложений SAP (требуется для режима входа в систему сервера приложений).
<i>Системный номер</i>	Введите системный номер, например 00 (требуется для режима входа в систему сервера приложений).
<i>Системный идентификатор</i>	Укажите <i>Сервер сообщений</i> , <i>Группу входа</i> , а при необходимости – и <i>Системный идентификатор</i> при использовании режима входа <i>Сервер сообщений</i> .
<i>Группа регистрации</i>	
<i>Сервер сообщений</i>	

9.2.6 Параметры входа для соединений MSAS OLAP

Диалоговое окно *Параметры входа в систему* мастера *нового соединения* может содержать следующие параметры:

Параметр входа в систему	Описание
<i>Режим аутентификации</i>	<ul style="list-style-type: none"> <i>Используйте указанные имя пользователя и пароль</i>: используйте учетные данные для входа в качестве аутентификации. <i>Использовать сопоставление учетных данных Business Objects</i>: для обновления отчета у пользователя запрашивается пароль пользователя для базы данных, связанный с его учетной записью BusinessObjects. Это множество, использующее параметры <i>dbuser</i> и <i>dbpass</i>. Это множества на уровне администрирования. Подробнее о настройке данного параметра см. в документе <i>Руководство администратора платформы SAP Business Objects Business Intelligence</i>. <i>Использовать единый вход при обновлении отчетов во время просмотра</i>: если выбраны имя пользователя и пароль, используемые для доступа в CMS, то они же автоматически используются в качестве параметров входа в базу данных. Подробнее о настройке единого входа (SSO) см. в документе <i>Руководство администратора платформы SAP Business Objects Business Intelligence</i>.

Параметр входа в систему	Описание
<i>Сервер</i>	<p>Введите одно из следующих значений:</p> <ul style="list-style-type: none"> • URL-адрес библиотеки MSAS, настроенной и предоставляемой сервером MSAS. • Имя сервера для источника данных MSAS • Полный путь и имя файла куба MSAS. Введите полный путь к файлу, заключив его в двойные кавычки, например: "Z:\All cubes\test.cub" <div> <p>Примечание</p> <p>Если файл с кубом расположен в системе, отличной от узла SAP BusinessObjects, эти два компьютера должны использовать общее соединение. Необходимо создать соединение с файлом куба данных непосредственно на узле SAP BusinessObjects.</p> </div>
<i>Имя пользователя</i>	Имя пользователя, используемое для доступа к серверу OLAP, когда выбран <i>режим аутентификации Использовать указанные имя пользователя и пароль</i> .
<i>Пароль</i>	Пароль, используемый для доступа к серверу OLAP, когда выбран <i>режим аутентификации Использовать указанные имя пользователя и пароль</i> .
<i>Язык</i>	Язык, который будет использоваться в соединении.

9.2.7 Определение параметров входа в систему для соединений Essbase

На странице *Параметры входа в систему* мастера новых соединений определите элементы конфигурации входа в систему для соединения с базой данных Essbase.

Параметр входа в систему	Описание
<i>Режим аутентификации</i>	<p>Для того чтобы от пользователя требовалось введение данных для входа в систему при использовании данного соединения, выберите параметр <i>Использовать заданные имя пользователя и пароль</i>. Для синхронизации защиты Essbase с системой BusinessObjects Enterprise введите Essbase DBUser и DBpass в качестве <i>Имени пользователя</i> и <i>Пароля</i>.</p> <p>Чтобы применять для данного соединения мандат пользователя для входа в систему BusinessObjects Enterprise, выберите параметр <i>Использовать сопоставление из мандата BusinessObjects</i>.</p> <div> <p>Примечание</p> <p>Для соединений Essbase не поддерживается параметр <i>Использовать единый вход при обновлении отчетов во время просмотра</i>.</p> </div>

Параметр входа в систему	Описание
Имя пользователя	Введите значение Essbase DBUser.
Пароль	Введите значение Essbase DBpass.
Сервер	Введите имя сервера Essbase.

9.2.8 Выбор куба или запроса источника для соединения OLAP

Обозреватель куба отображает доступные кубы OLAP целевого сервера.

Разверните узлы куба, чтобы отобразить доступные кубы и запросы. Обозреватель содержит следующие средства для помощи в поиске:


Средство обозревателя куба	Описание
Избранное	Каталог, содержащий ссылки на выбранные кубы, для быстрого доступа к ним. Чтобы добавить куб в "Избранное", нажмите куб правой кнопкой мыши в обозревателе куба OLAP и нажмите "Добавить в избранное" в контекстном меню.
Поиск	Выполняет поиск имен доступных кубов или запросов для текстовой строки. Введите строку в текстовом поле и нажмите "Поиск". Каждый найденный экземпляр будет подсвечен. Нажмите "Поиск", чтобы продолжить.
Каталог \$INFOCUBE	Источники данных SAP BW, InfoCube и MultiCube группируются в папки, называемые \$INFOCUBE.

Выберите куб или запрос для создания юниверса.

9.2.9 Определение параметров конфигурации для соединений OLAP

На странице [Параметры конфигурации](#) в мастере новых соединений определите параметры соединения для управления сроком его службы. Можно принять установки по умолчанию при создании соединения и изменить их позже.

Параметр конфигурации	Описание
<i>Режим пула соединений</i>	<p>Срок службы определяется при помощи параметров <i>Режим пула соединений</i> и <i>Время ожидания пула</i>.</p> <p>По умолчанию <i>Режим пула соединений</i> устанавливается в вариант <i>Сохранять соединение активным в течение</i>, по умолчанию значение параметра <i>Время ожидания пула</i> равно 10 минутам.</p> <div> <p>Примечание</p> <p>В системе Business Objects рекомендуется оставлять срок службы соединения по умолчанию. Процесс построения юниверса существенно замедляется, если выбран <i>Режим пула соединений</i> разрывать соединение после каждой транзакции. Разрыв соединения после каждой транзакции также влияет на ключевые потоки работ конечного пользователя, такие как работа с иерархическими списками значений.</p> </div> <p>Срок службы соединения может иметь значительное влияние при работе с системой SAP BW.</p> <p>Тем не менее срок существования соединений может также влиять на обновление существующих юниверсов при изменениях в запросах BEx. Это происходит по той причине, что интерфейс OLAP BAPI формирует кэш метаданных со стороны клиента каждый раз при установлении соединения с системой SAP BW. Этот кэш очищается только при завершении соединения.</p> <p>Чтобы минимизировать риск потери синхронизации кэша метаданных при обновлении запросов SAP BEx, можно изменить <i>Время ожидания пула</i> с 10 минут на 1 минуту.</p> <p>При параллельном редактировании запросов BW и сопоставлении новых юниверсов этим запросам перед построением новых юниверсов рекомендуется закрыть средство создания юниверсов (чтобы соединения с юниверсами также закрылись, а кэш метаданных был очищен), чтобы были учтены изменения, внесенные перед этим на стороне запросов BEx.</p>
<i>Время ожидания пула</i>	
<i>Размер выборки из массива</i>	С помощью параметра <i>Размер выборки массива</i> можно задать оптимальное число строк, которое будет использоваться в процедуре FETCH.
<i>Размер привязки массива</i>	Параметры <i>Размер временного массива</i> и <i>Время ожидания входа в систему</i> не используются для соединений OLAP.
<i>Время ожидания входа в систему</i>	

Параметр конфигурации	Описание
Максимальное количество параллельных запросов	Параметр Максимальное количество параллельных запросов позволяет задать максимальное число задач, которые могут выполняться параллельно для соединения, в диапазоне от 1 до 64.
<div>  Примечание </div> <p>Если максимальное количество параллельных запросов для одного соединения превышает максимальное количество параллельных заданий, установленное в Cental Management Console для приложения Web Intelligence, применяется меньшее из этих двух ограничений.</p>	

9.2.10 Определение пользовательских параметров для соединений Essbase

На странице [Параметры конфигурации](#) мастера новых соединений укажите таблицу под псевдонимом и укажите размерность, которая будет использоваться при формировании юниверса в качестве параметра "Мера, определяемая количеством измерений".

Параметр входа в систему	Описание
Таблица под псевдонимом	Чтобы создать юниверс на основе таблицы под псевдонимом, отличной от установленной по умолчанию, выберите нужную таблицу из списка.
Мера, определяемая количеством измерений	Укажите меру, которая будет использоваться в качестве параметра "Мера, определяемая количеством измерений". Средство создания юниверсов создает элементы измерения при выборе мер в юниверсе.

9.3 Редактирование юниверсов OLAP

9.3.1 Создание юниверсов OLAP с дополнительными параметрами

Эта функция применяется только к юниверсам OLAP и позволяет определять дополнительные параметры метаданных при создании юниверса OLAP с помощью MSAS, SAP Business Warehouse (BW) или Essbase.

При создании юниверса OLAP можно определить следующие параметры:

Общие параметры OLAP	Описание
Создание технических имен в качестве сведений	Можно настроить создание приложением технического имени в качестве объекта сведений для измерений в юниверсе. При создании юниверса генерируются объекты сведений, указывающие на технические имена.
Параметры OLAP для SAP	Описание
Настройка агрегирования мер для делегирования	Можно задать настройку приложением функции агрегирования мер для делегированной базы данных.
Заменить префиксы L00, L01,	Префиксы уровня юниверса обозначают уровень объекта в иерархии. Уровень "L00" представляет собой верхний или корневой уровень, а "L01" является следующим нижестоящим уровнем. В Мастере нового юниверса можно заменить префиксы уровня юниверса OLAP на другой префикс. Нумерация уровней останется прежней, но префикс "L" может быть заменен на "Level", например. Введите собственный префикс в поле <i>Новый префикс</i> . Этот префикс будет присоединен ко всем уровням в юниверсе OLAP.
Переименовать уровень 00 в уровень "Все"	Этот параметр неактивен, если параметр <i>Уровень генерирования 00</i> имеет значение <i>Нет</i> . Можно переименовать верхний (корневой) уровень "L00" в уровень "Все" при следующем создании юниверса.
Уровень генерирования 00	Этот параметр применяется только к характеристикам SAP. Этот параметр можно отключить для характеристик и иерархий. Уровень 00 всегда создается для переменных иерархий. При создании или обновлении юниверса можно повторно сгенерировать номера уровней (L00, L01, L02...). Номера уровней добавляются к именам уровней (например, "Monthly Sales_L01"). Это полезно в случае отчетов Web Intelligence, где уровень "Все" используется для агрегирования результатов запроса. Это позволяет не создавать поле агрегирования в отчете Web Intelligence.

❗ Примечание

Если при создании юниверса был отключен параметр "Уровень генерирования 00", корневой уровень для иерархий не создается

9.3.2 Определение параметров OLAP для юниверса OLAP

Используйте параметры OLAP, чтобы определить способ создания конкретных метаданных юниверса из источника OLAP. Параметры OLAP можно выбрать на странице *OLAP* диалогового окна *Параметры* (► *Сервис* ► *Параметры* ► *OLAP* ►). Все содержимое источника OLAP извлекается и создается в юниверсе в зависимости от выбранных параметров. Можно выбрать следующие параметры создания юниверса OLAP:

Общие параметры OLAP	Описание
Создание технических имен в качестве сведений	Можно настроить создание приложением технического имени юниверса в качестве свойства. Тогда при создании юниверса создается объект, указывающий на техническое имя.
Параметры OLAP для SAP	Описание
Настройка агрегирования мер для делегирования	Можно задать создание приложением делегированных мер для мер, использующих функцию агрегирования. При создании юниверса любые меры, использующие функцию агрегирования, задаются для делегированной базы данных.
Заменить префиксы L00, L01,	Префиксы уровня юниверса обозначают уровень объекта в иерархии. Уровень "L00" представляет собой верхний или корневой уровень, а "L01" является следующим нижестоящим уровнем. В Мастере нового юниверса можно заменить префиксы уровня юниверса OLAP на другой префикс. Нумерация уровней останется прежней, но префикс "L" может быть заменен на "Level", например. Введите собственный префикс в поле <i>Новый префикс</i> . Этот префикс будет присоединен ко всем уровням в юниверсе OLAP.
Переименовать уровень 00 в уровень "Все"	Этот параметр неактивен, если параметр "Уровень генерирования" имеет значение "Нет". Можно переименовать верхний (корневой) уровень "L00" в уровень "ВСЕ" при следующем создании юниверса. Это полезно в случае отчетов Web Intelligence, где уровень "Все" используется для агрегирования результатов запроса. Это позволяет не создавать поле агрегирования в отчете Web Intelligence.
Уровень генерирования 00	<p>Этот параметр применяется только к характеристикам SAP. Этот параметр можно отключить только для характеристик. Уровень 00 всегда создается для иерархий и переменных иерархий.</p> <p>При создании или обновлении юниверса можно повторно сгенерировать номера уровней (L00, L01, L02...). Номера уровней добавляются к именам уровней (например, "Monthly Sales_L01")</p>

9.3.3 Определение объектов в юниверсах OLAP

Можно использовать редактор SQL для определения инструкции Select или утверждения Where для объекта, а также для вставки операторов и функций MDX для объектов юниверсов OLAP. Параметры и функции, доступные в редакторе SQL, зависят от соответствующей базы данных.

9.3.4 Функции средства создания юниверсов, поддерживаемые для юниверсов OLAP

Юниверсы OLAP создаются автоматически. После создания юниверса OLAP любой из его компонентов может быть изменен.

Следующие функции средства создания юниверсов поддерживаются для создаваемых юниверсов OLAP:

- Скрыть, дублировать, переименовать классы и объекты (измерение, сведения и показатель)
- Вставить новые классы и объекты (измерение, сведения и показатель)
- Изменить формат объекта
- Изменить тип данных объекта
- Установить первичный и внешний ключи
- Произвести синтаксический анализ объектов MDX измерения, сведений и показателя
- Проверить целостность юниверса
- Изменить иерархии
- Создать каскадный список значений
- Определить делегированный поиск списка значений, позволяющий пользователям ограничить загрузку списка значений во время выполнения запроса.
- Использовать значения по умолчанию для переменных
- Задать показателю помощью функции проекции, делегированной базой данных (интеллектуальные показатели)
- Обновить структуру юниверса

Кроме того, следующие функции доступны только для юниверсов OLAP:

- Создать вычисляемые показатели (только SAP BW и MSAS)
- Создать ранее определенные условия
- Установить дополнительные подсказки

Все объекты, основанные на юниверсах OLAP, создаются с поддержкой индекса. Если присутствуют дубликаты значений в иерархии для объекта, поддержка индекса устраняет несогласованность в списке значений. Например, сказано, что значение "Париж" встречается в иерархии дважды: один раз в родительском объекте "Франция", другой – в родительском объекте "Техас". При выборе пользователем значения "Париж" в объекте "Франция" будут возвращены только строки для Парижа во Франции.

Следующие функции средства создания юниверсов не поддерживаются для юниверсов OLAP:

- Невозможно установить защиту на уровне строк в юниверсе OLAP.
- Невозможно изменить список значений в юниверсе OLAP.
- Невозможно просмотреть и изменить схему отношений сущностей юниверса, так как для юниверсов OLAP такая схема не создается.

9.3.5 Функция проекции "Делегированная база данных"

Любая мера юниверса может обладать функцией проекции (*Sum*, *Min*, *Max*, *Count* и *Avg*). Функция проекции используется для локального агрегирования меры в приложении Web Intelligence, когда

количество отображаемых в отчете измерений меньше, чем количество измерений в наборе результатов запроса.

Неаддитивные меры, такие как отношение, среднее значение и вес могут быть отражены только на том же уровне агрегирования, что и набор результатов запроса. Поэтому функция проекции неаддитивных мер в юниверсе обычно устанавливается на значение *Нет*.

Функция проекции *Делегированная база данных* позволяет делегировать агрегирование неаддитивных мер на сервер баз данных. В приложении Web Intelligence они называются интеллектуальными мерами. На странице свойств объекта функция проекции интеллектуальной меры устанавливается на значение *Делегированная база данных*. Дополнительные сведения об использовании этих и других функций в Web Intelligence см. в разделе *Расчет значений с интеллектуальными мерами* в документе *Использование функций, формул и вычислений в Web Intelligence*.

❗ Примечание

Для юниверсов OLAP, основанных на источниках данных MSAS и Essbase, все меры юниверса создаются с функцией проекции, установленной по умолчанию на значение *Делегированная база данных*.

❗ Примечание

При использовании интеллектуальной меры, основанной на мере с поддержкой агрегирования, необходимо учесть следующие ограничения: настоятельно рекомендуем убедиться, что таблицы агрегирования, используемые при определении меры, содержат непротиворечивые данные (значения агрегирования соответствуют значениям объектов-атрибутов). В противном случае интеллектуальные меры могут выдать неточные данные. Например, если таблицы агрегирования по году и по дню используются для интеллектуальной меры, годовая таблица агрегирования будет соответствовать дневной таблице по завершенным годам, но по текущему году годовая таблица может быть пустой, в то время как дневная таблица будет содержать точные данные по дням. В этом случае отчет, использующий интеллектуальные меры на основе текущих таблиц года и дня, может выдать неточные результаты.

Пример: Интеллектуальная мера

В данном примере запрос содержит два измерения: "Страна" и "Регион" и три меры: "Сумма заказа", "Сумма доставки" и "% доставки".

Регион L01	Сумма доставки	Сумма заказа	% доставки
Reg1	497318880	497332680	99,997
Reg2	199463776	199466536	99,998
Reg3	198927552	198933072	99,997
Сумма:			299,992

Сумма меры "% доставки" неверна, поскольку представляет собой сумму данных столбца "% доставки".

Если в юниверсе функция проекции этой меры имеет значение *База данных делегирована*, то при обновлении отчета Web Intelligence соединяется с базой данных для расчета верного значения.

Регион LO1	Сумма доставки	Сумма заказа	% доставки
Reg1	497318880	497332680	99,997
Reg2	199463776	199466536	99,998
Reg3	198927552	198933072	99,997
		Сумма:	299,992
		Всего:	99,997

📘 Примечание

Некоторые функции, например функцию соотношения (Average), необходимо использовать с осторожностью. При расчете среднего значения столбца поведение этой функции может оказаться непредвиденным, если она была неправильно сконфигурирована.

Например, функция SQL `sum(Shop_facts.Margin)/sum(Shop.facts.Quantity_sold)` может выдать непредвиденные результаты. Если она сконфигурирована неправильно, то она произведет расчет среднего значения каждой ячейки и выдаст сумму средних значений. Чтобы исправить такое поведение, необходимо задать параметры функции следующим образом.

1. Перейдите к параметру функции *Изменить свойства*.
2. Для параметра *Выберите способ проектирования меры при агрегировании* в раскрывающемся списке "Функция" выберите функцию *Db delegated*.
3. Сохраните изменения.

Связанные сведения

[Установка проекции агрегирования для меры \[страница 453\]](#)

9.3.6 Настройка делегированных мер для юниверсов OLAP

Можно задать создание приложением делегированных мер для мер, использующих функцию агрегирования. При создании юниверса любые меры, использующие функцию агрегирования, задаются для делегированной базы данных...

Связанные сведения

[Замена префиксов уровня юниверса OLAP \[страница 471\]](#)

[Регенерация уровня ОО для юниверсов OLAP \[страница 470\]](#)

[Переименование уровня "LOO" в уровень "BCE" \[страница 470\]](#)

9.3.7 Установка проекции агрегирования для меры

Чтобы создать меру, необходимо указать путь, с помощью которого функция агрегирования будет спроецирована в отчете.

Возвращенные значения меры объекта агрегируются на двух уровнях в процессе запроса.

- Уровень запроса. Данные агрегируются с помощью унаследованной инструкции SELECT.
- Из микрокуба на уровень блока. Если данные проецируются из микрокуба в блок отчета. Функция проекции мер позволяет локальное агрегирование в микрокубе.

❗ Примечание

Микрокуб – это концептуальный путь, чтобы представить данные, возвращенные с помощью запроса, перед проецированием в отчет. Он представляет возвращенные значения, хранящиеся в памяти, с помощью продукта отчета Business Objects. Уровень блока – это отчет на основе двух измерений, который пользователь создает с помощью возвращенных данных. Пользователь может использовать все или только некоторые данные, хранящиеся в микрокубе, чтобы создать отчет. Пользователь может также выполнить функции агрегирования на основе возвращенных значений в микрокубе (локальное агрегирование), чтобы создать новые значения в отчете.

Два уровня агрегирования соответствуют процессу запроса, как показано ниже.

- Пользователь создает запрос в Web Intelligence.
- Web Intelligence выводит SQL из запроса и отправляет инструкцию SELECT в целевую базу данных.
- Данные возвращаются в микрокуб. Это первый уровень агрегирования.
- Микрокуб проецирует агрегированные данные в отчет. На панели запросов данные разделяются для агрегирования на низших уровнях. Это второй уровень агрегирования.

После выполнения запроса набор результатов инструкции Select хранится в микрокубе, а все данные микрокуба проецируются в блок. Так как данные проецируются из низшего уровня микрокуба, агрегирование проекции не производится.

Однако когда панель запросов используется для проецирования лишь части данных из микрокуба, требуется агрегирование, чтобы отобразить значения мер на более высоком уровне.

Если в предыдущем примере не спроецировать данные за год в блок, три строки, связанные с годом, необходимо сократить до одной, чтобы отобразить общий доход от продаж для курорта. В данном случае используется агрегирование суммы.

Агрегирование проекции указывается на странице [Свойства](#) листа [Изменить свойства](#) меры (правой кнопкой щелкните "Объект" > "Свойства объекта" > "Свойства").

Агрегирование проекции отличается от агрегирования оператора SELECT.

9.3.8 Вычисляемые показатели в юниверсах OLAP

Для сужения запросов можно создавать в юниверсе вычисляемые показатели. Вычисляемые показатели в юниверсах OLAP аналогичны определениям в объекте показателя в юниверсах типа, отличного от OLAP, за исключением того, что ограничение определяется при помощи заключенных в теги XML функций MDX вместо использования языка SQL.

Вычисляемые показатели доступны для следующих источников данных OLAP:

SAP Business Warehouse (BW)

MSAS 2000 и 2005

Вычисляемые показатели могут использоваться в фильтрах или в утверждении WHERE.

Синтаксис выражений с вычисляемыми показателями

Синтаксис для вычисляемого показателя состоит из выражений, заключенных в теги <EXPRESSION></EXPRESSION>.

Функции средства создания юниверсов можно использовать в выражениях с вычисляемыми показателями, таких как:

@Select

@Prompt

@Variable

@Where

❗ Примечание

Выражения с вычисляемыми показателями не могут включать в себя функцию @Aggregate_Aware. Функция проверки целостности проверяет синтаксис XML и все функции @Function, перечисленные выше, включая вставленные в MDX-инструкции. Однако синтаксический разбор MDX-инструкций не выполняется.

Допускается использование констант в выражениях, например, "10" или "ABC".

Вычисляемые показатели могут обращаться к любым метаданным OLAP:

показатели

измерения

уровни измерений

выражения MDX

Рекомендации по построению выражений с вычисляемыми показателями

Используйте функцию @SELECT (имя показателя) вместо определения показателя, по следующим причинам:

функция @SELECT вычисляется во время запроса.

Вычисляемые показатели могут обращаться к другим вычисляемым показателям, если находятся в выражении с функцией @SELECT.

Проверяется применимость объектов внутри функции @SELECT.

Сгенерируйте и настройте индексную таблицу для определения каждого объекта.

Используйте ссылку на объект или деталь, определение которого ссылается на техническое имя или уникальное имя уровня или атрибута.

Пример: Выражение с вычисляемым показателем

```
<EXPRESSION>@Select(Key Figures\Order Amount)*@Select(Key Figures\Order  
Quantity)</EXPRESSION>
```

Связанные сведения

[Создание вычисляемой меры в юниверсе OLAP \[страница 455\]](#)

9.3.8.1 Создание вычисляемой меры в юниверсе OLAP

Для создания вычисляемой меры в юниверсе OLAP выполните следующие действия.

1. Откройте юниверс OLAP в средстве создания юниверсов.
2. Вставьте новый объект меры в юниверс.
3. В операторе *Where*: введите или вставьте определение объекта как выражение XML/MDX.
4. Нажмите *Разбор* для анализа определения объекта и исправления ошибок.
5. Нажмите *OK*, чтобы сохранить определение объекта.
6. Выберите **Инструменты** > *Проверить целостность* .
В рамках проверки целостности проверяются синтаксис XML и любые функции @FUNCTION средства создания юниверсов.

Связанные сведения

[Вычисляемые показатели в юниверсах OLAP \[страница 454\]](#)

9.3.9 Функции MDX для запросов к кубу

Для определения запросов к кубу используется редактор MDX.

При добавлении нового объекта или предварительно определенного фильтра к юниверсу OLAP доступен список поддерживаемых выражений MDX, соответствующих конкретному соединению с источником данных.

Библиотека доступных объектов хранится в файле соединений .prm. При открытии панели изменения свойств для объекта и открытия панели редактирования SELECT для запроса доступные выражения отображаются на панели "Функции". Чтобы вставить выражение в инструкцию SELECT или WHERE, щелкните место в инструкции, где необходимо вставить выражение, и дважды щелкните соответствующее выражение.

Словарь MDX юниверсов OLAP – список функций (файл PRM)

При добавлении нового объекта или предварительно определенного фильтра к юниверсу OLAP в редакторах объекта и фильтра доступен явный список операторов и функций MDX (в основном функций элементов) для соответствующего соединения OLAP (SAP или MSAS), которые можно использовать в выражении. Описание настройки соединения для SAP или MySQL (sap.prm, sqlsrv_as.prm) см. в Руководстве по доступу к данным. Доступные функции и операторы зависят от типа соединения для юниверса. Для каждого соединения в файле PRM приведен список функций. В этом списке приведены не все поддерживаемые функции, а только наиболее часто используемые функции.

Для запросов доступны следующие операторы MDX:

- Равно
- Не равно
- В списке
- Не в списке
- Больше
- Больше или равно
- Меньше
- Меньше или равно
- В промежутке
- Не в промежутке
- Подобно
- Не подобно

В приведенном ниже списке показаны примеры некоторых функций папки MDX, доступных при редактировании условия. Доступные функции и операторы зависят от соответствующей базы данных.

- Функции множества (ADDCALCULATEDMEMBERS, ALLMEMBERS ...)
- Статистические/числовые функции (AGGREGATE, AVG ...)
- Функции навигации/элементов (ANCESTOR, ASCENDANTS...)
- Функции метаданных (AXIS, HIERARCHY...)

9.3.10 XML-синтаксис для инструкций фильтра и WHERE

В этом разделе описывается XML-синтаксис для определения утверждений WHERE или инструкций фильтра в юниверсе OLAP. Следует вручную добавить теги FILTER или FILTER EXPRESSION, а затем ввести выражение между тегами вручную или с помощью редактора MDX средства создания юниверсов.

- Если используется одно определение объекта, воспользуйтесь синтаксисом `<FILTER= "определение_объекта">`. Введите определение объекта в двойных кавычках.
- Если используется сложное выражение MDX, содержащее один или несколько объектов, воспользуйтесь синтаксисом `<FILTER EXPRESSION= "сложное_выражение_MDX">`. Введите выражение в двойных кавычках.

Ниже приведен синтаксис фильтра для одного объекта:

```
<FILTER = "определение_объекта"><CONDITION OPERATORCONDITION="оператор"><CONSTANT  
VALUE=" значение " /></CONDITION></FILTER>
```

Где:

- выражение_MDX – определение одного объекта, заключенное в двойные кавычки.
- CONSTANT VALUE – либо CONSTANT CAPTION, либо CONSTANT TECH_NAME.
- оператор – оператор выражения фильтра (equals, inlist и т. д.). Если используется оператор Inlist, необходимо вставить элемент CONSTANT CAPTION или CONSTANT TECH_NAME для каждой строки в списке.
- значение – значение определенного фильтра (если используется элемент CONSTANT CAPTION) или идентификатор объекта (если используется элемент CONSTANT TECH_NAME).

Ниже приведен синтаксис для одного объекта фильтра при использовании оператора InList для списка, содержащего три страны:

```
<FILTER= "определение_объекта"><CONDITION OPERATORCONDITION="InList"><CONSTANT  
CAPTION="Великобритания" /><CONSTANT CAPTION="Франция" /><CONSTANT  
CAPTION="Германия" /></CONDITION></FILTER>
```

Ниже приведен синтаксис для сложного выражения фильтра с использованием элемента TECH_NAME для отфильтрованного значения:

```
<FILTER EXPRESSION="сложное_выражение_MDX"><CONDITION  
OPERATORCONDITION="Equal"><CONSTANT TECH_NAME="1" /></CONDITION></FILTER>
```

Пример: Фильтр с вычисляемым элементом в выражении фильтра

```
<FILTER EXPRESSION="IIF([OCALYEAR].CurrentMember > "2000", 1,0)"><CONDITION  
OPERATORCONDITION="Equal"><CONSTANT CAPTION="1" /></CONDITION></FILTER>
```

9.3.11 Предопределенные условия в юниверсах OLAP

Предопределенные условия в юниверсах OLAP аналогичны условиям в других юниверсах, за исключением того что утверждение WHERE определяется с использованием XML, а не SQL. Фильтры можно объявлять вручную или с помощью редактора предварительно определенных фильтров.

9.3.11.1 Синтаксис XML для параметров предварительно определенных фильтров

Синтаксис для предопределенных условий

Одно предопределенное условие может включать несколько фильтров, объединенных с помощью операторов AND и OR. По умолчанию все фильтры скомбинированы с операторами AND. Для включения фильтров с OR необходимо использовать теги операторов AND и OR.

В предопределенном определении фильтра можно использовать функции @Select, @Prompt и @Variable.

В предварительно определенных фильтрах может содержаться один или несколько подсказок. Подсказки могут быть необходимыми или дополнительными.

Пример: Использование тегов AND и OR для предопределенных условий

```
<OPERATOR VALUE="AND">
  <FILTER "[Level Object definition]">
    <CONDITION OPERATORCONDITION="Operator">
      <CONSTANT Level Attribute="Value"/>
    </CONDITION>
  </FILTER>
<OPERATOR VALUE="OR">
  <FILTER "[Level Object definition]">
    <CONDITION OPERATORCONDITION="Operator">
      <CONSTANT Level Attribute="Value"/>
    </CONDITION>
  </FILTER>
  <FILTER "[Level Object definition]">
    <CONDITION OPERATORCONDITION="Operator">
      <CONSTANT Level Attribute="Value"/>
    </CONDITION>
  </FILTER>
</OPERATOR>
</OPERATOR>
```

9.3.11.2 Создание предопределенных условий в универсе OLAP вручную

Для создания предопределенных условий выполните следующие шаги.

1. В средстве создания универсов откройте универс OLAP и установите переключатель условий в нижней части панели "Юниверс".
Отобразится представление условий панели "Юниверс". В нем содержится древовидное представление классов универса.
2. Щелкните класс правой кнопкой мыши и выберите [Условие...](#) в контекстном меню.
3. В поле [Where:](#) отредактируйте фильтр шаблона XML.

Фильтр шаблона имеет следующий формат.

```
<FILTER "[Level Object definition]">
  <CONDITION OPERATORCONDITION="Operator">
    <CONSTANT Level Attribute="Value"/>
  </CONDITION>
</FILTER>
```

Замените элементы в шаблоне, как показано ниже:

Элемент шаблона:	Возможные значения:
Определение объекта уровня	Введите уровень измерения или меру, включенную в фильтр. Введите определение объекта, но не имя объекта.
Оператор	Введите один из следующих вариантов: <ul style="list-style-type: none">• Равно• Не равно• Больше• Меньше• Больше или равно• Меньше или равно• В промежутке• Не в промежутке• InList• Не в списке• Подобно• Не подобно
Атрибут уровня	Введите один из следующих вариантов: <ul style="list-style-type: none">• NAME• CAPTION• TECH_NAME• DESCRIPTION

Элемент шаблона:	Возможные значения:
Значение	Введите значение или подсказку. Определите одно значение через тег CONSTANT.

Пример отредактированного предопределенного условия.

```
<FILTER KEY="[OD_DIV].[LEVEL01]">
  <CONDITION OPERATORCONDITION="InList">
    <CONSTANT CAPTION="Internal"/>
    <CONSTANT CAPTION="Service"/>
  </CONDITION>
</FILTER>
```

4. Нажмите кнопку [Разбор](#) для анализа синтаксиса и исправления ошибок.
5. Нажмите кнопку [OK](#), чтобы сохранить условие.

Связанные сведения

[Предопределенные условия в юниверсах OLAP \[страница 458\]](#)

[Дополнительные подсказки в юниверсах OLAP \[страница 463\]](#)

9.3.11.3 О редакторе предварительно определенных фильтров

Редактор [предварительно определенных фильтров](#) предназначен для редактирования предварительно определенных фильтров в юниверсах OLAP. Он применяется для выбора объектов, операторов, списков значений, подсказок, функций и других необязательных элементов, которые можно использовать для определения фильтра для юниверса OLAP.

На панели свойств условия фильтра можно вручную ввести выражение фильтра или нажать **>>** для открытия редактора [предварительно определенных фильтров](#). Когда редактор открыт, в выражение фильтра можно вставить @Prompt: щелкните правой кнопкой мыши соответствующее место в выражении фильтра и выберите [Новый @Prompt](#) в контекстном меню. Редактор предварительно определенных фильтров вставляет выражение фильтра в определение объекта/запроса.

Пример: Ограничение в измерении "Страна" на уровне страны для ограничения страны Канадой

```
<FILTER KEY="[Клиент].[Страна].[Страна]"> <CONDITION OPERATORCONDITION="Равно"> <CONSTANT CAPTION="Канада" /> </CONDITION> </FILTER>
```


Связанные сведения

[Параметры для редактора предварительно определенных фильтров \[страница 461\]](#)

[Редактирование предварительно определенного фильтра с помощью редактора предварительно определенных фильтров \[страница 462\]](#)

[Функции MDX для запросов к кубу \[страница 456\]](#)

9.3.11.4 Параметры для редактора предварительно определенных фильтров

Редактор [предварительно определенных фильтров](#) позволяет легко определить фильтр юниверса для юниверса OLAP. Можно определить следующие параметры:

Действие	Описание
Выбрать оператор	Выберите оператор из имеющегося списка. По умолчанию = равно
Применить фильтрацию по	Фильтрация либо по существующему объекту юниверса, либо по свободному определению (например, [Меры].[Сумма Интернет-продаж]). По умолчанию = Объект юниверса .
Выбрать список значений	Выберите список объектов в текущем юниверсе, если фильтрация основана на существующем объекте. Выбор по умолчанию = корневой класс (Root) в списке объектов.
Значения для сравнения	Определите значения для сравнения с объектом/выражением. В зависимости от выбранного оператора можно ввести один или два набора значений. Значения могут быть статическими или на основе подсказки. По умолчанию = статические значения .
Добавить подсказку	Измените подсказку вручную или используйте редактор @Prompt . Нажмите >>, чтобы открыть редактор @Prompt .
Установить поддержку индексирования	Включает функцию поддержки индексирования. Для правильной работы необходимо объявить первичные ключи. Если в средстве создания юниверсов настроена поддержка индексирования, для ускорения извлечения данных и создания более эффективных фильтров SQL в средстве создания юниверсов используются столбцы и первичного, и внешнего ключа. По умолчанию = Не выбран.
Использовать вычисленное выражение	Если выбран, выражение фильтра помещается в теги <EXPRESSION> </EXPRESSION>. По умолчанию = Не выбран.
Дополнительно	Настройка текущего выражения фильтра как необязательного. Применяется только к текущему выражению фильтра в редакторе фильтров, а не ко

Действие	Описание
	всему объекту предварительно определенного условия. По умолчанию = Не выбран.

📘 Примечание

В Web Intelligence нельзя использовать "необязательные" теги для предопределенных фильтров. Если эти теги используются, они рассматриваются как обязательная часть запроса, поэтому запрос не выполняется.

Связанные сведения

[Редактирование предварительно определенного фильтра с помощью редактора предварительно определенных фильтров \[страница 462\]](#)

9.3.11.5 Редактирование предварительно определенного фильтра с помощью редактора предварительно определенных фильтров

Выполняется редактирование фильтра в юниверсе OLAP.

Редактор [предварительно определенных фильтров](#) обновляется при вводе или выборе значений. Можно щелкнуть выражение фильтра правой кнопкой мыши, чтобы вставить в него выражение @Prompt. Если щелкнуть правой кнопкой мыши и выбрать [Новый @Prompt](#), откроется [редактор подсказок](#).

1. В панели [Свойства](#) панели условий (фильтров) щелкните кнопку [>>](#).
Появится редактор [предварительно определенных фильтров](#).
2. Чтобы задать фильтр для объекта юниверса, нажмите [Объект юниверса](#) и выберите объект на панели [Доступные объекты](#). Чтобы задать предварительно определенный фильтр для собственного выражения, выберите [Свободное определение](#) и введите выражение на панели [Доступные объекты](#).
3. Выберите оператор из списка [Операторы](#). Множественные значения (правый операнд) разрешены только для операторов "В списке" и "Не в списке".
4. Выберите [Статическое значение](#), чтобы определить одно или несколько фиксированных значений, либо [Подсказка](#), чтобы вставить выражение подсказки.
Если был выбран параметр [Подсказка](#), активируется кнопка [Редактировать](#). Нажмите [Редактировать](#), чтобы открыть редактор [@Prompt](#) и определить необходимое выражение подсказки.
5. Нажмите кнопку [ОК](#), чтобы подтвердить определение фильтра.
Синтаксический анализатор проверяет синтаксис на предмет ошибок, в том числе выполняет проверку целостности. При обнаружении ошибок выводится предупреждение с сообщением об ошибке. Если ошибок не обнаружено, к юниверсу добавляется новый объект условия с определением фильтра.

Связанные сведения

[Параметры для редактора предварительно определенных фильтров \[страница 461\]](#)

[О редакторе предварительно определенных фильтров \[страница 460\]](#)

9.3.12 Дополнительные подсказки в юниверсах OLAP

Юниверсы, генерируемые из источников данных OLAP, поддерживают дополнительные подсказки.

Фильтры с дополнительными условиями автоматически генерируются в юниверсе для дополнительных переменных SAP BW.

Чтобы в предварительно определенных условиях или утверждении WHERE объекта сделать подсказку дополнительной, встройте выражение фильтра XML между двумя метками XML: <OPTIONAL> и </OPTIONAL>.

Пример: Дополнительная подсказка в заранее определенном условии

```
<OPTIONAL>
  <FILTER KEY="[Products].[Family]" >
    <CONDITION OPERATORCONDITION="InList" >
      <CONSTANT CAPTION="@prompt('Enter value(s) for Product
family:', 'A', 'Products\Family', Multi, primary_key, persistent)"/>
    </CONDITION>
  </FILTER>
</OPTIONAL>
```

Связанные сведения

[Создание предопределенных условий в юниверсе OLAP вручную \[страница 459\]](#)

9.3.13 Повышение производительности определенных запросов для юниверсов SAP BW

Для запросов к юниверсам SAP BW, которые включают в себя только ключ и объекты-атрибуты среднего имени для измерения, с целью увеличения производительности запросов можно изменять сгенерированную синтаксическую конструкцию объектов.

Чтобы изменить синтаксис, выполните следующие действия.

1. Откройте юниверс в средстве создания юниверсов.

2. Нажмите два раза ключевой объект-атрибут, который необходимо изменить.
3. В выбранном текстовом окне во вкладке *Определение* диалогового окна *Изменить свойства* измените синтаксис на атрибут имени характеристики SAP.
Например, для объекта *Ключ пользователя L01* измените выбор заданного синтаксиса.

```
[ Z_CUSTOM ] . [ LEVEL01 ] . [ [ 2Z_CUSTOM ] ] . [ Value ]
```

чтобы обратиться к атрибуту имени:

```
[ Z_CUSTOM ] . [ LEVEL01 ] . [ NAME ]
```

4. Нажмите кнопку *OK*, чтобы сохранить изменения.
5. Выполните те же действия для объекта имени. Измените синтаксис, чтобы обратиться к атрибуту описания характеристики SAP.
Например, для объекта *Имя носителя данных пользователя L01* измените выбор заданного синтаксиса.

```
[ Z_CUSTOM ] . [ LEVEL01 ] . [ [ 5Z_CUSTOM ] ] . [ Value ]
```

чтобы обратиться к атрибуту описания, необходимо:

```
[ Z_CUSTOM ] . [ LEVEL01 ] . [ DESCRIPTION ]
```

9.4 Управление жизненным циклом юниверса OLAP

9.4.1 Управление жизненным циклом юниверса OLAP

📌 Примечание

При открытии юниверса, созданного с помощью более ранней версии Universe Designer, чем XIR3.1 SP2, необходимо обновить и сохранить юниверс перед внесением изменений в него или в источник OLAP.

Юниверсы OLAP автоматически создаются из источника данных OLAP (например, из запроса SAP BEx или куба MSAS 2005). Средство создания юниверсов позволяет создавать и изменять объекты существующего юниверса OLAP.

Мастер *обновления юниверса OLAP* позволяет автоматически обновлять структуру юниверса OLAP с изменениями, сделанными в источнике данных OLAP. Мастер сравнивает юниверс с обновленным источником данных. Этот мастер отличает генерируемые объекты от добавленных или измененных вручную объектов, позволяя сохранять изменения, сделанные вручную в средстве создания юниверсов. Мастер не обновляет объекты, добавленные в средство создания юниверсов вручную.

Обнаружение и обновление объектов зависит от элементов и источника данных, как показано в таблице ниже:

Что может обнаружить мастер	Новые элементы, которые могут быть обнаружены в	Измененные элементы, которые могут быть обнаружены в	Удаленные элементы, которые могут быть обнаружены в
Измерения	Все источники данных	Все источники данных	Все источники данных
Иерархии	Только SAP BW и MSAS	Все источники данных	Все источники данных
Уровни	Все источники данных	Все источники данных	Все источники данных
Свойства	Только MSAS	Только MSAS	Только MSAS
Меры	Все источники данных	Все источники данных	Все источники данных
Переменные SAP BW	Только SAP BW	Только SAP BW	Только SAP BW
Подклассы	Все источники данных	Все источники данных	Все источники данных

📌 Примечание

При обновлении юниверса, созданного с помощью более ранней версии Universe Designer, чем XIR3.1 SP2, при изменении имени измерения в кубе SAP обновление этого измерения не выполняется: создается дубликат этого измерения в юниверсе. Необходимо вручную обновить классы в юниверсе.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Обзор: взаимосвязь между состоянием объекта юниверса и состоянием объекта OLAP \[страница 465\]](#)

[Управление измерениями при обновлении юниверсов OLAP \[страница 472\]](#)

[Управление иерархиями и характеристиками при обновлении юниверсов OLAP \[страница 476\]](#)

[Управление уровнями при обновлении юниверсов OLAP \[страница 483\]](#)

[Управление переменными SAP при обновлении юниверсов OLAP \[страница 486\]](#)

[Управление контрольными цифрами и мерами при обновлении юниверсов OLAP \[страница 490\]](#)

[Управление контрольными датами SAP при обновлении юниверсов OLAP \[страница 493\]](#)

9.4.2 Обзор: взаимосвязь между состоянием объекта юниверса и состоянием объекта OLAP

В приведенной ниже таблице дан краткий обзор взаимосвязи между статусом объекта OLAP в SAP и статусом объекта юниверса. Для большинства действий существуют специальные примечания. Дополнительные сведения см. в более подробных разделах этой главы.

		Состояние объекта юниверса				
Метаданные OLAP		Не изменено	Обновлено	Удалено	Перемещено	Скрыто
Измерение		Эквивалент юниверса = класс				
Статус	Не изменено	NoC	Upd	NoC	NoC	NoC
	Обновлено	Upd	Upd	NoC	Upd	Upd
	Удалено	Del/Ob	Del/Ob	NoC	Del/Ob	NoC
	Перемещено	Переместить	NoC	NoC	NoC	Переместить
	Создана характеристика	CreS	CreS	Н/Д	CreS	CreS
	Создан	Cre	Cre	Н/Д	Cre	Cre
Иерархия или характеристика		Эквивалент юниверса = подкласс				
Статус	Не изменено	NoC	Upd	NoC	NoC	NoC
	Обновлено	Upd	Upd	NoC	Upd	Upd
	Изменено	UpdMDX	UpdMDX	NoC	UpdMDX	UpdMDX
	Атрибут отображения	Cre	Cre	Cre	Cre	Cre
	Атрибут навигации	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Удалено	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Перемещено	Переместить	Переместить	NoC	Переместить	Переместить
	Создать	Cre	Cre	Cre	Cre	Cre
Уровень		Эквивалент юниверса = уровень				
Статус	Не изменено	NoC	NoC	NoC	NoC	NoC
	Обновлено	Upd	Upd	NoC	Upd	Upd
	Удалено	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Перемещено	Переместить	Переместить	NoC	Переместить	Переместить
	Создать	Cre	Cre	Cre	Cre	Cre
Переменная		Эквивалент юниверса = фильтр				
Статус	Не изменено	NoC	NoC	NoC	NoC	NoC
	Обновлено	Upd	Upd	Cre	Upd	Upd
	Удалено	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Создать	Cre	Cre	Cre	Cre	Cre
Контрольная цифра		Эквивалент юниверса = мера				
Статус	Не изменено	NoC	NoC	NoC	NoC	NoC

Метаданные OLAP		Состояние объекта универса				
		Не изменено	Обновлено	Удалено	Перемещено	Скрыто
	Обновлено	Upd	Upd	NoC	Upd	Upd
	Удалено	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Перемещено	Переместить	Переместить	NoC	Переместить	Переместить
	Создать	Cre	Cre	Cre	Cre	Cre
контрольная дата;		Эквивалент универса = параметр				
Статус	Не изменено	NoC	Н/Д	Cre	Н/Д	Н/Д
	Удалено	Del	Н/Д	Н/Д	Н/Д	Н/Д
	Создать	Cre	Н/Д	Cre	Н/Д	Н/Д

Условные обозначения:

- *:Изменено одно из свойств объекта (имя, описание...).
- Cre: создание эквивалентного объекта
- CreS: создание эквивалентного объекта подкласса
- Del/Ob: удален или устарел (устаревшие объекты скрываются, их имена предваряются символами ##)
- Переместить: перемещение объекта
- Н/П: не применяется
- NoC: не изменен
- Upd: обновлен
- UpdMDX: обновление определения MDX

Связанные сведения

[Обновление универса OLAP \[страница 467\]](#)

[Управление измерениями при обновлении универсов OLAP \[страница 472\]](#)

[Управление иерархиями и характеристиками при обновлении универсов OLAP \[страница 476\]](#)

[Управление уровнями при обновлении универсов OLAP \[страница 483\]](#)

[Управление переменными SAP при обновлении универсов OLAP \[страница 486\]](#)

[Управление контрольными цифрами и мерами при обновлении универсов OLAP \[страница 490\]](#)

[Управление контрольными датами SAP при обновлении универсов OLAP \[страница 493\]](#)

9.4.3 Обновление универса OLAP

Чтобы обновить структуру универса OLAP, выполните следующие действия.

- Откройте в средстве создания юниверсов юниверс, который нужно обновить.
- Нажмите ► [Просмотр](#) ► [Обновить структуру](#) ►.
Отобразится мастер обновления юниверса OLAP.
- Нажмите [Начать](#).

Открывается страница обновленных определений метаданных.

Выберите "Сохранить параметры", если необходимо сохранить изменения для объектов в юниверсе, сделанные вручную. Все сохраненные параметры выбираются по умолчанию. Можно выбрать сохранение следующих атрибутов:

Действие	Описание
Сохранить название фирмы	Имена классов, измерений, мер, сведений и условий.
Сохранить тип объектов	Если в юниверсе изменен объект (например, сведения заменены измерением, при обновлении не будет снова вводиться тип начального объекта). Это относится к измерениям, мерам и сведениям.
Сохранить описание объектов	Если выбран этот параметр, при обновлении описания в источнике OLAP юниверс не обновляется согласно этим данным.
Сохранить тип данных объектов	Символьные, числовые объекты, дата и длинная строка.
Сохранить варианты списков значений для объектов	Можно сохранить те параметры, которые были установлены изначально: <ul style="list-style-type: none"> • Связать со списком значений • Автоматическое обновление перед использованием • Показ иерархии • &Экспорт с юниверсом • Делегировать поиск
Сохранить дополнительные параметры объектов	Доступны следующие параметры: Уровень защиты доступа Объект можно использовать в <ul style="list-style-type: none"> • Использовать в результате • Использовать в условии • Использовать в сортировке
Удалить неупотребляемые объекты	Элементы, которые больше не содержатся в источнике данных, будут удалены из юниверса.
Скрыть неупотребляемые объекты	Элементы, которые больше не содержатся в кубе, будут скрыты в юниверсе, а также перед ними будет добавлено /##/.

- Выберите необходимые параметры и нажмите кнопку [Далее](#).

Страница с [Результатами управления изменениями](#) отображается, показывая добавленные, удаленные и скрытые объекты. Скрытые объекты перемещаются в разделяющие классы в юниверсе и отображаются курсивом с предшествующими символами /##/.

- На панели параметров добавляемых метаданных задайте параметры для обработки добавляемых метаданных

Общие параметры OLAP	Описание
Создание технических имен в качестве сведений	Можно настроить создание приложением технического имени юниверса в качестве свойства. Тогда при создании юниверса создается объект, указывающий на техническое имя.
Повторно сгенерированы все объекты, удаленные вручную	Все объекты юниверса, удаленные вручную, будут создаваться повторно.

Параметры OLAP для SAP	Описание
Настройка агрегирования мер для делегирования	Можно задать настройку приложением функции агрегирования мер для делегированной базы данных.
Заменить префиксы L00, L01,	Префиксы уровня юниверса обозначают уровень объекта в иерархии. Уровень "L00" представляет собой верхний или корневой уровень, а "L01" является следующим нижестоящим уровнем. В Мастере нового юниверса можно заменить префиксы уровня юниверса OLAP на другой префикс. Нумерация уровней останется прежней, но префикс "L" может быть заменен на "Level", например. Введите собственный префикс в поле Новый префикс . Этот префикс будет присоединен ко всем уровням в юниверсе OLAP.
Переименовать уровень 00 в уровень "Все"	Этот параметр неактивен, если параметр Уровень генерирования 00 имеет значение <i>Нет</i> . Можно переименовать верхний (корневой) уровень "L00" в уровень "Все" при следующем создании юниверса.
Уровень генерирования 00	Этот параметр применяется только к характеристикам SAP. Этот параметр можно отключить только для характеристик. Уровень 00 всегда создается для иерархий и переменных иерархий. При создании или обновлении юниверса можно повторно сгенерировать номера уровней (L00, L01, L02...). Номера уровней добавляются к именам уровней (например, "Monthly Sales_L01"). Это полезно в случае отчетов Web Intelligence, где уровень "Все" используется для агрегирования результатов запроса. Это позволяет не создавать поле агрегирования в отчете Web Intelligence.

- На странице результатов управления изменениями выберите:

Параметр	Описание
OK	Если результаты не удовлетворительны, можно нажать OK , затем закрыть юниверс без сохранения или экспорта.
Экспорт	Если изменения удовлетворительны, нажмите Экспорт , чтобы сохранить и экспортировать обновленный юниверс на сервер центрального управления.

Параметр	Описание
Проверка целостности	Нажмите Проверить целостность , чтобы произвести проверку целостности. Происходит проверка структуры, анализ объектов, объединений, условий и свойств связи. Когда проверка закончена, отображается страница с результатами проверки целостности . С этой страницы можно распечатать результаты проверки.

Если отображаются не все ожидаемые изменения юниверса, остановите процесс, а затем перезапустите средство создания юниверсов, прежде чем повторять попытку обновления. Это позволит установить соединение с источником данных заново и очистить кэш.

Связанные сведения

[Синхронизация юниверса и куба OLAP \[страница 471\]](#)

[Управление жизненным циклом юниверса OLAP \[страница 464\]](#)

9.4.4 Регенерация уровня 00 для юниверсов OLAP

При создании или обновлении юниверса можно повторно сгенерировать номера уровней (L00, L01, L02...). Номера уровней добавляются к именам уровней (например, "Monthly Sales_L01")

Связанные сведения

[Настройка делегированных мер для юниверсов OLAP \[страница 452\]](#)

[Замена префиксов уровня юниверса OLAP \[страница 471\]](#)

[Переименование уровня "L00" в уровень "BCE" \[страница 470\]](#)

9.4.5 Переименование уровня "L00" в уровень "BCE"

Можно переименовать верхний (корневой) уровень "L00" в уровень "BCE" при следующем создании юниверса. Это полезно в случае отчетов SAP BusinessObjects Web Intelligence, где уровень "BCE" используется для агрегирования результатов для запроса. Это позволяет не создавать поле агрегирования в отчете Web Intelligence.

Связанные сведения

[Настройка делегированных мер для юниверсов OLAP \[страница 452\]](#)

[Замена префиксов уровня юниверса OLAP \[страница 471\]](#)

9.4.6 Замена префиксов уровня юниверса OLAP

Префиксы уровня юниверса обозначают уровень объекта в иерархии. Уровень "L00" представляет собой верхний или корневой уровень, а "L01" является следующим нижестоящим уровнем. В [Мастере нового юниверса](#) можно заменить префиксы уровня юниверса OLAP на другой префикс. Нумерация уровней останется прежней, но префикс "L" может быть заменен на "Level", например. Введите собственный префикс в поле [Новый префикс](#). Этот префикс будет присоединен ко всем уровням в юниверсе OLAP.

Связанные сведения

[Настройка делегированных мер для юниверсов OLAP \[страница 452\]](#)

[Регенерация уровня 00 для юниверсов OLAP \[страница 470\]](#)

[Переименование уровня "L00" в уровень "ВСЕ" \[страница 470\]](#)

9.4.7 Синхронизация юниверса и куба OLAP

При обновлении юниверса объекты в юниверсе сравниваются с объектами в кубе OLAP. Сравнение обеспечивает отсутствие обратного влияния изменений в кубе на юниверс. Это означает, что любые объекты, используемые (или даже удаленные) в юниверсе, должны быть всегда доступны. Все новые объекты в кубе OLAP становятся доступными для юниверса. Чтобы увидеть, как изменения влияют на различные объекты, см. ссылки ниже.

При обновлении свойств объекта в юниверсе обновляется только определенная часть свойств; другие свойства не должны изменяться. Процесс проиллюстрирован следующей таблицей.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Управление измерениями при обновлении юниверсов OLAP \[страница 472\]](#)

[Управление иерархиями и характеристиками при обновлении юниверсов OLAP \[страница 476\]](#)

[Управление уровнями при обновлении юниверсов OLAP \[страница 483\]](#)

[Управление переменными SAP при обновлении юниверсов OLAP \[страница 486\]](#)

[Управление контрольными цифрами и мерами при обновлении юниверсов OLAP \[страница 490\]](#)

[Управление контрольными датами SAP при обновлении юниверсов OLAP \[страница 493\]](#)

9.4.8 Управление измерениями при обновлении юниверсов OLAP

Это относится к источникам данных SAP, MSAS и Essbase. Класс юниверса соответствует измерению OLAP. Управление объектами юниверса в соответствии с объектами OLAP зависит от типа изменения. См. в перечисленных ниже тематических разделах, как на объекты юниверса влияют определенные изменения объекта OLAP.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если измерение не изменяется \[страница 472\]](#)

[Если измерение обновляется \(имя, описание\) \[страница 473\]](#)

[Если удаляется измерение \[страница 474\]](#)

[Если измерение перемещается \[страница 474\]](#)

[Если создается иерархия или характеристика \[страница 475\]](#)

[Если создается измерение \[страница 476\]](#)

9.4.8.1 Если измерение не изменяется

В следующей таблице показано, что происходит с эквивалентным классом юниверса в различных возможных ситуациях, если измерение не изменяется:

Если класс юниверса	Что происходит с классом юниверса
Не изменено	Класс юниверса не изменяется
Обновляется	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Остается без изменений, если эти параметры выбраны.
Удаляется	Отсутствуют изменения для класса юниверса. Создается объект, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение Да. Дочерние объекты, которые не были удалены, не создаются повторно
Перемещено	Отсутствуют изменения для класса юниверса.
Скрывается	Отсутствуют изменения для класса юниверса.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если измерение обновляется \(имя, описание\) \[страница 473\]](#)

[Если удаляется измерение \[страница 474\]](#)

[Если измерение перемещается \[страница 474\]](#)

[Если создается иерархия или характеристика \[страница 475\]](#)

[Если создается измерение \[страница 476\]](#)

9.4.8.2 Если измерение обновляется (имя, описание)

В следующей таблице показано, что происходит с эквивалентным классом юниверса в различных возможных ситуациях при обновлении имени или описания измерения:

Если класс юниверса	Что происходит с классом юниверса
Не изменено	Обновляется бизнес-имя, если не выбран параметр Сохранить бизнес-имя . Обновляется описание, если не выбран параметр Сохранить описание . Остается без изменений, если эти параметры выбраны.
Обновляется	Обновляется бизнес-имя, если не выбран параметр Сохранить бизнес-имя . Обновляется описание, если не выбран параметр Сохранить описание . Остается без изменений, если эти параметры выбраны.
Удаляется	Отсутствуют изменения для класса юниверса. Создается, если выбран параметр Повторно генерировать объекты, удаленные вручную . Дочерние объекты, которые не были удалены, не создаются повторно.
Перемещено	Обновляется бизнес-имя, если не выбран параметр Сохранить бизнес-имя . Обновляется описание, если не выбран параметр Сохранить описание . Остается без изменений, если эти параметры выбраны.
Скрывается	Обновляется бизнес-имя, если не выбран параметр Сохранить бизнес-имя . Обновляется описание, если не выбран параметр Сохранить описание . Остается без изменений, если эти параметры выбраны.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если измерение не изменяется \[страница 472\]](#)

[Если удаляется измерение \[страница 474\]](#)

[Если измерение перемещается \[страница 474\]](#)

[Если создается иерархия или характеристика \[страница 475\]](#)

[Если создается измерение \[страница 476\]](#)

9.4.8.3 Если удаляется измерение

В следующей таблице показано, что происходит с эквивалентным классом юниверса в различных возможных ситуациях при удалении измерения:

Если класс юниверса	Что происходит с классом юниверса
Не изменено	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс становится устаревшим, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Не удаляется, если класс содержит пользовательские объекты
Обновляется	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс становится устаревшим, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Не удаляется, если класс содержит пользовательские объекты
Удаляется	Отсутствуют изменения для класса юниверса.
Перемещено	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс становится устаревшим, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Не удаляется, если класс содержит пользовательские объекты
Скрывается	Отсутствуют изменения для класса юниверса

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если измерение не изменяется \[страница 472\]](#)

[Если измерение обновляется \(имя, описание\) \[страница 473\]](#)

[Если измерение перемещается \[страница 474\]](#)

[Если создается иерархия или характеристика \[страница 475\]](#)

[Если создается измерение \[страница 476\]](#)

9.4.8.4 Если измерение перемещается

В следующей таблице показано, что происходит с эквивалентным классом юниверса в различных возможных ситуациях при перемещении измерения:

Если класс юниверса	Что происходит с классом юниверса
Не изменено	Соответственно перемещается класс
Обновляется	Без изменений
Удаляется	Без изменений. Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i> Дочерние объекты, которые не были удалены, не создаются повторно
Перемещено	Без изменений
Скрывается	Соответственно перемещается класс

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если измерение не изменяется \[страница 472\]](#)

[Если измерение обновляется \(имя, описание\) \[страница 473\]](#)

[Если удаляется измерение \[страница 474\]](#)

[Если создается иерархия или характеристика \[страница 475\]](#)

[Если создается измерение \[страница 476\]](#)

9.4.8.5 Если создается иерархия или характеристика

Иерархия относится к источникам данных MSAS и Essbase, а характеристика относится к источникам данных SAP. В следующей таблице показано, что происходит с эквивалентным классом юниверса в различных возможных ситуациях при создании характеристики SAP:

Если класс юниверса	Что происходит с классом юниверса
Не изменено	Создается подкласс
Обновляется	Создается подкласс
Удаляется	Не применимо
Перемещено	Создается подкласс
Скрывается	Создается подкласс

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если измерение не изменяется \[страница 472\]](#)

[Если измерение обновляется \(имя, описание\) \[страница 473\]](#)

[Если удаляется измерение \[страница 474\]](#)

[Если измерение перемещается \[страница 474\]](#)

[Если создается измерение \[страница 476\]](#)

9.4.8.6 Если создается измерение

При создании измерения создается класс юниверса.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если измерение не изменяется \[страница 472\]](#)

[Если измерение обновляется \(имя, описание\) \[страница 473\]](#)

[Если удаляется измерение \[страница 474\]](#)

[Если измерение перемещается \[страница 474\]](#)

[Если создается иерархия или характеристика \[страница 475\]](#)

9.4.9 Управление иерархиями и характеристиками при обновлении юниверсов OLAP

Этот раздел относится к иерархиям для источников данных MSAS и Essbase, а также характеристикам для источников данных SAP. Подкласс юниверса соответствует характеристике OLAP. Управление объектами юниверса в соответствии с объектами OLAP зависит от типа изменения. См. в перечисленных ниже тематических разделах, как на объекты юниверса влияют определенные изменения объекта OLAP.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если иерархия или характеристика не изменяются \[страница 477\]](#)

[Если обновляется название фирмы или описание характеристики \[страница 477\]](#)

[Если изменяется активная иерархия характеристики \[страница 478\]](#)

[Если атрибут отображения характеристики меняется на атрибут навигации \[страница 479\]](#)

[Удаление иерархии или характеристики \[страница 481\]](#)

[Перемещение иерархии или характеристики \[страница 481\]](#)

[Если создается иерархия или характеристика \[страница 482\]](#)

9.4.9.1 Если иерархия или характеристика не изменяются

В следующей таблице показано, что происходит с эквивалентным подклассом юниверса в различных возможных ситуациях, если иерархия или характеристика не изменяются:

Если подкласс юниверса	Что происходит с подклассом юниверса
Не изменено	Без изменений
Обновляется	Без изменений
Удаляется	Без изменений. Создается, если для параметра Повторно генерировать объекты, удаленные вручную установлено значение Да Дочерние уровни, которые не были удалены, не создаются повторно
Перемещено	Без изменений
Скрывается	Без изменений

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если обновляется название фирмы или описание характеристики \[страница 477\]](#)

[Если изменяется активная иерархия характеристики \[страница 478\]](#)

[Если атрибут отображения характеристики меняется на атрибут навигации \[страница 479\]](#)

[Удаление иерархии или характеристики \[страница 481\]](#)

[Перемещение иерархии или характеристики \[страница 481\]](#)

[Если создается иерархия или характеристика \[страница 482\]](#)

9.4.9.2 Если обновляется название фирмы или описание характеристики

В следующей таблице показано, что происходит с эквивалентным подклассом юниверса в различных возможных ситуациях при обновлении характеристики:

Если подкласс юниверса	Что происходит с подклассом юниверса
Не изменено	Обновляется бизнес-имя, если не выбран параметр Сохранить бизнес-имя . Обновляется описание, если не выбран параметр Сохранить описание . Остается без изменений, если эти параметры выбраны.

Если подкласс юниверса	Что происходит с подклассом юниверса
Обновляется	<p>Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i>.</p> <p>Обновляется описание, если не выбран параметр <i>Сохранить описание</i>.</p> <p>Остается без изменений, если эти параметры выбраны.</p>
Удаляется	<p>Без изменений.</p> <p>Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i></p> <p>Дочерние уровни, которые не были удалены, не создаются повторно</p>
Перемещено	<p>Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i>.</p> <p>Обновляется описание, если не выбран параметр <i>Сохранить описание</i>.</p> <p>Остается без изменений, если эти параметры не выбраны.</p>
Скрывается	<p>Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i>.</p> <p>Обновляется описание, если не выбран параметр <i>Сохранить описание</i>.</p> <p>Остается без изменений, если эти параметры выбраны.</p>

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если иерархия или характеристика не изменяются \[страница 477\]](#)

[Если изменяется активная иерархия характеристики \[страница 478\]](#)

[Если атрибут отображения характеристики меняется на атрибут навигации \[страница 479\]](#)

[Удаление иерархии или характеристики \[страница 481\]](#)

[Перемещение иерархии или характеристики \[страница 481\]](#)

[Если создается иерархия или характеристика \[страница 482\]](#)

9.4.9.3 Если изменяется активная иерархия характеристики

Это относится только к источникам данных SAP. В следующей таблице показано, что происходит с эквивалентным подклассом юниверса в различных возможных ситуациях при изменении активной иерархии характеристики:

Если подкласс юниверса	Что происходит с подклассом юниверса
Не изменено	<p>Обновляются определения MDX существующих объектов в подклассе для ссылки на новую активную иерархию.</p> <p>Создаются отчеты перед продолжением обновления</p>

Если подкласс юниверса	Что происходит с подклассом юниверса
Обновляется	Обновляются определения MDX существующих объектов в подклассе для ссылки на новую активную иерархию. Создаются отчеты перед продолжением обновления
Удаляется	Без изменений. Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i> Дочерние уровни, которые не были удалены, не создаются повторно.
Перемещено	Обновляются определения MDX существующих объектов в подклассе для ссылки на новую активную иерархию. Создаются отчеты перед продолжением обновления.
Скрывается	Обновляются определения MDX существующих объектов в подклассе для ссылки на новую активную иерархию.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если иерархия или характеристика не изменяются \[страница 477\]](#)

[Если обновляется название фирмы или описание характеристики \[страница 477\]](#)

[Если атрибут отображения характеристики меняется на атрибут навигации \[страница 479\]](#)

[Удаление иерархии или характеристики \[страница 481\]](#)

[Перемещение иерархии или характеристики \[страница 481\]](#)

[Если создается иерархия или характеристика \[страница 482\]](#)

9.4.9.4 Если атрибут отображения характеристики меняется на атрибут навигации

Это относится только к источникам данных SAP. В следующей таблице показано, что происходит с эквивалентным подклассом юниверса в различных возможных ситуациях, если атрибут отображения характеристики меняется на атрибут навигации:

Если подкласс юниверса	Что происходит с подклассом юниверса
Не изменено	Создать
Обновляется	Создать
Удаляется	Создать
Перемещено	Создать
Скрывается	Создать

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если иерархия или характеристика не изменяются \[страница 477\]](#)

[Если обновляется название фирмы или описание характеристики \[страница 477\]](#)

[Если изменяется активная иерархия характеристики \[страница 478\]](#)

[Удаление иерархии или характеристики \[страница 481\]](#)

[Перемещение иерархии или характеристики \[страница 481\]](#)

[Если создается иерархия или характеристика \[страница 482\]](#)

9.4.9.5 Если атрибут навигации характеристики меняется на атрибут отображения

Это относится только к источникам данных SAP. В следующей таблице показано, что происходит с эквивалентным подклассом юниверса в различных возможных ситуациях, если атрибут навигации иерархии или характеристики меняется на атрибут отображения:

Если подкласс юниверса	Что происходит с подклассом юниверса
Не изменено	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс скрывается, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Не удаляется, если класс содержит пользовательские объекты.
Обновляется	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс скрывается, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Не удаляется, если класс содержит пользовательские объекты.
Удаляется	Без изменений
Перемещено	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс скрывается, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Не удаляется, если класс содержит пользовательские объекты.
Скрывается	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс скрывается, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Не удаляется, если класс содержит пользовательские объекты.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если иерархия или характеристика не изменяются \[страница 477\]](#)

[Если обновляется название фирмы или описание характеристики \[страница 477\]](#)

[Если изменяется активная иерархия характеристики \[страница 478\]](#)

[Удаление иерархии или характеристики \[страница 481\]](#)

[Перемещение иерархии или характеристики \[страница 481\]](#)

[Если создается иерархия или характеристика \[страница 482\]](#)

9.4.9.6 Удаление иерархии или характеристики

В следующей таблице показано, что происходит с эквивалентным подклассом юниверса в различных возможных ситуациях при удалении иерархии или характеристики:

Если подкласс юниверса	Что происходит с подклассом юниверса
Не изменено	Удаляется, если выбран параметр Удалить неупотребляемые объекты . Подкласс становится устаревшим, если выбран параметр Скрыть неупотребляемые объекты . Не удаляется, если подкласс содержит пользовательские объекты.
Обновляется	Удаляется, если выбран параметр Удалить неупотребляемые объекты , становится устаревшим, если выбран параметр Скрыть неупотребляемые объекты . Не удаляется, если подкласс содержит пользовательские объекты.
Удаляется	Без изменений
Перемещено	Удаляется, если выбран параметр Удалить неупотребляемые объекты , становится устаревшим, если выбран параметр Скрыть неупотребляемые объекты . Не удаляется, если подкласс содержит пользовательские объекты.
Скрывается	Удаляется, если выбран параметр Удалить неупотребляемые объекты , становится устаревшим, если выбран параметр Скрыть неупотребляемые объекты . Не удаляется, если подкласс содержит пользовательские объекты.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если иерархия или характеристика не изменяются \[страница 477\]](#)

[Если обновляется название фирмы или описание характеристики \[страница 477\]](#)

[Если изменяется активная иерархия характеристики \[страница 478\]](#)

[Если атрибут отображения характеристики меняется на атрибут навигации \[страница 479\]](#)

[Перемещение иерархии или характеристики \[страница 481\]](#)

[Если создается иерархия или характеристика \[страница 482\]](#)

9.4.9.7 Перемещение иерархии или характеристики

Если характеристика перемещается в рамках одного измерения, изменений не происходит: игнорируйте приведенную ниже таблицу. В следующей таблице показано, что происходит с эквивалентным подклассом юниверса в различных возможных ситуациях при перемещении иерархии или характеристики в другое измерение:

Если подкласс юниверса	Что происходит с подклассом юниверса
Не изменено	Соответственно перемещается подкласс.

Если подкласс юниверса	Что происходит с подклассом юниверса
Обновляется	Соответственно перемещается подкласс.
Удаляется	Без изменений. Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i> . Дочерние уровни, которые не были удалены, не создаются повторно.
Перемещено	Без изменений.
Скрывается	Соответственно перемещается подкласс.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если иерархия или характеристика не изменяются \[страница 477\]](#)

[Если обновляется название фирмы или описание характеристики \[страница 477\]](#)

[Если изменяется активная иерархия характеристики \[страница 478\]](#)

[Если атрибут отображения характеристики меняется на атрибут навигации \[страница 479\]](#)

[Удаление иерархии или характеристики \[страница 481\]](#)

[Если создается иерархия или характеристика \[страница 482\]](#)

9.4.9.8 Если создается иерархия или характеристика

При создании иерархии или характеристики создается подкласс юниверса.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если иерархия или характеристика не изменяются \[страница 477\]](#)

[Если обновляется название фирмы или описание характеристики \[страница 477\]](#)

[Если изменяется активная иерархия характеристики \[страница 478\]](#)

[Если атрибут отображения характеристики меняется на атрибут навигации \[страница 479\]](#)

[Удаление иерархии или характеристики \[страница 481\]](#)

[Перемещение иерархии или характеристики \[страница 481\]](#)

9.4.10 Управление уровнями при обновлении юниверсов OLAP

❗ Примечание

Не перемещайте уровни в юниверсе в другие иерархии. Если необходимо переместить уровень, скопируйте и вставьте этот уровень в новую иерархию.

Уровень юниверса или объект измерения соответствует уровню OLAP. Управление объектами юниверса в соответствии с объектами OLAP зависит от типа изменения. См. в перечисленных ниже тематических разделах, как на объекты юниверса влияют определенные изменения объекта OLAP.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если уровень не изменяется \[страница 483\]](#)

[Если обновляется имя или описание уровня \[страница 484\]](#)

[Если удаляется уровень \[страница 485\]](#)

[Если уровень перемещается \[страница 485\]](#)

[Если создается уровень \[страница 486\]](#)

9.4.10.1 Если уровень не изменяется

В следующей таблице показано, что происходит с уровнем юниверса в различных возможных ситуациях, если уровень не изменяется:

Если уровень юниверса	Что происходит с уровнем юниверса
Не изменено	Без изменений
Обновляется	Без изменений
Удаляется	Без изменений. Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i>
Перемещено	Без изменений
Скрывается	Без изменений

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если обновляется имя или описание уровня \[страница 484\]](#)

[Если удаляется уровень \[страница 485\]](#)

[Если уровень перемещается \[страница 485\]](#)

[Если создается уровень \[страница 486\]](#)

9.4.10.2 Если обновляется имя или описание уровня

В следующей таблице показано, что происходит с уровнем юниверса в различных возможных ситуациях при обновлении имени или описания уровня:

Если уровень юниверса	Что происходит с уровнем юниверса
Не изменено	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Остается без изменений, если эти параметры выбраны.
Обновляется	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Остается без изменений, если эти параметры выбраны.
Удаляется	Без изменений. Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i> .
Перемещено	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Остается без изменений, если эти параметры выбраны.
Скрывается	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Остается без изменений, если эти параметры выбраны.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если уровень не изменяется \[страница 483\]](#)

[Если удаляется уровень \[страница 485\]](#)

[Если уровень перемещается \[страница 485\]](#)

[Если создается уровень \[страница 486\]](#)

9.4.10.3 Если удаляется уровень

В следующей таблице показано, что происходит с уровнем юниверса в различных возможных ситуациях при удалении уровня:

Если уровень юниверса	Что происходит с уровнем юниверса
Не изменено	Удаляется, если выбран параметр Удалить неупотребляемые объекты . Подкласс становится устаревшим, если выбран параметр Скрыть неупотребляемые объекты .
Обновляется	Удаляется, если выбран параметр Удалить неупотребляемые объекты . Подкласс становится устаревшим, если выбран параметр Скрыть неупотребляемые объекты .
Удаляется	Без изменений.
Перемещено	Удаляется, если выбран параметр Удалить неупотребляемые объекты . Подкласс становится устаревшим, если выбран параметр Скрыть неупотребляемые объекты .
Скрывается	Удаляется, если выбран параметр Удалить неупотребляемые объекты . Подкласс становится устаревшим, если выбран параметр Скрыть неупотребляемые объекты .

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если уровень не изменяется \[страница 483\]](#)

[Если обновляется имя или описание уровня \[страница 484\]](#)

[Если уровень перемещается \[страница 485\]](#)

[Если создается уровень \[страница 486\]](#)

9.4.10.4 Если уровень перемещается

В следующей таблице показано, что происходит с уровнем юниверса в различных возможных ситуациях при перемещении уровня:

Если уровень юниверса	Что происходит с уровнем юниверса
Не изменено	Соответственно перемещается уровень (в рамках той же иерархии).
Обновляется	Соответственно перемещается уровень (в рамках той же иерархии).
Удаляется	Без изменений. Создается, если для параметра Повторно генерировать объекты, удаленные вручную установлено значение Да
Перемещено	Без изменений. Создается, если для параметра Повторно генерировать объекты, удаленные вручную установлено значение Да

Если уровень юниверса	Что происходит с уровнем юниверса
Скрывается	Соответственно перемещается уровень (в рамках той же иерархии).

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)
[Если уровень не изменяется \[страница 483\]](#)
[Если обновляется имя или описание уровня \[страница 484\]](#)
[Если удаляется уровень \[страница 485\]](#)
[Если создается уровень \[страница 486\]](#)

9.4.10.5 Если создается уровень

При создании уровня OLAP создается уровень юниверса.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)
[Если уровень не изменяется \[страница 483\]](#)
[Если обновляется имя или описание уровня \[страница 484\]](#)
[Если удаляется уровень \[страница 485\]](#)
[Если уровень перемещается \[страница 485\]](#)

9.4.11 Управление переменными SAP при обновлении юниверсов OLAP

Этот раздел относится только к источникам данных SAP. Фильтр юниверса и связанные объекты списка значений соответствуют переменной OLAP. Управление объектами юниверса в соответствии с объектами OLAP зависит от типа изменения. См. в перечисленных ниже тематических разделах, как на объекты юниверса влияют определенные изменения объекта OLAP.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)
[Если переменная SAP не изменяется \[страница 487\]](#)

[Если обновляется имя или описание переменной SAP \[страница 487\]](#)

[Если удаляется переменная SAP \[страница 488\]](#)

[Если создается переменная SAP \[страница 489\]](#)

9.4.11.1 Если переменная SAP не изменяется

В следующей таблице показано, что происходит с фильтрами юниверса в различных возможных ситуациях, если переменная SAP не изменяется:

Если фильтр юниверса	Что происходит с фильтром юниверса
Не изменено	Без изменений
Обновляется	Без изменений
Удаляется	Создается. Если характеристика, указанная в переменной, отсутствует в юниверсе, для этой характеристики также создается подкласс.
Перемещено	Без изменений
Скрывается	Без изменений

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если обновляется имя или описание переменной SAP \[страница 487\]](#)

[Если удаляется переменная SAP \[страница 488\]](#)

[Если создается переменная SAP \[страница 489\]](#)

9.4.11.2 Если обновляется имя или описание переменной SAP

В следующей таблице показано, что происходит с фильтрами юниверса в различных возможных ситуациях при обновлении имени или описания исходной переменной SAP:

Если фильтр юниверса	Что происходит с фильтром юниверса
Не изменено	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> .
	Обновляется описание, если не выбран параметр <i>Сохранить описание</i> .
	Остается без изменений, если эти параметры выбраны.

Если фильтр юниверса	Что происходит с фильтром юниверса
Обновляется	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Остается без изменений, если эти параметры выбраны.
Удаляется	Создается. Если характеристика, указанная в переменной, отсутствует в юни-версе, для этой характеристики также создается подкласс.
Перемещено	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Остается без изменений, если эти параметры выбраны.
Скрывается	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Остается без изменений, если эти параметры выбраны.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если переменная SAP не изменяется \[страница 487\]](#)

[Если удаляется переменная SAP \[страница 488\]](#)

[Если создается переменная SAP \[страница 489\]](#)

9.4.11.3 Если удаляется переменная SAP

В следующей таблице показано, что происходит с фильтром юниверса в различных возможных ситуациях при удалении переменной SAP:

Если фильтр юниверса	Что происходит с фильтром юниверса
Не изменено	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Под-класс скрывается, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Если становится устаревшим: также изменяется с обязательного на необяза-тельный, чтобы избежать автоматического применения в запросах.
Обновляется	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Под-класс скрывается, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Если становится устаревшим: также изменяется с обязательного на необяза-тельный, чтобы избежать автоматического применения в запросах.
Удаляется	Без изменений.

Если фильтр юниверса	Что происходит с фильтром юниверса
Перемещено	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс скрывается, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Если становится устаревшим: также изменяется с обязательного на необязательный, чтобы избежать автоматического применения в запросах.
Скрывается	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс скрывается, если выбран параметр <i>Скрыть неупотребляемые объекты</i> . Если становится устаревшим: также изменяется с обязательного на необязательный, чтобы избежать автоматического применения в запросах.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если переменная SAP не изменяется \[страница 487\]](#)

[Если обновляется имя или описание переменной SAP \[страница 487\]](#)

[Если создается переменная SAP \[страница 489\]](#)

9.4.11.4 Если создается переменная SAP

В следующей таблице показано, что происходит с фильтрами юниверса в различных возможных ситуациях при создании переменной SAP:

Если фильтр юниверса	Что происходит с фильтром юниверса
Не изменено	Создается. Если характеристика, указанная в переменной, отсутствует в юниверсе, для этой характеристики также создается подкласс.
Обновляется	Создается. Если характеристика, указанная в переменной, отсутствует в юниверсе, для этой характеристики также создается подкласс.
Удаляется	Создается. Если характеристика, указанная в переменной, отсутствует в юниверсе, для этой характеристики также создается подкласс.
Перемещено	Создается. Если характеристика, указанная в переменной, отсутствует в юниверсе, для этой характеристики также создается подкласс.
Скрывается	Создается. Если характеристика, указанная в переменной, отсутствует в юниверсе, для этой характеристики также создается подкласс.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если переменная SAP не изменяется \[страница 487\]](#)

[Если обновляется имя или описание переменной SAP \[страница 487\]](#)

[Если удаляется переменная SAP \[страница 488\]](#)

9.4.12 Управление контрольными цифрами и мерами при обновлении юниверсов OLAP

Источники данных SAP используют контрольные цифры, а источники данных MSAS и Essbase используют меры. Мера юниверса соответствует контрольной цифре OLAP. Управление объектами юниверса в соответствии с объектами OLAP зависит от типа изменения. См. в перечисленных ниже тематических разделах, как на объекты юниверса влияют определенные изменения объекта OLAP.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если контрольная цифра или мера не изменяется \[страница 490\]](#)

[Если обновляется имя, описание или тип данных контрольной цифры или меры: \[страница 491\]](#)

[Если удаляется контрольная цифра или мера \[страница 492\]](#)

[Если перемещается контрольная цифра или мера \[страница 492\]](#)

[Если создается контрольная цифра или мера \[страница 493\]](#)

9.4.12.1 Если контрольная цифра или мера не изменяется

В следующей таблице показано, что происходит с мерой юниверса в различных возможных ситуациях, если контрольная цифра SAP или мера MSAS/Essbase не изменяется:

Если мера юниверса	Что происходит с мерой юниверса
Не изменено	Без изменений
Обновляется	Без изменений
Удаляется	Без изменений. Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i>
Перемещено	Без изменений
Скрывается	Без изменений

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если обновляется имя, описание или тип данных контрольной цифры или меры: \[страница 491\]](#)

[Если удаляется контрольная цифра или мера \[страница 492\]](#)

[Если перемещается контрольная цифра или мера \[страница 492\]](#)

[Если создается контрольная цифра или мера \[страница 493\]](#)

9.4.12.2 Если обновляется имя, описание или тип данных контрольной цифры или меры:

В следующей таблице показано, что происходит с мерой юниверса в различных возможных ситуациях при обновлении контрольной цифры SAP или меры MSAS/Essbase:

Если мера юниверса	Что происходит с мерой юниверса
Не изменено	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Обновляется тип данных, если не выбран параметр <i>Сохранить тип данных объектов</i> . Остается без изменений, если эти параметры выбраны.
Обновляется	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Обновляется тип данных, если не выбран параметр <i>Сохранить тип данных объектов</i> . Остается без изменений, если эти параметры выбраны.
Удаляется	Без изменений. Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i>
Перемещено	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Обновляется тип данных, если не выбран параметр <i>Сохранить тип данных объектов</i> . Остается без изменений, если эти параметры выбраны.
Скрывается	Обновляется бизнес-имя, если не выбран параметр <i>Сохранить бизнес-имя</i> . Обновляется описание, если не выбран параметр <i>Сохранить описание</i> . Обновляется тип данных, если не выбран параметр <i>Сохранить тип данных объектов</i> . Остается без изменений, если эти параметры выбраны.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если контрольная цифра или мера не изменяется \[страница 490\]](#)

[Если удаляется контрольная цифра или мера \[страница 492\]](#)

[Если перемещается контрольная цифра или мера \[страница 492\]](#)

[Если создается контрольная цифра или мера \[страница 493\]](#)

9.4.12.3 Если удаляется контрольная цифра или мера

В следующей таблице показано, что происходит с параметром юниверса в различных возможных ситуациях при удалении контрольной цифры SAP или меры MSAS/Essbase:

Если мера юниверса	Что происходит с мерой юниверса
Не изменено	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс становится устаревшим, если выбран параметр <i>Скрыть неупотребляемые объекты</i> .
Обновляется	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс становится устаревшим, если выбран параметр <i>Скрыть неупотребляемые объекты</i> .
Удаляется	Без изменений.
Перемещено	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс становится устаревшим, если выбран параметр <i>Скрыть неупотребляемые объекты</i> .
Скрывается	Удаляется, если выбран параметр <i>Удалить неупотребляемые объекты</i> . Подкласс становится устаревшим, если выбран параметр <i>Скрыть неупотребляемые объекты</i> .

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если контрольная цифра или мера не изменяется \[страница 490\]](#)

[Если обновляется имя, описание или тип данных контрольной цифры или меры: \[страница 491\]](#)

[Если перемещается контрольная цифра или мера \[страница 492\]](#)

[Если создается контрольная цифра или мера \[страница 493\]](#)

9.4.12.4 Если перемещается контрольная цифра или мера

В следующей таблице показано, что происходит с параметром юниверса в различных возможных ситуациях при перемещении контрольной цифры SAP или меры MSAS/Essbase:

Если мера юниверса	Что происходит с мерой юниверса
Не изменено	Соответственно перемещается объект.
Обновляется	Соответственно перемещается объект.
Удаляется	Без изменений. Создается, если для параметра <i>Повторно генерировать объекты, удаленные вручную</i> установлено значение <i>Да</i> .
Перемещено	Без изменений.

Если мера юниверса	Что происходит с мерой юниверса
Скрывается	Соответственно перемещается объект.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если контрольная цифра или мера не изменяется \[страница 490\]](#)

[Если обновляется имя, описание или тип данных контрольной цифры или меры: \[страница 491\]](#)

[Если удаляется контрольная цифра или мера \[страница 492\]](#)

[Если создается контрольная цифра или мера \[страница 493\]](#)

9.4.12.5 Если создается контрольная цифра или мера

При создании контрольной цифры или меры создается мера юниверса.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если контрольная цифра или мера не изменяется \[страница 490\]](#)

[Если обновляется имя, описание или тип данных контрольной цифры или меры: \[страница 491\]](#)

[Если удаляется контрольная цифра или мера \[страница 492\]](#)

[Если перемещается контрольная цифра или мера \[страница 492\]](#)

9.4.13 Управление контрольными датами SAP при обновлении юниверсов OLAP

Этот раздел относится только к источникам данных SAP. Параметр юниверса соответствует контрольной дате OLAP. Управление объектами юниверса в соответствии с объектами OLAP зависит от типа изменения. См. в перечисленных ниже тематических разделах, как на объекты юниверса влияют определенные изменения объекта OLAP.

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если контрольная дата SAP не изменяется \[страница 494\]](#)

[Если удаляется контрольная дата SAP \[страница 494\]](#)

[Если создается контрольная дата SAP \[страница 495\]](#)

9.4.13.1 Если контрольная дата SAP не изменяется

Параметр юниверса соответствует контрольной дате OLAP. В следующей таблице показано, что происходит с параметром юниверса в различных возможных ситуациях, если контрольная дата SAP не изменяется:

Если параметр юниверса	Что происходит с параметром юниверса
Не изменено	Без изменений
Обновляется	Не применимо
Удаляется	Не применимо
Перемещено	Не применимо
Скрывается	Не применимо

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если удаляется контрольная дата SAP \[страница 494\]](#)

[Если создается контрольная дата SAP \[страница 495\]](#)

9.4.13.2 Если удаляется контрольная дата SAP

Параметр юниверса соответствует контрольной дате OLAP. В следующей таблице показано, что происходит с параметром юниверса в различных возможных ситуациях при удалении контрольной даты SAP:

Если параметр юниверса	Что происходит с параметром юниверса
Не изменено	Удаление
Обновляется	Не применимо
Удаляется	Не применимо
Перемещено	Не применимо
Скрывается	Не применимо

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если контрольная дата SAP не изменяется \[страница 494\]](#)

[Если создается контрольная дата SAP \[страница 495\]](#)

9.4.13.3 Если создается контрольная дата SAP

Параметр юниверса соответствует контрольной дате OLAP. В следующей таблице показано, что происходит с параметром юниверса в различных возможных ситуациях при создании контрольной даты SAP:

Если параметр юниверса	Что происходит с параметром юниверса
Не изменено	Создать
Обновляется	Не применимо
Удаляется	Создать
Перемещено	Не применимо
Скрывается	Не применимо

Связанные сведения

[Обновление юниверса OLAP \[страница 467\]](#)

[Если контрольная дата SAP не изменяется \[страница 494\]](#)

[Если удаляется контрольная дата SAP \[страница 494\]](#)

9.5 Сопоставление различных кубов OLAP с юниверсами

9.5.1 Сопоставление и использование объектов SAP BW в юниверсе

При создании юниверса с помощью InfoCube или запроса BEx средство создания юниверсов сопоставляет структуры OLAP SAP BW с классами и объектами в юниверсе.

Все объекты InfoObjects в запросе BEx, используемые как строки, столбцы, свободные характеристики и фильтры, представляются в юниверсе. К ним относятся характеристики, иерархии, контрольные цифры, структуры и переменные.

Иерархии сопоставляются, позволяя пользователям Web Intelligence выполнять развертку иерархии, согласно иерархиям BW.

Для InfoCubes сопоставляются все измерения, контрольные цифры и иерархии.

В следующей таблице отображены объекты юниверса, созданные для каждого объекта BW.

Объект SAP BW:	Созданные объекты юниверса
Группа измерений	Класс
Характеристика	Подкласс с объектами измерений и атрибутов
Характеристика с иерархией	<p>Если источник данных – запрос BEx: подкласс, содержащий измерение и объекты сведений для каждого из уровней иерархии в определенной на данный момент иерархии.</p> <p>Если источник данных – InfoCube: подклассы, содержащие измерение и объекты сведений для каждого из уровней иерархии для всех иерархий, определенных для характеристики.</p>
Структуры, основанные на характеристиках (только запросы BEx)	Класс с единым объектом измерения для структуры
Атрибут навигации	Подкласс с объектами измерений и атрибутами (идентично характеристикам)
Отображаемый атрибут	Объект-атрибут для измерений
Структура контрольных цифр	Класс
Контрольная цифра	Объект меры в классе структуры контрольных цифр с объектами измерений для денежных единиц/валюты.
Вычисляемые контрольные цифры (только запросы BEx)	Показатели и объекты измерений (аналогичны контрольным цифрам)
Ограничиваемые контрольные цифры (только запросы BEx)	Показатели и объекты измерений (аналогичны контрольным цифрам)
Переменные (только запросы BEx)	<p>Обязательный фильтр в запросе</p> <p>В классе измерений, к которым применяется переменная, два объекта измерений, поддерживающих список значений: один – для заголовка, другой – для описания.</p>
Переменные контрольных дат (только запросы BEx)	Параметры юниверса, определяющие переменную контрольной даты в юниверсе

Характеристики в разделе "Фильтры" запроса BEx не сопоставляются. Однако к юниверсу применяется фильтрация. Если фильтр имеет фиксированное значение, он применяется при запуске запроса Web

Intelligence. Если характеристика имеет определенную переменную, эта переменная сопоставима с ограничениями.

- Переменная всегда является необходимой
- Иерархия и переменные узла иерархии поддерживаются в полном объеме, за исключением переменной версии иерархии.

Чтобы избежать применения этих ограничений, рекомендуется переместить эту характеристику из раздела "Фильтр" в раздел "Свободно" запроса ВЕх.

Связанные сведения

[Сопоставление и использование характеристик \[страница 497\]](#)

[Сопоставление и использование контрольных цифр \[страница 498\]](#)

[Сопоставление и использование иерархий \[страница 498\]](#)

[Поддержка переменных в юниверсах \[страница 499\]](#)

[Сопоставление переменных для юниверса \[страница 501\]](#)

9.5.1.1 Сопоставление и использование характеристик

Если для характеристики в запросе ВЕх или в InfoCube не определена иерархия, то средство создания юниверсов создает класс, содержащий характеристику в качестве двух объектов измерения: Уровень 00 и уровень 01. Измерение уровня 00 представляет собой агрегирование характеристики при всех выбранных элементах (элемент, возвращаемый из платформы SAP NetWeaver – [Все элементы](#)). В измерении уровня 01 содержатся все элементы характеристики в качестве простого списка значений.

Для каждого объекта измерения средство создания юниверсов создает объект-атрибут для ключа, до трех объектов-атрибутов для описания (короткие, средние и длинные описания) и объект-атрибут для каждого отображения.

Утверждение SELECT определяется при использовании технического имени характеристики.

Атрибуты навигации, определенные в BW Query сопоставляются в родительском классе объектов таким же образом, как сопоставляются характеристики.

❗ Примечание

Большое количество атрибутов навигации, определенных в юниверсе, негативно влияют на производительность запроса в Web Intelligence.

Определенные в запросе ВЕх структуры, основанные на характеристиках, включаются в юниверс в качестве объектов одного измерения с элементами структуры в качестве элементов измерения.

9.5.1.2 Сопоставление и использование контрольных цифр

Все контрольные цифры в InfoCube (или определенные в запросе ВЕх) объединяются в юниверсе в единый класс объектов "Контрольные цифры".

Большинство контрольных цифр определяются в ВВ с помощью характеристики валюты или другой единицы измерения. Для каждой контрольной цифры средство создания юниверсов создает следующие элементы:

- объект меры с числовым форматом, соответствующим контрольной цифре без единицы измерения;
- объект измерения с символьным форматом, который содержит единицу измерения или валюту. Например, 'руб.', '€', 'км'.
- Объект измерения символьного формата, включающий контрольную цифру и единицу измерения (форматированное значение) основан на предпочтениях пользователя, настроенных на сервере SAP. Например, '200 руб.', '345 €', '25 км'.

Класс контрольных цифр включает вычисляемые и ограничиваемые контрольные цифры, определенные в запросе ВЕх. Исходные вычисления и ограничения применяются к запросу, но не отображаются в юниверсе.

9.5.1.3 Сопоставление и использование иерархий

Иерархии сопоставляются, чтобы разрешить пользователям Web Intelligence выполнить развертку иерархии SAP ВВ таким же образом, как и по иерархиям пользовательского юниверса.

❗ Примечание

Параметр *Использовать детализацию запроса* в диалоговом окне "Свойства" Web Intelligence значительно повышает производительность развертки.

Если для характеристики в запросе ВЕх определена иерархия, то средство создания юниверсов создает одну иерархическую структуру в юниверсе с подклассом для каждого уровня иерархии. Структура зависит от текущего определения запроса ВЕх.

- Если в запросе ВЕх определена иерархия, то средство создания юниверсов создает структуру этой иерархии в юниверсе.
- Если в запросе ВЕх определена переменная иерархии, что дает возможность пользователю выбирать иерархию во время выполнения, то средство создания юниверсов создает в юниверсе общую иерархию. Структура содержит самое большое количество уровней, определенных для доступных структур иерархии характеристики.

При построении юниверса на вершине InfoCube все определенные для характеристик иерархии выставляются в юниверсе результата. Средство создания юниверсов создает подклассы для каждой иерархической структуры. Каждый из этих подклассов содержит подклассы для уровней иерархии.

В юниверсе уровень иерархии 00 представляет собой верхний узел структуры. Если в иерархической структуре присутствуют несколько вершук, измерение уровня 00 содержит все верхние узлы в качестве списка значений. При установке атрибута иерархии на отмену неназначенных узлов фильтра,

необходимо включить уровень 00 в верхний узел для неназначенных элементов. Неназначенные элементы сгруппированы на нижнем уровне иерархии.

📘 Примечание

Наиболее часто в иерархиях SAP BW присутствует только один верхний узел. Можно удалить объект уровня 00 из юниверса по умолчанию, чтобы упростить юниверс для использования. Если существует потребность в неназначенных элементах запроса/отчета, необходимо, как правило, сохранять уровень 00.

Если в запросе BEx изменяется количество уровней иерархии, необходимо обновить юниверс.

Связанные сведения

[Управление жизненным циклом юниверса OLAP \[страница 464\]](#)

9.5.1.4 Поддержка переменных в юниверсах

Переменные SAP могут быть представлены как пользовательские подсказки для ввода, определенные в BW Query. Переменные могут являться обязательными или дополнительными и иметь значения по умолчанию.

Переменные для характеристик используются для фильтрации значений характеристик. Переменные заполняются значениями при выполнении запроса. В них хранятся значения характеристик, иерархии, узлы иерархий, тексты и элементы формул.

Переменные SAP BW применяются только к запросам BEx.

📘 Примечание

При определении переменной в Query Designer в диалоговом окне "Другие параметры мастера переменных SAP BW" необходимо выбрать параметр "Готово для ввода данных".

В юниверсах поддерживаются следующие типы переменных SAP BW:

- Переменные характеристик
- Переменные иерархии, кроме переменной версии иерархии
- Переменные узла иерархии
- Переменные валюты
- Переменные формулы
- Текстовые переменные (в качестве пути замены)
- Переменные контрольной даты

В следующей таблице отображена поддержка юниверса для переменных BW записи пользователя. Переменные могут являться обязательными или дополнительными и иметь значения по умолчанию.

Тип переменной	Уровень поддержки	
Характеристика (включая контрольную дату и валюту)	подсказка единого значения	поддерживается
	подсказка для нескольких единых значений	поддерживается
	внутренняя подсказка	поддерживается
		это не поддерживается для переменной контрольной даты, которая является переменной с одним значением
	подсказка на выбор параметра	поддерживается как подсказка для ввода интервала
		это не поддерживается для переменной контрольной даты, которая является переменной с одним значением
	ранее вычисленный набор значений	не поддерживается
Текст	поддерживается	
Формула	поддерживаются цена, квота и числовые значения	
Иерархия	поддерживается кроме переменной версии	
Узел иерархии	поддерживается	

В следующей таблице отображена поддержка юниверса для других типов обработки переменных BW.

Тип переменной	Тип обработки			
	Путь замещения	Авторизация	Вход для клиента	Вход SAP
Характеристика	поддерживается	поддерживается	поддерживается, подсказка в юниверсе не создается	поддерживается
Текст	поддерживается	Н/Д	поддерживается	Н/Д
Формула	поддерживается	Н/Д	поддерживается	поддерживается без записи пользователя
Иерархия	Н/Д	Н/Д	поддерживается	поддерживается
Узел иерархии	Н/Д	Н/Д	поддерживается	поддерживается без ввода пользователя

Оператор "Исключить" поддерживается, однако Web Intelligence не указывает, что выбранное значение исключается из запроса. Другие операторы, такие как "Меньше" и "Больше" используются с типом записи параметра выбора. Тип параметра выбора преобразуется в интервал для подсказок Web Intelligence.

❗ Примечание

Для обработки переменных BW в Web Intelligence необходимо включить, по крайней мере, одну меру в запрос Web Intelligence.

Связанные сведения

[Сопоставление переменных для юниверса \[страница 501\]](#)

[Поддержка переменных контрольных дат в юниверсе \[страница 503\]](#)

[Поддержка переменных иерархий и узлов иерархий в юниверсе \[страница 504\]](#)

9.5.1.4.1 Сопоставление переменных для юниверса

Пользователю необходимо выдавать подсказки на ввод всех дополнительных и обязательных переменных, даже если в результирующем наборе не используется измерение, позволяющее пользователю ограничивать результирующий набор. Поэтому переменная, определенная в запросе ВЕх, сопоставляется даже в том случае, если соответствующая характеристика отсутствует в запросе.

Пользователь должен знать, является ли переменная обязательной или необязательной, и иметь возможность пропускать необязательные переменные. Необязательные переменные определены как необязательные в юниверсе и являются необязательными подсказками в Web Intelligence. Обязательные переменные являются обязательными подсказками в Web Intelligence.

Для переменных характеристики средство создания юниверсов создает в юниверсе обязательный фильтр. Обязательный фильтр – это ранее определенный объект фильтра запроса, который скрыт от пользователей Web Intelligence, но систематически применяется ко всем запросам Web Intelligence, построенных на юниверсе.

Тип переменной	Сопоставляется с
Переменной характеристики, включая переменную валюты и формулы.	Обязательный фильтр юниверса
Переменная иерархии	Обязательный фильтр юниверса
Переменная узла иерархии	Обязательный фильтр класса
Переменная контрольной даты	Параметрах юниверса

Для каждого обязательного фильтра создается два объекта измерений в качестве объектов ссылки функции @Prompt для отображения ожидаемого списка значений. Измерения списка значений скрыты в юниверсе. Они необходимы для правильного функционирования подсказки. Поэтому необходимо не удалять и осторожно передвигать или изменять.

Значения по умолчанию переменных определены в функции @Prompt фильтра, который использует первичный ключ, постоянный/непостоянный, параметры значения по умолчанию. Синтаксис функции @Prompt виден на странице "Свойства" фильтра в юниверсе.

Чтобы избежать конфликта между переменными BW и фильтрами, определенными пользователями Web Intelligence, объекты в определении переменной SAP создаются в том случае, если параметр *Может быть использовано в условии* не выбран на странице *Дополнительно* свойств объекта. Это ограничивает пользователей Web Intelligence от включения измерений переменной SAP в панели фильтра.

Пример: Утверждение WHERE, созданное для переменной SAP BW

Данный пример отображает утверждение WHERE, созданное для переменной BW в объекте измерения "Клиент2". Синтаксис созданного утверждения WHERE для переменной отображается на странице свойств фильтра.

```
<FILTER KEY="[Z_VAR002]">
  <CONDITION OPERATORCONDITION="Equal">
    <CONSTANT TECH_NAME="@Prompt(
      'Customer Variable Single Value Mandatory',
      'A',
      'Customer2\LovCustomer Variable Single Value MandatoryBase',
      mono,
      primary_key)"/>
    <CONDITION>
  </FILTER>
```

Текст подсказки создается с помощью имени переменной BW. Можно редактировать текст, чтобы сделать его более описательным.

Customer2\LovCustomer Variable Single Value MandatoryBase – это имя скрытого объекта юниверса, который используется для построения списка значений.

📘 Примечание

При переименовании класса или передвижении списка объекта значений в другое поле необходимо обновить синтаксис ключа фильтра.

9.5.1.4.2 Поддержка переменных и списков значений

Запрос ВЕх может содержать более десяти переменных, соответственно, может загружаться более десяти списков значений. Важное влияние на производительность оказывает загрузка и обновление списка значений. Следующие параметры доступны для повышения производительности запроса для запросов с переменными.

- Во время генерации юниверса переменные SAP BW (кроме контрольной даты) сопоставляются с обязательными фильтрами. По умолчанию объекты фильтра не связаны со списком значений (кроме переменных узла иерархии). Необходимо связать список значений на странице свойств объекта.
- Дополнительные переменные создаются в качестве дополнительных подсказок. Дополнительная подсказка не загружает список значений во время запуска запроса.
- Параметр делегированного поиска в списке свойств значений предоставляет пользователю пустой список значений во время запуска запроса. Пользователь вводит критерий поиска для ограничения числа значений, возвращенных в список значений.

Для активизации параметра делегированного поиска списка значений измените объект на странице свойств объекта, к которому применяется список значений.

📘 Примечание

Делегированный поиск каскадных списков значений не поддерживается.

Связанные сведения

[Дополнительные подсказки в юниверсах OLAP \[страница 463\]](#)

9.5.1.4.3 Поддержка переменных контрольных дат в юниверсе

Переменная контрольной даты в запросе ВЕх позволяет задавать дату для данных, зависящих от времени. Контрольные даты могут влиять на данные, извлекаемые для измерения (например, со временем описание продукта может изменяться). Контрольная дата влияет на структуру иерархии, например центр специальной цены может быть для одного года на уровне 01, а для другого – на уровне 02.

Переменная контрольной даты – особая переменная SAP BW, так как значение даты, введенное пользователем, не включено в измерение BW Query. Контрольная дата – свойство запроса.

Переменная контрольной даты в запросе ВЕх может определяться двумя способами.

- Чтобы указать допустимую дату особой иерархии с воздействием только на нее.
- Чтобы указать дату полного запроса. В этом случае контрольная дата, установленная в запросе, влияет на следующие показатели:
 - master-данные, зависящие от времени
 - курсы обмена валюты
 - список иерархий
 - зависящие от времени структуры иерархии

❗ Примечание

В юниверсе использование контрольной даты ограничивается всем юниверсом. Поэтому контрольная дата, созданная в юниверсе, влияет на другие переменные SAP и данные.

SAP BW поддерживает только одну переменную контрольной даты на один BW-запрос, поэтому в юниверсе содержится только одна переменная контрольной даты.

Контрольная дата может являться обязательной или дополнительной со значением по умолчанию. Если значение по умолчанию не определено, а пользователь не вводит значение, в запросе используется текущая дата системы.

Свойства переменной контрольной даты запроса сопоставляются с пятью параметрами юниверса, описанными в следующей таблице.

Параметр	Описание
KEYDATE_ENABLED	Выберите " Да ", если контрольная дата включена в юниверс.
KEYDATE_NAME	Техническое имя переменной контрольной даты
KEYDATE_CAPTION	Заголовок переменной контрольной даты отображается при запросе значения у пользователя.

Параметр	Описание
KEYDATE_DEFAULT_VALUE	Значение по умолчанию для существующей контрольной даты
KEYDATE_MANDATORY	Выберите "Да", если пользователю необходимо ввести значение или использовать его по умолчанию.

Во время выполнения запроса Web Intelligence предлагает одинаковую контрольную дату для всех запросов. Пользователь может изменить контрольную дату. Диалоговое окно [Свойства контрольной даты](#) доступно для управления выбором контрольной даты. У пользователя запрашивается контрольная дата перед подсказкой типа переменной.

9.5.1.4.4 Поддержка переменных иерархий и узлов иерархий в универсе

Переменная иерархии используется для вывода пользователю подсказки для ввода иерархии, которая будет включена в запрос. Пользователи Web Intelligence могут создавать запросы и отчеты для получения и отображения элементов из иерархии.

Если переменная иерархии дополнительная, а пользователь не заполняет подсказку, иерархия в отчете не используется.

В отчете содержится большое количество уровней иерархии, независимых от выбранной иерархии. В отчете уровни иерархии, не возвращенные в набор результатов, остаются пустыми.

Переменная узла иерархии требует от пользователя узел, определенный в качестве узла для иерархии запроса.

Если в запросе содержится иерархия и переменная узла иерархии, пользователю Web Intelligence необходимо выбрать иерархию в списке доступных иерархий. Далее, пользователь выбирает узел иерархии. В списке доступных узлов иерархии отображаются узлы иерархии для всех иерархий. Список не фильтруется на основе выбранной иерархии. Пользователь является ответственным при выборе узла для верной иерархии. Из-за выбора узла иерархии из другой иерархии отчет может опустеть.

Связанные сведения

[Сопоставление и использование иерархий \[страница 498\]](#)

9.5.2 Сопоставление кубов Essbase и компонентов универса

Средство создания универсов создает универс из куба Essbase, преобразуя структуры Essbase в соответствии с классами и объектами. При создании соединения указывается источник данных куба.

Таблицы псевдонимов Essbase определяют множество альтернативных имен для измерений, уровней и элементов в структуре. Средство создания юниверсов создает юниверс, используя имена в соответствии с таблицей псевдонимов, выбранной при создании соединения с источником данных Essbase.

В структуре Essbase меры определяются как измерения. Выберите измерение в качестве показателя мер при создании соединения с источником данных Essbase. Средство создания юниверсов создает элементы этого измерения как меры в юниверсе.

Любое измерение поддерживает многоуровневые иерархии. Для каждого измерения может быть установлена максимум одна иерархия.

В следующей таблице представлены объекты, созданные в юниверсе для каждого элемента структуры Essbase.

Объект Essbase	Созданные объекты юниверса.
Измерение	Класс, содержащий генерации для измерения.
Генерация	Объект в классе измерения с двумя детализированными объектами: объектом заголовка и объектом имени.
Измерение мер	Класс, названный в соответствии с размерностью, выбранной в качестве измерения мер в соединении юниверса (класс Меры или класс Учетные записи).
Показатель	Объект мер в классе или подклассе мер. Меры создаются по структуре класса или подкласса в соответствии со структурой Essbase.

Показатели генерируются с помощью функции агрегирования проекции, для которой по умолчанию установлено значение Database delegated. При обновлении отчета Web Intelligence, агрегирование меры делегируется на сервер базы данных.

9.5.3 Сопоставление кубов MSAS с компонентами юниверса

Средство создания юниверсов создает юниверс из кубов MSAS, сопоставляя структуры MSAS с эквивалентными классами и объектами. При создании соединения указывается источник данных куба.

В следующей таблице представлены объекты, созданные в структурах юниверса для каждого объекта MSAS. Данное сопоставление применимо для виртуальных и локальных кубов MSAS (файлы .cub) наряду со стандартными кубами MSAS.

Объект MSAS.	Созданные объекты юниверса.
Измерение	Класс, содержащий объекты измерения.
Отображаемый каталог (MSAS 2005)	Подкласс в классе измерения.

Объект MSAS.	Созданные объекты универса.
Иерархия	Подкласс в соответствующем классе измерения или подкласс в соответствующем классе отображаемого каталога.
Атрибут (MSAS 2005)	Подкласс в соответствующем классе измерения или подкласс в соответствующем классе отображаемого каталога.
Меры	Класс "Мера", содержащий все объекты мер. Объекты мер создаются в классе Меры или в подклассе для группы мер.
Группа мер (MSAS 2005)	Подкласс в классе Меры.
Уровень	Объект в классе или подклассе измерения, а также объект всеобщего уровня, который отображает агрегирование всех подуровней.
Свойство уровня	Сведения о соответствующем объекте уровня.

Показатели генерируются с помощью функции агрегирования проекции, для которой по умолчанию установлено значение `Database delegated`. При обновлении отчета Web Intelligence, агрегирование меры делегируется на сервер базы данных.

10 Развертывание юниверсов

10.1 Обзор

Эта глава посвящена развертыванию юниверсов и управлению ими. В ней описывается следующее.

- [Как выполняется развертывание юниверсов? \[страница 507\]](#)
- [Установка ограничений доступа к юниверсу \[страница 508\]](#)
- [Управление пользователями и регистрационными именами \[страница 518\]](#)

10.2 Как выполняется развертывание юниверсов?

Развертывание юниверса – это действие, с помощью которого юниверс становится доступным для пользователей Web Intelligence или для других дизайнеров. Разверните юниверс, экспортировав его в репозиторий сервера центрального управления (CMS).

Протестируйте юниверс с помощью экспорта в тестовый репозиторий и запуска проверки в Web Intelligence до экспорта в рабочий репозиторий.

Необходимо разворачивать юниверс для пользователей Web Intelligence, только в случае завершения фаз разработки, создания и тестирования.

Сведений о развертывании юниверсов с помощью импорта юниверса в репозиторий и экспорта юниверса из репозитория см. в следующих разделах.

- [Импорт юниверса \[страница 42\]](#)
- [Экспорт юниверса \[страница 44\]](#)

10.2.1 Определение юниверса в репозитории

Юниверс определяется в репозиторий с помощью следующих параметров:

Идентификатор	Описание
Имя файла	Максимум 100 символов и расширение .unv.
Длинное имя	Включает до 35 символов. Это имя, по которому конечный пользователь определяет юниверс в Web Intelligence. Поэтому это должно быть имя, которое описывает цель юниверса.

Идентификатор	Описание
Уникальный системный идентификатор	Идентификатор, назначенный с помощью CMS.

10.2.1.1 Правила идентификатора юниверса

Следующие правила применяют для идентификаторов юниверса для юниверсов, хранящихся в папке юниверса в репозитории.

- Идентификатор юниверса в CMS уникален.
- Совокупность имени файла и положения папки (путь). Юниверс уникален в родительской папке.

10.3 Предоставление доступа к юниверсу для всех пользователей

Чтобы сделать юниверс доступным для всех пользователей средства создания юниверсов в режимах рабочей группы и предприятия, необходимо сохранить юниверс в режиме рабочей группы. Соединение с юниверсом не может быть защищенным соединением. Чтобы сделать юниверс доступным для всех пользователей, необходимо сохранить юниверс с незащищенным соединением.

Чтобы сделать юниверс доступным для всех пользователей средства создания юниверсов, выполните следующие действия:

1. Убедитесь, что юниверс, который необходимо сделать доступным для всех пользователей, не имеет защищенного соединения.
2. Безопасные соединения необходимы для экспорта юниверса в CMS. Если юниверс имеет безопасное соединение, выберите или создайте новое общедоступное соединение. Дополнительные сведения см. в разделе [Изменение параметров идентификации юниверсов \[страница 78\]](#).
3. Выберите **► Файл ► Сохранить как ►**.
Откроется окно [Сохранение файла](#).
4. Установите флажок [Сохранить для всех пользователей](#).
5. Нажмите кнопку [ОК](#).

10.4 Установка ограничений доступа к юниверсу

Можно применить ограничения для определенного пользователя или группы пользователей, использующих юниверс.

Безопасность юниверса контролируется на двух уровнях:

Уровень безопасности	Описание
CMS	В CMS можно установить ограничения, которые применяются к юниверсам, хранящимся в CMS. Можно установить, к каким юниверсам пользователи имеют доступ, и в зависимости от прав, определенных для пользователя, можно предотвратить просмотр, редактирование, удаление и другие действия в юниверсе. Данное руководство не охватывает ограничения по настройке на уровне CMS. Дополнительные сведения об использовании системы центрального управления см. руководство администратора BusinessObjects Enterprise.
Юниверс	Можно определить ограничения для пользователей, разрешенные для использования юниверса. В ограничение входит доступ к объекту, доступ к строке, средство управления запроса и генерации SQL, а также средство управления соединения. В этом руководстве описываются типы ограничений, которые можно установить для юниверса.

10.4.1 Что представляет собой ограничение?

Ограничение – это названная группа ограничений, применяемых в юниверсе. Можно применять ограничения для выбранной группы пользователей или учетной записи пользователя юниверса. Если пользователь подключается к юниверсу, объекты, строки, типы запроса и соединения, которые используются в юниверсе, определяются с помощью применяемого ограничения.

Можно назначить ограничение пользователю или группе пользователей BusinessObjects. Это ограничивает доступ к объектам или ресурсам юниверса на основе профиля группы пользователя.

10.4.2 Какие ограничения можно применять в юниверсе?

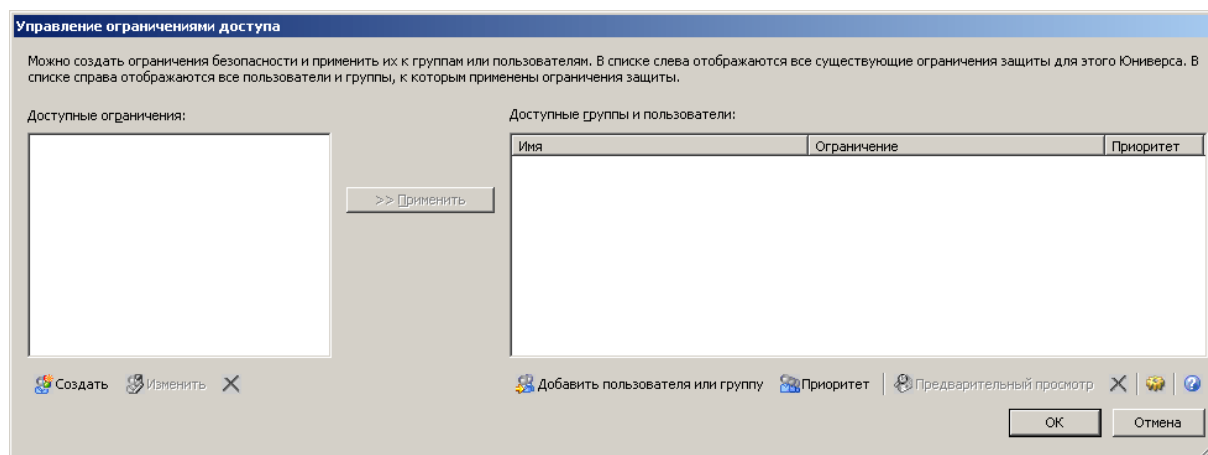
Ограничения доступа, которые применяются к группе пользователей, установлены в ограничении. Для юниверса можно установить несколько ограничений. Ограничения можно редактировать или удалять в любое время.

Ограничение может установить следующие типы ограничений для применения к выбранному пользователю или группе пользователей юниверса.

Тип ограничения	Описание
Соединение	Соединение юниверса с источником данных. Для юниверса можно выбрать альтернативное соединение. Создание и редактирование соединений описано в разделе Изменение параметров идентификации юниверсов [страница 78].
Средство управления запроса	Параметры для ограничения результатов и время выполнения запроса. Дополнительные сведения см. в разделе Определение ограничений SQL [страница 94].
Параметры генерации SQL	Параметры управления сгенерированного SQL для запросов. Дополнительные сведения см. в разделе Установка параметров генерации SQL [страница 96].
Доступ к объекту	Можно выбрать объекты, недоступные к юниверсу.
Доступ к строке	Можно определить утверждение WHERE, которое ограничивает доступ к строке и результаты, возвращаемые запросом.
Доступ к альтернативной таблице	Можно заменить таблицу, на которую имеется ссылка в юниверсе, другой таблицей в базе данных.

10.4.3 Как управлять ограничениями доступа




Ограничения доступа управляются в диалоговом окне "Управлять ограничениями доступа". Получите доступ к данному окну, выбрав "Инструменты" > "Управлять безопасностью" > "Управлять ограничениями доступа". Ниже откроется диалоговое окно.



Ограничения, доступные для юниверса в данный момент, перечислены на панели "Доступное ограничение".

Пользователи и группы пользователей, которые определены для каждого юниверса, отображаются на панели "Доступные пользователи и группы пользователей".

Параметры, доступные для управления ограничениями доступа, описаны ниже.

Параметр ограничения	Описание
Создать	Определить новое ограничение.
Правка	Изменить существующее ограничение.
Удалить выбранное ограничение	Удалите ограничение в списке.
	
Добавить пользователя или группу пользователей	Добавьте пользователя или группу пользователей в список пользователей BusinessObjects, определенных для системы BusinessObjects.
Приоритет	Позволяет назначить уровень приоритета для одной или нескольких групп пользователей.
Предварительный просмотр	Позволяет просматривать всех пользователей или групп пользователей, определенных в системе BusinessObjects.
Удалите параметр безопасности для выбранных пользователей или группы пользователей	Снимает все ограничения, установленные для выбранных пользователей или групп пользователей.
	
Параметры ограничений	Позволяют выбрать, с каким оператором применяются ограничения строк: AND или OR.
	

10.4.4 Создание ограничения

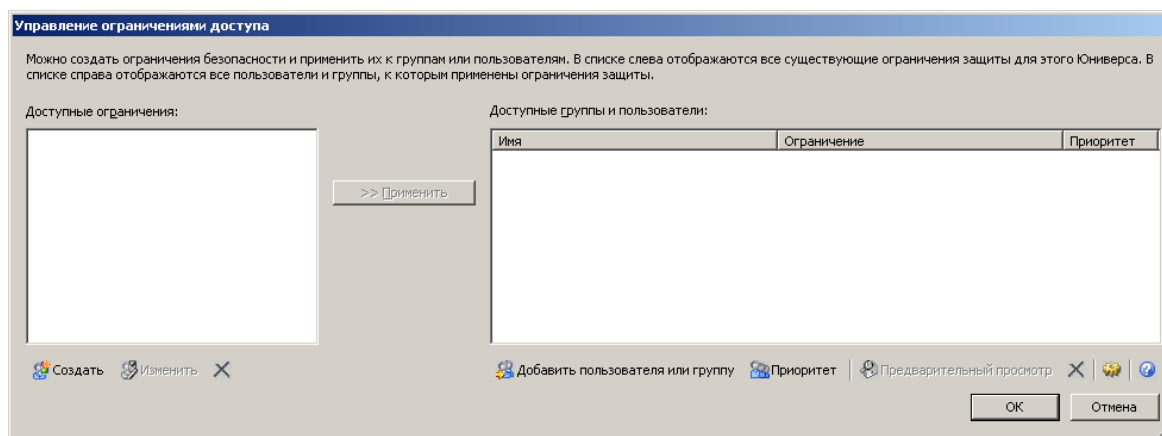
Совершив экспорт юниверса в CMS, можно создавать, редактировать или удалять ограничение в любое время.

Можно создавать несколько ограничений, зависящих от требований запроса целевой группы пользователей.

10.4.4.1 Создание ограничения

1. Выберите "Инструменты" > "Управлять безопасностью" > "Управлять ограничениями доступа".

Откроется диалоговое окно "Управлять ограничениями доступа".



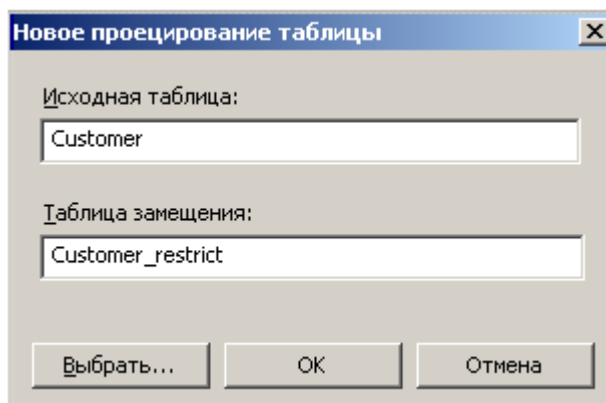
2. Нажмите кнопку [Создать](#).

На экран выводится диалоговое окно [Редактировать ограничение](#). Можно выбрать замещающее соединение для соединения юниверса или отредактировать свойства соединения с помощью кнопок, расположенных рядом с диалоговым окном [Соединение](#).

3. Выполните одно или несколько из следующих действий.

Чтобы установить...	Выполните...
новое соединение	Выберите имя соединения в окне списка "Соединения".
средство управления запроса	<ul style="list-style-type: none">• Выберите вкладку "Средства управления".• Выберите параметр запроса и введите значение для каждого параметра.
Параметры генерации SQL	<ul style="list-style-type: none">• Выберите вкладку "SQL".• Установите флажки на соответствующие параметры "Запрос", "Несколько путей" или "Декартово произведение".
ограничения доступа к объекту	<ul style="list-style-type: none">• Нажмите вкладку "Объекты".• Нажмите кнопку "Добавить". Откроется окно "Ограниченные объекты".• Нажмите кнопку "Выбрать". Отобразится обзорщик объекта.• Выберите объект для ограничения.• Нажмите кнопку "ОК", чтобы закрыть окна "Обзорщик объектов" и "Ограниченные объекты".

Чтобы установить...	Выполните...
ограничения доступа у строке	<ul style="list-style-type: none"> Откройте вкладку "Строки". Нажмите кнопку "Добавить" Нажмите кнопку "Обозреватель" рядом с окном "Таблица". Выберите имя таблицы и нажмите кнопку "ОК". Нажмите кнопку "Обзор" рядом с окном "Утверждение Where". Введите утверждение WHERE в окно определений. Или создайте утверждение WHERE, выбрав столбец, оператор и функцию в редакторе SQL. Дополнительные сведения об использовании редактора см. в разделе Использование редактора SQL-кода объединения [страница 157].
ссылка на альтернативную таблицу	<ul style="list-style-type: none"> Откройте вкладку "Сопоставление таблицы". Нажмите кнопку "Добавить" Откроется окно "Создать сопоставление таблиц". Переведите курсор в окно исходной таблицы и нажмите кнопку "Выбрать". Отобразится обозреватель таблицы. Выберите таблицу и нажмите кнопку "ОК". Переведите курсор в окно заменяемой таблицы и нажмите кнопку "Выбрать". Выберите таблицу в обозревателе таблиц и нажмите кнопку "ОК".



- Нажмите кнопку "ОК".
- В списке отобразится новое ограничение.
- Нажмите "ОК".

10.4.5 Применение ограничений доступа к юниверсу

Установите ограничения доступа к юниверсу, применив ограничение доступа к одному или нескольким пользователям или группам пользователей.

10.4.5.1 Применение ограничения к пользователю юниверса

Можно выбрать замещающее соединение для соединения юниверса.

1. Выберите "Инструменты" > "Управление безопасностью" > "Управление ограничениями доступа".

Откроется диалоговое окно "Управлять ограничениями доступа".

2. Выберите ограничение на панели доступных ограничений.
3. Выберите пользователя или группу пользователей на панели доступных пользователей или групп пользователей.

Или

Если необходимо выбрать несколько пользователей или групп пользователей, щелкните мышью несколько пользователей или групп пользователей, удерживая клавишу CTRL.

4. Нажмите кнопку "Применить".
5. Нажмите кнопку "OK".


10.4.6 Добавление группы пользователей в список доступных пользователей юниверса

Примените ограничения к группам пользователей, определенных для системы BusinessObjects. Данные пользователи выбраны в разделе "Управлять учетными записями пользователя или группы пользователей" консоли администрирования BusinessObjects. Сведения о выборе пользователя или группы пользователей для системы BusinessObjects см. руководство администратора BusinessObjects Enterprise XI 3.0.

Если необходимо применить ограничение к группе пользователей, которая отсутствует на панели доступных пользователей и групп пользователей, можно добавить данную группу пользователей в список следующим образом.

10.4.6.1 Добавление группы пользователей в области "Доступные группы и пользователи"

1. В окне "Управлять ограничениями доступа" ("Инструменты" > "Управлять безопасностью" > "Управлять ограничениями доступа") щелкните мышью значок "Добавить пользователя или группу

пользователей".  Добавить пользователя или группу

Откроется диалоговое окно "Выбрать пользователя или группу пользователей". В нем перечислены все группы пользователей, определенные в консоли администрирования BusinessObjects, которые имеют доступ в систему BusinessObjects. Если список пользователей очень велик для легкого поиска целевого пользователя или группы пользователей, можно найти список следующим образом.

- В раскрывающемся списке выберите "Имя" или "Описание".
- Введите текстовую строку в окно "Для текста", чтобы найти список текстовой строки в полях "Имя" или "Описание", выбранных ранее.
- Нажмите значок "Поиск", чтобы начать поиск.

Чтобы отфильтровать список, можно установить флажок "Пользователи или группы пользователей" для отображения только пользователей или групп пользователей в списке.

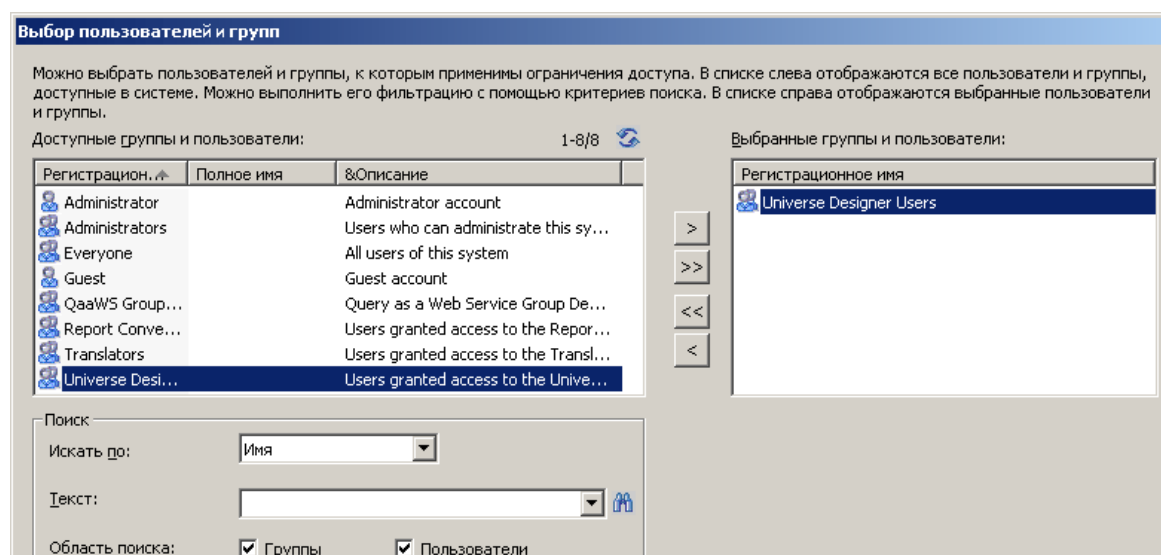
2. Выберите пользователя или группу пользователей.

Или

Щелкните мышью нескольких пользователей или групп пользователей, удерживая клавишу CTRL.

3. Нажмите на стрелку вправо.

На панели списка выбранных пользователей и групп пользователей правой части диалогового окна отобразится пользователь или группа пользователей.



4. Нажмите кнопку "ОК".

Теперь пользователь или группа пользователей отобразятся в списке доступных пользователей или групп пользователей диалогового окна "Управлять ограничениями доступа".

10.4.7 Установка приоритета группы ограничений

Можно указать, какое ограничение применить к пользователю, который принадлежит к нескольким группам пользователей, использующим универс. Например, пользователь принадлежит к двум группам: "Продажи" с ограничением для просмотра данных 5000 строк и "Маркетинг" для просмотра 10000 строк. Когда пользователь обновляет отчет, применяется ограничение, соответствующее группе самого низкого уровня. В примере, приведенном выше, если у группы "Продажи" порядок 1, а у группы "Маркетинг" порядок 2, ограничение распространяется на маркетинг (10000).

Можно упорядочить группы пользователей. Ограничение для низшей группы используется в указанном порядке.

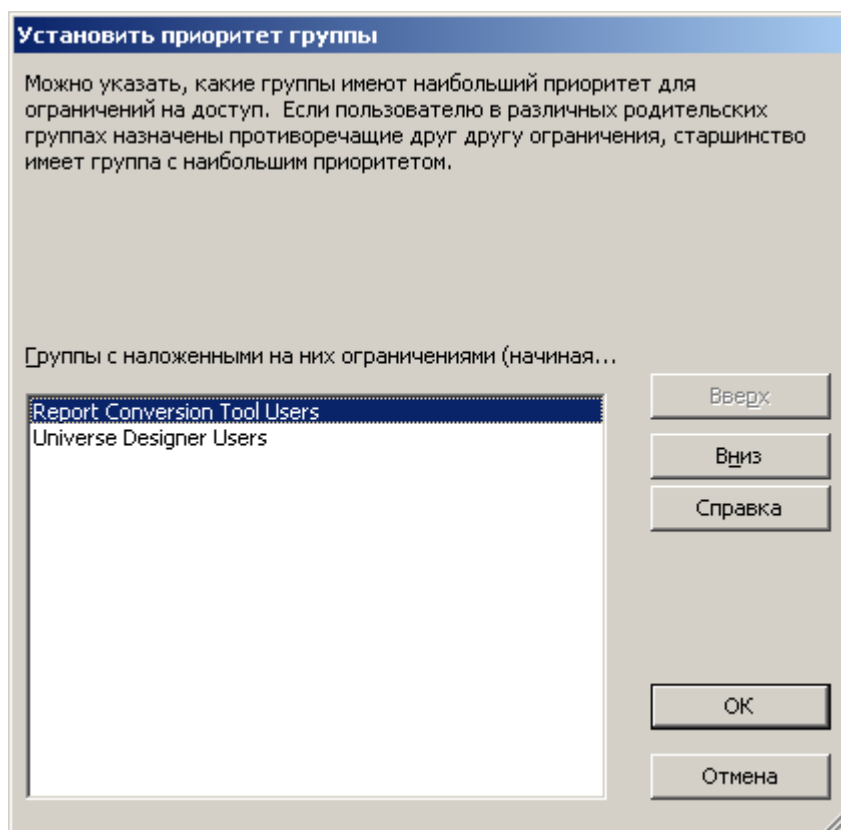
❗ Примечание

Это применимо только для исключительных ограничений, например соединение, связывание таблиц или средства управления SQL. При установке ограничений объекта на обе группы, применяться будут все.

10.4.7.1 Чтобы установить приоритет для нескольких использований ограничения

1. Выберите "Инструменты" > "Управлять безопасностью" > "Управлять ограничениями доступа".
Откроется диалоговое окно "Управлять ограничениями доступа".
2. Выберите пользователя или группу пользователей на панели доступных пользователей или групп пользователей.
3. Щелкните мышью значок "Приоритет".

Откроется окно "Установить приоритет группы".



4. Выберите пользователя или группу пользователей и нажмите кнопку "Переместить вверх" или "Переместить вниз", чтобы изменить уровень приоритета.

5. Нажмите кнопку "OK".

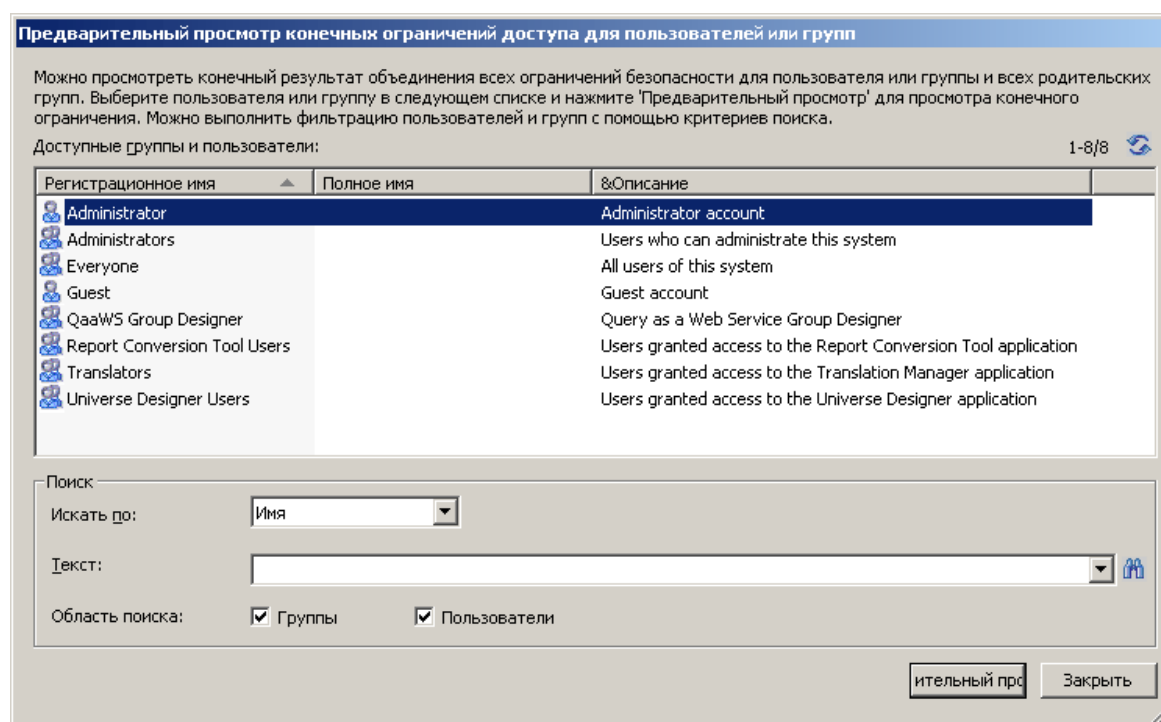
10.4.8 Просмотр ограничений безопасности пользователей и групп пользователей

Можно просматривать ограничения, применяемые для пользователя или группы пользователей.

10.4.8.1 Чтобы просмотреть все ограничения для пользователей и групп пользователей юниверса

1. Выберите "Инструменты" > "Предварительный просмотр ограничений безопасности".

Откроется диалоговое окно "Предварительный просмотр пользователей или групп пользователей".



2. Выберите имя учетной записи пользователя в списке.
3. Нажмите "Предварительный просмотр".

Отобразится ограничение безопасности, которое применяется для учетной записи пользователя. Параметры, отображенные красным – это измененные параметры. Их применяют специально для ограничений.

4. Нажмите "OK", чтобы закрыть окно.

10.5 Управление пользователями и регистрационными именами

Можно выполнить вход в средство создания юниверсов в качестве другого пользователя, а также изменить собственные данные для входа в систему. Учетные записи пользователей должны быть действительными для целевого репозитория.

Кроме того, можно выполнить вход в средство создания юниверсов в автономном режиме. Можно использовать средство создания юниверсов, создавать юниверсы, персональные и общие соединения, но нельзя получать доступ к соединениям и юниверсам в системе CMS.

10.5.1 Управление регистрационными именами

Можно выполнить вход в средство создания юниверсов в качестве другого пользователя, не выходя из текущего рабочего сеанса. Учетные записи пользователей должны быть действительными для целевого репозитория. Можно выполнять вход в качестве другого пользователя, только если известны соответствующие имя пользователя и пароль.

10.5.1.1 Чтобы войти в систему под именем другого пользователя, выполните следующие действия.

1. Выберите пункт меню "Инструменты" > "Войти как".

Любые открытые юниверсы при этом будут автоматически закрыты средством создания юниверсов. Откроется диалоговое окно "Идентификация пользователя".

2. Наберите имя пользователя в поле "Имя пользователя".
3. Введите новые имя пользователя и пароль.
4. Нажмите кнопку "ОК".

Когда выполняется вход под именем другого пользователя, автоматически предоставляются все его права, но некоторые операции могут быть запрещены, если это установлено в профиле пользователя.

10.5.2 Управление паролями

В течение сеанса можно изменять пароль, предназначенный для входа в систему при наличии прав на такое изменение у учетной записи пользователя. Однако невозможно изменить имя пользователя.

10.5.2.1 Чтобы изменить пароль

1. Выберите "Инструменты" > "Изменить пароль".
Откроется диалоговое окно "Изменить пароль".
2. В поле "Ввести старый пароль" введите существующий пароль.
3. В поле "Введите новый пароль" введите новый пароль.
4. В поле "Подтвердить новый пароль" введите новый пароль снова, чтобы подтвердить его.
5. Нажмите кнопку "ОК".

Пароль изменен.

11 Использование образцов материалов

11.1 Обзор

В данном приложении содержится подробная информация о структуре данных Клуба, построенная при помощи базы данных Microsoft Access. Большая часть примеров и изображений руководства взята из данной базы данных.

Файл базы данных, Club.mdb, можно найти в \ Samples \ <язык> \ Databases в подпапке Business Objects. Также в данной папке находится база данных eFashion.

11.2 База данных Клуба

База данных Клуба используется во многих примерах приведенных в данном руководстве.

11.2.1 Структура таблиц

База данных Клуба используется менеджером продаж Island Resorts, фиктивный бизнес специализированный на туристических путевках. Менеджер продаж выполняет маркетинговый анализ и анализ продаж, на основании сведений данной базы данных. База данных состоит из следующих таблиц:

- Age_group
- Город
- Страна
- Заказчик
- Invoice_Line
- Регион
- Region_Sline
- Reservation_Line
- Предварительные заказы
- Курорт
- Продажи
- Sales_Person
- Служба
- Service_Line

В следующих разделах описываются каждая из вышеуказанных таблиц и их столбцы.

11.2.1.1 Таблица Age_group

В таблице Age_group находятся сведения о возрастном диапазоне клиентов.

Имя столбца	Описание
age_min	нижний предел возрастного диапазона
age_max	верхний предел возрастного диапазона
age_range	возрастной диапазон клиентов

11.2.1.2 Таблица "Город"

В таблице "Город" находятся сведения о городе, в котором проживают клиенты.

Имя столбца	Описание
city_id	номер города сгенерированный системой
город	город, в котором проживают клиенты (Альбертвиль, Амстердам, Аугсбург...Версаль, Вашингтон, округ Колумбия, Иокогама)
region_id	номер региона сгенерированный системой

11.2.1.3 Таблица "Страна"

Таблица "Страна" относится к стране, в которой проживает клиент.

Имя столбца	Описание
country_id	номер страны сгенерированный системой
страна	Название страны, в которой проживает клиент (Австралия, Франция, Германия, Голландия, Япония, Великобритания, США).

11.2.1.4 Таблица "Клиент"

Таблица "Клиент" содержит сведения относящиеся к идентификации клиента, такие как имя и адрес.

Имя столбца	Описание
cust_id	номер клиента сгенерированный системой
first_name	имя клиента
last_name	фамилия клиента
возраст	возраст клиента
phone_number	телефонный номер клиента
адрес	первая строка адреса клиента
city_id	номер города сгенерированный системой
sales_id	номер продавца сгенерированный системой (лицо продавшее туристическую путевку).
sponsor_id	номер спонсора сгенерированный системой (не обязательно)

11.2.1.5 Таблица Invoice_Line

Данная таблица содержит сведения о счете, и используется для оплаты счета клиентом.

Имя столбца	Описание
inv_id	номер счета сгенерированный системой
service_id	номер услуги сгенерированный системой
дни	Число (от 3 до 15) дней проживания на курорте. В целях удобства выставления счета, срок проживания не превышает 15 дней. Свыше 15 дней, система считает оставшиеся дни, как новое проживание.
nb_guests	число гостей для которых составлен счет

11.2.1.6 Таблица "Регион"

Таблица "Регион" содержит сведения о географическом регионе, проживания клиент.

Имя столбца	Описание
region_id	номер региона сгенерированный системой
регион	географический регион проживания клиента (Бавария, Восточное побережье, Восточная Германия...Уэльс, Запад, Западная Япония)
country_id	номер страны сгенерированный системой

11.2.1.7 Таблица "Region_Sline"

Данная таблица позволяет вычислять агрегирования годового дохода с продаж в юниверсе. Поддержка агрегирования описывается в Главе 5 данного руководства.

Имя столбца	Описание
sl_id	номер сервисной линии сгенерированный системой (сведения о сервисной линии даны в таблице "Service_Line")
region_id	номер региона сгенерированный системой
sales_revenue	общая сумма годового дохода по региону.

11.2.1.8 Таблица "Reservation_Line"

Сведения относящиеся к предварительным заказам клиентов находятся в таблице "Reservation_Line".

Имя столбца	Описание
res_id	номер предварительного заказа сгенерированный системой
service_id	номер услуги сгенерированный системой
res_days	зарезервированное количество дней недели (от 1 до 7)
future_guests	число будущих гостей (от 1 до 5)

11.2.1.9 Таблица "Предварительные заказы"

Таблица "Предварительные заказы" содержит сведения о дате предварительного заказа клиента.

Имя столбца	Описание
res_id	номер предварительного заказа, сгенерированный системой
cust_id	номер клиента, сгенерированный системой
res_date	дата предварительного заказа клиента

11.2.1.10 Таблица "Курорт"

Таблица "курорт" содержит сведения о каждом курорте.

Имя столбца	Описание
resort_id	номер курорта сгенерированный системой
курорт	название курорта: Австралийский Риф, Багамский пляж, Французская Ривьера, Гавайский клуб, Королевские Карибы
country_id	номер страны сгенерированный системой

11.2.1.11 Таблица "Продажи"

Таблица "Продажи" содержит сведения о продажах.

Имя столбца	Описание
inv_id	номер счета сгенерированный системой
cust_id	номер клиента сгенерированный системой
invoice_date	дата счета

11.2.1.12 Таблица "Sales_Person"

Таблица "Sales_Person" содержит сведения о продавце Island Resorts.

Имя столбца	Описание
sales_id	номер продавца сгенерированный системой
sales_person	имя продавца (Андерсен, Барро, Бауман... Мур, Нагата, Шмидт)

11.2.1.13 Таблица "Услуги"

Таблица "Услуги" содержит сведения о ценах и типах услуг, доступных для данного курорта.

Имя столбца	Описание
service_id	номер услуги сгенерированный системой
услуга	услуги курорта (см. результат запроса)
sl_id	номер сервисной линии сгенерированный системой (сведения о сервисной линии представлены в следующей таблице)
стоимость	стоимость услуг

11.2.1.14 Таблица Service_Line

Таблица "Service_Line" содержит сведения о сервисной линии курортов. Сервисная линия означает категорию под которую подходят услуги.



Имя столбца	Описание
sl_id	номер сервисной линии сгенерированный системой
service_line	Сервисная линия включает в себя: размещение, питание и напитки, развлечения
resort_id	номер курорта сгенерированный системой (значения от 1 до 5)

Важные положения об отказе от ответственности в отношении правовых вопросов

Гиперссылки

Некоторые ссылки обозначаются значком и/или текстом, отображаемым при наведении мыши. Эти ссылки обеспечивают доступ к дополнительной информации.

Подробнее о значках:

- Ссылки со значком  Вы собираетесь перейти на сайт, размещенный не на сервере SAP. Используя такие ссылки, вы соглашаетесь (если иное не оговорено особо в соглашениях с SAP) со следующим:
 - Сайт по ссылке не содержит документацию SAP. Не разрешается подавать рекламации в отношении любых продуктов SAP на основе содержащейся на таком сайте информации.
 - SAP не выражает ни согласия, ни несогласия с информацией, содержащейся на сайте по ссылке, а также не гарантирует ее доступность и правильность. SAP не несет ответственности за любой ущерб, вызванный использованием такой информации, за исключением тех случаев, когда такой ущерб вызван намеренными нарушениями или халатностью со стороны компании SAP.
- Ссылки со значком  Вы закрываете документацию по определенному продукту или сервису SAP и переходите на веб-сайт, расположенный на сервере SAP. Используя такие ссылки, вы соглашаетесь (если иное не оговорено особо в соглашениях с SAP) с тем, что не разрешается подавать рекламации в отношении любых продуктов SAP на основе содержащейся на таком сайте информации.

Видео-ролики, размещенные на внешних платформах

Некоторые видео-ролики могут указывать на сторонние платформы размещения видео-роликов. SAP не может гарантировать в будущем доступность видео-роликов, сохраненных на этих платформах. Кроме того, любые рекламные объявления или другой контент, размещенные на этих платформах (например, предлагаемые видео-ролики или ссылки на другие видео-ролики, размещенные на одном сайте), не входят в сферу управления или ответственности SAP.

Бета-версии и другие экспериментальные функции

Экспериментальные функции не являются частью официально поставляемого SAP объема, гарантируемого для будущих версий. Это означает, что экспериментальные функции могут быть изменены компанией SAP в любое время и по любой причине без предварительного уведомления. Экспериментальные функции не предназначены для продуктивного использования. Не разрешается демонстрировать, тестировать, проверять, анализировать или иначе использовать экспериментальные функции в фактической операционной среде либо с использованием данных, для которых не выполнено резервное копирование. Экспериментальные функции предназначены для получения обратной связи, которая позволяет нашим клиентам и партнерам влиять на разработку будущих продуктов. Предоставляя обратную связь (например, в SAP Community), вы соглашаетесь с тем, что права на интеллектуальную собственность относительно ваших отзывов и производных работ останутся в исключительной собственностью SAP.

Пример кода

Примером кода является любой код и/или фрагменты кода программного обеспечения. Они не предназначены для продуктивного использования. Этот код предназначен только для пояснения и иллюстрирования синтаксиса и правил составления текста программ. SAP не гарантирует правильность и полноту примеров кода. SAP не несет ответственности за любые ошибки и ущерб, вызванные использованием примеров кода, за исключением тех случаев, когда такой ущерб вызван намеренными нарушениями или халатностью со стороны компании SAP.

Язык, свободный от предрассудков

SAP поддерживает культуру многообразия и инклюзивности. Когда это возможно, в нашей документации мы используем безоценочный язык для обозначения людей из любой культуры или этнической группы, любого пола и уровня способностей.

© SAP SE или аффилированная компания SAP, 2024. Все права защищены.

Полное или частичное воспроизведение или передача в какой-либо форме и в каких-либо целях настоящей публикации без явного образом выраженного разрешения SAP SE или аффилированной компании SAP запрещены. Информация, содержащаяся в настоящей публикации, может быть изменена без предварительного уведомления.

Некоторые программные продукты, предлагаемые на рынке компанией SAP SE и ее дистрибьюторами, содержат компоненты программного обеспечения, исключительными правами в отношении которых обладают иные поставщики программного обеспечения. Возможны различные варианты спецификаций продуктов для разных стран.

Материалы предоставлены компанией SAP SE и ее аффилированной компанией исключительно в информационных целях, без предоставления каких-либо гарантий. Компания SAP или ее аффилированные компании не несут ответственности за ошибки или пропуски в настоящих материалах. Гарантии, если таковые предоставляются, в отношении продуктов и услуг компании SAP или ее аффилированной компании содержатся исключительно в документах, которые прилагаются к соответствующим продуктам и услугам. Ничто, изложенное в настоящем документе, не должно трактоваться как предоставление дополнительных гарантий.

SAP, а также упомянутые здесь продукты и услуги SAP, как и соответствующие логотипы, являются товарными знаками или зарегистрированными товарными знаками SAP SE (или аффилированной компании SAP) на территории Германии и других стран. Все иные названия продуктов и услуг являются товарными знаками соответствующих компаний.

Для получения дополнительной информации и уведомлений о товарных знаках см. <https://www.sap.com/cis/about/legal/trademark.html>.