



Universe 设计工具用户指南

■ SAP BusinessObjects Business Intelligence platform 4.1

2013-06-29

版权所有

© 2013 SAP AG 股份公司或其关联公司版权所有，保留所有权利。未经SAP股份公司明确许可，不得以任何形式或为任何目的复制或传播本文的任何内容。本文包含的信息如有更改，恕不另行事先通知。由SAP股份公司及其分销商营销的部分软件产品包含其它软件供应商的专有软件组件。各国的产品规格可能不同。上述资料由SAP股份公司及其关联公司（统称“SAP集团”）提供，仅供参考，不构成任何形式的陈述或保证，其中如若存在任何错误或疏漏，SAP集团概不负责。与SAP集团产品和服务相关的保证仅限于该等产品和服务随附的保证声明（若有）中明确提出之保证。本文中的任何信息均不构成额外保证。SAP和本文提及的其它SAP产品和服务及其各自标识均为SAP股份公司在德国和其他国家的商标或注册商标。如欲了解更多商标信息和声明，请访问：
<http://www.sap.com/china/about/company/legal/copyright/index.epx#trademark>。

2013-06-29

目录

第 1 章	Universe 设计工具简介.....	13
1.1	文档历史记录: Universe 设计工具..	13
1.2	概述..	13
1.3	Universe 设计工具和 Universe 基础知识..	14
1.3.1	什么是 Universe? ..	14
1.3.2	Universe 有何作用? ..	14
1.3.3	Universe 包含什么? ..	15
1.3.4	关于 Universe 窗口..	17
1.3.5	Universe 设计工具安装根路径..	18
1.4	使用 Universe 设计工具创建 Universe 的方法..	18
1.4.1	对象如何生成 SQL? ..	19
1.4.2	支持哪些类型的数据库模式? ..	19
1.4.3	如何使用 Universe? ..	20
1.5	谁是 Universe 的设计员? ..	21
1.5.1	必需的技能 and 知识..	21
1.5.2	Universe 设计员的任务是什么? ..	22
1.6	Universe 的基本创建步骤..	22
1.7	Universe 开发过程简介..	23
1.7.1	Universe 设计方法..	23
1.7.2	Universe 开发周期..	25
1.7.3	优化 Universe 规划和实施时间..	26
1.8	多语言 Universe..	27
1.8.1	多语言 Universe..	27
1.8.2	语言和区域设置的定义..	28
1.8.3	不同的区域设置..	29
1.8.4	设置 Universe 设计工具用户界面的产品语言..	29
1.8.5	使用多语言 Universe..	30
1.8.6	确定链接的 Universe 中的回退区域设置..	30
1.8.7	翻译管理工具..	30
1.8.8	多语言数据..	31
1.9	Universe 设计工具示例材料..	31
1.9.1	演示数据库..	31
1.9.2	演示 Universe..	32
1.10	使用信息设计工具处理 Universe..	32

第 2 章	执行基本操作.....	33
2.1	概述..	33
2.2	启动 Universe 设计工具..	33
2.2.1	启动 Universe 设计工具..	34
2.2.2	使用“快速设计”向导 ..	35
2.3	通过 Designer XI R3 使用 XI R2 连接和 Universe..	35
2.4	使用“快速设计”向导创建基本 Universe..	36
2.4.1	为何使用快速设计向导? ..	36
2.4.2	使用“快速设计向导” ..	36
2.4.3	对利用快速设计向导创建的 Universe 进行后续处理..	44
2.5	导入 Universe..	44
2.5.1	从资源库导入 Universe..	44
2.5.2	打开操作和导入操作有什么不同? ..	45
2.6	打开 Universe..	45
2.7	导出 Universe..	46
2.7.1	Universe 在资源库文件系统上是如何排列的? ..	47
2.7.2	将 Universe 导出到资源库..	47
2.7.3	导出操作和保存操作有什么不同? ..	48
2.8	保存 Universe..	48
2.8.1	作为标识符的 Universe 文件名..	49
2.8.2	保存 Universe..	49
2.8.3	将 Universe 定义另存为 PDF..	49
2.9	关闭 Universe..	51
2.10	与多个设计员一起工作..	51
2.10.1	锁定 Universe..	51
2.10.2	修订版号..	52
2.11	使用 Universe 设计工具用户界面..	52
2.11.1	用户界面的主要构成项..	52
2.11.2	Universe 设计工具用户界面..	53
2.11.3	使用窗口 ..	54
2.11.4	使用工具栏..	54
2.11.5	在 Universe 设计工具中执行操作..	56
2.12	使用“查找并替换” ..	58
2.12.1	使用“查找” ..	58
2.12.2	使用“快速查找” ..	61
2.13	组织表显示..	61
2.13.1	表是如何表示的? ..	62
2.13.2	处理表..	62
2.13.3	使用列表模式..	63
2.13.4	自动排列表..	65
2.13.5	更改表的显示..	65
2.14	选择模式显示选项..	67
2.14.1	为“结构”窗格显示设置图形选项..	68
2.14.2	查看表和列值..	69

2.14.3	查看数据库表中的行数..72
2.15	打印 Universe..75
2.15.1	设置打印选项..76
第 3 章	创建 Universe 并设置 Universe 参数.....79
3.1	什么是 Universe 参数? ..79
3.2	创建新的 Universe..80
3.2.1	从头创建新的 Universe..80
3.3	查看和输入摘要信息..82
3.4	设置 Universe 参数..83
3.4.1	标识 Universe ..83
3.4.2	定义和编辑连接..85
3.4.3	设置 Universe 摘要参数..93
3.4.4	查看和输入摘要信息..93
3.4.5	选择策略..94
3.4.6	指明资源控制..98
3.4.7	哪些系统资源选项可用? ..98
3.4.8	输入资源控制信息..99
3.4.9	限制生成多个 SQL 语句的查询的执行时间..99
3.4.10	指明 SQL 限制..100
3.4.11	为链接的 Universe 指明选项..102
3.4.12	设置 SQL 生成参数..103
3.4.13	关于 SQL 生成参数..105
3.4.14	在用户界面中设置的 SQL 参数..106
3.4.15	在 PRM 文件中设置的 SQL 参数..126
第 4 章	创建包含表和联接的模式.....143
4.1	概述..143
4.2	什么是模式? ..143
4.2.1	模式设计是成功的 Universe 的基础..144
4.2.2	模式设计和 Universe 创建过程..144
4.2.3	模式设计有哪些阶段? ..145
4.3	插入表..145
4.3.1	使用表浏览器..145
4.3.2	排列“结构”窗格中的表..149
4.4	使用派生表..150
4.4.1	添加、编辑和删除派生表..150
4.5	嵌套派生表..153
4.5.1	使用派生表编辑器..153
4.5.2	创建嵌套派生表..154
4.5.3	重命名嵌套的派生表..155
4.6	使用包含输入列的表..155
4.6.1	定义硬编码的值列表..156
4.6.2	定义供用户输入或选择的值列表..156
4.7	定义联接..157

4.7.1	什么是联接? .. 157
4.7.2	为何要在模式中使用联接? .. 157
4.7.3	联接推导出什么 SQL? .. 158
4.7.4	无需联接哪些表? .. 158
4.7.5	联接主键和外键.. 159
4.7.6	了解联接的基数.. 160
4.7.7	创建联接.. 160
4.7.8	联接属性.. 165
4.7.9	编辑联接.. 168
4.7.10	对 Universe 中联接的 ANSI 92 支持.. 172
4.7.11	删除联接.. 175
4.8	定义特定的联接类型.. 176
4.8.1	创建相等联接.. 176
4.8.2	Theta 联接.. 180
4.8.3	外部联接.. 183
4.8.4	快捷联接.. 187
4.8.5	自我限制的连接.. 188
4.9	使用基数.. 190
4.9.1	基数在 Universe 设计工具中的使用方法.. 191
4.9.2	手动设置基数.. 193
4.10	检查 Universe.. 200
4.10.1	自动检查 Universe 完整性.. 201

第 5 章 解决模式中的联接问题..... 207

5.1	概述.. 207
5.2	什么是联接路径问题? .. 207
5.2.1	什么是查找表.. 208
5.2.2	什么是事实表.. 208
5.2.3	什么类型的联接路径返回不正确的结果? .. 208
5.2.4	检测 and 解决联接问题.. 209
5.3	定义别名.. 210
5.3.1	如何在模式中使用别名? .. 210
5.3.2	创建别名.. 211
5.4	定义上下文.. 214
5.4.1	如何在模式中使用上下文? .. 214
5.4.2	创建上下文.. 216
5.4.3	编辑上下文.. 218
5.4.4	删除上下文.. 220
5.4.5	更新上下文.. 220
5.4.6	阻碍上下文检测的联接路径.. 221
5.4.7	上下文如何影响查询? .. 222
5.5	解决环路.. 225
5.5.1	什么是环路? .. 226
5.5.2	环路如何影响查询? .. 227
5.5.3	直观地标识环路.. 234

5.5.4	自动标识和解决环路..234
5.5.5	用于检测 and 解决环路的工具功能..235
5.5.6	解决环路的示例..243
5.6	解决断层陷阱..251
5.6.1	什么是断层陷阱? ..251
5.6.2	断层陷阱如何使结果膨胀? ..253
5.6.3	检测断层陷阱..254
5.6.4	解决断层陷阱..254
5.7	解决扇形陷阱..257
5.7.1	什么是扇形陷阱? ..257
5.7.2	如何检测扇形陷阱? ..259
5.7.3	如何解决扇形陷阱? ..260
5.8	以图形方式检测联接问题..263
5.8.1	可能的断层陷阱..264
5.8.2	可能的扇形陷阱..265
5.9	检查 Universe..266
5.9.1	自动检查 Universe 完整性..266
5.9.2	手动检查 Universe 完整性..267
5.9.3	刷新 Universe 结构..271

第 6 章 创建 Universe.....273

6.1	概述..273
6.2	Universe 构建简介..273
6.2.1	什么是对象? ..274
6.2.2	Universe 中使用什么类型的对象? ..275
6.2.3	什么是类? ..275
6.2.4	使用类和对象..275
6.3	使用 “Universe” 窗格..276
6.3.1	显示类和对象或条件..276
6.4	有关类、对象和条件的基本操作..277
6.4.1	剪切、复制、粘贴..277
6.4.2	移动类、对象或条件..277
6.4.3	显示或隐藏类、对象和条件..278
6.5	定义类..279
6.5.1	创建类..279
6.5.2	类属性..281
6.5.3	修改类..282
6.5.4	使用子类..282
6.6	定义对象..283
6.6.1	创建对象..283
6.6.2	对象属性..285
6.6.3	修改对象..286
6.6.4	对象定义..286
6.6.5	属性..289
6.6.6	高级..290

- 6.6.7 定义索引感知..292
- 6.6.8 源信息..296
- 6.6.9 使用 SQL 编辑器来定义对象..297
- 6.6.10 定义对象格式..300
- 6.6.11 查看对象定义中使用的表..301
- 6.6.12 定义维..302
- 6.6.13 定义明细..302
- 6.6.14 定义度量..303
- 6.6.15 为对象定义限制..309
- 6.6.16 定义条件对象..314
- 6.6.17 使用自我限制的联接来应用限制..320
- 6.6.18 通过推导多个表来应用限制..320
- 6.6.19 拼接对象..322
- 6.7 定义层次结构..324
 - 6.7.1 什么是多维分析?..324
 - 6.7.2 标识层次结构的方法..325
 - 6.7.3 设置层次结构..326
- 6.8 使用层次结构的级联值列表..329
 - 6.8.1 创建级联值列表..330
 - 6.9 使用值列表..332
 - 6.9.1 如何使用值列表?..333
 - 6.9.2 定义如何将值列表用于对象..334
 - 6.9.3 值列表属性和选项..335
 - 6.9.4 编辑值列表..339
 - 6.9.5 导出值列表..343
 - 6.9.6 刷新值列表中的值..346
 - 6.9.7 使用个人数据文件中的数据..346
 - 6.9.8 管理 Universe 中的值列表..348
 - 6.9.9 优化和定制 LOV 文件..349
 - 6.10 链接 Universe..350
 - 6.10.1 什么是链接的 Universe?..350
 - 6.10.2 链接 Universe 的不同方法..352
 - 6.10.3 链接 Universe 的优点..353
 - 6.10.4 链接 Universe 的要求..354
 - 6.10.5 链接 Universe 时的限制..354
 - 6.10.6 在两个 Universe 之间创建链接..355
 - 6.10.7 编辑派生 Universe..357
 - 6.10.8 删除链接..358
 - 6.10.9 重新放置核心 Universe..359
 - 6.10.10 派生 Universe 和值列表..359
 - 6.10.11 按核心 Universe 的顺序显示对象..360
 - 6.11 在一个 Universe 中包含另一个 Universe..360
 - 6.11.1 将核心 Universe 复制到派生 Universe 中..360
 - 6.12 创建存储过程 Universe..361
 - 6.12.1 Java Bean Universe 中的存储过程..362
 - 6.12.2 根据存储过程创建 Universe..363

6.13	测试 Universe..367	
6.13.1	在“查询面板”中测试对象..367	
6.13.2	测试 Universe 的完整性 ..367	
6.13.3	使用 Web Intelligence 测试 Universe..368	
 第 7 章	 优化 Universe.....	 369
7.1	概述..369	
7.2	使用聚合表..369	
7.2.1	什么是聚合感知? ..370	
7.2.2	将聚合感知应用于数据仓库..370	
7.2.3	设置聚合感知..370	
7.2.4	构建对象..372	
7.2.5	确定聚合对象的所有组合..372	
7.2.6	按聚合级别顺序排列对象..372	
7.2.7	使用 @Aggregate_Aware 函数定义聚合对象..373	
7.2.8	指定不兼容的对象..375	
7.2.9	指定不兼容的对象..378	
7.2.10	解决包含聚合表的环路..380	
7.2.11	测试聚合感知..382	
7.3	在对象的 SQL 中使用 @ 函数..383	
7.3.1	在对象中插入 @函数..384	
7.3.2	@Aggregate_Aware..385	
7.3.3	@Prompt..386	
7.3.4	@Script..399	
7.3.5	@Select..400	
7.3.6	@Variable..402	
7.3.7	@Where..408	
7.4	使用外部策略定制 Universe 创建..410	
7.4.1	将外部策略迁移到 Universe 设计工具..411	
7.4.2	外部策略概述..411	
7.4.3	什么是外部策略? ..412	
7.4.4	为外部策略创建帮助文本..413	
7.4.5	验证是否声明了外部策略文件..415	
7.4.6	使用示例外部策略..416	
7.4.7	策略文件 (STG) 具有什么样的结构? ..416	
7.4.8	策略的输出格式..418	
7.4.9	创建外部策略..422	
7.4.10	创建数据的文本文件..424	
7.4.11	在 Universe 设计工具中应用外部策略..424	
7.5	使用分析函数..425	
7.5.1	什么是分析函数? ..426	
7.5.2	使用分析函数有哪些优点? ..426	
7.5.3	支持哪些分析函数族? ..426	
7.5.4	如何在 Universe 设计工具中使用分析函数? ..427	
7.5.5	IBM DB2 UDB 和 Oracle..428	

7.5.6	RedBrick (RISQL 函数) .. 432
7.5.7	Teradata (OLAP 函数) .. 435
7.5.8	在 Select 语句中自动插入语法.. 438
7.6	使用 SQL Prefix 函数.. 439
7.6.1	使用 BEGIN_SQL Universe 参数为 SQL 语句添加前缀.. 439
7.7	优化数组提取参数.. 440
7.7.1	修改数组提取参数.. 440
7.8	分配表的权重.. 441
7.8.1	修改 PRM 文件以分配表的权重.. 441
7.9	修改为某个表返回的行数.. 442
7.9.1	修改返回的行数.. 442
7.10	使用快捷联接.. 442

第 8 章 使用 OLAP Universe..... 444

8.1	关于 OLAP Universe.. 444
8.1.1	什么是 OLAP Universe? .. 444
8.1.2	可用于创建 Universe 的 OLAP 数据源 .. 445
8.2	定义 OLAP 数据源的连接.. 451
8.2.1	关于 OLAP 数据源的连接.. 451
8.2.2	启动“新建连接”向导.. 452
8.2.3	为 OLAP 连接选择数据库中间件.. 453
8.2.4	SAP BW OLAP 连接的登录参数.. 454
8.2.5	MSAS OLAP 连接的登录参数 .. 455
8.2.6	为 Essbase 连接定义登录参数.. 457
8.2.7	为 OLAP 连接选择源多维数据集或查询.. 457
8.2.8	定义 OLAP 连接的配置参数.. 458
8.2.9	为 Essbase 连接定义定制参数.. 459
8.3	自定义 OLAP Universe.. 460
8.3.1	使用附加参数创建 OLAP Universe.. 460
8.3.2	针对 OLAP Universe 定义 OLAP 选项.. 461
8.3.3	定义 OLAP Universe 中的对象.. 462
8.3.4	支持 OLAP Universe 的 Universe 设计工具功能.. 462
8.3.5	投影函数“委派给数据库”.. 463
8.3.6	为 OLAP Universe 设置委派度量.. 465
8.3.7	为度量设置聚合投影.. 466
8.3.8	OLAP Universe 中的计算所得度量.. 467
8.3.9	关于多维数据集查询的 MDX 函数.. 468
8.3.10	过滤器和 WHERE 语句的 XML 语法.. 469
8.3.11	OLAP Universe 中的预定义条件.. 470
8.3.12	OLAP Universe 中的可选提示.. 475
8.3.13	提高 SAP NetWeaver BW Universe 上某些查询的性能.. 475
8.4	OLAP Universe 生命周期管理.. 476
8.4.1	关于 OLAP Universe 生命周期管理.. 476
8.4.2	概述: Universe 对象状态和 OLAP 对象状态之间的关系 .. 477
8.4.3	刷新 OLAP Universe.. 480

8.4.4	为 OLAP Universe 重新生成级别 00..483
8.4.5	将级别 L00 重命名为“所有”..483
8.4.6	替换 OLAP Universe 级别前缀..483
8.4.7	正在同步 Universe 和 OLAP 多维数据集..484
8.4.8	在 OLAP Universe 更新中管理维的方式..484
8.4.9	在 OLAP Universe 更新中管理层次结构或特性的方式 ..489
8.4.10	在 OLAP Universe 更新中管理级别的方式 ..496
8.4.11	在 OLAP Universe 更新中管理 SAP 变量的方式 ..500
8.4.12	在 OLAP Universe 更新中管理关键值或度量的方式 ..504
8.4.13	在 OLAP Universe 更新中管理 SAP 关键日期的方式 ..508
8.5	不同 OLAP 多维数据集映射到 Universe 的方式..510
8.5.1	如何在 Universe 中映射和使用 SAP NetWeaver BW 对象..511
8.5.2	如何将 Essbase 多维数据集映射到 Universe 元素..519
8.5.3	将 MSAS 多维数据集映射到 Universe 元素的方法 ..520
第 9 章	使用来自元数据源的 Universe.....522
9.1	有关从元数据源生成 Universe 的介绍..522
9.2	概述..522
9.3	Universe 创建概述..523
9.4	选择元数据源..523
9.5	依据 XML 源创建 Universe..524
9.5.1	XML 元数据源..524
9.5.2	依据 XML 元数据源生成 Universe ..525
9.5.3	选择连接和 Universe 选项..526
9.5.4	依据 XML 元数据源更新 Universe..527
9.6	将 Universe 导出到 DB2CV..528
9.6.1	针对导出的 Universe 先决条件..528
9.6.2	标识 Universe 元数据..529
9.6.3	将 Universe 导出为 DB2CV XML 文件..530
9.6.4	Universe 到 DB2CV 元数据的映射..530
9.6.5	映射特定 SQL 表达式..534
9.7	Oracle 分析工作区..535
9.7.1	如何从 OLAP 多维数据集中生成 Universe? ..536
9.7.2	将 Oracle OLAP 结构映射到 Universe 组成部分..537
9.7.3	分析关系视图..537
9.7.4	Universe 中的快捷联接有何具体用途? ..538
9.7.5	Oracle OLAP 结构如何映射到 Universe 组成部分? ..540
9.7.6	创建视图并生成 Universe..544
9.7.7	用于从 Oracle 分析工作区创建 Universe 和视图的选项..544
9.7.8	创建视图并生成 Universe..545
9.7.9	仅从 Oracle 分析工作区创建视图..546
9.7.10	从现有的视图中生成 Universe..547
9.7.11	从现有的视图中生成 Universe..547

第 10 章	部署 Universe.....	548
10.1	概述..	548
10.2	Universe 的部署方法..	548
10.2.1	在资源库中标识 Universe..	548
10.3	使所有用户都能访问 Universe..	549
10.4	针对 Universe 设置访问限制..	550
10.4.1	什么是限制? ..	550
10.4.2	可以在 Universe 中应用什么限制? ..	551
10.4.3	如何管理访问限制? ..	552
10.4.4	创建限制..	553
10.4.5	应用 Universe 访问限制..	555
10.4.6	将用户组添加到 Universe 的可用用户列表..	556
10.4.7	设置限制组优先级..	557
10.4.8	查看用户和组的安全限制..	559
10.5	管理用户和登录..	559
10.5.1	管理登录..	560
10.5.2	管理密码..	561
第 11 章	使用示例资料.....	562
11.1	使用示例资料 — 概览..	562
11.2	Club 数据库..	562
11.2.1	表的结构..	562
附录 A	更多信息.....	571
索引.....		573

第 1 章 Universe 设计工具简介

1.1 文档历史记录：Universe 设计工具

下表概述了最重要的文档更改。

版本	日期	说明
SAP BusinessObjects Universe 设计工具 4.0	2010 年 11 月 30 日	本文档的第一个版本。 Universe Designer 更名为 “Universe 设计工具”
SAP BusinessObjects Universe 设计工具 4.0 Service Pack 1	2011 年 2 月 25 日	
SAP BusinessObjects Universe 设计工具 4.0 Service Pack 2	2011 年 6 月 15 日	连接对象还有一个由管理员定义的安全权限，叫做 “本地下载连接”。
SAP BusinessObjects Universe 设计工具 4.0 Feature Pack 3	2012 年 2 月 20 日	

1.2 概述

本章简要介绍了 Universe 设计工具这一用于构建 Universe 的工具。它说明了 Universe，Universe 包含什么，如何创建 Universe 以及 Universe 在您的业务环境中具有的作用。

说明了典型的 Universe 开发周期，并提供了最佳设计做法的建议。也说明了随本版本附带的演示数据库和 Universe。

本章介绍了 Universe 设计工具、开发过程以及 Universe 的不同可用语言。涵盖了以下主题：

相关主题

- 第 14 页上的[“Universe 设计工具和 Universe 基础知识”](#)
- 第 18 页上的[“使用 Universe 设计工具创建 Universe 的方法”](#)
- 第 21 页上的[“谁是 Universe 的设计员？”](#)
- 第 23 页上的[“Universe 开发过程简介”](#)
- 第 27 页上的[“多语言 Universe”](#)

1.3 Universe 设计工具和 Universe 基础知识

Business Objects Universe 设计工具是一个软件工具，可为 Web Intelligence 和 Desktop Intelligence 用户创建 Universe。

1.3.1 什么是 Universe？

Universe 是一个包含以下内容的文件：

- 一个或多个数据库中间件的连接参数。
- 称为对象的 SQL 结构，映射到数据库中的实际 SQL 结构，如列、表和数据库函数。对象按类分组。Web Intelligence 用户既可以看到对象，也可以看到类。
- 在数据库中使用的表和联接的模式。对象是通过您包括在模式中的数据库结构来构建的。模式只能由 Universe 设计工具用户使用。Web Intelligence 和 Desktop Intelligence 用户看不到该模式。

Web Intelligence 用户连接到 Universe，并依据数据库运行查询。他们可以进行数据分析，并使用 Universe 中的对象创建报表，而无需看到或了解有关数据库中的基本数据结构的任何信息。

1.3.2 Universe 有何作用？

Universe 的作用是为 Web Intelligence 的非技术用户提供一个易于使用 and 理解的接口，使他们能依据数据库运行查询，以创建报表和执行数据分析。

Universe 设计员使用 Universe 设计工具来创建表示数据库结构的对象（例如，列和数据库函数），用户需要访问和查询这些对象，以获得满足其业务需求所需的信息。

您在 Universe 中创建的对象必须与最终用户的业务环境和词汇相关。这些对象的作用是将以业务为焦点的前端呈现为数据库中的 SQL 结构。

1.3.3 Universe 包含什么？

Universe 包含以下结构：

- 类
- 对象

1.3.3.1 类

类是 Universe 中的对象的逻辑分组。它表示对象的类别。类的名称应指明类包含的对象的类别。可以按层次将类划分为子类。

1.3.3.2 对象

对象是映射到数据库中的数据或数据派生的命名元素。应根据目标用户组的业务词汇指定对象的名称。例如，由产品经理使用的 Universe 中所用的对象可以是“Product（产品）”、“Life Cycle（生命周期）”或“Release Date（发布日期）”。财务分析员使用的 Universe 可以包含诸如“Profit Margin（利润）”或“Return on Investment（投资回报）”等对象。

1.3.3.3 对象类型

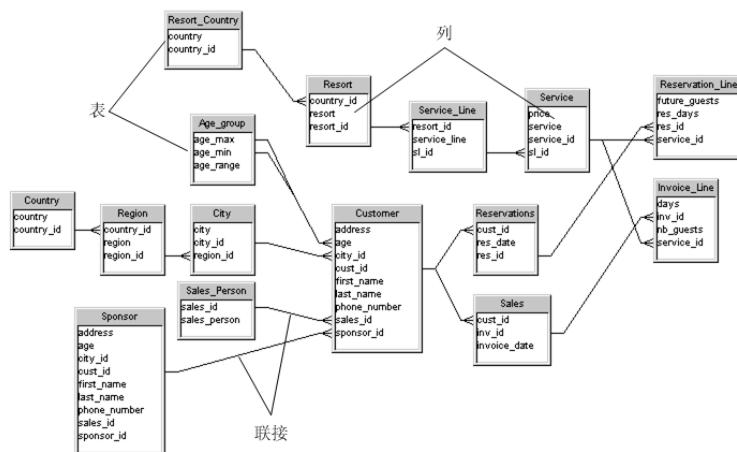
在 Universe 设计工具中，对象被限定为以下三种类型之一：维、明细或度量。

对象类型	说明
维	用于分析的参数。维通常与诸如地理、产品或时间的层次相关。例如：姓氏和 City_Id
明细	提供维的说明，但并不是分析的焦点。例如：电话号码
度量	传递用于确定维对象数量的数值信息。例如：“Sales Revenue”（销售收入）

1.3.3.4 对象推导出模式中显示的 SQL 结构

Web Intelligence 用户在 Universe 中看到的对象推导出您已插入到数据库模式中的 SQL 结构。作为 Universe 设计员，您根据返回数据所需的表和联接创建此模式（用户需要这些数据进行分析 and 创建报表）。

此模式是 Universe 文件的一部分，但只在 Universe 设计工具中可见及可访问。在 Universe 窗口的“结构”窗格中创建模式。以下显示了 Beach.unv 这一示例 Universe 的模式。

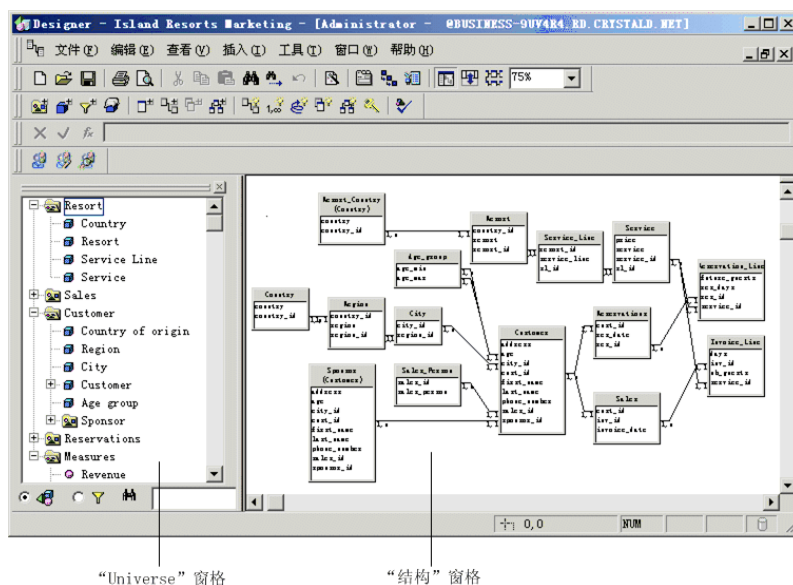


1.3.3.5 在 Universe 中如何表示对象？

在“Universe”窗格中，对象显示为树形资源管理器视图中的节点。您使用对象资源管理器来创建、删除、复制、查看和移动类及对象。

1.3.4 关于 Universe 窗口

以下显示了 Universe 设计工具中的“Universe 窗口”。它同时包含“Universe”窗格（在 Web Intelligence 中也可见）和“结构”窗格（仅在 Universe 设计工具中可见）。



1.3.5 Universe 设计工具安装根路径

在本指南中，变量 \$INSTALLDIR 是 Universe 设计工具和 Web Intelligence 使用的数据库访问文件的安装根路径。这是带有操作系统子目录的 Business Objects 安装路径，其中包含 Universe 设计工具可执行文件和数据访问驱动程序。

在 Windows 下，\$INSTALLDIR = \\...\\Business Objects\\BusinessObjects Enterprise 12.0\\win32_x86。

例如，C:\\Program Files\\Business Objects\\BusinessObjects Enterprise 12.0\\win32_x86。

1.4 使用 Universe 设计工具创建 Universe 的方法

Universe 设计工具提供了一个允许连接到数据库中间件的连接向导。可以使用该工具创建多个连接，但只能为每个 Universe 定义一个连接。此数据库连接随 Universe 一起保存。

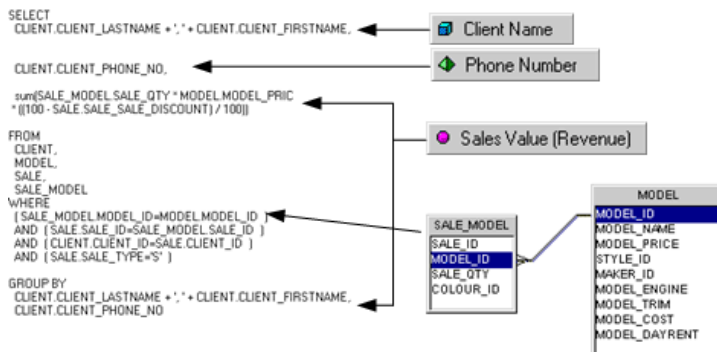
Universe 设计工具提供了用于选择和查看数据库表的图形界面。在模式图中，以表符号来表示数据库表。可以使用此界面操作表、创建链接表的联接、创建别名表、上下文和解决模式中的循环。Web Intelligence 用户看不到此模式。

Universe 设计工具提供了对象资源管理器视图，即“树形资源管理器”。使用树形资源管理器可以创建映射到在模式视图中显示的列和 SQL 结构的对象。Web Intelligence 用户可以处理这些对象以依据数据库运行查询。

使用 Universe 设计工具可以通过导入 Universe 并将其导出到中央管理系统（CMS）资源库来分发 Universe。

1.4.1 对象如何生成 SQL？

Web Intelligence 用户通过将对象拖到“查询”工作区中来创建查询。每个对象的定义推导出 SELECT 语句。运行查询时，会依据目标数据库运行 SELECT 语句和 WHERE 子句（对于所有对象均为可选）。

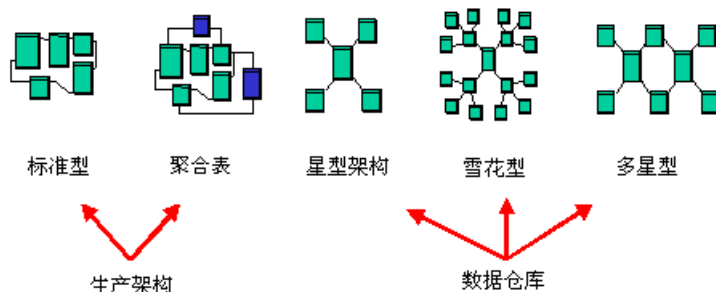


当用户选择在“查询”工作区中随度量对象一起包含维和/或明细对象时，包含这些维和明细对象的内容的 GROUP BY 子句被自动添加到 SELECT 语句中。

对于包含在 FROM 子句中的表和包含在 WHERE 子句中的联接，它们是通过用户在“结构”窗格中建立的表模式推导出的。

1.4.2 支持哪些类型的数据库模式？

Universe 设计工具可以支持大多数类型的数据库模式，包括以下所示的所有模式。在使用 Universe 设计工具之前，无需重新定义或优化数据库。



1.4.3 如何使用 Universe?

Universe 由 Web Intelligence 用户使用。Universe 存储在中央管理系统 (CMS) 资源库中。最终用户可通过 Web 浏览器连接到 Universe。

与数据库的连接在 Universe 中定义，因此，通过连接到 Universe，最终用户自动就能访问数据。而对数据的访问受到 Universe 中可用的对象限制。这些对象是由作为 Universe 设计员的您根据定义的用户组的用户需求配置文件创建的。

1.4.3.1 重新表示目标数据需求

Universe 可以表示任何特定应用程序、系统或用户组的数据需求。例如，Universe 可以包含表示公司中营销或会计部门的数据需求的对象。

Universe 也可以表示部门中某一组的数据需求，或者任何一组有组织的过程（如工资制度或盘存制度）的数据需求。

类示例包括“员工信息”、“出勤信息”和“部门信息”。

1.4.3.2 Universe 和数据库模式

数据库模式可用于创建三个 Universe：包含人事信息的 PERSONNEL，包含库存信息的 INVENTORY，包含销售信息的 SALES。每个 Universe 都包含类和对象。每个对象都映射到数据库结构的一个部分。

1.4.3.3 谁使用 Universe？

Web Intelligence 用户使用 Universe 进行报告和分析。Universe 应为他们提供与其业务领域相关的类和对象。

1.5 谁是 Universe 的设计员？

Universe 由 Universe 设计员使用 Universe 设计工具创建。并不存在针对 Universe 设计员的标准个人特征。在公司中，指定为 Universe 设计员的人员可以是数据库管理员、应用程序管理员或开发人员、项目经理，或者有足够的专业技能为其他用户创建 Universe 的报表创建者。

公司中可以有多个 Universe 设计员。Universe 设计员的数量取决于公司的数据需求。例如，可以为每个应用程序、项目、部门或职能领域指定一个 Universe 设计员。

如果有多位人员创建 Universe，请务必定义一套规则或术语指南，以便对象的表示方式保持一致。

1.5.1 必需的技能 and 知识

Universe 设计员应具有以下技能和技术知识水平：

技能/知识	说明
能够分析用户需求	创建 Universe 以满足用户的数据需求。Universe 设计员必须具有开展用户需求分析的技能，以便创建与用户词汇相关的类和对象，以及开发满足用户群需求的 Universe。这些需求包括适合于分析的报表创建和查询结果
数据库知识	Universe 设计员需要掌握有关以下方面的实用知识：公司的数据库管理系统 (DBMS)、数据库是如何部署的、逻辑数据库结构，以及公司数据库中存储的数据类型
结构化查询语言 (SQL)	必须具有 SQL 的实用知识

1.5.2 Universe 设计员的任务是什么？

Universe 设计员通常负责以下任务：

- 开展用户需求分析
- 设计和创建 Universe
- 分发 Universe
- 维护 Universe

1.6 Universe 的基本创建步骤

Universe 是 .unv 文件的组成部分，该文件由使用报表工具创建报表的最终用户使用。这是 .unv 文件中最终用户唯一可见的部分。创建 Universe 时，需要执行以下步骤：

1. 创建新的（空）Universe 文件，并设置连接和参数详细信息。

- 。 2 创建要使用的数据库表模式，然后定义表之间的联接。该模式对使用报表工具创建报表的最终用户不可见。
- 。 3 解决模式中的所有联接问题。
- 。 4 创建构成 Universe 的类和对象。该 Universe 由使用报表工具创建报表的最终用户使用。

相关主题

- 第 33 页上的[“执行基本操作”](#)
- 第 79 页上的[“创建 Universe 并设置 Universe 参数”](#)
- 第 143 页上的[“创建包含表和联接的模式”](#)
- 第 207 页上的[“解决模式中的联接问题”](#)
- 第 273 页上的[“创建 Universe”](#)
- 第 369 页上的[“优化 Universe”](#)

1.7 Universe 开发过程简介

以下几节概述了如何手动创建 Universe，并说明了 Universe 的创建如何适应典型的 Universe 开发周期。

1.7.1 Universe 设计方法

本手册所述的 Universe 设计方法包含一个规划阶段和三个实施阶段：

- 业务问题的分析和规划 Universe 解决方案
- 设计模式
- 构建 Universe
- 将 Universe 分发给用户

每个实施阶段均基于您已完成了初始规划阶段的假设。无需使用 Universe 设计工具即可完成规划阶段，而且此阶段决定了 Universe 的成败。如果 Universe 未基于对用户报表需求的调查并且规划不佳，则将难于设计、实施和维护它，对目标用户也没什么用处。

以下说明了上述各个阶段：

1.7.1.1 使用 Universe 设计工具之前规划 Universe

在开始第一阶段之前，您应花费一部分为 Universe 创建项目分配的时间（最高为 80%）来规划 Universe。应注意以下几点：

- 必须分析 Universe 的目标用户的数据分析和报表需求。您用于创建模式的结构应基于清楚定义的用户需求，这样才能访问那些表和列中包含的数据。
- 在开始使用 Universe 设计工具之前，应对需要创建的对象具有清晰的概念。不要通过查看数据库中可用的列来创建对象，但要识别那些与您已通过用户需求分析确定的对象匹配的列。

1.7.1.2 设计模式

为 Universe 的基本数据库结构创建一个模式。此模式包含目标数据库的表和列，以及链接这些表和列的联接。可能需要通过使用别名或上下文来解决可能会在结构中出现的联接问题，如循环、断层陷阱和扇形陷阱。测试整体结构的完整性。在本指南中，第 143 页上的 [“创建包含表和联接的模式”](#) 和第 207 页上的 [“解决模式中的联接问题”](#) 章节中描述了设计模式阶段。

1.7.1.3 构建 Universe

创建 Universe 对象，该对象推导出基于模式的元素的 SELECT 语句。将这些对象组织为类。这些是通过对用户报表需求进行分析所确定的对象。可以创建许多种类型的对象，以改进用户的报告能力、多维分析和优化查询性能。

测试 Universe 结构的完整性。应该通过在 Web Intelligence 中运行报表来执行测试。

第 273 页上的 [“创建 Universe”](#) 一章中描述了构建阶段。

1.7.1.4 分发 Universe

可以将 Universe 分发给用户进行测试并最终用于生产，方法是将它们导出到中央管理系统（CMS）资源库。第 548 页上的 [“部署 Universe”](#) 一章中描述了此阶段。

1.7.2 Universe 开发周期

Universe 开发是一个循环过程，它包括规划、设计、构建、分发和维护阶段。可使用 Universe 设计工具来设计和构建 Universe，但是，任何 Universe 的可用性直接与开发周期的其他各阶段的相互配合成功程度相关。

本节概述了一种 Universe 设计方法，您可以使用此方法规划和实施 Universe 开发项目。

下表概述了典型 Universe 开发周期的主要阶段：

开发阶段	说明
准备	<ul style="list-style-type: none">• 确定目标数据源并熟悉其结构。• 了解各目标数据库的各个表中包含哪些数据。• 了解联接。• 确定基数。• 了解可能发生的情况。
分析	<ul style="list-style-type: none">• 确定用户群及其组织方式，例如是按部门或按任务组织的用户组。• 确定用户需要哪些信息。• 确定用户需要哪些标准报表。• 熟悉用户的业务术语，以便您能切合实际地命名对象。

开发阶段	说明
规划	确定项目策略。例如，应创建多少个 Universe、哪些 Universe 应具有被链接的能力以及被链接的程度。
实施	<ul style="list-style-type: none"> 使用该工具构建 Universe。本手册论述 Universe 开发周期的此部分：Universe 设计工具的实际使用。 在构建过程中经常进行测试，以确保推导出的 SQL 的有效性和可靠性。
测试	组织一小组用户，最好是 Web Intelligence 的超级用户，这些用户在一定程度上了解他们期望从 Universe 获得的信息。要求这些用户执行模拟 Universe 的实际使用的彻底测试。
部署	通过将 Universe 导出到中央管理系统（CMS）资源库来分发 Universe，最终用户可在资源库中访问它们。
发展	在数据源与用户需求更改和发展时更新和维护 Universe。

注意：

Universe 设计应始终主要由用户需求而不是数据源结构推动。

1.7.3 优化 Universe 规划和实施时间

用户需求的分析和设计是整个过程中最重要的阶段。如果 Universe 将满足用户的需求，则用户必须通过用于命名对象的业务语言和可访问的数据大量参与到开发过程中。

如果正确执行前面三个阶段，则将能非常迅速轻松地进行实施。

可以花费一部分为 Universe 的开发分配的时间（最高为 80%）来执行前面三个阶段：

- 准备
- 分析

- 规划

如果花时间打好 Universe 的基础，则其余 20% 花在实际使用 Universe 设计工具来构建 Universe 的时间，将比没有花费必要的时间来规划和分析时要高效得多。

1.8 多语言 Universe

1.8.1 多语言 Universe

SAP BusinessObjects Enterprise XI4 的其中一项主要功能是可基于同一个 Universe 生成多语言元数据和报表。此功能为用户提供了一个一站式多语言报表解决方案，它区分区域设置，由单一元数据 Universe 模型支持，并提供全面的 Unicode 支持。用户只需依据同一个 Universe 生成报表一次，然后就可以根据用户首选项使用多种语言来显示这些报表。

可翻译的 Universe 元数据如下：

- Universe 名称
- Universe 说明
- 类名称
- 对象名称
- 对象名称、说明及格式
- 自定义层次结构名称
- 提示和输入列问题

注意：

只可翻译 Universe 元数据中定义的提示。无法翻译使用 @Prompt 函数定义的提示。

Universe 可能包含多个区域设置下的译文。当基于该 Universe 创建报表时，用于显示元数据的区域设置将取决于用户的首选查看区域设置。

Universe 也可定义回退区域设置，在没有任何可用的区域设置时将使用该区域设置。

基于 Universe 创建报表的用户只能看到已将状态设置为“可供使用”的区域设置。

在这些可见的区域设置下，基于 Universe 创建报表的用户只能看到状态在“翻译可见”类别中的元数据。“翻译可见”类别包含具有以下状态的元数据：

- NEEDS_REVIEW_TRANSLATION

- NEEDS_REVIEW_LOCALIZATION
- NEEDS_REVIEW_ADAPTATION
- TRANSLATED
- FINAL
- SIGNED_OFF

翻译 Universe 元数据、设置 Universe 区域设置以及设置元数据状态均通过翻译管理工具工具完成。译文与区域设置参数存储在附加的 XML 流中并以 .unv 文件格式保存。

Universe 设计工具用户界面也可以使用不同语言显示。下文介绍了该工具的多语言 Universe 功能。

1.8.2 语言和区域设置的定义

有些语言可能与多个国家/地区关联。例如，以下几个国家都讲法语 (fr)：法国 (FR)、比利时 (BE) 和瑞士 (CH)。在本例中，使用缩写 fr-FR、fr-BE、fr-CH 来分别代表法国 (FR)、比利时 (BE) 和瑞士 (CH) 使用的法语 (fr)。

语言	国家/地区
法语	法国
法语	比利时
法语	瑞士

同样，一个国家/地区可能与多种语言关联 (fr-CH、de-CH、it-CH)。例如，在瑞士，德语、法语和意大利语这三种语言都使用。

语言	国家/地区
法语	瑞士
德语	瑞士
意大利语	瑞士

一个区域设置定义一种语言和一个地理区域、以及数据的排序方式。日期和时间常常按特定格式进行格式化。当将语言与国家/地区组合在一起时（例如法国法语），区域设置显示如下，具体取决于操作系统或应用程序：

操作系统	区域设置的格式
Windows	Français (France) 区域设置源自用户的系统设置（国家/地区）。
Java	fr_FR
Sun Solaris	fr_FR.IS08859-1
Infoview 6.5.1	Français (France) [fr-FR]

为简化文档编写和应用，可能会使用同一术语“语言”来表示语言和区域设置两种含义。

1.8.3 不同的区域设置

术语	定义
产品语言	这是 Universe 设计工具用户界面的语言。菜单和消息以该语言显示。
首选查看区域设置（PVL）	首选的查看语言设置。它定义资源（文档或 Universe）内容或属性列表的组成部分 — 字符串、文本和格式在 InfoView 或 Web Intelligence 胖客户端上的应用程序以何种区域设置显示。
替代（回退）区域设置	首选查看区域设置不可用时所使用的区域设置。
源语言	创建文档时使用的区域设置。

1.8.4 设置 Universe 设计工具用户界面的产品语言

在 Universe 设计工具的“工具”>“选项”设置的“常规”选项卡中，从可用语言列表中选择一种“语言”，这就是产品语言，也称为“用户界面语言（UIL）”。这对 Universe 元数据没有影响：对

象名称、上下文名称和类采用数据库元素的原始语言显示。要翻译 Universe 元数据，请使用翻译管理工具。

1.8.5 使用多语言 Universe

当用户在多语言 Universe 的基础上创建报表时（例如在 Web Intelligence 中），显示元数据所使用的区域设置取决于该区域设置的可用性/状态以及元数据类别：

- 当且仅当元数据译文的状态为“可见”并且已将区域设置定义为“可供使用”时，才会显示该区域设置下的元数据译文。
- 显示元数据译文所用的区域设置的优先顺序如下：
 - 用户的首选查看区域设置。
 - 如果首选查看区域设置不可用，则使用回退区域设置（如果已在此 Universe 中定义）。
 - 如果在此 Universe 中未定义回退区域设置，则使用用户首选查看区域设置的主区域设置。
 - 如果它也不可用，则显示原始内容。该原始内容就是在 Universe 设计工具中定义的原样元数据。

1.8.6 确定链接的 Universe 中的回退区域设置

派生的 Universe 可以重复使用来自不同核心 Universe 的元数据。当派生的 Universe 与核心 Universe 定义不同的回退区域设置时：

- 如果在派生的 Universe 级别定义了回退区域设置，则使用该回退区域设置。
- 如果没有在派生的 Universe 级别定义回退区域设置，则使用在派生的 Universe 中定义的第一个核心 Universe 的回退区域设置（如果有）。
- 如果在任何核心 Universe 中均未定义回退区域设置，则该 Universe 没有回退区域设置。

1.8.7 翻译管理工具

Universe 设计工具不能用于翻译 Universe 元数据，甚至不能显示元数据译文。在 Universe 设计工具中，只显示原始的 Universe 内容。对于翻译任务，BusinessObjects Enterprise 套件提供了翻译管理工具。该工具是一个独立式应用程序，仅可以在 Windows 平台上使用。

通过翻译管理工具，Universe 的设计者可以执行以下任务：

- 在 Universe 中添加新区域设置并将其设置为“可供使用”。
- 定义 Universe 的回退区域设置。
- 通过界面本身，在添加的区域设置下翻译 Universe 元数据。
- 设置不同区域设置下的元数据译文的状态。
- 将 Universe 元数据导出到 XLIFF 文件以进行外部翻译，然后再导回该应用程序。

Universe 元数据进行翻译后，可以重新保存，这样，多语言报表便可以使用这些译文。有关更多信息，请参阅翻译管理工具用户指南。

1.8.8 多语言数据

通过 PREFERRED_VIEWING_LOCALE 和 DOMINANT_PREFERRED_VIEWING_LOCALE 变量，设计者可以对 Universe 进行自定义，以便在查询时过滤多语言数据并仅检索用户首选查看区域设置下的数据。这可以使用 @Variable 函数实现。

1.9 Universe 设计工具示例材料

下面是 Universe 设计工具随附的示例：

1.9.1 演示数据库

本指南中的大多数示例均基于利用 Microsoft Access 2000 构建的 Club 数据库。此数据库由虚构的企业（Island Resorts）的销售经理用来执行销售和营销分析。可以在 Business Objects 安装路径中的 Databases 子文件夹下找到数据库文件 Club.mdb。

有关此数据库结构的更多信息，请参阅本指南后面的附录。

此版本还附带了 efashion 数据库。此 MS Access 2000 数据库跟踪 3 年来在 13 间商店（12 间在美国，1 间在加拿大）销售的 211 种产品（663 种产品色调变化）。

此数据库包含：

- 一个包含 89,000 行销售信息（以周为基础）的中央事实表。
- 包含促销信息的第二个事实表。
- 两个聚合表（利用聚合导航建立）。

1.9.2 演示 Universe

BusinessObjects 安装路径中 Samples 文件夹的 Universes 子文件夹中包含了一个名为 beach.unv 的完整演示 Universe。它是利用上述的 Club 数据库构建的。

可以使用此 Universe 来了解如何使用 Universe 设计工具构建特定的对象和类。

Universe 设计工具还随附了使用 efashion 数据库构建的 efashion Universe。

1.10 使用信息设计工具处理 Universe

可使用信息设计工具处理通过 Universe 设计工具、Universe 设计工具桌面版、Universe Designer 或 Universe Designer 个人版创建的 .unv 格式 Universe。这些文件不能直接通过信息设计工具使用，必须先进行转换，或升级再转换，具体取决于文件的版本。请参见信息设计工具以获取有关为使用不同版本的 .unv Universe 文件而需执行的步骤的信息，以及有关转换文件之后支持哪些功能的信息。

注意：

如果为了使用信息设计工具进行处理而转换了 .unv 文件，就无法使用该文件的原始创建工具将其打开。

第 2 章 执行基本操作

2.1 概述

本章介绍了在 Universe 设计工具中创建、修改和更新 Universe 所需执行的基本操作。涵盖了以下主题：

- 第 33 页上的 [“启动 Universe 设计工具”](#)
- 第 44 页上的 [“导入 Universe”](#)
- 第 45 页上的 [“打开 Universe”](#)
- 第 46 页上的 [“导出 Universe”](#)
- 第 48 页上的 [“保存 Universe”](#)
- 第 79 页上的 [“创建 Universe 并设置 Universe 参数”](#)
- 第 52 页上的 [“使用 Universe 设计工具用户界面”](#)
- 第 58 页上的 [“使用“查找并替换””](#)
- 第 61 页上的 [“组织表显示”](#)
- 第 67 页上的 [“选择模式显示选项”](#)
- 第 75 页上的 [“打印 Universe”](#)

2.2 启动 Universe 设计工具

Universe 设计工具只能与中央管理系统（CMS）资源库一起使用。在启动该工具之前，必须登录到资源库。

如果是第一次启动该工具并且想依据现有 Universe 进行操作，您首先需要直接打开 Universe，使用安全的连接保存该 Universe，然后将其导出到资源库。接着，您导入该 Universe 以进行更新，并导出更新后的版本。这样可以确保 CMS 和本地 Universe 版本保持同步。

启动 Universe 设计工具之后，就可以采用以下方式之一打开 Universe：

- 创建新的 Universe
- 从 CMS 资源库导入 Universe
- 从文件系统中直接打开 Universe

只有在将 Universe 导出到资源库时，Web Intelligence 用户才能使用它。使用 Universe 设计工具的最常见方法是：导入 Universe，进行更改，然后导出更新后的 Universe。这种方法确保 CMS（资源库）版本可与文件版本保持同步。

注意：

可以将 Universe 保存到文件系统。如果要与可能没有目标 CMS 连接权限的其他用户共享 Universe，您可以这样做。有关更多信息，请参阅第 48 页上的 [“保存 Universe”](#) 一节。

通过在此版本的已安装 Business Objects 产品组中单击该工具的图标，可以从任务栏中启动 Universe 设计工具。在该工具启动之前，将会提示用户登录到 CMS。

2.2.1 启动 Universe 设计工具

要启动 Universe 设计工具：

- 1 单击任务栏上的“开始”按钮。
 - 2 指向“程序”菜单。
 - 3 从 BusinessObjects 命令中单击 Universe 设计工具程序。
- 将出现 CMS 的登录框。
- 4 键入以下信息。此信息通常由 BusinessObjects 管理员提供。

登录信息	说明
系统	CMS 服务器的名称。
用户名	您的资源库用户名。
密码	您的资源库密码。

登录信息	说明
身份验证	您的安全级别

5 单击“确定”。

Universe 设计工具启动屏幕将显示，同时会打开一个空会话。

用户名和 CMS 名称出现在标题栏中。

根据为 Universe 设计工具设置的选项，在启动 Universe 设计工具时可自动启动“快速入门”Universe 设计向导。单击“取消”关闭向导。有关禁用其他向导选项的更多信息，请参阅第 43 页上的[“取消激活“快速设计向导””](#)一节。如果要使用“快速设计向导”，用户可以参考第 36 页上的[“使用“快速设计向导””](#)一节。

2.2.2 使用“快速设计”向导

初次启动会话时，默认情况下会出现“快速设计”向导。可以使用该向导快速创建 Universe，或熟悉一下 Universe 设计工具，但是它并不太适合用于创建响应最终用户报告需求的完整 Universe。

建议禁用“快速设计”向导，只将其用作熟悉 Universe 设计工具的一种方式，而不要用于设计 Universe。本手册中的所有 Universe 设计、构建和维护信息及过程均假定已禁用“快速设计”向导，但专门论述如何使用此向导的第 36 页上的[“使用“快速设计向导””](#)一章除外。有关禁用其他“快速设计”向导选项的信息，请参阅第 43 页上的[“取消激活“快速设计向导””](#)一节。

2.3 通过 Designer XI R3 使用 XI R2 连接和 Universe

在此 Universe 设计工具版本中可以访问连接，并打开或导入存储在 XI R2 CMS 中的 Universe。在使用 XI R2 Universe 和连接时，需要注意以下几点：

- Desktop Intelligence XI R2 用户可以基于 XI 3.1 Universe 和 XI R2 连接来刷新通过 Desktop Intelligence XI 3.1 创建的文档。
- Desktop Intelligence XI R2 用户可以基于 XI 3.1 Universe 和 XI R2 连接创建文档。
- 如果要编辑和保存 XI R2 连接，将显示一则警告消息，告知用户如果保存该连接，则会将它另存为 XI 3.1 连接，而且将无法刷新使用该连接的 XI R2 报表。

- 用户可以使用 XI 3.1 Universe Designer 打开 XI R2 Universe，但不能使用以前的 Designer 版本打开 XI 3.1 Universe。

Desktop Intelligence XI R2 与 XI 3.1 安装之间的互连功能使管理员可以在升级服务器的同时，保留连接到已升级的 XI 3.1 服务器的 DeskTop Intelligence XI R2 和 XI 3.1 客户端。这是当需要管理的客户端群比较大时采取的临时阶段。

2.4 使用“快速设计”向导创建基本 Universe

对于基于简单的关系模式的演示 Universe 或快速测试 Universe，请使用“快速设计向导”创建基本但完整的 Universe。您可以立即使用产生的 Universe，也可以修改对象并创建复杂的新对象。通过这种方式，您可以逐渐改进 Universe 的质量和结构。

如果设计的是用于生产的 Universe，则应手动创建该 Universe。本指南的其他所有章节均建立在介绍如何手动创建 Universe 这一基础上。本节是唯一一个说明自动创建 Universe 的章节。

2.4.1 为何使用快速设计向导？

快速设计向导能帮助您完成创建 Universe 的整个过程。它引导您建立与数据库的连接，然后让您创建简单的类和对象。此向导还提供了用于自动创建对象、联接和表的内置策略。

使用“快速设计向导”有以下好处：

- 对于 Universe 设计工具的新用户，它可以帮助熟悉用户界面和基本的 Universe 设计。
- 如果您创建的是演示 Universe，它通过自动执行大部分设计过程节省您的时间。利用此向导，您可以快速建立 Universe 的工作模型，然后可以按照目标用户的需求定制 Universe。

2.4.2 使用“快速设计向导”

“快速设计向导”是用于自动创建 Universe 的向导的名称。以下各节说明了此向导中的每一步。

2.4.2.1 启动“快速设计向导”

启动“快速设计向导”：

- 1 启动 Universe 设计工具。

将显示“用户标识”对话框。



- 2 在“用户标识”对话框中，输入用户名和密码。
- 3 单击“确定”。

将出现快速设计向导的欢迎屏幕。

注意：

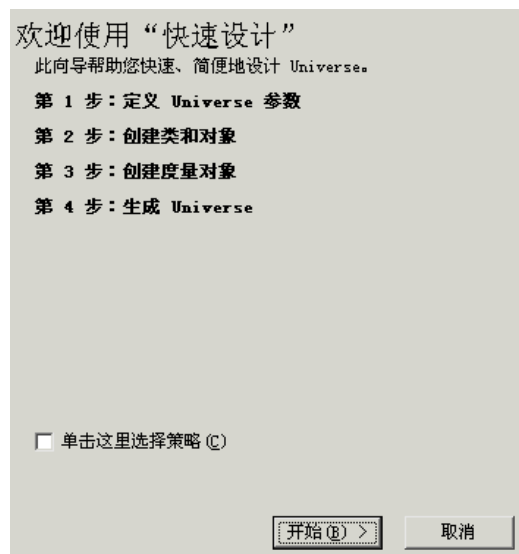
如果不想在下次启动会话时显示此向导，则清除“在启动时运行此向导”复选框。此外，可以在“选项”对话框的“常规”选项卡中找到两个与此向导的显示相关的选项：“显示欢迎向导”和“文件/新建启动快速设计向导”（“工具”菜单，“选项”命令）。

2.4.2.2 欢迎屏幕

欢迎屏幕显示创建基本 Universe 所需的四个步骤的概览。它还提供了一个复选框：单击这里选择策略。如果单击此复选框，则将能够选择用于创建 Universe 的策略；否则，Universe 设计工具会应用默认的内置策略。

在随后的每个对话框中，“快速设计向导”会提示您提供执行操作所需的信息。

要从一个对话框移到另一个对话框，请单击“下一步”。可以通过单击“后退”返回到前一个对话框。可以随时单击“取消”按钮，以结束过程并退出“快速设计向导”。



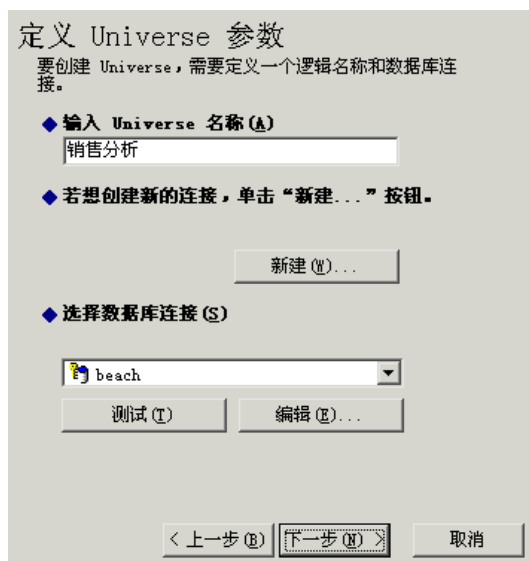
当您选择“单击这里选择策略”复选框时，将出现一个列出策略的对话框。第 39 页上的[“选择策略”](#)中描述了此对话框。可以选择策略或接受缺省的策略。

单击“开始”以启动创建过程。

2.4.2.3 定义 Universe 参数

在此步骤中定义 Universe 参数：Universe 名称和数据库连接。

可以为 Universe 输入长名称（最多为 35 个字母数字字符）。



可以创建连接或选择一个现有连接。要创建连接，请单击“新建”按钮，然后在后续的对话框中指定必要的参数。有关这些对话框的更多说明，请参阅第 85 页上的 [“定义和编辑连接”](#) 一节。

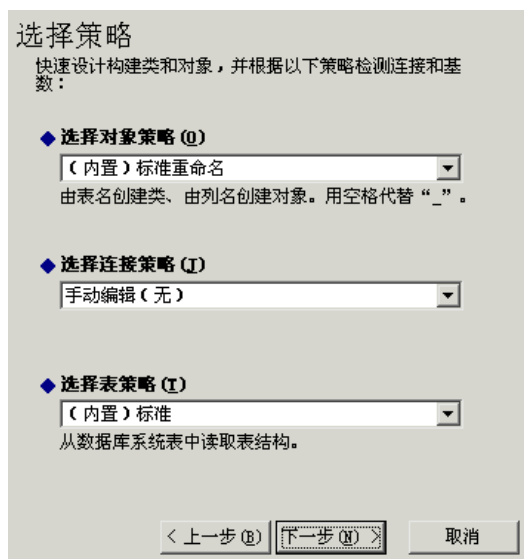
要检查连接是否有效，请单击“测试”按钮。“编辑”按钮可让您修改连接的参数。

单击“下一步”按钮继续执行下一步。

2.4.2.4 选择策略

如果您单击了欢迎屏幕中策略的复选框，则“快速向导”会提示您指定用于创建对象、联接和表的策略。

一个策略是从数据库或平面文件中读取结构信息的脚本。Universe 设计工具使用这些脚本来自动创建对象、联接和表。



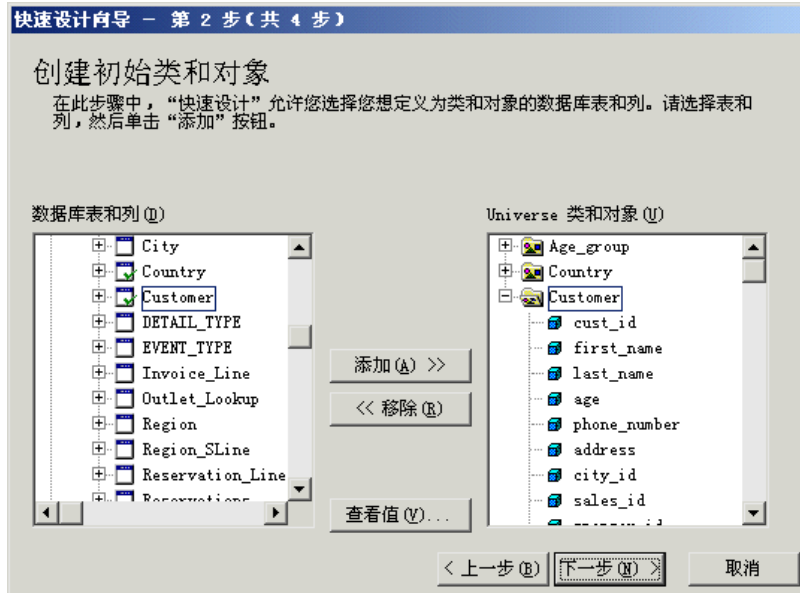
可以从列表框中选择另一个策略或不选择策略。列表框的正下方显示了当前策略的简短说明。

除了 Universe 设计工具提供的内置内部策略之外，还可以创建自己的外部策略。请参阅第 410 页上的 [“使用外部策略定制 Universe 创建”](#) 一节。

单击“下一步”按钮继续执行下一步。

2.4.2.5 创建初始的类和对象

根据数据库连接的参数，此向导为您显示了一个数据库表和列的列表。通过从左侧窗格中选择表和列，然后将它们添加到右侧的“Universe 类和对象”窗格，创建初始的类和对象。



默认情况下，左侧窗格只显示表的名称。可以使用以下方法浏览文件树，然后将类和对象添加到右侧窗格中：

- 要查看任何表的列，请单击表名称左侧的加号 (+)。
- 要查看任何表或列的数据值，请单击它，然后单击“查看值”按钮。
- 要选择一个表，请单击该表，然后单击“添加”按钮。
- 要选择几个连续的表，请按住 Shift 键，然后单击第一个表和最后一个表。将会突出显示所选择的表之间的所有表。然后，单击“添加”按钮。
- 要选择几个不连续的表，请按住 Ctrl 键并单击每个表。单击“添加”按钮。
- 选择表的另一个方法是将它们从左侧窗格拖放到右侧窗格。

插入表时，Universe 设计工具将包含表中的所有列。

在右侧窗格中，类的名称显示在文件夹图标的旁边。单击类名称旁边的加号 (+) 可查看对象。可以重命名类或对象，方法是双击类或对象，然后在对话框中输入新名称。

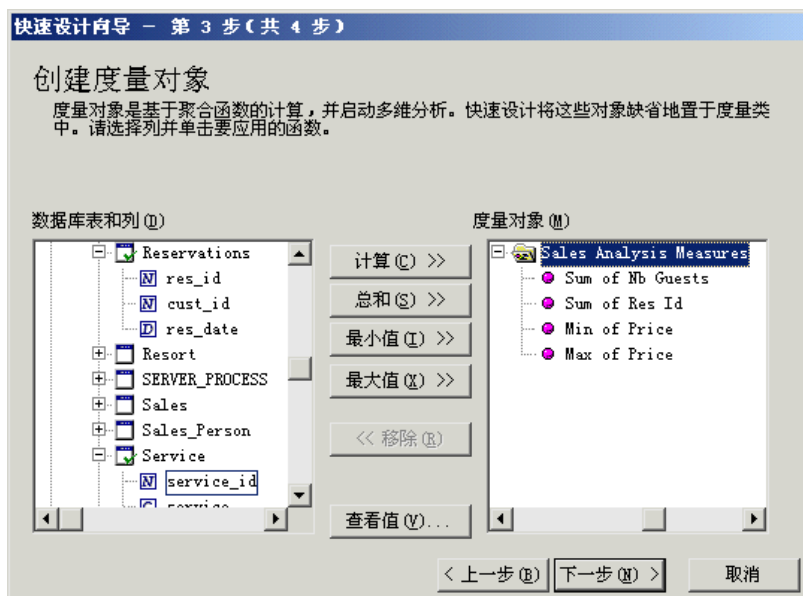
默认情况下，对象被限定为维对象，这由对象名称前的立方体符号指示。

要删除类或对象，请单击类或对象，然后单击“移除”按钮。

单击“下一步”，继续执行下一步。

2.4.2.6 创建度量对象

度量对象是由聚合函数派生的：Count、Sum、Minimum 或 Maximum。此类型的对象提供了数值信息。以下对话框的右侧窗格中显示了度量对象的示例：



如果想查看与对象关联的数据值，请单击对象，然后单击“查看值”按钮。

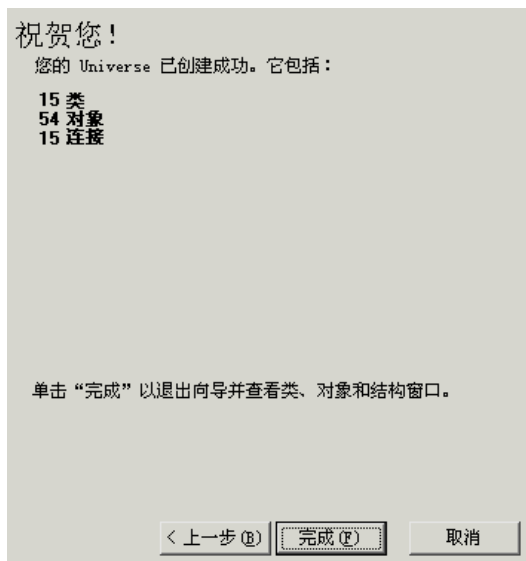
要创建度量对象，请单击左侧窗格中的适当对象，然后单击聚合按钮。可以重命名您创建的任何度量对象。

将度量对象分组为一个或多个度量类可以改善 Universe 的组织。这还使最终用户能轻松进行导航。有关度量对象的更多信息，请参阅第 303 页上的 [“定义度量”](#) 一节。

单击“下一步”时，“快速设计向导”将开始创建 Universe。

2.4.2.7 生成 Universe

“快速设计向导”会根据您指定的参数自动生成新的 Universe。它指出在 Universe 中创建的类、对象和联接数。



在上面的对话框中有一条消息，说明 Universe 的联接中存在环路。Universe 设计工具支持通过别名和上下文解决环路。有关更多信息，请参阅“设计模式向导”一章。

单击“完成”按钮时，将出现新 Universe 的 Universe 窗格和结构窗格。

2.4.2.8 结束工作会话

选择“文件”>“另存为”以保存 Universe，然后选择“文件”>“关闭”以关闭 Universe。

保存 Universe 时，Universe 设计工具会提示输入文件名。Universe 文件名可以包含操作系统允许的最大字符数。它的扩展名为 .unv。默认情况下，Universe 设计工具将这些文件存储在 BusinessObjects 文件夹的 Universe 子文件内。在 Windows 2000 中，此文件夹显示在您的用户配置文件的 Local Data 文件夹下。

要退出 Universe 设计工具，请选择“文件”>“退出”。

2.4.2.9 取消激活“快速设计向导”

初次启动会话时，默认情况下会出现“快速设计”向导。可以按以下所述在创建新的 Universe 时阻止此向导自动出现：

取消激活“快速设计”向导：

- 1 选择“工具” > “选项”。
- “选项”对话框即会打开到“常规”页面。
- 2 清除“显示欢迎向导”复选框。（如果已清除“启动向导”欢迎页中的“在启动时运行此向导”复选框，则“显示欢迎向导”复选框已被清除。）
- 3 清除“文件/新建启动快速设计向导”复选框。
- 4 单击“确定”。

可以随时激活“快速设计”向导，方法是从“选项”对话框的“常规”页中选择上述复选框。第 36 页上的 [“使用“快速设计向导””](#) 一节论述了使用“快速设计”向导的信息。

2.4.3 对利用快速设计向导创建的 Universe 进行后续处理

利用“快速设计”向导创建了基本的 Universe 后，可能会发现必须编辑联接，以及使用别名或上下文解决所有环路。此外，可以选择使用各种 Universe 设计工具功能通过更复杂的组件来改进 Universe。有关的适当信息，应参阅本手册中相关的章节。

2.5 导入 Universe

可以导入一个或多个存储在资源库的 Universe 文件夹中的 Universe。只能导入已导出到资源库的 Universe。

当您导入 Universe 时，CMS 将检查资源库文件系统上的 Universe 版本。如果版本相同，则 Universe 设计工具可使用该 Universe。如果资源库文件系统上的 Universe 版本比 CMS 版本更新，则会出现一个消息框，询问您是否要替换文件夹中的 Universe。如果回答“是”，则会将资源库文件系统中的 Universe 替换为 CMS 中的版本。

2.5.1 从资源库导入 Universe

2.5.1.1 从资源库导入 Universe

- 1 选择“文件” > “导入”。
将出现“导入 Universe”对话框。
- 2 从下拉列表框中选择一个 Universe 文件夹。
或
单击“浏览”按钮，并使用文件夹浏览器选择一个 Universe。
您想从此文件夹中导入 Universe。
- 3 如果要锁定 Universe，请双击 Universe 名称。
锁定的 Universe 显示为带有挂锁符号。要取消锁定 Universe，请再次双击它。
- 4 单击一个 Universe 名称。
这是您想导入的 Universe。
- 5 验证“导入文件夹”框中的导入文件夹的文件路径。
Universe 将被导入到此文件夹。
- 6 单击“确定”。

2.5.2 打开操作和导入操作有什么不同？

可以从文件系统中直接打开 Universe。当您保存此 Universe 时，只会将它保存到文件系统，而不会在 CMS 中更新此 Universe。此 Universe 的更新不可供 Web Intelligence 用户使用。

当导入 Universe 时，会将资源库中的最新可用版本提供给 Universe 设计工具。当您完成修改 Universe 时，可将其导出到资源库。CMS 即会被最新的更改加以更新。

2.6 打开 Universe

使用菜单命令或单击“打开”按钮打开 Universe。当您不执行导入而直接打开 Universe 时，所打开的是本地文件系统上的版本，该版本可能与 CMS 中的最新版本不对应。

2.6.1 直接打开 Universe

- 1 选择“文件” > “打开”。

“打开”框即会打开到指定为 Universe 文件默认存储位置的目录。可以在“选项”对话框的“保存”页（“工具” > “选项” > “保存”）中设置此目录。

- 2 如有必要，浏览到包含 Universe 文件（.UNV）的目录。
- 3 选择一个 Universe 文件并单击“打开”

或

双击 Universe 文件。

Universe 在当前的 Universe 设计工具窗口中打开。

2.7 导出 Universe

通过将 Universe 导出到资源库，可以将 Universe 提供给 Web Intelligence 用户和其他设计员使用。

当您导出某个 Universe 时，将对该 Universe 进行如下操作：

- 将其移动到资源库文件系统上的选定 Universe 文件夹
- 并且
- 在中央管理系统（CMS）中创建它。

每次将 Universe 导出到资源库时，CMS 中的 Universe 版本都会得到更新。这是 Web Intelligence 用户可以使用的版本。

注意：

保存 Universe 与导出 Universe 不同。保存操作将更新资源库本地文件系统上的 Universe，而不是 Universe 的 CMS 资源库版本。更多信息，请参阅第 48 页上的 [“导出操作和保存操作有什么不同？”](#) 一节。

2.7.1 Universe 在资源库文件系统上是如何排列的？

资源库在本地文件系统和 CMS 服务器中存储 Universe。您处理的是本地文件系统上的 Universe 版本。本地文件系统是指安装了该工具的服务器。默认情况下，Universe 会保存在您的用户配置文件路径的 universes 文件夹中，如下所示：

```
\\Documents and Settings\<用户>\Application Data\Business Objects\Business Objects  
12.0\universes\@<资源库名称>\universe folder\<universe>.unv
```

存储在 CMS 服务器中的 Universe 用于版本控制。当您将更新的 Universe 导出到资源库时，更新的 Universe 将被复制到 CMS 服务器。

2.7.2 将 Universe 导出到资源库

2.7.2.1 将 Universe 导出到资源库

- 1 选择“文件” > “导出”。

将出现“导出 Universe”对话框。

- 2 从文件夹下拉列表框中选择一个 Universe 文件夹。

或

单击“浏览”按钮，并在文件夹浏览器中选择一个 Universe 文件夹。

您想将 Universe 导出到此文件夹。

- 3 如果要锁定 Universe，请双击 Universe 名称。

锁定的 Universe 显示为带有挂锁符号。要取消锁定 Universe，请再次双击它。

- 4 在“组”列表框中单击某个组。这是使用导出的 Universe 的用户组。
- 5 在“Universe”列表框中单击某个 Universe。“Universe”列表框显示了活动 Universe 的名称。
- 6 如果想导出其他未打开的 Universe，请单击“添加 Universe”按钮，然后使用浏览器选择其他 Universe。
- 7 单击“确定”。

2.7.3 导出操作和保存操作有什么不同？

当您保存 Universe 时，所更新的是资源库文件系统中的版本。这不会更新 CMS 版本。

当您导出 Universe 时，资源库文件系统中的版本更新将与 CMS 中的 Universe 更新同步。

如果保存 Universe 而没有导出更新的版本，则不会更新 CMS。其他用户将无法使用保存的 Universe。

资源库中的每个 Universe 均获分配有一个系统标识符。有关标识符的更多信息，请参阅第 548 页上的 [“在资源库中标识 Universe”](#) 一节。

如果另一个设计员在资源库中锁定了某个 Universe，您则不能导出它。

只能导出利用安全连接定义的 Universe。

2.8 保存 Universe

应定期保存工作会话中的 Universe。当保存 Universe 时，Universe 设计工具会在本地文件系统中将其保存为具有 .unv 扩展名的文件。

在 Web Intelligence 中，用户通过 Universe 名称（长名称）标识 Universe。

当您保存 Universe，更改将不会保存到 CMS。在完成更新 Universe 时，必须将 Universe 导出到 CMS。

Universe 名称（长名称）和 .unv 文件名中可使用的最大字符数如下：

名称类型	最大字符数
Universe 名称	100
.unv 名称	操作系统最大长度

2.8.1 作为标识符的 Universe 文件名

在根据某个 Universe 创建报表后，不得更改该 Universe 的文件名 .unv。如果更改文件名，则对于任何根据旧名称的 Universe 构建的报表，一旦更改了该 Universe 的名称，这些报表将不会指向该 Universe。

2.8.2 保存 Universe

Universe 名称可以不同于 .unv 名称。

在使用“另存为”以新名称保存 Universe 时，将不会在 CMS 中关联新的 Universe。必须将新的 Universe 导出到 CMS 才能创建新 Universe 的一个版本。

可以使用以下方法保存 Universe：

- 从菜单栏中选择“文件” > “保存”
- 单击“保存”图标
- 在键盘上按 Ctrl+S

2.8.3 将 Universe 定义另存为 PDF

将 Universe 信息另存为 Adobe PDF 文件。可以保存您能够为 Universe 打印的那些信息。这些信息包括：

- 常规信息：参数、链接的 Universe 和图形表模式。
- 元素列表：Universe 中的元素的列表，包括对象、条件、层次、表、联接和上下文。
- 元素说明：Universe 中的对象、条件、层次、表、联接和上下文的说明。

可以从“打印选项”对话框（“工具” > “选项” > “打印”）中选择要以 PDF 格式显示的元素。第 76 页上的 [“设置打印选项”](#) 一节中对这些选项进行了说明。

将 Universe 信息另存为 PDF 文件：

- 1 选择“文件” > “另存为”
- 2 从“另存为类型”下拉列表框中选择“可移植文档格式 (*.pdf)”。



- 3 单击“保存”。

2.8.3.1 设置默认保存选项

默认情况下，Universe 设计工具将用户保存的文件存储在 Business Objects 路径中的 Universe 子文件夹内。可以按以下所述指定另一个默认保存文件夹：

- 1 选择“工具” > “选项”。
将出现“选项”对话框。
- 2 单击“保存”选项卡。
将出现“保存”页面。

- 3 在“默认 Universe 文件夹”文本框中键入文件路径。
或
- 4 浏览到包含 .unv 文件的文件夹。
- 5 如果想指定自动保存时间，则选择“自动保存”复选框，然后在“选择分钟值”框中选择或键入时间间隔数字。
- 6 单击“确定”。

2.9 关闭 Universe

可以使用以下方法关闭 Universe。

关闭 Universe:

- 从菜单栏中选择“文件” > “保存”
- 单击 Universe 窗口右上角的“关闭窗口”按钮
- 在键盘上按 Ctrl+W。

2.10 与多个设计员一起工作

可以在多用户的环境中使用 Universe 设计工具，在此环境中，多个设计员可以处理同一 Universe，而不会在版本之间造成冲突。

可以锁定某个 Universe，以便同一时间只有一个设计员可以修改该 Universe，而且还可以为 Universe 分配一个版本号，以便跟踪更改。

2.10.1 锁定 Universe

当 Universe 存储在 Universe 文件夹中时，多个设计员可以共享该 Universe，只要他们具有必要的用户权限。

同一时间只能有一个设计员处理指定的 Universe。对于想处理某个 Universe 的设计员而言，只有在另一个设计员未锁定此 Universe 时，前者才能这样做。

注意：

通过“导入”或“导出”对话框锁定 Universe。锁定 Universe 时，会在 Universe 名称旁边显示一个挂锁符号。当另一个设计员锁定 Universe 时，挂锁符号会变暗。

2.10.2 修订版号

每次将 Universe 导出到 Universe 文件夹时，Universe 设计工具会增加 Universe 的修订版号。这使您可以确定哪一个是最新的 Universe 版本。修订版号出现在 Universe 参数的摘要选项卡（文件 > Universe 参数 > 摘要）中。

2.11 使用 Universe 设计工具用户界面

Universe 设计工具用户界面符合 Microsoft Windows 标准。它包含窗口、菜单、工具栏、快捷键和联机帮助。

2.11.1 用户界面的主要构成项

每个 Universe 均包含在单个 Universe 窗口中，而后者又包含在主窗口中。

还可以使用一个称为“表浏览器”的独立窗口，它显示了所连接的数据库中所有可用的表。

2.11.1.1 Universe 窗口

Universe 窗口分为两个窗格：

窗格	显示
结构	Universe 的基本目标数据库的图形表示方式。它包含您将对象（最终用户使用这些对象来运行其查询）映射到的表和联接。
Universe	在 Universe 中定义的类和对象。它们是 Web Intelligence 用户看到的和用于创建其查询的 Universe 的元素。

2.11.1.2 表浏览器

表浏览器是一个窗口，它显示所连接的数据库中可用的表。可以通过以下方法将表插入“结构”窗格中：选择表并将其拖入“结构”窗格中，或者双击表浏览器中的适当表。

可以通过以下任一方法显示表浏览器：

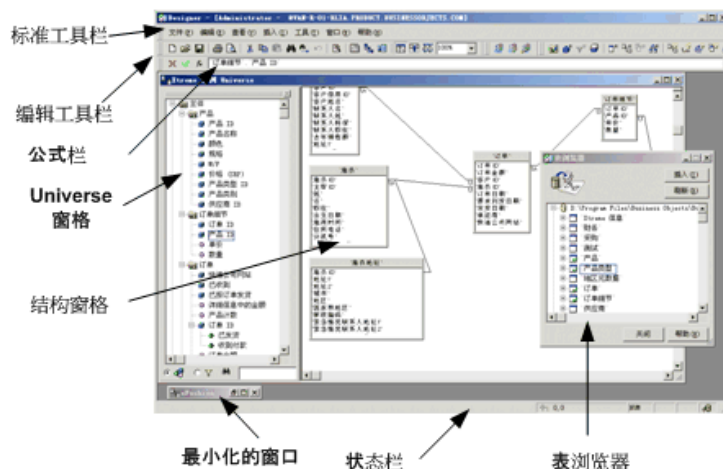
- 双击“结构”窗格的背景。
- 右键单击“结构”窗格的背景，然后从上下文菜单中选择“插入表”。
- 选择“插入” > “表”。

注意：

“设计模式”一章中详尽说明了表浏览器的使用。

2.11.2 Universe 设计工具用户界面

以下标了界面的主要构成项：



2.11.3 使用窗口

可以按以下方式使用该用户界面中的窗口：

- 在工作会话中，可以同时处理多于一个的 Universe。该工具在一个“结构”窗格和一个“Universe”窗格中显示每个 Universe。
- 最近打开的 Universe 列在“文件”菜单的底部。可以通过以下方式修改列出的 Universe 数量：选择“工具” > “选项” > “常规”，然后在“最近打开的文件列表”中指明 Universe 的数量。
- 可以在 Universe 设计工具窗口中移动任何窗口，调整这些窗口的大小，或者最小化这些窗口。
- 用户可以按自己觉得最方便的方式放置这些窗口，方法是选择“窗口” > “排列”，然后选择“层叠”、“水平平铺”或“垂直平铺”。
- 通过选择“窗口” > “排列图标”，可以排列工具窗口中已最小化的所有窗口。

2.11.4 使用工具栏

Universe 设计工具窗口包含两组工具栏：“标准”工具栏和“编辑”工具栏。

对于任一个工具栏，可选择的按钮取决于哪一个窗格处于活动状态（“Universe”窗格或“结构”窗格）。不可用的按钮会变暗。

工具栏是可停靠的。可以拖动工具栏，并将其放在 Universe 窗口中的任何地方。

2.11.4.1 移动工具栏

移动工具栏：

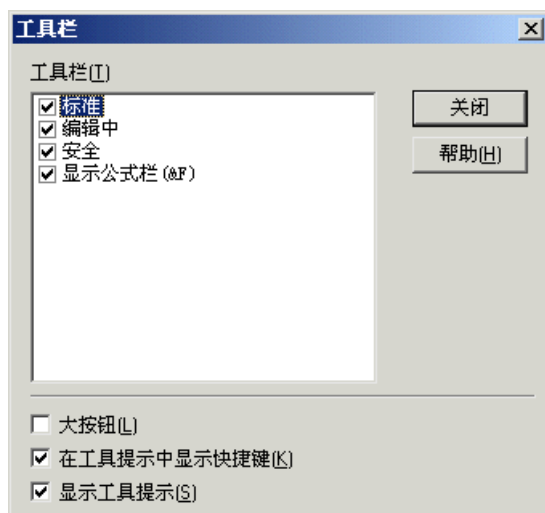
- 1 单击包含工具栏的矩形中的区域。
上图中显示了这两个工具栏的区域。
- 2 按住鼠标左键，将工具栏拖到所要的位置。
- 3 松开鼠标按钮。

工具栏将独自显示。

2.11.4.2 隐藏和显示工具栏

交替地显示或隐藏任一工具栏：

- 1 选择“查看” > “工具栏”。
将出现“工具栏”对话框。



- 2 选择或清除对应于工具栏的复选框。
- 3 选择或清除列在对话框底部的工具栏按钮、工具提示和快捷键显示选项。
- 4 单击“确定”。

2.11.5 在 Universe 设计工具中执行操作

在 Universe 设计工具中，可以按照以下方式执行操作：

- 从菜单中选择一个命令。
- 按 Alt 键并通过键盘输入快捷键。
- 单击工具栏上的按钮。

2.11.5.1 使用鼠标

可以按以下所述单击和双击鼠标：

单击

单击可执行以下操作：

- 执行标准操作（选择命令或单击按钮）
- 从“Universe”窗格、“结构”窗格或表浏览器中选择元素。
- 如果在 Universe 设计工具窗口中选择一个或多个元素，单击鼠标右键将会显示弹出菜单。它包含了与所选的元素相关的命令。

双击

可以双击以下 Universe 结构以影响显示的更改或修改属性：

双击...	结果...
“结构”窗格中的空白空间	表浏览器将出现。
“结构”窗格中的表	修改表的显示。可以按三种视图中的任一种来显示表及其列。有关更多信息，请参阅第 65 页上的 “更改表的显示” 一节。
“结构”窗格中的联接	联接的“编辑联接”对话框将出现。可以从此对话框中修改联接属性。
“Universe”窗格中的类	类的“编辑属性”对话框将出现。可以从此对话框中修改类属性。
“Universe”窗格中的对象。	对象的“编辑属性”对话框将出现。可以从此对话框中修改对象属性。
“Universe”窗格的“条件”视图中的条件对象	条件对象的“编辑属性”对话框将出现。可以从此对话框中修改对象属性。

2. 11. 5. 2 撤消操作

可以按两种方式撤消以前执行的操作：

- 选择“编辑” > “撤消”。

- 单击“撤消”。

2.12 使用“查找并替换”

可以使用“查找”在 Universe 窗格和结构窗格中查找字符或文本字符串。可以使用“查找并替换”在 Universe 中的任何结构的名称和说明内查找并替换字符或文本。

2.12.1 使用“查找”

可以搜索 Universe 窗格和结构窗格中的 Universe 结构包含的文本。

2.12.1.1 设置“查找”选项

可用的“查找”选项取决于“Universe”窗格还是“结构”窗格处于活动状态。

可以设置以下用于查找字符串的搜索选项：

选项	选项可用	说明
查找内容	当“Universe”或“结构”窗格处于活动状态时	要搜索的文本字符串。
区分大小写	当“Universe”或“结构”窗格处于活动状态时	在搜索中包括大写和小写字符匹配项。
全字匹配	当“Universe”或“结构”窗格处于活动状态时	匹配整个字符串。

选项	选项可用	说明
也搜索名称	当“Universe”窗格处于活动状态时	选中时，搜索类和对象名称或预定义的条件名称（只限这些名称）。 清除时，在搜索中不包括类、对象名称或预定义的条件名称。
也搜索说明	当“Universe”窗格处于活动状态时	选中时，在搜索中包括 Universe 结构的所有说明。
也搜索 SQL	当“Universe”窗格处于活动状态时	选中时，在搜索中包括对象、联接和其他 Universe 结构的 SQL 定义。

2.12.1.2 在 Universe 中搜索

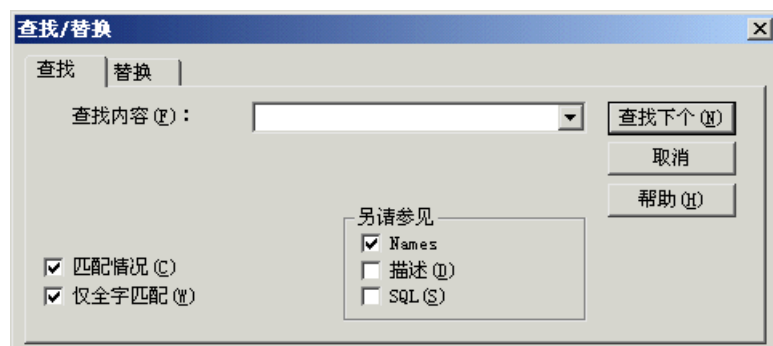
在 Universe 中搜索：

- 1 单击“Universe”窗格或“结构”窗格。

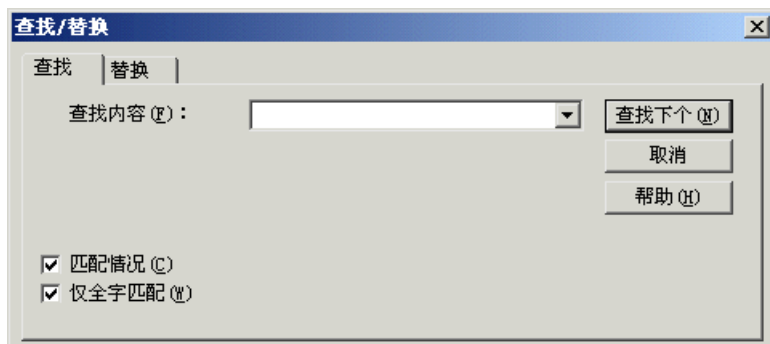
您想在此窗格中查找字符串。

- 2 选择“编辑” > “查找”。

将出现“查找并替换”框。以下是活动“Universe”窗格对应的“查找并替换”对话框。



以下是活动“结构”窗格对应的“查找并替换”对话框。



- 3 在“查找内容”文本框中键入字符或字符串。
- 4 选择或清除搜索选项文本框。
- 5 单击“查找下一个”。

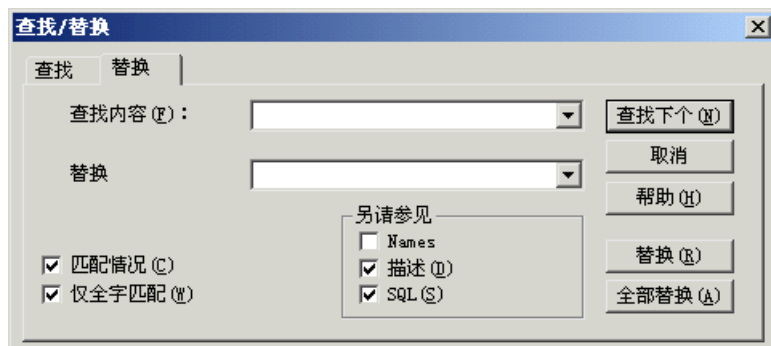
在“Universe”窗格中找到字符或字符串时，会突出显示对象。在对象说明或 SQL 定义中找到实例时，会自动打开对象的属性框，并突出显示字符或字符串。

- 6 单击“查找下一个”以搜索所搜索的字符串的另一个实例。
- 7 单击“取消”关闭“查找并替换”框。

2.12.1.3 在 Universe 中搜索并替换

在 Universe 中搜索并替换字符或字符串：

- 1 选择“编辑” > “替换下一个”。
将出现“查找并替换”框。
- 2 在“查找内容”文本框中键入字符或字符串。



- 3 在“替换”框中键入字符或字符串。这是要替换“查找内容”框的内容实例的文本项。
- 4 选择或清除搜索选项文本框。
- 5 如果要在每次找到实例时替换一个文本项，则单击“替换”。

或

单击“全部替换”以自动替换 Universe 中的所有实例。

如果逐个替换找到的项目，则在某个对象说明中出现项目时，该对象的属性框会自动打开并成为活动框。需要单击“查找并替换”框才能继续搜索。

2.12.2 使用“快速查找”

可以在 Universe 窗格底部的搜索框中键入搜索字符串的第一个字母，搜索活动窗格。

如果“Universe”窗格处于活动状态，则对类和对象名称执行搜索。

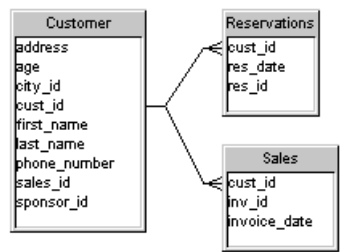
如果“结构”窗格处于活动状态，则对表名称执行搜索。

2.13 组织表显示

本节说明您可以用于组织和处理结构窗格中的表的图形功能。在第 143 页上的 [“创建包含表和联接的模式”](#)一章中，说明了用于设计模式的设计方法，以及若要在“结构”窗格中成功创建模式而需了解的知识。

2. 13. 1 表是如何表示的？

在“结构”窗格中，表用矩形表示。表的名称显示在矩形上半部分的条形区域中。矩形中的项目列表代表了表的列。连接表的线条代表联接。



2. 13. 2 处理表

可以执行以下操作来处理“结构”窗格中的表：

2. 13. 2. 1 选择表

可以按以下所述选择表：

选择对象	执行以下操作...
一个表	单击表。

选择对象	执行以下操作...
几个表	<ul style="list-style-type: none">• 按住鼠标左键并在几个表的周围画一个选择框。• 按住 Shift 键并单击多个表。
所有表（同时选择）	选择“编辑” > “全选”。

要撤消选择，将指针放在表的外面，然后再次单击。

2.13.2.2 删除表

删除表：

- 1 选择一个表。
- 2 执行以下操作之一：
 - 单击“标准”工具栏上的“剪切”按钮。
 - 选择“编辑” > “剪切”。
 - 按 Delete。

2.13.3 使用列表模式

可以使用列表模式来列出活动 Universe 中使用的表、联接和上下文。在列表模式下，Universe 设计工具的“结构”窗格上方将会增加三个窗格。这些窗格标记为“表”、“联接”和“上下文”。

可以按以下方式使用列表模式：

操作	结果
单击在任一个列表模式窗格中列出的元素。	元素在“结构”窗格中突出显示。
选择“结构”窗格中的表、联接或上下文。	在“列表”窗格中列出的对应元素会突出显示。
双击“表”窗格中的某个表名称。	“重命名表”框将出现。可以重命名表并视数据库而定编辑表所有者和限定符。
双击“联接”窗格中的某个联接名称。	该联接的“编辑联接”框将出现。可以编辑联接属性。
双击“上下文”窗格中的某个上下文名称。	“编辑上下文”框将出现。可以通过按 Ctrl 并单击列表中的联接，将联接添加到选定的上下文中。
单击一个元素，然后单击两个“列表”窗格之间的三角形。	将显示邻近列表窗格中与原始元素相关的元素。所有不相关的元素均被过滤。
单击“列表”窗格和“结构”窗格之间的分隔线，然后向上或向下拖动它。	视拖动方向而定，“列表”窗格将扩大或缩小。

2.13.3.1 使用窗格之间的三角形来过滤列出的元素

出现在窗格之间的小三角形用作显示元素的过滤器。例如：

- 单击“表”窗格中的某个表名称，然后单击指向“联接”窗格的三角形。“联接”窗格现在只显示所选表中的联接。
- 单击“联接”窗格中的某个联接名称，然后单击指向“表”窗格的三角形。“表”窗格现在只显示该联接链接的表。

2.13.3.2 从列表模式返回到普通视图

可以按两种方式删除“列表”视图并返回到普通视图：

- 在列表模式下，选择“查看” > “列表模式”。
- 在列表模式下，单击“列表模式”按钮。

2.13.4 自动排列表

可以按两种方式自动排列结构窗格中的表：

- 选择“查看” > “排列表”。
- 单击“排列”按钮。

2.13.5 更改表的显示

可以显示表的三种不同视图。每种视图都用作有关在表符号中显示多少信息的过滤器。

以下说明了每种视图：

表视图	说明
默认	最多显示每个表中的八列。可以修改此值。有关更多信息，请参阅第 67 页上的 “选择模式显示选项” 一节。

表视图	说明
只限名称	在表符号中只显示表名称。当具有许多表时，此操作会降低“结构”窗格中可能出现的混乱现象。
联接列	在每个表符号中只显示表之间的联接所涉及到的列。这些通常是键列。

以下显示了每种表视图：

2. 13. 5. 1 默认表视图

以下显示了包含首八列的表符号。

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

当表中的列数超过默认列数时，会在最后一列后面显示省略号 (...)。单击表时会显示滚动栏。可以通过向下拖动表的下边框扩大表。

2. 13. 5. 2 只限表名称的视图

可以按以下所述在表符号中只显示表名称：

- 双击某个表。

仅显示表的名称。

2.13.5.3 联接列的表视图

可以按以下所述在表符号中只显示联接列：

- 在“结构”中，双击已存在于仅显示名称视图中的表。该表仅显示联接列。

2.13.5.4 更改所有表的显示

同时更改所有选定表的视图：

- 选择“查看” > “更改表显示”。

2.14 选择模式显示选项

可以定制“结构”窗格中的表、列、联接和基数的形状或外观。

对于结构窗格中的元素的显示，可以选择以下图形选项：

选项	说明
联接形状	可以将联接表示为不同类型的简单线条，或者包含基数指示符（如鸟足状末端）或基数比率的线条。
最佳边	选择此选项时，会自动将符合以下所述的链接两个表的联接评为较佳：显示在一个表的左边或右边，在另一个表的左边或右边结束，并且具有最短的长度。

选项	说明
表	表可以具有 3D 效果、显示别名或显示行数。要显示每个表中的行数，还需要通过选择“查看”>“表中的行数”来刷新行计数。第 72 页上的 “查看数据库表中的行数” 一节中对这一点进行了说明。
列	可以在列旁边显示列数据类型。可以为键列加下划线，而且列在表符号中也可以按左对齐或居中方式显示。
默认列数	可以键入在表符号中显示的列的默认数量。如果表的列数多于默认值，则表符号会在列的列表的结尾显示省略号 (...)。单击表时，表旁会显示一个滚动栏。
以所选内容为中心	“结构”窗格的视图基于计算所得的中点。

2. 14. 1 为“结构”窗格显示设置图形选项

可以按以下所述为“结构”窗格的元素设置图形选项：

- 1 选择“工具”>“选项”。
将出现“选项”对话框。
- 2 单击“图形”选项卡。
将出现“图形”页面。它列出了用于“结构”窗格中的元素的图形选项。
- 3 选择或键入图形显示选项。
- 4 单击“确定”。

2. 14. 1. 1 图形选项的示例

以下列出了一些示例，说明“结构”窗格中的元素的可能图形表示方式，它们是通过使用“选项”对话框中提供的图形选项（“工具” > “选项” > “图形”）实现的。

别名

选择此选项时，对于“结构”窗格中具有别名的表，会在显示它时附带其名称及派生它的表的名称（在括号中）。

“显示行计数”和“显示格式”

选择“显示行计数”时，每个表中的行数显示在每个表符号的底部。需要选择“查看” > “表中的行数”才能在显示行计数之前刷新所有表的行数。

选择“显示格式”时，会在列名称旁边显示一个代表列类型的字母。列类型可以是：

- C 代表字符
- D 代表日期
- N 代表数字
- T 代表长文本
- L 代表 blob（二进制大型对象）。

在“结构”窗格中，数字显示在表左下角的下面，而数据类型显示在列名称的旁边。

2.14.2 查看表和列值

可以查看特定表或列的数据值。在任何表中可以查看的行数缺省为 100。视您的需求而定，可以更改此值以返回更多或更少的行。

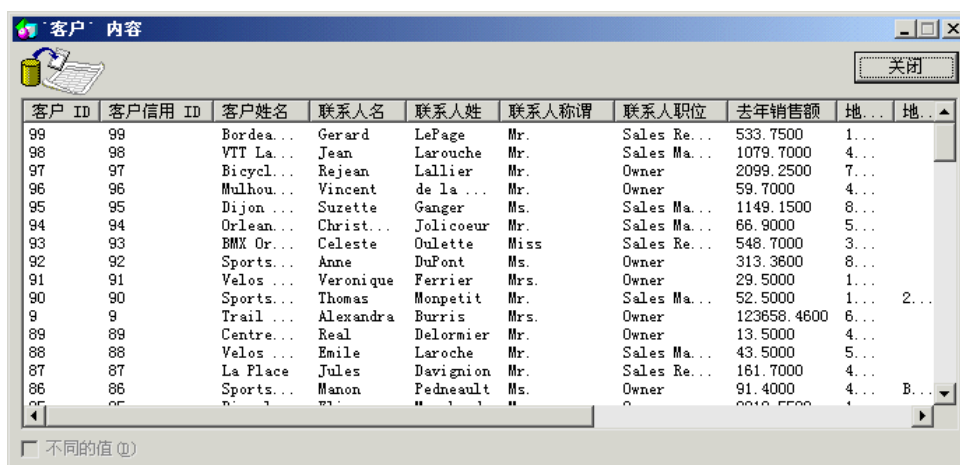
2.14.2.1 查看表的值

查看某个表中的值：

- 1 在“结构”窗格中单击该表。

- 2 选择“查看” > “表值”。

将出现该表的内容对话框，其中列出了表中每列的值。



客户 ID	客户信用 ID	客户姓名	联系人姓名	联系人姓	联系人称谓	联系人职位	去年销售额	地...	地...
99	99	Bordea...	Gerard	LePage	Mr.	Sales Re...	533.7500	1...	
98	98	VTT La...	Jean	Larouche	Mr.	Sales Ma...	1079.7000	4...	
97	97	Bicycl...	Rajean	Lallier	Mr.	Owner	2099.2500	7...	
96	96	Mulhou...	Vincent	de la ...	Mr.	Owner	59.7000	4...	
95	95	Dijon ...	Suzette	Ganger	Ms.	Sales Ma...	1149.1500	8...	
94	94	Orlean...	Christ...	Jolicoeur	Mr.	Sales Ma...	86.9000	5...	
93	93	BMX Or...	Celeste	Oulette	Miss	Sales Re...	548.7000	3...	
92	92	Sports...	Anne	DuPont	Ms.	Owner	313.3600	8...	
91	91	Velos ...	Veronique	Ferrier	Mrs.	Owner	29.5000	1...	
90	90	Sports...	Thomas	Monpetit	Mr.	Sales Ma...	52.5000	1...	2...
9	9	Trail ...	Alexandra	Burris	Mrs.	Owner	123858.4600	6...	
89	89	Centre...	Real	Delormier	Mr.	Owner	13.5000	4...	
88	88	Velos ...	Emile	Laroché	Mr.	Sales Ma...	43.5000	5...	
87	87	La Place	Jules	Davignon	Mr.	Sales Re...	161.7000	4...	
86	86	Sports...	Manon	Pedneault	Ms.	Owner	91.4000	4...	B...

☐ 不同的值 (U)

- 3 如果只想显示非重复的值，请选择“不同的值”复选框。
- 4 单击“关闭”。

2.14.2.2 查看列的值

查看列的值时，可以通过选择“查看” > “放大”来放大列视图。这使您更容易选择列。

可以按以下所述查看单一列的值：

- 1 将指针放在“结构”窗格中某个表的列上。
指针将变为手形符号。
 - 2 右键单击该列，然后从上下文菜单中选择“查看列值”。
- 将出现该列的内容对话框，其中列出了列值。



- 3 如果只想显示非重复的值，请选择“不同的值”复选框。
- 4 单击“关闭”。

2.14.2.3 修改返回的行数的默认值

可以修改在查看表或列值时返回的行数的默认值。在您只想查看表中的一小部分样本值时，这会很有用，因此，您可以将返回的值数限制为很小的数目。

修改为某个表提取的行数：

- 1 选择“工具” > “选项”。
将出现“选项”对话框。
- 2 单击“数据库”选项卡。
将出现“数据库”页面。
- 3 在“表和列值”列表框中，键入数字或使用向上和向下箭头选择数字。
在以下的“数据库”页中，指定了在查看表或列的值时返回 20 行。



4 单击“确定”。

2.14.3 查看数据库表中的行数

可以显示每个表中的行数。可以分两个阶段执行此操作：

- 激活图形选项“显示行计数”（“工具” > “选项” > “图形”），
- 通过选择“查看” > “表中的行数”来刷新所有表的行计数。

可以显示数据库中每个表的行数，也可以为所选的表设置固定的行数，以优化查询性能。这样用户可以在 FROM 子句中控制表的顺序（基于表的权重）。第 74 页上的 [“修改表的行计数”](#) 一节中对这一点进行了说明。

注意：

显示表中的行数与为了查看表或列值而设置返回的行数并不同。

2.14.3.1 显示表中的行数

显示每个表中的行数：

- 1 选择“工具” > “选项”。
- 将出现“选项”对话框。
- 2 单击“图形”选项卡。
- 将出现“图形”页面。
- 3 选择“显示行计数”复选框。
- 4 单击“确定”。
- 5 选择一个或多个表。

或

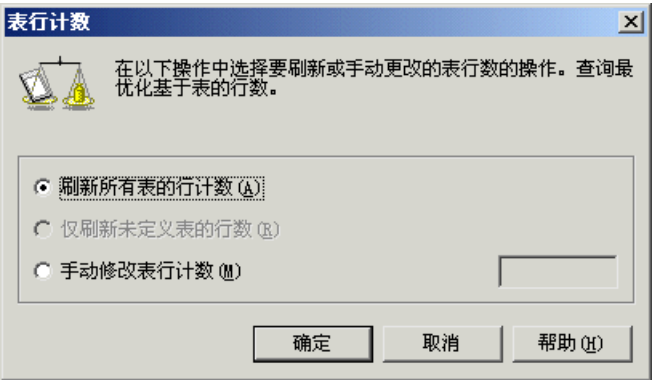
在“结构”窗格中的任意位置单击，并选择“编辑” > “全选”，以便选择结构窗格中的所有表。

注意：

在“结构”窗格中单击时，即激活了与“结构”窗格中的元素相关的菜单项。如果在选择某个菜单项前未在“结构”窗格中单击，则只能使用适用于“Universe”窗格的菜单项。

- 6 选择“查看” > “表中的行数”。

将出现“表行计数”框。



以下说明了此对话框中的选项：

选项	说明
刷新所有表的行计数	刷新“结构”窗格中选定表或所有表的行计数的显示。
只刷新未定义的表的行计数	显示以前未选择的表的行计数。因此，“结构”窗格中的所有表在显示时均附带其行计数。

选项	说明
手动修改表的行计数	让您为“结构”窗格中选定的表或所有表修改行计数。在此选项旁边的文本框中输入新的值。此选项用于优化查询（下一节中将论述此主题）。

- 7 选择“刷新所有表的行计数”单选按钮。
 - 8 单击“确定”。
- 每个选定表的行计数出现在“结构”窗格中每个表符号的左下角下面。

2. 14. 3. 2 修改表的行计数

可以修改表的行计数。以下说明了这样做的两个原因：

修改行计数以...	说明
优化查询	<p>查询优化依据生成的 SQL 的 FROM 子句中的表顺序来进行。具有许多行的表显示在行数较少的表之前。此顺序对于缺少优化程序功能的 RDBMS 尤其重要。</p> <p>通过修改表的行计数，您可以在 FROM 子句中更改表的顺序。</p>
使行计数适应以后的数据容量更改	<p>如果行计数不能反映表将会包含的行数，您可以修改表的行计数。例如，您可以使用一个行计数为 100 的测试表（即使此表将会包含 50,000 行）。</p>

修改一个或多个表的行计数：

- 1 选择“工具” > “选项”。
- 将出现“选项”对话框。
- 2 单击“图形”选项卡。

将出现“图形”页面。

3 选择“显示行计数”复选框。

4 单击“确定”。

5 选择一个或多个表。

或

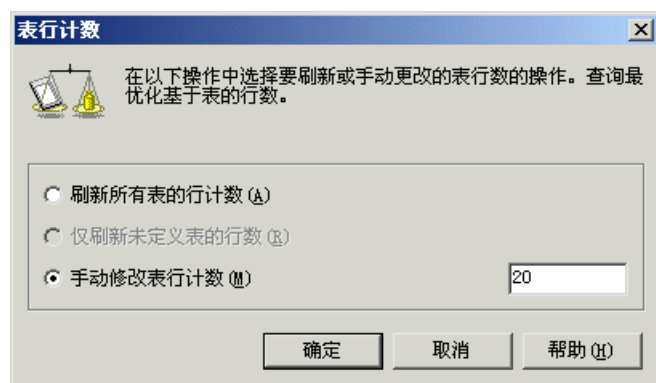
在“结构”窗格中的任意位置单击，并选择“编辑” > “全选”，以便选择结构窗格中的所有表。

6 选择“查看” > “表中的行数”。

将出现“表行计数”框。

7 选择“手动修改表行计数”单选按钮。

8 键入您想为表显示的行数。



9 单击“确定”。

每个选定表的行计数出现在“结构”窗格中每个表符号的左下角下面。

2.15 打印 Universe

Universe 设计工具提供了所有标准的 Windows 打印工具。可以打印模式以及“结构”窗格中的表、列和联接的列表。也可以控制元素和信息在打印的页面上的显示方式。

注意：

要打印出 Universe 定义和模式的 PDF 版本，请将 Universe 另存为 PDF 文件，然后打印该 PDF 文件。有关更多信息，请参阅第 49 页上的 [“将 Universe 定义另存为 PDF”](#) 一节。

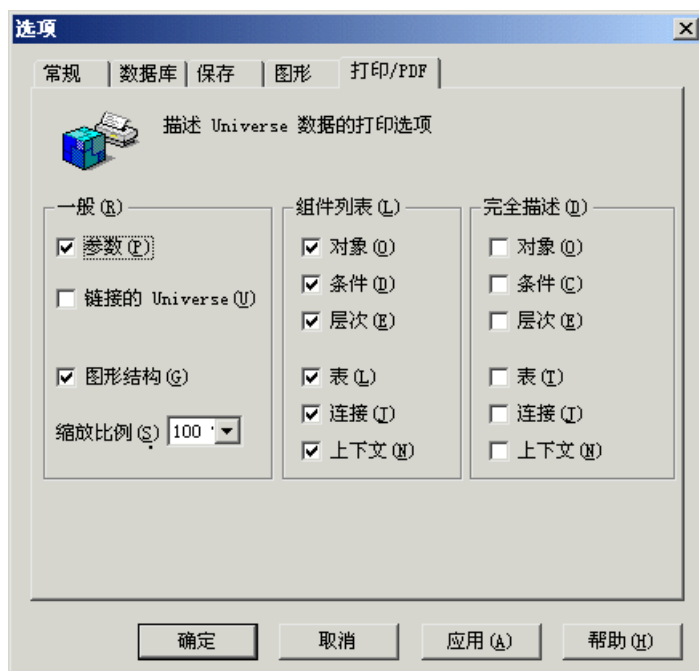
2.15.1 设置打印选项

可以通过“选项”对话框的“打印”页面（“工具” > “选项” > “打印”）选择打印选项。您设置的“打印”选项也适用于在将 Universe 定义另存为 PDF 时保存到 PDF 文件的选项。可以选择以下打印和 PDF 选项：

打印选项	打印...
一般信息	<p>有关以下项目的信息：</p> <ul style="list-style-type: none">• Universe 参数• 链接的 Universe <p>“结构”窗格中的模式的图形结构。可以为此图形选择刻度。</p>
元素列表	<p>Universe 中的元素的列表，按以下一种或多种类型分组：对象、条件、层次、表、联接和上下文。</p>
元素说明	<p>以下元素的说明：对象、条件、层次、表、联接和上下文。</p> <p>说明包括有关元素的属性的详细信息。对于对象，此信息可能包括 SQL 定义、限定和安全访问级别。</p>

为 Universe 设置打印选项：

- 1 选择“工具” > “选项”。
将出现“选项”对话框。
- 2 单击“打印 / PDF”选项卡。
将出现“打印”页面。



- 3 根据需要选择打印选项的复选框。
- 4 单击“确定”。

2.15.1.1 指定页面设置

指定页面设置选项：

- 1 选择“文件” > “页面设置”。
- 将出现“页面设置”表单。
- 2 选择或键入页面设置选项。
- 3 单击“确定”。

2.15.1.2 使用页面预览

可以按两种方式在打印前预览 Universe:

- 选择“文件” > “打印预览”。
- 单击“打印预览”按钮。

2.15.1.3 打印 Universe

可采用两种方式打印 Universe:

- 选择“文件” > “打印”。
- 单击“打印”按钮。

第 3 章 创建 Universe 并设置 Universe 参数

在开始构建 Universe 之前，您必须先创建一个新的 Universe 文件。

创建新的 Universe 文件时，必须定义一个连接参数，以允许 Universe 访问数据库中间件。也可以定义其他参数，这些参数决定 Universe 设计工具如何创建对象、从当前 Universe 到其他 Universe 的链接以及查询限制。

将新的 Universe 另存为 .unv 文件。新的 Universe 不包含类和对象。您在开发 Universe 的过程中，通过设计表模式然后创建映射到数据库结构的对象来创建这些类和对象。

3.1 什么是 Universe 参数？

Universe 参数是您为 Universe 定义的限制及定义，用于标识 Universe 及其数据库连接、指定在使用 Universe 时可运行的查询类型，以及设置有关系统资源的使用的控制。

在创建 Universe 时，用户通过“Universe 参数”对话框（“文件” > “参数”）来定义 Universe 参数。在创建新的 Universe 时，数据库连接是您必须手动选择或创建的唯一参数。

可以随时修改这些参数。可以定义以下 Universe 参数：

参数	说明
定义	Universe 名称、说明、连接参数和信息。它们是标识 Universe 的参数。有关定义和修改此参数的信息，请参阅第 83 页上的 “标识 Universe” 一节。
摘要信息	版本和修订信息、设计员说明和 Universe 统计信息。有关定义和修改此参数的信息，请参阅第 82 页上的 “查看和输入摘要信息” 一节。

参数	说明
策略	指明 Universe 使用的策略。策略是用于从数据库中抽取结构信息的脚本。有关定义和修改此参数的信息，请参阅第 94 页上的 “选择策略” 一节。
控制	指明为系统资源的使用设置的限制。有关定义和修改此参数的信息，请参阅第 98 页上的 “指明资源控制” 一节。
SQL	指明允许最终用户通过“查询”窗格运行的查询类型。有关定义和修改此参数的信息，请参阅第 98 页上的 “指明资源控制” 一节。
链接	指明为链接的 Universe 定义的设置。有关定义和修改此参数的信息，请参阅第 102 页上的 “为链接的 Universe 指明选项” 一节。

3.2 创建新的 Universe

以下过程说明可以如何通过定义 Universe 参数然后保存 Universe 来从头创建新的 Universe。此过程概述了可通过“参数”对话框使用的所有页面。

有关每个步骤的详细信息，应参阅本章中有关参数的各节。

并非一定要在创建 Universe 时定义所有参数。您必须选择一个连接，但可以接受其他参数的默认值，然后在必要时相应地修改这些值。

3.2.1 从头创建新的 Universe

从头创建新的 Universe：

- 1 选择“文件” > “新建”。

“Universe 参数”对话框即会打开到“定义”页面。有关此页面的信息，请参阅第 83 页上的[“标识 Universe ”](#)一节。

注意：

在为 Universe 参数选择任何选项时，“单击这里选择存储过程 Universe”选项都将变灰。将不能选定或取消选定它。如果要更改希望创建的 Universe 类型，请单击“取消”，然后重新开始。

- 键入 Universe 的名称和说明。
- 从“连接”下拉列表框中选择一种连接。

或

- 如果想定义未在下拉列表中列出的新连接，则单击“新建”按钮。有关定义新连接的信息，请参阅第 85 页上的[“定义和编辑连接”](#)一节。

- 2 单击“摘要”选项卡。

将出现“摘要”页面。有关此页面的信息，请参阅第 82 页上的[“查看和输入摘要信息”](#)一节。

- 在“说明”框中键入 Universe 信息。

- 3 单击“策略”选项卡。

将出现“策略”页面。它显示可用于所连接的数据源的策略。有关此页面的信息，请参阅第 94 页上的[“选择策略”](#)一节。

- 从每个“对象”、“联接”和“表”下拉列表框中选择一个策略。

取决于连接的 RDBMS，每个下拉列表框中可能提供了多个策略。

- 4 单击“控制”选项卡。

将出现“控制”页面。有关此页面的信息，请参阅第 98 页上的[“指明资源控制”](#)一节。

- 选择或清除“查询限制”组框中的复选框。
- 为您选择的复选框输入值。

- 5 单击“SQL”选项卡。

将出现“SQL”页面。有关此页面的信息，请参阅第 100 页上的[“指明 SQL 限制”](#)。

- 适当地选择或清除复选框。

- 6 如果想将新的 Universe 与现有的 Universe 链接，则单击“链接”选项卡。

将出现“链接”页面。有关此页面的信息，请参阅第 102 页上的[“为链接的 Universe 指明选项”](#)一节。

- 单击“添加链接”按钮选择要与新的 Universe 链接的 Universe。

- 7 单击“参数”选项卡。

将出现“参数”页面。它列出了可进行设置以优化 SQL 生成的 SQL 参数。有关此页面的信息，请参阅第 103 页上的 [“设置 SQL 生成参数”](#) 一节。

- 8 单击“确定”。

Universe 和结构窗格将在 Universe 设计工具中打开。

- 9 选择“文件” > “保存”。
- 键入 Universe 文件的名称。
 - 单击“保存”。

3.3 查看和输入摘要信息

“摘要”页显示了 Universe 的管理信息。可以使用这些信息帮助您跟踪活动的 Universe 的开发。

“摘要”页显示以下信息：

信息	说明
创建时间	Universe 的创建日期和创建者名称。
修改	上次修改的日期和修改者的名称。
修订	指出已将 Universe 导出到 CMS 的次数的修订次数。
注释	供您自己或另一位设计员参考的 Universe 相关信息。此信息只在 Universe 设计工具中可用。应在“标识”页上的“说明”字段中包括供用户参考的 Universe 相关信息。
统计信息	Universe 中包含的类、对象、表、别名、联接、上下文和层次的数量列表。

3.4 设置 Universe 参数

可以为以下目的而设置 Universe 参数：

- 第 83 页上的 [“标识 Universe ”](#)
- 第 85 页上的 [“定义和编辑连接”](#)
- 第 82 页上的 [“查看和输入摘要信息”](#)
- 第 94 页上的 [“选择策略”](#)
- 第 100 页上的 [“指明 SQL 限制”](#)
- 第 102 页上的 [“为链接的 Universe 指明选项”](#)
- 第 103 页上的 [“设置 SQL 生成参数”](#)

“参数”对话框（“文件” > “参数”）中的每个页面均包含了一种类型的参数。每组参数均在以下各自对应的节中说明。

3.4.1 标识 Universe

每个 Universe 均由以下参数标识：

标识符	使用方
文件名（8 个字符）	要引用 Universe 的文件系统和 Web Intelligence。
长名称（35 个字符）	Web Intelligence 用户。
说明	Web Intelligence 用户。
唯一的数字 ID	要标识 Universe 的 CMS。在第一次将 Universe 导出到 CMS，将为其分配此数字。

在创建 Universe 时，通过“Universe 参数”对话框的“定义”页定义名称和说明参数。您可以随时修改 Universe 标识参数。

您也通过此页定义数据库连接。

有关定义新连接的信息，可以参阅第 85 页上的[“定义和编辑连接”](#)一节。

可以为 Universe 定义以下标识参数：

标识参数	说明
名称	Universe 名称。向 Web Intelligence 用户标识 Universe。注册表支持的名称字符由总管理员定义。字符支持依赖于 RDBMS。
说明	Universe 用途和内容的说明。可选字段。此说明可由 Web Intelligence 用户查看，所以此字段中的信息可提供有关 Universe 的作用的有用信息。
连接	参数的命名集合，用于定义 Web Intelligence 如何访问数据库文件中的数据。所有可用的连接均显示在“连接”下拉列表框中。也可以创建新的连接。

3.4.1.1 修改 Universe 标识参数

修改 Universe 标识参数：

- 1 选择“文件” > “参数”。
或
单击工具栏中的“Universe 参数”按钮。
“Universe 参数”对话框即会打开到“定义”页面。
- 2 键入名称和说明。
- 3 从“连接”下拉列表框中选择一种连接。
- 4 单击“测试”按钮以验证该连接是否有效。

如果收到一则通知您服务器未响应的消息，则该连接无效。可以通过单击“编辑”按钮然后编辑连接属性来更正连接参数。如果此错误仍然存在，请参阅 RDBMS 文档中有关错误消息的章节。

5 单击“确定”。

3.4.2 定义和编辑连接

连接是一种命名的参数集，它确定 Business Objects 应用程序如何访问数据库文件中的数据。连接将 Web Intelligence 链接到中间件。必须具有连接才能访问数据。

创建 Universe 时，必须选择或创建一个连接。可以随时修改、删除或替换连接。

注意：

连接对象还有一个由管理员定义的安全权限，叫做“本地下载连接”。管理员需要定义与连接关联的安全性，例如，定义谁可以本地下载（敏感的）连接信息。

注意：

有关创建、修改和优化连接的完整信息，请参见《数据访问指南》。

可以以下面的方法管理连接：

管理连接的方法	说明
从“连接面板”	通过以下菜单中的命令：“工具” > “连接”。此面板显示可访问的连接列表：登录的 CMS 中的“个人”、“共享”和“安全”。如果 Universe 设计工具已经以独立模式启动，将仅显示“个人”和“共享”连接。连接可显示为简单列表或层次结构，其中可以在子文件夹内显示连接。可以通过此页面删除、编辑和创建新的连接。
从“Universe 参数”对话框	从“Universe 参数”对话框的“定义”页面（“文件” > “参数” > “定义”）。在不存在适合于当前 Universe 的连接时，创建新的连接。也可以编辑连接。

连接包含三个元素：

- 数据访问驱动程序
- 连接和登录参数
- 连接类型

以下各节中说明了各个元素：

3.4.2.1 关于管理连接文件夹

可以使用 Universe 设计工具管理连接文件夹。连接文件夹在显示或管理安全连接的任何位置中显示。可以以下面的方法管理连接：

管理连接的方法	说明
连接面板	通过以下菜单中的命令：“工具” > “连接”。此面板显示可访问的连接列表：登录的 CMS 中的“个人”、“共享”和“安全”。如果 Universe 设计工具已经以独立模式启动，将仅显示“个人”和“共享”连接。连接可显示为简单列表或层次结构，其中可以在子文件夹内显示连接。可以通过此页面删除、编辑和创建新的连接。验证之后，将在 CMS 中自动提交从“连接面板”运行的所有操作。
“Universe 参数”对话框	从“Universe 参数”对话框的“定义”页面（“文件” > “参数” > “定义”）。在不存在适合于当前 Universe 的连接时，创建新的连接。也可以编辑连接。

可执行的操作取决于用户权限，但是可用操作如下所示：

- 创建新连接
- 创建新连接文件夹
- 编辑连接或文件夹
- 删除连接或空文件夹
- 重命名连接或文件夹
- 编辑连接的说明
- 查看连接的属性
- 在编辑模式下剪切、复制和粘贴。

3.4.2.2 数据访问驱动程序

数据访问驱动程序是将 Universe 连接到中间件的软件层。

数据访问驱动程序随 Business Objects 产品提供。为每个支持的中间件都提供了一个数据访问驱动程序。在安装 Universe 设计工具时，用户的数据访问密钥决定安装哪些数据访问驱动程序。

创建新的连接时，您要为用于连接到目标 RDBMS 的 RDBMS 中间件选择适当的数据访问驱动程序。

3.4.2.3 连接和登录参数

通过指定以下连接和登录参数来配置数据访问驱动程序。

参数	说明
类型	连接的类型（个人、共享或安全）
名称	标识连接的名称。
用户名	您的数据库用户名。此名称通常由数据库管理员为您分配。
密码	您的数据库密码。此名称通常由数据库管理员为您分配。
在查看时刷新报表期间使用单一登录	选中时，将自动使用用于访问 CMS 的用户名和密码作为数据库登录参数。有关设置单一登录的信息，请参阅《BusinessObjects Enterprise 管理员指南》。
使用与 BusinessObjects 用户帐户关联的数据库凭据	选择此选项后，将会强制用户输入与其 BusinessObjects 帐户关联的数据库用户密码以刷新报表。此选项是在中央管理控制台级别设置的。有关设置此选项的信息，请参阅《BusinessObjects Enterprise 管理员指南》。
数据源/服务	数据源或数据库名称。如果使用了 ODBC 驱动程序，则数据源名称标识目标数据库。如果使用直连驱动程序，则数据库名称标识目标数据库。

注意：

用户可以创建通过 ODBC 到 Excel 文件和文本文件（.csv 格式）的连接。如果希望 Web Intelligence 可以使用通过 ODBC 访问的基于文本文件或 Excel 文件的 Universe，必须为此连接编辑 msjet.prm 文件。

此文件位于以下文件夹中：\$INSTALLDIR\$/BusinessObjects Enterprise 12.0/win32_x86/dataAccess/connectionserver/odbc，其中 \$INSTALLDIR\$ 是安装 Business Objects 应用程序的目录。在 msjet.prm 文件中，按如下所示更改 DB_TYPE 参数：

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

更改为：<Parameter Name='DB_TYPE'>MS Jet</Parameter>

更改完之后，必须停止并重新启动 Business Objects Enterprise 服务器。注意：如果与 Web Intelligence 服务器在同一台计算机中运行 Universe 设计工具，并且要在更改此值之后另外创建基于文本文件或 Excel 文件的 Universe，则必须将此值重置为 <Parameter Name='DB_TYPE'>MS Jet Engine</Parameter>

3.4.2.4 连接类型

确定谁能够使用连接来访问数据的连接类型。Universe 设计工具将自动存储用户在工作会话期间创建的所有连接。下次启动会话时，您将可以使用这些连接。

可以通过该工具创建以下三种类型的连接：

- 个人
- 共享
- 安全

每种连接类型如下所述：

个人连接

将对数据的访问限制为 Universe 创建者和在上面创建它的计算机。

连接参数存储在 PDAC.LSI 文件中，该文件位于用户配置文件目录的 Business Objects 12.0 文件夹中的 LSI 文件夹内，例如：

C:\Documents and Settings\<用户名>\Application Data\Business Objects\Business Objects 12.0\lsi

这些参数是静态的，因此无法更新。

就 Business Objects 产品安全性而言，个人连接是不安全的。

不要使用个人连接来分发 Universe。可以使用个人连接来访问本地计算机上的个人数据。

共享连接

允许访问所有用户的数据。就 Business Objects 产品安全性而言，这些连接是不安全的。

连接参数存储在 SDAC.LSI 文件中，该文件位于用户配置文件目录的 Business Objects 12.0 文件夹中的 lsi 文件夹内，例如：

C:\Documents and Settings\<用户名>\Application Data\Business Objects\Business Objects 12.0\lsi

安全连接

- 集中管理并控制对数据的访问。它是最安全的连接类型，应加以使用来保护对敏感数据的访问。
- 可以使用 Universe 设计工具创建安全连接。
- 如果要通过 CMS 分发 Universe，您必须使用安全连接。
- 可以随时使用并更新安全连接。

3.4.2.5 针对个人连接和共享连接设置密码

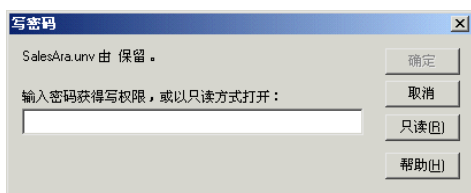
可以在任何具有个人或共享连接类型的 Universe 上设置密码。通过使用密码，可以在没有资源库的环境中保护 Universe 免遭未授权用户的访问。

注意：

如果忘记了密码，则无法恢复 Universe 文件。应备有 Universe 密码的备份文件。

您可以设置两个可用于密码的不同选项：

- “保护密码”会显示一个对话框；它只是提示用户输入密码。如果密码正确，则会打开 Universe。
- “写保护密码”会显示以下对话框：



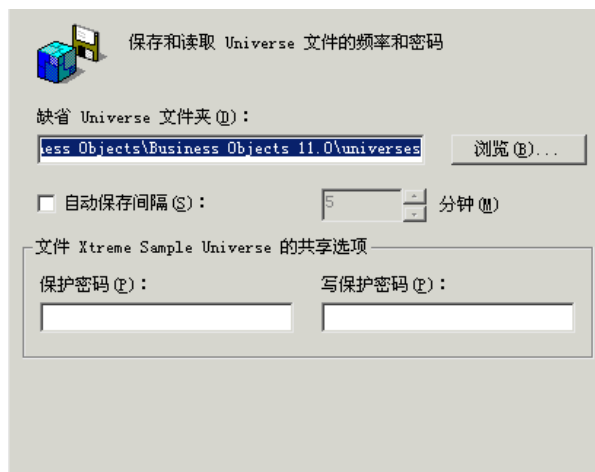
用户随后能以只读模式打开 Universe，或通过输入正确的密码以读写模式打开 Universe。

使用个人或共享连接时设置密码：

1. 选择“工具” > “选项”
将出现“选项”对话框。

- 2 单击“保存”选项卡。

将出现“保存”页面。



- 3 在“保护密码”或“写保护密码”文本框中键入密码。最多可以输入 40 个字母数字字符。
- 4 单击“确定”。

3.4.2.6 在用户的 DBPass 更改之后访问数据库

当数据库的用户密码被更改之后，BusinessObjects 管理员可以允许使用 BusinessObjects 用户登录信息（名称和密码）继续访问数据。

设置了以下参数后，BusinessObjects 用户可以在更改了数据库密码之后继续访问数据，而无需与 BusinessObjects 管理员联系：

- 在中央管理控制台中，必须选定“登录时启用并更新用户的数据源凭据”复选框。
- 在 Universe 设计工具的“新建连接”向导的“定义新连接”页面中，必须选定“使用与 Business Objects 用户帐户关联的数据库凭据”和“在查看时刷新报表期间使用单一登录”复选框。

选中上述复选框后，更新的 DBUser 和 DBPass 参数会自动与 BusinessObjects 用户帐户关联。

注意：

DBUser 和 DBPass 是静态参数，必须在中央管理控制台中进行更新。如果更改了数据库密码，则建议为中央管理控制台中的每个用户帐户更新密码。

3.4.2.7 定义新的连接

使用“新建连接”向导可以定义新的数据库连接。可通过以下位置访问此向导：

- “Universe 参数”对话框的“定义”页（“文件” > “参数” > “定义”）。在不存在可用于 Universe 需访问的数据的连接时，通常要定义新的连接。
- “连接”列表（“工具” > “连接”）。

可以使用连接向导为连接设置高级和定制的参数。有关创建、编辑和优化连接的完整信息，请参阅《数据访问指南》。

当使用“Universe 参数”对话框创建连接时，“Universe 参数”对话框出现，并且新连接列在“连接”框中。

当使用“连接”对话框创建连接时，该连接将出现在列表中。

相关主题

- 第 452 页上的[“启动“新建连接”向导”](#)

3.4.2.8 查看可用连接

可以在“连接”列表中查看所有已存储的可用连接。可以编辑现有的连接和创建新的连接。

查看可用连接：

1. 选择“工具” > “连接”。

将出现“连接”列表。它显示当前 Universe 可用的所有连接。

2. 单击“取消”关闭对话框。

可以通过“连接”对话框编辑连接。

只有当您在联机模式中工作时才能编辑安全连接。可以在任意模式中修改个人和共享连接。

无法修改现有连接的名称。

3.4.2.9 编辑连接

编辑连接：

- 1 选择“工具” > “连接”。
- 将出现“连接”列表。
- 2 在可用连接的列表中单击一个连接名称。
- 3 单击“编辑”。
- 将出现该连接的“登录”页。
- 4 如有必要，在“数据源或服务”框中键入新的数据源或数据库名称。
- 5 如有必要，键入对登录参数的修改。
- 6 单击“下一步”。
- 将出现“执行测试”页面。
- 7 单击“测试数据源”以验证修改后的连接。
- 8 单击“下一步”以移到“高级”和“定制”页面。可以在必要时修改参数。也可以接受默认值或现有的值。
- 9 在“定制”页中单击“完成”以将更改应用到连接。

3.4.2.10 删除连接

可以从“连接”列表中删除连接。只有当您在联机模式中工作时才能删除安全连接。可以在任意模式中删除个人和共享连接。

删除连接：

- 1 选择“工具” > “连接”。
- 将出现“连接”列表。
- 2 在列表中选择连接名称。
- 3 单击“删除”。

将出现一个确认框。

- 4 单击“是”。

即会从列表中移除该连接。

3.4.2.11 添加新的连接

可以通过“连接”页添加新的连接，方法是选择“工具” > “连接”，单击“添加”按钮，然后按照“定义新连接”向导进行操作。有关按照此连接向导进行操作的完整说明位于第 93 页上的[“添加新的连接”](#)一节中。

3.4.3 设置 Universe 摘要参数

“摘要”页显示了 Universe 的管理信息。可以使用这些信息帮助您跟踪活动的 Universe 的开发。

“摘要”页显示以下信息：

信息	说明
创建时间	Universe 的创建日期和创建者名称。
修改	上次修改的日期和修改者的名称。
修订	指出已将 Universe 导出到资源库的次数的修订次数。
注释	供您自己或另一位设计员参考的 Universe 相关信息。
统计信息	Universe 中包含的类、对象、表、别名、联接、上下文和层次的数量列表。

3.4.4 查看和输入摘要信息

- 1 选择“文件”>“参数”
或
单击“参数”工具。
将出现“Universe 参数”对话框。
- 2 单击“摘要”选项卡。
将出现“摘要”页面。
- 3 在“说明”文本框中键入说明。
- 4 单击“确定”。

3.4.5 选择策略

策略是自动从数据库或平面文件中抽取结构信息的脚本。策略有两个基本作用：

- 自动检测联接和基数（联接策略）
- 自动创建类、对象和联接（对象和联接策略）

如果想根据数据库中的 SQL 结构自动检测和创建 Universe 中的结构，则策略会很有用。

注意：

自动创建 Universe 结构的策略不一定要是设计和创建 Universe 这一过程的基本部分。如果您要快速创建 Universe，则这些策略会很有用，因为它们允许您使用数据库或数据库设计工具中已存在的元数据信息。但是，如果您通过创建对象和联接（基于直接来自用户需求分析的关系）来构建 Universe，则将可能无法使用策略提供的自动创建能力。

在 Universe 设计工具中，可以指定两种类型的策略：

策略	说明
内置策略	此工具随附的默认策略。无法定制内置策略。
外部策略	用户定义的脚本，包含与内置策略同一类型的信息，但经过定制以便优化从数据库检索的信息。

3.4.5.1 选择策略

选择策略：

- 1 选择“文件” > “参数”。
- 或
- 单击“参数”工具。
- 将出现“Universe 参数”对话框。
- 2 单击“策略”选项卡。
- 将出现“策略”页面。
- 3 从“对象”、“联接”或“表”下拉列表框中选择一个策略。
- 4 单击“确定”。

3.4.5.2 使用内置策略

内置策略是 Universe 设计工具随附的默认策略。为所有支持的数据库都提供了内置策略。无法修改这些策略。在策略下拉列表中，内置策略默认情况下将出现在外部策略的前面。

可以将内置策略用于以下目的：

策略	用于...
对象	在表模式中创建表时，会自动创建默认类和对象。*

策略	用于...
联接	<ul style="list-style-type: none"> 在表模式中创建表时，会自动提取默认的联接。* 在创建联接时会自动插入基数。* 自动检测表模式中的联接。选择“工具” > “自动检测” > “检测联接”时，Universe 设计工具使用此策略来自动检测候选联接。可以选择是否实现联接。 自动为表模式中的现有联接检测和插入基数。选择“工具” > “自动检测” > “检测基数”时，Universe 设计工具使用此策略为表模式中选定的联接检测基数。
表	过滤可用于表浏览器中的表的信息。

* 必须通过“选项”对话框的“数据库”页激活策略的这些自动创建用法。

使用对象策略

对象策略只用于在您将表添加到表模式时自动创建类和对象。要使用此策略，必须通过“选项”对话框的“数据库”页激活它。有关更多详细信息，请参阅第 96 页上的 [“使用策略的自动创建功能”](#) 一节。

使用联接策略

选定的联接策略决定了 Universe 设计工具如何自动检测表模式中的基数和联接。

视数据库而定，列表中可能有一个或多个联接策略。例如，在使用 Oracle 数据库时，可以指定联接策略根据匹配的列名称或匹配的列号名称自动检测联接。

如果不选择策略，Universe 设计工具则使用与列名称匹配的默认连接策略来检测联接。无需激活所选的联接策略就能使用它检测联接。在您选择检测表模式中的联接或基数时，始终会使用策略。

联接策略也用于自动创建联接以及在创建联接时实现基数。要使用此策略的默认的自动创建功能，必须通过“选项”对话框的“数据库”页激活它。有关更多详细信息，请参阅第 96 页上的 [“使用策略的自动创建功能”](#) 一节。

使用表策略

选定的表策略会读取数据库表的结构。视策略而定，策略可决定在表浏览器中显示哪一类信息，例如，列数据类型和说明。

3.4.5.3 使用策略的自动创建功能

默认情况下，并未激活策略的自动创建和插入功能。要使用这些功能，必须选择与要在创建对象或联接时应用的策略相对应的“默认创建”复选框。这些复选框列在以下所示的“选项”对话框的“数据库”页（“工具” > “选项” > “数据库”）上。

下面说明了“数据库”页上的每个默认创建选项：

选项	清除时	选择时
抽取表的联接	必须手动创建联接。如果选择“工具” > “自动检测” > “检测联接”，Universe 设计工具会使用该策略来检测联接并推荐候选联接。可以选择是否实现候选联接。	检索表（具有按照选定的联接策略链接它们的联接）。
检测联接中的基数	必须手动定义基数。如果选择“工具” > “自动检测” > “检测基数”，Universe 设计工具会使用该策略为选定的联接检测和实现基数。	在创建联接时检测和实现联接中固有的基数。
通过表创建默认类和对象	必须手动创建类和对象 — 可通过在“Universe”窗格中直接创建，或通过将表或列从“结构”窗格拖到“Universe”窗格中来创建。	当将表添加到“结构”窗格中时，会自动在“Universe”窗格中创建默认类和对象。类对应于表名称，而对象对应于列名称。此选项会用空格替换所有下划线字符（_）

为策略选择默认的创建选项：

- 1 选择“工具” > “选项”。
将出现“选项”对话框。
- 2 单击“数据库”选项卡。
将出现“数据库”页面。
- 3 选择对应于要用于策略的默认创建功能的复选框。
- 4 单击“确定”。

3.4.5.4 设置要查看的行数

在“数据库选项”对话框中，还可以指明从数据库的各表中查看的最大行数。用户不能限制从数据库检索的物理行数，但可以修改在查看表或列值时一次可查看的列数默认值。这只适用于在 Universe 设计工具中返回的行，不适用于 Web Intelligence 中运行的查询。

设置可查看的行数：

- 在“获取的最大行数”选项的文本框中输入一个值。也可以单击一次或多次向上或向下箭头，增加或减少默认值（100）。

3.4.5.5 使用外部策略

外部策略是一种用户定义的 SQL 脚本，它采用定义的输出结构来执行定制的自动 Universe 创建任务。外部策略存储在一个外部 XML 策略文件（<RDBMS>.STG）中。此文件中的 SQL 脚本将和其他策略一起出现在“策略”页面上的下拉列表中。

外部策略包含与内置策略相同类型的信息，但常常经过定制，以允许 Universe 设计工具检索特定类型的数据库信息，或优化从数据库中检索信息的方式。

有关定义外部策略的完整信息，请参阅第 410 页上的 [“使用外部策略定制 Universe 创建”](#) 一节。

3.4.6 指明资源控制

Universe 设计工具提供了一系列可用于控制系统资源使用的选项。

注意：

如果在“限制预览”对话框中查看此选项卡，则应用于限制的修改参数显示为红色。

3.4.7 哪些系统资源选项可用？

可以指定以下有关系统资源的限制：

查询限制	说明
限制结果集的大小为指定的值	在查询中返回的行数被限制为您指定的数量。这限制了返回的行数，但并不限制 RDBMS 处理查询中的所有行。它只在 RDBMS 开始发送行时限制返回的行数。
限制执行时间为指定值	查询执行时间被限制为您指定的分钟数。这限制了将数据发送到 WebIntelligence 的时间，但不会停止数据库上的进程。
限制长文本对象的大小为指定值	<p>为长文本对象指定最大字符数。</p> <p>未选中该复选框时，参数未被激活。它会自动设置为默认的最大值（1000）。要确保您允许大于默认值的结果，则必须选中此复选框并且输入值。</p>

3.4.8 输入资源控制信息

- 1 选择“文件”>“参数”。

或

单击“参数”工具。

将出现“Universe 参数”对话框。

- 2 单击“控制”选项卡。

将出现“控制”页面。

- 3 在“查询限制”组框中选择一个复选框。

在对应于所选“查询限制”选项的文本框中键入一个值。可以单击文本框末尾的向上和向下箭头，以增加或减少所输入的值。

- 4 单击“确定”。

3.4.9 限制生成多个 SQL 语句的查询的执行时间

您为查询执行指定的时间限制是某个查询的总执行时间。如果该查询包含多个 SQL 语句，则会为每个语句分配一个执行时间（等于查询总执行时间除以语句数），因此查询中的每个语句都具有相同的执行时间。

如果某个语句要求的运行时间比其他语句长很多，则它可能无法完成，因为它的执行时间与其在查询中获分配的执行时间不对应。

当您为多个 SQL 语句指定执行时间限制时，您需要考虑运行时间最长的单个语句的正常执行时间，并将此值乘以查询中语句的个数。

3.4.10 指明 SQL 限制

对于最终用户可通过 Web Intelligence 中的“查询”窗格以公式表达的查询，可以对它们的类型设置控制。

可以为以下有关查询生成的几个方面指明控制：

- 在各查询中使用子查询、运算符和复合操作数。
- 多个 SQL 语句的生成。
- 阻止或警告有关笛卡儿积的出现。

以下各节中说明了上述每种控制：

3.4.10.1 查询控制

可以为各查询设置以下控制：

选项	说明
允许使用“并”、“交”和“减”运算符	使最终用户能使用数据集运算符（并、交和减）合并查询，以获得一个结果集。

3.4.10.2 多个 SQL 语句控制

可以设置以下控制，以决定如何处理多个 SQL 语句：

选项	说明
为每个上下文使用多个 SQL 语句	使最终用户在使用一个上下文时能创建包含多个 SQL 语句的查询。如果您在 Universe 中有任何上下文，请选择此选项。
为每个度量使用多个 SQL 语句	<p>查询包含从不同表中的列派生的度量对象时，将 SQL 分为几个语句。有关使用此选项的更多信息，请参阅第 256 页上的 “为每个度量使用多个 SQL 语句” 一节。</p> <p>如果度量对象基于相同表中的列，则即使选择了此选项，也不会分割 SQL。</p>
允许选择多个上下文	<p>使最终用户能在多个上下文中的对象上创建查询，以及通过多个上下文生成一个结果集。</p> <p>如果您使用上下文来解决环路、断层陷阱、扇形陷阱或任何其他联接路径问题，则应清除此复选框。</p>

3.4.10.3 笛卡儿积控制

笛卡儿积是一个结果集，它包含查询中所包括的每个表中每一行的所有可能组合。笛卡儿积所代表的结果几乎总不正确。

可以为笛卡儿积的产生设置以下控制。

选项	说明
阻止	选择此选项时，不会执行会产生笛卡儿积的查询。
警告	选择此选项时，会显示一则警告消息，通知最终用户查询会产生笛卡儿积。

3.4.10.4 输入 SQL 限制选项

输入 SQL 限制选项：

- 1 选择“文件” > “参数”。
或
单击“参数”工具。
将出现“Universe 参数”对话框。
- 2 单击“SQL”选项卡。
将出现“SQL”页面。
- 3 选择或清除“查询”和“多重路径”组框中的选项。
- 4 在“笛卡儿积”组框中选择一个单选按钮。
- 5 单击“确定”。

3.4.11 为链接的 Universe 指明选项

“链接”选项卡用于动态链接的 Universe（第 548 页上的 [“部署 Universe”](#) 一章中论述了此主题）。

3.4.12 设置 SQL 生成参数

在 Universe 设计工具中，可以动态配置大多数 RDBMS 共用的某些 SQL 参数，以便优化在 Web Intelligence 产品中使用 Universe 生成的 SQL。

3.4.12.1 使用 Universe 设计工具以前的版本中的参数（PRM）文件

在 Designer 6.5 以前的版本中，Universe 使用的 SQL 生成参数是在一个名为参数（PRM）文件的独立文件中维护和编辑的。PRM 文件中设置的值适合于使用为连接定义的关联数据访问驱动程序的所有 Universe。

现在，可以在单独的 Universe 文件内控制用于优化查询生成的多个 SQL 参数。对于可以在 Universe 设计工具 PRM 文件中设置的查询生成参数，不能再使用 PRM 文件，但 PRM 文件仍可用于特定于数据库的参数。

注意：

有关用于您的数据访问驱动程序的 PRM 文件的更多信息，请参阅《数据访问指南》。可以通过选择“帮助” > “数据访问指南”来访问此指南。

3.4.12.2 在 Universe 设计工具中动态设置 SQL 参数

可以在“Universe 参数”对话框（“文件” > “参数” > “参数”）的“参数”选项卡中编辑大多数受支持的 RDBMS 中间件公用的多个参数。

这些参数仅适用于活动的 Universe，并保存在 UNV 文件中。在 Universe 设计工具中修改 Universe 的 SQL 参数时，将使用 Universe 设计工具内定义的值，而不是在与用于连接的数据访问驱动程序关联的 PRM 文件内定义的值。

3.4.12.3 编辑 SQL 生成参数

可以修改在产品中使用 Universe 确定 SQL 生成的 SQL 参数的值。

编辑 SQL 生成参数：

- 1 选择“文件” > “参数”。

将出现“参数”对话框。

- 2 单击“参数”选项卡。

将出现“参数”页面。



- 3 按如下方式编辑、添加或删除参数：

目的	需执行的操作
添加新参数	<ul style="list-style-type: none"> 单击列表中的任何参数。 在“名称”框中键入名称。 在“值”框中键入值。 单击“添加”。 新值即会出现在列表的底部
更改名称或值	<ul style="list-style-type: none"> 单击列表中的某个参数。 在“名称”框中键入新的名称。 在“值”框中键入新的值。 单击“替换”。 <p>值即会被新的定义替换。</p>
删除参数	<ul style="list-style-type: none"> 从列表中单击要删除的参数。 单击“删除”。

4 单击“确定”。

注意：

在某个 Universe 中设置的 SQL 生成参数值仅可用于使用该 Universe 的产品。

3.4.13 关于 SQL 生成参数

以下参考介绍了可影响查询脚本的生成的参数。这些参数分为两组，以字母顺序列出：

- 在 Universe 设计工具或信息设计工具的用户界面中设置的 SQL 参数。这些是大多数数据访问驱动程序公用的 SQL 参数。每个参数对在其中设置该参数的 Universe 有效。
- 在 PRM 文件中设置的 SQL 参数。这些是连接特定参数，列在目标数据访问驱动程序的数据访问参数（PRM）文件中。

相关主题

- 第 106 页上的[“在用户界面中设置的 SQL 参数”](#)
- 第 126 页上的[“在 PRM 文件中设置的 SQL 参数”](#)

3.4.14 在用户界面中设置的 SQL 参数

3.4.14.1 ANSI92

ANSI92 = Yes|No

值	Yes/No
默认值	No
说明	指定生成的 SQL 是否符合 ANSI92 标准。 Yes: 使 SQL 生成符合 ANSI92 标准。 No: SQL 生成的行为依 PRM 参数“OUTER_JOIN_GENERATION”而定。

3.4.14.2 AUTO_UPDATE_QUERY

AUTO_UPDATE_QUERY = Yes|No

值	Yes/No
默认值	No

说明	确定在查询中的对象不可用于用户配置文件时执行什么操作。 Yes: 更新查询, 并从查询中删除对象。 No: 对象保留在查询中。
----	---

3. 4. 14. 3 BACK_QUOTE_SUPPORTED

BACK_QUOTE_SUPPORTED = Yes|No

值	YES: SQL 使用后引号。 NO: SQL 不使用后引号。
默认值	YES 对于 OpenAccess, 为 No。
说明	指定 SQL 是否使用后引号将其中包含空格或特殊字符的表或列名称引起来。
结果	Table name=`我的表`

3. 4. 14. 4 BEGIN_SQL

BEGIN_SQL = <字符串>

值	字符串
---	-----

默认值	空字符串
说明	<p>用于为执行核算、优化和工作负载管理的 SQL 语句添加前缀。此参数应用于任何 SQL 生成，包括文档生成和 LOV 查询。</p> <p>此参数在 Web Intelligence、LiveOffice 和 QaaWS 中受支持。但被 Desktop Intelligence 和 Crystal Reports 忽略。</p> <p>Teradata 的示例：</p> <pre>BEGIN_SQL=SET QUERY_BAND='string' for transaction;</pre> <p>此参数需要提供一个包含一个或多个名称值对（用分号隔开，所有值都必须括在单引号中）的字符串。所有 SQL 语句均以 BEGIN_SQL 之后的参数作为前缀。在此参数中输入的名称值对被写入到 GetQueryBand Pairs 系统表中。</p> <p>三个名称值对的示例：</p> <pre>BEGIN_SQL=SET QUERY_BAND=' UserID=Jones;JobID=980;AppID=TRM' for transaction;</pre> <p>也可以将 @Variable 函数用作名称值对中的值，返回的值将括在单引号内：BEGIN_SQL=SET QUERY_BAND=' USER=' @Variable(' BOUSER');Document=' @Variable(' DPNAME')';' for transaction;</p>

3. 4. 14. 5 BLOB_COMPARISON

BLOB_COMPARISON = Yes|No

值	Yes/No
默认值	No
是否能够编辑?	否

说明	<p>指定当 SELECT 语句中使用 BLOB 文件时，是否能够使用 DISTINCT 语句生成查询。它与查询属性中的 “No Duplicate Row” 设置相关。</p> <p>Yes: 可在查询内使用 DISTINCT 语句。</p> <p>No: 即使启用了查询设置 “No Duplicate Row”，仍然不能在查询内使用 DISTINCT 语句。</p>
----	--

3. 4. 14. 6 BOUNDARY_WEIGHT_TABLE

BOUNDARY_WEIGHT_TABLE = Integer 32bits [0-9]

值	Integer 32bits [0-9 或负整数]
默认值	-1
说明	<p>允许用户在表包含多个行时优化 FROM 子句。</p> <p>如果表大小（行数）超过输入的值，则将该表声明为子查询：</p> <p>FROM (SELECT col1, col2,....., coln, ,....., FROM Table_Name WHERE 简单的条件).</p> <p>简单的条件被定义为不包含子查询。</p> <p>-1、0 或任何负值均表示不使用此优化。</p>
限制	<p>当出现以下情况时，将不会执行优化：</p> <ul style="list-style-type: none">• 运算符 OR 在查询条件中• SQL 中只涉及一个表• 查询包含外部联接• 所优化的表中未定义条件• 所优化的表是派生表。

3. 4. 14. 7 COLUMNS_SORT

COLUMNS_SORT = Yes|No

值	Yes/No
默认值	No
说明	确定列在“结构”窗格的表中的显示顺序。 Yes: 按字母顺序显示列 No: 按从数据库中检索列的顺序显示列

3. 4. 14. 8 COMBINE_WITHOUT_PARENTHESIS

COMBINE_WITHOUT_PARENTHESIS= Yes|No

值	Yes/No
默认值	No
说明	指定当某个查询包含 UNION、INTERSECT 或 MINUS 运算符时是否用圆括号将它括起。与 RedBrick 一起使用。 Yes: 删除圆括号。 No: 保留圆括号。

3. 4. 14. 9 COMBINED_WITH_SYNCHRO

COMBINED_WITH_SYNCHRO = Yes|No

值	Yes No
默认值	No
说明	<p>指定是否允许执行包含 UNION, INTERSECTION 或 EXCEPT 运算符, 并且它的对象在每个子查询中不兼容的查询。</p> <p>Yes: 指定允许执行包含 UNION, INTERSECTION 和 EXCEPT 运算符, 并且它的对象在每个子查询中不兼容的查询。这种类型的查询会产生同步 (报表中的两个块)。</p> <p>No: 指定不允许执行包含 UNION, INTERSECTION 和 EXCEPT 运算符, 并且它的对象在每个子查询中不兼容的查询。当执行查询时, 将显示以下错误消息: “此查询太复杂。某个子查询包含不兼容的对象。” 这是默认值。</p>

3. 4. 14. 10 COMPARE_CONTEXTS_WITH_JOINS

COMPARE_CONTEXTS_WITH_JOINS = Yes|No

值	Yes No
默认值	Yes

说明	<p>指明如何比较上下文。</p> <p>Yes: 系统验证上下文提供相同的联接。</p> <p>No: 系统验证上下文提供相同的表集合。这是默认值。</p>
----	---

3.4.14.11 CORE_ORDER_PRIORITY

CORE_ORDER_PRIORITY = Yes|No

值	Yes No
默认值	No
说明	<p>此参数应用于添加到所链接的派生 Universe 中的类或对象。此参数不应用于核心 Universe 或原始的派生 Universe 中的类或对象。此参数指定希望在 Universe 设计工具中组织新类和对象的方式。</p> <p>另请参见 FIRST_LOCAL_CLASS_PRIORITY 参数。</p> <p>Yes: 指定按以下方式组织类和对象:</p> <ul style="list-style-type: none">• 第一个核心 Universe 类 <p>核心 Universe 对象</p> <p>任何属于第一个核心 Universe 类的派生 Universe 对象</p> <ul style="list-style-type: none">• 第二个核心 Universe 类 <p>核心 Universe 对象</p> <p>任何属于第二个核心 Universe 类的派生 Universe 对象</p> <ul style="list-style-type: none">• 其它核心 Universe 类...• 派生的 Universe 类和对象 <p>No: 指定类和对象采用派生 Universe 中定义的原始顺序。这是默认值。</p>

3. 4. 14. 12 CORRECT_AGGREGATED_CONDITIONS_IF_DRILL

CORRECT_AGGREGATED_CONDITIONS_IF_DRILL = Yes|No

值	Yes No
默认值	No
说明	<p>仅应用于 Desktop Intelligence。指定 Desktop Intelligence 是否能够聚合查询和条件中的度量。</p> <p>Yes: Desktop Intelligence 能够单独聚合主要查询和条件中的度量（如果查询启用了钻取）。</p> <p>No: Desktop Intelligence 无法单独聚合主要查询和条件中的度量（如果查询启用了钻取）。</p>

3. 4. 14. 13 CUMULATIVE_OBJECT_WHERE

CUMULATIVE_OBJECT_WHERE = Yes|No

值	Yes No
默认值	No

说明	<p>此参数仅应用于过滤的对象。指定如何将对象的 WHERE 子句与基于这些对象的查询条件合并在一起。</p> <p>Yes: 指定使用 AND 运算符将 WHERE 子句与主要查询条件合并。</p> <p>No: 指定将该对象的 WHERE 子句与该对象的条件合并。</p> <p>示例:</p> <p>如果条件是为了找到除 John 之外的所有 France (法国) 客户, 或者除 New York (纽约) 之外的 USA (美国) 城市, 则 SQL 为:</p> <p>Yes:</p> <pre>(customer.first_name <> 'John') OR (city.city <> 'New York' AND customer_country.country = 'France' AND city_country.country = 'USA'</pre> <p>No:</p> <pre>(customer.first_name <> 'John' AND customer_country.country = 'France') OR (city.city <> 'New York' AND city_country.country = 'USA')</pre>
----	--

3. 4. 14. 14 DECIMAL_COMMA

DECIMAL_COMMA = Yes|No

值	Yes No
默认值	No
说明	<p>指定 Business Objects 产品在必要时插入逗号作为小数分隔符。</p> <p>Yes: Business Objects 产品在必要时插入逗号作为小数分隔符。</p> <p>No: Business Objects 产品不会插入逗号作为小数分隔符。这是默认值。</p>

3. 4. 14. 15 DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION

DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION = Yes|No

值	Yes/No
默认值	No
说明	<p>可以不采用默认设置，而是采用优化算法来优化所返回的数组的大小。</p> <p>No: 依据 Universe 运行的所有查询将受益于此优化。</p> <p>Yes: 查询使用默认值设置。</p> <p>此参数也适用于 OLAP 连接。</p>

3. 4. 14. 16 DISTINCT_VALUE

DISTINCT_VALUES = GROUPBY|DISTINCT

值	GROUPBY DISTINCT
默认值	DISTINCT

说明	<p>指定在启用了“不检索重复行”选项时，是否使用值列表和“查询”窗格中的 DISTINCT 或 GROUP BY 子句生成 SQL。</p> <p>DISTINCT: SQL 是使用 DISTINCT 子句生成的，例如：</p> <pre>SELECT DISTINCT cust_name FROM Customers</pre> <p>GROUPBY: SQL 是使用 GROUP BY 子句生成的，例如：</p> <pre>SELECT cust_name FROM Customers GROUP BY cust_name</pre>
----	--

3. 4. 14. 17 END_SQL

END_SQL = 字符串

值	字符串
默认值	<空字符串>
说明	此参数中指定的语句将添加到每个 SQL 语句的结尾。
示例	<p>对于 IBM DB2 数据库。您可以使用以下语句：</p> <pre>END_SQL=FOR SELECT ONLY</pre> <p>服务器将以更快的速度读取数据块。</p> <p>另一个示例：</p> <pre>END_SQL=' write ' UNVID To Usage_Audit.Querieded_universe</pre> <p>Universe ID 被写入到审核表，这可用于记录其它记录，例如，查询的用户和表。</p>

3. 4. 14. 18 EVAL_WITHOUT_PARENTHESES

EVAL_WITHOUT_PARENTHESES = Yes|No

值	Yes No
默认值	No
说明	<p>默认情况下，对于括在括号中的对象 <Class\object>，将使用 SELECT 语句替换函数 @Select(Class\object)。</p> <p>例如，在合并两个 @Select 语句 @Select(object1) *@Select(object2) 时。</p> <p>如果 SQL(object1) = A-B 而 SQL(object2) =C，</p> <p>则运算为 (A-B) * (C)。</p> <p>可通过设置 EVAL_WITHOUT_PARENTHESES = Yes 来避免默认情况下添加括号的行为。则运算为 A - B * C。</p> <p>Yes: 从函数 @Select(Class\object) 的 SELECT 语句中删除括号。</p> <p>No: 在函数 @Select(Class\object) 的 SELECT 语句两边加上括号。</p>

3. 4. 14. 19 FILTER_IN_FROM

FILTER_IN_FROM = Yes|No

值	Yes No
默认值	No

说明	<p>确定是否在 FROM 子句中包括查询条件。只有在另一个 Universe 参数设置 ANSI92 设定为 Yes. 时，此设置才适用。</p> <p>Yes: 编辑外部联接时，在 Universe 设计工具内“高级联接属性”对话框的下拉列表框中选择的默认行为属性将设置为“FROM 中的所有对象”。</p> <p>No: 编辑外部联接时，在 Universe 设计工具内“高级联接属性”对话框的下拉列表框中选择的默认行为属性将设置为“FROM 中没有对象”。</p>
----	--

3. 4. 14. 20 FIRST_LOCAL_CLASS_PRIORITY

FIRST_LOCAL_CLASS_PRIORITY = Yes|No

值	Yes No
默认值	No
说明	<p>此参数仅适用于 Desktop Intelligence</p> <p>只有在 CORE_ORDER_PRIORITY=Yes 时才会考虑。</p> <p>Yes: 派生 Universe 中的类最先列出。</p> <p>No: 派生 Universe 中的对象和子类出现在核心 Universe 的对象和子类之后。</p>

3. 4. 14. 21 FORCE_SORTED_LOV

FORCE_SORTED_LOV = Yes|No

值	Yes No
默认值	No
说明	<p>检索已经过排序的值列表。</p> <p>Yes: 指定对值列表进行排序。</p> <p>No: 指定不对值列表进行排序。</p>

3. 4. 14. 22 INNERJOIN_IN_WHERE

INNERJOIN_IN_WHERE = Yes|No

值	Yes No
默认值	No。必须手动添加参数以激活它。
说明	<p>当 ANSI92 设置为 “Yes” 时，允许用户强制系统在 WHERE 子句中使用所有内部联接生成 SQL 语法。只有查询仅包含内部联接（不包含 “FULL OUTER”、“RIGHT OUTER” 或 “LEFT OUTER” 联接）时，才能执行此操作。</p> <p>Yes: 如果 ANSI92 设置为 “Yes”，除非查询仅包含内部联接，否则系统会在 FROM 子句中生成 ANSI92 联接语法。在此情况下，内部联接将转入 WHERE 子句中。</p> <p>No: 如果 ANSI92 设置为 Yes，系统则在 FROM 子句中生成 ANSI 92 联接语法。</p>

3. 4. 14. 23 JOIN_BY_SQL

JOIN_BY_SQL = Yes|No

值	Yes No
默认值	No
说明	<p>指定如何处理多个 SQL 语句。可以合并多个语句（前提是数据库允许合并）。</p> <p>Yes: 指定将多个 SQL 语句合并起来。</p> <p>No: 指定不将多个 SQL 语句合并起来。这是默认值。</p>

3. 4. 14. 24 MAX_INLIST_VALUES

MAX_INLIST_VALUES = [0-99]

值	Integer: 最小值为 1, 最大值取决于数据库
默认值	-1
说明	<p>允许设置当使用“IN LIST”运算符时最多可在一个条件中输入多少个值。</p> <p>99: 指定当使用“IN LIST ”运算符创建条件时, 最多可输入 99 个值。</p> <p>可输入的最大授权值视数据库而定。</p> <p>值“-1”表示对返回的值数没有限制（数据库的强制限制除外）。</p>

3. 4. 14. 25 OLAP_UNIVERSE

OLAP_UNIVERSE = Yes|No

值	Yes No
默认值	无默认值。
说明	<p>指明是否使用 OLAP Universe。如果 Universe 设计工具使用 OLAP Universe，值将设置为 Yes，而 SQL 参数列表中也会显示该参数。如果 Universe 不是 OLAP Universe，则此参数不会显示在 SQL 参数列表中。</p> <p>Yes: Universe 是一个 OLAP Universe。</p> <p>No: Universe 不是 OLAP Universe。</p>

3. 4. 14. 26 PATH_FINDER_OFF

默认情况下未列出参数。必须手动将参数添加到列表并设置值。

PATH_FINDER_OFF= Yes|No

值	Yes No
默认值	无默认值。必须手动输入该参数。

说明	<p>用于 HPIW，因为联接生成是由数据库完成的。</p> <p>Yes：在查询中不生成联接。</p> <p>No：在查询中生成联接。这是默认行为。</p>
----	---

3. 4. 14. 27 REPLACE_COMMA_BY_CONCAT

REPLACE_COMMA_BY_CONCAT= Yes|No

值	Yes No
默认值	No
说明	<p>在 Universe 设计工具以前的版本中，可以使用逗号来分隔对象 Select 语句中的多个字段。逗号被视为拼接运算符。对于已经按这种方式使用逗号的 Universe，可以通过将 REPLACE_COMMA_BY_CONCAT 设置为 No 来保留这一行为。在该 Universe 设计工具的当前版本中，默认情况下此参数设置为 Yes，因此按这种方式使用逗号的表达式将被自动更改为使用拼接语法。</p> <p>Yes：当发现多字段对象时，用拼接表达式替换逗号。</p> <p>No：按原样保留逗号。</p>

3. 4. 14. 28 SELFJOINS_IN_WHERE

SELFJOINS_IN_WHERE = Yes|No

值	Yes No
---	--------

默认值	No
说明	<p>自联接通常包括在 FROM 子句中。这样可促使系统使用 WHERE 子句中自联接的所有条件来生成 SQL 语法。必须将 ANSI92 参数设置为“Yes”，才会考虑此参数。</p> <p>必须手动将参数添加到列表中以激活它。</p> <p>Yes: 自联接的条件加入 SQL 查询的 WHERE 子句中。</p> <p>No: 根据 ANSI 92 约定生成自联接的语法，并在 SQL 查询的 FROM 子句中将自联接的条件加入到表联接定义的 ON 子句中。</p>

3. 4. 14. 29 SHORTCUT_BEHAVIOR

SHORTCUT_BEHAVIOR = Global|Successive

值	Global Successive
默认值	Successive
说明	<p>指定如何应用快捷联接。在 PRM 文件中，此参数以前被列为 GLOBAL_SHORTCUTS。值也发生相应更改，Yes 改为 Global，No. 改为 Successive。</p> <p>Global: 指定逐个考虑快捷联接。只有当某个快捷联接真正忽略一个或多个表，并且未从后面的快捷联接使用的联接路径中删除表时，才会应用该快捷联接。</p> <p>Successive: 指定将应用所有快捷联接。注意：如果它生成了笛卡儿积，则不会应用快捷联接。</p>

3. 4. 14. 30 SMART_AGGREGATE

SMART_AGGREGATE = Yes|No

值	Yes No
默认值	No
说明	<p>确定聚合表用于智能度量（以聚合表为基础）的方式。这样可以确保正确聚合基于比率的 Universe 对象。默认情况下，如果这些表在一段时间（不同时间段）内不一致，系统会使用聚合表的预计算值，用户可使用此参数确保使用最详细的聚合表。</p> <p>此参数在 Universe 参数列表中不可见（默认情况下没有激活）。在激活此参数（值为“Yes”）之前，Universe 设计员必须手动将它插入到参数列表中。</p> <p>Yes：基于聚合表的智能度量的任何附加分组集查询应基于初始查询的聚合表。</p> <p>No：系统采用最适合的聚合表。</p>

3. 4. 14. 31 STORED_PROC_UNIVERSE

STORED_PROC_UNIVERSE = Yes|No

值	Yes No
默认值	No
说明	<p>在创建包含存储过程的 Universe 时，此值自动设置为 Yes。不要手动更改此值。</p> <p>Yes：所创建/编辑的 Universe 包含存储过程。</p> <p>No：Universe 不包含存储过程。</p>

3. 4. 14. 32 THOROUGH_PARSE

THOROUGH_PARSE = Yes|No

值	Yes No
默认值	No
说明	<p>指定用于“查询”窗格中的默认分析和独立对象分析的方法。</p> <p>Yes: PREPARE, DESCRIBE 和 EXECUTE 语句用于分析对象的 SQL。</p> <p>Prepare+DescribeCol+Execute</p> <p>No: PREPARE 和 DESCRIBE 语句用于分析对象的 SQL。</p>

3. 4. 14. 33 TRUST_CARDINALITIES

TRUST_CARDINALITIES = Yes|No

值	Yes No
默认值	No
说明	<p>允许在出现过度膨胀结果的情况下优化 SQL。</p> <p>Yes: 对于包括度量的查询，导致度量过度膨胀并且不会出现在“结果对象”中的所有条件都将转换为子查询，以确保可能会为度量返回错误结果的表不会包括在查询中。</p> <p>No: 不实施任何优化。</p>

3. 4. 14. 34 UNICODE_STRINGS

UNICODE_STRINGS = Yes|No

值	Yes No
默认值	No
说明	<p>指定当前的 Universe 是否能够处理 Unicode 字符串。只适用于 Microsoft SQL Server 和 Oracle 9。如果 SBO 文件中的数据库字符集设置为 Unicode，则必须修改 SQL 生成才能处理特定 Unicode 列类型（如 NCHAR 和 NVARCHAR）。</p> <p>Yes：依据 PRM 文件中 UNICODE_PATTERN 参数的值，在 SQL 中设置基于字符串的条件格式，例如，对于 MS SQL Server (sqlsrv.prm)：UNICODE_PATTERN=N\$</p> <p>条件 Customer_name='Arai' 将变为</p> <p>Customer_name=N'Arai'。</p> <p>注意：当基于 Unicode 值使用 @Prompt 语法创建提示时，数据类型应为“U”（而不是“C”）</p> <p>No：以标准 SQL 设置基于字符串的所有条件的格式。例如，条件 Customer_name='Arai' 保持为 Customer_name='Arai'</p>

3. 4. 15 在 PRM 文件中设置的 SQL 参数

3. 4. 15. 1 CASE_SENSITIVE

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

说明	指定数据库是否区分大小写。此参数用于 Oracle。
值	YES: 数据库区分大小写。 NO: 数据库不区分大小写。
默认值	NO

3. 4. 15. 2 CHECK_OWNER_STATE

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

说明	指定 SQL 是否检查数据库是否支持按所有者名称列出的表分类。
值	YES: SQL 检查数据库是否支持按所有者名称列出的表分类。 NO: SQL 不检查数据库是否支持按所有者名称列出的表分类。
默认值	YES

3. 4. 15. 3 CHECK_QUALIFIER_STATE

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

说明	指定 SQL 是否检查数据库是否支持按限定符列出的表分类。
值	YES: SQL 检查数据库是否支持按限定符列出的表分类。 NO: SQL 不检查数据库是否支持按限定符列出的表分类。
默认值	YES

3. 4. 15. 4 COMMA

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

说明	指定应使用什么数据库拼接运算符来替换具有以下语法的对象的逗号: Tab.Col1, Tab.Col2。 此参数用于所有数据访问驱动程序。
值	' ' +' '+
默认值	' '
结果	Tab.Col1 ' ' Tab.Col2

3. 4. 15. 5 CONCAT


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

说明	指定拼接运算符。此参数用于所有数据访问驱动程序。
值	双竖线 () 或加号 (+)
默认值	

3. 4. 15. 6 DATE_WITHOUT_QUOTE

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

说明	指定在 SQL 语法中是否将日期引在单引号内。此参数用于 MS Access。
值	YES: 不将日期引在单引号内。 NO: 将日期引在单引号内。
默认值	YES

3. 4. 15. 7 DELIMIT_LOWERCASE

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

说明	指定是否使用引号分隔小写标识符。
值	YES: 使用引号分隔小写标识符。 NO: 不使用引号分隔小写标识符。

3.4.15.8 EXTERN_SORT_EXCLUDE_DISTINCT

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

说明	指定当查询包含 ORDER BY 子句时应用程序是否生成 SELECT DISTINCT。
值	YES: 当查询包含 ORDER BY 子句时不生成 SELECT DISTINCT。 NO: 当查询包含 ORDER BY 子句时生成 DISTINCT。
默认值	YES

3.4.15.9 GROUPBY_WITH_ALIAS

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

说明	指定数据库是否能在 SELECT 语句中创建包含别名的 GROUP BY 子句。
----	--

值	YES: 允许在 SELECT 语句中创建 GROUP BY 子句。 NO: 不允许在 SELECT 语句中创建带有别名的 GROUP BY 子句。
默认值	YES

3.4.15.10 IDENTIFIER_DELIMITER

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

说明	<p>指定以下功能:</p> <ul style="list-style-type: none">在激活了 BACK_QUOTE_SUPPORTED 参数时, 将包含空格或特殊字符的表或列名称用引号引起来。在激活了 DELIMIT_IDENTIFIERS 参数时, 将表或列名称 (不管其字符如何) 用引号引起来。 <p>要使用此参数, BACK_QUOTE_SUPPORTED 或 DELIMIT_IDENTIFIERS 必须设置为 "YES"。这是这两个参数的默认值。</p>
值	<p>" (双引号): 将包含空格或特殊字符的表或列名称用双引号引起来。</p> <p>' (单引号): 将包含空格或特殊字符的表或列名称用单引号引起来。此值只能用于 Microsoft Access。</p>
默认值	"
结果	Table name="我的表"

3.4.15.11 IF_NULL

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

说明	指定带有两个参数的函数。如果第一个参数返回 NULL，则使用第二个参数值。
值	依赖于数据库。
默认值	依赖于数据库。

3. 4. 15. 12 OUTERJOINS_COMPLEX

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

说明	此参数和 OUTERJOINS_GENERATION 参数一起控制外部联接查询的生成。
值	YES NO

3. 4. 15. 13 OUTERJOINS_GENERATION

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

此参数控制默认的外部连接生成行为。可以设置该参数，使

外部联接生成符合 ANSI92 规范。

外部联接生成与 Universe 设计工具以前版本的外部联接生成相同。

注意：

PRM 文件的 OUTERJOINS_GENERATION 参数以下列方式关联到 Universe 的 ANSI92 设置：

如果 PRM 文件的 OUTERJOINS_GENERATION 参数设置为 ANSI92 且 Universe ANSI92 设置已设定为 NO，则 PRM 参数将覆盖该 Universe 设置，外部连接将符合 ANSI92 行为。

- 如果 PRM 文件的 OUTERJOINS_GENERATION 参数设置为 USUAL，则 Universe ANSI92 设置优先，外部联接是否符合 ANSI92 将取决于 Universe ANSI92 设置是 YES 还是 NO。

记忆：

值 ANSI92 将使 REVERSE_TABLE_WEIGHT 参数对于 SQL 生成的优化无效。符合 ANSI92 行为的外部联接决定表在 SQL 语句中的顺序。

说明	<p>指定外部联接的 SQL 语法。</p> <p>值 ANSI 92 在 FROM 子句中生成外部联接。其他值将在 WHERE 子句中生成外部联接。</p> <p>在修改此设置时，您应检查连接属性，以便验证外部联接表达式是否有效，以及基数是否正确。ANSI92 不支持在联接语法中进行任何手动自定义。</p>
值	<p>OUTERJOINS_GENERATION 的主值包括：</p> <p>ANSI92：无论 Universe 设计工具的 ANSI92 参数值是什么，默认外部联接行为都符合 ANSI92 标准。</p> <p>No：不支持外部联接。</p> <p>USUAL：默认外部联接行为与 Universe 设计工具以前的版本相同。如果 Universe 设计工具的 ANSI92 参数设置为“YES”，此行为将被覆盖。</p> <p>其他设置是否可用取决于数据库。请参阅以下默认值。</p>

默认值	<p>ANSI_92: Oracle、MS SQL Server 2005 和 Sybase 的默认值。</p> <p>DB2: IBM DB2 的默认值。</p> <p>FULL_ODBC: Microsoft SQL Server 的默认值。</p> <p>INFORMIX: IBM Informix 的默认值。</p> <p>INGRES: Teradata 的默认值。</p> <p>NO: ODBC 的默认值。</p> <p>USUAL: HP Neoview、Nettezza、IBM Red Brick 和 MS SQL Server 2000 的默认值。</p>
-----	--

OUTERJOINS_GENERATION 参数设置的示例

Setting = USUAL:

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

Setting = DB2:

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

Setting = ODBC:

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

Setting = INFORMIX

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

Setting = FULL-ODBC

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

Setting = ANSI_92:

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

将 OUTERJOINS 用于 Oracle

无论 ANSI92 参数的 Universe 级设置如何，默认的 OUTERJOINS_GENERATION 设置都会影响现有 Universe 的行为。

将现有的 Oracle Universe 行为设置为与 Universe 设计工具以前的版本相同：

- 1 在 PRM 文件中，确保 OUTERJOINS_GENERATION 参数设置为 USUAL。

- 。 2 在 PRM 文件中，将 LEFT_OUTER 和 RIGHT_OUTER 参数设置为 \$(+)。

有关 Universe 设计工具以前版本中的 Universe SQL 参数和 PRM 文件的更多信息，请参阅《Designer 指南》。

3. 4. 15. 14 OVER Clause

```
<Parameter Name="OVER Clause">YES</Parameter>
```

说明	允许 SAP BusinessObjects 应用程序在生成 SQL 时包括 RISQL 函数。数据库支持的 RISQL 函数列在 ANALYTIC_FUNCTIONS 参数中。
值	YES: 应用程序在生成 SQL 时可以包括 RISQL 函数。 NO: 应用程序在生成 SQL 时不能包括 RISQL 函数。
默认值	YES

3. 4. 15. 15 OWNER

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

说明	指定数据库是否支持所有者名称作为表的前缀。
值	YES: 数据库支持使用所有者名称为表加上前缀。 NO: 指定数据库不支持使用所有者名称为表加上前缀。
默认值	YES

3.4.15.16 PREFIX_SYS_TABLE

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

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

说明	指定是否在 Universe 设计工具中显示系统表。
值	MSys: 在 Universe 设计工具表浏览器中隐藏 MS Access 系统表。 RBW_: 在 Universe Designer 表浏览器中隐藏 IBM Red Brick 系统表。 无值: 在 Universe 设计工具表浏览器中显示数据库系统表。
默认值	MSys: MS Access 的默认值 RBW_: IBM Red Brick 的默认值

3.4.15.17 QUALIFIER

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

说明	指定数据库是否支持标识符名称作为表的前缀。
值	YES: 数据库支持使用标识符名称为表加上前缀。 NO: 指定数据库不支持使用标识符名称为表加上前缀。
默认值	取决于 RDBMS。

3. 4. 15. 18 QUOTE_OWNER

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

说明	指定是否应将所有者名称用单引号引起来。仅由 IBM Informix 使用。
值	YES: 用单引号引起来的所有者名称为表名称加上前缀。对于符合 ANSI 的 IBM Informix 数据库而言, 这是必需的。否则, IBM Informix 会将所有者名称转换为大写字符。 NO: 不用单引号引起来的所有者名称为表名称加上前缀。
默认值	YES
结果	SELECT Alias.col (<别名> 是本地别名) FROM 'Owner'.table.col Alias

3. 4. 15. 19 REFRESH_COLUMNS_TYPE

<Parameter Name="REFRESH_COLUMNS_TYPE">0</Parameter>

说明	指定如何刷新列。
值	0: 按所有者名称刷新列。 Q: 按限定符名称刷新列。 T: 按表名称刷新列。

默认值	0: Oracle 的默认值 Q: IBM Red Brick、Sybase、MS SQL Server 和 MS Access 的默认值
-----	--

3. 4. 15. 20 REMOVE_SEMICOLONS

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

说明	指示 SAP BusinessObjects 应用程序的查询面板是否在自由编写 SQL 中删除分号。
值	YES: 查询面板删除分号。 NO: 查询面板不删除分号。

3. 4. 15. 21 REVERSE_TABLE_WEIGHT

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

说明	<p>指定应按哪种顺序生成表。此参数用于 Oracle。此参数也可用于某些其他数据库，但 YES 和 NO 可能颠倒。</p> <p>注意： Teradata 不支持此参数。</p> <p>记忆： 如果 OUTERJOINS_GENERATION 参数设置为 ANSI92，或者 Universe ANSI92 设置设置为 YES，则 REVERSE_TABLE_WEIGHT 参数不会影响对 SQL 生成进行优化。</p>
----	--

值	YES: 按从最小到最大的顺序生成表。 NO: 按从最大到最小的顺序生成表。
默认值	YES

3. 4. 15. 22 UNICODE_PATTERN

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

说明	仅当 Universe SQL 生成参数 UNICODE_STRINGS 设置为 YES 时才适用。随后基于字符串的所有条件的格式会使用该字符串值设置。仅用于 MS SQL Server 和 Oracle。
值	N\$: 用于 MS SQL Server UNISTR(\$): 用于 Oracle

3. 4. 15. 23 USER_INPUT_DATE_FORMAT

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

说明	指定在 SQL 语句的 WHERE 子句中生成的默认日期和小时格式。
----	------------------------------------

值	<p>{\d 'yyyy-mm-dd'}: ODBC 的默认日期格式。</p> <p>'DD-MM-YYYY HH:MM:SS': Oracle 的默认日期和小时格式。</p> <p>'MM/DD/YYYY': Informix 的默认日期格式。</p> <p>'yyyy-mm-dd HH:mm:ss': MS SQL Server 和大部分 IBM DB2 服务器的默认日期和小时格式。</p> <p>'mm/dd/yyyy hh:m:s am/pm': Sybase 的默认日期和小时格式。</p> <p>'yyyy-mm-dd': Sybase 网关的默认日期格式。</p> <p>注意: 如果需要将时间或时间戳变量用于 ODBC, 则必须在 odbc.sbo 文件中将默认日期格式值替换为: {\t 'hh:mm:ss'} 或 {\t\s 'yyyy-mm-dd hh:mm:ss'}。</p>
默认值	请参阅上面的值。

3. 4. 15. 24 USER_INPUT_NUMERIC_SEPARATOR

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

说明	指定生成的 SQL 脚本中使用的默认小数分隔符。
值	“.”（句点）
默认值	“.”

3. 4. 15. 25 DELIMIT_IDENTIFIERS

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

说明	指定是否将数据库标识符引起来。使用 IDENTIFIER_DELIMITER 参数中指定的定界符将标识符引起来。
值	YES: 可以将标识符引起来。 NO: 不能将标识符引起来。
默认值	YES
结果	Table name="my_table"

3. 4. 15. 26 EXT_JOIN_INVERT

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

说明	指定如何在联接表达式中显示外部联接符号。此参数用于 IBM DB2、IBM Informix、Oracle 和 Teradata。
值	YES: 当在 Universe Designer 的“编辑联接”对话框中单击“外部联接”复选框时，外部联接符号将出现在联接表达式中对面的位置。 NO: 当在 Universe Designer 的“编辑联接”对话框中单击“外部联接”复选框时，外部联接符号将出现在创建了外部联接的同一侧。
默认值	YES

3. 4. 15. 27 KEY_INFO_SUPPORTED

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

说明	指定是否能从数据库中检索主要和辅助键定义。
值	YES: 数据库允许从数据库中检索主要和辅助键定义。此参数使 Universe Designer 能够在“结构”窗口中显示键。 NO: 数据库不允许从数据库中检索主要和辅助键定义。
默认值	YES

3. 4. 15. 28 ORDER_BY_STRINGS

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

说明	指定数据库是否可根据字符串列正确处理 ORDER BY 子句。此参数对应于 SAP BusinessObjects Data Federator 的 ORDERBYSTRINGS 功能。如果数据库无法执行处理，Data Federator 查询服务器将执行排序。
值	YES: 数据库可执行排序处理。 NO: 数据库不可执行排序处理。

第 4 章 创建包含表和联接的模式

4.1 概述

本章介绍了如何可以创建一个包含构建对象所需的所有 SQL 结构的模式（Web Intelligence 用户使用这些对象来构建报表）。这些 SQL 结构包括表、列、联接和数据库函数。构建正确的模式是构建满足所有最终用户报表需求的 Universe 的基础。

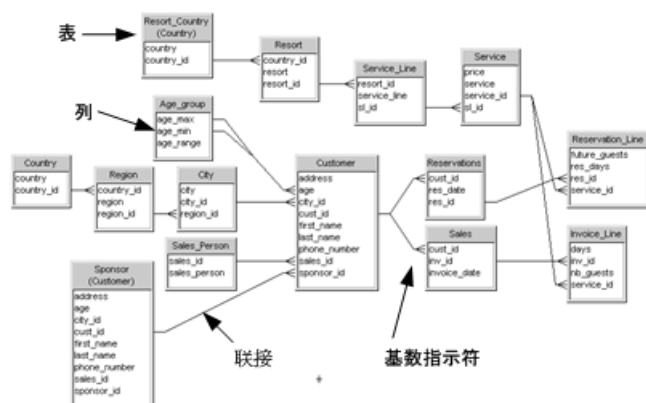
4.2 什么是模式？

模式是数据库结构的图形表示方式。在 Universe 设计工具中，为数据库中 Universe 表示的部分创建模式。

模式包含表和联接。表包含列，而您最终会将这些列映射到最终用户用于创建报表的对象。联接链接表，从而为在多个表上运行的查询返回正确的数据。

通过使用表浏览器从目标数据库中选择表，在“结构”窗格中设计模式。创建联接以链接表。当您为 Universe 设计了模式时，可以使用自动完整性检查功能验证模式。

以下显示了 Beach 这一示例 Universe 的模式：



4.2.1 模式设计是成功的 Universe 的基础

必须很好地设计模式才能产生好的 Universe 设计。您使用表填充模式，而这些表基于与最终用户创建报表所需的对象相对应的列。应通过用户需求分析定义这些对象。您应在数据库中查看可让您创建这些必需对象的表。

4.2.2 模式设计和 Universe 创建过程

创建模式是 Universe 开发周期的实施阶段的第一个子阶段。无需使用 Universe 设计工具即可完成用户分析和计划阶段；但是，创建模式是使用该工具来构建 Universe 的第一个步骤。

下面指明了模式设计阶段出现在一个典型 Universe 开发周期（实现，步骤 1）的哪个位置：

- 准备
 - 1 用户需求分析
 - 2 规划
- 使用 Universe 设计工具的实现
 - 1 设计和测试模式
 - 2 构建和测试 Universe 对象
 - 3 使用资源库部署 Universe
- 维护

- 。 1 根据用户需求或数据源中的更改更新和维护 Universe

4.2.3 模式设计有哪些阶段？

本章论述了模式设计的以下阶段：

- 插入和组织表。
- 创建联接和设置基数。
- 解决诸如环路、断层陷阱和扇形陷阱的联接问题。
- 测试模式的完整性。

4.3 插入表

通过从目标数据库中选择表，并在“结构”窗格中插入表示这些表的符号来开始设计模式。在 Universe 设计工具中，表符号简称为表。

使用表浏览器来选择插入到模式中的表。表浏览器是一个独立的窗口，它显示了目标数据库中可用的表的树形视图。

注意：

选择表之前，您可以指明想用于协助创建 Universe 的策略。有关此主题的更多信息，请参阅第 94 页上的 [“选择策略”](#)。

4.3.1 使用表浏览器

表浏览器是一个独立的窗口，它显示了目标数据库中的表和列的树形视图。使用“表浏览器”可查看和选择数据库中要插入模式内的表。展开表名称旁边的节点可显示表的列。

4.3.1.1 激活表浏览器

默认情况下，表浏览器不可见。若要将表添加到“结构”窗格中时，必须激活表浏览器。可以使用以下列出的任一方法激活表浏览器。

激活表浏览器：

- 选择“插入” > “表”。

或

- 在“结构”窗格的空白空间中双击。

或

- 单击“表浏览器”按钮。

“表浏览器”窗口将出现在“结构”窗格中。

4.3.1.2 从表浏览器插入表

可以使用以下任一方法用“表浏览器”插入一个或多个表：

插入单个表

插入单个表：

- 单击一个表并单击“插入”按钮。

或

- 右键单击一个表，然后从上下文菜单中选择“插入”。

或

- 双击某个表。

或

- 单击一个表，然后将其拖到“结构”窗格中。

该表将出现在“结构”窗格中。

插入多个表

插入多个表：

- 1 按住 Ctrl 键并逐个单击表。
或
- 2 按住 Shift 键并单击连续的一系列表中的第一个表和最后一个表。
多个表即被选中。
- 3 单击“插入”按钮。

或

将表拖入“结构”窗格中。

或

右键单击选中的表，然后从上下文菜单中选择“插入”。

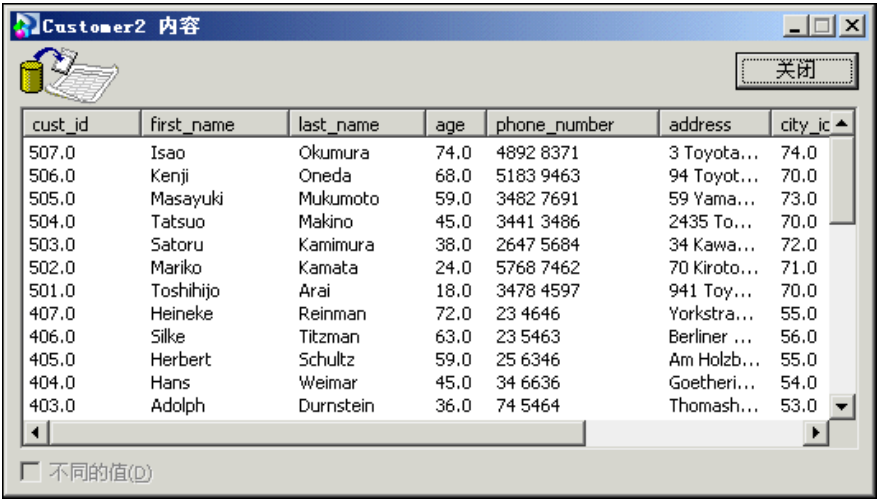
包含自己的所有列的各个表将出现在“结构”窗格中。在表浏览器中，用户插入到 Universe 中的所有表在其名称旁均显示一个复选标记。

4.3.1.3 从表浏览器查看数据

可以使用表浏览器查看表中或单个列中包含的数据。

从表浏览器查看数据：

- 1 在表浏览器中右键单击某个表
或
展开表浏览器中的一个表节点，然后右键单击该表的列。
- 2 从上下文菜单中选择“查看表值”。
或
从上下文菜单中选择“查看列值”。
将出现一个列出表或列中包含的数据的框。



cust_id	first_name	last_name	age	phone_number	address	city_ic
507.0	Isao	Okumura	74.0	4892 8371	3 Toyota...	74.0
506.0	Kenji	Oneda	68.0	5183 9463	94 Toyot...	70.0
505.0	Masayuki	Mukumoto	59.0	3482 7691	59 Yama...	73.0
504.0	Tatsuo	Makino	45.0	3441 3486	2435 To...	70.0
503.0	Satoru	Kamimura	38.0	2647 5684	34 Kawa...	72.0
502.0	Mariko	Kamata	24.0	5768 7462	70 Kiroto...	71.0
501.0	Toshihijo	Arai	18.0	3478 4597	941 Toy...	70.0
407.0	Heineke	Reinman	72.0	23 4646	Yorkstra...	55.0
406.0	Silke	Titzman	63.0	23 5463	Berliner ...	56.0
405.0	Herbert	Schultz	59.0	25 6346	Am Holz...	55.0
404.0	Hans	Weimar	45.0	34 6636	Goetheri...	54.0
403.0	Adolph	Durnstein	36.0	74 5464	Thomash...	53.0

提示：
如果列的宽度不足以查看完整的行值，可通过按组合键 Ctrl-Shift 和 “+” 键来扩宽列。

4. 3. 1. 4 优化表浏览器的性能

从表浏览器将表插入 “结构” 窗格中所用的时间会视以下因素变动：

插入表很慢，原因是...	优化插入表的操作，通过...
数据库中有大量的表。Universe 设计工具会查询系统目录，因此当目录很大时，检索表就可能很慢。	使用想插入到单独的数据库帐户的表构建数据仓库。创建与新的仓库的连接。

插入表很慢，原因是...	优化插入表的操作，通过...
正在自动插入联接和检查所插入的表的基数。	<p>只插入表。可以按如下方式执行此操作：</p> <ol style="list-style-type: none">1 选择“工具” > “选项”。 <p>将出现“选项”对话框。</p> <ol style="list-style-type: none">2 单击数据库选项卡。 <p>此时将显示“数据库”页面。</p> <ol style="list-style-type: none">3 清除下列复选框：“用表抽取联接”和“检测联接中的基数”。 <ol style="list-style-type: none">4 单击“确定”。

4.3.2 排列“结构”窗格中的表

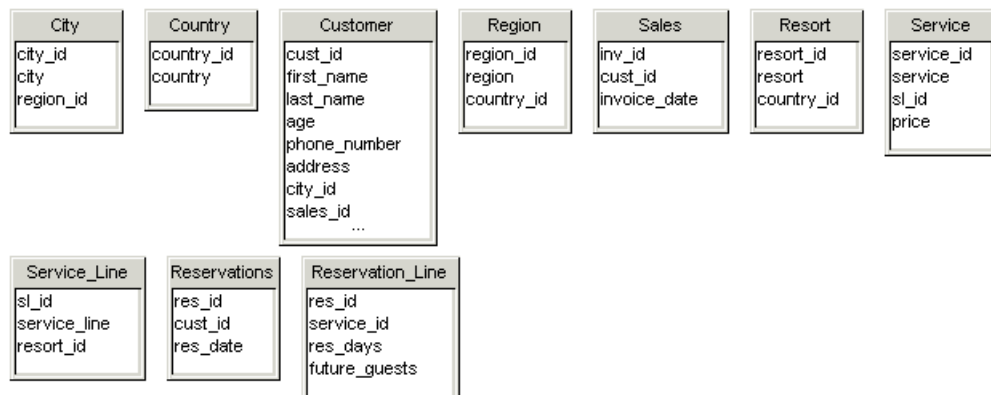
在开始手动重新排列表以创建联接之前，可以自动排列“结构”窗格中的表，以整理初始模式。

4.3.2.1 自动排列“结构”窗格中的表

自动排列表：

- 选择“查看” > “排列表”。

将按有序的方式排列表。



4.4 使用派生表

派生表是您在 Universe 模式中定义的表。您在这些表上创建对象，就像在任何其他表上创建一样。派生表由 SQL 查询在 Universe 级别定义，可用作 Universe 设计工具中的逻辑表。

派生表具有以下优点：

- 减少了返回到文档供分析用的数据量。

可以在派生表中包含复杂的计算和函数。这些操作在将结果集返回到文档之前执行，从而节省了时间和减少了在报表级别对大量数据进行复杂分析的需求。

- 减少了对数据库汇总表的维护。

有时，派生表可以取代保存复杂计算的结果的统计表（通过使用聚合感知将这些结果合并到 Universe 中）。经常维护和刷新这些聚合表要付出很高的代价。派生表可以返回相同的数据并提供实时数据分析。

派生表与数据库视图类似，并具有以下优势：派生表的 SQL 可以包括提示。

4.4.1 添加、编辑和删除派生表

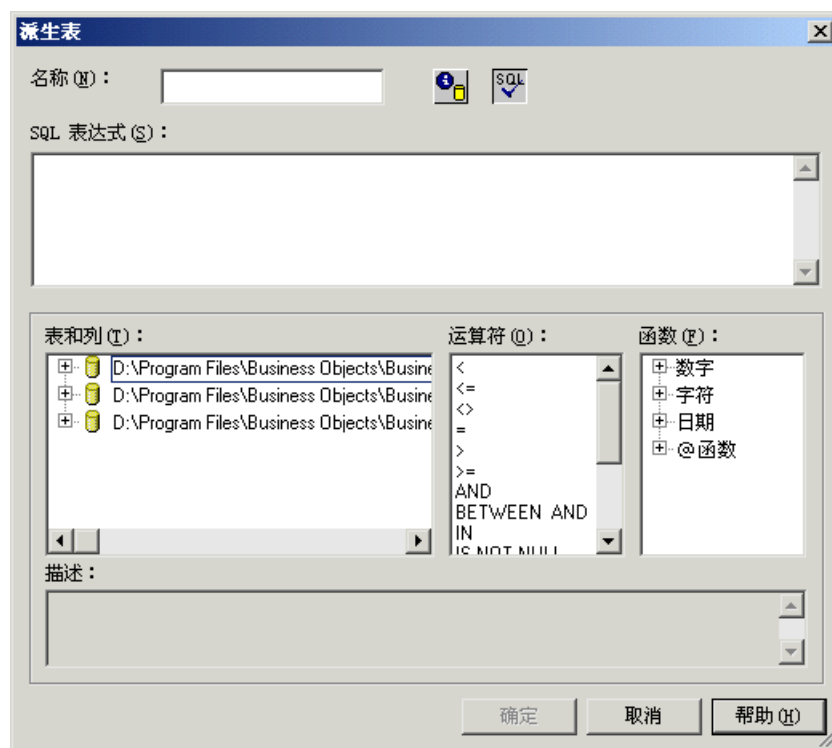
派生表在 Universe 设计工具模式的显示方式与在普通数据库表的显示方式完全一样，但创建这两种表的工作流程却不同。以下各节中描述了添加、编辑和删除派生表的过程。

4.4.1.1 添加派生表

添加派生表：

1. 单击“插入”菜单上的“派生表”。

将出现“派生表”对话框。



2. 在“表名称”框中键入表名称。
 3. 在“表名称”框下面的框中构建表 SQL。
可以直接键入 SQL 或使用“表和列”、“运算符”和“函数”框来构建它。
 4. 单击“确定”。
- 派生表将与物理数据库表一起出现在模式中。
5. 根据派生表的列构建对象（方式与您处置常规表的方式完全一样）。

4.4.1.2 编辑派生表

编辑派生表：

- 1 右键单击 Universe 设计工具模式中的表，然后从快捷菜单中选择“编辑派生表”。
- 2 编辑派生表，然后单击“确定”。

4.4.1.3 删除派生表

- 1 在模式窗格中，选择要删除的派生表。
- 2 按 Delete 键。

4.4.1.4 示例：创建派生表

示例：创建派生表以返回服务器信息

在本例中，您想创建使用户可以向其报表添加有关数据库服务器的信息的对象。您在某个运行于 SQL Server 数据库上的 Universe 中创建两个对象：“服务器名称”和“版本”，它们返回内置变量 @@SERVERNAME 和 @@VERSION 的值。

执行以下操作：

- 1 选择“插入”菜单上的“派生表”。
- 将出现“派生表”对话框。
- 2 在“表名称”框中键入“serverinfo”。
 - 3 在“SQL”框中键入 SQL Select @@SERVERNAME as servername, @@VERSION as version。

注意:

必须在 SQL 中为所有派生列提供别名。Universe 设计工具使用这些别名来命名派生表的列。

- 4 单击“确定”。

派生表 serverinfo 出现在 Universe 设计工具模式中。

- 5 创建一个称为“服务器信息”的类，然后根据 serverinfo 派生表的 servername 和 version 列以及该表的列，在此类下面添加两个维对象。请注意，serverinfo 表会像任何普通的数据库表一样出现在表的列表中，而它的列会像普通的表的列一样出现在列的列表中。

用户现在可以将服务器名称和版本对象放在报表上。

示例：显示每个国家/地区中的地区数

在本例中，您创建一个显示每个国家/地区中的地区数的表。SQL 如下所示：

```
select country,
count (r.region_id) as number_of_regions
from country c,
region r
where r.country_id = c.country_id
group by country
```

在本例中重要的是，要为包含计算的列提供别名。Universe 设计工具使用这些别名作为派生表中的列名。在本例中，表具有两个列：country 和 number_of_regions。

4.5 嵌套派生表

嵌套派生表（也称为“基于派生表的派生表”）是一个至少依据一个现有派生表推导出的表。嵌套派生表还可以引用数据库表。

请使用“派生表”编辑器输入 SQL 表达式，选择派生表（以及数据库中的任何物理表，如果需要）以创建嵌套派生表。在报表生成时，派生表的 SQL 表达式被插入到嵌套派生表的 SQL 表达式中。

4.5.1 使用派生表编辑器

使用派生表编辑器可以定义派生表或嵌套派生表。请在编辑器中输入 SQL 表达式，并双击对象（表、派生表、列、函数），以便为派生表或嵌套派生表创建 SQL 表达式。

使用 @DerivedTable 函数可在嵌套派生表中引用派生表。

- 函数 @DerivedTable(Derived_table_name) 包括在派生表编辑器的函数目录中。
- 派生表编辑器底部的中间窗格显示现有派生表和嵌套派生表。仅当 Universe 中存在派生表时，才会显示此窗格。

单击“检查完整性”将对派生表和嵌套派生表执行以下检查：

- 检测删除引用的派生表时对派生表的影响。
- 检查是否存在循环引用。
- 检查对象定义（SELECT 和 WHERE）中是否使用了 @DerivedTable()，这种用法不被允许。

4.5.2 创建嵌套派生表

创建嵌套派生表的方式与创建派生表相同。添加和重命名嵌套派生表的方式也与添加和重命名派生表相同。

创建嵌套派生表：

- 1 在示例目录中打开 Universe (*.unv)：Business Objects\BusinessObjects Enterprise 12\Samples\en\UniverseSamples。
- 2 右键单击“Universe 结构”窗格，从上下文菜单中选择“派生表”。
“派生表”编辑器打开，可用派生表列在“派生表”编辑器底部的中间窗格中。
- 3 键入嵌套派生表的名称。
- 4 键入 SQL 表达式。可以键入整个文本，也可使用编辑器助理键入。
- 5 双击对象（表、派生表、列、函数）。
- 6 使用 @DerivedTable 函数，语法如下：@DerivedTable(派生表名称) 以选择派生表。
- 7 单击“检查语法”，以检查派生表的语法并更正任何可能存在的错误，然后验证嵌套派生表。
嵌套派生表被添加到 Universe 中。
- 8 单击“确定”以验证嵌套派生表。

嵌套派生表出现在“结构”窗格中。派生表和嵌套派生表的颜色比那些实际数据库表的颜色浅。

注意：

要显示表值，请右键单击其它表。

现在在 Universe 中已创建嵌套派生表。

4.5.3 重命名嵌套的派生表

在重命名派生表时，新名称将通过引用它的所有其它派生表进行传播和更新。

4.6 使用包含输入列的表

在 Universe 中插入包含输入列的表时，会要求 Web Intelligence 或 Query as a Web Service 用户选择值或输入值以计算输入列。输入列与值绑定。输入列的数据不一定在原始数据库中可用，数据可以是：

- 在创建 Universe 时硬编码的值
- 由最终用户提供的值（提示之后）或从列表中选择
- 通过与另一个表的联接提供的值

仅当数据库连接是 BusinessObjects Data Federator 服务器时，才支持包含输入列的表。

在分析可用于解析输入列的联接时，请注意：

- 解析算法中仅考虑简单联接
- 不允许复杂联接，例如，包含多个左列或多个右列的联接
- 仅考虑包含“Equal”或“IN (INLIST)”操作数的联接。Between 等运算符不能用于解析输入列

在“结构”窗格中，包含输入列的表在输入列的一侧有一个箭头，而在“表浏览器”窗格中，输入列用特定图标标识。

当插入包含输入列的表时，请使用“输入列”编辑器输入设置。

注意：

在将表添加到 Universe 时，必须为该表的输入列分配默认值。

此功能可与下列产品和组件一起使用：

- Web Intelligence
- Query as a Web Service

相关主题

- 第 156 页上的[“定义硬编码的值列表”](#)
- 第 156 页上的[“定义供用户输入或选择的值列表”](#)

4.6.1 定义硬编码的值列表

数据库至少包含一个具有一个或多个输入列的表。

硬编码的值列表用作表的输入，以确定输入列的值。最终用户不输入任何值。要定义值列表，请执行以下步骤。

- 1 从数据库中选择表，将它添加到 Universe 设计工具的“结构”窗格中。
将出现“输入列”编辑器。
- 2 在“输入列”编辑器中，单击某个参数。
- 3 在“值”字段中，键入值或值列表。将每个值括在双引号中，并用分号(;)隔开。
值出现在“值”列中。
- 4 在“下一次执行”列表中，确保已选定“使用此值”。
“使用此值”出现在“下一次执行”列中。
- 5 单击“确定”。

该表将在 Universe 设计工具的“结构”窗格内显示。输入列用箭头标识。

4.6.2 定义供用户输入或选择的值列表

数据库至少包含一个具有一个或多个输入列的表。

用户可以输入值或从值列表中选择一个值，表将使用此值来确定输入列的值。要为模式中的输入列表格定义值，请执行以下步骤。

- 1 从数据库中选择表，将它添加到 Universe 设计工具的“结构”窗格中。
- 2 在“输入列”编辑器中，单击某个参数。
- 3 在“下一次执行”列表中，单击“提示我输入值”。
在执行 Web Intelligence 或 Query as a Web Service 查询时，将提示用户从关联的值列表中选择一个值。
- 4 在提示标签编辑字段中，编辑要对最终用户显示的默认提示。
- 5 单击“浏览 Universe 对象”，从 Universe 中选择值列表。
如果要从已添加到设置的值列表中删除对象，请在“选定对象”窗格中单击该对象，然后单击“清除”。

6 单击“确定”。

该表将在 Universe 设计工具的“结构”窗格内显示。输入列用箭头标识。在“表浏览器”中，输入列用特定图标标识。

4.7 定义联接

在模式中插入了多个表后，必须在相关的表之间创建联接。在模式中，联接和表一样重要，因为它们允许您通过一种有意义的方式组合多个表中的数据。

4.7.1 什么是联接？

联接是一种条件，链接各不相同但又相关的表中的数据。表通常具有父子关系。如果查询不包含联接，则数据库返回的结果集将包含查询表中的行的所有可能组合。这种结果集称为笛卡儿积，它并无多大用处。

例如，引用两个表（分别包含 100 和 50 行）的查询的笛卡儿积包含 5000 行。在包含许多表的大型数据库或查询中，笛卡儿积会迅速变得难以管理。在 Universe 设计工具中，联接表示为模式中链接表的线条。

4.7.2 为何要在模式中使用联接？

使用联接可确保返回多个表中的数据的查询不会返回不正确的结果。如果查询中包含两个表，则这两个表之间的联接定义了如何返回数据。

模式中的每个表均在对应用于用户需求的一列或多列中包含数据。在生产 Universe 中，Web Intelligence 用户可能需要运行这样的查询：这些查询组合许多个不同的对象（每个对象推导出一系列），并且返回任意表组合中的数据。

通过用联接链接模式中的所有表，可确保限制了在查询中组合不同表中的列数据的方式数。联接将表之间的列组合限制为匹配列或共有列。这防止了返回的结果数据包含来自毫无意义地匹配的列中的信息。

注意：

应始终在“结构”窗格中创建联接。不通过“结构”窗格创建的联接（例如在对象的 WHERE 子句中手动定义的联接）是在运行时创建的，因此 Universe 设计工具在进行完整性检查和上下文检测时将不考虑这些联接。在设计时需要这些过程的信息。本章后面将论述上下文和 Universe 完整性。

4.7.3 联接推导出什么 SQL？

默认情况下，Universe 设计工具通过引用表的匹配列或公用列，在 WHERE 子句中隐含地指定联接。

通常，所联接的每对表都会有一个 WHERE 子句。因此，如果组合四个表，则需要三个 WHERE 条件。

运行的查询（包含由联接链接的两个表）的结果是一个表，包含全都来自组合的表的列。此表中的每行都包含不同的输入表中的行数据（这些表的共有列具有匹配的值）。

4.7.3.1 ANSI 92 支持

如果目标 RDBMS 支持 ANSI 92，则可以将 Universe 参数（“文件” > “参数” > “参数”）ANSI92 设置为 Yes，以便为用户模式中创建的联接激活 ANSI 92 支持。当 Universe 支持将 ANSI 92 标准用于联接时，会在 FROM 子句中指定新创建的联接。也可以选择由将包含在 FROM 子句中的列推导出的对象。第 172 页上的 [“对 Universe 中联接的 ANSI 92 支持”](#) 一节中对 ANSI 92 支持进行了说明。

4.7.4 无需联接哪些表？

应对模式中的下列表建立联接：在依据 Universe 运行的 Web Intelligence 查询中，从由对象生成的 SQL 中推导出的所有表。上述这些表的唯一例外是以下类型的表：

- 模式中已针对每一用途为其创建了别名的基表。它们是您出于重命名或解决联接问题的原因而为其创建了别名的原始表。通常不会在任何对象定义中使用这些基表。
- 作为聚合感知语法的目标的表（虽然这需要依据具体情况而定）。例如，在示例 efashion Universe 中的聚合表（名称以“Agg_”开头）不会联接到模式中的任何表：

4.7.5 联接主键和外键

通常在一个表的主键和另一个表的外键之间创建联接。也可以在两个主键之间创建联接。联接至少有一侧不包含表的主键的这种情况很少见。

您需要了解数据库中的每个键是如何构建的。多个列键可能会影响您如何为联接设置基数，并且可能会影响您如何在模式中设置上下文。

第 209 页上的 [“检测 and 解决联接问题”](#) 中描述了检测和使用上下文的过程

4.7.5.1 显示键

可以在“结构”窗格中显示所有表中的主键和外键。在包含键的每个表中，键列在显示时带有下列线。当您选择显示键的选项时，必须刷新结构才能使键在显示时带有下列线。

将键列显示为带有下列线的能力取决于目标数据库中定义的主键。

注意：

显示带有下列线的键列时，信息存储在 .UNV 文件中。将 Universe 导出到中央管理服务器（CMS）资源库时，会丢失此信息。每次导入 Universe 时，必须重新显示它的键。

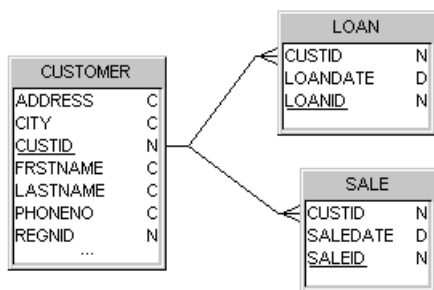
显示键：

- 1 选择“工具”>“选项”。
- “选项”对话框即会打开到“常规”页面。
- 2 单击“图形”选项卡。
- 将出现“图形”页面。
- 3 选择“列”组框中的“为键加下列线”复选框。
- 4 单击“确定”。

必须刷新结构才能使键列在显示时带有下列线。

- 5 选择“查看”>“刷新结构”。

将刷新数据库结构。模式中的键列将带有下列线，如下所示：



4.7.6 了解联接的基数

基数通过指明一个表中将有多少行与另一个表中的行匹配，进一步说明了两个表之间的联接。这对于检测联接问题和创建上下文以修正目标 RDBMS 结构的限制非常重要。

应为模式中的每个联接设置基数。Universe 设计工具可以自动检测和设置基数，但应始终手动检查基数，并考虑联接的键的性质。

第 190 页上的 [“使用基数”](#) 一节中对设置和使用基数进行了说明。

4.7.7 创建联接

可以通过下面几种方法使用 Universe 设计工具来创建联接：

- 在模式中手动描画联接。
- 直接定义联接属性。
- 选择自动检测的联接。
- 在插入表时自动创建联接。

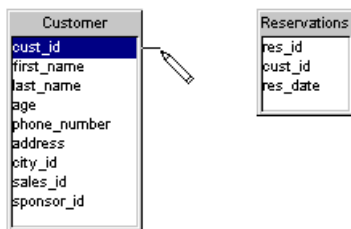
以下详细说明了上述每种方法。

4.7.7.1 在模式中手动描画联接

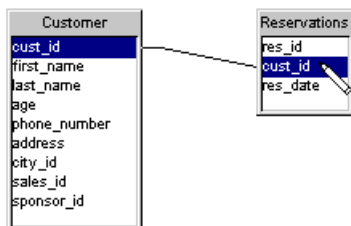
可以通过图形方式在表之间创建各个联接，方法是使用鼠标从一个表中的列向另一个表中的匹配列画一条线。

通过手动描画创建联接：

- 1 将指针放在您想作为联接的一端的列上。
指针将显示为手形符号。
- 2 单击并按住鼠标左键。
该列将突出显示。
- 3 将鼠标拖到另一个表中您想作为联接的另一端的列。
拖动鼠标时，指针会变为铅笔符号。



- 4 将铅笔符号放在目标列上。
目标列将突出显示。



- 5 松开鼠标按钮。
两个表之间的联接即被创建。
- 6 双击新的联接。

将出现“编辑联接”对话框。它列出了联接属性。第 165 页上的[“联接属性”](#)一节中说明了您可以为联接设置的属性（包括基数和连接类型）。

7 输入和选择联接的属性。

8 单击“确定”。

4.7.7.2 直接定义联接属性

通过在“编辑联接”对话框中直接定义连接属性来创建连接。

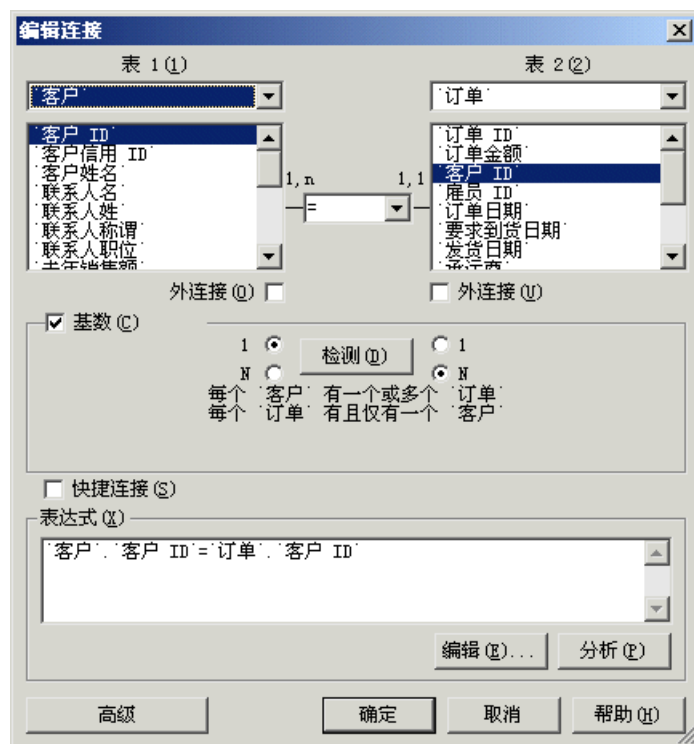
直接创建联接：

1 选择“插入”>“联接”。

或

单击“插入联接”按钮。

将出现“编辑联接”对话框。



- 2 从“表 1”下拉列表选择一个表。
选定表的列将出现在表名下方的列表框中。
- 3 单击您想作为新联接一端的列的名称。
- 4 从“表 2”下拉列表框中选择一个表。
选定表的列将出现在表名下方的列表框中。
- 5 单击您想作为新联接另一端的列的名称。
第 165 页上的 [“联接属性”](#) 一节中说明了您可以为联接设置的属性（包括连接运算符、基数和联接类型）。
- 6 输入和选择联接的属性。
- 7 单击“确定”。
新联接将出现在模式中，它链接了在“编辑联接”对话框中定义的两个表。

4.7.7.3 选择自动检测的联接

可以使用 Universe 设计工具的“检测联接”功能自动检测模式中的选定联接。此工具会识别跨越目标数据库中各个表的列名称，然后为模式中的表推荐候选联接。然后，您可以选择想创建哪个推荐的联接（或全部接受）。

如何自动检测联接？

根据“参数”对话框的“策略”页（“文件”>“参数”>“策略”选项卡）中显示的联接策略来检测联接。

策略是自动从数据库中抽取结构信息的脚本文件。Universe 设计工具附带了大量内置策略。这些策略列在“参数”对话框的“策略”页上的下拉列表框中。

默认的自动联接检测策略会根据匹配列的名称（键信息除外）检测联接。可以选择您在使用自动联接检测时想应用的联接策略。

注意：

有关使用策略的更多信息，请参阅第 94 页上的 [“选择策略”](#)。

适当使用自动联接检测

自动检测联接对于帮助您快速在模式中创建联接很有用。但是，在设计模式时，您需要知道自动联接检测的限制。

用于检测候选联接的联接策略匹配数据库中的列名称。在目标数据库中可能会存在这样的情况：主键、外键和其他联接列在不同的表中具有不一致的名称。Universe 设计工具将忽略这些列。应始终手动验证您同意创建的每个自动检测到的联接。您应意识到可能还有其他未检测到的必需联接。

使用自动检测创建联接：

- 1 验证在“参数”对话框的“联接”下拉列表框中选择了您想用于检测联接的联接策略。可以按以下所述验证此策略：
 - 选择“文件”>“参数”，然后单击“策略”选项卡。
 - 从“联接”下拉列表框中选择您想用于检测联接的策略，然后单击“确定”。
- 2 选择“结构”窗格中的多个表。

可以通过按住 Shift 并单击每个表来选择多个表，或者可以通过以下方式选择某个区域中的所有表：单击空白的空间，然后拖动鼠标以划定一个包含任意数量的表的矩形区域。

- 3 选择“工具”>“自动检测”>“检测联接”。

或

单击“检测联接”按钮。

将出现“候选联接”对话框。它列出了所选表的候选联接或推荐的联接。在“结构”窗格中，候选联接还显示为所选的表之间的蓝色线条。



- 4 单击“插入”以创建所有候选联接。

- 5 或

选择一个或多个联接，然后单击“插入”。

可以通过以下方式选择一个或多个联接：按住 Ctrl 并单击各个表，或者按住 Shift 并单击连续的一系列联接中的第一个和最后一个联接。

联接即被插入到模式中。

- 6 单击“关闭”。

4.7.7.4 自动插入相关表之间的联接

可以选择在将使用了联接的表插入到结构窗格的同时自动在模式中插入这些联接。自动联接创建由两个过程决定：

- 活动的联接策略决定用于检测联接的列信息。
- 必须选择默认创建选项“抽取表的联接”，才能允许自动在与联接相关的表之间创建联接。此选项位于“选项”对话框的“数据库”页上。

自动插入联接的限制

自动将相关表之间的联接插入到模式中是一种将联接加入到模式中的快速方式，但它可能会导致模式中出现严重的设计错误。联接是根据数据库结构插入的，因此将不会选择已在数据库中重命名的多个表共有的列。

不应使用此方式在用于生产的 Universe 中创建联接。相反，在演示时使用此方式，或将其用作构建 Universe 的快速方式（此时，您将在插入每个联接后仔细对它们进行验证）。

自动创建相关表的联接：

- 1 验证在“参数”对话框的“策略”页面中选择了要用于检测联接的联接策略。
- 2 选择“工具”>“选项”。
- 将出现“选项”对话框。
- 3 单击“数据库”选项卡。
- 此时将显示“数据库”页面。
- 4 选择“抽取表的联接”复选框。
- 5 单击“确定”。

现在，当您插入一个表而该表中的列引用了已插入到“结构”窗格中的表的其他列时，这些表之间的引用将会作为相应表之间的联接自动插入。

4.7.8 联接属性

在“编辑联接”对话框中定义联接属性。可以为联接定义以下属性：

属性	说明
表 1	联接左端的表。选定表的列在下拉列表框中列出。
表 2	联接右侧的表。选定表的列在下拉列表框中列出。
运算符	定义如何联接表的运算符。第 166 页上的 “联接运算符” 一节中说明了可用于联接的运算符。
外部联接	选择它时，它确定了在外部联接关系中哪个表包含不匹配的数据。第 183 页上的 “外部联接” 一节中详细说明了外部联接。
基数	选择它时，您可以定义联接的基数。第 190 页上的 “使用基数” 一节中说明了定义和使用基数。
快捷联接	将联接定义为快捷联接。第 187 页上的 “快捷联接” 一节中说明了快捷联接。
表达式	WHERE 子句，用于限制在查询中包含两个联接的表时返回的数据。
高级	<p>可在为 Universe 激活了 ANSI 92 支持的情况下使用。单击时将打开另一个联接属性框，其中列出了依据联接中的两个表的各个列构建的对象。可以选择要包括在 FROM 子句中的对象。</p> <p>有关为联接语法激活 ANSI 92 支持的信息，请参阅第 172 页上的 “对 Universe 中联接的 ANSI 92 支持” 一节。</p>

4.7.8.1 联接运算符

可以从“表 1”和“表 2”框之间的下拉列表框中为联接选择一个运算符。运算符使您可以定义联接用于匹配联接的列之间的数据的限制。

可以为联接选择以下运算符：

运算符	说明
=	等于
!=	不等于
>	大于
<	小于
>=	大于或等于
<=	小于或等于
Between	介于两者之间（Theta 联接）
Complex	复合关系

4.7.8.2 编辑和分析

“编辑联接”对话框还具有两个可用功能，可让您编辑和验证联接语法：

编辑

“编辑”按钮打开 SQL 编辑器。可以使用此图形编辑器修改联接中使用的表、列、运算符和函数的语法。有关使用此编辑器的更多信息，请参阅第 169 页上的 [“使用联接 SQL 编辑器”](#) 一节。

分析

“分析”按钮启动一个验证联接表达式的 SQL 语法的分析函数。如果分析成功，则您会收到结果为成功分析的消息。如果 Universe 设计工具遇到错误，则会收到一条错误消息，指出问题的根源。

4.7.9 编辑联接

可以使用以下任一方法编辑联接：

- 通过“编辑联接”对话框修改联接属性。
- 使用联接 SQL 编辑器直接修改联接 SQL 语法。
- 使用公式栏直接修改联接 SQL 语法。

本节中论述了上述每种方法。

4.7.9.1 使用“编辑联接”对话框

可以使用“编辑联接”对话框定义和编辑联接属性。也可以通过此对话框访问联接 SQL 编辑器，以直接编辑联接语法。第 165 页上的 [“联接属性”](#) 一节中对联接属性进行了说明。

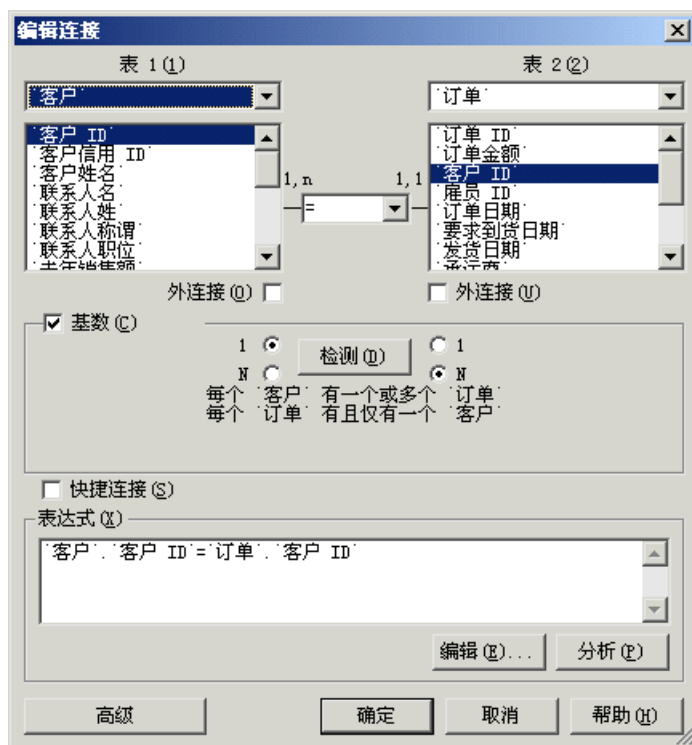
使用“编辑联接”对话框编辑联接：

1. 双击“结构”窗格中的某个联接。

或

单击某个联接，然后选择“编辑”>“联接”。

将出现“编辑联接”对话框。



- 2 从表之间的下拉列表框中选择一个运算符。
- 3 根据需要选择其他属性。
- 4 如果要使用 ANSI 92 语法定义联接，则单击“高级”按钮。
- 5 单击“确定”。

提示：

可以通过单击“编辑”按钮并使用联接 SQL 编辑器来直接编辑联接的 SQL。有关更多信息，请参阅第 169 页上的[“使用联接 SQL 编辑器”](#)。

4.7.9.2 使用联接 SQL 编辑器

可以使用图形编辑器来直接修改联接的 SQL 表达式。通过“编辑联接”对话框访问此编辑器。

使用联接 SQL 编辑器修改联接：

- 1 双击“结构”窗格中的某个联接。
- 或

单击某个联接，然后选择“编辑”>“联接”。

将出现“编辑联接”对话框。

- 2 单击“编辑”按钮。

将出现“联接 SQL 定义”框。联接的 SQL 表达式将出现在文本框中。



- 3 在编辑框的联接表达式中，单击您想添加或修改 SQL 语法的位置。

可以按以下所述使用编辑功能来修改或添加 SQL 语法：

您想要...	则执行以下操作...
更改联接任一端的列	<ul style="list-style-type: none"> 展开“表和列”框中的表节点。 双击列名称。
更改联接使用的运算符	双击“运算符”框中的一个运算符。
在联接中使用函数	<ul style="list-style-type: none"> 展开函数族节点。 双击一个函数。

列、运算符或函数将出现在联接定义中。

- 4 单击“确定”。

4.7.9.3 使用公式栏

公式栏是一个位于 Universe 窗口上方的文本框，它显示“结构”窗格中任何选定的联接或“Universe”窗格中选定对象的公式或表达式。可以使用三个位于公式栏左边的编辑按钮：

编辑按钮	说明
	取消尚未验证的上一次修改。如果对联接表达式做了几处更改但未验证这些更改，则单击“取消”按钮会将表达式恢复到其原始状态。如果想撤消任何单独的修改，应使用“编辑” > “撤消”选项，或者单击“撤消”。
	验证表达式。这会将任何更改应用到联接表达式。如果想撤消验证后所做的修改，请使用“编辑” > “撤消”选项，或者单击“撤消”。
	为选定的联接打开“编辑连接”对话框。

显示公式栏：

- 选择“查看” > “公式栏”。

公式栏将出现在 Universe 窗口的上方。

使用公式栏修改联接：

1. 单击想编辑的联接。
该联接的公式将出现在公式栏中。
 2. 在公式栏的联接表达式中单击要修改语法的位置。
 3. 根据需要修改表达式。
 4. 单击“验证”使更改生效。
 5. 按“返回”键以退出公式栏。
- 或
- 单击公式栏外部的任何位置。

4.7.10 对 Universe 中联接的 ANSI 92 支持

Universe 设计工具支持对联接使用 ANSI 92 语法。默认情况下不支持 ANSI 92。必须通过将 SQL Universe 参数 ANSI92 设置为 YES 来激活此支持。此参数列在 Universe 参数对话框的“参数”页面（“文件”>“参数”>“参数”）上。激活之后，您就可以选择为 Universe 中的联接使用 ANSI 92 语法。

注意：

ANSI 92 设置也在 .prm 文件中声明。如果 .prm 设置为“usual”，则 Universe 设计工具的设置优先。如果 .prm 设置为“ANSI92”，将覆盖 Universe 设计工具级设置。有关 .prm 文件和 ANSI 92 设置的更多详细信息，请参阅《数据访问指南》。具体行为取决于用户的数据库版本。有关更多信息，请参阅数据库技术详细信息。

在联接中使用该语法之前，确保您验证了目标 RDBMS 是否支持 ANSI 92。

下面描述了在 Universe 中激活 ANSI 92 支持以及使用 ANSI 92 语法定义联接的过程。

4.7.10.1 示例：比较缺省联接语法和 ANSI 92 语法

以下显示了用于两个联接的联接语法。第一种语法显示了在 WHERE 子句中定义联接这一缺省行为，而第二种语法显示了 FROM 子句（使用 ANSI 92 标准）中的同一联接。

默认联接语法

```
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')
```

使用 ANSI 92 标准的相同联接

```

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')

```

4.7.10.2 在 Universe 中激活 ANSI 92 支持

为联接激活 ANSI 92 支持:

- 1 选择“文件”>“参数”。

将出现“Universe 参数”对话框。

- 2 单击“参数”选项卡。

将出现“参数”页面。它列出了某些 SQL 生成参数，您可以在 Universe 级别设置这些参数，以便为当前 Universe 优化 SQL 生成。在以前版本的 Business Objects 产品中，这些参数包含在目标 RDBMS 的 PRM 文件内。某些特定于 RDBMS 的参数仍然包含在 PRM 文件中，但许多标准的 SQL 参数现在列于“参数”页面中。有关可用参数的完整列表，请参阅第 103 页上的[“设置 SQL 生成参数”](#)一章。

- 3 单击列表中的 ANSI92 参数。
- 4 在值框中键入“YES”。
- 5 单击“替换”。
- 6 单击“确定”。

现在即可将 ANSI 92 标准应用于当前 Universe 的联接定义。单击“编辑联接”对话框上的“高级”按钮时，将出现“高级联接”框。可以定义一个过滤器，以确定想在联接的 FROM 子句中包含哪个维。

4.7.10.3 使用 ANSI 92 语法定义联接

可以使用 ANSI 92 语法通过“编辑联接”属性对话框定义联接。通过使用高级编辑框（允许您选择要包括在联接定义中的对象），您可以达到此目的。

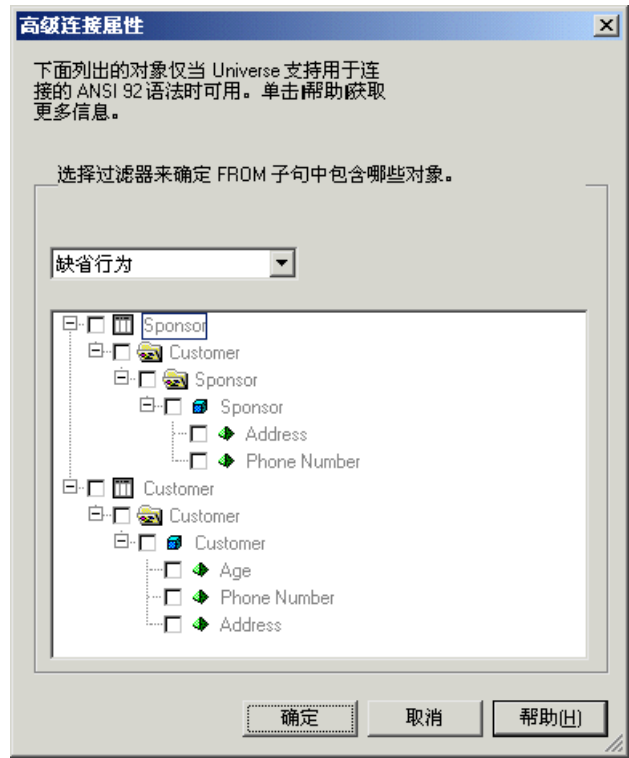
使用 ANSI 92 语法定义联接：

- 1 为 Universe 激活 ANSI 92 支持。有关信息，请参阅第 173 页上的 [“在 Universe 中激活 ANSI 92 支持”](#) 一节。
- 2 双击模式中的某个联接。

将出现该联接的“编辑联接”框。

- 3 单击“高级”按钮。

将出现“高级联接属性”对话框。



- 4 从下拉列表中选择以下 FROM 子句过滤器之一。

FROM 选项	说明
默认行为	应用联接的默认语法。联接是在 WHERE 子句中定义的。

FROM 选项	说明
FROM 中的所有对象	在联接的右边和左边的表中的列上定义的所有对象将包括在 FROM 子句中。
FROM 中没有对象	FROM 子句中不包括对象。
FROM 中的选定对象	只会在 FROM 子句中包括从联接表的“高级联接属性”树视图中选择的对象。

- 5 如果在 FROM 过滤器中选择了选定对象，则选择要包括在 FROM 子句中的对象。
- 6 单击“确定”。
- 7 在“编辑联接”框中输入任何其他联接参数。
- 8 单击“确定”。

4.7.11 删除联接

删除联接：

- 1 单击联接。
该联接即被选定。
- 2 执行以下任一操作：
 - 按键盘上的 backspace 键
 - 按键盘上的 Delete 按钮
 - 右键单击该联接，然后从上下文菜单中选择“清除”。将出现一个确认框，请您确认删除联接的操作。
- 3 单击“是”。
联接即被删除。

注意：

删除联接时，确保您清楚这会在模式和 Universe 中造成的所有后果。验证删除联接不会影响上下文。如果试图删除某个联接，但该联接已用在一个或多个上下文中，Universe 设计工具会发出警告。

您需要手动验证会影响哪个上下文，而如果删除联接会影响该上下文，还需了解这对 Universe 造成的影响。

4.8 定义特定的联接类型

可以在 Universe 设计工具中定义以下联接类型：

联接类型	说明
相等联接 (包括复合的相等联接)	<p>根据一个表的列中的值和另一个表的列中的值之间的相等性链接表。因为同一列均出现在两个表中，所以联接使两个表同步。</p> <p>也可以创建复合相等联接，在这种情况下一个联接链接两个表之间的多个列。</p>
Theta 联接（条件联接）	根据两个列之间的关系（相等性除外）链接表。
外部联接	链接两个表，其中一个表包含与另一个表的共有列中的行不匹配的行。
快捷联接	在两个表之间提供替代路径的联接，此连接忽略中间表，并且不管是什么方向结果都一样。通过尽可能切短较长的联接路径来优化查询时间。
自我限制的连接	用于在表上设置限制的单一表联接。

每种联接类型均在本章中各自对应的节中详细说明。创建每种联接类型时使用的方法都相同；但是，在创建联接时，必须在“编辑连接”框中为每种连接定义不同的属性。

4.8.1 创建相等联接

相等联接链接两个表（表 1 中的某列与表 2 中的某列具有相等的值）。限制遵守以下语法：

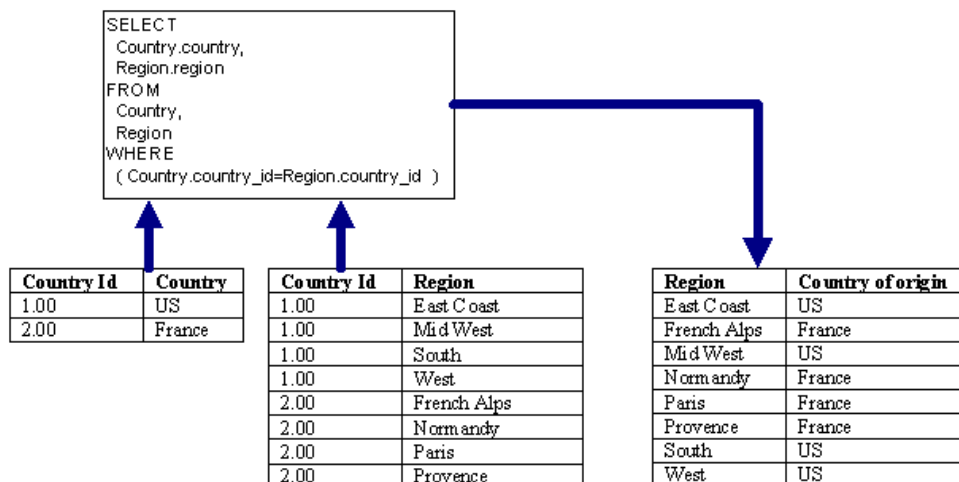
Table1.column_a = Table2.column_a

在标准化的数据库中，相等联接中使用的列通常是一个表中的主键和另一个表中的外键。有关键的信息，请参阅第 159 页上的 [“联接主键和外键”](#) 一节。

创建新联接时，默认情况下它是相等连接。模式中的大多数连接应为相等联接。

示例：相等联接限制数据

在下例中运行 Select 语句时，Select 和 From 子句创建笛卡儿积。但是，在返回任何数据之前，Where 子句应用某个限制，使得只返回 Country ID 列在两个表中互相匹配的行。



4.8.1.1 创建新的相等联接

创建新的相等联接：

- 在两个表之间创建联接。

缺省的新联接是相等联接。

提示：

第 160 页上的 [“创建联接”](#) 一节中说明了可用于创建联接的不同方法。

4.8.1.2 通过现有联接创建相等联接

通过现有联接创建相等联接：

1. 双击某个现有联接。
将出现“编辑联接”框。
2. 在“表 1”列表框中选择一列。
3. 在“表 2”列表框中选择匹配列。
4. 从“运算符”下拉列表框中选择“=”。

以下的“编辑联接”框显示了表 Customer 和 Reservations 之间的相等联接。

The screenshot shows the "编辑联接" (Edit Join) dialog box. It is configured for a 1:n relationship between two tables:

- 表 1 (1):** 客户 (Customer). The selected field is 客户 ID.
- 表 2 (2):** 订单 (Orders). The selected field is 客户 ID.
- 运算符 (Operator):** =
- 基数 (Cardinality):** 1, n (One-to-many).
- 外连接 (Outer Join):** 未选中 (Not selected).
- 快速连接 (Quick Join):** 未选中 (Not selected).
- 表达式 (Expression):** 客户. 客户 ID = 订单. 客户 ID

Buttons at the bottom include: 高级 (Advanced), 确定 (OK), 取消 (Cancel), 帮助 (Help), 编辑 (Edit), and 分析 (Analyze).

注意：

共有的列并不总是具有相同的名称。您需要验证数据库中的主键和外键列名称。不同的表可以使用相同的键列，但要根据表在数据库中的作用为每个表重命名键列。

- 5 单击“分析”按钮以检查联接语法。

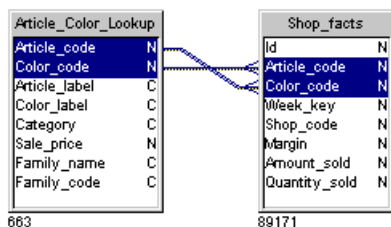
如果收到错误消息，请检查以确定该列是否由两个表共用。

- 6 单击“确定”。

4.8.1.3 创建复合相等联接

也可以创建复合相等联接。这是链接两个表之间的多个列的单一联接。可以通过在联接的“编辑属性”表单中为联接使用“Complex”运算符来创建复合相等联接。

eFashion 这一示例 Universe 包含了一个如下所示的复合联接。



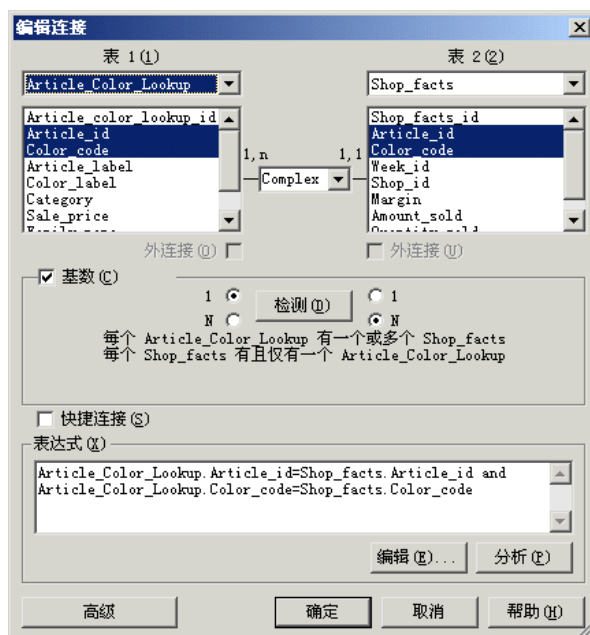
在联接的列之间使用复合相等联接而不是多个单一相等联接具有以下优点：

- 只检测一个基数。这可以在检测基数时节省时间，同时还使模式保持清晰和易于阅读。
- 可以在联接的“编辑属性”框的“表达式”文本框中，查看两个表之间的所有联接的 SQL。在两个表之间使用多个单一相等联接时，每个联接都有一个表达式。

创建复合相等联接：

- 1 双击某个现有联接。
将出现“编辑联接”框。
- 2 在“表 1”列表框中选择多个列。
- 3 在“表 2”列表框中选择匹配列。
- 4 从“运算符”下拉列表框中选择“Complex”。

以下的“编辑联接”框显示了表 Article_Color_Lookup 和 Shop_facts 之间的复合（complex）相等联接。



- 5 单击“分析”按钮以检查联接语法。

如果收到错误消息，请检查以确定该列是否由两个表共用。

- 6 单击“确定”。

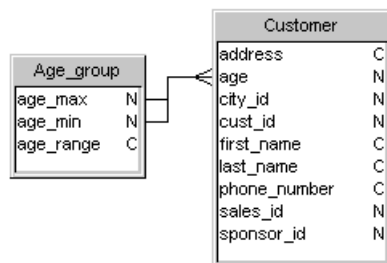
4.8.2 Theta 联接

Theta 联接是一种根据两列之间的关系（相等性除外）链接表的联接。Theta 联接可以使用除“等于”运算符之外的任何运算符。

以下示例和过程显示了如何创建使用“介于”运算符的 Theta 连接。

示例：Theta 联接

以下的 Age_Group 表包含了年龄范围信息，可用于分析有关客户年龄的数据。

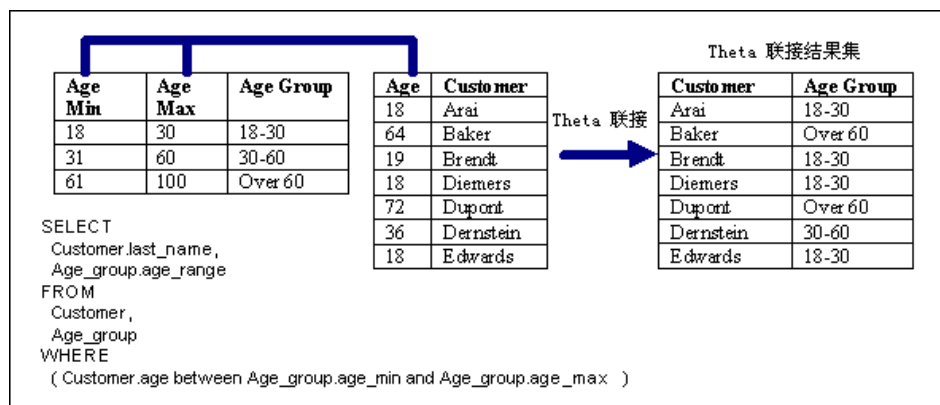


您需要在 Universe 中包含此表，但 Customer 表和 Age_Group 表之间没有共有列，因此无法使用相等联接。

您为最大年龄范围和最小年龄范围创建一个使用运算符“介于”的 Theta 联接。通过使用 Theta 联接，您推导出存在着一个联接，在此联接中，Customer 表中 Age 列的行中的值介于 Age_Group 表中 Age_Min 列和 Age_Max 列的行中的值之间。以下表达式定义了此联接：

Customer.age between Age_group.age_min and Age_group.age_max

下图显示了 Age max、Age min 和 Age 之间的联接以及结果集（在运行于 Age_Group 和 Customer 表上的查询中使用此 Theta 联接时返回此结果集）。



4.8.2.1 创建 Theta 联接

使用范围列创建 Theta 联接：

1. 在两个表之间创建联接。
默认情况下将创建相等联接。

- 2 双击该联接。

将出现“编辑联接”对话框。

- 3 单击“表 1”列的列表框中的一列。
- 4 按住 Ctrl 键并单击“表 2”列的列表框中的两列。

下例显示了选定的两列：age_min 和 age_max。“Between”运算符自动出现在运算符下拉列表中。

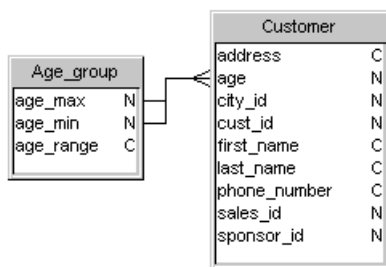


- 5 单击“分析”按钮以测试联接的有效性。

如果您收到错误消息，请进行检查，查看是否正确地选择了列。

- 6 单击“确定”。

将在“结构”窗格中创建联接。



4.8.3 外部联接

外部联接是一种链接两个表的联接，其中一个表包含与另一个表的共有列中的行不匹配的行。

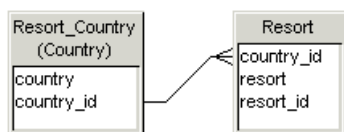
通过指定原始相等联接中的哪个表是外部表来定义外部联接。外部表包含您想返回所有值的列（即使这些值不匹配）。从“编辑联接”对话框中为选定的联接指定外部表。

4.8.3.1 完全外部联接

默认情况下，取决于指定了在哪一边指定了外部表，您可以创建左外部联接或右外部联接。通过为 Universe 中的连接激活 ANSI 92，您还可以创建完全外部联接。通过将 Universe SQL 参数 ANSI 92 设置为 YES（“文件”>“参数”>“参数”），可以达到此目的。这将允许 Universe 支持对联接使用 ANSI 92 语法，并且您可以选择联接任一侧的表作为外部表。有关创建完全外部联接的信息，请参阅第 185 页上的[“定义完全外部联接”](#)一节。

示例：外部联接

以下的 Resort_Country 表和 Resort 表由相等联接链接。



每个休假地都属于某个国家/地区，但每个国家/地区可能并不具有休假地。如果使用相等联接，则查询的结果集只会显示有关具有休假地的国家/地区的信息；Australia（澳大利亚）、France（法国）和 US（美国）。

Country	Resort
Australia	Australian Reef
France	French Riviera
US	Bahamas Beach
US	Hawaiian Club
US	Royal Caribbean

但是，您可能想显示所有国家/地区，而不考虑 Resort 表的外键中的等值。要做到这一点，请定义一个外部联接，以便返回所有国家/地区（即使 Resort 列中没有匹配值）。

外部联接的语法（Microsoft Access）如下所示：

```
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 }
```

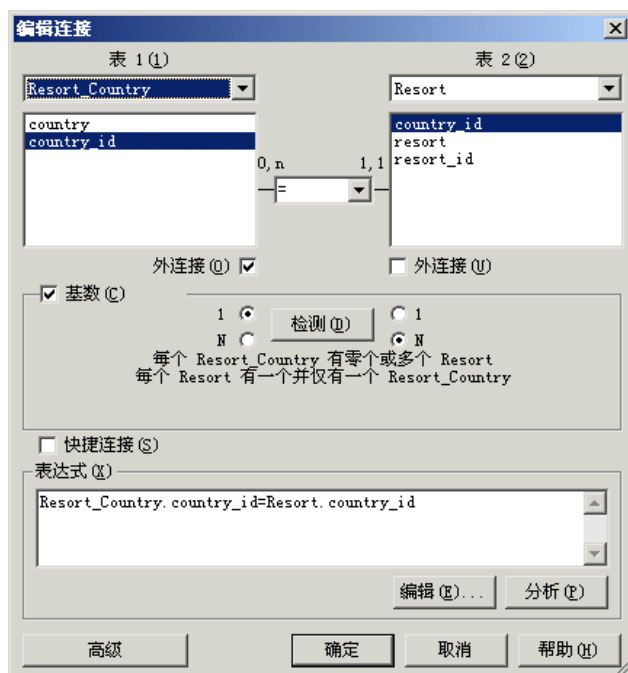
注意：

上例使用了 Microsoft Access，因此，尾随 Resort 表的任何一对多联接也必须使用外部联接。否则，如果尾随的联接未返回匹配的 NULL，则将不会考虑原始外部联接返回的 NULL。外部联接的处理是特定于 RDBMS 的，因此，请参阅您的 RDBMS 文档以了解有关信息。有关限制使用外部联接的更多信息，另请参阅第 186 页上的 [“针对外部联接使用的限制”](#) 一节。

4.8.3.2 创建外部联接

创建外部联接：

- 1 双击某个现有相等联接。
将出现“编辑联接”对话框。
- 2 为在查询中返回所有值的表选择“外部联接”复选框。
在下例中，您想返回 Resort_Country 的所有值。



- 3 单击“分析”按钮以验证联接语法。

如果您收到错误消息，请进行检查，查看是否正确地选择了列。

- 4 单击“确定”。

Universe 设计工具将在“结构”窗格中显示该联接。外部联接由一个小圆圈指示，它位于指向返回不匹配值的表的联接上（在另一侧上）。



4.8.3.3 定义完全外部联接

可以使用用于定义外部联接的 ANSI 92 标准来定义外部联接。这样，您将能够指定完全外部联接。要为外部联接使用 ANSI 92 标准，您必须将 ANSI 92 参数设置为 YES。此参数可在“参数”页面（“文件”>“参数”>“参数”）上找到。

注意：

有关为 Universe 设置此参数和其他 SQL 生成参数的信息，请参阅第 103 页上的 [“设置 SQL 生成参数”](#) 一节。

将 ANSI 92 参数设置为 YES 后，您就可以选择联接两侧的表作为外部表。在设置此参数之前，必须确保目标 RDBMS 支持为外部联接使用 ANSI 92 语法。

定义完全外部联接的过程分为两个阶段：

- 为 Universe 的外部联接激活 ANSI 92 支持：有关信息，请参阅第 173 页上的 [“在 Universe 中激活 ANSI 92 支持”](#) 一节。
- 使用“编辑联接”对话框来定义完全外部联接。

定义完全外部联接：

- 1 为 Universe 激活 ANSI 92 支持。
- 2 双击模式中的某个联接。

将出现“编辑联接”对话框。

- 3 为联接中包括的两个表选中“外部联接”复选框。
- 4 单击“确定”。

Universe 设计工具将在“结构”窗格中显示该联接。完全外部联接是由两个表之间联接链接上的两个圆圈指明的。

4.8.3.4 针对外部联接使用的限制

使用外部联接可能会很有用，但您应知道以下的性能和实施问题：

- 可能会降低性能。包含外部联接时，会返回更多的行并且某些数据库将不会使用索引，因此，大量的数据可能会降低查询性能。
- 有关外部联接使用的数据库限制。并非所有数据库都允许在 WHERE 子句中控制外部联接。使用自我限制的联接时这一控制是必需的。例如，自我限制的连接“TYPE_CODE=10”会返回 TYPE=10 或 Type 为 NULL 的所有行，这是因为当类型代码为 NULL 时 TYPE=10 将永远不会为真（而 NULL 值是由外部连接生成的）。
- 应验证目标 RDBMS 如何处理外部联接，以避免尾随原始外部联接的查询路径不完整。例如，在 Club.mdb 这一 Microsoft Access 示例数据库中，也必须将联接路径中所有尾随原始外部联接的一对多联接定义为外部联接。否则，最终的查询将会忽略原始外部联接。在下例中，Resort 和 Service_Line 之间的联接会忽略 Resort_Country 和 Resort 之间的外部联接返回的 NULL 值。运行包含这三个表的查询时，会返回数据库错误，该错误建议用户创建执行第一个联接的独立查询，然后将该查询包括在 SQL 语句中。这种类型的错误会使许多用户觉得混乱，因此，在这种情况下，较好的做法是不使用外部联接，或者使路径中全都是外部联接。



4.8.4 快捷联接

快捷联接是一种在两个表之间提供替换路径的联接。快捷联接通过不考虑中间表提升了查询性能，并因而缩短了通常较长的联接路径。

快捷联接常用于将共享的查找表链接到联接路径中更靠前的另一个表。联接路径由同一上下文中的几个不同的表组成。

在这种情况下，只有符合以下所述快捷联接才有效：已将所查找的值反标准化到表的层次中的较低级别，因而使相同的值存在于所联接的所有级别中。

如果不能“简化”指定上下文的任意联接路径，则快捷联接将被忽略。针对相关 Web Intelligence 查询生成的 SQL 不会考虑无效的快捷联接。

注意：

Universe 设计工具在自动环路和上下文检测的过程中不会考虑快捷联接。但是，如果您为快捷联接设置基数，则可避免在检测上下文时收到“未设置所有基数”消息。

4.8.4.1 创建快捷联接

创建快捷联接：

- 1 确定联接路径中可直接链接的两个表。
- 2 在这两个表之间创建联接。
- 3 双击新的联接。
将出现“编辑联接”对话框。
- 4 选择“快捷联接”复选框。
- 5 根据需要选择或键入其他联接属性。
- 6 单击“确定”。

将出现联接这两个表的快捷联接。在“结构”窗格中，以点线显示快捷联接。

注意：

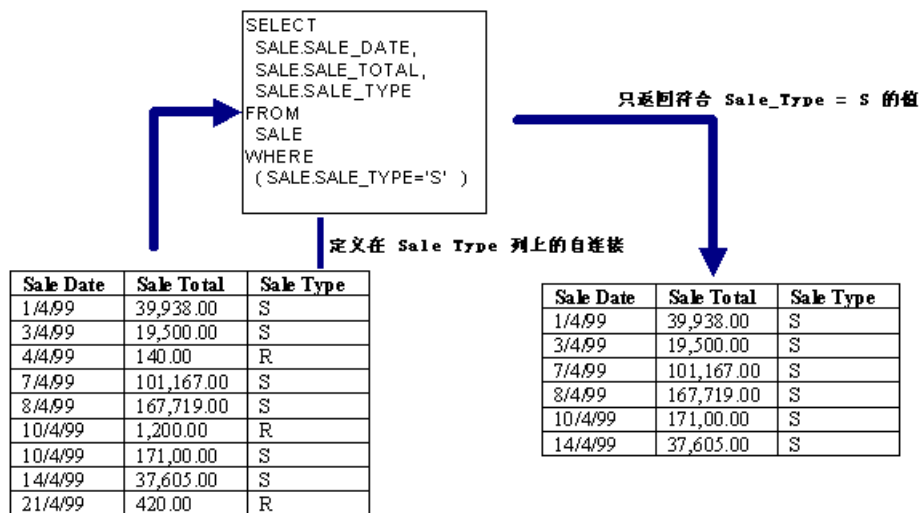
应将快捷联接的基数设置为与它替换的联接路径的基数一样。

4.8.5 自我限制的连接

自我限制的联接其实根本不是联接，而是在单个表上的自我限制。可以使用自我限制的联接通过固定值限制表值返回的结果。

示例：自我限制的联接

以下所示的 Sales 表包含了销售和租借的汽车的数据行。Sale_Type 列用作指明交易类型的标志（S = 汽车销售，R = 汽车租赁）。自我限制的联接将 Sales 返回的数据限制为 Sale_Type = S。这确保了任何基于 Sales 表的对象或经过该表的联接只产生覆盖汽车销售的查询结果。



没有自我限制的联接，查询的结果集会产生 Sale_Type 列等于“S”或“R”的行。

提示：

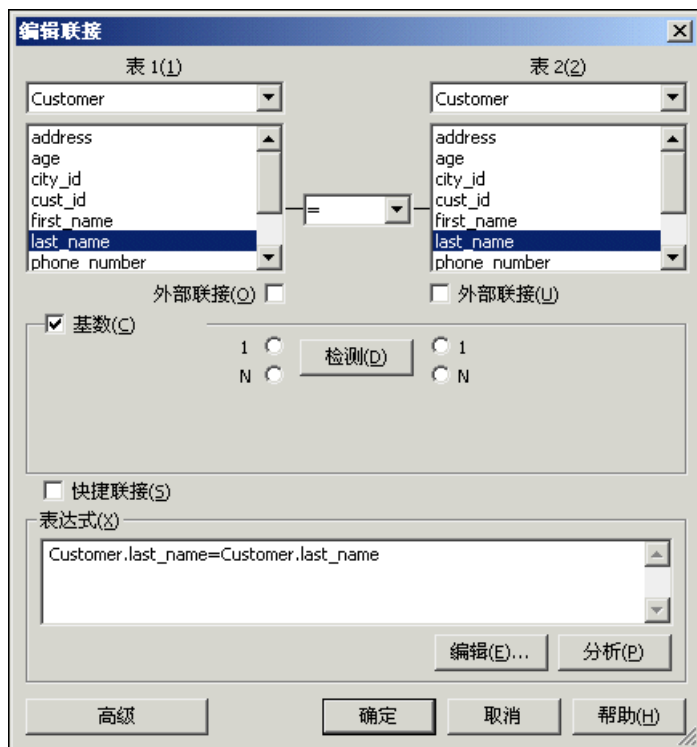
为自我限制的联接设置基数有助于防止在检测上下文时收到“未设置所有基数”消息。只要是设置基数，就应将每个基数都设置为一致（尽管实际的设置并不重要）。

4.8.5.1 创建自我限制的联接

创建自我限制的联接：

- 1 选择“插入”>“联接”。
将出现“编辑联接”对话框。
- 2 从“表 1”下拉列表框中选择想针对其设置自我限制联接的表。
选定表的列将出现在表的列的列表中。
- 3 在列下拉列表框中单击想用于定义限制的列。
- 4 从“表 1”下拉列表框中选择您已选择的同一个表。
- 5 在“表 1”列的列表框中单击您所选的同列。

联接的表达式将出现在“表达式”文本框中。



- 6 用您想在联接列上设置的限制值替换联接表达式中的操作数值。

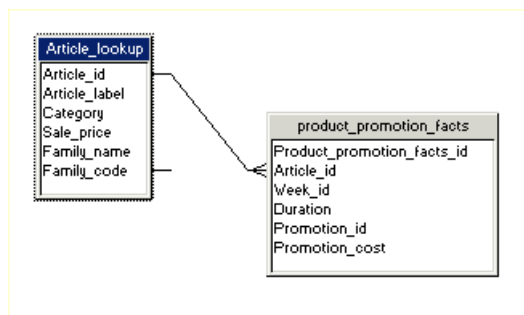
例如，如果要将 Family_code 列中的返回值限制为“F3”，则可以将 = 号后面的“Article_lookup.Family_code”替换为“F3”，如下所示：



7 单击“分析”按钮以验证语法。

8 单击“确定”。

自我限制的联接将显示为一条短线，并显示在已在其上定义了自我限制联接的列的旁边。



4.9 使用基数

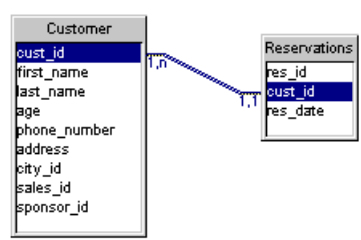
基数是联接的属性，它描述了一个表中有多少行与另一个表中的行匹配。

基数表示为位于联接一端的列中的最小行数和最大行数，这些行与联接另一端的列中的行匹配。

最小匹配行数和最大匹配行数可以等于 0、1 或 N。联接表示双向关系，因此，它必须始终具有两个基数，一个基数代表联接的一端。

示例：联接的基数

Customer 和 Reservations 这两个表由联接链接。



上述联接中的基数可以按以下所述表示：

说明	表示法
对于每个客户，可以有一个或多个预订	(1, N)
对于每个预订，只能有一个客户	(1, 1)

4.9.1 基数在 Universe 设计工具中的使用方法

运行查询时，联接的基数在生成的 SQL 中不起作用。但是，Universe 设计工具使用基数来确定上下文和有效的查询路径。

上下文是提供有效查询路径的联接的集合。使用上下文可解决可能返回太多或太少行的联接问题，这些问题是由在目标数据库中链接表的方式造成的。第 209 页上的 [“检测 and 解决联接问题”](#) 中描述了上下文。

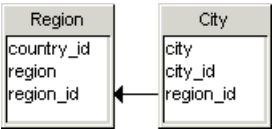
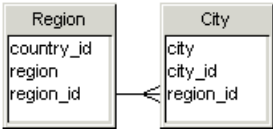
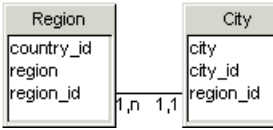
上下文会影响为查询生成的 SQL，原因是它们会指引最终用户采用特定的联接路径，或解决联接路径问题：

需要验证是否为模式中的所有联接正确设置了基数，以确保具有正确的上下文以及有效的联接路径。

设置基数也可以帮助您了解数据库中的表是如何相关的，以及以图形方式确定模式中的潜在联接路径问题。

4.9.1.1 显示基数

可以使用以下符号在“结构”窗格中显示基数：

基数符号	示例	说明
箭头		箭头指明联接的“一”端。如果基数为 1, 1，则会在联接的每一端显示一个箭头。
数量		鸟足状符号指示联接的“许多”末端。如果基数为 1, 1，则会显示一条直线。
1, N		基数在联接的每端显示为比率。

显示基数：

- 1. 选择“工具”>“选项”。

“选项”对话框即会打开到“常规”页面。

2 单击“图形”选项卡。

将出现“图形”页面。

3 单击“箭头”、“元数”或“1, n”单选按钮。

4 单击“确定”。

4.9.1.2 可以为联接设置什么基数？

可以为联接设置以下基数：

基数	说明
一对一 (1, 1)	对于表 1 中的每行，期望表 2 中有匹配的一行（只一行）
一对多 (1, N)	对于表 1 中的每行，期望表 2 中有匹配的一行或多行
多对一 (N, 1)	与一对多 (1, N) 相同，但行匹配的方向相反。
多对多 (N, N)	<p>对于表 1 中的每行或多行，期望表 2 中有匹配的一行或多行。</p> <p>多对多基数在关系数据库中很少见，而且它们将返回重复的行，这会造成性能下降和结果可能不准确。如果具有 (N, N) 基数，则应重复检查所涉及的联接，并确保您了解表之间的关系。</p>

可以手动设置基数，或使用 Universe 设计工具中的自动基数检测工具。以下各节中说明了这两种方法。

4.9.2 手动设置基数

可以通过在联接的“编辑联接”框中为联接定义基数来手动设置联接的基数。

为何手动设置基数？

手动设置基数时，必须考虑每个联接。这将帮助您发觉模式中的潜在联接路径问题。如果只选择自动检测基数，您可能无法发觉这些问题；例如，联接路径末端的孤立一对一连接，或者在并不需要所有列以确保唯一性的情况下有过多的主键。

了解键

通过评估每个表中的主键和外键来为大多数联接情况确定基数。以下是对主键和外键的说明：

键	说明
主	表中的单列或列的组合（其值标识表中的每行）。主键保证了行在表中的唯一性。每个表只有一个主键。
外	<p>单列或列的组合，需要其值才能匹配另一个表中的主键或另一个唯一的键。</p> <p>外键实施约束，如“如果尚未创建客户，则您不能为其创建销售”。每个表可以有多个外键。</p>

设置基数的标准是什么？

通过评估主键和外键之间的关系来确定联接的基数，如下所示：

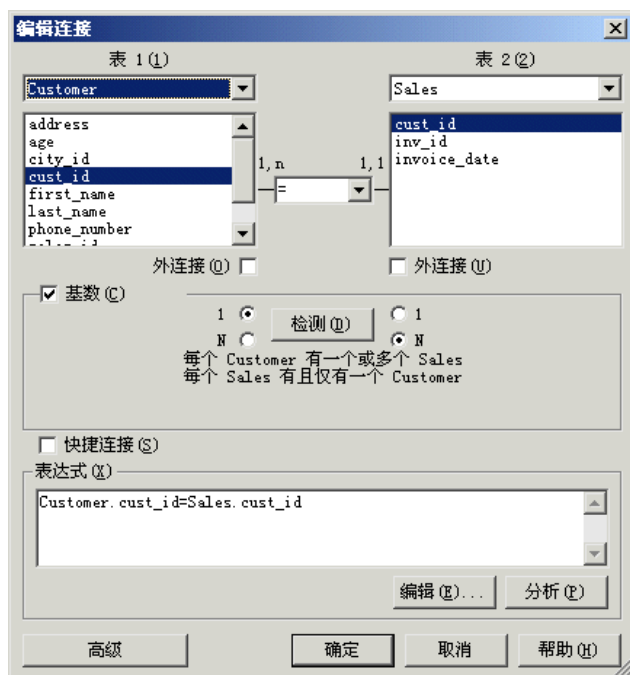
如果联接...	基数可能是...
将表 1 的完整主键与表 2 的完整主键相联接。例如：	<p>一对一 (1, 1)。</p> <p>将只为每个主键值返回每个表中的一行。</p>

如果联接...	基数可能是...
将表 1 的完整主键与表 2 的对应外键相联接。例如：	一对多 (1, N)。 不能保证表的外键值唯一，因此，可能会为原始表上的主键的单一值返回许多个匹配值。

如果联接...	基数可能是...
将表 1 的完整主键与表 2 的部分主键相联接。例如：	一对多 (1, N)。不完整的主键匹配可能会为原始表上的主键的单一值返回许多个匹配值。

4.9.2.1 手动设置基数：

- 1 双击某个联接。
或
单击某个联接，然后选择“编辑” > “属性”。
将出现“编辑联接”对话框。
- 2 选择“基数”复选框。
- 3 为表 1 选择“1”或“N”单选按钮。
- 4 为表 2 选择“1”或“N”单选按钮。



5 单击“确定”。

4.9.2.2 自动检测基数

可以使用 Universe 设计工具的“检测基数”功能在以下情形中自动检测基数：

- 选定的联接
- 所有联接
- 在创建联接时
- 通过“编辑联接”框

使用自动基数检测时，会在检测时自动实施基数。

注意：

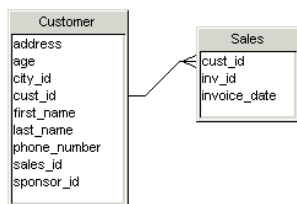
应适当地使用自动基数检测。对于快速检测模式中的所有基数而言，这可能很有用；但是，许多关系数据库中存在着不少固有的结构问题，这些问题可能会导致不正确的基数检测。其中包括不完整的主联接和过量设计的主键。第 199 页上的[“使用基数解决数据库限制”](#)一节中讨论了这些问题。

自动为选定的联接检测基数

自动为选定的联接检测基数：

- 单击一个联接并选择“工具”>“检测基数”。
- 右键单击该联接，然后从上下文菜单中选择“检测基数”。

将显示基数，并且“多”端为鸟足状。



如果您不选择联接而直接选择“工具”>“检测基数”，则会收到指出未选择联接的消息，而且该消息将询问您是否想检测所有联接的基数。

自动为所有联接检测基数

自动为所有联接检测基数：

- 1 选择“工具”>“自动检测”>“检测基数”。

或



单击“检测基数”按钮。

将出现一个消息框，询问您是否想检测所有联接的基数。

- 2 单击“是”。

将显示“结构”窗格中的所有联接（带有基数）。

在创建联接时自动检测基数

在创建联接时自动检测基数：

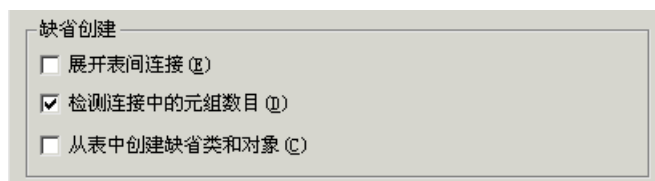
- 1 选择“工具”>“选项”。

“选项”对话框即会打开到“常规”页面。

- 2 单击“数据库”选项卡。

此时将显示“数据库”页面。

- 3 选择“检测联接中的基数”复选框。



- 4 单击“确定”。
- 5 创建新的联接时，会自动检测基数并将其显示在联接上。

通过“编辑联接”框自动检测基数

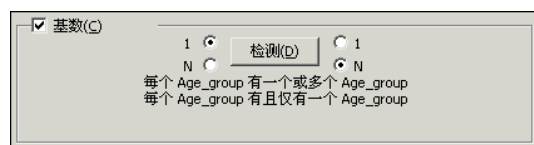
通过“编辑联接”框自动检测基数：

- 1 双击某个联接。

将出现“编辑联接”对话框。

- 2 选择“基数”复选框。
- 3 单击“检测”按钮。

将自动为检测的基数选择基数单选按钮。也会以句子的形式表示这两个基数。



- 4 单击“确定”。

4.9.2.3 优化自动基数检测

可以通过修改目标 RDBMS 的 PRM 文件中的参数来改进基数检测的响应时间。这会指引检测算法读取两个而不是三个 SQL 语句，从而改进了此算法的性能。

PRM 文件是一个文本文件，它列出了用于在 Web Intelligence 中配置 Universe 创建和 SQL 查询生成的参数。每个支持的 RDBMS 都有一个 PRM 文件。

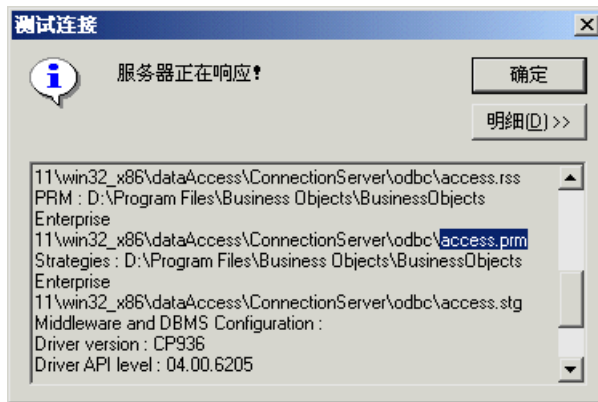
PRM 文件位于 <INSTALLDIR>\win32_x86\dataAccess\ConnectionServer\ 下的数据库文件夹中。

验证连接使用哪个 PRM 文件

验证 Universe 连接使用哪个 PRM 文件：

- 1 选择“文件”>“参数”。

- 将出现“参数”对话框。
- 单击“测试”按钮。
- 将出现“测试连接”消息框。
- 单击“明细”按钮。
- 连接的明细将出现在下拉消息框中。
- 将消息框向下滚动到以 PRM 开始的行。
- 此行指明了活动 Universe 当前使用的 PRM 文件的文件路径和名称。



- 单击“确定”。
- 您将返回到“参数”对话框。
- 单击“取消”。

使用 PRM 文件优化基数检测

使用 PRM 文件优化基数检测：

- 在文本编辑器中打开目标数据库的 PRM 文件。
- PRM 文件存储在 Business Objects 路径中的 Data Access 文件夹内。
- 将 LIGHT_DETECT_CARDINALITY 参数设置为 YES。
 - 保存并关闭 PRM 文件。
- 下次打开 Universe 时，会优化自动基数检测。

4.9.2.4 使用基数解决数据库限制

可以使用以下准则来确定特殊联接情况下的基数，而如果不处理这些情况，可能会导致模式设计出错：

问题	解决方法
查找表的主键有两列。每列均被联接到不同的事实表。由于两个联接中的主键均不完整，因此，每个事实表的联接均为多对多形式。	<p>对于在查找表端的联接，将“多”端更改为“一”端。按如下方式执行此操作：</p> <p>在此类型的查找表上添加一个自我限制的联接（一对一）；<code>lookup.pk_column = pk_column value</code>。这将确保查找表的主键中的值为唯一。在查找表端的联接的基数现在为一。</p>
主键过多，因此并不需要主键中的所有列来保证唯一性。	<p>如果您是目标数据库的 DBA，则可以将多列主键更改为单一列字母数字标识符。这将使表可以占据联接的“一”侧（这在多列主键的情况下则困难得多）。如果您不是 DBA，则可以向管理员提出这一点。</p>

4.10 检查 Universe

在设计 Universe 时，应定期测试其完整性。可以按如下方式验证 Universe 的完整性：

检查 Universe	说明
自动	可以设置 Universe 设计工具选项，以便在创建、导出 Universe 或在打开 Universe 时检查 Universe 结构的 SQL 语法。
手动	可以运行“检查完整性”以检查选定 Universe 结构。

4.10.1 自动检查 Universe 完整性

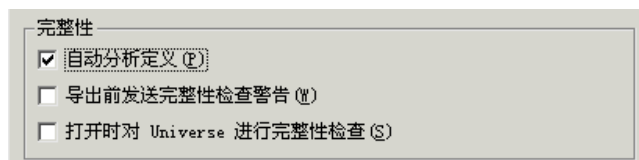
可以在 Universe 设计工具中设置以下完整性检查选项，以便在创建、导出 Universe 或在打开 Universe 时分析 SQL 结构：

自动检查选项	说明
依据定义的自动解析	Universe 设计工具将在创建时自动检查所有对象、条件和联接的 SQL 定义。它将在您单击“确定”以验证结构创建时应用。
导出前发送完整性检查警告	当每次尝试导出未经检查的 Universe 时，Universe 设计工具都会显示一条警告。
打开时检查 Universe 完整性	所有 Universe 在打开时均会自动受到检查。

4.10.1.1 设置自动 Universe 检查选项

设置自动 Universe 检查选项：

- 1 选择“工具”>“选项”。
- “选项”对话框即会打开到“常规”页面。
- 2 在“完整性”组框中，选择或清除相应 Universe 自动检查选项的复选框。



- 3 单击“确定”。

4.10.1.2 手动检查 Universe 完整性

可以使用“检查完整性”来测试活动 Universe 的设计是否准确并保持最新。

“检查完整性”将检测以下各项：

- Universe 的对象、连接、条件及基数中的错误。
- 联接路径中的环路。
- 任何必需的上下文。
- 对目标数据库所做的更改。

在依据这些数据库检查 Universe 的各个元素之前，该功能将检查与数据库的连接是否有效。如果连接无效，则功能将停止并返回错误消息。

4.10.1.3 “检查完整性”检测的错误类型

“检查完整性”可以检测：

- 对象、条件或联接的 SQL 定义中的无效语法。
- 环路
- 孤立的表
- 孤立联接
- 上下文内的环路
- 缺少或不正确的基数

“检查完整性”如何确定连接数据库中的更改？

“检查完整性”功能会向数据库发送一个请求，以获得表的列表。然后，它会将此列表与 Universe 中的表进行比较。它将对列执行同样的操作。

在“结构”窗格中，“检查完整性”会将与列表中的表或列不匹配的任何表或列标记为不可用。这些表或列在数据库中可能已被删除或重命名。请参阅第 205 页上的 [“刷新 Universe 结构”](#) 一节。

注意：

如果检查大量的数据，则“检查基数”选项可能运行很慢。如果有数据不明确或缺少数据，结果也可能不准确。如果数据库较大并且可能包含不完整的数据条目，则不应选择“检查基数”选项。如果确实要使用此选项，则可以通过修改 PRM 文件来优化基数检测。有关更多信息，请参阅第 198 页上的[“优化自动基数检测”](#)一节。

4.10.1.4 使用“检查完整性”验证 Universe 的完整性

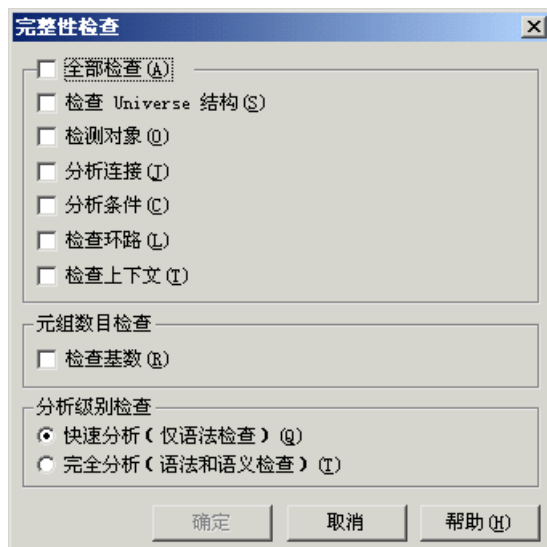
验证 Universe 完整性：

1. 选择“工具”>“检查完整性”。

或

单击“检查完整性”按钮。

2. 将出现“完整性检查”对话框。



3. 选择要验证的元素的复选框。

注意：

可以独立于“全部检查”选项选择“检查基数”。这使您无需检查基数即可验证 Universe 结构（视数据库而定，检查基数可能要花很长时间）。

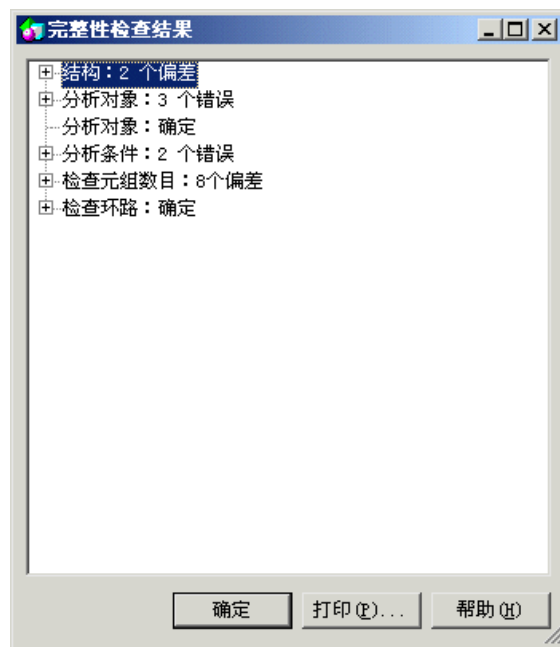
4. 清除不要验证的元素的复选框。
5. 选择“快速分析”复选框，以便只验证元素的语法。

或

选择“完全分析”复选框，以便同时验证元素的语法和语义。

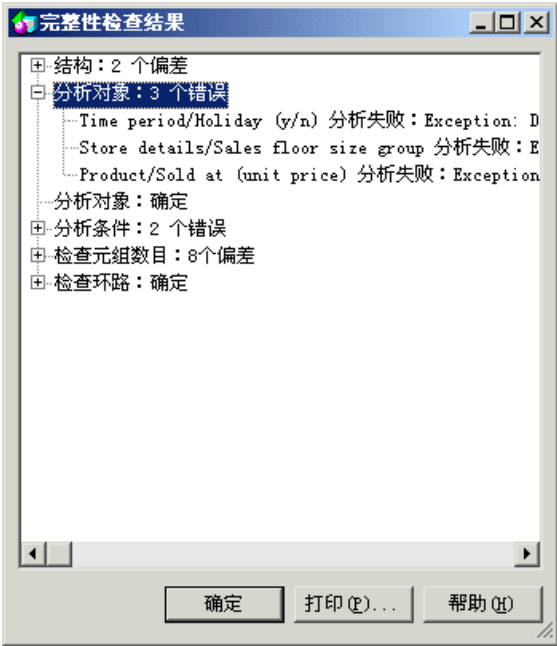
- 6 单击“确定”。

消息框将显示 Universe 检查进度。



如果“检查完整性”未遇到错误，则会在每种错误类型旁边显示“确定”。

- 7 单击错误类型旁边的加号 (+)，查看发生错误的元素的列表。



可以双击列表中的项目，以便突出显示“结构”窗格中对应的元素。

- 8 单击“打印”按钮打印窗口内容。
- 9 单击“确定”。

注意：

在选择“检查环路”复选框之前，请确保已检测了联接的基数。否则，该功能将错误地标识联接中的环路。

4. 10. 1. 5 刷新 Universe 结构

如果“检查完整性”指出 Universe 连接的数据库已被修改，则可以使用“刷新结构”来更新“结构”窗格的内容。

“刷新结构”可以修改 Universe 结构以符合数据库中的更改，如下所示：

如果	则工具执行以下操作
向表中添加了列	将列添加到 Universe 中对应的表。

如果	则工具执行以下操作
从表中删除了列	显示一则警告消息，指明应删除的列和关联联接。
从数据库中删除了表	显示一则警告消息，指明应删除的表和关联联接。
在数据库中重命名了表	显示一则消息，说明它不再识别 Universe 中对应的表。应重命名这些表，以便与数据库中的表匹配。如果名称仍然不匹配，Universe 设计工具将返回一条消息，指明重命名的表在数据库中不存在。
未对数据库进行更改	显示一则消息，通知您无需进行更新。

刷新 Universe 结构：

- 选择“查看”>“刷新结构”。
- 将出现一个消息框，通知您数据库中的更改，或通知您无需更新（如果未进行更改）。

第 5 章 解决模式中的联接问题

5.1 概述

本章描述在模式中的表之间创建联接时可能出现的问题的类型。它说明了如何能够检测 and 解决这些联接问题，以便确保在 Universe 上运行的查询所采用的联接路径返回正确的结果。必须先解决联接问题，才能构建 Universe。

5.2 什么是联接路径问题？

联接路径是一系列的联接，查询可以使用它们来访问这些联接所链接表中的数据。

从查找和事实表与关系数据库相关的有限方式中，可能会产生联接路径问题。在设计模式时，您碰到的三个主要联接路径问题如下：

- 环路
- 断层陷阱
- 扇形陷阱

所有这些问题都能解决，方法是创建别名（基表的副本）、上下文（定义的联接路径），然后使用 Universe 设计工具中提供的功能来分隔对度量或上下文的查询。

本节简要定义了查找和事实表，并描述了在使用这些表时可能会遇到的联接路径问题的类型。它说明可以如何使用别名、上下文和其他 Universe 设计工具功能来解决 Universe 模式中的联接路径问题。

在 Universe 设计工具中，用户通常会在查找表和事实表之间创建联接。

5.2.1 什么是查找表

查找（或维）表包含与特定实体或主体关联的信息。例如，查找表可以容纳有关客户的地理信息，如他们的姓名、电话号码以及他们居住的城市和国家/地区。

在 Universe 设计工具中，维和明细对象通常是从查找表中派生的。

5.2.2 什么是事实表

事实表包含有关事务的统计信息。例如，它可能包含诸如“Sales Revenue（销售收入）”或“Profit（利润）”等数字。

在 Universe 中，大多数（但不是全部）度量都是通过事实表定义的。

5.2.3 什么类型的联接路径返回不正确的结果？

由于在关系数据库中执行联接的方式存在限制，因此查询可能会返回不正确的结果。取决于查找表和事实表在表模式中的相关方式，联接路径可能会生成查询在其中返回太少或太多行的实例。

以下类型的联接路径可能会产生不正确的结果：

联接路径的类型	返回	说明
环路	太少行	联接在查找表之间形成多个路径。

联接路径的类型	返回	说明
会聚式多对一联接	太多行	两个事实表中的多对一联接会聚在单一查找表上。此类型的联接会聚可能会导致名为断层陷阱的联接路径问题。
系列式多对一联接	太多行	一对多联接链接了依次由一对多联接链接的表。一对多联接这种类型的分散方式可能会导致名为扇形陷阱的联接路径问题。

5.2.4 检测和解决联接问题

Universe 设计工具提供了多种用于检测和解决联接问题的方法。相应的各节中详细描述了其中的每种方法。

可以使用以下方法来检测和解决联接路径问题：

联接问题	检测方式	解决方式
环路	<ul style="list-style-type: none"> 检测别名 检测上下文 检测环路 检查完整性 模式的直观分析 	创建别名和上下文以中断环路。
断层陷阱（会聚式多对一联接）	表模式的直观分析。	<ul style="list-style-type: none"> 创建上下文。 使用功能“为每个度量使用多个 SQL 语句”。 创建多个 Universe（仅限 Web Intelligence）。

联接问题	检测方式	解决方式
扇形陷阱（序列式多对一联接）	表模式的直观分析。	<ul style="list-style-type: none">创建别名，使用别名创建上下文，然后依据别名建立受影响的度量对象。为每个度量使用多个 SQL 语句。

大多数联接路径问题都可以通过创建别名或实施上下文来解决。可以使用 Universe 设计工具中的自动环路检测工具来确定模式中的环路，并使用自动上下文检测来确定断层陷阱出现在何处。但是，要解决扇形陷阱，您必须能够对模式进行直观的分析 and 创建别名，并能够在必要时手动创建上下文。

5.3 定义别名

别名是对模式中现有的表的引用。别名是一个表，它是与原始表（基表）完全相同的副本，只是名称不同。该表中的数据与原始表完全相同，但不同的名称使查询的 SQL “误以为” 您在使用两个不同的表。

Beach Universe 模式包含两个别名表：Resort_Country（Country 表的别名）和 Sponsor（Customer 表的别名）。针对每个别名表，会以括号形式显示原始表的名称。

5.3.1 如何在模式中使用别名？

您出于以下两个主要原因使用别名：

- 在一个查询中多次使用表。这是使用别名的主要原因，并包括使用别名来解决环路和扇形陷阱。示例 Beach Universe 包含两个别名：Country 的 Resort_Country，以及 Customer 的 Sponsor。
- 在编写自由编写 SQL 时简化表名称以减少键入。

提示：

别名的另一种使用方法是：在模式中插入表时为每个表创建一个别名。然后，使用别名表（而不是原始基表）来构建模式。将基表一起放在远离主要 Universe 结构的位置。这允许您为表指定有意义的名称，并且，如果在稍后需要为基表指定别名，则无需重建 Universe 结构的主要部分。

5.3.1.1 使用别名来解决环路

在 Universe 开发中，别名的最常见用法是解决使用常见表时可能出现的环路。环路是一组连接，它定义了经过模式中的一组表的闭合联接。当联接形成查找表之间的多个联接时，将出现环路

可以使用别名来中断环路，方法是正在用于多个查询路径的原始查找表提供另一个表。第 225 页上的 [“解决环路”](#) 一节中讨论了别名的这种用法。

5.3.1.2 使用别名来解决扇形陷阱

别名还用于解决潜在的扇形陷阱。这些扇形陷阱可能会出现在这样的序列式一对多联接路径中：在联接的“多”端对聚合进行加总时，该路径可能会返回过度膨胀的结果。第 251 页上的 [“解决断层陷阱”](#) 一节中讨论了别名的这种用法。

5.3.2 创建别名

可以手动创建别名，也可以让 Universe 设计工具自动检测将可能解决联接路径环路的别名。

需要手动创建别名来解决扇形陷阱。如果要创建只使用别名而不使用基表的模式，则也需要手动创建别名。

第 236 页上的 [“检测和创建别名”](#) 一节描述了自动检测和创建别名来解决环路的过程。

5.3.2.1 手动创建别名

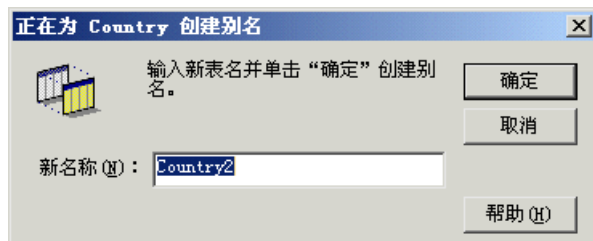
手动创建别名：

- 1 单击要用于创建别名的表。
- 2 选择“插入”>“别名”

或

单击“插入别名”按钮。

将出现“创建别名”框。它提示您输入新别名的名称。



- 3 输入已指定别名的表的名称，或保留建议名称。

注意：

为别名指定的名称应与别名的角色相关，以便将别名与基表区分开来。例如，Resort country 是 Country 的别名。Resort Country 用于返回休假地国家/地区数据的查询，基表 Country 用于返回客户国家/地区数据的查询。

- 4 单击“确定”。

指定了别名的表即出现在“结构”窗格中。

- 5 在模式中的别名和其他表之间创建任何必需的联接。

提示：

为了避免将基表与别名混淆，您可以将别名与它在表标题中所代表基表的名称一起显示，如下所示：选择“工具”>“选项”>“图形”，然后选择“别名”复选框。

5.3.2.2 重命名别名

可以随时重命名别名。别名和表命名约定取决于 RDBMS。您可以通过重命名表直接重命名别名，也可以从 Universe 的别名值列表中重命名别名。

直接重命名别名

直接重命名别名：

- 1 单击一个表，然后选择“编辑”>“重命名表”。

或

右键单击一个表，然后从上下文菜单中选择“重命名表”。

将出现“重命名表”对话框。



- 2 在“表名称”框中键入新的名称。

“所有者”和“限定”字段的可用性特定于数据库。如果这些字段处于活动状态，则可以根据需要修改他们。

- 3 如果希望别名显示为全大写，请选择“大写”复选框。

或

如果希望别名显示为全小写，请选择“小写”复选框。

- 4 单击“确定”。

从列表中重命名别名

从列表中重命名别名：

- 1 选择“工具”>“别名列表”。
- 2 将出现“别名列表”。它列出了活动 Universe 中的所有别名。
- 3 单击列表中的一个别名。
- 4 为选定别名在“新名称”文本框中键入一个新名称。
- 5 单击“应用”。
- 6 单击“确定”。

5.3.2.3 删除别名

可以按与删除表相同的方式删除别名。如果已经使用别名定义了对象，则必须在删除别名之前修改这些对象以使它们使用另一个表，或删除不再必要的对象。

如果不修改或删除使用已删除别名的对象，则使用这些对象的查询将在 Web Intelligence 中产生错误。

删除别名：

1. 单击一个别名，然后选择“编辑”>“清除”。

或

右键单击一个别名，然后从上下文菜单中选择“清除”。

或

单击一个别名并按 DELETE 键。

如果任何对象使用该别名，则会出现以下消息：



如果没有对象使用该别名，您不会收到确认框。别名将被立即删除。

2. 单击“是”。

即会从“结构”窗格中删除别名。

5.4 定义上下文

上下文是一组联接的集合，它为 Web Intelligence 提供了生成 SQL 的有效查询路径。

5.4.1 如何在模式中使用上下文？

可以出于以下目的在 Universe 模式中使用上下文：

- 解决环路。
- 解决断层陷阱。
- 协助某些解决扇形陷阱的情况。
- 协助检测使用聚合感知的对象的不兼容性。

5.4.1.1 使用上下文来解决环路

上下文的最常见用法是分隔两个查询路径，以使一个查询返回一个事实表的数据，另一个查询返回另一个事实表的数据。可以使用上下文在包含多个事实表的模式中引导联接路径。在此类模式中，别名并不适合。第 225 页上的 [“解决环路”](#) 一节中论述了上下文的这种用法。

5.4.1.2 使用上下文来解决断层陷阱和扇形陷阱

上下文还用于解决潜在的断层陷阱。当两个多对一联接路径会聚在一个表上时，可能会出现这些陷阱。可能会为单一维返回导致结果过度膨胀的多个行。上下文可以分割出查询，以便为维返回正确的行数。上下文也可以与别名一起使用以解决扇形陷阱。第 251 页上的 [“解决断层陷阱”](#) 一节中论述了上下文的这些用法。

5.4.1.3 使用上下文来确定聚合感知不兼容性

可以使用上下文来排除与定义中使用 @AggregateAware 函数的对象不兼容的对象，以便不在包含聚合感知对象的查询中使用它们。

5.4.2 创建上下文

可以让 Universe 设计工具自动检测上下文，也可以手动创建上下文。

如果要使用上下文来解决环路或断层陷阱，则应始终让 Universe 设计工具检测上下文。但是，为了解决扇形陷阱（另一种联接路径问题），您可能必须手动建立上下文。

第 225 页上的 [“解决环路”](#) 一节中说明了用于解决环路的上下文自动检测。

注意：

在创建一个或多个上下文时，所有联接均必须包括在一个或多个上下文内。如果通过不包括在上下文中的某个联接链接了表，则查询在运行时将不会考虑该联接。

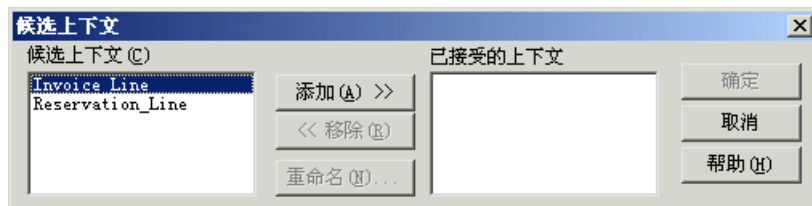
以下过程描述了如何能够自动和手动创建上下文。

5.4.2.1 自动创建上下文

自动创建上下文

- 1 选择“工具”>“自动检测”>“检测上下文”。

将出现“候选上下文”框。它列出了模式的建议候选上下文。这些候选上下文对于解决环路或断层陷阱可能是必需的，因为断层陷阱存在于两个上下文相交之处的分支中。



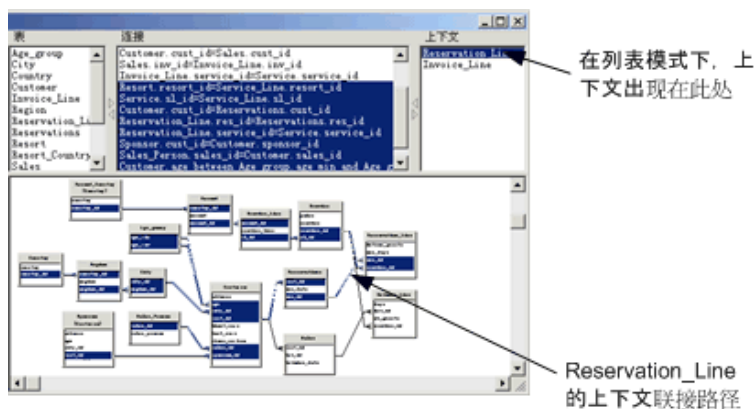
- 2 单击“候选上下文”列表中的一个上下文，然后单击“添加”按钮。
- 3 为列表中的每个候选上下文重复步骤 2。

注意：

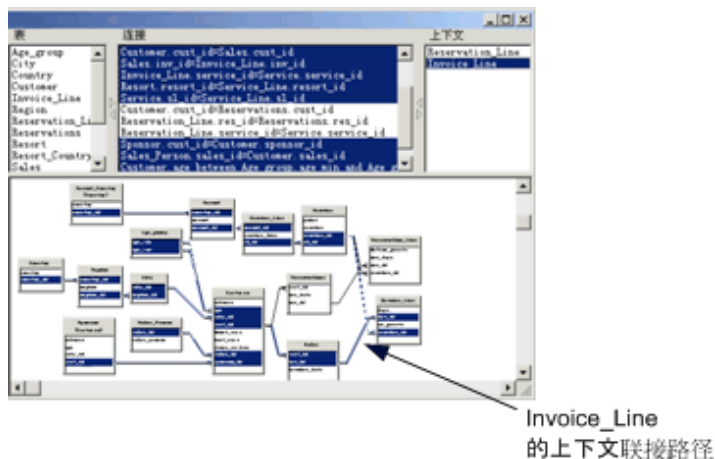
将候选上下文添加到“接受的上下文”列表之后，您就可以按如下方式重命名上下文：单击一个上下文，然后单击“重命名”按钮。将出现一个编辑框。键入新的名称，然后单击“确定”。

- 4 单击“确定”。

当列表模式（“查看”>“列表模式”）处于活动状态时，这些上下文列在“上下文”窗格中。Invoice_Line 的上下文如下所示。



5 Invoice_Line 的上下文如下所示。

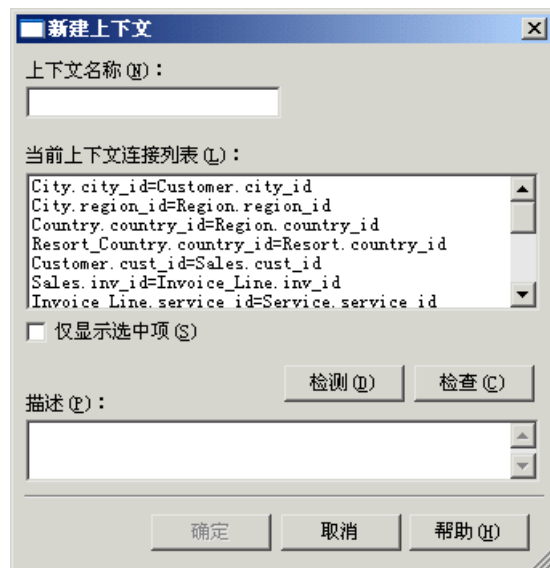


5.4.2.2 手动创建上下文

手动创建上下文：

- 1 选择“插入”>“上下文”。
- 或
- 单击“插入上下文”按钮。

将出现“新建上下文”框。



- 2 在“上下文名称”文本框中键入上下文的名称。
- 3 在“当前上下文联接”列表中选择定义上下文的所有联接。

在创建上下文时，您可以选择进行以下操作：

- 4 单击“检测”按钮，以便显示和上下文名称一起组成建议上下文的联接。
- 5 选择“仅显示选定项”复选框，以便仅查看选定的联接。
- 6 单击“检查”按钮。

Universe 设计工具检查选定联接以查看是否存在任何环路。

- 7 键入上下文所返回数据的说明。这是 Web Intelligence 用户在运行采用该上下文路径的查询时所看到的帮助文本。此上下文应对最终用户有所帮助。
- 8 单击“确定”。

上下文即已创建。

5.4.3 编辑上下文

可以使用上下文编辑器来修改上下文的以下属性：

- 名称

- 上下文中包括的联接
- 说明

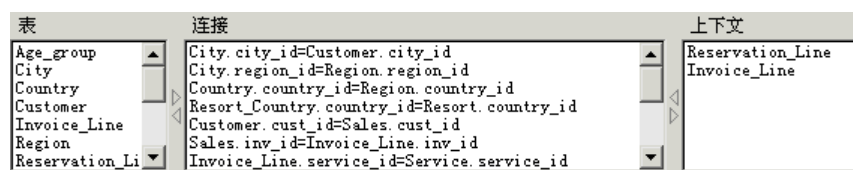
也可以检查任何未解决环路的上下文。

5.4.3.1 编辑上下文属性

编辑上下文属性：

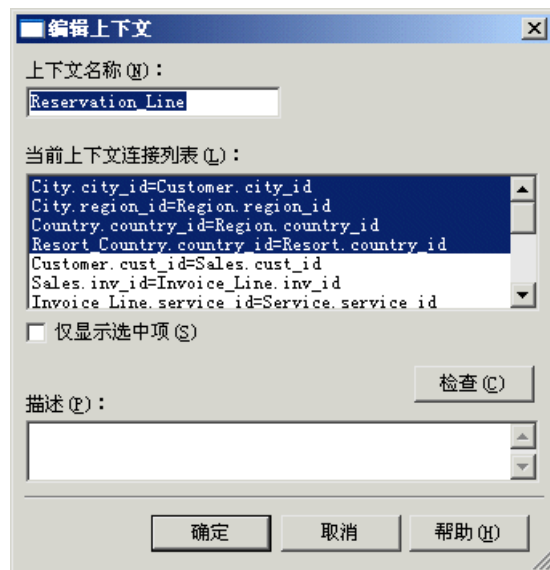
1. 选择“查看”>“列表模式”。

“列表”窗格将出现在“结构”窗格的上方。它包含“结构”窗格中所有表、联接和上下文的列表框。



2. 双击“上下文”列表窗格中的某个上下文名称。

将出现“编辑上下文”框。



3. 如果要更改上下文名称，请在“上下文名称”框中键入一个新名称。

- 4 单击突出显示的联接，将其从上下文中删除。
或
单击未突出显示的联接，将其添加到上下文中。
- 5 键入上下文的说明。
- 6 单击“确定”。
修改内容即会出现在上下文中。

5.4.4 删除上下文

可以随时从“列表”窗格的“上下文”列表中删除上下文。如果要在上下文内添加或删除表或联接，应在修改表或联接之前删除上下文。

完成修改之后，您可以手动重新创建上下文（如果它用于解决断层陷阱），也可以使用“检测上下文”来自动检测新的上下文（如果它用于解决环路）。有关检测上下文的信息，请参阅第 238 页上的[“检测和创建上下文”](#)一节。

5.4.4.1 从上下文列表中删除上下文

从上下文列表中删除上下文：

- 1 确保列表模式处于活动状态（选择“查看”>“列表模式”）。
- 2 右键单击“上下文”列表框中的某个上下文名称，然后从上下文菜单中选择“清除”。
或
单击“上下文”列表框中的某个上下文名称，然后选择“编辑”>“清除”。
即会从列表中删除该上下文。

5.4.5 更新上下文

当 Universe 结构发生变化时，上下文不会自动更新。如果将任何表添加到结构或从中删除了任何表，或者添加或删除了任何联接，则必须更新所有上下文。

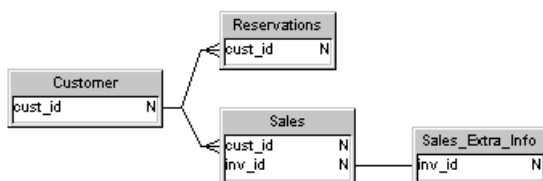
如果只对结构进行了简单更改，您可以使用“编辑上下文”框或“列表”窗格手动更新每个上下文中包括的联接。但是，如果对 Universe 结构进行了大量的更改，则应删除当前上下文并重新创建它们。

5.4.6 阻碍上下文检测的联接路径

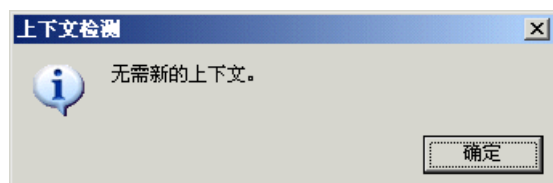
位于联接路径末端的一对一基数可能会阻止 Universe 设计工具中的“上下文检测”功能检测上下文。通过将联接路径末端的表的基数更改为一对多，您可以解决此问题。

示例：阻碍上下文检测的一对一基数

下面的模式显示了一个表 Sales_Extra_Info，该表包含有关每个销售的特定信息。它通过一对一联接来联接到 Sales 表。



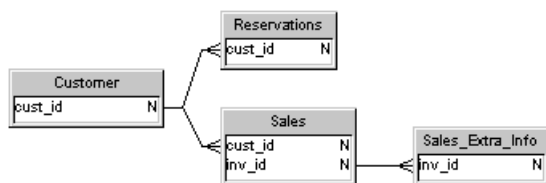
当您以直观方式对联接路径进行检测时，此模式中明显有两个上下文：一个 Reservations 上下文和一个 Sales 上下文。但是，在此类型的联接路径上自动检测上下文时（“工具”>“自动检测”>“检测上下文”），将会收到以下消息：



Universe 设计工具没有在上文检测中考虑到联接路径末端的一对一联接，因此不会考虑存在两个上下文的情况。

5.4.6.1 更改基数以允许上下文检测

通过将联接（将 Sale_Extra_Info 链接到 Sales）的基数设置为一对多，您可以解决此问题。它也可以为多对一，关键因素在于联接路径的末端不能有一对一联接。现在，下面的模式在联接路径的末端具有一对多联接。



当您运行“检测上下文”时，即会检测到两个上下文，如下所示：



5.4.7 上下文如何影响查询？

取决于您如何允许 Web Intelligence 用户使用模式结构上定义的对象，上下文可能会导致三种类型的查询运行：

- 不明确查询
- 推导式查询
- 不兼容查询

可以在 Web Intelligence 中运行这些类型的查询，以便测试上下文生成的 SQL。如果任何这些查询类型产生了错误或返回了不正确的数据，则需要分析有关的联接路径。

5.4.7.1 不明确查询




提示最终用户在两个查询路径之间进行选择。当综合使用查询所含对象时，生成的信息不足以确定上下文，则出现不明确查询。

如果查询不明确，Web Intelligence 将显示一个对话框，提示用户在两个上下文中选择其中之一。当用户选择了上下文时，即会将对应的表和联接插入 SQL 查询。

示例：运行不明确查询

Web Intelligence 用户可运行以下查询：

为我提供每个休假地每个年龄段的游客所使用的服务

 Service  Age group  Resort

在运行查询时，将出现一个要求用户选择上下文（在本例中为 Reservations 或 Sales 上下文）的对话框：

用户必须选择他们是需要有关某个年龄段的游客所预订服务的信息，还是需要某个年龄段的游客已付款服务的信息。如果选择 Reservation 上下文，则会生成以下 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 )
```

另一个上下文 (Sales) 所引用的联接不会出现在 SQL 中。

5.4.7.2 推导式查询

Web Intelligence 查询在运行时不会提示最终用户选择上下文。查询包含用于推导出正确上下文的足够信息。例如，用户运行以下查询：

为我提供每个可用服务按年龄段列出的未来宾客数：

 Service  Age group  Future guests

当查询运行时，无需提示用户选择上下文即会返回数据。Future Guests 对象是 Reservation_Line 表的总和，该表是 Reservations 上下文的一部分。Web Intelligence 推导出查询应使用 Reservation 上下文。




5.4.7.3 不兼容查询

来自两个不同上下文的对象合并在一个查询中。两个 Select 语句在各自的表中同步显示返回的数据。

示例：运行不兼容查询

Web Intelligence 用户可运行以下查询：

为我提供公司范围内宾客的总数以及进行预订的月份。

 Number of guests  Age group  Reservation Month

Web Intelligence 推导出查询将同时使用 Sales 和 Reservation 上下文，因此在运行查询时不会出现提示。两个上下文的 Select 语句将会同步，如下所示：

```
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
```

查询分为两个部分：

- 年龄段和宾客数
- 预订月份

在检索两个查询的结果时，Web Intelligence 会（使用“年龄组”）合并结果。然后，将在同一报表的两张表中显示结果，如下所示。

18-30

Number of guests	Reservation Month
451.00	Apr
	Aug
	Feb
	Jan
	Jun
	May
	Nov
	Oct
	Sep

为了在 Web Intelligence 中运行不兼容查询，必须在 Universe 设计工具中选择“对上下文使用多重 SQL 语句”选项。下一节中描述了这一点。

5.4.7.4 选择“为每个上下文使用多个 SQL 语句”

选择“为每个上下文使用多个 SQL 语句”：

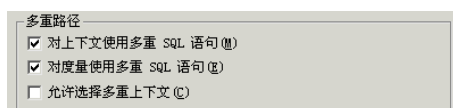
- 1 选择“文件”>“参数”。

将出现“Universe 参数”对话框。

- 2 单击“SQL”选项卡。

将出现“SQL”页面。

- 3 选择“为每个上下文使用多个 SQL 语句”复选框。



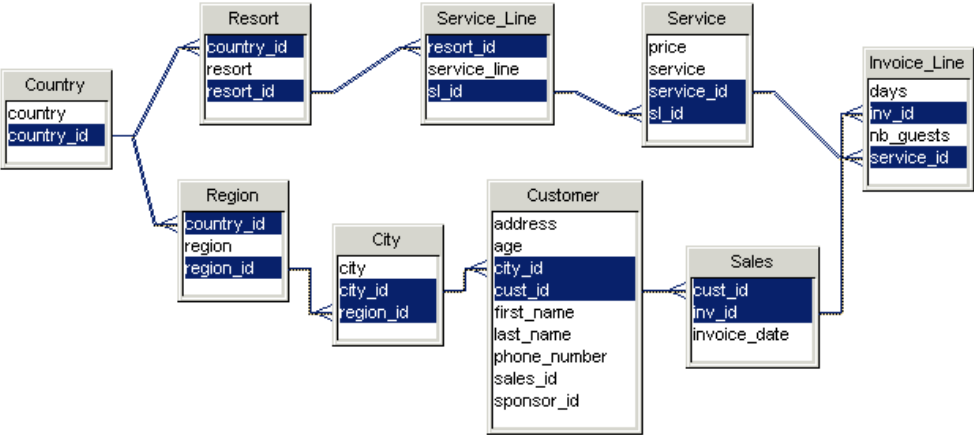
- 4 单击“确定”。

5.5 解决环路

在关系数据库模式中，返回太少行的一种常见联接路径称为环路。

5.5.1 什么是环路？

环路是一组连接，它定义了经过模式中的一组表的闭合联接。当联接形成查找表之间的多个路径时，将出现环路。下面显示了环路的一个示例。



该模式包含两个链接的信息集：

对于每个...	链接了以下信息
休假地	可用服务系列、每个服务系列的服务、每个服务的发票信息以及休假地所在的国家/地区。
客户	客户居住的城市、地区和地区、客户的销售以及每个销售的发票信息。

这两组信息链接在一个形成环路的常见联接路径中。查找表 Country 可以是休假地所在的国家/地区，也可以是客户居住的国家/地区。

5.5.1.1 为何环路出现在 Universe 模式中而不出现在数据库中？

在数据库中，表之间的多个路径可能是有效的，并可以实施以满足特定的用户需求。如果在查询中单独包括每个路径，查询将返回不同的结果集。

但是，在 Universe 设计工具中设计的模式通常需要允许包括多个路径的查询，但关系数据库可能没有设计为可处理该查询，因此返回的信息可能不正确。

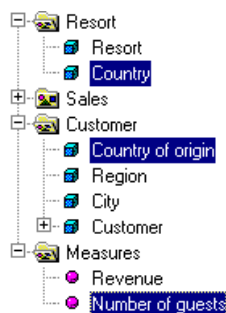
返回的行是每个路径的结果的交集，因此返回的行比预计的要少。并且，通常很难在检查结果时确定问题。

5.5.2 环路如何影响查询？

如果基于上述结构创建了 Universe，则依据环路中的表运行的任何查询都只会返回度假地的国家/地区值与原始客户所在国家/地区值相等的结果。这种对共享查找表 Country 的双重限制返回的行比预计的要少。




示例：环路返回不正确的结果

使用包含上述环路的模式创建了以下对象：



用户可以在 Web Intelligence 中运行以下查询：

对于每个度假地国家/地区，为我提供在每个度假地停留的来自每个国家/地区的宾客数量。

 Country  Country of origin  Number of guests

您期望以下类型的结果：

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

对于法国和美国的休假地，您有在这些国家/地区的休假地停留的德国、日本和美国游客数。

但是，当您使用包含环路的 Universe 运行查询时，将会收到以下结果：

Country	Country of origin	Number of guests
US	US	431.00

此结果仅列出停留在美国休假地的来自美国的游客。没有来自任何其他国家/地区的其他游客。

5.5.2.1 环路会对查询产生什么影响？

“结构”中的联接用于在查询的推导 SQL 中创建 Where 子句。这些联接的目的是限制查询返回的数据。在环路中，联接会应用比预期更多的限制，因此返回的数据不正确。

环路创建的 Where 子句如下所示：

```
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' )
```

以下两个联接都对 Country 表应用了限制：

- Country.country_id=Resort.country_id
- Country.country_id=Region.country_id

Country 用于两个目的：

- 查找休假地国家/地区。

- 查找客户原始国家/地区。

这样就创建了一个限制，以便仅在休假地国家/地区和客户所在国家/地区相同时才返回数据。生成的报表只显示游览美国休假地的美国游客的数量。

视环路的特性而定，用户可以使用别名来中断联接路径，或使用上下文来分隔两个联接路径以使查询只能使用其中的一个路径，从而解决 Universe 设计工具中的环路。

5.5.2.2 别名如何中断环路？

通过在同一查询中出于不同的目的使用同一个表两次，别名可以中断环路。别名与基表是一样的，只是名称不同。别名中的数据与原始表完全相同，但不同的名称使 SQL “误以为” 您在使用两个不同的表。

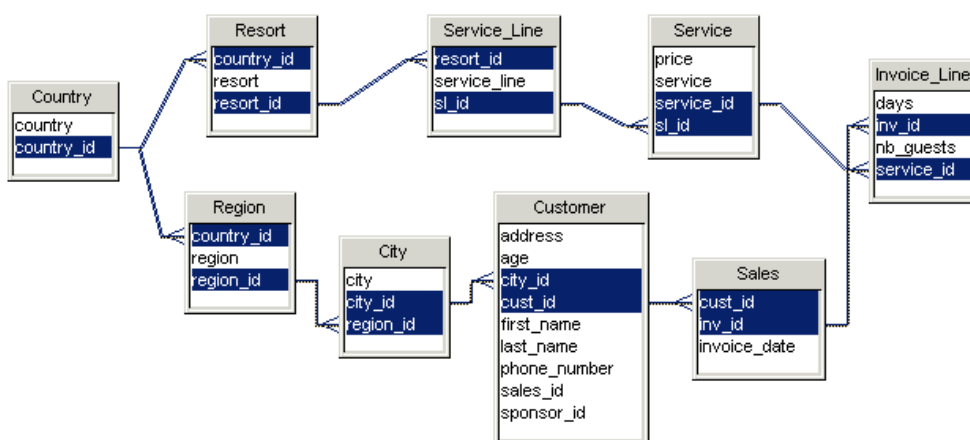
注意：

通过在我们曾经使用的示例中只创建一个别名表，您可以令人满意地解决环路。Region 联接使用原始的 Country 表，而 Showroom 联接使用别名表。但是，可以为原始表中的每个联接都创建一个单独的别名表。在某些关系数据库系统中，这是必需的。

示例：使用别名中断环路

下面的模式与上一节中包含环路的模式相同。它显示了一个联接路径，其中 Country 查找表只接收两个联接的“一”末端，因此可在联接路径中将其用于以下两个目的：

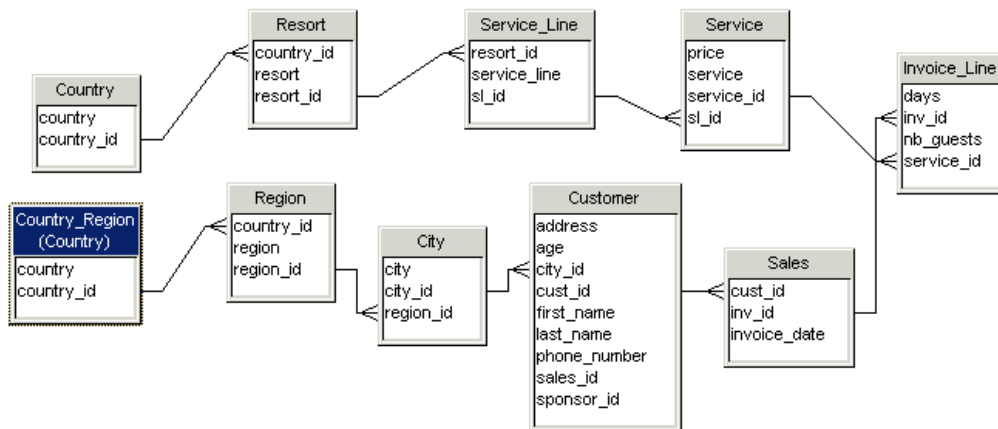
- 休假地国家/地区
- 客户所在国家/地区



您为 Country 创建了一个别名，并将其重命名为 Country_Region。这两个“一”端的联接现在已经分隔开来，如下所示：

- Country 保持与 Resort 表的联接。
- Country_Region 联接到 Region 表。

模式现在看起来如下所示：



当您运行上一示例中所生成行数太少的同一查询时：

对于每个休假地国家/地区，为我提供在每个休假地停留的来自每个国家/地区的宾客数量。

Country Country of origin Number of guests

此查询的 Where 子句现在为：

```

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' )
    
```

现在，一个联接在 Country 表上应用限制，而另一个联接在 Resort_Country 表上应用限制。环路即已中断。

当查询运行时，将返回以下表：

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

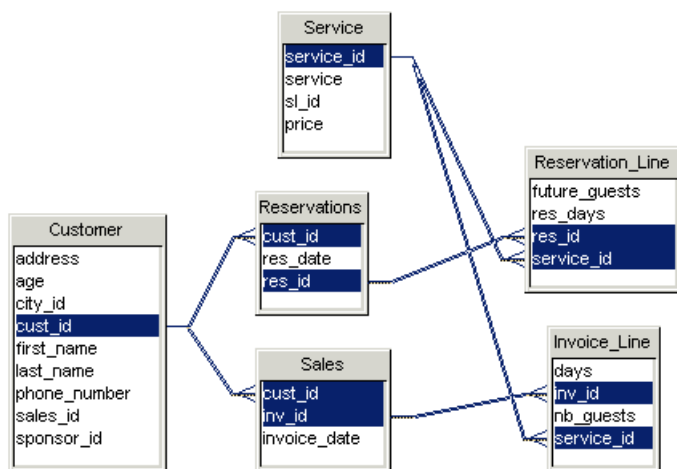
5.5.2.3 上下文如何解决环路？

上下文通过定义一组联接来解决环路，这些联接指定在环路中穿过表的一个特定路径。它确保了不会在相同 SQL 查询的不同路径中包括联接。

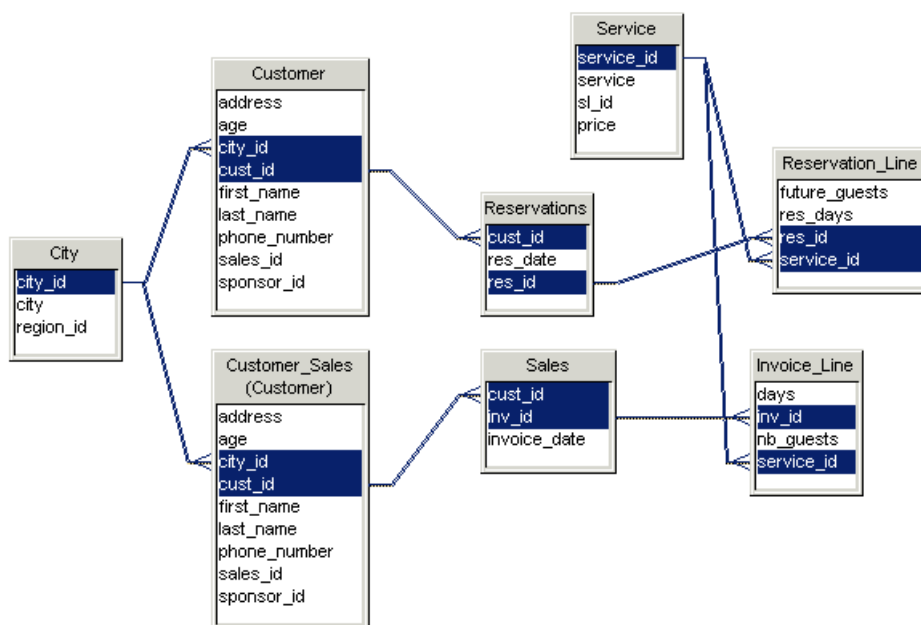
通常在包含多个事实表（“多个星”）的模式中使用上下文，这些事实表共享查找表。

示例：使用上下文解决环路

下面的模式包含有关销售和预订的统计信息。与每种类型的事务相关的统计数据存储在 Sales 和 Reservations 事实表中。该模式包含了一个环路，因为联接路径可以通过销售路径或预订路径来获取服务信息。



如果为 Customer 创建了别名，而因此拥有了 Customer 到 Reservation 的联接和 Customer_Sales 到 Sales 的联接，则您中断了环路。但是，如果要将 City 表添加到模式中，则会再次出现环路，如下所示：



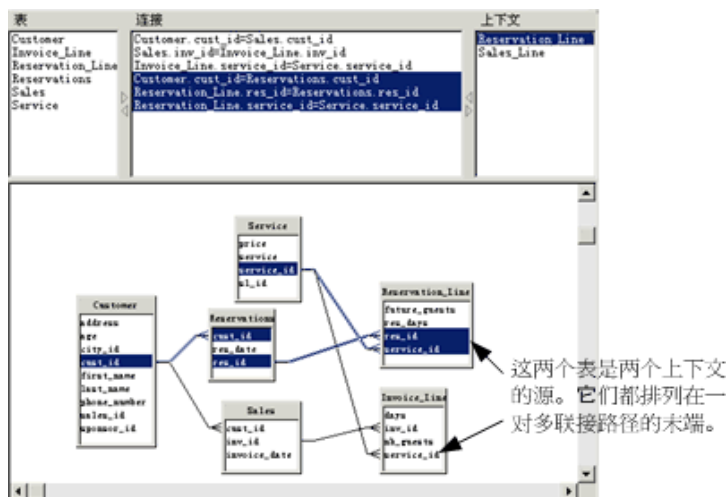
必须继续为添加到模式的每个新表创建别名。这难于维护，并同时阻止了使用每个表的类似对象数在 Universe 中大量增加。

解决此环路的唯一方式是将环路放在适当的位置，并创建一个指定模式周围一个或其他路径的上下文。这确保了查询可以回答一个或另一个事务的问题，诸如：从销售或预订的角度来看，客户信息是否是必需的？

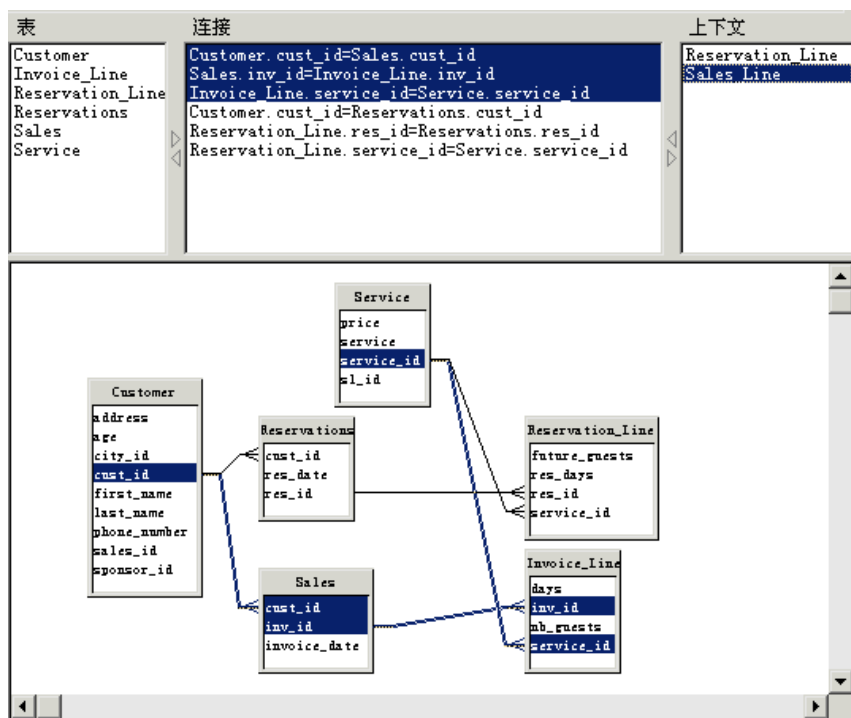
在该示例中，您可以采用从 Customer 表到 Service 表的两条不同路径：

对于此路径...	工具检测以下上下文...
Reservations 和 Reservation_Line	Reservation_Line
Sales 和 Invoice_Line	Sales_Line

以下是 Reservation_Line 上下文：



以下是 Sales_Line 上下文:



然后，您可以根据不同上下文中的表创建不同的对象集。用户随后即可运行 Reservation 查询或 Sales 查询，具体情况取决于他们选择的对象。

5.5.3 直观地标识环路

可以使用以下准则来帮助您分析模式，以便确定别名或上下文是否适合于解决环路。这些准则对于理解模式可能十分有用，但您应使用“检测别名”和“检测上下文”来正式地标识和解决环路。有关更多信息，请参阅第 236 页上的[“检测和创建别名”](#)和第 238 页上的[“检测和创建上下文”](#)两节。

如果环路包含...	则可以通过以下方式解决环路...
唯一一个查找表	别名
仅接收联接的“一”端的查找表	别名
两个或两个以上事实表	上下文

5.5.4 自动标识和解决环路

可以使用 Universe 设计工具来自动检测环路，并建议可插入模式以解决环路的候选别名和上下文。

5.5.4.1 检测环路之前必须设置基数

在使用自动环路检测和解决功能之前，必须为模式中的所有联接设置所有基数。

好的设计做法是：手动定义基数，或手动验证 Universe 设计工具在使用自动例程时建议的每个基数。

可以用两种方式设置基数：

- 手动。有关更多信息，请参阅第 190 页上的 [“使用基数”](#) 一节。
- 使用“检测基数”。有关更多信息，请参阅第 190 页上的 [“使用基数”](#) 一节。

5.5.5 用于检测 and 解决环路的工具功能

可使用 Universe 设计工具中的以下功能确定和解决环路：

使用以下功能来标识和解决环路...	说明
检测别名	<p>检测可指定别名以解决结构中的环路的表，并为每个表建议一个候选别名。可以直接从该框中插入和重命名别名。</p> <p>应在“检测上下文”之前运行“检测别名”，以确保创建的别名包括在您实施的任何上下文中。</p> <p>它不会检测用于解决扇形陷阱的别名的需求。</p>
检测上下文	<p>检测可用于解决结构中的环路的上下文，并建议候选上下文。可以直接从该框中实施和重命名每个上下文。</p> <p>在“检测别名”之后运行“检测上下文”，以确保您实施的任何上下文包括所有新的别名。</p> <p>它不会总是检测用于解决断层陷阱的上下文的需求。如果未检测，则需要手动标识上下文。</p>
检测环路	<p>检测并突出显示结构中的环路。它建议插入别名或上下文来解决每个环路。可以直接从“检测环路”框中实施建议的别名或上下文。</p> <p>使用“检测环路”来快速地检测模式，或以形象的方式显示环路。不要使用它来标识并随后解决环路，因为您无法在插入之前编辑或查看候选别名。</p>

5.5.5.1 用于标识和解决环路的一般方法

下面描述了用于检测和解决环路的一般过程。同时提供了详细描述步骤的各个小节。

- 1 验证是否设置了所有基数。
请参阅第 190 页上的 [“使用基数”](#) 一节。
- 2 如果模式需要别名来解决任何环路，请运行“检测别名”进行标识。
有关更多信息，请参阅第 236 页上的 [“检测和创建别名”](#) 一节。
- 3 插入“检测别名”建议的候选别名。
- 4 如果模式需要上下文来解决仅使用别名无法解决的环路，请运行“检测上下文”进行标识。
有关更多信息，请参阅第 238 页上的 [“检测和创建上下文”](#) 一节。
- 5 实施“检测上下文”建议的候选上下文。
- 6 通过创建对象并运行查询来测试已解决的环路。
有关创建对象和测试 Universe 结构的信息，请参阅第 273 页上的 [“创建 Universe”](#) 一章。

注意：

如果要解决已在表上定义了对象的模式的环路，则必须重新定义现在使用别名（而不是基表）的任何对象。

5.5.5.2 检测和创建别名

可以使用“检测别名”来自动检测和指明导致活动 Universe 中出现环路的表。“检测别名”将建议您可以编辑并插入模式的候选表。

注意：

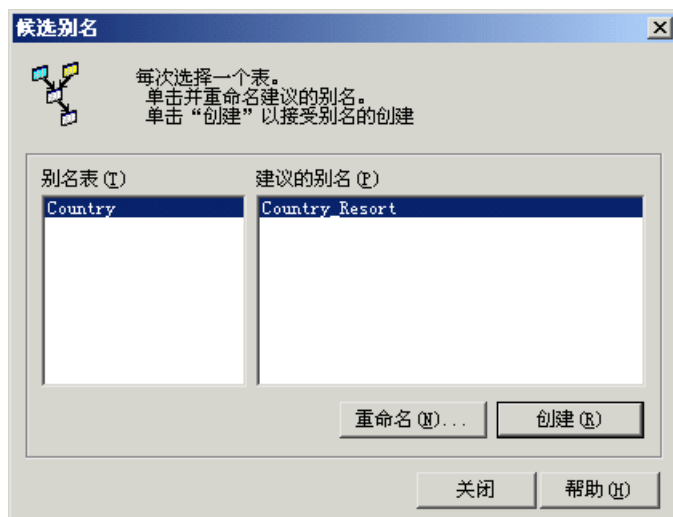
在使用“检测别名”之前，请验证是否通过联接链接了模式中的所有表，以及是否设置了所有基数。

检测和创建别名：

- 1 选择“工具” > “自动检测” > “检测别名”。
- 或

单击“检测别名”按钮。

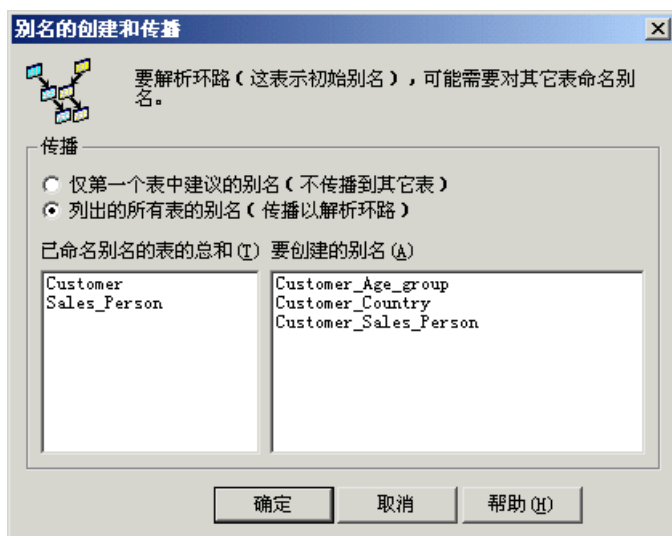
将出现“候选联接”对话框。左边的窗格列出了需要别名的一个或多个表。右边的窗格列出了可插入以中断环路的建议别名。



- 2 在左边的窗格中选择一个表。
候选别名的建议名称即会出现在右边的窗格中。
- 3 如果要重命名建议的别名，请单击“重命名”按钮，并在“重命名”框中输入一个新名称。
- 4 单击“创建”。
一个消息框将提示您确认别名的创建。
- 5 单击“确定”。
别名将出现在“结构”窗格中。
- 6 为所有其他表重复步骤 2 到 5。
- 7 单击“关闭”。

5.5.5.3 检测和创建多个别名

有时在创建别名时，您需要创建其他别名来适应新的联接路径。在使用“检测别名”时，如果 Universe 设计工具检测到需要其他别名，则当用户单击“创建”按钮时，将出现以下对话框：



在这种情况下，您可以选择进行以下两种操作：

- 可以接受只为建议的第一个表指定别名。
- 可以为列出的所有表指定别名。

5.5.5.4 检测和创建上下文

可以使用“检测上下文”来自动检测上下文的需求。“检测上下文”还会建议一个候选上下文。可以在实施候选上下文之前对其进行编辑。

检测和创建上下文：

1. 选择“工具”>“自动检测”>“检测上下文”。

或

单击“检测上下文”按钮。

将出现“候选上下文”对话框。建议的上下文出现在左边的窗格中。



- 单击上下文名称。

候选上下文中包含的表即会在模式中突出显示。

- 单击“添加”按钮。

上下文名称出现在“接受的上下文”窗格中。可以从右边的窗格中删除任何上下文，方法是将其选定，然后单击“移除”按钮。

- 如果适用，请重复步骤 3 和 4 以添加其他上下文。

- 如果要重命名某个上下文，请从右边的窗格中选择该上下文，然后单击“重命名”按钮。

将出现“重命名上下文”对话框。键入一个新名称。

- 单击“确定”按钮。

这些上下文即会列在“Universe”窗口的“上下文”框中。

表	连接	上下文
Age_group	City.city_id=Customer.city_id	Sales
City	City.region_id=Region.region_id	Reservations
Country	Country.country_id=Region.country_id	
Customer	Resort_Country.country_id=Resort.country_id	
Invoice_Line	Customer.cust_id=Sales.cust_id	
Region	Sales.inv_id=Invoice_Line.inv_id	
Reservation_Line	Invoice_Line.service_id=Service.service_id	
Reservations	Resort.resort_id=Service_Line.resort_id	
Resort	Service.sl_id=Service_Line.sl_id	
Resort_Country (Cou	Customer.cust_id=Reservations.cust_id	
Sales	Reservation_Line.res_id=Reservations.res_id	
Sales_Person	Reservation_Line.service_id=Service.service_id	
Service	Sponsor.cust_id=Customer.sponsor_id	
Service_Line	Sales_Person.sales_id=Customer.sales_id	
Sponsor (Customer)	Customer.age between Age_group.age min and Age_group.age max	
Sponsor2 (Customer)	Sponsor2.address=Customer.sponsor_id	
	Sponsor2.first_name=Sales.inv_id	

注意：

如果 Universe 包含意义对于用户不甚明确的环路，则应始终为解决环路的上下文指定一个用户易于理解的名称。Web Intelligence 用户应该清楚地知道上下文所代表的信息路径。

5.5.5.5 自动检测环路

可以使用“检测环路”来检测 Universe 中的环路。此功能将自动检查模式中的环路，并建议用于解决环路的别名或上下文。

要运行快速检查以查找模式中的环路，“检测环路”十分有用。它还会建议用于解决检测到的环路的别名和上下文；但是，相对于使用“检测别名”和“检测上下文”来解决环路而言，您对别名和上下文的创建顺序的控制程度将较低。

第 236 页上的 [“用于标识和解决环路的一般方法”](#) 一节中说明了解决环路的建议过程。

注意：

也可以使用“检查完整性”来自动检查 Universe 结构（包括联接、基数和环路）中的错误。“检查完整性”会为所发现的任何错误提出解决方法。有关更多信息，请参阅第 267 页上的[“手动检查 Universe 完整性”](#)一节。

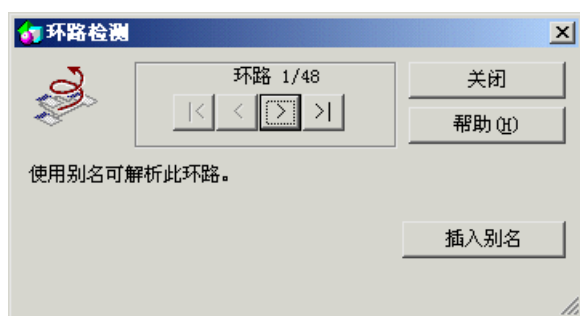
检测模式中的环路：

- 1 验证是否已为模式中的所有联接设置了基数。
- 2 选择“工具”>“自动检测”>“检测环路”。

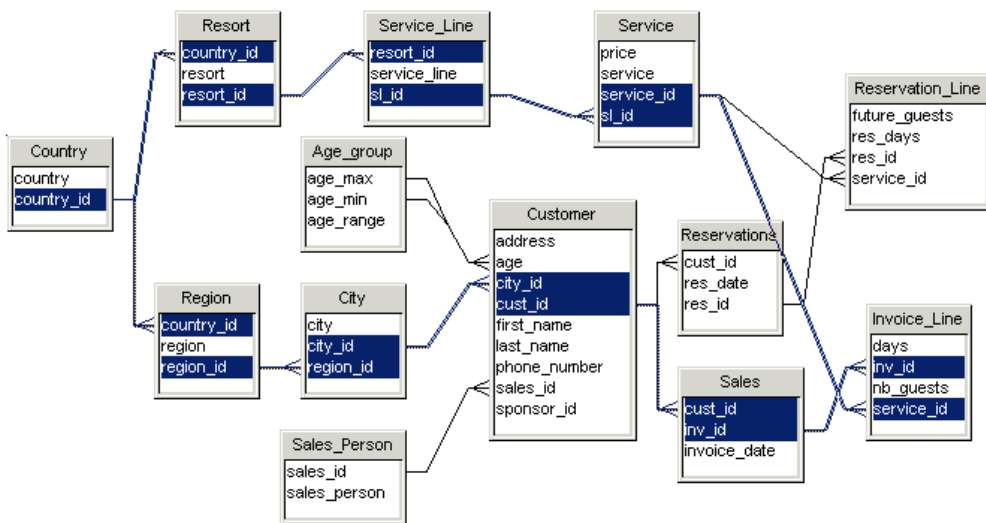
或

单击“检测环路”按钮。

将出现“环路检测”框。它指明检测到了多少环路，并提出可能的解决方法。



检测到的联接路径（形成环路）将同时在“结构”窗格中突出显示，如下所示：



- 3 单击向前按钮可显示下一个环路和建议的解决方法。对于 Universe 设计工具检测到的每个环路，联接路径都会在“结构”窗格中突出显示。

- 单击“关闭”。

5.5.5.6 自动创建别名和上下文

当用户运行“检测环路”时，Universe 设计工具将建议用于解决环路的候选别名或上下文。可以选择从“检测环路”框中直接插入候选别名或实施候选上下文。

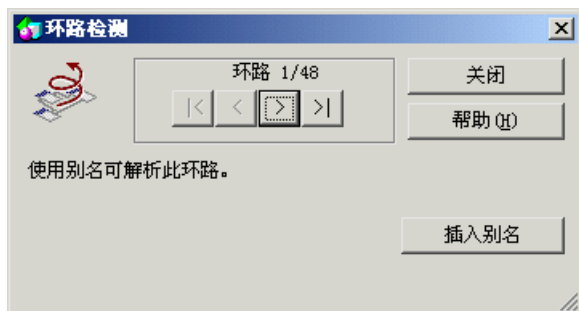
使用“检测环路”创建别名：

- 选择“工具”>“自动检测”>“检测环路”。

将出现“检测环路”框。它指明了在模式中检测到的一个或多个环路，并为每个环路建议了一个候选别名或上下文。

- 单击向前箭头按钮，直至为检测到的环路显示了以下消息：

此环路可用别名解决。



- 单击“插入别名”按钮。

别名即会自动插入“结构”窗格中。它联接到导致模式中出现环路的表。

5.5.5.7 使用“检测环路”创建上下文

使用“检测环路”创建上下文：

- 选择“工具”>“自动检测”>“检测环路”。

将出现“检测环路”框。它指明了在模式中检测到的一个或多个环路，并为每个环路建议了一个候选别名或上下文。

- 2 单击向前箭头按钮，直至为检测到的环路显示了以下消息：

此环路未被任何上下文涵盖。



- 3 单击“候选上下文”按钮。

将出现“候选上下文”对话框。



- 4 单击上下文名称。

候选上下文中包含的表即会在模式中突出显示。

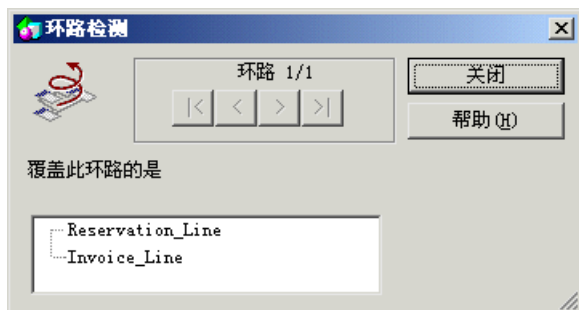
- 5 单击“添加”按钮。

上下文名称出现在“接受的上下文”窗格中。可以从右边的窗格中删除任何上下文，方法是将其选定，然后单击“移除”按钮。

- 6 如果适用，请重复步骤 3 和 4 以添加其他上下文。

- 7 单击“确定”。

将出现一个上下文确认框。



8 单击“关闭”。

这些上下文即会列在“Universe”窗口的“上下文”框中。

5.5.6 解决环路的示例

以下是一些说明如何执行以下操作的成功示例：

- 创建一个别名，以中断由于共享查找表引起的环路
- 创建一个别名，以中断由于共享查找表引起的环路
- 确定别名何时不适合于中断环路
- 创建上下文以解决环路
- 结合使用别名和上下文以解决环路

这些模式不基于 Beach Universe。它们使用一个基于货运公司的模式，并显示某个环路解决方法示例的另一个方面，本章中已使用 Beach Universe 显示了该示例。

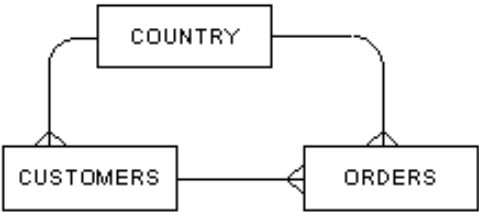
5.5.6.1 创建一个别名，以中断由于共享查找表引起的环路

销售数据库包含有关在全球范围内销售给客户的产品的信息。这些客户可能：

- 居住在世界上的任何地方
- 从公司订购产品
- 请求将这些产品发运到任何国家/地区的目的地

例如，居住在英国的客户可能会订购一辆汽车，并要求将其发运到巴西。

此类型的数据库的模式如下所示：



可以按如下方式解释此模式：

- 每个客户都来自一个国家/地区。
- 每个客户都可能对某项产品下达一个或多个订单。
- 公司将每项订购的产品发运到目的地国家/地区，该国家/地区不必与客户所居住的国家/地区相同。

这些表和它们的列如下所示：

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	WÄLDRON	3
103	HANCOCK	4
104	DAVIS	2
105	BÄRBIERI	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

可以运行查询来获取以下信息：

- 客户的名称
- 客户居住的国家/地区
- 每张订单的日期
- 发运的目的地国家/地区

用于抽取此数据的 SQL 如下所示：

```
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)
```

在执行时，此 SQL 将返回不完整的结果；仅返回了请求发运到居住地国家/地区的那些客户。未返回为发运选择另一个国家/地区的客户。

返回的行是客户居住地国家/地区和发运目的地国家/地区的交集。SQL 不会生成如下所示的完整结果

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

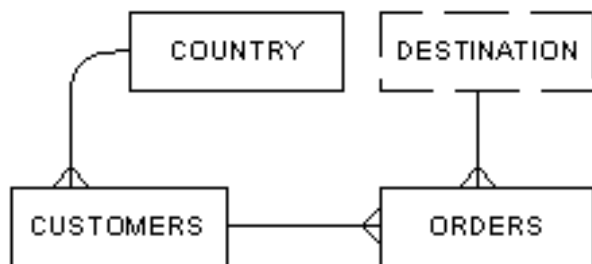
SQL 只会返回以下结果：

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

可以通过插入别名来中断环路。创建别名的第一步是确定在数据库结构中具有多个用途的查找表。下一节中描述了这一点。

5.5.6.2 确定多用途查找表

COUNTRY 表用于查找客户的居住地国家/地区和发运目的地国家/地区。此类型的表称为共享查找表。您在模式中创建了一个名为 DESTINATION 的别名。



原来的三个联接仍然存在，但环路已被 DESTINATION 别名中断，因此不再有闭合的联接路径。

5.5.6.3 在 FROM 子句中引用共享查找表和别名

现在，您需要在 From 子句中引用表名称两次，第一次使用正常名称，第二次使用别名；因此将使用备用名称为原始名称加上后缀。

生成的 SQL 如下所示：

```
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 (CUS
TOMER.CUST_LOC_ID=COUNTRY.COUNTRY_ID)
```

5.5.6.4 创建一个别名，以中断由于共享查找表引起的环路

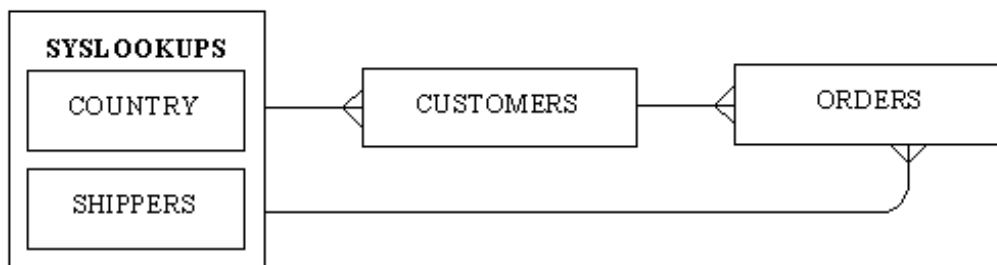
销售数据库包含有关居住在不同国家/地区的客户的信息。这些客户可能会订购可由多个承运商或货运公司运送的货物。

在此数据库中，已将国家/地区和发运人的名称标准化为查找表。标准化是一种通过删除冗余项来细化表关系的过程。

出于结构原因，将只会创建一个具有代码、说明和类型字段的查找表 (SYSLOOKUPS)，而不是两个查找表。类型字段指明记录所包含信息的特定类型，例如，国家/地区或发运人。

此类型的表也称为“灵活查找”，通常出现在 CASE 工具自动生成的模式中。

模式和表布局如下所示：



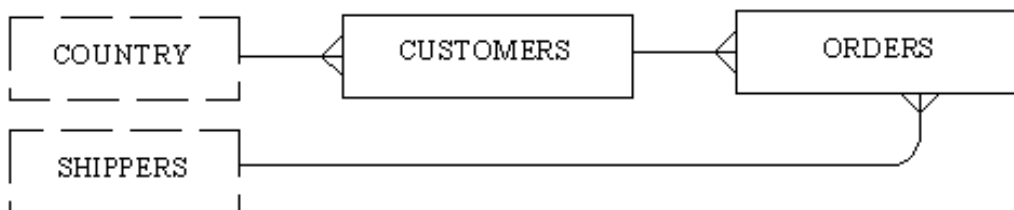
cust_id	last_name	loc_country
100	COLTRANE	1
101	MULLIGAN	1
102	WALDRON	3
103	HANCOCK	4
104	DAVIS	2
105	BARBIERI	5
106	STREETS	5

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

SYSLOOKUPS 表用于多个目的，因此您必须创建与表所拥有的域相同数量的别名（类型字段的不同值）。基于 SYSLOOKUPS 表中体现的两个目的，您可以创建两个别名：COUNTRY 和 SHIPPERS。

生成的模式如下所示：



在 Universe 设计工具中，用户可以创建对象 Customer's Country（定义为 COUNTRY.DESCRPTION）和对象 Shipper（定义为 SHIPPERS.DESCRPTION）。

对应的联接将为：

CUSTOMERS.LOC_COUNTRY=COUNTRY.CODE

ORDERS.SHIP_ID=SHIPPERS.CODE

使用自我限制的联接来限制结果

定义了对象之后，您现在需要限制每个别名以使其仅返回自己的域信息（而不是其他别名的域信息）。有关创建自我限制联接的更多信息，请参阅第 188 页上的[“自我限制的连接”](#)一节。

例如，如果要了解将两笔订单发运给客户 101 的发运人的名称，您将期望返回两个行。

但是，以下 SQL

```
SELECT ORDERS.ORDER_ID, ORDERS.CUST_ID, ORDERS.ORDER_DATE, SHIPPERS.DESCRPTION SHIPPER FROM ORDERS, SYSLOOKUPS
SHIPPERS WHERE (ORDERS.SHIP_ID=SHIPPERS.CODE)
```

将产生下面的结果：

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

查询返回了国家/地区和发运人的名称。“Man With a Van”和“USA”共享代码 1，而“France”和“Parcel Fun”共享代码 3。

可以按如下方式更正错误：

- 将一个新的自我限制的联接应用到 SHIPPERS 别名。在“编辑联接”对话框中，将表 1 和表 2 都设置为 SHIPPERS，并输入 SQL 表达式 SHIPPERS.TYPE='SHIP'。
- 将一个新的自我限制的联接应用到 COUNTRY 别名。在“编辑联接”对话框中，将“表 1”和“表 2”都设置为 COUNTRY，并输入 SQL 表达式 COUNTRY.TYPE='CTRY'。

使用限制时的问题

将限制添加到对象的 Where 子句或别名和 CUSTOMERS/ORDERS 表之间的现有联接时，将可能会产生以下问题：

- 在将限制添加到对象的 Where 子句时，必须同时将相同的限制添加到依据别名构建的每个对象。如果基于包含多个列的别名创建了若干对象，则在维护 Universe 时可能会出现问題。
- 只有在调用联接时，针对别名和另一个表之间的联接的限制才会生效。如果运行只包含 Shipper 对象的简单查询，则会返回 SHIPPERS 别名中的每一行（包括不需要的 Country 行），因为没有包括 ORDERS 表的理由。由于联接看起来并不必要，因此未应用限制。

小结

在此示例中，我们将模式与共享查找表一起考虑。采取的操作可以概括为以下几点：

- 1 为共享查找表创建 COUNTRY 和 SHIPPERS 别名。
- 2 为别名创建自我限制的联接作为限制。

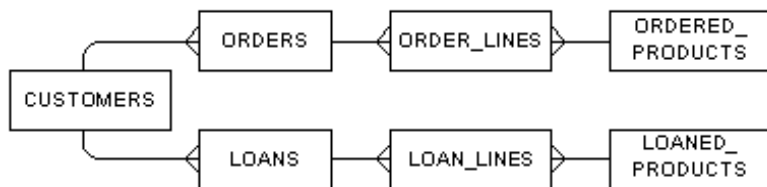
通过将一个合并的查找表用作两个不同的查找表，此示例中的别名解决了环路。这些别名还需要设置限制（自联接），因此在某些结构中，别名可能会导致额外需要进行调整 and 限制。

5.5.6.5 确定别名何时不适合于中断环路

上文所述的使用别名来解决环路并不是最佳的解决方法。在这种情况下，使用上下文是一种更好的解决方法。以下示例描述了别名为何不适当的原因，以及在这种情况下为何上下文是更好的解决方法。

如果尝试标识用作多个目的的查找表，则查找表是 PRODUCTS 表还是 CUSTOMERS 表将不会很清楚。

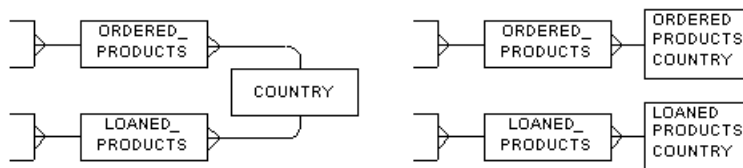
如果决定为 PRODUCTS 表创建如下所示的两个别名：



这两个别名为 ORDERED_PRODUCTS 和 LOANED_PRODUCTS。这可能会使用户产生混淆，因为他们更可能理解产品，而不是已订购产品或借贷产品。

如果还决定添加一个 COUNTRY 表来指明产品是在若干不同的国家/地区生产的，则必须将其直接联接到 PRODUCTS 表。

生成的模式将如下所示：



在上面的模式中，必须要创建两个别名：ORDERED_PRODUCTS_COUNTRY 和 LOANED_PRODUCTS_COUNTRY。对于这一特定的模式而言，使用别名明显是一种不能令人满意而且复杂的解决方法。

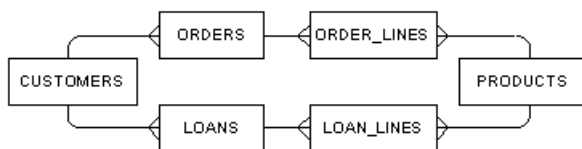
在这种情况下，您应创建上下文。

5.5.6.6 创建上下文以解决环路

数据库包含有关可能购买或租用产品的客户的信息。在此数据库中，使用了两种不同的方式来表示客户和产品之间的关系：

- 按客户已订购（或销售给客户）的产品。
- 按已租借给客户的產品。

数据库具有以下类型的模式：



如果要运行仅返回客户名称列表和产品列表的查询，我们将使用 ORDER 和 ORDER_LINES 表。结果将是每个客户所订购产品的列表。

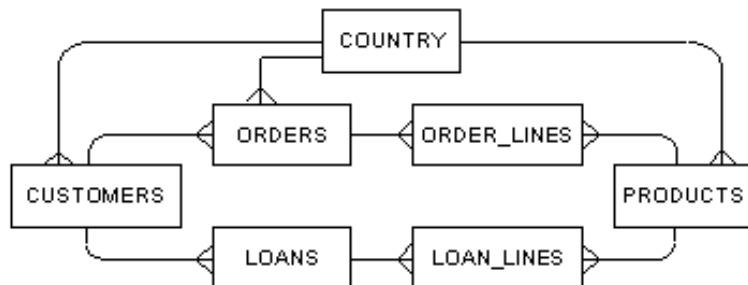
通过使用 LOANS 和 LOAN_LINES 表，我们将获得每个客户所租用产品的列表。

此模式包含一个环路，该环路将导致任何查询（同时包括全部六个联接）生成由销售给客户的产品和租借给客户的产品组成的列表。如果某个产品已售出，但从未租借给客户（反之亦然），它将不会出现在结果列表中。

结合使用别名和上下文以解决环路

可以使用上下文和别名来解决 Universe 中的环路。以下示例显示了如何在环路解决过程中结合使用别名和上下文。

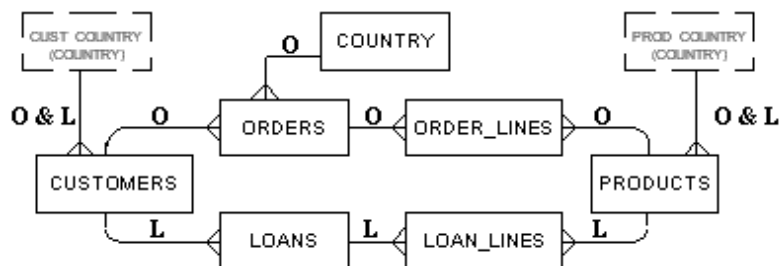
Universe 具有以下模式：



可以使用别名和上下文来解决环路，如下所示：

- 为 COUNTRY 表创建两个别名：CUST_COUNTRY 和 PROD_COUNTRY
- 定义两个上下文来解决 CUSTOMERS 到 PRODUCTS 环路（Orders 和 Loans）
- 确保 CUSTOMERS 和 CUST_COUNTRY 以及 PRODUCTS 和 PROD_COUNTRY 之间的两个联接同时出现在两个上下文中。

生成的模式如下所示：



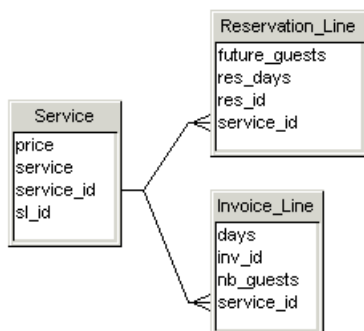
5.6 解决断层陷阱

断层陷阱是关系数据库模式中的一种常见问题，其中联接路径返回了比预计更多的数据。

5.6.1 什么是断层陷阱？

断层陷阱是当两个“多对一”联接会聚在一个表上时三个表之间的一种联接路径，并且没有适当地分隔会聚联接路径的上下文。

下面的示例显示了 Beach Universe 模式的一部分。已从模式的其余部分中分离出了这三个表，以便举例说明断层陷阱。它为数据使用相同的 Club 连接。Service 表接收两个一对多联接的一个末端。



只有当所有以下条件存在时，才会获得不正确的结果：

- Universe 结构中的三个表之间存在“多对一对多关系”。
- 查询包括基于两个表的对象，这两个表均位于其各自连接的“多”端。
- 单一维返回了多个行。

以下示例显示了在上述所有条件存在时运行的查询如何返回笛卡尔积。

示例：断层陷阱在不进行警告的情况下使结果膨胀

通过使用上面的模式，Web Intelligence 用户可运行以下独立的查询：

查询	返回的结果				
Service • Number of guests Service Equal to 'Sports'	<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				
Service • Number of future guests Service Equal to 'Sports'	<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				

用户现在运行既包括已付款宾客又包括将来宾客的查询：

Service • Number of guests • Number of future guests
 Service Equal to 'Sports'

返回了以下结果：

Service	Number of guests	Number of future guests
Sports	188.00	96.00

已使用过或已预订要使用体育服务的宾客数量已经大量增加。返回了笛卡尔积，因此结果不正确。如果未检测到，这可能是一个严重的问题。上面的示例可能会使群岛度假地经理认为：度假地的体育活动服务对宾客而言比实际数字所指明的更有吸引力。

5.6.2 断层陷阱如何使结果膨胀？

断层陷阱导致查询使用另一个度量所有可能的行组合返回一个度量所有可能的行组合。在上面的示例中发生了以下情况：

- Number of guests transactions * Number of future guest transactions (宾客数 * 将来宾客数)
- Number of future guest transactions * Number of guests transactions (将来宾客数 * 宾客数)

以下示例详细说明了断层陷阱如何返回笛卡尔积。

示例：检查断层陷阱的笛卡尔积

我们需要检查查询所返回的行以生成聚合数字。在我们的示例中，通过将“Days Billed（开票天数）”和“Days Reserved（预订天数）”维添加到查询以返回单独的交易明细，我们可以达到此目的。

宾客数报表看起来如下所示：

Service	Days billed	Number of guests
Sports	3.00	4.00
Sports	4.00	133.00
Sports	6.00	8.00

将来宾客数报表看起来如下所示：

Service	Days reserved	Number of future guests
Sports	1.00	7.00
Sports	2.00	1.00

这两个报表显示以下交易数：

- 宾客数 = 3 次交易
- 将来宾客数 = 2 次交易

当两个维都添加到查询中时，将返回以下结果：

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
	Sum:	188.00	Sum:	96.00

查询返回了每个可能的“Number of Guests（宾客数）”行组合以及每个可能的“Number of Future Guests（将来宾客数）”行组合：“Number of Guests（宾客数）”交易每个出现两次，而“Number of Future Guests（将来宾客数）”交易每个出现三次。

在对返回的数据进行总计时，总计的结果不正确。

与环路不同，Universe 设计工具不会自动检测到断层陷阱，但是，可以使用“检测上下文”（“工具”>“检测上下文”）在模式中自动检测和建议候选上下文。

“检测上下文”将检查模式中的多对一联接。它挑选出接收会聚性多对一联接的表，并建议用于分隔在表上运行的查询的上下文。这是确保模式不会包含断层陷阱的最有效方式。

通过分析模式中的一对多联接路径，您也可以使用图形方式检测断层陷阱。

如果既不运行“检测上下文”，也没有找出模式中的断层陷阱，则发现问题的唯一方式是查看明细行。否则您将不会收到该情况的警报。

5.6.3 检测断层陷阱

通过使用“检测上下文”来检测和建议候选上下文，并随后检查任何两个上下文分叉处的表，您可以找到断层陷阱。两个上下文相交的这一点是断层陷阱的来源。

如果有多对一联接会聚在一个查找表上的两个事实表，则可能存在断层陷阱。

提示：

有关组织表模式以检测联接问题的信息，请参阅第 263 页上的[“以图形方式检测联接问题”](#)。

5.6.4 解决断层陷阱

要解决断层陷阱，您需要运行两个单独的查询，然后合并结果。取决于为事实表定义的对象类型，以及最终用户环境的类型，可以使用以下方法来解决断层陷阱。

- 为每个事实表创建一个上下文。此解决方法适用于所有情况。
- 修改 Universe 的 SQL 参数，以便能够为每个度量生成单独的 SQL 查询。此解决方法仅适用于度量对象。它不会为维或明细对象生成单独的查询。

以下各节中论述了上述每种方法。

5.6.4.1 使用上下文来解决断层陷阱

可以为位于联接的“多”端的每个表定义一个上下文。在我们的示例中，您可以定义从 SERVICE 到 RESERVATION_LINE 以及从 SERVICE 到 INVOICE_LINE 的上下文。

如果查询包括来自两个上下文的对象，运行查询时将创建两个 Select 语句，并且同步在 Web Intelligence 中生成两个单独的表，从而避免了创建笛卡尔积。

5.6.4.2 何时使用上下文？

创建上下文总是可以解决 Universe 中的断层陷阱。如果一个或两个事实表中都有维对象，则应始终使用上下文。

5.6.4.3 使用上下文来解决断层陷阱

使用上下文来解决断层陷阱：

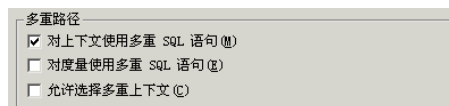
1. 通过分析模式中的“一对多对一”联接路径关系来确定可能的断层陷阱。

- 2 选择“工具”>“检测上下文”。

将出现“候选上下文”框。



- 3 在“候选上下文”列表框中选择一个建议的上下文，然后单击“添加”按钮将其添加到“接受的上下文”列表框。
- 4 为其他列出的上下文重复该步骤。
新的上下文即会列在“列表视图”栏的“上下文”窗格中。
- 5 选择“文件”>“参数”。
将出现“Universe 参数”对话框。
- 6 单击“SQL”选项卡。
将出现“SQL”页面。
- 7 选择“为每个上下文使用多个 SQL 语句”复选框。



- 8 单击“确定”。
当您在断层陷阱中的表上运行查询时，将为在受影响的表上定义的度和维分隔查询。

5.6.4.4 为每个度量使用多个 SQL 语句

如果只为两个事实表定义了度量对象，则可以使用“Universe 参数”选项“为每个度量使用多个 SQL 语句”。这样会强制为“查询”窗格中出现的每个度量生成独立的 SQL 查询。

此解决方法不适用于维和明细对象。

下表描述了何时可以使用“为每个度量使用多个 SQL 语句”以及何时应避免使用它：

是否使用多个 SQL 语句	场合
为每个度量使用多个 SQL 语句	在只包含为两个事实表同时定义的度量对象的 Universe 中。使用多个 SQL 语句的优点在于可以避免使用需要稍后维护的上下文。
不为每个度量使用多个 SQL 语句	<p>适用于为一个或两个事实表定义了维或明细对象的情况。如果基于使用此解决方法的 Universe 在查询中包括了维或明细对象，则将会返回笛卡尔积。</p> <p>由于此解决方法可能会减慢查询的响应时间并产生不正确的结果，您应考虑创建上下文来解决断层陷阱。</p>

激活“为每个度量使用多个 SQL 语句”：

- 1 从菜单栏中选择“文件”>“参数”。
将出现“Universe 参数”对话框。
- 2 单击“SQL”选项卡。
- 3 在“多路径”组框中选择“为每个度量使用多个 SQL 语句”复选框。
- 4 单击“确定”。

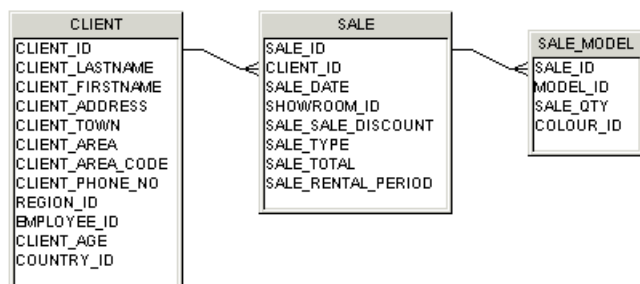
5.7 解决扇形陷阱

扇形陷阱是关系数据库模式中仅次于断层陷阱的常见问题。它的影响与返回比预计更多的数据相同。

5.7.1 什么是扇形陷阱？

当“一对多”连接链接了依次由另一个“一对多”连接链接的表时，扇形陷阱是三个表之间的一种连接路径。当查询包括同时基于两个表的对象时，“一对多”连接的分散影响可能会导致返回不正确的结果。

下面显示了扇形陷阱的一个简单示例：



当您为特定客户运行查询（该查询请求按每个模型生产线列出的汽车模型总数）时，将会返回不正确的结果，因为您是在位于连接的“一”端但仍然连接到“多”端的表上执行聚合函数。

示例：扇形陷阱在不进行警告的情况下使结果膨胀

通过使用上面的模式，Web Intelligence 用户运行了以下查询：

ClientName SaleValue SaleQTY
ClientName Equal to 'WendyCraig'

返回了以下结果：

ClientName	SaleQTY	SaleValue
WendyCraig	2.00	57,092.00

此结果是正确的。但是，最终用户将维 Model ID 添加到查询，如下所示：

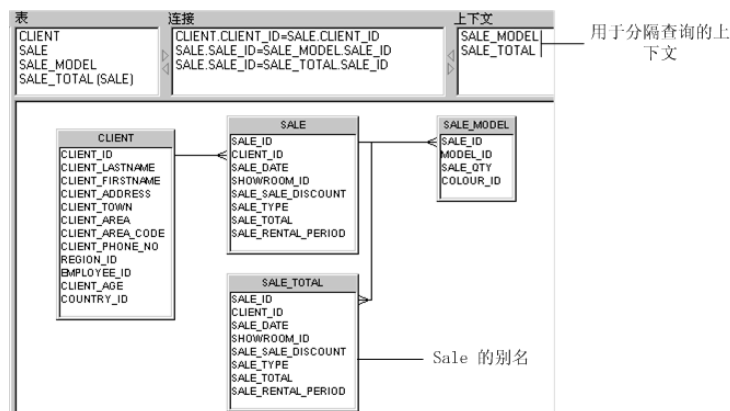
ClientName SaleValue Model Id SaleQTY
ClientName Equal to 'WendyCraig'

以下报表是使用返回的结果创建的：

WendyCraig		
Model Id	SaleValue	SaleQTY
1,034.00	57,092.00	1.00
1,081.00	57,092.00	1.00
Sum:	114,184.00	2.00

Sale Value 聚合出现了两次。Model_ID 的每个实例一次。在报表中聚合这些结果时，总和不正确。扇形陷阱返回了笛卡尔积。Wendy 购买了两辆汽车，总值为 57,092.00 美元，而不是报表中总计出的 114,184.00。由于在查询中包括了 Model_ID，因此导致按与 Model_ID 相同的行数对 SaleValue 进行了聚合。

查询中使用维对象的扇形陷阱是通过使用别名和上下文解决的。以下模式是扇形陷阱模式的解决方法：



在使用上面的解决方法运行时，为 Wendy Craig 返回笛卡尔积的原始查询现在返回了以下表：

WendyCraig

Sale Qty	Model Id	Sale Total
1.00	1,034.00	57,092.00
1.00	1,081.00	

5.7.2 如何检测扇形陷阱？

无法自动检测扇形陷阱。您需要以直观方式检查表模式中所显示基数的方向。

如果有两个由度量对象引用并且用一系列多对一联接加以联接的表，则可能存在扇形陷阱。

有关排列表模式以检测联接问题的说明，请参阅第 263 页上的 [“以图形方式检测联接问题”](#) 一节。

5.7.3 如何解决扇形陷阱？

可以通过两种方式来解决扇形陷阱问题。

- 为包含初始聚合的表创建别名，然后使用“检测上下文”（“工具”>“检测上下文”）来检测和建议用于别名表的上下文和用于原始表的上下文。这是解决扇形陷阱问题的最有效方式。
- 更改 Universe 的 SQL 参数。此方式仅适用于度量对象。

下面描述了这两种方法。

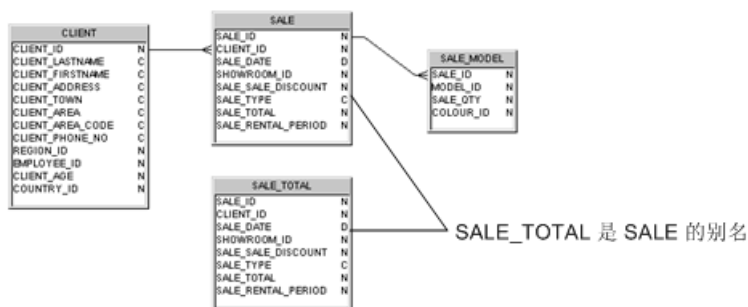
5.7.3.1 使用别名和上下文来解决扇形陷阱

您为产生聚合的表创建一个别名表，然后检测和实施用于分隔查询的上下文。可以按如下方式执行此操作：

使用别名和上下文来解决扇形陷阱：

- 1 通过分析模式中的“一对多对一对多”联接路径关系来确定可能的扇形陷阱。
- 2 为产生增殖性聚合的表创建一个别名。

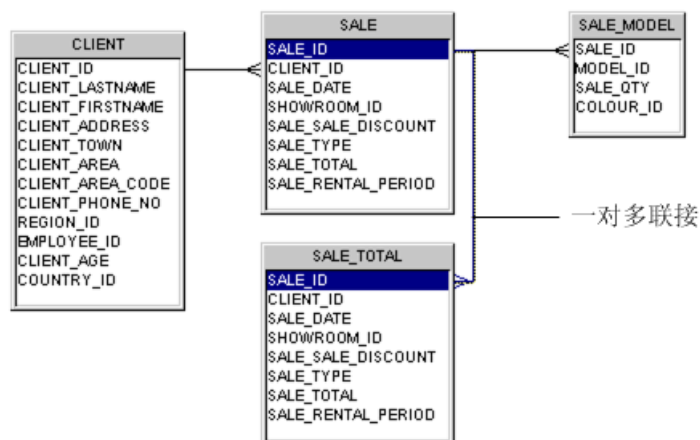
例如，上一个示例中的 SaleValue 是 Sales 表中 Sale_Total 列的聚合。您为 Sale 创建了一个名为 Sale_Total 的别名。



- 3 在原始表和别名表之间创建一个联接。

如果创建一对一联接，Universe 设计工具将不会检测上下文，因此必须手动构建上下文。大多数情况下，可以使用允许自动检测和实施上下文的一对多连接。

例如，您在 Sale 和 Sale_Total 之间创建了一对多联接。



- 4 构建在别名表上进行聚合的对象。

例如，按如下方式定义了原始 SaleValue 对象：

sum(SALE.SALE_TOTAL)。SaleValue 的新定义为：

sum(Sale_Total.SALE_TOTAL)。

- 5 选择“工具”>“检测上下文”。

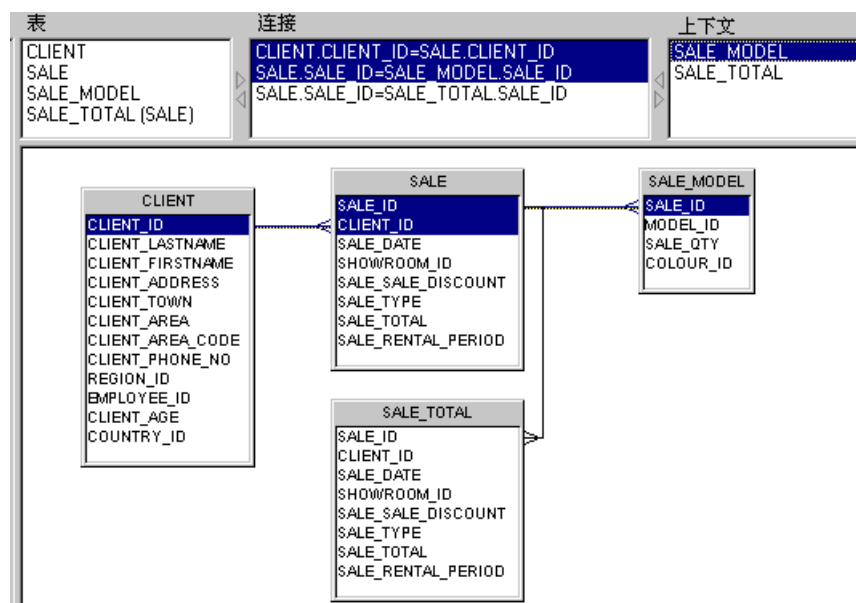
将出现“候选上下文”框。它为基表的联接路径和别名表的新联接路径建议了候选上下文。

注意：

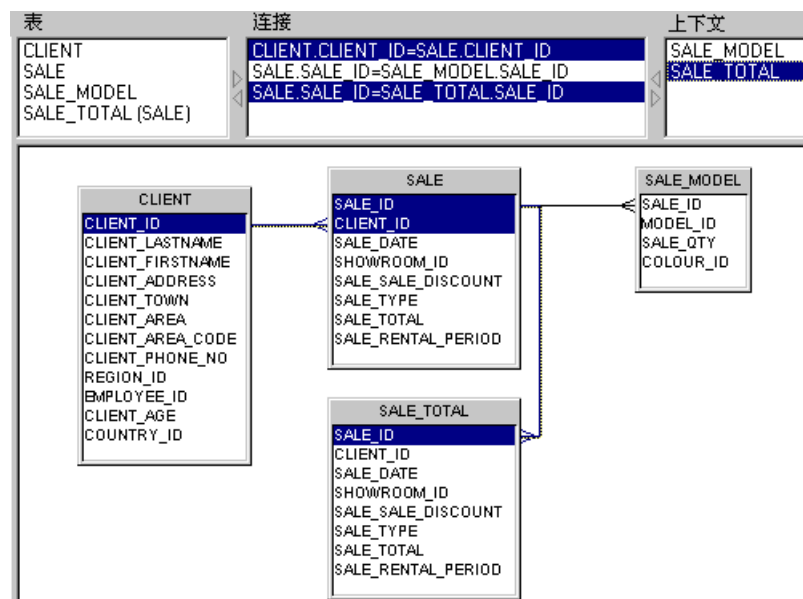
如果在别名和基表之间使用了一对一联接，则需要手动创建上下文。

- 6 单击一个候选上下文，然后单击“添加”。
- 7 为其他候选上下文重复该步骤。
- 8 单击“确定”。

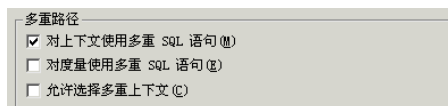
既会在模式中创建上下文。当“列表模式”处于活动状态时（“查看”>“列表模式”），您可以在“上下文”窗格中查看这些上下文。联接路径 CLIENT>SALE>SALE_MODEL 的上下文看起来如下所示：



CLIENT>SALE>SALE_TOTAL 联接路径的另一个上下文如下:



9. 选择“文件”>“参数”。
- 将出现“Universe 参数”对话框。
10. 单击“SQL”选项卡。
- 将出现“SQL”页面。
11. 选择“为每个上下文使用多个 SQL 语句”复选框。



12. 单击“确定”。
13. 运行查询以测试扇形陷阱解决方法。

5.7.3.2 为每个度量使用多个 SQL 语句

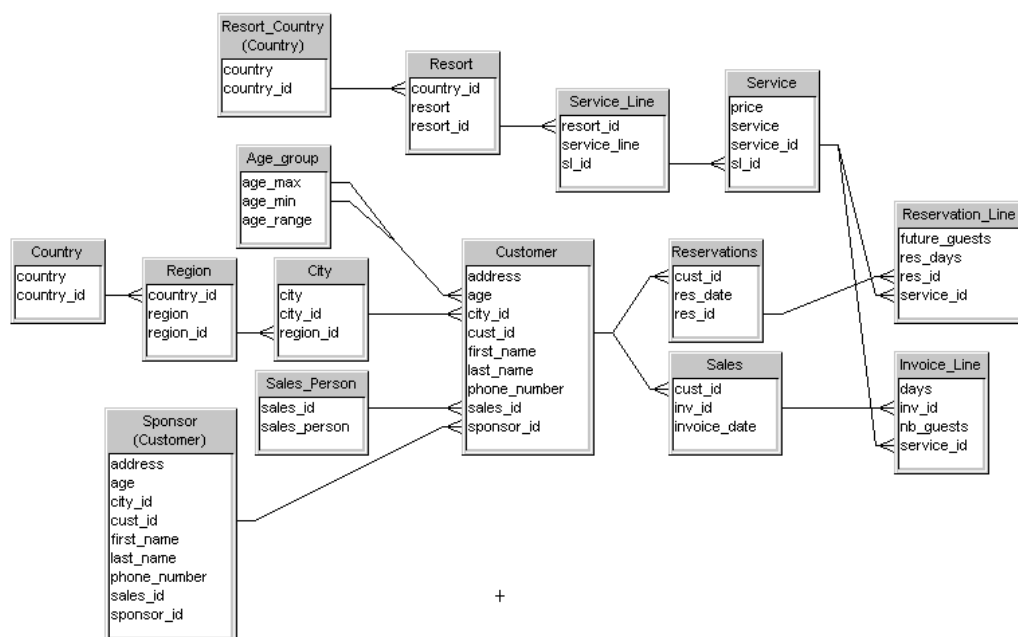
如果只为位于串联一对多联接的“多”端的表定义了度量对象，则可以使用“Universe 参数”选项“为每个度量使用多个 SQL 语句”。这样会强制为“查询”窗格中出现的每个度量生成独立的 SQL 查询。

不能使用此方法为维生成多个查询。如果最终用户能够纳入在查询中引用度量对象的任何表中的维，则必须使用别名和上下文来解决扇形陷阱。

有关激活此选项的更多信息和过程，请参阅第 263 页上的[“为每个度量使用多个 SQL 语句”](#)一节。

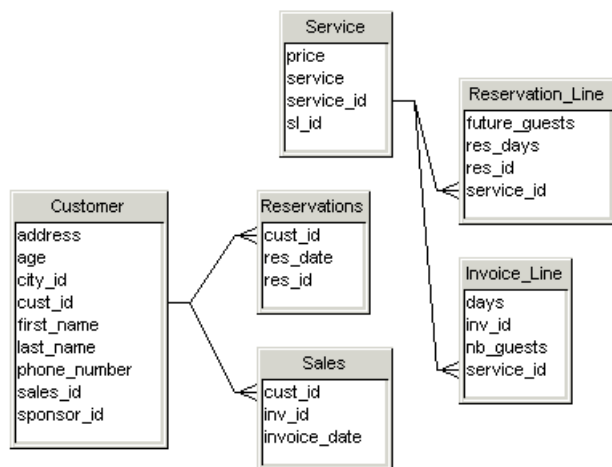
5.8 以图形方式检测联接问题

您可以以直观方式检测表模式中的潜在断层陷阱和扇形陷阱，方法是在“结构”窗格中排列各个表，以使联接的“多”端位于窗格的一侧，而“一”端位于另一侧。下面的示例显示了按一对多连接从左流向右的方式进行排列的 Beach Universe 模式。



5.8.1 可能的断层陷阱

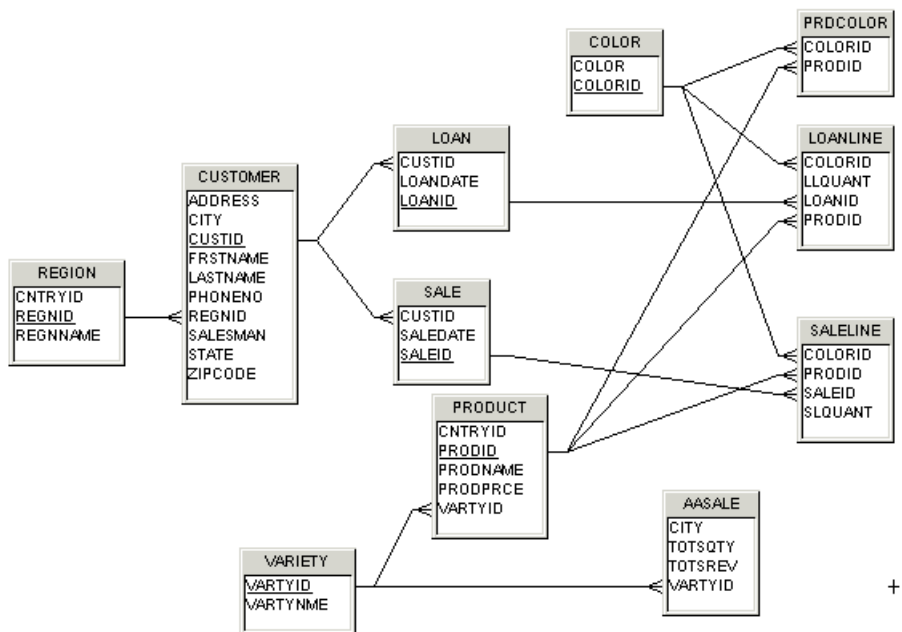
可能的断层陷阱如下所示：



这两个联接路径都已使用上下文 Sales 和 Reservations 分隔。

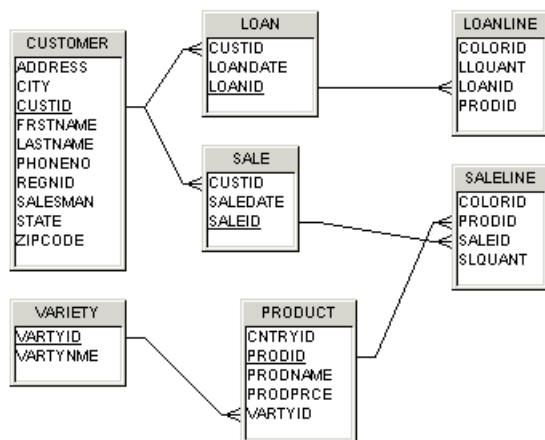
5.8.2 可能的扇形陷阱

汽车销售数据库的 Universe 模式如下所示：



可能的扇形陷阱涉及到以下各个表

- CUSTOMER、LOAN 和 LOANLINE
- CUSTOMER、SALES 和 SALELINE
- VARIETY、PRODUCT 和 SALELINE

**提示：**

使用必要的表填充了模式之后，请不要立即启动定义对象。留出一些时间来移动表，以便将所有一对多联接设置为方向相同。Universe 设计工具是一种图形工具，因此请使用此产品的可视功能来帮助设计 Universe。花一个多小时来移动表可以在以后的设计过程中节省大量的时间。

5.9 检查 Universe

在设计 Universe 时，应定期测试其完整性。可以按如下方式验证 Universe 的完整性：

检查 Universe	说明
自动	可以设置 Universe 设计工具选项，以便在创建、导出 Universe 或在打开 Universe 时检查 Universe 结构的 SQL 语法。
手动	可以运行“检查完整性”以检查选定 Universe 结构。

5.9.1 自动检查 Universe 完整性

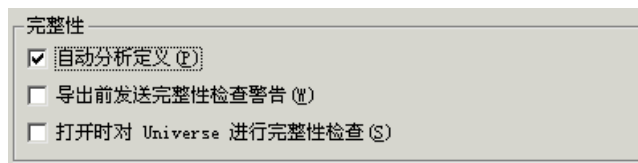
可以在 Universe 设计工具中设置以下完整性检查选项，以便在创建、导出 Universe 或在打开 Universe 时分析 SQL 结构：

自动检查选项	说明
依据定义的自动解析	Universe 设计工具将在创建时自动检查所有对象、条件和联接的 SQL 定义。它将在您单击“确定”以验证结构创建时应用。
发送检查完整性	当每次尝试导出未经检查的 Universe 时，Universe 设计工具都会显示一条警告。
打开时检查 Universe 完整性	所有 Universe 在打开时均会自动受到检查。

5.9.1.1 设置自动 Universe 检查选项

设置自动 Universe 检查选项：

- 1 选择“工具”>“选项”。
- “选项”对话框即会打开到“常规”页面。
- 2 在“完整性”组框中，选择或清除相应 Universe 自动检查选项的复选框。



- 3 单击“确定”。

5.9.2 手动检查 Universe 完整性

可以使用“检查完整性”来测试活动 Universe 的设计是否准确并保持最新。

“检查完整性”将检测以下各项：

- Universe 的对象、连接、条件及基数中的错误。
- 联接路径中的环路。
- 任何必需的上下文。
- 对目标数据库所做的更改。

在依据这些数据库检查 Universe 的各个元素之前，该功能将检查与数据库的连接是否有效。如果连接无效，则功能将停止并返回错误消息。

5.9.2.1 “检查完整性”检测的错误类型

“检查完整性”可以检测：

- 对象、条件或联接的 SQL 定义中的无效语法。
- 环路
- 孤立的表
- 孤立联接
- 上下文内的环路
- 缺少或不正确的基数

5.9.2.2 “检查完整性”如何确定连接数据库中的更改？

“检查完整性”功能会向数据库发送一个请求，以获得表的列表。然后，它会将此列表与 Universe 中的表进行比较。它将对列执行同样的操作。

在“结构”窗格中，“检查完整性”会将与列表中的表或列不匹配的任何表或列标记为不可用。这些表或列在数据库中可能已被删除或重命名。请参阅第 271 页上的 [“刷新 Universe 结构”](#) 一节。

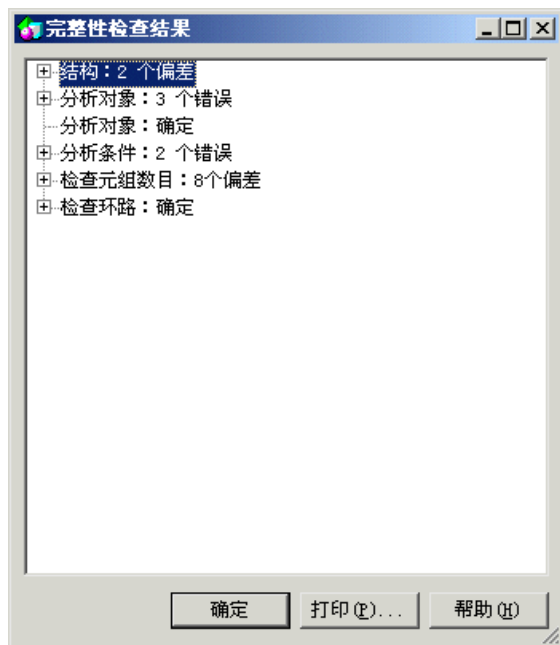
注意：

如果检查大量的数据，则“检查基数”选项可能运行很慢。如果有数据不明确或缺少数据，结果也可能不准确。如果数据库较大并且可能包含不完整的数据条目，则不应选择“检查基数”选项。如果确实要使用此选项，则可以通过修改 PRM 文件来优化基数检测。有关更多信息，请参阅第 198 页上的[“优化自动基数检测”](#)一节。

5.9.2.3 使用“检查完整性”验证 Universe 的完整性

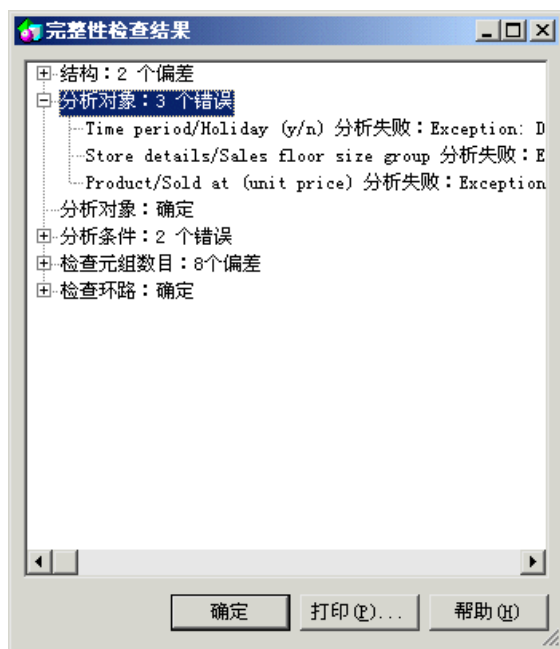
验证 Universe 完整性：

- 1 选择“工具”>“检查完整性”。
 - 或
单击“检查完整性”按钮。
 - 将出现“完整性检查”对话框。
 - 2 选择要验证的元素的复选框。
 - 3 清除不要验证的元素的复选框。
 - 4 选择“快速分析”复选框，以便只验证元素的语法。
 - 或
选择“完全分析”复选框，以便同时验证元素的语法和语义。
 - 5 单击“确定”。
- 消息框将显示 Universe 检查进度。



如果“检查完整性”未遇到错误，则会在每种错误类型旁边显示“确定”。

- 6 单击错误类型旁边的加号 (+)，查看发生错误的元素的列表。



可以双击列表中的项目，以便突出显示“结构”窗格中对应的元素。

- 7 单击“打印”按钮打印窗口内容。
- 8 单击“确定”。

注意：

在选择“检查环路”复选框之前，请确保已检测了联接的基数。否则，该功能将错误地标识联接中的环路。

5.9.3 刷新 Universe 结构

如果“检查完整性”指出 Universe 连接的数据库已被修改，则可以使用“刷新结构”来更新“结构”窗格的内容。

“刷新结构”可以修改 Universe 结构以符合数据库中的更改，如下所示：

如果	则工具执行以下操作
向表中添加了列	将列添加到 Universe 中对应的表。
从表中删除了列	显示一则警告消息，指明应删除的列和关联联接。
从数据库中删除了表	显示一则警告消息，指明应删除的表和关联联接。
在数据库中重命名了表	显示一则消息，说明它不再识别 Universe 中对应的表。应重命名这些表，以便与数据库中的表匹配。如果名称仍然不匹配，Universe 设计工具将返回一条消息，指明重命名的表在数据库中不存在。
未对数据库进行更改	显示一则消息，通知您无需进行更新。

5.9.3.1 刷新 Universe

刷新 Universe 结构：

- 选择“查看”>“刷新结构”。

将出现一个消息框，通知您数据库中的更改，或通知您无需更新（如果未进行更改）。

第 6 章 创建 Universe

创建模式、执行完整性检查并解决了环路问题之后，即可创建将被报表工具使用的 Universe。

6.1 概述

本章描述如何创建 Web Intelligence 用户用于运行查询和创建报表的类和对象。它还论述 Universe 优化，以及优化对象定义以提升最终用户报表的性能。

前面各章已经描述了如何规划 Universe、创建包含 Universe 数据库结构（表、列和联接）的表模式以及如何解决联接路径中的环路。

Web Intelligence 用户看不到您创建的模式。完成此数据库结构之后，即可以建立类和对象，用户将在“Universe”窗格中看到这些类和对象，并且将使用它们依据数据库结构运行查询，以便生成文档和报表。

6.2 Universe 构建简介

构建 Universe 是 Universe 开发周期的对象创建阶段。您创建的对象必须基于用户需求调查，并且必须使用已针对联接路径问题测试过的可靠模式设计。

下表指明了构建（和测试）阶段出现在一个典型 Universe 开发周期（实现，步骤 2）的哪个位置：

- 准备
 - 。 1 用户需求分析
 - 。 2 规划
- 实现
 - 。 1 设计和测试模式
 - 。 2 构建和测试 Universe 对象
 - 。 3 使用资源库部署 Universe

- 维护
 - 1 根据用户需求或数据源中的更改更新和维护 Universe

6.2.1 什么是对象？

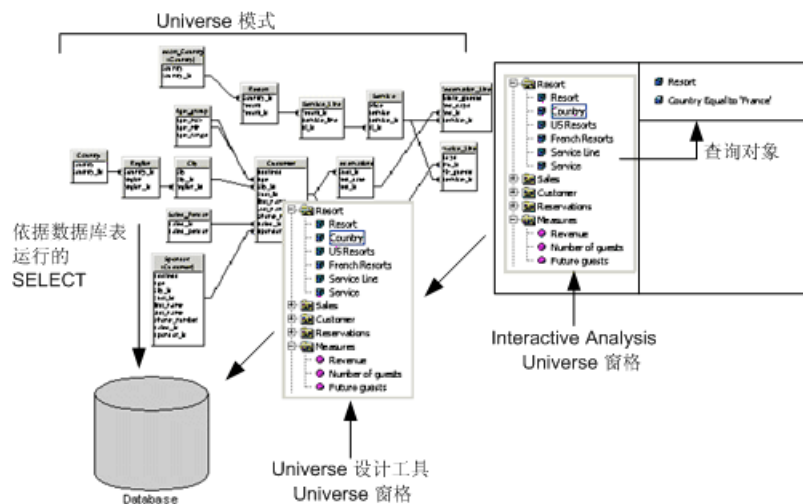
在 Business Objects 产品中，对象是 Universe 中一种命名元素，它代表数据库中的一个列或函数。

对象以图标形式显示在“Universe”窗格中。每个对象都代表最终用户业务环境中使用的一个有意义的实体、事实或计算。在 Universe 设计工具的“Universe”窗格中创建的对象就是最终用户在报表工具中看到和使用的对象。也可以创建仅在 Universe 设计工具中使用的对象，在 Web Intelligence 用户可以看到“Universe”窗格中可以隐藏这些对象。

Web Intelligence 用户将对象从“Universe”窗格拖到“查询”窗格即可运行查询，并使用返回的数据创建报表。

每个对象都映射到目标数据库中的一个列或函数，并且，当在“查询”窗格中使用时，每个对象都会推导出一个 SELECT 语句。当组合了多个对象时，SELECT 语句依据包括 SQL（由每个对象推导出）并应用默认 WHERE 子句的数据库运行。











下图显示了 Web Intelligence 的“Universe”窗格中的对象，以及 Universe 设计工具的“Universe”窗格中的相同对象。Universe 设计工具“Universe”窗格中的每个对象都映射为 Universe 模式中的一个列，并在查询中使用推导出一个 SELECT 语句。



Universe 设计员使用 Universe 设计工具创建对象，Web Intelligence 用户可以在“查询”窗格中包括这些对象以运行查询。

6.2.2 Universe 中使用什么类型的对象？

可以将对象限定为以下三种类型之一：

对象限定	示例	说明
维	 Resort  Country  Service Line	查询中分析的焦点。维将映射到数据库中的一个或多个列或函数，这些列或函数是查询的关键所在。
明细	 Customer  Age  Phone Number  Address	提供有关维的描述性数据。明细始终会附加到维。它映射到数据库中的一个或多个列或函数，这些列或函数提供了与维相关的详细信息。
度量	 Revenue  Number of guests  Future guests	包含映射为数据库中的统计数据的聚合函数。

在创建对象时，可以基于您希望对象在查询中具有的角色为其分配一个限定。此角色确定了在“查询”窗格中使用对象时推导出的 SELECT 语句。

6.2.3 什么是类？

类是对象的容器。类相当于 Windows 环境中的文件夹。可以创建类来容纳在 Universe 中具有共同用途的对象。

6.2.4 使用类和对象

可以在“Universe”窗格中将类和对象排列在一起，以便符合 Web Intelligence 用户使用对象所代表信息的惯常工作方式。

6.3 使用“Universe”窗格

可以使用“Universe”窗格创建 Universe 中的类和对象。

“Universe”窗格在活动的 Universe 中呈现类和对象的分层视图。使用“Universe”窗格来查看、创建、编辑并组织类和对象。

“Universe”窗格如下所示。类名出现在文件夹图标旁边，而对象名称出现在限定符号的旁边。



6.3.1 显示类和对象或条件

可以使用窗口底部的两个单选按钮来显示类和对象，或显示“Universe”窗格中的条件对象。条件对象是预定义的 Where 子句，可在一个或多个 Select 语句内使用。

可以显示“Universe”窗格的两个视图：

视图	要显示该视图...	显示内容
类/对象	选择左边的单选按钮	所有类和对象
类/条件	选择右边的单选按钮	所有类以及应用于每个类中所包含对象的条件

相关主题

- 第 309 页上的“[为对象定义限制](#)”

6.4 有关类、对象和条件的基本操作

可以在“Universe”窗格中执行以下操作，这些操作是类、对象和条件所共用的：

6.4.1 剪切、复制、粘贴

可以使用 Windows 环境中使用的常用标准命令剪切、复制和粘贴选定的元素。

6.4.2 移动类、对象或条件

可以将元素移动到窗口中的另一个位置，方法是将其拖放到所需的位置。

6.4.3 显示或隐藏类、对象和条件

可以在“Universe”窗格中隐藏一个或多个元素。这些元素对 Web Intelligence 用户隐藏，但是在 Universe 设计工具中保持可见。

如果出于以下任何原因，对最终用户隐藏对象可能十分有用：

- 元素来自链接的 Universe，并且在活动的 Universe 中不需要该元素。
- 对象仅用于优化 SQL 语法，最终用户不应看到对象。
- 您正在开发不想要最终用户通过“查询”窗格查看的元素。
- 想要在不删除元素的情况下暂时禁用元素。

6.4.3.1 隐藏类、对象或条件

隐藏类、对象或条件：

- 1 在“Universe”窗格中单击元素。
- 2 选择“编辑” > “隐藏项目”。

或

单击“编辑”工具栏上的“显示/隐藏”按钮。

元素名称在“Universe”窗格中显示为斜体。

6.4.3.2 显示隐藏的类、对象或条件

隐藏元素的名称显示为斜体。

显示隐藏的类、对象或条件：

- 1 在“Universe”窗格中单击隐藏的元素。
- 2 选择“编辑” > “显示项目”。

元素的名称不再显示为斜体。最终用户现在可看到该元素。

6.5 定义类

类是一个或多个对象的容器。Universe 中的每个对象都必须包含在类中。可以使用类将相关的对象分组。通过使用类，最终用户可以更轻松地查找特定对象。可以创建新类，也可以编辑现有类的属性。类以文件夹的形式呈现在“Universe”窗格的树形层次中。

提示：

使用类的一种有效方法是将相关维和明细对象分组为一个类，并将度量对象放在单独的类中。通过使用子类将对象分解为子集，可以对相关对象的分组进一步加以组织。第 282 页上的 [“使用子类”](#)一节中对子类进行了说明。

6.5.1 创建类

可以使用两种方式在“Universe”窗格中创建类：

- 手动定义类。
- 通过将表从表模式拖到“Universe”窗格中自动创建类

下面描述了这两种方法：

6.5.1.1 手动创建类

可以在“Universe”窗格内手动创建类。如果分析了用户需求，并且列出了可能的对象并将它们分组为类，则依据列表手动创建类是确保 Universe 结构符合最终用户需求的最佳方式。

在空“Universe”窗格中创建类：

- 1 选择“插入”>“类”。
- 或
- 单击“插入类”按钮。
- 将出现一个类属性框。
- 2 在“类名”文本框中键入名称。
- 3 在“描述”文本框中键入类的说明。
- 4 单击“确定”。

新命名的类文件夹即出现在“Universe”窗格中。

提示：

如果单击“应用”而不是“确定”，则会应用类的名称和说明，但属性框将保持打开。如果创建另一个类，您可以在相同的框中键入新类的属性。这样，您将能够使用一个属性框创建一系列类。由于避免了在创建每个类时都显示一个新的属性框，因此可以节省时间和避免不必要的单击。

6.5.1.2 使用现有类在“Universe”窗格中创建类

使用现有类创建类：

- 1 单击要在树形视图中放在新类之前的类，然后选择“插入”>“类”。
- 或
- 单击要在树形视图中放在新类之前的类，然后单击“插入类”按钮。
- 将出现一个类属性框。
- 2 键入新类的名称和说明。
- 3 单击“确定”。

新命名的类文件夹即出现在“Universe”窗格中。

6.5.1.3 依据表模式自动创建类

通过在表模式中选择一个表并将其拖入“Universe”窗格，您可以自动创建类。默认情况下，表名称就是类名。还会在该类的下方自动创建新的对象。每个新对象都对应于表中的一个列。

您应对新类和对象的属性进行编辑，以确保它们已正确命名，并且与最终用户的需求相关。第 283 页上的 [“定义对象”](#) 一节中描述了编辑对象属性。

在“Universe 参数”对话框的“策略”页上选定的对象策略确定如何自动创建类和对象（“文件”>“参数”>“策略”选项卡）。可以修改此策略。也可以创建策略以定制类和对象创建过程。有关策略的更多信息，请参阅第 410 页上的 [“使用外部策略定制 Universe 创建”](#) 和第 94 页上的 [“选择策略”](#) 两节。

注意：

在自动创建类和对象时，即会根据数据库结构直接创建 Universe 元素。应该在分析用户需求的基础上创建类和对象，而不是直接通过数据库中提供的列和表创建类和对象。第 23 页上的 [“Universe 设计方法”](#) 一节中描述了依据用户需求设计 Universe 的过程。

依据表模式自动创建类

- 1 在表模式中选择一个表。
- 2 将该表拖放到“Universe”窗格的类层次中的所需位置。

新的类即会出现在层次中。对于拖入“Universe”窗格的表中的每个列，它都包含一个对象。默认情况下，类名与表名称相同，而每个对象名称与其对应的列名称相同。

6.5.2 类属性

可以为类定义以下属性：

属性	说明
名称	可以包含特殊字符。在 Universe 中必须唯一。类名区分大小写。可以随时重命名类。
说明	描述类的说明。用户可在“查询”窗格中查看此说明。应采用用户的业务语言表述此字段中的信息，并且该信息应与用户的查询需求相关。按 CTRL + Return 键创建换行符。

6.5.3 修改类

可以从类属性对话框中随时修改类的名称和说明。可以通过以下任意方法访问类属性对话框：

- 双击某个类文件夹。
- 右键单击某个类文件夹，然后选择“编辑”>“类属性”。
- 单击某个类文件夹，然后选择“编辑”>“类属性”。

注意：

通过在类文件夹或类名称上执行以上任何单击操作，您可以访问类属性对话框。

6.5.4 使用子类

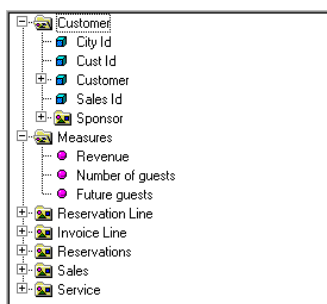
子类是类中的类。可以使用子类来帮助组织相关的对象组。子类本身可以包含其他子类或对象。

6.5.4.1 创建子类

创建子类：

- 单击某个类文件夹或类名称，然后选择“插入”>“子类”。
- 右键单击某个类文件夹或名称，然后从上下文菜单中选择“插入子类”。

下面的“Universe”窗格显示了“客户”类下方列出的子类“赞助商”。



6.6 定义对象

对象是一种 Universe 元素，它映射到 Universe 数据库模式中一个或多个表中的一个或多个列。对象也可以映射到一个或多个列上定义的函数。

每个对象都会为它所映射到的列或函数推导出一个 Select 语句。当 Web Intelligence 用户使用“查询”窗格中的一个或多个对象构建查询时，则使用每个对象所代表的列或函数推导出 Select 语句中 Select 子句行的内容。

6.6.1 创建对象

在“Universe”窗格中创建对象。Web Intelligence 用户通过名称和限定标识对象。可以在“Universe”窗格中手动创建对象，也可以通过将相应的数据库结构从“结构”窗格拖到“Universe”窗格来自动创建对象。

6.6.1.1 手动创建对象

通过在“Universe”窗格中插入对象然后定义对象属性，您可以手动创建对象。对象必须从属于类。

手动创建对象

- 1 右键单击“Universe”窗格中的某个类，然后从上下文菜单中选择“插入对象”。
或
单击某个类，然后单击“插入对象”工具。
对象即会插入到选定类的下方，并出现该对象的“编辑属性”框。
- 2 在“名称”框中键入名称。
确保始终用最终用户的业务词汇表述对象名称。此名称可能与在数据库模式中与该对象关联的实际列名称不同。
- 3 单击“属性”选项卡并选择对象属性。
- 4 在“选择”框中键入一个 Select 语句，或者单击“选择”按钮以便使用 SQL 编辑器。
- 5 单击“确定”。

相关主题

- 第 299 页上的[“使用 SQL 编辑器”](#)
- 第 285 页上的[“对象属性”](#)

6.6.1.2 自动创建对象

通过在“结构”窗格的表中选择一个列并将其拖到“Universe”窗口，您可以自动创建对象。将在最靠近放置列的位置的类下方创建对象。对象的默认名称是列名称。将使用空格替换所有下划线。默认对象数据类型是从列数据类型中派生的。通过从对象“编辑属性”表单的下拉列表框中选择一个新数据类型，您可以更改此值。

您应编辑新对象的属性，以确保其已正确命名，并且与最终用户的需求相关。第 283 页上的[“定义对象”](#)一节中描述了编辑对象属性。

在“Universe 参数”对话框的“策略”页上选定的对象策略确定如何自动创建类和对象（“文件”>“参数”>“策略”选项卡）。可以修改此策略。也可以创建策略以定制类和对象创建过程。

有关使用策略的更多信息，请参阅第 410 页上的[“使用外部策略定制 Universe 创建”](#)和第 94 页上的[“选择策略”](#)。

注意：

在自动创建类和对象时，即会根据数据库结构直接创建 Universe 元素。应该在分析用户需求的基础上创建类和对象，而不是直接通过数据库中提供的列和表创建类和对象。第 23 页上的[“Universe 设计方法”](#)一节中描述了依据用户需求设计 Universe 的过程。

自动创建对象：

- 1 在“结构”窗格中单击某个表列。
- 2 将该列拖过“Universe”窗格，并将表放到类层次中的所需位置。必须将该列放在现有类的下方。
新的对象即会出现在层次中。默认情况下，对象名称与列名称相同。

应确保始终用最终用户的业务词汇表述对象名称。此名称可能与在数据库模式中对象关联的实际列名称不同。

6.6.2 对象属性

可以通过“编辑属性”对话框为选定对象定义以下对象属性：

“编辑属性”页面	属性
<p>定义</p> <p>有关可用对象定义属性的完整信息，请参阅第 286 页上的 “对象定义”。</p>	<ul style="list-style-type: none">• 名称• 数据类型• 说明• Select 语句• Where 子句 <p>可以从页面中访问 SQL 编辑器以便定义 SELECT 和 WHERE 语法。</p>
<p>属性</p> <p>有关可用对象属性的完整信息，请参阅第 289 页上的 “属性”。</p>	<ul style="list-style-type: none">• 对象限定• 关联值列表
<p>高级</p> <p>有关可用高级对象属性的完整信息，请参阅第 290 页上的 “高级”。</p>	<ul style="list-style-type: none">• 安全性• 用户的对象权限• 日期格式
<p>键</p> <p>有关为对象定义索引感知的信息，请参阅第 292 页上的 “定义索引感知”。</p>	<ul style="list-style-type: none">• 键类型• Select• Where• 启用

“编辑属性” 页面	属性
<p>源信息</p> <p>有关使用此选项卡的信息，请参阅第 296 页上的 “源信息”。</p>	<ul style="list-style-type: none"> • 技术信息 • 映射 • 沿袭

可以随时修改对象属性。第 286 页上的 [“修改对象”](#) 一节中为每个“编辑属性”页面详细论述了上面列出的每个对象属性。

6.6.3 修改对象

您可以在创建对象时定义对象属性，也可以随时修改对象属性。通过“编辑属性”对话框为对象定义对象属性（右键单击对象 > “对象属性”）。可在“编辑属性”对话框的每个页面上定义的属性如下所述：

6.6.4 对象定义

“定义” 页面如下所示：



The screenshot shows the 'Definition' page of the 'Edit Properties' dialog box. It includes the following fields and controls:

- 名称(N):** A text box containing the value 'Country'.
- 类型(T):** A dropdown menu showing '字符' (Character).
- 描述(D):** A text box containing the value 'Resort's country'.
- 选择(S):** A text box containing the value 'Resort_Country.country', with navigation buttons (left arrow, right arrow, and a double right arrow button labeled '>>').
- 其中(W):** An empty text box, with navigation buttons (left arrow, right arrow, and a double right arrow button labeled '>>').
- Buttons:** At the bottom, there are two buttons: '表(B)...' (Table...) and '分析(P)' (Analysis).

可以通过“编辑属性”对话框的“定义”页面定义以下属性。

属性	说明	必需/可选
名称	对象名称。可以包含字母数字字符（包括特殊字符和空格）。名称区分大小写。对象名称在类中必须唯一。不同类中的对象可以具有相同的名称。	必需
类型	对象数据类型。可以为四种类型之一： <ul style="list-style-type: none"> • 字符 • 日期 • 长文本 • 数字 Universe 设计工具的当前版本中不支持 Blob。	必需
说明	对象的说明。可以通过“查询”窗格查看此字段，因此可以包括有关可能对最终用户有用的对象的信息。按 Ctrl+Return 键将指针移到下一行。	可选
Select	对象推导出的 Select 语句。可以使用 SQL 编辑器来创建 Select 语句。请参阅第 289 页上的 “属性” 一节。	必需
Where	对象推导出的 Select 语句的 Where 子句。Where 子句限制查询中返回的行数。可以使用 SQL 编辑器来创建 Where 子句。*	可选

* 可在 Select 语句或 Where 子句中插入或编辑 @Prompt。右键单击 Select 语句或 Where 子句，如果语句中不包含 @Prompt，则快捷菜单可提供“新建 @Prompt”，如果单击内部现有 @prompt，则快捷菜单可提供“编辑 @prompt”。@Prompt 编辑器开启。

“表”按钮

当您单击“表”按钮时，将会出现模式中所使用表的列表。可以从此列表中选择要包括在对象定义中的其他表中的其他列。这样，对象将可以在 Select 语句中推导出来自多个表中的列。有关更多信息，请参阅第 320 页上的[“通过推导多个表来应用限制”](#)一节。

“分析”按钮

当您单击“分析”按钮时，将会分析对象的 Select 语句。如果检测到语法错误，则会出现一个描述错误的消息框。

相关主题

- 第 299 页上的[“使用 SQL 编辑器”](#)
- 第 467 页上的[“OLAP Universe 中的计算所得度量”](#)
- 第 387 页上的[“@Prompt 编辑器”](#)

6.6.4.1 编辑对象定义

编辑对象定义：

1. 双击一个对象。
“编辑属性”对话框即会打开到“定义”页面。
2. 根据需要键入或选择对象定义和属性。
3. 单击“确定”。

6.6.4.2 将对象定义为动态超级链接

可以将单元格中的文本定义为超级链接。此方法对于报表中的动态超级链接很有用，在这种情况下，一系列单元格中的文本将成为指向特定资源（取决于结果对象）的超级链接。

编辑对象的 select 语句以包含超级链接声明，并选择对象的格式属性“作为超级链接读取”。

6.6.4.3 将对象定义为动态超级链接

通过使用此方法，对象将在结果报表中创建一个动态超级链接。

- 1 右键单击对象，然后选择“对象属性”。
将显示“编辑对象的属性”对话框。
- 2 输入 select 语句，并包括相应超级链接。
- 3 保存新属性。
- 4 右键单击对象，然后选择“对象格式”。
将显示“对象格式”窗格。
- 5 选择选项“作为超级链接读取”。
- 6 单击“确定”保存格式设置。
- 7 使用报表工具创建报表并测试链接。

在报表中使用对象时，结果列将包括超级链接。

示例：使用指向日历信息的超级链接

以下 Select 语句从 timeanddate.com 中检索年份日历信息，具体取决于列单元格中的年份值。注意，该声明采用四个最右侧的字符（年份），并且从字符串中删除目标 URL 无法识别的“FY”（表示财务年度）。

```
'<a href=http://www.timeanddate.com/calendar/?year=>' +right(@Select(Reservations\Reservation Year),4) +'</a>'
```

6.6.5 属性

可以通过“编辑属性”对话框的“属性”页面指定以下对象限定和值列表属性：

属性	说明
限定	<p>在“查询”窗格中使用对象时，对象所采用的定义角色。可以将对象限定为以下三种类型之一：</p> <ul style="list-style-type: none"> • 维 • 明细 • 度量 <p>有关对象限定的更详细说明，请参阅第 275 页上的“Universe 中使用什么类型的对象？”一节。</p>
关联值列表	<p>处于选定状态时，将包含数据值的文件与对象关联。默认情况下处于激活状态。有关更多信息，请参阅第 332 页上的“使用值列表”一节。</p>

6.6.5.1 指定对象限定和值列表属性

指定对象的限定和值列表属性：

- 1 双击一个对象。
将出现该对象的“编辑属性”框。
- 2 单击“属性”选项卡。
将出现“属性”页面。
- 3 单击限定单选按钮，以便确定对象是维、明细还是度量。
如果要将返回值的列表与对象关联，请选择“关联值列表”复选框。
有关创建和使用值列表的信息，请参阅第 332 页上的[“使用值列表”](#)一节。
- 4 单击“确定”。

6.6.6 高级

“高级” 页面如下所示。

安全访问级别(L)
用户要使用此对象，其优先权要高于或等同于：

公用

可以使用在

☒ 结果(R)

☒ 条件(C)

☒ 排序(S)

数据库格式(D)
在缺省情况下，下列格式决定了区域设置。您可以通过读取对象数据的方式指定另一种格式。

可以通过“编辑属性”对话框的“高级”页面定义以下属性：

属性	说明
安全访问级别	<p>定义对象的安全访问级别。可以选择一个安全级别，它将限制具有相应安全级别的用户使用对象。</p> <p>可以分配以下安全访问级别：</p> <ul style="list-style-type: none">• 公用• 受控制• 受限• 保密• 专用 <p>如果分配“公用”，则所有用户都能看到并使用对象。如果分配“受限”，则只有具有“受限”或更高级别用户配置文件的用户才能查看并使用对象。</p>
可在结果中使用	选中时，将可以在查询中使用对象。
可在条件中使用	选中时，可以在条件中使用对象进行设置。

属性	说明
可在排序中使用	选中时，可对返回的值进行排序。
数据库格式	<p>仅适用于日期对象的选项。</p> <p>默认情况下，对象的日期格式是在 MS-Windows “控制面板” 的 “区域设置属性” 对话框中定义的。可以修改此选项，以便使用目标数据库格式存储日期。例如，日期格式可以为美国格式或欧洲格式。有关修改此值的信息，请参阅第 300 页上的 “定义对象格式” 一节。</p>

6.6.6.1 定义对象安全性和用户权限

定义对象的安全性和用户权限：

- 1 双击一个对象。
将出现该对象的 “编辑属性” 框。
- 2 单击 “高级” 选项卡。
将出现 “高级” 页面。
- 3 从 “安全访问级别” 下拉列表框中选择一个安全访问级别。
- 4 选择 “可以使用在” 组框中的一个或多个复选框。
- 5 在 “数据库格式” 文本框中键入一个日期格式（如果要修改默认日期格式）。
- 6 单击 “确定”。

6.6.7 定义索引感知

“键”选项卡允许您为对象定义索引感知。索引感知是指利用键列上的索引来加速数据检索过程的功能。

通过 Universe 设计工具创建的对象基于对最终用户有意义的数据库列。例如，“客户”对象将检索包含客户名称的字段。在这种情况下，客户表通常具有对最终用户没有意义，但对数据库性能则至关重要的主键（例如，整数）。在 Universe 设计工具中设置索引感知时，请告知该工具哪些数据库列是主键，哪些是外键。这一点可能会通过以下方式对查询性能产生巨大的影响：

- Universe 设计工具可以利用键列的索引来加快数据检索的速度。
- Universe 设计工具可以生成以最有效的方式进行过滤的 SQL。在星型模式的数据库中，这一点特别重要。如果构建涉及对维表中的值进行过滤的查询，Universe 设计工具可通过使用维表外键直接在事实表上应用过滤器。这样就避免了既不必要又耗费资源地联接到维表。

Universe 设计工具在使用索引感知时不会忽略重复项。如果两个客户具有相同名称，Universe 设计工具将只会检索一个名称，除非它感知到每个客户都有独立的主键。

示例：在城市列表中查找客户

在此示例中，您将在 Island Resorts Marketing Universe 上构建按客户汇总的收入报表，该报表包含来自于“Houston（休斯顿）”、“Dallas（达拉斯）”、“San Francisco（旧金山）”、“San Diego（圣迭哥）”或“Los Angeles（洛杉矶）”的客户。为此，您将“客户”和“销售收入”对象拖入“查询”窗格中的“结果对象”窗格，然后将“城市”对象拖到“条件”窗格并将城市限制为上面列出的城市。

如果不使用索引感知，Universe 设计工具会生成以下 SQL：

```
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
```

在此示例中，此工具创建了一个指向 City 表的联接，以便限制检索出的城市。

如果有索引感知，则可以告知 Universe 设计工具 city_id 是 City 表的主键，并且它还会以外键的形式出现在 Customer 表中。通过使用此信息，此工具可以在不联接到 City 表的情况下限制城市。SQL 如下所示：

```
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
```

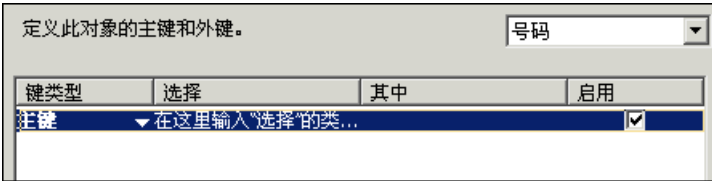
在此示例中，此工具能够生成只需通过过滤 city_id 外键的值即可限制城市的 SQL。

6.6.7.1 设置主键索引感知

设置主键索引感知：

- 1 右键单击要在上面设置索引感知的对象，然后从菜单中选择“对象属性”。
将出现“编辑属性”对话框。
- 2 单击“键”选项卡。
- 3 单击“插入”。

即会在“键”页面上插入如下所示的主键行。



- 4 执行以下操作，以便为主键创建键感知：
 - 在“键类型”列表中选择“主要”。单击“选择”字段中的“...”按钮，打开 SQL 编辑对话框。



将出现 SQL 编辑器。

- 使用 SQL 编辑器来构建主键 SQL SELECT 子句，或直接键入该子句。例如，对于 City 对象，主键 SQL 为 City.city_id。

有关 SQL 编辑器的更多信息，请参阅第 299 页上的 [“使用 SQL 编辑器”](#)。

- 从键类型的下拉列表中选择主键数据类型。
- 5 如果要添加 WHERE 子句，请执行以下操作：
- 在如下所示的“Where”列下的行内单击：



- 单击“Where”字段中的“...”按钮，打开 SQL 编辑对话框。
- 将出现 SQL 编辑器。
- 使用 SQL 编辑器来构建主键 SQL WHERE 子句，或直接键入该子句。上面的示例中没有 Where 子句。
 - 从关键字类型下拉列表中选择“数字”。
- 6 选择“启用”。
- 7 单击“确定”。

注意：

要为主键添加多个列，可以拼接多个列以定义主键。这些列应当属于同一个表，而且数据类型应当相同。

例如，在示例数据库“club.mdb”中，Resort 表有一个基于“Country_id”和“Resort_id”的多列主键。

因此在对 <resort> 对象定义索引感知时，用户可以拼接“Country_id”和“Resort_id”以定义主键：

```
Resort.country_id & Resort.resort_id
```

& 是 MS Access 拼接运算符。

6.6.7.2 设置外键感知

设置外键感知：

- 1 右键单击要在上面设置索引感知的对象：
从菜单中选择“对象属性”。
将出现“编辑属性”对话框。
- 2 单击“键”选项卡。

3 单击“插入”。

即会将一个键行插入“键”页面。

4 执行以下操作，以便为外键创建键感知：

- 在“键类型”列表中选择“外键”。
- 单击“选择”字段中的“...”按钮，打开 SQL 编辑对话框。

将出现 SQL 编辑器。

- 使用 SQL 编辑器来构建外键 SQL SELECT 子句，或直接键入该子句。
- 从键类型的下拉列表中选择外键数据类型。

5 为组成外键的所有列重复步骤 3 和 4。

6 如果要添加 WHERE 子句，请执行以下操作：

- 在“Where”列下突出显示的行中单击。
- 单击“Where”字段中的“...”按钮，打开 SQL 编辑对话框。

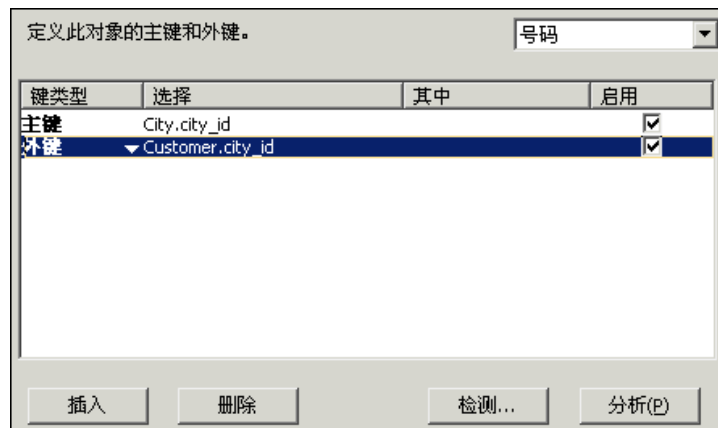
将出现 SQL 编辑器。

- 使用 SQL 编辑器来构建外键 SQL WHERE 子句，或直接键入该子句。
- 从关键字类型下拉列表中选择“数字”。

7 选择“启用”。

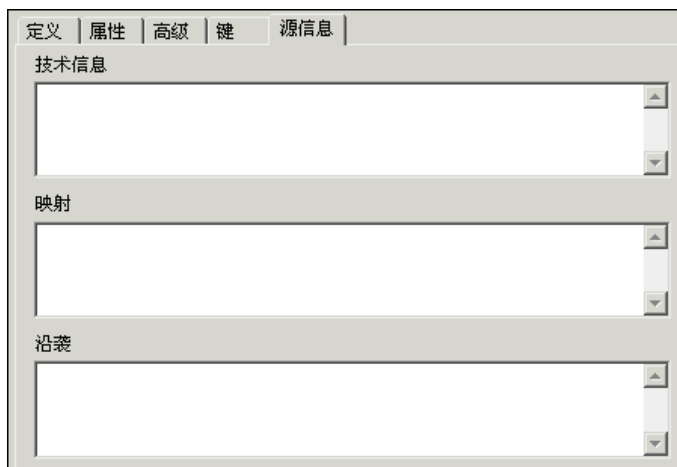
8 为外键中的所有列重复上面的步骤。

在“定义索引感知”一节的示例中，“键”选项卡应如以下所示：



6.6.8 源信息

“源信息”页面由从 Data Integrator 中生成的 Universe 所使用。“源信息”选项卡如下所示：



对于通过 Data Integrator 生成的 Universe，用于依据源表计算目标表的技术说明和公式显示在此选项卡中。Web Intelligence 用户可以使用此信息。

可以在“源信息”选项卡中指定以下类型的信息：

- 技术说明：Universe（从 Data Integrator 中生成）中可用的技术说明。
- 映射信息：在 Data Integrator 内的源表和目標表之间应用的映射。目的不是为了提供映射表达式，而是将映射表达式显示为描述性说明，以通知对象定义中所使用的源列的用户。
- 数据沿袭信息：目标列中包含的源列的列表。此信息有助于通过 Data Integrator 和 Web Intelligence 报表进行影响分析。

6.6.9 使用 SQL 编辑器来定义对象

可以使用 SQL 编辑器帮助为对象定义 SELECT 语句或 WHERE 子句，以及为 OLAP Universe 对象插入 MDX 运算符和函数。SQL 编辑器是一种在树形视图中列出表、列、对象、运算符和函数的图形式编辑器。双击任何列出的结构即可将其插入“选择”或“位置”框。

可以在 SQL 编辑器中编辑以下选项：

编辑选项	说明
表和列	<p>出现在“结构”窗格中的所有表及它们各自的列。</p> <p>注意： 此选项仅可用于关系 Universe，而不可用于 OLAP Universe。</p>
类和对象	出现在“Universe”窗格中的所有类及它们各自的对象。
运算符	可用于在 Select 语句中组合 SQL 结构或在 Where 子句中设置条件的运算符。
函数	<ul style="list-style-type: none"> 数据库函数，比如数值、字符和日期函数。 特定于 Business Objects 产品的 @ 函数。 <p>可用的函数列在目标数据库参数 (.PRM) 文件的“函数”条目下。每个支持的数据库都有一个 .PRM 文件。这些文件存储在 BusinessObjects 路径的 Data Access 文件夹中。可以通过编辑 .PRM 文件来添加或修改可用函数。</p> <p>《数据访问指南》中描述了编辑 .PRM 文件的过程。</p>
显示对象 SQL	如果处于选定状态，则会为“选择”或“位置”框中出现的对象显示 SQL 语法。
分析	单击时将分析语法。如果语法无效，则会出现一个描述问题的消息框。
说明	显示选定对象或函数的说明。

相关主题

- 第 468 页上的[“关于多维数据集查询的 MDX 函数”](#)
- 第 299 页上的[“使用 SQL 编辑器”](#)

6.6.9.1 关于多维数据集查询的 MDX 函数

使用 MDX 编辑器定义多维数据集查询。

向 OLAP Universe 添加新对象或预定义过滤器时，可使用针对特定数据源连接的一系列 MDX 表达式。

可用表达式库存储于 .prm 连接文件中。针对某对象打开“编辑属性”窗格以及针对某查询打开“编辑 SELECT 语句”窗格时，“函数”窗格中将显示可用的表达式。要在 SELECT 或 WHERE 语句中插入表达式，请在语句中单击需要插入该表达式的位置，然后双击适合的表达式。

OLAP Universe MDX 字典 — 函数列表 (PRM 文件)

向 OLAP Universe 添加新对象或预定义过滤器时，针对表达式中使用的正确 OLAP 连接 (SAP 或 MSAS)，可在对象和过滤器编辑器中使用 MDX 函数 (主要是成员函数) 和运算符。有关如何设置 SAP 或 MySQL (sap.prm、sqlsrv_as.prm) 的说明，请参阅《数据访问指南》。可用的函数和运算符取决于 Universe 的连接类型。PRM 文件针对每个连接提供此函数列表。不提供所支持函数的完整列表，仅提供最常用的函数。

以下 MDX 运算符可用于查询。

- Equal
- NotEqual
- InList
- NotInList
- Greater
- GreaterOrEqual
- Less
- LessOrEqual
- Between
- NotBetween
- Like
- NotLike

下表列出了编辑条件时可用的部分 MDX 文件夹函数的示例。可用函数取决于基础数据库。

- 集合函数 (ADDCALCULATEDMEMBERS、ALLMEMBERS……)
- 统计/数字函数 (AGGREGATE、AVG……)
- 导航/成员函数 (ANCESTOR、ASCENDANTS……)
- 元数据函数 (AXIS、HIERARCHY……)

6.6.9.2 使用 SQL 编辑器

可以使用 SQL 编辑器在对象定义中插入 SQL 和 MDX 表达式。还可右键单击 SQL 语句，选择“新建 @Prompt”，以在 SQL 中插入 @Prompt 表达式，或选择“编辑 @Prompt”，以对现有 @Prompt 表达式进行编辑。可打开 @Prompt 编辑器。

使用 SQL 编辑器：

1. 双击一个对象。
将出现该对象的“编辑属性”对话框。
2. 单击“选择”或“位置”框旁边的 >> 按钮。
将出现“编辑 Select 语句”或“编辑 Where 子句”对话框。
3. 在 Select 语句或 Where 子句中要添加结构语法的位置单击。如果该框为空，请在框中的任意位置单击。光标自动出现在对话框的左上角。
4. 展开表节点以显示列。
5. 双击一个列，以便在 Select 语句或 Where 子句中插入列定义。
提示：
要从选定列的值列表中选择一个或多个值，请右键单击该列并选择“值列表”。
6. 展开类节点以显示对象。
7. 双击一个对象，以便在 Select 语句或 Where 子句中插入 @Select 或 @Where 函数。这些函数指示当前对象使用选定对象的 Select 语句或 Where 子句。有关使用 @ 函数的更多信息，请参阅第 383 页上的 [“在对象的 SQL 中使用 @ 函数”](#) 一节。
8. 双击一个运算符，以便在编辑框中插入运算符。
9. 展开函数节点以显示变量函数。
10. 双击一个函数，以便在编辑框中插入函数。
11. 单击“分析”按钮以验证语法。
12. 单击“确定”。

6.6.10 定义对象格式

可以为选定对象的数据值定义格式。格式适用于 Web Intelligence 报表的单元格中显示的相关数据值。

“对象格式”对话框中的各个选项卡包括了数值、对齐方式、字体、边框和阴影的设置。

例如，用户可以使用如 \$1,000 的格式显示一个整数，而不是用默认格式 1,000.00。用户也可以对重要数据值应用颜色，例如红色。

“数值”、“货币”、“科学型”和“百分比”类别仅适用于具有数值类型的对象和变量，而“日期/时间”类别仅适用于具有日期类型的对象和变量。

有关格式的信息将随 Universe 一起导出和导入。

可以使用“删除对象格式”命令来删除定义的任何格式。

相关主题

- 第 468 页上的“[关于多维数据集查询的 MDX 函数](#)”

6.6.10.1 修改对象格式

修改对象格式：

- 1 右键单击一个对象。
- 2 从上下文菜单中选择“对象格式”。
将出现“对象格式”表单。
- 3 单击一个格式选项卡，然后选择或键入对象的格式。
- 4 单击“确定”。

6.6.10.2 删除对象格式

可以随时删除对象的格式。

删除对象格式：

- 选择一个对象，然后选择“文件”>“删除格式”。
- 或
- 右键单击对象，然后从上下文菜单中选择“删除格式”。

6.6.11 查看对象定义中使用的表

可以在“结构”窗格中查看在“Universe”窗格的对象定义中使用的表。如果要在对象名称不能明确指示特定表的情况下快速地识别一个表，这一点可能十分有用。

6.6.11.1 查看对象使用的表

查看对象使用的表：

- 1 右键单击“Universe”窗格中的一个对象。
将出现一个上下文菜单。
- 2 从该上下文菜单中选择“查看关联的表”。
关联的表即会在“结构”窗格中突出显示。

6.6.12 定义维

维是一种对象，它是查询中分析的重点。维将映射到数据库中的一个或多个列或函数，这些列或函数是查询的关键所在。例如，“Country（国家/地区）”、“Sales Person（销售人员）”、“Products（产品）”或“Sales Line（销售线）”。

维是创建对象时的默认限定。可以随时将限定更改为维。

定义维对象：

- 1 双击一个对象。
将出现该对象的“编辑属性”对话框。
- 2 单击“属性”选项卡。
将出现“属性”页面。
- 3 在“限定”组框中选择“维”单选按钮。
- 4 单击“确定”。

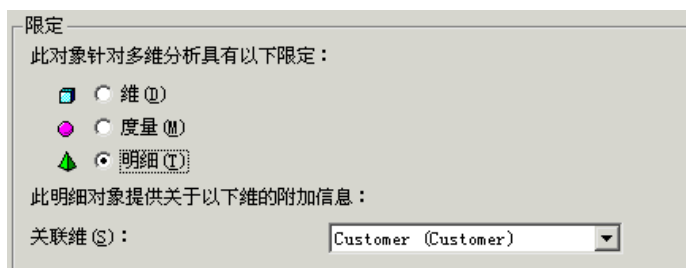
6.6.13 定义明细

明细提供有关维的描述性数据。明细始终会附加到维。它映射到数据库中的一个或多个列或函数，这些列或函数提供了与维相关的详细信息。

可以通过以下方式定义明细对象：选择“明细”作为对象的限定，并指定附加到明细的维。

定义明细对象：

- 1 双击一个对象。
将出现该对象的“编辑属性”对话框。
- 2 单击“属性”选项卡。
将出现“属性”页面。
- 3 在“限定”组框中选择“明细”单选按钮。
将出现“关联维”下拉列表框，并列出 Universe 中的所有维对象。
- 4 从该下拉列表框中选择一个维。明细描述了这个维的质量或属性。



- 5 单击“确定”。

6.6.14 定义度量

可以通过选择“度量”作为对象的限定来定义度量对象。由于度量是动态对象，因此它们非常灵活。度量对象返回的值因查询中与度量一起使用的维和明细对象而异。例如，如果在一个查询中与“Sales Revenue（销售收入）”对象一起使用，然后在别的查询中与“Region（地区）”和“Country（国家/地区）”对象一起使用，“Sales Revenue”度量将返回不同的值。

由于度量对象相对于维和明细而言更为复杂并且功能更为强大，因此以下各节中将对它们进行更为深入的讨论。

6.6.14.1 度量返回何种类型的信息？

度量对象返回数值信息。可以通过使用聚合函数创建度量。下面是五种最常用的聚合函数：

- Sum
- Count
- Average
- Minimum
- Maximum

6.6.14.2 度量与维和明细有何不同？

度量在以下几方面有别于维和明细：

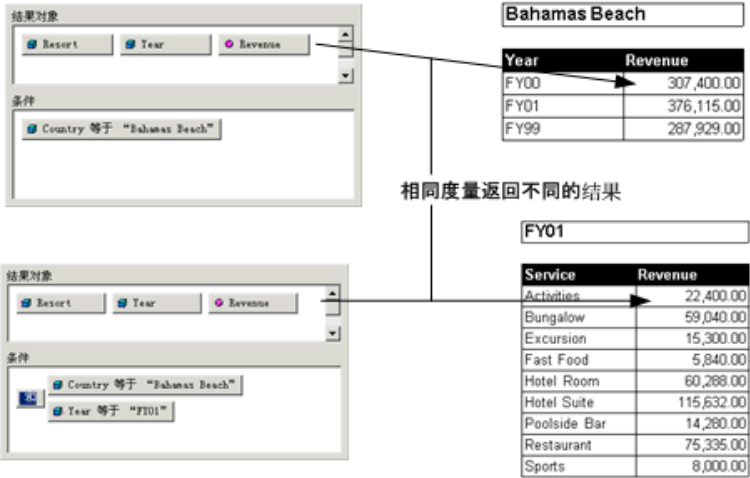
- 度量是动态的
- 度量能够投影聚合

下面描述了这两个属性：

6.6.14.3 度量如何动态地表现？

度量对象的返回值因查询中与度量对象一起使用的维或明细对象而异。

以下示例显示了两个具有不同维（导致度量返回不同的值）的独立查询中使用的相同“收入”度量对象。



6.6.14.4 度量推导出 Group By 子句

当您运行包括度量对象（具有其他类型的对象）的查询时，将自动在 Select 语句中推导出 Group By 子句。

Group By 子句的推导取决于以下 SQL 规则：

如果 Select 子句行包含聚合，则子句中该聚合外部的所有内容必须也出现在 Group By 子句中。

根据此规则，在与度量对象同一查询中使用的任何维或明细将始终包含在自动推导出的 Group By 子句中。为了确保查询返回正确的结果，维和明细对象“不能”包含聚合。

以下示例显示，“休假地”、“服务系列”和“年度”维对象都是在 Select 子句和 Group By 子句中推导出的。

在 GROUP BY 语句中推导出的维

聚合到最低级别休假地 (Resort)，然后按 Service Line 和 Year 列出的结果

```

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')
  
```

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

注意：

如果查询只包含一个度量对象，则不会推导出 Group By 子句。

6.6.14.5 为度量设置聚合投影

在创建度量时，您必须指定将聚合函数投影到报表上的方式。

将在查询过程的以下两个级别聚合度量对象的返回值：

- 查询级别。使用推导出的 SELECT 语句对数据进行聚合。
- 微多维数据集到块级别。将数据从微多维数据集投影到报表中的块时。度量的这种投影功能允许在微多维数据集中以本地方式进行聚合。

注意：

微多维数据集是用于在将查询投影到报表中之前表示查询所返回数据的一种概念性方式。它表示由 Business Objects 报表产品保留在内存中的返回值。块级别是用户使用返回数据创建的二维报表。用户可以选择使用保留在微多维数据集中的全部数据或只使用部分数据来创建报表。用户还可以在微多维数据集中的返回值上执行聚合函数（本地聚合），以便在报表上创建新的值。

这两个聚合级别按如下方式放置在查询过程中：

- 用户在 Web Intelligence 中创建查询。
- Web Intelligence 从查询中推导出 SQL，然后将 SELECT 语句发送到目标数据库。
- 数据返回到微多维数据集。这是第一个聚合级别。
- 微多维数据集将聚合数据投影到报表上。在“查询”窗格中分解出需要聚合到更低级别的数据。这是第二个聚合级别。

当您开始进行查询时，Select 语句的结果集存储在微多维数据集中，并且微多维数据集中包含的所有数据将被投影到块中。由于数据是从微多维数据集中包含的最低级别投影的，因此不会发生投影聚合。

但是，当您使用“查询”窗格以便仅从微多维数据集中投影部分数据时，则必须进行聚合才能在较高级别显示度量值。

例如，在前一示例中，如果不将年度数据投影到块中，则需要将与“年度”相关的三个行缩减为一个行以显示该休假地的总“销售收入”，因此使用了加总聚合。

可以在度量的“编辑属性”表单的“属性”页面上设置投影聚合（右键单击“对象”>“对象属性”>“属性”）。

投影聚合与 SELECT 聚合不同。

相关主题

- 第 463 页上的[“投影函数“委派给数据库”](#)”

6.6.14.5.1 投影函数“委派给数据库”

在 Universe 中，任何度量均可以具有投影函数（Sum、Min、Max、Count 和 Avg）。当报表中显示的维数小于查询结果集中的维数时，投影函数用于在 Web Intelligence 中对度量进行本地聚合。

非叠加度量（例如比率、平均和权重）只能显示在与查询结果集相同的聚合级别上。因此，在 Universe 中非叠加度量的投影函数通常设置为“无”。

使用投影函数“委派给数据库”可以将非叠加度量的聚合委派给数据库服务器。在 Web Intelligence 中这些称为智能度量。在对象属性的属性页上，智能度量的投影函数设置为“委派给数据库”。有关如何在 Web Intelligence 中使用这些函数以及其他函数的更多信息，请参阅文档《在 Web Intelligence 中使用函数、公式和计算》中的“使用智能度量计算值”一节。

注意：

对于基于 MSAS 和 Essbase 数据源的 OLAP Universe，默认情况下通过设置为“委派给数据库”的投影函数创建 Universe 中的所有度量。

注意：

在使用基于已设置聚合感知的度量的智能度量时，应注意以下限制：强烈建议确保度量定义中使用的各聚合表具有一致的数据（就明细值而言聚合值是准确的），否则智能度量会生成不一致的数据。例如，如果为智能度量使用了一个年度聚合表和一个日聚合表，就全部年度而言，年度聚合表的数据与日聚合表的数据保持一致，然而对于当前年度，年度表可以为空，而日聚合表在每日基础上的数据是准确的。在这种情况下，使用基于当前年度和日聚合表的智能度量的报表会给出不连贯的结果。

示例：智能度量

在本例中，一个查询包含两个维（Country 和 Region）和三个度量（Order Amount、Amount Delivered，以及 % Delivered）。

L01 Region	Amount Delivered	Order Amount	% Delivered
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		总和:	299.992

“% Delivered” 的总和不正确，因为它等于 “% Delivered” 列的合计。

如果在 Universe 中将此度量的投影函数设置为“委派给数据库”，当用户刷新报表时，Web Intelligence 会连接到数据库以计算正确的值。

L01 Region	Amount Delivered	Order Amount	% Delivered
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		总和:	299.992
		总计:	99.997

注意：

使用某些函数时必须特别谨慎，例如比率函数（Average）。在计算列的平均值时，如果函数配置不正确，其行为可能会异常。

例如，SQL 函数 `sum(Shop_facts.Margin)/sum(Shop_facts.Quantity_sold)` 可能得到意外的结果。如果配置不正确，它会计算每个单元格的平均值，并返回所有平均值的总和。要更正结果，必须按以下所示配置函数的参数：

- 。 1 转到函数的“编辑属性”选项。
- 。 2 从“函数”下拉列表中为选项“选择在聚合时如何投影此度量”选择“委派给数据库”。

- 。 3 保存更改。

相关主题

- 第 466 页上的“[为度量设置聚合投影](#)”

6.6.14.6 创建度量

创建度量：

- 1 双击一个对象。
将出现该对象的“编辑属性”对话框。
- 2 单击“属性”选项卡。
将出现“属性”页面。
- 3 在“限定”组框中选择“度量”单选按钮。
将出现一个列出聚合函数的“函数”下拉列表框。
- 4 选择一个函数。
- 5 单击“确定”。

相关主题

- 第 466 页上的“[为度量设置聚合投影](#)”
- 第 463 页上的“[投影函数“委派给数据库”](#)”

6.6.15 为对象定义限制

限制是 SQL 中的一种条件，它设置用于限制查询所返回数据的标准。

可以在对象上定义限制，以便限制可供用户使用的数据。限制用户访问数据的考虑应基于目标用户的数据需求。用户可能不需要访问对象返回的所有值。也可能出于安全考虑想要限制用户访问某些值。

可以在 Universe 中定义两种类型的限制：

限制类型	说明
强制	在对象的 Where 子句中定义的限制。用户无法访问该限制，因此无法在 Web Intelligence 中改写它。
可选	在特殊条件对象中定义的限制，用户可选择在查询中使用或不使用这些条件对象。条件对象是一个预定义的 Where 子句，可以插入到“查询”窗格中由对象推导出的 Select 语句。

注意：

在 Web Intelligence 中，用户可以在“查询”窗格中应用条件。Universe 设计员应避免创建容易在用户级别应用的可选限制。用户可以在必要时自己创建这些条件。

6.6.15.1 为对象定义 Where 子句

通过在对象的“编辑属性”对话框的“定义”页上“位置”框中添加条件，可以对对象应用进一步的限制。

可以在创建对象时定义条件，也可以随时将条件添加到对象定义。

在 Universe 中，可通过两种方式在 SQL 语句中使用 Where 子句来限制查询返回的行数。

- 借助模式中的联接链接表，为对象在 SELECT 语句中自动推导出 WHERE 子句。联接通常基于表之间的等同性。通过限制从联接表中返回的数据，它们可以防止创建笛卡尔积。
- 为对象在 WHERE 子句中添加条件。这是联接所推导出的现有 WHERE 子句的附加条件。例如，若要限制用户查询数据的某个子集，则可以定义 WHERE 子句来进一步限制查询中返回的数据。

示例：修改对象的默认（仅联接）Where 子句

下面的报表是一个不受限制的块，包含来自所有国家/地区的销售人员的数据：

Sales Person	Country of origin
Barrot	France
Carlin	France
Edwood	UK
Fischer	Germany
Galagers	US
Ishimoto	Japan
Nagata	Japan

此查询的 SQL 如下所示。Where 子句只包含由 Customer、City、Region 和 Sales_Person 表之间的联接推导出的限制。

```
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 )
```

如果想要限制用户只能看到特定于 France（法国）的返回值，您可以将一个条件添加到“Country（国家/地区）”对象的 Where 子句。以下报表只显示法国的销售人员：

Sales Person	Country of origin
Barrot	France
Carlin	France

该查询的 SQL 如下所示：

```
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' )
```

WHERE 子句多出了一行。这一行就是刚添加到“Country”对象的 WHERE 子句的限制。

注意：

除了自我限制的联接之外，用户不能在 WHERE 子句中创建联接。“检测上下文”（自动上下文检测）或聚合感知不兼容性检测将不会考虑 WHERE 子句中的联接。应确保“结构”窗格中的所有联接都是可见的。这样可确保所有联接都可供 Universe 设计工具自动检测工具使用。

6.6.15.2 定义 Where 子句

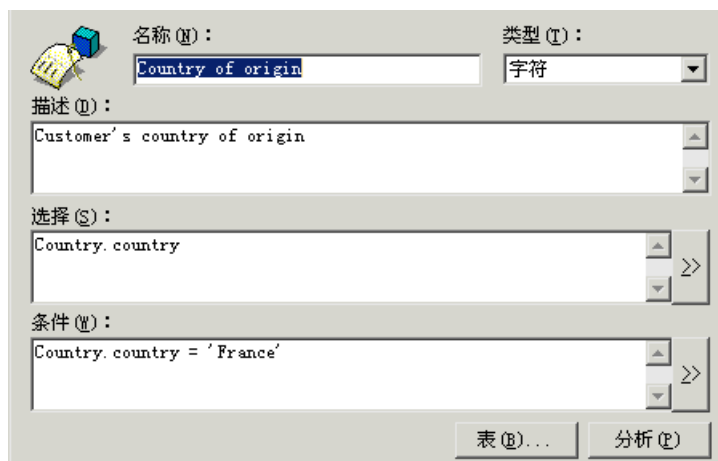
定义 Where 子句：

- 1 双击一个对象。
“编辑属性”对话框即会打开到“定义”页面。
- 2 在 Where 子句文本框中直接键入语法。
或
单击“位置”框旁边的 >> 按钮打开 Where 子句编辑器。
- 3 双击 SQL 结构和功能列表中显示的列、对象、运算符或函数。

提示：

可以按如下方式为 Where 子句选择值：右键单击“表和列”列表中的一个列。选择“查看值”。将出现该列所有值的列表。可以选择要插入 Where 子句的一个或多个值（例如，在使用 In 运算符时）。

- 4 单击“确定”以关闭编辑器。
“Country”对象的 Where 子句如下所示。它将“Country”的值限制为只限“France”。



名称 (N): Country of origin 类型 (T): 字符

描述 (D): Customer's country of origin

选择 (S): Country.country

条件 (C): Country.country = 'France'

表 (T)... 分析 (A)

- 5 单击“确定”。

6.6.15.3 使用 Where 子句时的问题

Where 子句是一种十分有用的限制数据的方式，但在 Universe 中必须小心使用它们，以避免以下问题：

问题	说明	解决方法
产生大量的相似对象。	如果通过创建若干对象（每个对象推导出一个数据部分的 Where 子句）限制对象数据，您可能会使用相似的名称作为多个对象的结尾。例如，法国客户、美国客户和日本客户。对于看到多个看起来相似的对象的用户而言，这可能会产生混淆。	为每个限制创建条件对象。
难于创建层次。	如果有多个在相同数据上推导出 Where 子句的对象，用户将难于构建用于向下钻取的逻辑默认层次。	为每个限制创建条件对象。
对象名称和应用的限制之间出现混淆。	除非对象采用非常精确的命名方式，否则，仅从对象的名称来看，限制对于用户来说可能并非显而易见。用户可以通过查看查询的 SQL 来查看 Where 子句，但并非所有用户都会在运行查询之前查看 SQL。	<ul style="list-style-type: none"> • 为每个限制创建条件对象。 • 以适当的方式命名每个对象。
Where 子句之间出现冲突。	如果同一查询中包括了两个或两个以上采用类似方式加以限制的对象，则 Where 子句之间的冲突将导致不会返回任何数据。	为每个限制创建条件对象，并确保用户在报表级别合并或同步了查询。

创建条件对象将会解决多个对象、难于创建层次和对象名称冲突问题。

通过创建条件对象，并确保用户知道他们必须使用 UNION 或 SYNCHRONIZE 运算符在报表级别联接查询，可以解决 Where 子句之间的冲突。

如果对象定义中定义的 Where 子句可能存在问题，您应避免使用这些子句，并在可能时创建条件对象，正确地使用了条件对象时，将可以避免硬编码 Where 子句的问题。

注意：

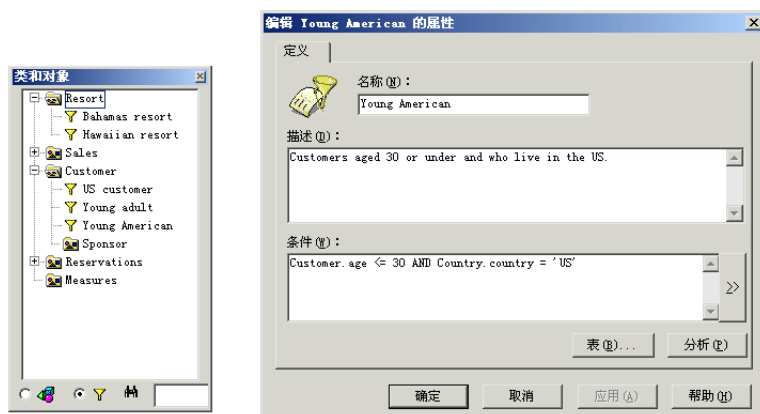
除了自我限制的联接之外，您不能在条件对象中创建联接。条件对象中的联接相当于在可重用 Where 子句中创建联接，因此，“检测上下文”（自动上下文检测）或聚合感知不兼容性检测将不会考虑条件对象中的联接。应确保“结构”窗格中的所有联接都是可见的。这样可确保所有联接都可供自动检测工具使用。

6.6.16 定义条件对象

条件对象是一个预定义的 Where 子句，可以插入到“查询”窗格中由对象推导出的 Select 语句。

条件对象存储在“Universe”窗格的“条件”视图中。可通过单击“Universe”窗格右下角的“条件”单选按钮来访问条件视图。

Beach Universe 的条件对象和 Young American 条件推导出的 Where 子句如下所示。



6.6.16.1 使用条件对象的优点和限制

使用条件对象具有以下优点：

- 对于复杂或频繁使用的条件十分有用。
- 使用户能够选择应用条件。
- 无需多个对象。

- 条件对象不会更改“Universe”窗格中类和对象的视图。

注意：

您可能需要指示用户使用“Universe”窗格的条件对象视图。

使用条件对象的唯一缺点是：可能需要强制对用户使用条件才能限制他们访问部分数据集。在这种情况下，您需要在对象定义中定义 Where 子句。

6.6.16.2 条件对象未解决相互冲突的 Where 子句问题

使用条件对象不能解决相互冲突的 Where 子句（返回空数据集）的问题。如果用户运行包含两个条件对象（访问相同数据）的查询，由于这两个条件使用 AND 运算符合并在一起，因此这两个条件未得到满足，从而不会返回数据。用户可通过创建两个查询（每个条件对象一个），然后将这些查询合并，从而在报表级别解决此问题。

6.6.16.3 必需过滤器

必需过滤器分为两种类型：

- **Universe:** Universe 必需过滤器与其所属的类之间不存在依赖关系。Universe 必需过滤器包括在查询中，且独立于查询中包括的对象（维、度量和详细信息）。

大多数 SAP NetWeaver Business Warehouse (BW) 变量是在 NetWeaver BW 上生成 OLAP Universe 时作为 Universe 必需过滤器创建的。

- **类:** 仅当查询中使用对象的类中的项目时，类必需过滤器才会出现。

当用户执行以下操作时会触发类必需过滤器：

- 将对象（维、度量或详细信息）添加至 Web Intelligence 的“查询面板”的“结果”窗格中。
- 将 Universe 预定义过滤器添加至“查询面板”的“过滤器”窗格中，即使没有在“结果”窗格中选定属于相同类的对象。
- 使用属于必需过滤器的类的对象（维、度量或详细信息）创建过滤器。

必需过滤器可以具有默认值，也可以与值列表关联。

必需过滤器被隐藏，无法在 Web Intelligence 的“查询面板”中被选定。在 Universe 设计工具中，如果在查询中将过滤器设置为必需，该过滤器将自动隐藏，并且会禁用“显示项目”命令。当取消选择“必需”选项之后，该过滤器不再隐藏。“隐藏项目”命令变为启用。

最终用户查询可以包括多个必需过滤器。默认情况下，查询中的所有必需过滤器使用 And 运算符联接。

所有子类从其父类继承必需过滤器。但是，请注意：

- 如果某个对象（维、度量、详细信息）使用 @Select 函数引用其它对象，该对象将不会继承所引用对象的类必需过滤器。
- 如果某个对象的 WHERE 子句中使用 @Where 函数引用另一个对象的 WHERE 子句，则不会继承所引用对象的类必需过滤器。
- 如果某个预定义过滤器使用 @Where 函数引用另一个预定义过滤器或对象的 WHERE 子句，则同样不会继承所引用对象的类必需过滤器。

示例：OLAP Universe 中的必需过滤器

以下过滤器（以 XML 代码显示）可验证用户在提示中输入的代码。

```
<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>
```

相关主题

- 第 316 页上的[“必需过滤器示例”](#)
- 第 317 页上的[“必需过滤器和值列表”](#)

6.6.16.3.1 必需过滤器示例

以下示例演示了如何使用 Universe 必需过滤器：

用表中存储的登录信息来验证用户输入的登录信息：

```
1 = (Select 1 from Club.dbo.Login
where Login = @Variable('BOUSER')
AND Password = @Prompt('Password?', 'A',, mono, free) )
```

将 Universe 的使用限定在上午 9 点至下午 6 点之间：

```
1 = (select 1
where datepart(HH, getdate()) between 9 and 18)
```

下面是一个类必需过滤器的示例：

在包含国家/地区/地区/城市/客户的类中定义，将查询限定为特定周期的销售信息。向用户提示此周期。

```
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))
```

相关主题

- 第 315 页上的“[必需过滤器](#)”

6.6.16.4 必需过滤器和值列表

必需过滤器可以与值列表关联。要与值列表关联，必须在过滤器所应用的对象的对象属性页上明确选择值列表选项。

Universe 必需过滤器可以与级联值列表关联。

类必需过滤器可以与级联值列表关联，前提是相应类中至少有一个对象是级联值列表的一部分。即使级联值列表将来自不同类的对象分组，情况同样如此。

建议

仅在级联值列表的顶层生成必需过滤器。

不要将级联值列表与包含提示的必需过滤器关联。Web Intelligence 不支持在级联值列表中使用提示。

6.6.16.5 创建条件对象

创建条件对象：

- 1 单击“Universe”窗格右下角的“条件”单选按钮。
将出现“Universe”窗格的“条件”视图。它包含 Universe 中所有类的树形视图。
 - 2 右键单击一个类，然后从上下文菜单中选择“插入条件”。
- 或

单击某个类，然后单击“插入条件”按钮。

将出现“编辑属性”对话框。默认名称出现在“名称”框中。“Where”框为空。

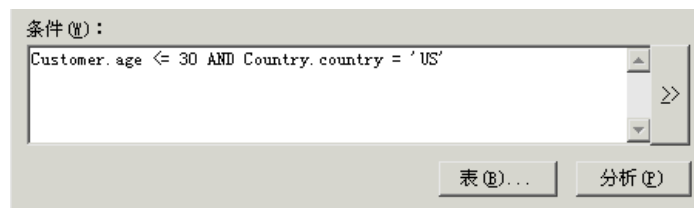
- 3 键入条件的名称。
- 4 在“Where”子句框中直接键入 WHERE 子句语法。

或

单击“Where”子句框旁边的 >> 按钮打开“Where”子句编辑器。

- 5 双击 SQL 结构和功能列表中显示的列、对象、运算符或函数。
- 6 单击“确定”以关闭编辑器。

下面显示了一个名为 Young American 的条件的定义。它将返回的值限制为 30 岁以下（包括 30 岁）的美国客户。

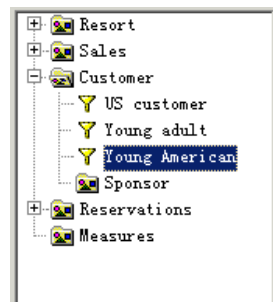


- 7 单击“分析”按钮以验证语法。
- 8 要将过滤器定义为强制的过滤器，请选择“强制的过滤器”复选框。
默认情况下，强制的过滤器应用于类，但不应用于值列表。
- 9 选择相应的单选按钮以将强制的过滤器定义为应用于类或应用于 Universe。
- 10 要将强制的过滤器应用于值列表，请选择“在值列表上应用”复选框。
11. 单击“确定”。

新的条件对象即出现在“Universe”窗格的“条件”视图中。

注意：

要使用预定义的过滤器编辑器编辑条件对象，请单击“>>”。



相关主题

- 第 315 页上的[“必需过滤器”](#)
- 第 317 页上的[“必需过滤器和值列表”](#)
- 第 470 页上的[“OLAP Universe 中的预定义条件”](#)
- 第 475 页上的[“OLAP Universe 中的可选提示”](#)

6.6.16.6 在同一查询中使用条件对象

如果有两个为同一对象定义的条件对象，并且这两个对象都用在同一查询中，则不会返回数据，因为这两个 WHERE 子句会产生 False 条件。在可能时，应避免在对象的定义中对 WHERE 子句进行硬编码，并且当使用条件对象时，用户需要知道这些潜在问题。

通过联接两个查询（每个条件对象一个查询），用户可以解决返回空数据集的问题。

注意：

为了避免 Web Intelligence 用户在同一查询中合并两个条件对象，您可以在条件对象“X”的说明中指明：该对象不能和对象“Y”一起使用。

6.6.16.7 为何多个 Where 子句会返回空数据集？

当您将在 Where 子句添加到对象的定义时，使用 AND 运算符的联接会将限制添加到限制集中。如果在一个查询中合并两个对象（都在同一数据集上应用限制），则会在连续的 AND 子句中合并两个 Where 子句。此类查询的结果是不会有同时满足两个条件的数据，因此不会返回数据。

例如，用户想要了解“Bahamas Resort（巴哈马群岛休假地）”和“Hawaiian Club（夏威夷俱乐部）”酒店休假地提供的服务。以下查询是使用“Bahamas Resort（巴哈马群岛休假地）”和“Hawaiian Club（夏威夷俱乐部）”的条件对象运行的：

此查询的 SQL 如下所示：

```
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' ) )
```

两个 Where 子句限制合并并在 Where 子句结尾的 AND 子句中。

在运行查询时，由于无法在同一查询中满足对国家/地区的两个限制，因此未返回数据。将出现一个消息框，通知您没有要提取的数据。

创建两个查询来合并限制

通过运行两个查询（每个 Where 子句一个），并使用“并”运算符来合并结果，用户可以解决在同一查询中使用两个条件对象的问题。

6.6.17 使用自我限制的联接来应用限制

可以使用自我限制的联接将数据限制为表中的一个或另一个列（基于用于在两个列之间进行切换的标志）。标志是第三个列，它的值确定了在查询中使用这两个备用列中的哪一个。

有关创建和使用自我限制联接的更多信息，请参阅第 188 页上的 [“自我限制的连接”](#) 一节。

6.6.18 通过推导多个表来应用限制

可以将为对象返回的数据限制为该对象所推导出的表中的值，这些值也与另一个表中的值相匹配。

例如，名为“Country of Origin（原始国家/地区）”的对象将推导出 Country 表。“Country of Origin”对象返回以下数据：

Country of origin
Australia
France
Germany
Holland
Japan
UK
US

如果要在“Sales_Person”类下方使用对象“Country of Origin”，以使其仅返回销售人员所在的国家/地区，您可以将对象重命名为“Sales people countries（销售人员所在国家/地区）”，并将 Country 表限制为仅返回 Sales_Person 表中销售人员所在的国家/地区的值。

Sales people countries（销售人员所在国家/地区）对象具有以下 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 )
```

“Sales people countries（销售人员所在国家/地区）”对象返回以下数据：

Sales people countries
France
Germany
Japan
UK
US

通过指定在查询中使用“Country”对象时，必须同时在 Select 语句的 From 子句中推导 Sales_Person 表，您可以应用限制。

则“Sales_Person”类下面的“Country”只会返回销售人员所在的国家/地区。您通过使用对象定义表单中的“表”按钮来应用限制。

必须通过使用相等联接的中间联接将 Country 表联接到 Sales_Person 表。

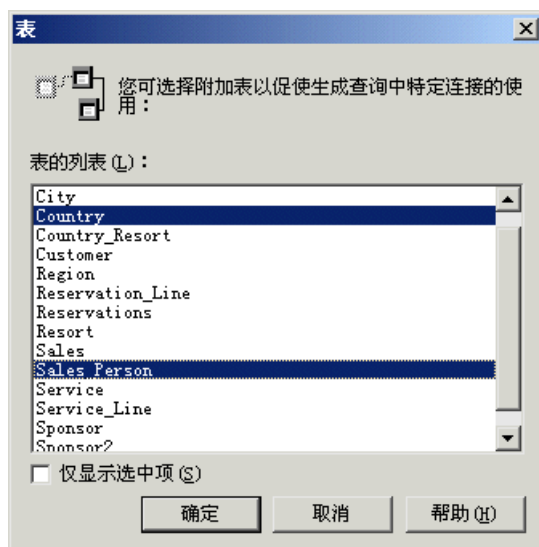
注意：

如果对在 Select 语句中定义了表限制的对象 SQL 进行任何更改，则 Universe 设计工具会自动重新确定对象的 Select 语句和 Where 子句需要哪些表。如果表限制在对象所推导的表中被改写，您将不会收到通知。

6.6.18.1 推导多个表以应用条件

推导为对象应用条件的多个表：

- 1 双击一个对象。
将出现该对象的“编辑属性”对话框。
- 2 单击“表”按钮。
将出现 Universe 中的表的列表。
- 3 除了当前表之外，选择一个或多个您希望对象推导的表。可以通过按下 CTRL 键并单击列表中的表名来选择多个表。下面的 Country 和 Sales_Person 表处于选定状态：



- 4 在每个对话框中单击“确定”。
- 5 在 Web Intelligence 中运行查询以测试表限制。

6.6.18.2 何时使用每种方法来应用限制？

可以使用以下准则在 Universe 中设置限制：

- 避免在对象定义中使用 Where 子句。如果需要使用 Where 子句，您应意识到使用多个对象和 Where 子句相互冲突的潜在问题。
- 当您要通过提供预定义的条件来帮助用户时，请使用条件对象，并避免使用多个对象和对“Universe”窗格的类和对象视图进行更改。
- 如果不管在 SQL 中的何处使用表您都要应用限制，请使用自我限制的联接将限制应用到表。如果表使用标志在两个以上的域之间切换，则此方法十分理想。
- 当查找表在 Universe 中用作多种用途时，请使用附加联接。

6.6.19 拼接对象

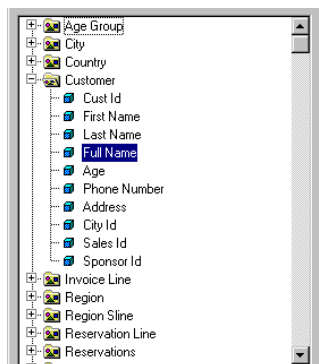
拼接对象是两个现有对象的组合。例如，您创建了对象“Full Name（全名）”，该对象是由“Customer（客户）”类中的对象“Last Name（姓氏）”和“First Name（名字）”拼接起来的。

6.6.19.1 创建拼接对象

创建拼接对象：

- 1 创建一个对象。

例如，用户在“客户”类中创建了新的对象“Full Name（全名）”。还应键入对象的说明，如“此对象由客户的姓氏和名字拼接而成。”



- 2 双击该对象。

将出现“编辑属性”对话框。

- 3 在“选择”框中键入拼接对象的语法。

例如，为“Full Name（全名）”对象键入以下语法（MS Access 语法）：

```
rtrim (Customer.first_name + ' ' + Customer.last_name)
```

其中，`rtrim` 是删除字符串结尾空格的函数，而两个引号用于在名字和姓氏之间插入一个空格。

名称 (N): Full Name

类型 (T): 字符

描述 (D): Returns a concatenation of the customer's first name and last name.

选择 (S): rtrim(Customer.first_name + ' ' + Customer.last_name)

条件 (C):

表 (B)... 分析 (A)

注意:

也可以单击“编辑”按钮打开 SQL 编辑器。可以使用编辑器中的图形工具来帮助您指定对象的 SQL 语法。有关此编辑器的更多信息，请参阅“设计模式”一章。

- 4 在每个对话框中单击“确定”。

在全名对象上运行查询时，会返回全名，按第一个名称的首字母顺序列出。

6.7 定义层次结构

可以创建对象层次以便允许用户执行多维分析。

6.7.1 什么是多维分析？

多维分析是对按一定意义排列的维对象所进行的分析。

多维分析允许用户从多种角度观察数据。这使得他们能够发现数据中的趋势或异常。

层次是一系列已排序的相关维。层次结构的一个示例是“Geography（地理）”，它可以将诸如“Country（国家）”、“Region（地区）”和“City（城市）”等维加以分组。

在 Web Intelligence 中，可以使用向上钻取或向下钻取来执行多维分析。

6.7.1.1 Drill

用户可以使用“钻取”来浏览明细的分层级别。用户可以在层次上“向上钻取”或“向下钻取”。

例如，经理想要跟踪一段时间内的预订情况。Universe 设计员可以设置一个“Reservation（预订）”时间层次结构，以包括“Reservation Year（预订年度）”、“Reservation Quarter（预订季度）”、“Reservation（预订月份）”和“Reservation Date（预定日期）”维。

经理可以从较高的聚合级别（例如，“Reservation Quarter（预订季度）”）向下钻取到更详细的级别，如“Reservation Month（预订月份）”或“ReservationDate（预订日期）”。他或她还可以从“Reservation Quarter（预订季度）”向上钻取到“Reservation Year（预订年度）”，以便查看更加概括性的数据视图。

6.7.2 标识层次结构的方法

层次可以采用不同的形式。传统层次的示例包括：

- 地理：洲 国家/地区 地区 城市
- 产品：类别 品牌 产品
- 时间：年度 季度 月份 周 日

层次也可能是“混合型”的，如以下层次：

地理/产品：洲 国家/地区 类别 品牌 产品

数据中隐含的层次取决于数据的特性和数据在数据库中的存储方式。您可能需要非常仔细的分析数据，以便在特定系统中找到最符合用户组的分析要求的层次。

尽管没有明确的规则用于确定数据位于层次中的何处，但数据库结构中固有的一对多（1-N）关系可以指明层次的存在。

在下面的模式中，表之间的一对多关系暗示了一个图形式层次。



6.7.3 设置层次结构

默认情况下，Universe 设计工具提供了一组供多维分析使用的默认层次。这些层次是类和对象，它们按在“Universe”窗格中出现的顺序排列。在创建对象时，应按层次排列对象，以便确保用户能够明白默认层次结构的意义。

通常需要创建包括来自不同类的对象的定制层次。在这些情况下，您需要创建新的层次。

可以查看默认值，并通过“层次编辑器”创建新的层次。层次编辑器是一种图形式编辑器，它允许您管理 Universe 中的层次。

注意：

定义自定义层次结构时，默认层次结构不再处于活动状态，并且无法供最终用户使用。如果要将其设置为活动状态，必须在“层次结构编辑器”中显式选择它们，并且将其添加到自定义层次结构列表。

6.7.3.1 查看层次

可按如下方式查看层次结构：

6.7.3.1.1 查看 Universe 中的层次

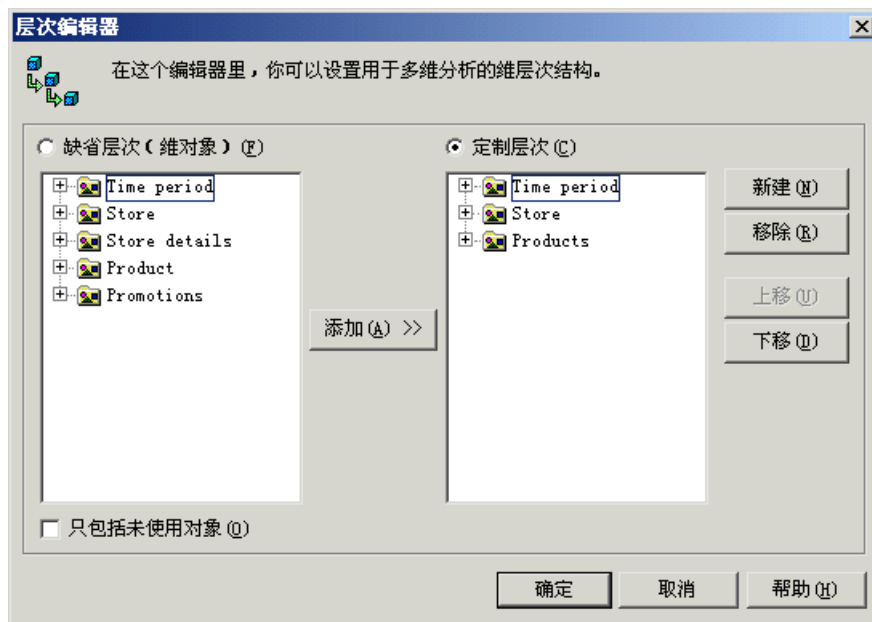
- 1 选择“工具”>“层次”。

或

单击“层次”按钮。

将出现层次编辑器。Universe 设计工具用文件夹符号代表层次，用立方体符号代表维。

左边的窗格列出了所有包含活动 Universe 中的维对象的类。右边的窗格显示您创建的所有定制层次。



- 2 单击层次节点 (+ 号) 以查看按层次排列的维。
- 3 单击“取消”。

6.7.3.2 设置层次

通过在“定制层次”窗格中创建一个新文件夹，然后按层次顺序添加相应的维，您可以创建新的层次。

通过选择层次或维并单击“删除”按钮，您可删除层次或维。

6.7.3.2.1 创建新的层次结构

- 1 从层次编辑器中，单击“新建”按钮。

或

从层次编辑器左边的窗格中选择一个类，然后将其拖到右边的窗格。

代表层次的文件夹即会出现在右边的窗格中。

- 2 键入层次名称。
- 3 按 RETURN 键应用该名称。
- 4 选择新的层次。

层次将突出显示。

- 5 展开左边窗格中的默认层次结构节点。

此层次包含您要添加到新的定制层次中的维。

- 6 单击一个维。要选择一系列维，请按住 CTRL 键并单击每个维。

一个或多个维将突出显示。

- 7 单击“添加”按钮。

一个或多个维即会出现在右边窗格中选定层次的下方。

注意：

如果要仅查看尚未选定要包括在层次中的维对象，则“仅未使用的对象”复选框十分有用。

6.7.3.3 重新排列维和层次的顺序

可以重新排列维对象在层次内的出现顺序。要移动某个对象，请单击该对象，然后单击“上移”或“下移”按钮。也可以用同样的方式重新排列层次的顺序。

您还可以通过拖放来移动维对象或层次。

层次和维对象的示例如下所示：



在上面的层次编辑器中，已经设置了三个定制层次：Time Period、Store 和 Products。“Products（产品）”层次由以下维组成：Lines（产品线）、Category（类别）、SKU desc（库存单位说明）、Color（颜色）和 Unit Price MSRP（建议零售单价）。

6.8 使用层次结构的级联值列表

可以将默认层次结构或定制层次结构与值列表（称为级联的值列表）相关联。

注意：

值列表（LOV）是包含与对象关联的数据值的列表。第 332 页上的 [“使用值列表”](#) 一节中详尽论述了值列表。

级联的值列表是与 Universe 中的层次结构关联的一系列值列表。目前已经为每个层次结构级别定义了提示，以便为级别返回值列表。

在刷新含有与级联值列表关联的层次结构的报表时会显示层次结构，并会提示先选择一个级别，再从值列表选择一个或多个值，然后才会运行查询。

例如，“Reservation Quarter（预订季度）”与“Year（年度）”层次结构关联。在查询中使用 Reservation Quarter 的月份时会显示 Year 层次结构，并提示用户为季度选择一个年度，然后才运行查询。

6.8.1 创建级联值列表

可以为默认层次结构或定制层次结构创建级联的值列表。会为每个级别创建一个 .LOV 文件。运行查询时，仅返回提示的层次结构级别的 LOV。

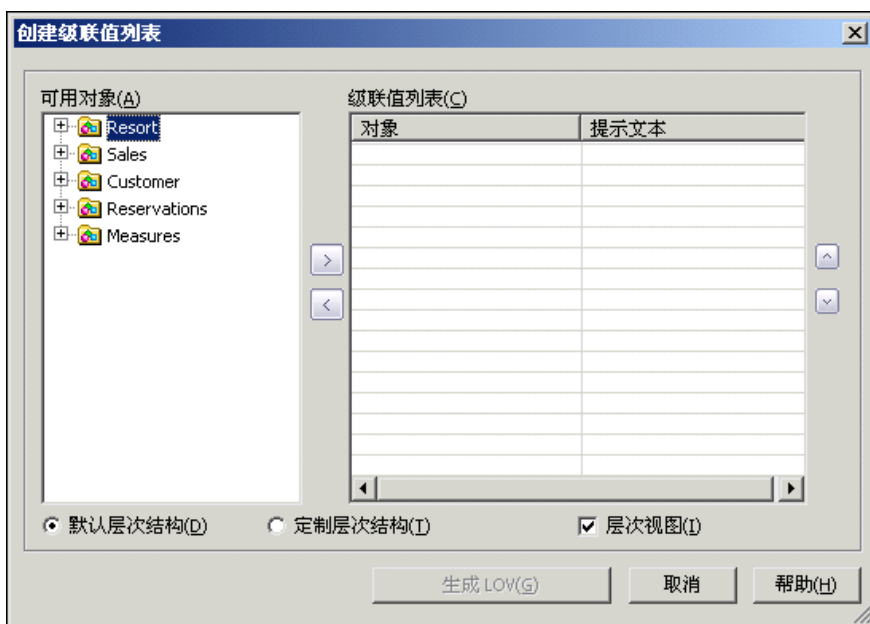
注意：

重复使用层次结构值列表不同于创建值列表层次结构，值列表层次结构的微多维数据集中将返回层次结构所有级别的所有值列表。如果使用级联值列表，在填写级别提示前，微多维数据集中不会返回 LOV，填写级别提示后，微多维数据集中仅返回该级别的 LOV。

6.8.1.1 创建级联值列表

1. 选择“工具” > “值列表” > “创建级联的值列表”。

将出现“创建级联的值列表”对话框。



可以选择以下选项：

级联的 LOV 选项	说明
默认层次结构 定制层次结构	如果选择一项，在 Universe 中定义的相应的默认层次结构或定制层次结构将出现在“可用”窗格中。有关这些层次结构类型的更多信息，请参阅第 326 页上的 “设置层次结构” 一节。
层次视图	如果选定此选项，“查询面板”的树视图中将显示层次结构。这样便于在层次结构中导航。单击级别时，值列表将出现在“查询面板”右边的窗格中。
对象	维的层次结构级别。
提示文本	级别值列表提示中出现的文本。

- 单击“默认层次结构”或“定制层次结构”单选按钮。

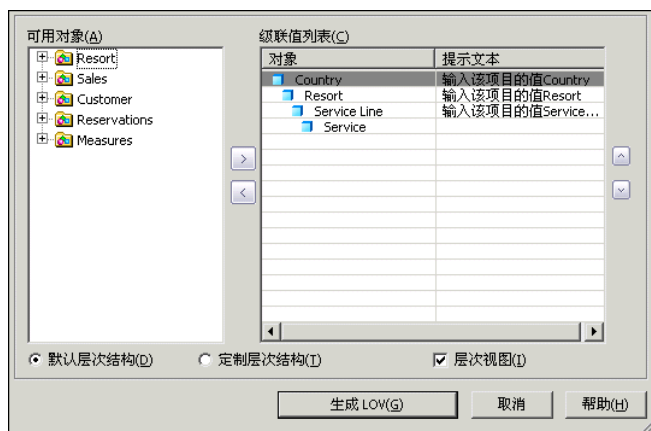
Universe 中可用的层次结构的相应列表将会出现。

- 单击或展开类，然后选择一个或多个对象。
- 单击向右箭头。

类中的所有对象将出现在“对象”列表中。

或

选定的对象将出现在“对象”列表中。



- 为每个对象键入提示文本。

- 6 如果要在“级联的值列表”中更改对象的位置，请单击该对象，并使用向上和向下箭头在列表中向上或向下移动它。

如果要删除对象，请单击对象，然后单击向左箭头。

- 7 选中或清除“层次视图”复选框。

- 8 单击“生成 LOV”。

“创建级联的值列表”对话框将被删除。将为级联值列表的每个级别创建一个 LOV。每个 .LOV 文件都保存在文件系统上的 Universe 子文件夹中，例如：C:\Documents and Settings\<用户>\Application Data\Business Objects\Business Objects 12.0\Universes\<CMS 名称>\beachXI3.0\。

注意：

有关编辑值列表、将值列表导出到 CMS 以及为个别对象创建值列表的信息，请参阅第 332 页上的[“使用值列表”](#)一节。

6.9 使用值列表

值列表是一种列表，它包含与对象关联的数据。值列表可以包含两种类型数据源中的数据：

值列表数据源	说明
数据库文件	<p>创建对象时，Universe 设计工具会自动将一列值与该对象关联。在用户或您（设计员）选择为“查询”窗格中的对象显示值列表之前，将不会创建值列表。然后，依据对象推导出的一个或多个列运行 SELECT DISTINCT 查询。</p> <p>返回的数据存储在 universe 子文件夹（在与存储 Universe 文件的相同文件夹下创建）中的一个扩展名为 .LOV 的文件中。该 .LOV 文件随后将用作列表的值来源。</p>
外部文件	<p>个人数据（例如文本文件或 Excel 文件）可与值列表关联。</p> <p>基于外部文件的值列表是固定的。无法与外部文件建立动态链接。如果更改了外部文件，您必须刷新 .LOV 文件。</p>

6.9.1 如何使用值列表？

在 Web Intelligence 中，用户可以使用操作数“显示值列表”在“查询”窗格中创建查询，以便在应用条件时应用对象。

注意：

在将任何条件应用于“查询”窗格中的对象（该对象要求在对象所推导出的列值上有所限制）时，还会创建一个 .LOV 文件。

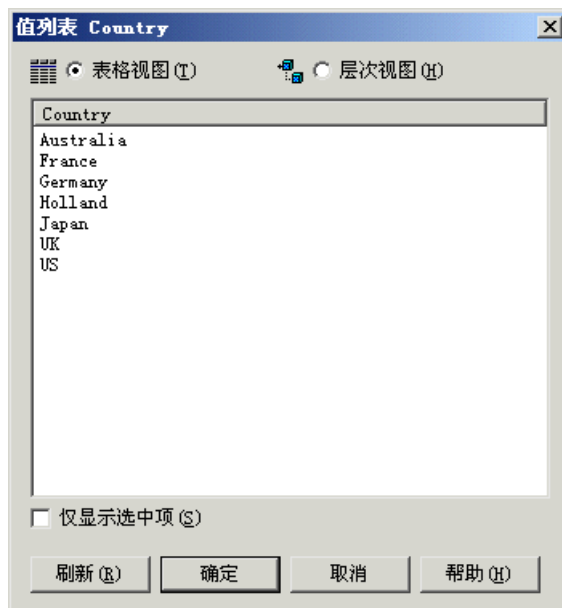
将出现对象的值列表并显示对象的可用值，从而允许用户为条件选择条款。第一次使用值列表时，它会以 .LOV 文件的形式保存在文件系统上的 Universe 子文件夹中。这使得 SELECT DISTINCT 查询只能为一个对象运行一次。

此文件夹还存储 Universe 设计工具中创建的 .LOV 文件，这些文件用于限制针对设计员想要控制其数据访问的对象返回的值列表。

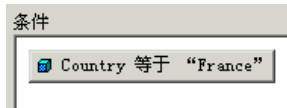
示例：为“Country（国家/地区）”使用值列表

名为 Country 的对象具有以下 Select 子句定义：COUNTRY.COUNTRY_NAME。与对象关联的默认值列表包含 COUNTRY_NAME 列中所有不同的国家/地区名称。在查询的条件中使用 Country 对象时，将返回此列表。

想要将查询中的值限制为仅为“France（法国）”的用户可以从以下列表中选择 France，该列表显示了条件的 Country 表中的所有国家/地区值：



在从列表中选择了 France 时，该条件将出现在“查询”窗格的“条件”窗格中，如下所示：



查询仅返回法国的值。

6.9.2 定义如何将值列表用于对象

当用户在 Universe 设计工具中创建维或明细对象时，它会自动分配一个关联的值列表。当您创建对象时，此列表实际上不存在，但是，默认情况下，在“查询”窗格中使用对象时，对象能够查询数据库来返回值列表。

注意：

没有为度量对象分配默认值列表。

当第一次在“查询”窗格中的对象上放置需要在 Universe 设计工具 中显示值列表的条件时，将依据对象所推导出的相应列运行 SELECT DISTINCT 语句，并返回值列表。

将自动在 universe 子文件夹中创建一个 .LOV 文件，以容纳值列表。下一次 Universe 设计工具中的对象需要值列表时，将从 .LOV 文件（而不是数据库）中返回值。

6.9.2.1 设计员在控制值列表时担任的角色

作为 Universe 设计员，您可以定义如何在列表中呈现数据，并定义对返回到列表的数据量和数据类型的限制。

可以设置对象的属性，以便确定值列表的以下操作：

- 如果值列表与对象关联。
- 在刷新列表时。
- 定义一个查询，该查询在对象用于返回值列表的 SELECT DISTINCT 查询中设置条件。将此查询保存在对象的属性中。
- 将列表值显示为简单的列表，或显示为对象层次。

- 如果列表基于列的值或外部文件（例如 Excel 电子表格）中的值。

也可以为对象创建永久性值列表，并将此列表导出到资源库。然后，将始终使用此 .LOV 文件作为该对象的值列表。它不会更新。

6.9.3 值列表属性和选项

可以定义以下对象属性，这些属性用于控制如何在 Web Intelligence 中使用对象的值列表。

属性	说明
关联值列表	<ul style="list-style-type: none"> • 选中时，允许将值列表与对象关联。默认情况下处于选定状态。 • 清除时，则没有值列表与对象关联。 • 对于维和明细，默认情况下处于选定状态。不会为度量选择此属性。
列表名称	对返回列表数据进行储存的 .LOV 文件的名称。限制为 8 个字符。
允许用户编辑此值列表	<ul style="list-style-type: none"> • 选中时，用户能够编辑 Web Intelligence 中的值列表文件。 • 清除时，用户无法编辑此列表。 <p>注意： 此属性不适用于诸如 Excel 电子表格等个人数据文件。这些文件不会导出到资源库。它们保留在本地计算机上。用户能够编辑本地文件，或更改另一个本地数据文件的目标值列表。</p> <p>值列表通常用于限制可供用户使用的值。如果用户可以编辑列表，您就不再能够控制他们选择的值。通常，如果不使用个人数据文件作为值列表来源，请清除此选项以确保用户不会编辑值列表。</p>
使用之前自动刷新（仅限 BusinessObjects）	<ul style="list-style-type: none"> • 如果选中，则每次在“查询”窗格中显示对象的值列表时，都会刷新列表数据。每次刷新 .LOV 时，这可能会对性能产生影响。此选项不适用于 Web Intelligence 报表。 • 如果清除，则只会在用户登录会话开始时刷新列表一次。 <p>如果列表包含定期更改的值，您可以选择此选项，但应考虑性能影响。</p> <p>如果列表内容保持不变，则应清除此选项。</p>

属性	说明						
分层显示	如果选中“分层显示”属性，则在 Web Intelligence 中按层次结构显示级联值列表。						
随 Universe 导出	<ul style="list-style-type: none">如果选中，则将与对象关联的 .LOV 文件随 Universe 一起导出到资源库。必须创建与对象关联的值列表，以便将其导出。此列表将保存为 .LOV 文件。如果清除，则不将对象的 .LOV 文件导出到资源库。 如果定期定制此列表，请选择此选项。这允许您随 Universe 一起导出和导入您所做的修改。						
委派搜索	<p>如果选中“委派搜索”属性，则允许 Web Intelligence 用户限制值列表中返回的值数。当选中“委派搜索”属性时，Web Intelligence 在查询运行时向用户显示一个空的值列表框。用户可以输入一个值以定义用于过滤值列表的搜索条件。</p> <p>许多数据源支持使用通配符来加快数据库搜索。Web Intelligence 支持以下通配符：</p> <table><tr><td>*</td><td>匹配任意个字符，包括零个字符</td></tr><tr><td>?</td><td>只匹配一个字符</td></tr><tr><td>\</td><td>将下一个字符换码，用于搜索通配符</td></tr></table> <p>“委派搜索”选项受到以下限制：</p> <ul style="list-style-type: none">级联值列表不支持“委派搜索”。只能为“字符”类型的对象的值列表激活“委派搜索”。如果为值列表输入自定义的 SQL，则不会激活“委派搜索”。当为值列表使用“分层显示”选项时，则不会激活“委派搜索”。	*	匹配任意个字符，包括零个字符	?	只匹配一个字符	\	将下一个字符换码，用于搜索通配符
*	匹配任意个字符，包括零个字符						
?	只匹配一个字符						
\	将下一个字符换码，用于搜索通配符						

通过单击以下按钮，您可以编辑、显示默认名称或将其分配给值列表：

选项	说明
恢复默认值	在创建对象时恢复分配给 .LOV 文件的默认名称。
编辑	允许您编辑列表中显示的值。当在“查询”窗格中使用时，可以使用编辑器来限制列表中显示的值。
显示	显示对象的值列表。在创建要随 Universe 一起导出到资源库的永久性列表时，必须单击“显示”以创建 .LOV 文件。可以随后编辑该文件。

6.9.3.1 定义值列表的属性和选项

定义值列表（.LOV）文件的属性和选项：

- 1 双击一个对象。
“编辑属性”对话框即会打开到“定义”页面。
- 2 单击“属性”选项卡。
将出现“属性”页面。
- 3 选择或清除页面底部值列表组框中的复选框。
- 4 在“列表名称”框中键入关联 .LOV 文件的名称。
- 5 如果要在列表值上定义限制，请单击“编辑”按钮。
- 6 使用“查询”窗格在列表数据上创建查询。
- 7 单击“显示”按钮以查看值列表。
当您单击此按钮时，将依据数据库中的对象所推导出的列运行 SELECT DISTINCT 查询。此方法与报表产品中用于为对象创建 .LOV 文件的方法相同。
- 8 单击“确定”。

6.9.3.2 查看与对象关联的值列表

在 Universe 设计工具中，可以查看与对象关联的值列表。查看值列表时，会自动在 UserDocs 目录中创建默认的 .LOV 文件，用以保存返回数据。默认情况下，查看值列表时，会自动创建 .LOV 文件。

可以按列表格式或对象层次格式查看值列表。

查看值列表：

- 1 双击一个对象。

“编辑属性”对话框即会打开到“定义”页面。

- 2 单击“属性”选项卡。

将出现“属性”页面。

- 3 单击“显示”按钮。

“值列表”对话框显示与对象关联的所有可能的数据值。



- 4 单击“取消”。

6.9.3.3 创建值列表

可以按如下方式创建值列表：

- 1 查看对象的值列表。

- 2 单击“确定”。

Universe 设计工具将值列表 (.LOV) 文件存储在包含 Universe 文件的相同文件夹中的 universe 子文件夹内。子文件夹的名称与用来创建 .LOV 的对象所在的 Universe 相同。

.LOV 文件创建完成后，就可以编辑列表，限制返回到 .LOV 文件的数据，或修改数据在列表中的呈现方式。

6.9.4 编辑值列表

可以使用两种方式修改值列表的内容：

- 将条件应用于生成列表的 SELECT DISTINCT 查询。例如，您可以将“休假地”对象值列表中的休假地限制为拥有多于最少保留游客数的休假地。
- 创建一个层次，以便为用户简化从列表中选择值的过程。如果列表包含大量的值，此方法可能非常有用。

6.9.4.1 将条件应用于值列表

将条件应用于值列表：

- 1 双击一个对象。
将出现对象“编辑属性”表单。
- 2 单击“属性”选项卡。
将出现“属性”页面。
- 3 选择“关联值列表”复选框。
- 4 如果要重命名列表，请在“列表名称”框中键入 .LOV 文件的名称。

☒ 关联值列表 (L)

列表 (N) :
Cust_FR

☐ 允许用户编辑此列的值 (U)

☐ 使用前自动刷新 (R)

☒ 与 Universe 一起导出 (X)

恢复缺省值 (R) 编辑 (E)... 显示 (V)...

- 5 单击“编辑”按钮。

将出现“查询”窗格。活动对象列在“结果对象”窗格中。

- 6 将想要在活动对象的值列表上充当条件的对象拖到“条件”窗格中。
- 7 双击“运算符”窗格中的一个运算符。
- 8 双击“操作数”窗格中的一个操作数。
- 9 根据需要选择或键入值。

例如，以下查询仅返回来自法国的客户。



- 10 单击“确定”。
- 11 单击“显示”以查看受限制的值列表。

将出现一个空白列表。
- 12 单击“刷新”。
- 13 值即会出现在列表中。

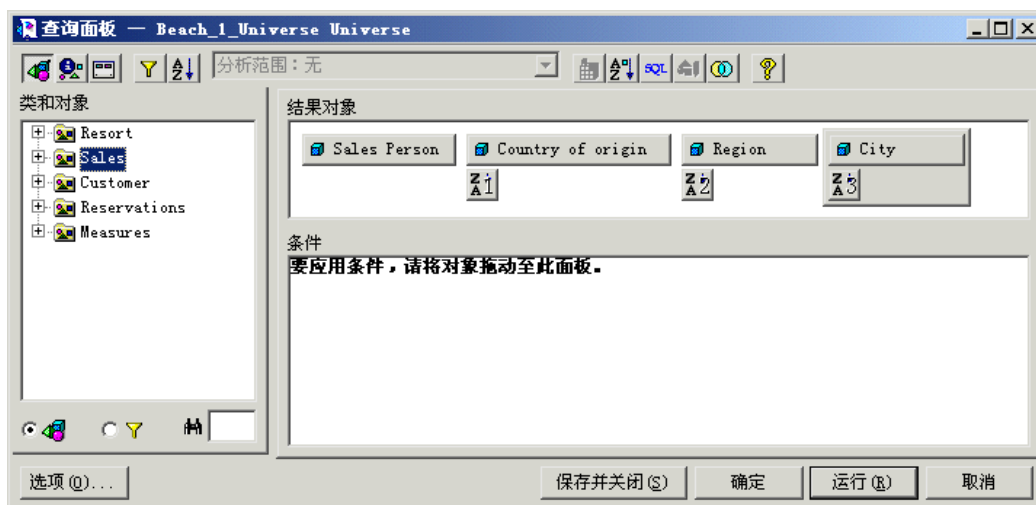


14. 在每个对话框中单击“确定”。

6.9.4.2 为值列表创建层次

为值列表创建层次：

1. 双击一个对象。
将出现对象“编辑属性”表单。
2. 单击“属性”选项卡。
将出现“属性”页面。
3. 选择“关联值列表”复选框。
4. 如果要重命名列表，请在“列表名称”框中键入 .LOV 文件的名称。
5. 单击“编辑”按钮。
将出现“查询”窗格。活动对象列在“结果对象”窗格中。
6. 将想要放在层次中的对象拖入现有对象右边的“结果对象”框，如下所示：



- 7 单击“确定”。
- 8 单击“显示”以查看受限制的值列表。

将出现一个空白列表。

- 9 单击“刷新”。

值即会出现在列表中。



- 10 在每个对话框中单击“确定”。

6.9.5 导出值列表

可以将值列表随 Universe 一起导出到 CMS。在文件系统上，关联的 .LOV 文件会被复制到与存储 Universe 文件的相同文件夹中的 universe 子目录。

6.9.5.1 如何在 Web Intelligence 中使用导出的 .LOV?

如果用户在 Web Intelligence 中运行查询时使用的对象与从 Universe 设计工具中导出的 .LOV 文件关联，为对象返回的值列表由以下各项之一确定：

- .LOV 文件中包含的数据。
- .LOV 文件中所定义 SELECT DISTINCT 查询的 SQL。

如果在 Universe 设计工具中创建了条件来限制为对象返回的数据值，则会出现限制列表，而不是所有数据值的默认列表。此列表将保留在 Universe 设计工具中实现的所有条件和格式设置。

如果未随 Universe 一起导出 .LOV 文件，则对象将只返回没有条件和格式设置的缺省列表。然后，将创建一个缺省 .LOV 文件来容纳数据。

6.9.5.2 导出包含或不包含数据的列表

可以采用两种方式将值列表导出到中央管理服务器（CMS）资源库：

导出 .LOV...	说明
仅包含查询定义（无数据）	将随 SELECT DISTINCT 查询的定义一起导出 .LOV 文件，以便将值返回到列表。在 Universe 设计工具“查询”窗格中为 .LOV 设置的所有条件将会保留。.LOV 文件不包含数据，并且，当第一次在“查询”窗格中使用对象来返回值时，将会填充该文件。对于定期更新的数据，或者在值列表可能非常庞大的情况下，您应使用此方法。
包含数据	当在 Universe 设计工具中显示或编辑值列表时，将随返回的所有数据一起导出或导入 .LOV 文件。如果 .LOV 中的数据不更改，这可能十分有用。但是，如果数据定期更新，或者列表包含大量的值，则不应将数据随 .LOV 一起导出，因为它可能会减慢导出过程的速度。

导出值列表定义

导出值列表定义（无数据）：

- 1 为对象创建值列表。
- 2 在对象的“属性”页上选择“随 Universe 导出”复选框。

下面的值列表 Cust_FR 与 Customer 关联，以便仅返回法国客户的值。

- 3 选择“工具”>“值列表”。

将出现“值列表”对话框。它列出了当前 Universe 中的类和对象，并包含用于管理每个对象的值列表的选项。

- 4 展开一个类，并选择带有要导出到资源库的关联 .LOV 文件的对象。



- 5 单击“清除”按钮。

即会从对象的 .LOV 文件中删除数据。.LOV 文件现在仅包含值列表的查询定义。

- 6 单击“确定”。

- 7 选择“文件”>“导出”。

将出现“导出 Universe”框。

- 8 从 Universe 列表中选择 Universe 文件名。

- 9 单击“确定”。

将出现一个消息框，通知您已成功导出了 Universe。

导出包含数据的值列表

导出包含数据的值列表：

- 1 为对象创建值列表。

- 2 在对象的“属性”页上选择“随 Universe 导出”复选框。

- 3 单击“显示”按钮。

此时将出现值列表。

- 4 如果列表为空，请单击“刷新”按钮来填充列表。

- 5 在每个对话框中单击“确定”。

- 6 选择“文件”>“导出”。

将出现“导出 Universe”框。

- 7 从 Universe 列表中选择 Universe 文件名。

- 8 单击“确定”。

将出现一个消息框，通知您已成功导出了 Universe。

6.9.6 刷新值列表中的值

可以使用两种方法在 Universe 设计工具中刷新值列表中的数据：

- 显示对象的值列表，然后单击“刷新”按钮。
- 选择“工具”>“值列表”，以便显示“值列表”管理框，选择对象并单击“刷新”按钮。

6.9.7 使用个人数据文件中的数据

可以将值列表分配给包含从数据库服务器中检索的个人数据（而不是公司数据）的对象。

个人数据是存储在平面文件（如文本文件）中的数据或来自以下应用程序之一的数据：Microsoft Excel、Lotus 1-2-3 或 dBASE。

使用个人数据文件作为值列表具有以下优势：

- 从个人数据文件中检索数据可能比访问公司数据库速度更快。
- 用户需要这些数据库中不存在的值。
- 您可以控制用户在使用值列表时所看到的值。

使用个人数据文件的缺点在于：数据是固定的。如果需要更改值，您必须手动更新数据。

6.9.7.1 通过个人数据文件创建值列表

通过个人数据文件创建值列表：

1. 选择“工具”>“值列表”。

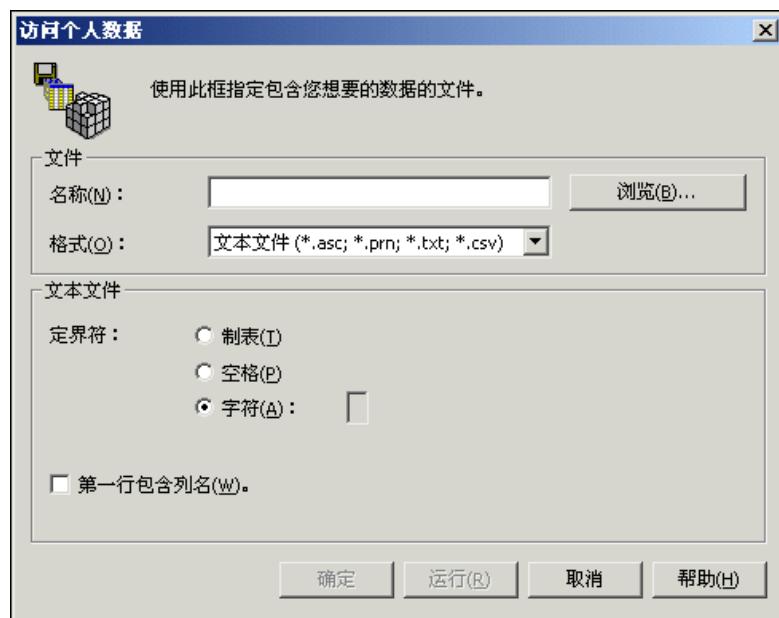
将出现“值列表”对话框。

- 2 展开某个类并单击对象。
- 3 单击“属性”组框中的“个人数据”单选按钮。

一个消息框告知您：将要把值列表类型从公司更改为个人。

- 4 单击“确定”。

将出现“访问个人数据”对话框。可用的选项取决于您选择的文件类型。



- 5 单击“浏览”按钮，并选择要用作值列表的文件。

或

在“名称”文本框中键入文件名。

- 6 从“格式”列表框中选择文件格式。
- 7 可以选择以下文件格式之一：
 - 文本文件 (*.asc; *.prn; *.txt; *.csv)
 - Microsoft Excel 文件
 - dBASE
 - Microsoft Excel 97。

注意：

如果文件是在 Excel 97 中创建的，您必须使用“Microsoft Excel 97”选项，而不是“Microsoft Excel 文件”选项。

- 8 根据需要指定剩余的选项。

在文本文件中，一行就相当于一个行。对于文本文件，请指明列分隔符的类型：制表符、空格或字符。如果选择字符作为类型，请在文本框中输入字符。

- 9 单击“确定”。

6.9.8 管理 Universe 中的值列表

可以通过“值列表”对话框（“工具”>“值列表”）管理活动 Universe 中的所有值列表。所有类和对象都呈现在树形视图中。可以选择任何对象并访问其值列表。可通过“值列表”对话框执行以下操作：

选项	说明
编辑	显示用于为选定对象定义查询的“查询”窗格。可以定义和编辑值列表的现有查询。
显示	显示选定对象的当前值列表。
清除	清除当前分配给选定对象的值列表的内容。
刷新	刷新值列表的显示。

6.9.8.1 访问值列表管理工具

访问值列表管理工具：

- 1 选择“工具”>“值列表”>“编辑值列表”。

将出现“值列表”对话框。



- 2 展开某个类并选择对象。
- 3 单击按钮或选择选项以执行管理任务。
- 4 单击“确定”。

6.9.9 优化和定制 LOV 文件

用于优化和定制 LOV 的某些常用方法如下所示：

方法	说明
将 LOV 指向较小的表	缺省情况下，LOV 指向与本身所附加对象相同的对象。但是，如果此对象指向较大的表（行数），则 LOV 的刷新速度可能较慢。如果有更小或更快的备用表（返回相同的值），则应编辑 LOV 以便指向该备用表。

方法	说明
合并代码和说明	自定义 .LOV 的一种典型方法是合并“代码”和“说明”。对象会返回“销售类型代码”，但该代码对于某些用户来说并没有有意义的值。如果编辑 LOV 以显示“销售类型说明”，则会在用户查看 LOV 时对他们大有帮助。也可以对“销售类型说明”对象执行相反的操作，以便与说明一起显示代码。

6.10 链接 Universe

可以动态链接一个或多个 Universe。

6.10.1 什么是链接的 Universe？

链接的 Universe 是共享公共元素（如参数、类、对象或联接）的 Universe。

链接两个 Universe 时，一个 Universe 的角色为核心 Universe，而另一个为派生 Universe。对核心 Universe 进行更改时，这些更改会自动传播到派生 Universe。

注意：

有关部署链接的 Universe 的信息，请参阅第 359 页上的 [“派生 Universe 和值列表”](#) 一节。

6.10.1.1 什么是核心 Universe？

核心 Universe 是其他 Universe 所链接到的 Universe。它包含链接到它的其他 Universe 所共有的元素。这些 Universe 称为派生 Universe。核心 Universe 表示可重用的元素库。

核心 Universe 可以是内核或主 Universe，具体取决于在派生 Universe 中使用核心 Universe 元素的方式。第 355 页上的[“在两个 Universe 之间创建链接”](#)一节中说明了内核和主 Universe。

6.10.1.2 什么是派生 Universe?

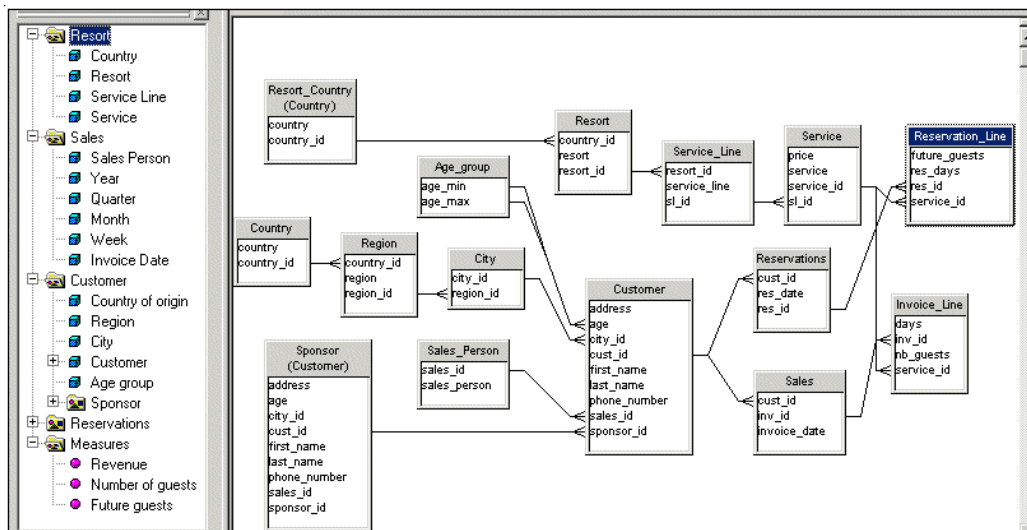
派生 Universe 是一种包含指向核心 Universe 的链接的 Universe。此链接使派生 Universe 可以共享核心 Universe 的公共元素：

- 如果链接的核心 Universe 是内核 Universe，则可以将元素添加到派生 Universe。
- 如果链接的核心 Universe 是主 Universe，则派生 Universe 将包含所有核心 Universe 元素。不会将类和对象添加到派生 Universe 中。可以在派生 Universe 中隐藏类和对象，这取决于目标用户的用户需求。

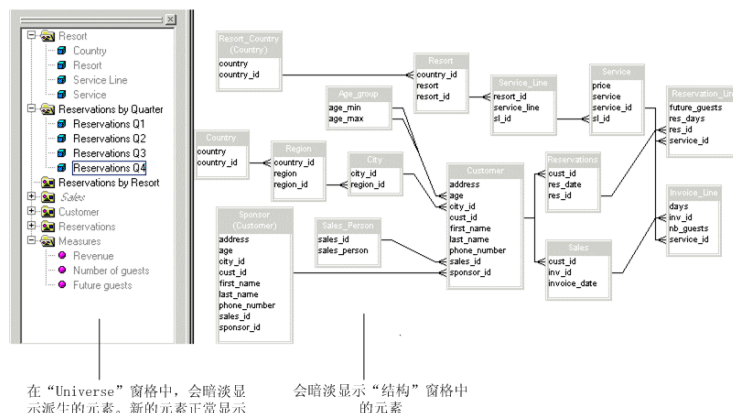
示例：链接的核心 Universe 和派生 Universe

下例显示了两个链接的 Universe：一个是包含公共元素的核心 Universe，另一个是使用核心结构的派生 Universe，但还具有特定于它自己的新类和对象。

Beach.unv 是核心 Universe。Island Resorts 的销售经理使用它执行营销分析。此 Universe 是随本版本附带的其中一个演示 Universe。此 Universe 的内容如下所示：



通过使用此核心 Universe，该经理创建了一个派生 Universe（侧重于预订情况）。



从核心 Universe 中派生元素会变暗。经理已经创建了两个新的类：“Reservations by Quarter（按季度列出的预订）”和“Reservations by Resort（按度假地列出的预订）”。这些类和它们的对象将正常显示。该经理还选择隐藏“Sales（销售）”类（在 Reservation（预订）Universe 中不需要此类）。对核心 Universe 元素所做的任何更改会自动传播到派生 Universe。

6.10.2 链接 Universe 的不同方法

在链接 Universe 时可以使用以下任何方法：

- 内核方法
- 主方法
- 组合方法

可以逐一使用上述三种方法，或联合使用上述一种以上的方法。

6.10.2.1 内核方法

在使用内核方法的情况下，一个 Universe 包含核心元素。这些元素是所有 Universe 都共有的元素。您通过此内核 Universe 创建的派生 Universe 包含这些核心元素和它们自己的特定元素。

您对此内核 Universe 所做的任何更改会自动反映在所有派生 Universe 的核心元素中。

6.10.2.2 主方法

主方法是另一种组织方法，用于组织链接的 Universe 的公共元素。

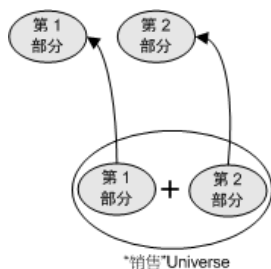
主 Universe 包含所有可能的元素。在从主 Universe 派生 Universe 中，某些元素会被隐藏，具体取决于它们与派生 Universe 的目标用户的相关性。

在派生 Universe 中可见的元素始终是主 Universe 的子集。没有特定于派生 Universe 添加的新元素。下面的示例显示，“人力资源”和“销售”这两个 Universe 都是从主 Universe 中派生的。它们包含此主 Universe 的派生（某些派生可能被隐藏）。

您对此主 Universe 所做的任何更改会自动反映在所有派生 Universe 的核心元素中。

6.10.2.3 组合方法

组合方法涉及到将两个或更多个 Universe 合并到一个 Universe 中。下面的“销售” Universe 是通过合并两个 Universe（第 1 部分和第 2 部分）而创建的。



6.10.3 链接 Universe 的优点

链接 Universe 有以下优点：

- 减少了开发和维护时间。修改核心 Universe 中的某个元素时，Universe 设计工具会将更改传播到所有派生 Universe 中的同一元素。
- 可以将常用的元素集中到一个核心 Universe 内，然后将它们包含在所有新的 Universe 中。您无需在每次创建新的 Universe 时重新创建公共元素。
- 促进了专业化。可以在数据库管理员（建立基本的核心 Universe）和更专业的设计员（根据他们的特定领域创建更实用的 Universe）之间划分开发任务。

6.10.4 链接 Universe 的要求

只有在满足以下要求时，才能将活动的 Universe 链接到核心 Universe：

- 核心 Universe 和派生 Universe 使用相同的数据帐户或数据库和相同的 RDBMS。将相同的连接用于核心 Universe 和派生 Universe 使用户更容易管理 Universe，但可以随时改变这一做法。
- 核心 Universe 和派生 Universe 必须位于同一资源库中。
- 至少导出和重新导入核心 Universe 一次。在创建链接前，无需将派生 Universe 进行导出。
- 导出的派生 Universe 与核心 Universe 位于同一个 Universe 域中。
- 您被授权链接指定的 Universe。

6.10.5 链接 Universe 时的限制

在链接 Universe 时，您需要了解以下限制：

- 不能链接到使用存储过程的 Universe
- 只能使用一种链接级别。不能从自我派生 Universe 中创建派生 Universe。
- 所有类和对象在核心 Universe 和派生 Universe 中均保持唯一。否则，将会发生冲突。
- 两个 Universe 结构必须允许在这两个 Universe 中的两个表之间创建联接。否则，会在利用这两个结构中的对象运行查询时产生笛卡儿积。

- 在派生 Universe 中，只有核心 Universe 的表模式、类和对象可用。必须在派生 Universe 中重新检测上下文。
- 将派生 Universe 与核心 Universe 的结构一起导出时，不会保存与核心 Universe 关联的值的列表。

6.10.6 在两个 Universe 之间创建链接

可以将一个活动的 Universe 链接到另一个 Universe。这样做时，活动的 Universe 将成为派生 Universe，而链接的 Universe 将成为核心 Universe。派生 Universe 继承核心 Universe 中的元素。

要将 Universe 链接到核心 Universe，必须已将核心 Universe 导出到资源库。

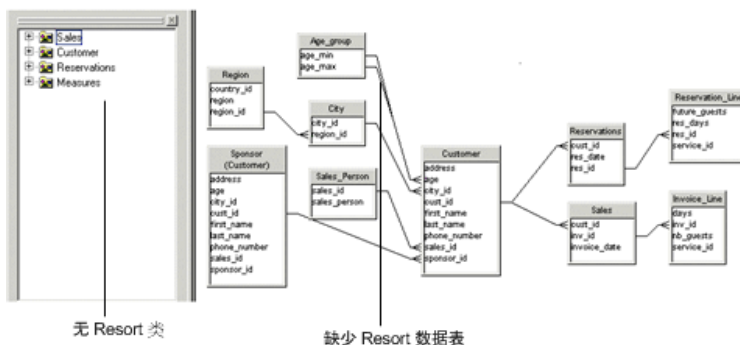
注意：

在链接 Universe 时，您可以在同一资源库内重新定位核心 Universe，而不会中断链接。这样，您将能够将核心 Universe 导出到其他资源库文件夹，并同时保持与派生 Universe 的链接有效。

6.10.6.1 在派生 Universe 和核心 Universe 之间创建链接：

- 1 确保活动的 Universe 是您想链接到核心 Universe 的 Universe。

例如，以下的 Universe 是 Beach Universe 的一个版本，它只包含国家/地区的销售信息，而不包含休假地数据。您想将此销售 Universe 与包含休假地数据的休假地 Universe 进行链接。以下的销售 Beach Universe 是派生 Universe，而 Resort Universe 是核心 Universe。



- 2 选择“编辑”>“链接”。

“Universe 参数”对话框即会打开到“链接”页面：



- 单击“添加链接”按钮。

将出现“要链接的 Universe”对话框。它列出了可用域中的 Universe。

- 浏览到您想链接的 Universe。这是核心 Universe，此 Universe 包含要在活动 Universe 中使用的元素。

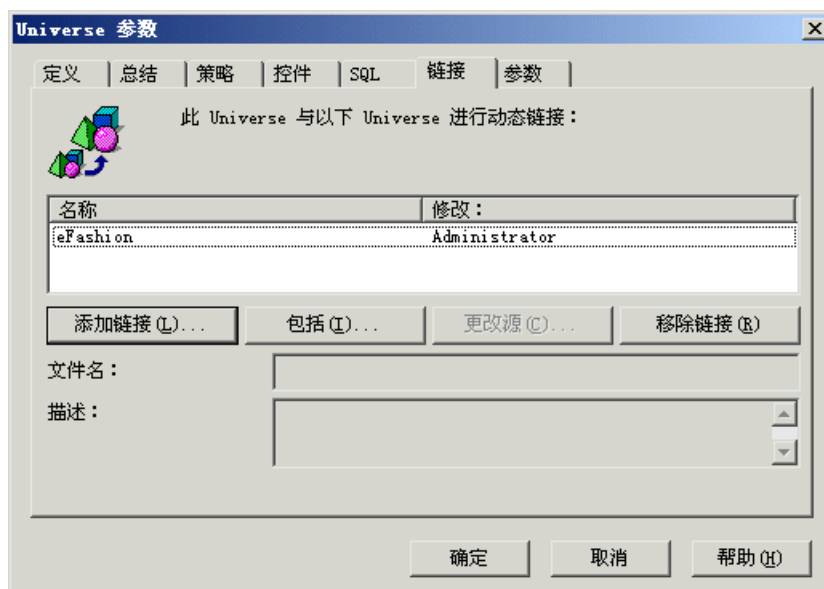
在本例中，您选择休假地 Universe。



如果从未导出过所选的 Universe，则您会收到错误消息。必须先导出 Universe，然后才能链接它。

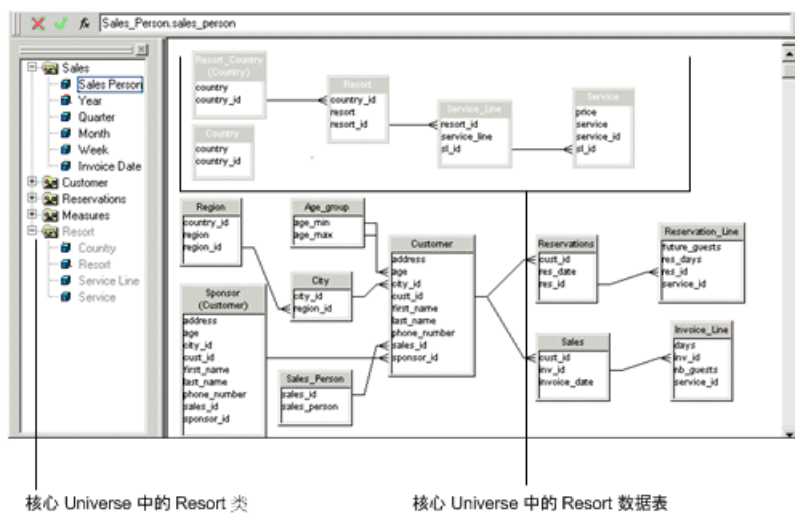
- 单击“打开”按钮。

所选的 Universe 即会出现在列表中。



6 单击“确定”。

将创建链接。活动的 Universe 中的核心元素将变暗。



6.10.7 编辑派生 Universe

通过在核心表和 Universe 的派生表之间创建联接来完成链接过程。必须删除所有当前上下文，然后为新的结构重新检测上下文。

注意：

不能在派生 Universe 中编辑来自链接的 Universe（核心 Universe）的任何结构、类或对象。

6.10.7.1 编辑派生 Universe

编辑派生 Universe：

- 1 在核心 Universe 的结构和派生 Universe 的结构之间创建联接。
创建联接确保了不会为包含在这两个结构的查询中的对象返回笛卡儿积。
- 2 删除现有上下文。
- 3 检测别名。
- 4 检测上下文。
- 5 根据需要隐藏或创建新的对象。

注意：

有关隐藏元素的信息，请参阅第 278 页上的 [“显示或隐藏类、对象和条件”](#) 一节。

6.10.8 删除链接

只有在派生 Universe 不包含基于核心元素的对象或与核心元素的联接时，才能删除指向核心 Universe 的链接。

6.10.8.1 删除派生 Universe 中的链接

删除派生 Universe 中的链接：

- 1 打开派生 Universe。
 - 2 选择“编辑”>“链接”。
- 将出现“Universe 参数”对话框的“链接”页面。
- 3 在列表中单击核心 Universe 的名称。
 - 4 单击“删除链接”按钮。
 - 5 单击“确定”。
- 将从活动的 Universe 中删除核心 Universe 中的元素。

6.10.9 重新放置核心 Universe

如果核心 Universe 的位置已改变，则需要指明新的位置，以便保持链接。

6.10.9.1 将链接更新到重新放置的核心 Universe

将链接更新到重新放置的核心 Universe：

- 1 打开派生 Universe。
 - 2 选择“编辑”>“链接”。
 - 3 在列表中单击链接的核心 Universe。
 - 4 单击“更改源”按钮。
- 将出现“要链接的 Universe”对话框。
- 5 浏览到核心 Universe 的新位置。
 - 6 单击“打开”按钮。
- 新的核心 Universe 将出现在“链接”列表中。

6.10.10 派生 Universe 和值列表

在将派生 Universe 导出到资源库时，不会随派生 Universe 一起保存与核心对象关联的值的列表。

以下说明了可用来保存与核心对象关联的值列表的一种方法：

1. 使用与包含以下值列表的对象相同的定义创建新的对象：要随派生 Universe 一起导出到资源库的值列表。
2. 为新的对象分配与核心对象相同的值列表。
3. 隐藏这些新的对象。

隐藏的对象的作用是保存值列表，这样就能随派生 Universe 一起导出和导入这些值列表。

6.10.11 按核心 Universe 的顺序显示对象

缺省情况下，排列派生 Universe 的对象时所依从的顺序就是 Universe 的用户将看到的顺序，即使此顺序后来在核心 Universe 中改变了也是如此。如果希望派生 Universe 始终按照对象在核心 Universe 中显示的顺序显示对象，则必须相应地在所使用的数据库的 *.PRM 文件中设置一个参数。

此参数设置是 `CORE_ORDER_PRIORITY = Y`。

有关如何在相关的 *.PRM 文件中设置参数的详细信息，请参阅《数据访问指南》（“帮助” > “数据访问指南”）。

6.11 在一个 Universe 中包含另一个 Universe

可以将核心 Universe 的元素复制到派生 Universe。派生 Universe 中的最终元素独立于核心 Universe 中的那些元素。这些元素未链接到核心 Universe。对核心 Universe 所做的任何更改都不会由派生 Universe 继承。

6.11.1 将核心 Universe 复制到派生 Universe 中

将核心 Universe 复制到派生 Universe 中时，派生 Universe 中的最终圆熟独立于核心 Universe 中的那些元素。这些元素未链接到核心 Universe。对核心 Universe 所做的任何更改都不会由派生 Universe 继承。

将核心 Universe 复制到派生 Universe 中的原因有以下两个：

- 将指定的 Universe 的内容复制到活动的 Universe 中。
- 不再保持两个 Universe 之间的动态链接。

注意：

如果在复制操作之前链接了两个 Universe，则复制过程会删除动态链接，而活动 Universe 中的元素将不再动态链接到外部 Universe。

6.11.1.1 将核心 Universe 复制到派生 Universe 中

将核心 Universe 复制到派生 Universe 中：

- 1 打开一个 Universe。
- 2 选择“编辑”>“链接”。

将出现“Universe 参数”对话框的“链接”页面。

- 3 单击“添加链接”按钮。

将出现“要链接的 Universe”对话框。它列出了可用域中的 Universe。

- 4 浏览到并选择您想复制的 Universe。这是核心 Universe，此 Universe 包含要在活动 Universe 中使用的元素。
- 5 单击“包含”按钮。
- 6 单击“确定”。

核心 Universe 中的元素将显示在活动的 Universe 中。

6.12 创建存储过程 Universe

存储过程 Universe 是一种特殊的 Universe，Web Intelligence 用户使用它来访问数据库中的存储过程。这是 Web Intelligence 用户可用来访问存储过程的唯一方式。Web Intelligence 用户可以使用这些存储过程 Universe 创建基于数据库中存储过程的报表。

存储过程是一个已编译的 SQL 程序，由一个或多个 SQL 语句构成，位于目标数据库中并在目标数据库中运行。

如果 Desktop Intelligence 创建的报表基于存储过程，Web Intelligence 将无法打开这些报表。这意味着 Web Intelligence 用户必须使用特定的存储过程 Universe 来访问存储过程和创建报表。

存储过程提供以下优点：

- 存储过程会封装代码。数据库操作只在存储过程中出现一次，而且在整个应用程序源代码中也不会多次出现。这样可以提高应用程序的调试性能和可维护性能。
- 对数据库模式所做的更改将只影响一处源代码，即存储过程，所以模式更改工作成为一项数据库管理任务，不再涉及代码版本。
- 由于存储过程驻留在服务器中，因此用户可以设置更严格的安全限制，从而为本身已得到妥善保护的存储过程保存更可信的权限。
- 由于存储过程是在客户端应用程序之外进行编译和存储的，因此可以在 SQL 语法中使用敏感度更高的变量，例如密码或个人数据。
- 使用存储过程可降低网络通信流量。

在 BusinessObjects XI Release 3.0 中，存储过程可以与 Desktop Intelligence 以及与 Universe 设计工具中的 Universe 一起使用。如果用户在 Universe 中包含用于 Crystal Reports 和 Web Intelligence 的存储过程，将会受益匪浅。

请注意，存储过程 Universe 受以下限制：

- 不允许在存储过程 Universe 中的对象之间使用联接。
- 不能对存储过程 Universe 使用过滤器。
- 无法将存储过程 Universe 链接至标准 Universe。
- Web Intelligence 管理员授予对存储过程所在的数据库或帐户的访问权限。
- 并非所有 RDBMS 都支持存储过程。请查阅数据库指南，以便了解您的 RDBMS 是否支持存储过程。
- 不会执行存储过程中包含的 COMPUTE、PRINT、OUTPUT 或 STATUS 语句。

有关在报表中使用存储过程的更多信息，请参见《Desktop Intelligence 指南》。

6.12.1 Java Bean Universe 中的存储过程

BusinessObjects XI Release 3.0 支持依据 Java Bean 创建 Universe。基于 Java Bean 的 Universe 返回一个结果集，以构建 Universe 实体关系。

基于 Java Bean 的 Universe 与基于存储过程的 Universe 使用相同的工作流，带来的益处相同，受到的限制也相同：

- 不允许使用联接
- 不能对 Universe 使用过滤器

有关如何访问 Java Bean 的更多信息，请参阅《数据访问指南》。

6.12.2 根据存储过程创建 Universe

Business Objects 支持存储过程：

- 不带参数
- 带参数 (IN)
- 带多个结果集
- 带多个语句（多个不同于 SELECT 的 SQL 语句）

要创建带参数的存储过程，请单击“快速设计向导”，然后按照“根据带参数的存储过程创建 Universe”中所述的步骤操作。

相关主题

- 第 363 页上的[“使用存储过程”](#)
- 第 364 页上的[“根据带输入参数的存储过程创建 Universe”](#)
- 第 366 页上的[“带多个结果集的存储过程”](#)

6.12.2.1 存储过程中的类和对象

- Universe 设计工具为每个选定的存储过程生成一个表（或在存在多个结果集时生成多个表），并为存储过程返回的每列生成一个对象。
- 结果集结构在用户描述函数时确定。

6.12.2.2 使用存储过程

用户可以根据一个或多个不带参数的存储过程来创建 Universe。请使用“快速设计向导”（从工具栏中选择）。

1. 单击“快速设计向导”工具栏按钮。
此时将出现“欢迎”窗格。

- 2 单击窗格底部的复选框“单击这里选择存储过程 Universe”。
- 3 单击“开始”。
此时将出现“定义 Universe 参数”面板。
- 4 在“输入 Universe 名称”字段中键入 Universe 名称。
- 5 在“选择数据库连接”列表的下拉列表中选择数据库连接。
- 6 单击“下一步”。
此时将出现“创建初始类和对象”面板。
- 7 单击存储过程。
- 8 单击“添加”。
“Universe 类和对象”窗格中将创建此存储过程。
- 9 单击“下一步”。
- 10 单击“完成”。出现“配置”面板。

6.12.2.2.1 Universe 中的存储过程参数

使多个存储过程基于相同数据源的相同 Universe，可以提高产品的性能（通过“快速设计向导”或“插入”>“存储过程”>“更新”）。

在 Universe 参数中，检查参数 STORED_PROC_UNIVERSE 是否设置为 YES。设置为“YES”则表明当前 Universe 是基于存储过程。

为避免在存储过程列中出现分析错误，建议依据复杂 SQL 为结果列提供别名（例如，使用聚合函数 Sum、Count），而且不能对别名对象的创建加以限制。

限制：

存储过程不支持 OUT 或动态结果集参数。

6.12.2.3 根据带输入参数的存储过程创建 Universe

如果存储过程提示用户从数据库的列表中选择一个值，则表明用户已声明对象。

用户可以根据要求一个或多个输入参数的存储过程来创建 Universe。根据所输入的值，存储过程从相应的构面表中返回该值的构面数据。

1. 单击“快速设计向导”工具栏按钮。
此时将出现“欢迎”窗格。
2. 单击窗格底部的复选框“单击这里选择存储过程 Universe”。
3. 单击“开始”。
此时将出现“定义 Universe 参数”面板。
4. 在“输入 Universe 名称”字段中键入 Universe 名称。
5. 在“选择数据库连接”列表的下拉列表中选择数据库连接。
6. 单击“下一步”。
此时将出现“创建初始类和对象”面板。
7. 单击存储过程。
8. 单击“添加”。
当存储过程需要输入参数时，将显示“存储过程编辑器”。
9. 单击参数列表中的某个参数。
10. 在提示中或在“值”字段中输入参数值。
11. 选择“使用此值”或“提示我输入值”。
在输入值后，当执行此过程时，它会检索列和结果集结构，从而将此值传递给过程。
12. 如果选定了“提示我输入值”，请在提示中键入值。
用户可以输入提示消息，也可以浏览以选择一个现有对象（例如，表中的“预订 ID”列表）。
13. 单击“确定”。
14. 单击“下一步”。
15. 单击“完成”。出现“配置”面板。

6.12.2.3.1 值提示

使用提示可以在执行存储过程时为参数定义值。

默认情况下，存储过程参数名称源自存储过程结构，并显示存储过程名称的提示。

用户可以调整其语义，并将值列表与此提示关联，以便从列表中添加更多值。

在存储过程参数对话框中，每个参数前面有一个按钮，通过此按钮可以打开高级对话框。

6.12.2.4 在存储过程的提示中使用值列表

在定义带动态参数的存储过程时，可以通过将值列表（LOV）与提示定义（此值列表必须来自标准表）关联的方式来选择基于标准表的 Universe 对象。它适用于向用户建议 Universe 对象值列表的情况。

值列表只能包含简单值。用户不能编辑或创建定制的值列表。

注意：

当插入不属于存储过程定义类或对象时，这些类或对象被隐藏。用户不能更改其状态以显示这些类和对象。

6.12.2.5 带多个结果集的存储过程

示例：一个返回多个结果集的存储过程。在设计时，是依据相同存储过程，采用 Universe 结构来创建多个表的。

```
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
```

处理多个结果集（RS）的思路是：

存储过程 RS1：a、b 以及 RS2：b、d、e

表 A1：A、B

表 A2：B、D、E

此示例语法依据相同 ID 生成两个表。当在 Universe 设计工具模块中编辑表 A1 时，同时还可以编辑表 A2。

列是根据存储过程的结果集结构进行分发的。这两个表也是依据相同的存储过程生成的。在此示例中，这两个流程的结果集结构是相同的。Universe 设计工具会重新对源自存储过程中第二个结果集的表名进行命名。Universe 设计工具可以精心构思业务元素。

Universe 设计工具在 Universe 中为每个结果集生成一个表，并为每个表创建一些相互独立的对应对象。用户可以按正常存储过程的方式来修改 Universe。

6.13 测试 Universe

可以测试 Universe 中对象和类的完整性，方法是使用“检查完整性”（“工具”>“检查完整性”）运行定期检查，并测试 Web Intelligence 中的对象。通过使用“查询面板”来创建带有 Universe 对象的查询并单击“查看 SQL”按钮，您还可以查看对象在查询中生成的 SQL。

6.13.1 在“查询面板”中测试对象

可以使用“查询面板”按如下方式查看查询的 SQL：

- 1 选择“工具”>“查询面板”。

将出现“查询面板”。

- 2 将对象拖到右边的“结果”窗格。
- 3 单击“SQL”按钮。



- 4 查询的 SQL 即会出现。
- 5 单击“确定”，然后单击“取消”关闭“查询面板”。

6.13.2 测试 Universe 的完整性

在创建和修改类和对象时，您应使用“检查完整性”来定期测试 Universe 的完整性。有关使用“检查完整性”的信息，请参阅第 266 页上的[“自动检查 Universe 完整性”](#)。

6.13.3 使用 Web Intelligence 测试 Universe

可以通过在 Web Intelligence 中运行测试查询来测试对象。在测试对象时，可以提出以下类型的问题：

- 对象是否存在？如果不存在，是否在最后一次创建之后保存了 Universe？
- SQL 是否正确？
- 查询的结果是否正确？

您还必须测试联接，方法是评估返回的结果是否正确，并使用“检查完整性”检查模式的各组成部分。

第 7 章 优化 Universe

通过优化 Universe 通常可缩短查询时间。可以使用几种方法优化 Universe:

- 优化“Universe 参数”中的“数组提取”参数。
- 向每个表分配一个权重。
- 使用快捷联接。
- 在数据库中创建和使用聚合表。

以下说明了上述每种方法:

7.1 概述

可通过使用以下方法来优化 Universe:

- 第 369 页上的 [“使用聚合表”](#)
- 第 383 页上的 [“在对象的 SQL 中使用 @ 函数”](#)
- 第 410 页上的 [“使用外部策略定制 Universe 创建”](#)
- 第 425 页上的 [“使用分析函数”](#)

7.2 使用聚合表

可以使用 Universe 设计工具中的功能,使自己能为对象定义 Select 语句,以依据数据库中的聚合表而不是基表运行查询。可以设置条件,以便依据聚合表运行查询,前提是这样做能优化查询,而如果不能,则依据基表运行查询。对象使用聚合表优化查询的能力称为聚合感知。

本章说明如何在 Universe 中设置聚合感知。

7.2.1 什么是聚合感知？

聚合感知是一个术语，用于说明 Universe 利用数据库中的聚合表的能力。这些表是包含预先计算的数据的表。在对象的 Select 语句中，可以使用一个称为 @Aggregate_Aware 的函数，该函数指示查询将依据聚合表而不是包含非聚合数据的表来运行。

使用聚合表会加快查询的执行速度，从而提高 SQL 事务的性能。

Universe 中的聚合感知的可靠性和实用性取决于聚合表的准确性。必须在刷新所有事实表的同时刷新聚合表。

如果 Universe 具有一个或多个对象，而这些对象包含基于聚合表的备用定义，则表示它是“聚合感知”的。这些定义对应于聚合的级别。例如，可以按月、按季或按年聚合一个称为“利润”的对象。这些对象称为聚合对象。

从使用聚合对象 Universe 构建的查询将以最佳的速度返回聚合到适当级别的信息。

7.2.2 将聚合感知应用于数据仓库

聚合感知在用于数据仓库时特别有用。例如，考虑一个组织为三个维的数据仓库：时间、地理和产品。

在其最低的级别上，此数据仓库可以存储有关客户和产品的每日信息。每个客户每日购买的产品都有一行记录；这可以用以下公式表示：

365 天 x 100 个城市 x 10 种产品 = 365,000 行。

如果您需要有关年度销售的信息，则数据库引擎必须将庞大数目的行累加起来。但是，公司的年度销售实际上可以包含较少的行，如下所示：

3 年 x 3 个国家/地区 x 3 家公司 = 27 行

因此，在本例中，表中的 27 行就足以解答您的问题。根据此信息，将这些行预先汇总到聚合表中将产生非常高的效率。

7.2.3 设置聚合感知

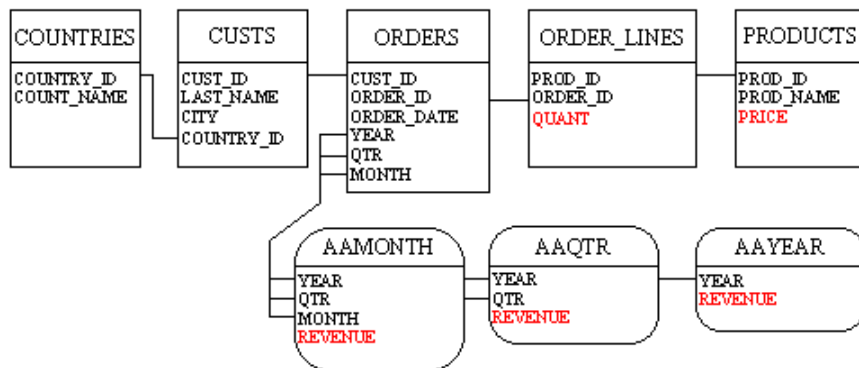
在 Universe 中设置聚合感知是一个分为四部分的过程。下面概述了此方法的主要步骤。

- 构建对象：
 - 。 1 确定对象的所有可能定义（表/列组合）。
 - 。 2 按聚合级别排列对象。
 - 。 3 使用 @Aggregate_Awareness 函数构建对象。
- 指定不兼容的对象：
 - 。 1 构建对象/聚合表矩阵。
 - 。 2 对于第一个聚合表，确定每个对象是兼容（在同一聚合级别或更高级别）还是不兼容的（在更低的聚合级别）。
 - 。 3 仅选定对于该表不兼容的那些对象的复选框。
 - 。 4 为其余的聚合表重复步骤 1 到 3。
- 定义任何必要的上下文

为每个聚合级别定义一个上下文。

- 测试结果
 - 。 1 运行几个查询。
 - 。 2 比较结果。

以下各节中详细说明了上述过程的每一阶段。为了说明每一阶段，使用了以下所示的示例模式：



此模式包含三个预定义的聚合表：AAMONTH、AAQTR 和 AAYEAR。

注意：

此示例模式并不代表典型的模式。使用它作为按照步骤来设置聚合感知的一种方式。在用于生产的模式中，聚合表通常会组合几个维而不是单一的基于时间的维。时间维（年度、季度和月份）也通常是在主表而不是在聚合表中定义的。

7.2.4 构建对象

在 Universe 中设置聚合感知的第一步是确定哪些对象将会是聚合感知的。可以使用度量对象或维对象。

根据以上模式，对象“Sales Revenue（销售收入）”具有以下定义：

PRODUCTS.PRICE*ORDER_LINES.QUANT

您想重新定义“Sales Revenue（销售收入）”，以便在可能时使用聚合表而不是使用非聚合表执行聚合。

如果您想在任何其他对象的定义中使用聚合表，则也需完成为了将“Sales Revenue（销售收入）”重新定义为聚合感知对象而完成的每个阶段。

7.2.5 确定聚合对象的所有组合

您需要确定各个表中的对象的所有可能组合。可以通过以下方式定义“Sales Revenue（销售收入）”对象：

- AAMONTH.REVENUE
- AAYEAR.REVENUE
- AAQTR.REVENUE
- PRODUCTS.PRICE*ORDER_LINES.QUANT

7.2.6 按聚合级别顺序排列对象

确定了对象的所有组合后，按照对象的聚合级别排列对象，如下所述：

- AAYEAR.REVENUE 是最高级别的聚合。

- AAQTR. REVENUE 是下一个级别。
- AAMONTH. REVENUE 是下一个级别。
- PRODUCTS. PRICE*ORDER_LINES. QUANT 是最低的聚合级别。

7.2.7 使用 @Aggregate_Aware 函数定义聚合对象

接下来，您使用 @Aggregate_Aware 函数为所有聚合感知对象重新定义 Select 语句。
@Aggregate_Aware 函数指示对象先查询作为其参数列出的所有聚合表。如果这些聚合表不适用，则会利用基于非聚合表的原始聚合运行查询。有关 @函数的更多信息，请参阅第 383 页上的 [“在对象的 SQL 中使用 @ 函数”](#) 一节。

“销售收入”的使用 @Aggregate_Aware 函数的 Select 语句如下所示。



@Aggregate_Aware 函数的语法如下所示：

```
@Aggregate_Aware(sum(agg_table_1), ... sum(agg_table_n))
```

其中 agg_table_1 是聚合级别最高的聚合，而 agg_table_n 是级别最低的聚合。
必须将所有聚合表的名称作为参数输入。从左到右按聚合的降序放置这些表的名称。

7.2.7.1 使用 @Aggregate_Aware 定义对象

使用 @Aggregate_Aware 重新定义对象：

- 1 双击某个对象。

将出现该对象的“编辑属性”对话框。

- 2 单击“选择”框旁边的“>>”按钮。

将出现“编辑 SELECT 语句”对话框。

- 3 单击 SELECT 语句的开头。

或

如果对象尚未具有 SELECT 语句，则单击选择框中的任何位置。

光标出现在对话框的左上角。

- 4 单击“函数”窗格中的“@函数”节点。

将出现可用 @函数的列表。



- 5 双击 @Aggregate_Aware。

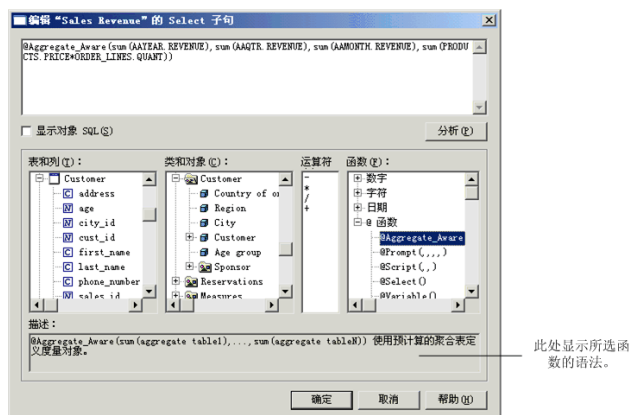
@Aggregate_Aware 的语法将插入到 Select 语句中。语法的说明出现在对话框底部的“说明”框中。可以使用这一点来帮助您为 @函数键入参数。

- 6 按顺序（从聚合数据的最高级别到最低级别）在 @AggregateAware 函数的括号内插入聚合。
- 7 用逗号分隔每个聚合。例如，“Sales Revenue（销售收入）”的语法是：

```
@Aggregate_Aware(sum (AAYEAR. REVENUE), sum(AAQTR. REVENUE), sum (AAMONTH. REVENUE),
sum(PRODUCTS. PRICE*ORDER_LINES. QUANT))
```

8 单击“分析”按钮以验证语法。

“Sales Revenue（销售收入）”的 SQL 编辑器的 Select 语句如下所示。



9 在每个对话框中单击“确定”。

在本例中，还利用 @Aggregate_Aware 函数重新定义了维对象“Year（年度）”和“Quarter（季度）”。

7.2.8 指定不兼容的对象

现在必须为 Universe 中的每个聚合表指定不兼容的对象。您指定的不兼容对象的集合确定了在生成 SQL 的过程中会忽略哪些聚合表。

就聚合表而言，对象是兼容或不兼容的。兼容性的规则如下所示：

- 当对象的聚合级别与表相同或高于表时，则它与表兼容。
- 当对象的聚合级别低于表时（或如果它与表根本不相关），则它与表不兼容。

7.2.8.1 使用矩阵分析对象

您可能会发觉，通过构建矩阵来分析对象和聚合表的兼容性很有用。在此矩阵的前两列中，可以列出类和对象的名称。然后，可以为 Universe 中的每个聚合表创建一个列标题。基于示例的模式空白矩阵如下所示：

类	对象	AAYEAR	AAQTR	AAMONTH
客户	客户代码 (CUSTOMER. CUST_ID)			
	客户名 (CUSTOMER. LAST_NAME)			
	客户所在城市 (CUSTOMER. CITY)			
	客户国籍 (COUNTRIES. COUNT_NAME)			
产品	产品代码 (PRODUCT. PROD_ID)			
	产品名称 (PRODUCT. PROD_NAME)			
订单	订单年度 (AAYEAR. PROD_NAME)			
	订单季度 (AAQTR. QTR)			
	订单月份 (AAMONTH. MONTH)			

类	对象	AAYEAR	AAQTR	AAMONTH
	订单日期 (ORDERS. ORDER_DATE)			
销售 度量	销售收入 (@Aggregate_Aware(...))			

对于每个表，如果对象是不兼容的，则输入 X。

基于示例的完成的矩阵如下所示。

类	对象	AAYEAR	AAQTR	AAMONTH
客户	客户代码 (CUSTOMER. CUST_ID)	X (n)	X (n)	X (n)
	客户名 (CUSTOMER. LAST_NAME)	X (n)	X (n)	X (n)
	客户所在城市 (CUSTOMER. CITY)	X (n)	X (n)	X (n)
	客户国籍 (COUNTRIES. COUNT_NAME)	X (n)	X (n)	X (n)
产品	产品代码 (PRODUCT. PROD_ID)	X (n)	X (n)	X (n)
	产品名称 (PRODUCT. PROD_NAME)	X (n)	X (n)	X (n)
订单	订单年度 (AAYEAR. PROD_NAME)	- (s)	- (h)	- (h)

类	对象	AAYEAR	AAQTR	AAMONTH
	订单季度 (AAQTR. QTR)	X (1)	- (s)	- (h)
	订单月份 (AAMONTH. MONTH)	X (1)	3 (1)	- (s)
	订单日期 (ORDERS. ORDER_DATE)	X (1)	X (1)	X (1)
销售 度量	销售收入 (@Aggregate_Aware(...))	-	-	-

X (n)：此对象与聚合表无关。因此它具有兼容性。

X (1)：此对象位于比此聚合表更低的聚合级别；它不能用于派生信息。因此它不兼容。

- (s)：此对象所在的聚合级别与此聚合表相同；它可用于派生信息。因此它具有兼容性。

- (h)：此对象位于比此聚合表更高的聚合级别；它可用于派生信息。因此它具有兼容性。

7.2.9 指定不兼容的对象

现在指定不兼容的对象。使用“聚合导航”对话框（“工具” > “聚合导航”）来指定不兼容的对象。

请按以下所述使用“聚合导航”对话框来指定不兼容的对象：

- 1 选择“工具” > “聚合导航”。

将出现“聚合导航”对话框。它包含两个窗格：

- Universe 表，列出了 Universe 的所有表。
- 相关的不兼容对象，列出了 Universe 的所有对象。

- 2 单击左边窗格中的一个聚合表。

- 3 在右边窗格中，选择每个不兼容对象的复选框。

例如，根据矩阵，“Customers（客户）”类中的所有对象对于 AAYEAR 表都是不兼容的。按以下所示选择类名称旁边的复选框：



- 4 为 Universe 中的每个聚合表重复以上步骤。

例如，AAQTR 表的不兼容对象如下所示。



对于 AAMONTH 表，只有一个对象不兼容。



5 为所有表指定了所有不兼容对象后单击“确定”。

注意：

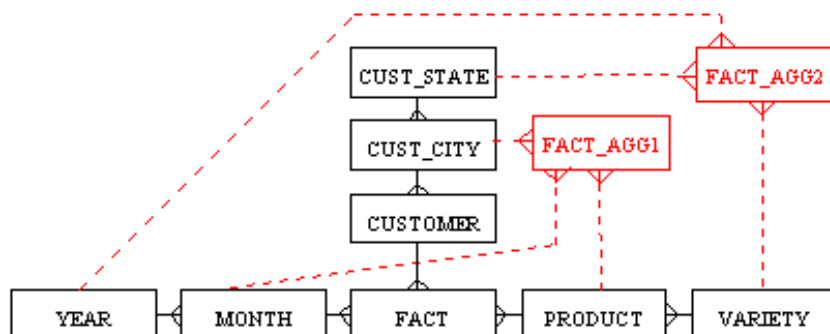
此对话框还具有一个“检测不兼容性”按钮，它可以引导用户完成指定不兼容对象的过程。单击某个表并随后单击此按钮时，Universe 设计工具会自动检查它认为不兼容的对象。应将“检测不兼容性”提出的不兼容对象视为建议的对象，而不是最终的选择。

7.2.10 解决包含聚合表的环路

当数据库包含一个或多个聚合表时，应使用上下文解决任何环路。

示例：解决包含聚合表的环路

以下显示了包含聚合表的一个简单模式：



请注意此模式中的以下要点：

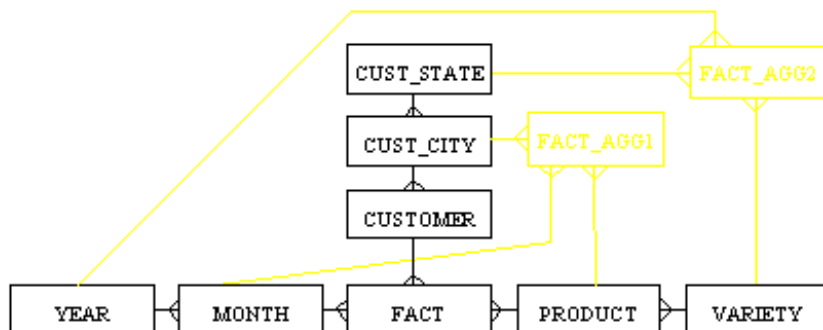
- FACT_AGG1 是一个与 FACT 表几乎相同的聚合表。它除了包含许多个聚合到 “Customer City（客户所在城市）”、“Product（产品）” 和 “Month（月份）” 的度量外，还包含 Customer City 键、Product 键和 Month 键。
- FACT_AGG2 也是一个类似于 FACT 表的聚合表。它的度量聚合到 “Customer State（客户所在省/州）”、“Product（产品）” 和 “Year（年度）”。
- 度量（关键绩效指标）存储在所有事实表中。“Sales Revenue（销售收入）” 存储在 FACT_AGG1、FACT_AGG2 和 FACT 中，但聚合到每个表的相应级别。

对于包含 “Sales Revenue（销售收入）” 和 “Customer State（客户所在省/州）” 的查询，您希望使用 CUST_STATE 和 FACT_AGG2 之间的联接而不是 CUST_STATE 和 CUST_CITY 之间的联接。

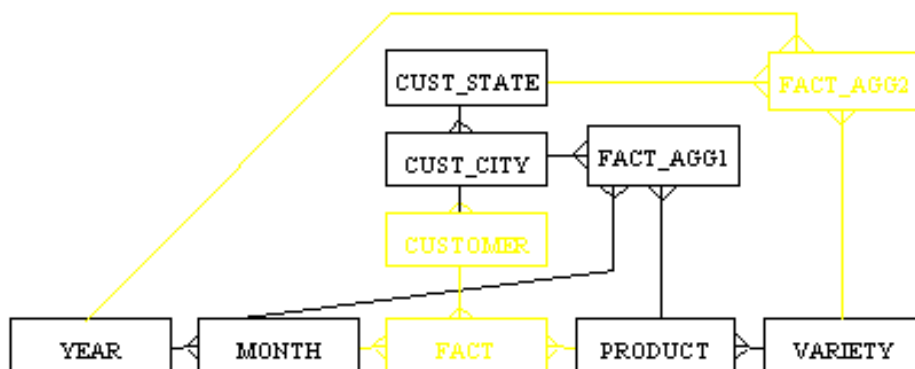
但是，在可以运行此查询之前，您需要定义三个上下文，例如 FACT、FACT_AGG1 和 FACT_AGG2。无需将上下文重命名为更有意义的标签，因为它们对于用户是透明的。

下一页说明了包含在这三个上下文中的联接。在每个模式中，较深色的一组联接代表着指定的上下文。

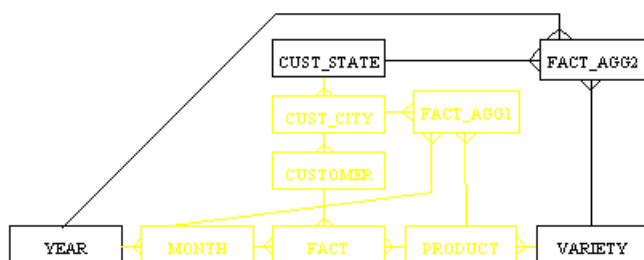
FACT 上下文



FACT_AGG1 上下文



FACT_AGG2 上下文



7.2.11 测试聚合感知

设置聚合感知的最后一步是在 Web Intelligence 中测试结果。

根据第一个示例，我们可以运行以下查询，然后比较不同的结果：

- “Order Year（订单年度）”对“Sales Revenue（销售收入）”。
- “Order Quarter（订单季度）”对“Sales Revenue（销售收入）”。
- “Order Month（订单月份）”对“Sales Revenue（销售收入）”。
- “Customer（客户）”对“Sales Revenue（销售收入）”。

- “Product（产品）”对“Sales Revenue（销售收入）”。

7.3 在对象的 SQL 中使用 @ 函数

@ 函数是一种特殊的函数，它们提供了更多用于为对象指定 SQL 的灵活方法。对象的“编辑 Select”框的“函数”窗格中提供了 @ 函数。

可以在对象的 SELECT 语句或 WHERE 子句中包括一个或多个 @ 函数。可使用以下 @ 函数：

@ 函数	说明	通常用于对象的以下语句中
@Aggregate_Aware	在对象中结合包含聚合和维数据的列。	SELECT 语句
@Prompt	用于在 SQL 中插入提示。每当在查询中包括使用 @Prompt 函数的对象时，如果用户运行查询，将提示用户输入一个值进行限制。	SELECT 语句 WHERE 子句
@Script	每次在查询中包括使用 @Script 函数的对象时运行脚本。	WHERE 子句
@Select	允许使用另一个对象的 SELECT 语句。	SELECT 语句
@Variable	调用内存中存储的变量的值，比如在引用的文本文件中调用。	WHERE 子句
@Where	允许使用另一个对象的 WHERE 子句。	WHERE 子句

示例：使用 @Prompt 函数可以将返回的值限定为输入的提示值

@Prompt 函数是 Universe 设计工具中的一种可用的 @Functions。在 Web Intelligence 查询中使用对象时，可以使用 @Prompt 函数来显示消息框。

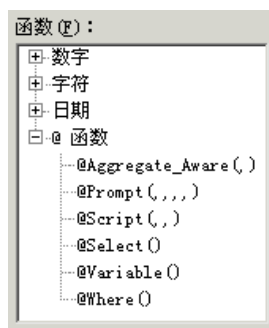
该消息框提示用户输入对象的值。查询将返回所输入提示的值，如下所示：



7.3.1 在对象中插入 @函数

在对象的 SQL 定义中插入 @函数：

- 1 双击“Universe”窗格中的对象。
将出现该对象的“编辑属性”对话框。
- 2 单击“选择”框旁边的“>>”按钮。
或
单击“位置”框旁边的“>>”按钮。
将出现“编辑 SELECT 语句”或“编辑 WHERE 子句”对话框。
- 3 在 Select 语句或 Where 子句中要添加 @函数的位置单击。如果该框为空（如上所示），请在框中的任意位置单击。光标自动出现在对话框的左上角。
- 4 Click the @Functions node in the Functions pane.
将出现可用 @函数的列表。



5 双击 @函数。

@函数的语法即被添加到对象的 SELECT 语句或 WHERE 子句中。语法的说明出现在对话框底部的“说明”框中。可以使用这一点来帮助用户为 @函数键入参数。



@函数语法的说明

6 在对话框的上部窗格中键入必需参数。

7 单击“分析”按钮以验证语法。

8 在每个对话框中单击“确定”。

7.3.2 @Aggregate_Aware

@Aggregate_Aware 函数允许对象使用包含数据库中的汇总数据的表。如果数据库包含汇总表，并且用户要运行将返回聚合数据的查询，在包含汇总数据的列（而不是包含事实或事件数据的列）上运行 SELECT 语句会更快。如果对象被声明为与聚合表不兼容，它将无法使用聚合表，但会改为使用基表来进行查询。

可以使用 @Aggregate_Aware 函数在 Universe 中设置聚合感知。此过程包括多个其它步骤，这些步骤与使用 @Aggregate_Aware 函数相关。

- 为每个聚合表指定不兼容的对象。
- 为聚合表解决任何环路问题。
- 测试聚合表以确保它们返回正确的结果。

7.3.2.1 @Aggregate_Aware 函数的语法

@Aggregate_Aware 函数的语法如下所示：

```
@Aggregate_Aware(sum(agg_table_1), ...
                  sum(agg_table_n))
```

必须将所有聚合表的名称作为参数输入。从左到右按聚合的降序放置这些表的名称。

语法	说明
agg_table_1	是最高聚合级别下的聚合。
agg_table_n	是最低聚合级别下的聚合。

示例

```
@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)
                )
```

在本例中，当使用此函数的对象声明为与任何表不兼容，这些表将被忽略。对于与 R_Country 表和 R_Region 表不兼容的对象，使用的 SQL 变为：

```
@Aggregate_Aware( R_City.Revenue,
                  R_Customer.Revenue,
                  R_Age_Range.Revenue,
                  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
                )
```

7. 3. 3 @Prompt

使用 @Prompt 函数可以在查询中插入提示。提示可用于限制数据，或用于使较大的值对象更方便地在用户创建报表时使用。@Prompt 函数可在对象的 SELECT 语句或 WHERE 子句中使用。当该对象用于查询时，它会强制用户输入一个或多个值以进行限制，或强制用户选择一个值或一个值列表。当用户运行查询时，将出现一个要求输入值的提示框。

如果要在推导出的 SQL 中强制使用限制，但不想预先设置条件的值，则 @Prompts 十分有用。

用户可以根据需要为提示定义默认值。包含默认值的提示的行为表现与包含默认值的 Web Intelligence 提示一样。

可以通过以下方式编辑 @Prompt 定义：

- 使用 @Prompt 编辑器。
- 在条件的“编辑属性”对话框的“定义”窗格中，在“SELECT”或“WHERE”字段中键入定义。
- 在“编辑属性”对话框的“高级”编辑窗格中键入定义。

注意：

- 例如，如果使用的两个提示非常相似，但是使用的主键不同，请勿使用相同的问题（提示文本）。这是因为系统无法区分这两个提示，因此可能提供不恰当的答案。
- 由于 @Prompt 定义可能很复杂，特别是在键入值列表时，因此建议使用 @Prompt 编辑器。
- 当 @Prompt 函数是单值时，可以在同一查询中合并 @Variable 函数与 @Prompt 函数。

相关主题

- 第 387 页上的[“@Prompt 编辑器”](#)
- 第 392 页上的[“编辑现有 @Prompt 表达式”](#)
- 第 398 页上的[“@Prompt 函数的语法”](#)
- 第 393 页上的[“手动定义 SQL 语句的 @Prompt 函数 ”](#)

7.3.3.1 @Prompt 编辑器

使用 @Prompt 编辑器可以定义或编辑提示，这些提示是在 Web Intelligence 或 Desktop Intelligence 用户对关系 Universe 或 OLAP Universe 运行查询时显示的。该编辑器可简化提示的定义或编辑过程。提示定义显示在窗格的底部，并在定义提示的不同值时自动进行更新。右键单击现有的 @Prompt 表达式，只有当 @Prompt 字符串的语法正确时才能启用“编辑 @Prompt”菜单项。

还可以直接在“编辑属性”对话框的“SELECT”或“WHERE”窗格中键入 @Prompt 定义。

相关主题

- 第 398 页上的[“@Prompt 函数的语法”](#)
- 第 393 页上的[“手动定义 SQL 语句的 @Prompt 函数 ”](#)
- 第 390 页上的[“为提示定义静态值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 391 页上的[“为提示选择自定义值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 392 页上的[“编辑现有 @Prompt 表达式”](#)

7.3.3.2 @Prompt 编辑器的 @Prompt 表达式的属性

用户可以编辑提示的以下属性：

属性	说明
消息	用户看到的提示消息。例如，可以键入“选择国家/地区”。提示文本（问题）。 默认值 = 输入值
值类型	用户输入或选择的数据类型。这样可以保证用户输入或选择正确类型的数据。从以下各项选择： <ul style="list-style-type: none"> • Alphanumeric (A) • Numeric (N) • Date (D) 默认值 = Alphanumeric
键类型	选择 Primary_key 作为选择模式时，设置用户输入或选择的键类型。从以下各项选择： <ul style="list-style-type: none"> • Alphanumeric (A) • Numeric (N) • Date (D) 默认值 = 无
允许多项选择	选择此选项时，用户可输入或选择多个值。 默认值 = 未选择 - 用户只能选择或输入一个值
选择模式	定义用户如何选择提示所需的值。选择： <ul style="list-style-type: none"> • Free: 用户可以输入任何值。 • Constrained: 用户必须从建议的值中进行选择。 • Primary_key*: 在查询中使用主键值，将会大大缩短响应时间。可以为 Universe 中的对象选择或输入主键。在提示时，用户选择对象名称，但查询实际使用的是相应的主键值。*只有在数据库中设置了“索引感知”时，才能使用主键设置。 默认值 = Free
保持上次选择的值	选择该选项后，在用户下次运行该查询时，会提示使用上次使用的值。如未选择该选项，则始终使用默认值。 默认值 = 未选择
显示值	用户必须从值列表中选择一项或多项。可通过以下方式定义值列表： <ul style="list-style-type: none"> • 在“值列表”框中键入值列表（这是静态列表） • 从 Universe 中选择一个对象 • 使用“文件输入向导”导入文件 默认值 = Static

属性	说明
默认值	可以声明提示的默认值。

相关主题

- 第 398 页上的“[@Prompt 函数的语法](#)”
- 第 393 页上的“[手动定义 SQL 语句的 @Prompt 函数](#)”
- 第 390 页上的“[为提示定义静态值列表](#)”
- 第 391 页上的“[选择 Universe 对象作为提示的值列表](#)”
- 第 391 页上的“[为提示选择自定义值列表](#)”
- 第 391 页上的“[选择 Universe 对象作为提示的值列表](#)”
- 第 392 页上的“[编辑现有 @Prompt 表达式](#)”

7.3.3.3 使用提示编辑器创建 @Prompt 表达式

将在 Universe 设计工具中打开一个 Universe，而用户则可以开始创建或编辑对象。

- 1 在条件的“SELECT”或“WHERE”字段中，右键单击要在表达式中添加提示的位置，并从快捷菜单中选择“@Prompt 编辑器”。
- 2 在“消息”框中，键入要向用户显示的消息。
- 3 如果希望提示建议上一个报表使用的值，请选择“保留上次选择的值”。
在首次运行报表时，建议默认值（如果有）。
- 4 如果用户可以输入或选择多个值，请选择“允许多项选择”。
- 5 设置“选择模式”。如果选择“free”，则用户可以输入允许的任何值。如果选择“constrained”，则用户必须从值列表中选择一个值。如果选择“primary_key”，则用户选择对象名称，但对象的主键由查询使用。只有设置了“索引感知”时，才能使用“primary_key”。

注意：

如果选择“constrained”选项并且未指定值列表，则相应的选项卡颜色将变为红色，“确定”按钮将被禁用，并且在突出显示的选项卡上移动光标时，上下文工具提示将显示一条消息，指示如何纠正此问题。

- 6 设置选择模式的“值类型”：Alphanumeric、Number 或 Date。
- 7 如果选择了“primary_key”作为选择模式，则将“键类型”设置为：Alphanumeric、Number 或 Date。
- 8 如果使用多选，则定义值列表。执行下列操作之一：可输入或导入值列表，或选择 Universe 对象。
- 9 单击“默认值”选项卡，并定义任何默认值。允许多项选择时，可以设置多个默认值。
- 10 单击“确定”。

@Prompt 编辑器关闭，然后可以在“编辑属性”窗格的条件语句中看到提示表达式。

11. 验证提示并单击“分析”。

如果语法不正确，将显示“分析失败”消息，指明含有错误的 SQL 片段。

相关主题

- 第 398 页上的[“@Prompt 函数的语法”](#)
- 第 393 页上的[“手动定义 SQL 语句的 @Prompt 函数”](#)
- 第 390 页上的[“为提示定义静态值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 391 页上的[“为提示选择自定义值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 392 页上的[“编辑现有 @Prompt 表达式”](#)

7.3.3.4 为提示定义静态值列表

@Prompt 编辑器的下半部分包含用于定义静态值列表的表窗格。用户可以使用向上箭头和向下箭头更改所输入值的位置。当“标题”标题文本为红色时，必须填写值或改正值。

- 1 在“标题”字段中输入第一个值。如果已选定“Primary_key”作为选择模式，请在第二个字段中键入索引值。
- 2 单击“+”以在包含静态值列表的表中插入值。
- 3 在表中输入更多值以完成值列表。
- 4 在“默认值”窗格中输入任何默认值。

用户运行报表时将建议默认值。在设置“保留上次选择的值”时，将建议上次运行报表时使用的值，否则每次运行报表时都会建议默认值。

注意：

- 要编辑静态列表中的值，请选择该值，然后在表中进行编辑，并单击“更新”按钮。
- 要删除一个值，请选择该值并单击“-”。

相关主题

- 第 398 页上的[“@Prompt 函数的语法”](#)
- 第 393 页上的[“手动定义 SQL 语句的 @Prompt 函数”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 391 页上的[“为提示选择自定义值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 392 页上的[“编辑现有 @Prompt 表达式”](#)

7.3.3.5 选择 Universe 对象作为提示的值列表

用户可以选择 Universe 对象作为提示的值列表。

- 1 在 @Prompt 编辑器的值列表窗格中选择“Universe 对象”。
- 2 导航到要在提示中使用的 Universe 对象，并选择该对象。
- 3 双击该对象。

@Prompt 编辑器关闭，然后可以在“编辑属性”窗格的条件语句中看到提示表达式。

相关主题

- 第 398 页上的[“@Prompt 函数的语法”](#)
- 第 393 页上的[“手动定义 SQL 语句的 @Prompt 函数”](#)
- 第 390 页上的[“为提示定义静态值列表”](#)
- 第 391 页上的[“为提示选择自定义值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 392 页上的[“编辑现有 @Prompt 表达式”](#)

7.3.3.6 为提示选择自定义值列表

使用“数据文件输入向导”可以将自定义值列表导入“@Prompt”编辑器的标题表中，以便插入 @Prompt 函数。仅支持 Excel 文件和文本文件。

- 1 在 @Prompt 编辑器的值列表窗格中选择“文件输入”。
此时将显示“数据文件输入向导”。
- 2 单击“浏览”，浏览到要使用的输入文件。
- 3 单击“打开”。
- 4 单击“下一步”。
- 5 如果将值列表的第一行用作列表中各列的标题值，请选择“第一行指定列标题”。
- 6 选择文件编码类型。
- 7 设置在输入文件中用于分隔数据的“分隔符”字符。
- 8 单击“下一步”。
- 9 单击“Get unique column values”（获取唯一列值），以确保仅建议唯一值。

10. 单击“Get number of TOP records”（获取前几个记录），以定义将向用户建议的值数量。
11. 使用“Column map”（列导航图）窗格选择用于标题和主键值的列。
12. 单击“Sort on column”（对列排序）对选定的列排序，选择“升序”以按升序对列排序。如果没有选择“升序”，则按降序对列排序。
13. 单击“完成”。
将显示“Column row append”（列/行追加）确认消息。单击“是”确认选择。
14. 自定义值列表插入到“@Prompt”编辑器的值列表表格，同时这些值插入到 @Prompt 定义中。
15. 单击“确定”，将 @Prompt 插入到 SELECT 或 WHERE 子句中。

相关主题

- 第 398 页上的[“@Prompt 函数的语法”](#)
- 第 393 页上的[“手动定义 SQL 语句的 @Prompt 函数 ”](#)
- 第 390 页上的[“为提示定义静态值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 392 页上的[“编辑现有 @Prompt 表达式”](#)

7.3.3.7 编辑现有 @Prompt 表达式

将在 Universe 设计工具中打开一个 Universe。选定的对象表达式包含 @Prompt 函数。

要编辑的 @Prompt 函数已存在于对象或条件的定义或 WHERE 子句中。通过 @Prompt 编辑器完成编辑。

1. 右键单击 @Prompt 函数，并从快捷菜单中选择“编辑提示”。
2. 使用“@Prompt”编辑器编辑提示表达式。
3. 单击“分析”检查 @Prompt 的语法。

相关主题

- 第 398 页上的[“@Prompt 函数的语法”](#)
- 第 393 页上的[“手动定义 SQL 语句的 @Prompt 函数 ”](#)
- 第 390 页上的[“为提示定义静态值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)
- 第 391 页上的[“为提示选择自定义值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)

7.3.3.8 手动定义 SQL 语句的 @Prompt 函数

注意：

如果以前没有定义过 @Prompt 函数，建议用户使用 @Prompt 编辑器。@Prompt 函数的语法比较复杂，请参见语法定义页面。

要为对象手动定义 @Prompt 函数，请执行以下操作：

- 1 打开“编辑属性”窗口：在 Universe 窗格中双击类、对象或条件，或右键单击对象并选择“对象属性”，或选择“编辑” > “属性”。
- 2 在“定义”窗格的“SELECT”或“WHERE”对话框中单击，然后根据所需的语法定义 @Prompt 值（请参阅下面的链接）。
- 3 单击“分析”以检查 SQL 的语法。

相关主题

- 第 398 页上的[“@Prompt 函数的语法”](#)
- 第 387 页上的[“@Prompt 编辑器”](#)
- 第 392 页上的[“编辑现有 @Prompt 表达式”](#)
- 第 390 页上的[“为提示定义静态值列表”](#)
- 第 391 页上的[“选择 Universe 对象作为提示的值列表”](#)

7.3.3.9 用于手动定义提示的 @Prompt 表达式的属性

@Prompt 函数的语法如下所示：

```
@Prompt('message','type','lov',Mono|Multi,free|constrained|primary_key
,persistent|not_persistent,{default value'})
```

示例如下所示：

```
@Prompt('Displayed text ','A','Store\City',Mono,constrained,Persistent,{ 'Paris' })
```

下表描述了 @Prompt 表达式值的属性：

属性	说明
'message'	<p>必需</p> <p>提示消息的文本。该文本必须括在单引号内，例如，'选择地区'、'选择时间段' 或 '选择陈列室'。当用户运行查询时，该文本显示在提示框中。</p>
'type'	<p>必需，但可以为空（“A”是默认设置）。</p> <p>第三个参数的数据类型。它可以是以下类型之一：</p> <ul style="list-style-type: none"> • “A”代表字母数字 • “N”代表数字 • 'D' 表示日期 <p>指定的数据类型必须括在单引号中。</p> <p>当使用硬编码的值对列表时，语法为：'value_type:key_type'，例如：'A:N'，其中第一个值是最终用户看到的标题，第二个值是为了加快查询而在查询中实际使用的主键值。每种类型（标题和主键）可以是上面指定的 A、N 或 D。例如：'A:A' 或 'A:N'。在此情况下，下一个参数 'lov' 包含参数对列表。同样，'default_value' 参数将包含值对。使用主键时，必须设置“索引感知”。</p>

属性	说明
lov	<p>必需，但可以为空。如果列表为空，仍必须使用逗号。当此参数是 Universe 对象时，第五个参数（选择模式 = free constrained primary_key）必须是“primary_key”，而且必须在 Universe 中设置了“索引感知”。</p> <p>可以指定两种类型的值列表：</p> <ul style="list-style-type: none"> 从现有 Universe 对象指向“值列表”的指针。通过双击包含要用于“类和对象”面板的值列表的对象，可以调用目标值列表。这指定了类名和对象名称，用反斜杠隔开。它必须括在单引号内。例如：'Client\Country'。如果正在使用“索引感知”，并且希望返回对象的键值，请将第五个值设置为“primary_key” 硬编码的单值或值对的列表。值对中的值用冒号隔开。每个值用单引号括住。值对用逗号隔开。整个列表括在花括号内。将约束设置为“primary_key”。 <p>一个单默认值的语法：</p> <p>{ '值' }</p> <p>多个单默认值的语法：</p> <p>{ '值1', '值2', ..., '值n' }</p> <p>可定义默认值对。</p> <p>一对默认值的语法：{ '值' : '键' }。</p> <p>冒号 (:) 用作值与键之间的分隔符。</p> <p>多对默认值的语法：</p> <p>{ '值1' : '键1', '值2' : '键2', ..., '值n' : '键n' }</p> <p>例如：{ 'Australia' : 'A', 'France' : 'F', 'Germany' : 'G', 'Japan' : 'J', 'Spain' : 'S', 'United Kingdom' : 'UK' }</p>
Mono Multi	<p>必需，但可以为空（Mono 是默认设置）。必须使用逗号。</p> <p>如果只能从值列表中选择一個值，则使用 Mono。</p> <p>如果可以从值列表中选择多个值，则使用 Multi。</p>

属性	说明
<code>free constrained primary_key</code>	<p>必需，但可以为空（<code>free</code> 是默认设置）。必须使用逗号。</p> <p>如果可以输入一个值或从值列表中选择一个值，则使用 <code>free</code>。</p> <p>如果必须从值列表中选择一个值，则使用 <code>constrained</code>。</p> <p>如果已在 Universe 中设置“索引感知”，则使用 <code>primary_key</code> 参数。使用对象关联的关键值，而不是输入或显示的值。在使用 <code>lov</code> 参数时，<code>primary_key</code> 是必需参数。</p>
<code>persistent not_persistent</code>	<p>可选。如果设置了此项，则以逗号作为参数结尾。如果未设置此项，但设置了第七个参数（默认值），还必须设置逗号。</p> <p>注意： 请注意，此参数在 Desktop Intelligence 中不起作用。</p> <p>如果在刷新文档时，即使定义了默认值，默认情况下仍显示提示中上次使用的值，则使用 <code>persistent</code>。</p> <p>如果在刷新文档时，默认情况下没有显示提示中使用的值，则使用 <code>not_persistent</code>。</p>
<code>'default value'</code>	<p>可选。'default value' 参数用于定义向用户显示的默认值。当使用硬编码的列表时，在此处输入的默认值必须显示在 [lov] 列表中。</p> <p>例如，对于单值：</p> <pre>{ 'France' }</pre> <p>对于一对值：</p> <pre>{ 'France' : 'F' }</pre> <p>对于两对值：</p> <pre>{ 'France' : 'F', 'Germany' : 'G' }</pre> <p>在刷新文档时，默认情况下将显示这些值，但是如果设置了“persistent”选项，则采用提示中上次使用的值，而不是默认值。</p> <p>用户可具有单值或值对。</p> <p>如果在提示定义中指定“primary_key”参数，则必须提供键值。</p>

7.3.3.10 示例：使用 @Prompt 函数

下面是 @Prompt 语法的示例。

@Prompt 函数的最简单用法：

```
@Prompt('Displayed text ','A',,,)
```

将 @Prompt 与不带默认值的值列表一起使用：

```
@Prompt('Displayed text ','A',{'Paris','London','Madrid'},,,)
```

将 @Prompt 与值列表和一个默认值一起使用：

```
@Prompt('Displayed text ','A',{'Paris','London','Madrid'},,,,{'Paris'})
```

将 @Prompt 与一个对象和一个默认值一起使用：

```
@Prompt('Displayed text ','A','Store\City',,,,{'Paris'})
```

将 @Prompt 与所有可能的设置一起使用：

```
@Prompt('Displayed text ','A','Store\City',Mono,Constrained,Persistent,{'Paris'})
```

将 @Prompt 与包含值对、但不包含默认值的值列表一起使用：

```
@Prompt('Displayed text ','A:N',{'Paris':'12','London':'7','Madrid':'15'},,,)
```

将 @Prompt 与包含值对及一对默认值的值列表一起使用：

```
@Prompt('Displayed text ','A:N',{'Paris':'12','London':'7','Madrid':'15'},,,,{'Paris':'12'})
```

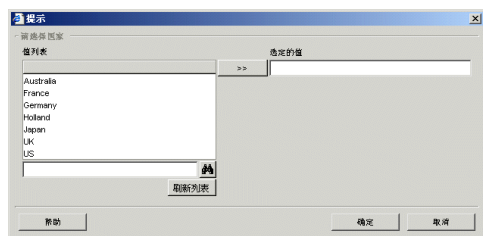
示例：使用 @Prompt 可以限制不带默认值的国家/地区

对象“Country（国家/地区）”返回休假地所在国家/地区的值。如果要将返回的值限制为仅为一个国家/地区的休假地，Universe 中的每个休假地国家/地区将需要一个单独的对象。然而，通过使用 @Prompt，只需要一个对象，如下所示：

```
Country.country=@prompt('Choose a country',
'A','Customer\Country of origin',Mono,primary_key,,,)
```

将会提示用户输入国家/地区的名称，并且返回的值是该特定国家/地区中的休假地。

当查询在 Web Intelligence 中运行时，将出现以下提示框：



示例：包含默认值的 @Prompt 语法

```
@prompt('Enter value(s) for Customer with IA:',
'A','Customer\Customer with I A',
Multi,primary_key,,{'Baker','Brendt'})
```

示例：使用硬编码值列表的 @Prompt 语法

以下示例显示如何建议带有默认值的国家/地区列表。如果对象是索引感知的，并且约束设置为“primary_key”，则可以将默认值设置为一组值对（值，键），例如：{'England: 21', 'Scotland:39'}。用户只能选择一个地区，因此设置了“Mono”参数。默认值必须显示在值列表中。

```
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'})
```

通过此功能，还可以在不支持 CASE WHEN ELSE 子句的数据库中（特别是在 OLAP 数据库中）执行此子句的行为。

示例：@Prompt 语法创建使用匹配模式提示的预定义条件

下例允许用户通过输入名称的首字母来选择客户名称。如果 Web Intelligence 用户输入 H%，报表将返回所有其姓氏以“H”开头的客户。

```
(@Select(Client\Client Name)
LIKE (@Prompt('enter','A',,,)+%))
```

要允许客户使用大小写字母，语法如下：

```
(@Select(Client\Client Name)
LIKE lower(@Prompt('enter','A',,,)+%) OR
(@Select(Client\Client Name)
LIKE upper(@Prompt('enter','A',,,)+%))
```

7.3.3.11 @Prompt 函数的语法

由于 @Prompt 函数的功能众多，因此此函数的语法较为复杂。编写一条提示消息并指定数据类型（数据是单一值还是多个值，数据是否为永久数据），而且可以指定默认值。语法如下所示：

```
@Prompt('message','type',[lov],Mono|Multi,
free|constrained|primary_key,persistent|not_persistent,[default_values])
```

相关主题

- 第 386 页上的“[@Prompt](#)”
- 第 393 页上的“[手动定义 SQL 语句的 @Prompt 函数](#)”
- 第 393 页上的“[用于手动定义提示的 @Prompt 表达式的属性](#)”

7.3.4 @Script

@Script 函数返回 Visual Basic for Applications 宏（VBA 宏）的结果。VBA 宏只能在 Windows 环境中运行。每次刷新或运行包含该对象的查询时，使用 @Script 函数运行指定的 VBA 宏。

一般在 WHERE 字句中使用 @Script 函数运行比简单提示框更复杂的进程（@Prompt 函数）。VBA 宏保存在 BusinessObjects 报表文件（.REP）中。这些报表的默认目录是 BusinessObjects 路径中的 UserDocs 文件夹，但是，您可以定义任何文件夹来存储 .REP 文件。

注意：

@Script 仅在 Universe 设计工具和 Desktop Intelligence 客户端版本或 Desktop Intelligence 三层模式中得到支持。强烈建议不要在 Desktop Intelligence 客户端版本之外使用 @Script 函数。InfoView 中的 Desktop Intelligence 服务器版本不支持使用此函数来发布或计划 Desktop Intelligence 报表，在 Web Intelligence 中也不支持。在 Web Intelligence 中，不应当为交互式对象使用 @Script 函数，而应当保留使用 @Prompt 函数的更简单设计。

7.3.4.1 @Script 函数的语法

@Script 函数的语法如下所示：

```
@Script('var_name', ['var_type'], 'script_name')
```

注意：

第二个参数是可选的；但是，即使忽略了该参数，您仍然必须包括逗号作为分隔符。

下表中描述了语法：

语法	说明
'var_name'	宏中声明的变量名称。此名称使所执行宏的结果可以在对象的 SQL 定义中重现。此名称在 VBA 宏和对象的 SQL 定义中必须相同。
'var_type'	<p>（可选）函数返回的数据类型。它可以是以下类型之一：</p> <ul style="list-style-type: none">• “A” 代表字母数字• “N” 代表数字• 'D' 表示日期 <p>指定的数据类型必须括在单引号中。</p>
'script_name'	要执行的 VBA 宏的名称。

7.3.5 @Select

用户可以使用 @Select 函数来重新使用另一个对象的 SELECT 语句。在对象的 SELECT 语句中使用 @Select 函数时，它将指定 Universe 中另一个对象的路径作为 @Select 函数的参数，格式为“类名\对象名称”。然后，此参数将充当指向所引用对象 SELECT 语句的指针。

使用 @Select 函数有以下好处：

- 必须只保留一个 SQL 代码实例。
- 确保代码的一致性。

注意：

当使用 @Select 和 @Where 函数时，一个对象现在将取决于 Universe 中的另一个对象。已创建了新的对象相关性。当删除一个对象时，需要手动更新使用 @Select 或 @Where 函数的另一个对象。

7.3.5.1 @Select 函数的语法

@Select 函数的语法如下：

```
@Select (Classname\ObjectName)
```

语法	说明
Classname	包含所引用对象的类的名称。
Objectname	所引用对象的名称。

7.3.5.2 使用 @Select 函数的示例

示例：使用 @Select 来重新使用服务系列 Select 语句

创建了一个名为“Promotional Service Line”（促销服务系列）的对象，该对象用于返回在 Club 数据库中不同休假地的促销商业活动中使用的服务系列。此对象位于一个名为“Promotions”（促销）的新类中。可以使用 @Select 来引用“Service_lines”（服务系列）对象的现有 SELECT 语句。

“Promotional Service Line”（促销服务系列）的 SELECT 语句如下所示：



名称 (N):
promotional service line

类型 (T):
字符

描述 (D):

选择 (S):
@Select (Resort\Service Line)

>>

条件 (C):

>>

表 (T)...

分析 (A)

7.3.6 @Variable

@Variable 函数用于（例如在 WHERE 子句中）调用分配给以下一种类型的变量的值：

- BusinessObjects 系统变量
- 报表变量
- 语言（区域设置）变量
- 操作系统变量
- Desktop Intelligence 的自定义变量

大多数情况下，可通过对象的“编辑属性”表单的“定义”页面，在该对象的 WHERE 子句条件的操作数端插入 @Variable。此查询将检索变量的值。

注意：

- @Variable 是单值函数，不能与 IN 或 INLIST 运算符结合使用。
- 当某个查询中多次执行相同的 @Variable 函数时，提示仅显示一次。
- @Variable 函数与具有以下设置的单值 @Prompt 函数所起的作用相同：

```
@Prompt( 'Question' , ' A' , ,mono, free)
```

当 @Prompt 函数是单值时，可以在同一查询中合并 @Variable 函数与 @Prompt 函数。

相关主题

- 第 402 页上的[“@Variable 函数的语法”](#)
- 第 403 页上的[“@Variable 属性说明”](#)

7.3.6.1 @Variable 函数的语法

@Variable 函数的语法如下：

```
@Variable(<Variablename>')
```

注意：

变量名称括必须用单引号括起。

示例：用于返回 BOUSER 值的 @Variable 语法
@Variable('BOUSER')

相关主题

- 第 402 页上的“[@Variable](#)”
- 第 403 页上的“[@Variable 属性说明](#)”

7.3.6.2 @Variable 属性说明

在所有情况下，都必须用单引号括起变量名称。

变量名称	说明
BusinessObjects 系统变量 <ul style="list-style-type: none"> • BOUSER — 用户登录名 • DBUSER — 数据库用户名 • DBPASS — 数据库用户密码 	BusinessObjects 系统变量的值。然后基于该 BusinessObjects 用户的登录限制返回的数据。 BusinessObjects 声明的数据户用户的值。
报表变量 <ul style="list-style-type: none"> • DOCNAME — 文档的名称 • DPNAME — 数据提供程序的名称 • DPTYPE — 数据提供程序的类型 • UNVNAME — Universe 的名称 • UNVID — 所使用的 Universe 的 ID 	例如，这些变量可通过 Begin_SQL 参数引用，此参数将在 SELECT 语句之前执行。若要审核数据库的使用（例如，要确定哪个报表查询或 Universe 的使用频率最高），可采用此方法。
语言变量 <ul style="list-style-type: none"> • PREFERRED_VIEWING_LOCALE • DOMINANT_PREFERRED_VIEWING_LOCALE 	语言变量 <ul style="list-style-type: none"> • PREFERRED_VIEWING_LOCALE — 用户的首选查看区域设置。此区域设置与用户选择用于显示 Universe 元数据的区域设置相同。 • DOMINANT_PREFERRED_VIEWING_LOCALE — 用户的首选查看区域设置的主区域设置。这可以防止用户翻译所有区域设置（fr_FR、fr_BE、fr_CA 等）的数据。如果提供了 fr_FR 的翻译，则当用户区域设置为“fr_BE”或“fr_CA”时，由于它们共享相同的主区域设置，因此可以重复使用 fr_FR 的翻译。

变量名称	说明
操作系统变量	输入 Windows 环境变量可以获取关于安装的信息。
自定义变量	通过 Desktop Intelligence, 用户可以使用预定义的文本文件以提供固定变量值的列表。

相关主题

- 第 402 页上的“[@Variable](#)”
- 第 402 页上的“[@Variable 函数的语法](#)”

7.3.6.3 使用 BusinessObjects 系统变量

可以将 @Variable 函数与 BusinessObjects 系统变量一起使用, 以便根据当前登录的 BusinessObjects 用户的身份来限制数据。

注意:

BusinessObjects 登录参数必须与数据库登录参数相同。

将保留分配给每位 BusinessObjects 用户的用户名作为以下 BusinessObjects 系统变量:

- BOUSER — 用户名

当用户登录到 Business Objects 的某个产品时, 这个变量会出现在“用户标识”框中。

当在查询中使用对象时, 可以在对象的 WHERE 子句中使用 @Variable 函数, 以限制用户的数据访问权限及其数据库配置文件。

通过对象的“编辑属性”表单的“定义”页面, 可以在该对象的 WHERE 子句条件的操作数端插入 @Variable。

示例:

@Variable

在人力资源数据库的 Universe 中, 有一个名为“雇员姓名”的对象。您想要将为“雇员姓名”返回的数据限制为数据库中为每个用户授权的值。这将允许您对每个用户能够看到的雇员信息加以控制。该信息是由雇员的数据库配置文件定义的。

在 WHERE 子句中插入 @Variable 函数，如下所示：

```
Employees.Employee_Name = @Variable(' BOUSER')
```

在查询中使用“雇员姓名”对象时，将此按表中与 BOUSER 值相匹配的值返回数据。

相关主题

- 第 402 页上的“[@Variable](#)”
- 第 402 页上的“[@Variable 函数的语法](#)”
- 第 403 页上的“[@Variable 属性说明](#)”

7.3.6.4 使用区域设置变量

使用 @Variable 函数的区域设置变量定义区域设置，使 Web Intelligence 可以检索报表并按相应的区域设置显示信息。数据库表必须包含一个列，用于为含有数据译文的行声明语言。一个区域设置定义一种语言和一个地理区域、数据的排序方式、日期的格式设置以及其他特定格式。@Variable 函数可用在对象的 WHERE 子句中。当该对象用在查询中时，此函数会强制用户选择一个区域设置。当用户运行查询时，将显示一个提示框，以提示用户输入区域设置。翻译管理工具指南中包含局部代码和主区域设置代码列表。

可以定义以下设置：

- @Variable(' PREFERRED_VIEWING_LOCALE')
- @Variable(' DOMINANT_PREFERRED_VIEWING_LOCALE')

示例：

下面的 PRODUCT（产品）表已经翻译成多种语言。用户要按特定的区域设置列出产品名称。

表7-13: PRODUCT 表

Product ID	LOCALE	Product_Name
DC1212	en_GB	Digital camera
DC1212	fr_FR	Appareil photo numérique
DC1212	de_DE	Digitalkamera
DC1212	es_ES	Cámaras digitales

Product ID	LOCALE	Product_Name
...

```
SELECT Product_Name
FROM PRODUCT
WHERE PRODUCT.LOCALE = @Variable('PREFERRED_VIEWING_LOCALE')
```

在查询时，用户将变量替换为正确的区域设置值，并且 Web Intelligence 检索按该区域设置列出的信息。

相关主题

- 第 402 页上的[“@Variable”](#)
- 第 402 页上的[“@Variable 函数的语法”](#)
- 第 403 页上的[“@Variable 属性说明”](#)

7.3.6.5 使用报表变量

在对象的 WHERE 子句中使用 @Variable 函数可以将报表变量包括在请求中。

这些变量可通过 Begin_SQL 参数引用，此参数的执行顺序先于 SELECT 语句。若要审核数据库的使用（例如，要确定哪个报表查询或 Universe 的使用频率最高），可采用此方法。

变量可在以下项中引用：

- 对象的定义：SELECT、WHERE 子句等
- 过滤器
- 联接表达式
- Begin_SQL 参数

相关主题

- 第 402 页上的[“@Variable”](#)
- 第 402 页上的[“@Variable 函数的语法”](#)
- 第 403 页上的[“@Variable 属性说明”](#)

7.3.6.6 使用操作系统变量

输入 Windows 环境变量可以获取关于安装的信息。例如 NUMBER_OF_PROCESSORS、USERNAME。

示例：

如果在查询中包括 @Variable(NUMBER_OF_PROCESSORS)，结果中将包含用户目前所用计算机的处理器数。

相关主题

- 第 402 页上的“[@Variable](#)”
- 第 402 页上的“[@Variable 函数的语法](#)”
- 第 403 页上的“[@Variable 属性说明](#)”

7.3.6.7 通过 Desktop Intelligence 使用自定义变量

通过 Desktop Intelligence，可以在对象的 WHERE 子句中使用 @Variable 函数，以引用关联文本文件中的变量。这允许您在对象上定义用户特定条件。

要使用此变量，需要通过包含 -vars 参数的命令行启动 BusinessObjects。您将需要在使用此功能的所有 PC 上的 Windows 快捷方式中更改命令行。

注意：

如果确保通过命令行启动 BusinessObjects，将难于为多个用户的 Universe 部署维护 @Variable 函数。如果有多个用户，或者用户群的地理分布较散，则不能将 @函数用于关联的文本文件以实施限制。

将 @Variable 函数用于文本文件变量的优点：

将 @Variable 函数用于文本文件变量的主要优点在于：无需对 Universe 进行任何更改即可更新文本文件中变量的值。

将 @Variable 函数用于文本文件变量的缺点：

- 每次客户端提交时都必须更改命令字符串，以包括 -vars <textfile.txt> 参数。
- 可能存在安全问题，因为能够以本地方式修改 PC 上的文本文件。

由于将 @Variable 函数用于文本变量有许多潜在问题，因此，如果要在企业环境中使用 Business Objects 产品，则应当使用 Supervisor 中提供的安全选项来控制数据访问。

相关主题

- 第 408 页上的[“通过 Desktop Intelligence 使用自定义变量”](#)
- 第 402 页上的[“@Variable”](#)
- 第 402 页上的[“@Variable 函数的语法”](#)
- 第 403 页上的[“@Variable 属性说明”](#)

7.3.6.7.1 通过 Desktop Intelligence 使用自定义变量

执行以下步骤可以使用一个或多个预定义的变量值。

- 1 创建一个文本文件，该文件包含具有对应值的变量的列表。使用以下格式：变量名称 = 值
- 2 将以下内容添加至用于启动 BusinessObjects 的命令行：Busobj.exe -vars <textfile.txt> 例如，如果用户的文本文件名称为 Bovars.txt，则需要键入以下内容：C:\BusinessObjects\Busobj.exe -vars Bovars.txt。-vars 语法是一种切换，通知操作系统将文本文件加载至内存以供 BusinessObjects 使用。
- 3 打开要引用文本变量的对象的“编辑属性”表单。
- 4 在 WHERE 子句中条件的操作数端插入 @Variable。例如：COUNTRY.COUNTRY_NAME = @Variable('国家/地区')。'国家/地区'是文本文件中变量的名称。该术语必须括在单引号内。
- 5 单击“确定”并保存 Universe。

7.3.7 @Where

使用 @Where 函数可以重新使用另一个对象的 WHERE 子句。当在对象的 WHERE 子句中使用 @Where 函数时，它将指定 Universe 中另一个对象的路径作为 @Where 函数的参数，格式为“类名\对象名称”。然后，此参数将充当指向所引用对象的 WHERE 子句的指针。

使用 WHERE 子句将在两个对象之间创建动态链接。在修改原始对象的 WHERE 子句时，引用对象的 WHERE 子句自动被更新。

通过使用 @Where 函数，用户可以使用现有代码。这具有以下优势：

- 必须只保留一个 SQL 代码实例。

- 确保代码的一致性。

当使用 @Select 和 @Where 函数时，一个对象现在将取决于 Universe 中的另一个对象。已创建了新的对象相关性。当删除一个对象时，需要手动更新使用 @Select 或 @Where 函数的另一个对象。

7.3.7.1 @Where 函数的语法

此函数的语法如下：

@Where (Classname\Objectname)

语法	说明
Classname	类的名称。
Objectname	所引用对象的名称。

7.3.7.2 示例：使用 @Where 函数以重新使用 WHERE 子句

示例：通过 @Where 可以重新使用“Resort”（度假地）的 WHERE 子句

您创建了一个名为“Resort Service Lines”（度假地服务系列）的对象，该对象用于返回每个度假地可用的服务系列。并且想要重新使用“Resort”（度假地）对象中定义的 @Prompt 函数，以便在用户查询特定度假地可用的服务时，提示他们输入度假地名称。

“Resort”（度假地）对象（要引用的对象）的 SQL 如下所示：

描述 (D):

Name of the resort

选择 (S):

Resort.resort

条件 (W):

Resort.resort = @Prompt('Enter a resort name', 'A', 'Resort\Resort', Mono,)

表 (T)... 分析 (A)

新对象“Resort Service Lines”（度假地服务系列）将使用“Resort”（度假地）的 WHERE 子句中的 @Prompt 函数，如下所示：

选择 (S):

Service_Line.service_line @' at '& Resort.resort

条件 (W):

@Where (Resort\Resort)

表 (T)... 分析 (A)

在对“Resort Service Line”（度假地服务系列）运行查询时，将会提示键入度假地的名称。在修改“Resort”（度假地）的 WHERE 子句时，将会自动在“Resort Service Line”（度假地服务系列）对象中进行相应更改。

7.4 使用外部策略定制 Universe 创建

Universe 设计工具使用内置的自动化例程，基于数据库结构自动创建 Universe 元素。这些例程称为策略，并且可通过“参数”对话框的“策略”页面（“文件”>“参数”>“策略”）使用。这些策略内置在 Universe 设计工具中。您无法访问或修改它们。第 94 页上的[“选择策略”](#)一节中说明了策略的使用和激活。

也可以创建采用已定义输出结构的 SQL 脚本，以便执行定制的自动 Universe 创建任务。可以从“策略”页面中随其他策略一起选择这些脚本。这些用户定义和定制脚本称为“外部”策略。

本节描述了外部策略及它们的使用。

7.4.1 将外部策略迁移到 Universe 设计工具

在 Universe Designer 6.5 之前的 Universe 设计工具版本中的外部策略是在一个外部文本文件（名为 st<xxxx>.txt 文件）中定义的。此文件不再受支持。

注意：

如果在从 Universe Designer 6.5 进行迁移，外部策略的处理方式在 Universe 设计工具中会保持相同。

要确保以前版本中使用的自定义策略和用户定义的外部策略在 Universe 设计工具中可用，必须执行以下操作：

- 按如下方式编辑新的外部策略文件（<RDBMS>.STG）：
 - 在 XML 编辑器中打开目标 RDBMS 的外部策略文件。
 - 为每个策略创建一个新条目。
 - 对于每个策略，使用 SQL 标记将 SQL 脚本直接复制到 STG 文件中。

或

- 输入文件路径，以便使用 FILE 标记引用外部文本文件中的数据。

第 422 页上的 [“创建外部策略”](#) 一节中详细说明了这两种方法。

- 将帮助文本复制到另一个 XML 文件（<RDBMS><语言>.STG）。第 413 页上的 [“为外部策略创建帮助文本”](#) 一节中对这一点进行了说明。
- 验证是否在常规参数文件（SBO）中声明了外部策略文件，而不是像 Universe Designer 以前的版本一样在参数文件（PRM）中。第 415 页上的 [“验证是否声明了外部策略文件”](#) 一节中对这一点进行了说明。

7.4.2 外部策略概述

下表概述了创建和管理外部策略时所使用的文件，以及这些文件担任的角色。

外部策略管理过程中的角色和文件	说明
外部策略是在外部策略文件（<RDBMS>.STG）中存储并创建的。	<p>XML 文件包含外部策略名称、类型、SQL 脚本或指向外部文本文件（包含数据）的文件引用。文件存储在此处：\$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS>.stg。每个 RDBMS 一个文件。使用此处的 strategy.dtd 文件：\$INSTALLDIR>/dataAccess/RDBMS/connectionServer/strategy.dtd 相关章节：</p> <ul style="list-style-type: none"> 第 416 页上的 “策略文件（STG）具有什么样的结构？” 第 422 页上的 “创建外部策略”
外部策略的帮助文本是在外部策略语言文件中存储并创建的。 (<RDBMS><语言>.STG)	<p>XML 文件包含外部策略文件中每个外部策略的帮助文本。当在“策略”页面上选择某个外部策略时，此文本将出现在该策略的下方。文件存储在此处：\$INSTALLDIR>/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS><language>.stg。使用位于此处的 strategy_localization.dtd 文件：\$INSTALLDIR>/dataAccess/RDBMS/connectionServer/strategy_localization.dtd。相关章节：第 413 页上的 “为外部策略创建帮助文本”。</p>
外部策略文件是在目标 RDBMS 的常规数据访问文件（SBO）中声明的。	<p>XML 文件包含目标 RDBMS 的常规数据访问参数。默认情况下，外部策略文件的名称被设置为“外部策略”参数的值。相关章节：第 415 页上的 “验证是否声明了外部策略文件”</p>

7.4.3 什么是外部策略？

外部策略是一种存储在 .unv 文件外部的 SQL 脚本，并且经过精心构建，以便 Universe 设计工具能够使用它来自动创建对象或联接，并自动在 Universe 中完成表检测任务。外部策略存储在扩展名为 STG 的外部策略文件中。外部策略文件为 XML 格式。每个支持的 RDBMS 都有一个外部策略文件。

外部策略文件存储在以下目录中：

```
$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<rdbms>.stg
```

注意：

您应使用 XML 编辑器来编辑外部策略文件。

7.4.3.1 访问外部策略

外部策略出现在“策略”页面上同时列出了内置策略的下拉列表框中。每个下拉列表框都对应于 XML 文件中的一种策略类型分类。外部策略出现在列表中，策略名称前面带有“外部策略”前缀，如下所示：

外部策略：<策略名称>

例如，用于联接创建的外部策略在策略文件中称为“约束”，以“外部策略：约束”的形式出现在“策略”页面上的“联接”下拉列表中。

7.4.4 为外部策略创建帮助文本

在“策略”页面上，每个选定策略下面都会显示一个注释性附注。这就是策略的帮助文本。无法访问或编辑内置策略的帮助文本。但是，您可以访问和编辑外部策略的帮助文本。

注意：

在 Universe 设计工具以前的版本中，帮助文本包含在策略文本文件的 [HELP] 节中。此部分中的文本现在存储在一个单独的文件（即下文所述的外部策略语言文件）中。

7.4.4.1 外部策略帮助文本存储在一个单独的文件中

外部策略的帮助文本存储在一个名为 <RDBMS><语言>.stg 的单独外部策略语言文件中。例如，oacleen.stg 是 oracle.stg 文件中的策略的帮助文本文件。

可以编辑和定制帮助文本条目。帮助文本应简要描述策略所执行的操作，以便帮助可能不熟悉策略的设计员。

对于出现在外部策略文件中的每个外部策略，应确保对应的帮助文本条目出现在外部策略语言文件中。

所安装 Universe 设计工具的每个语言版本都有一个策略语言文件。外部策略语言文件位于与外部策略文件相同的目录中。例如，如果有 Universe 设计工具的法语版本，Oracle 的外部策略语言文件则为 oraclefr.stg。英语版本为 oacleen.stg。

当您在外部策略文件中创建新的外部策略时，还需要在外部策略语言文件中创建一个帮助文本条目。此条目为使用 Universe 的其他设计员提供了有关外部策略的信息。

示例：Oracle 数据访问驱动程序随附策略的帮助文本条目

oacleen.stg 文件中列出的策略“类和对象”的帮助文本如下所示。这是 oracle.stg 文件中定义的“类和策略”外部策略的帮助文本。

```
<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>
```

7.4.4.2 为外部策略创建帮助条目

为外部策略创建帮助条目：

- 1 在 XML 编辑器打开目标 RDBMS 的外部策略语言文件。目标 RDBMS 的外部策略语言文件位于此处：
\$INSTALLDIR/dataAccess/RDBMS/connectionServer/<数据库>/<数据库><语言>.stg。
例如：
\$INSTALLDIR/dataAccess/RDBMS/connectionServer/oracle/oacleen.stg。
- 2 创建一个新的 Name 元素。
- 3 输入策略的名称。这是您正在为其创建帮助文本的策略。
- 4 创建一个名为“帮助”的 Message ID。此标记包含帮助文本。
- 5 输入帮助文本。
- 6 创建一个名为“名称”的 Message ID。此标记包含要在选择外部策略时出现在策略下拉列表中的名称。
- 7 输入策略名称。

验证、保存并关闭文件。

下一次启动 Universe 设计工具时，帮助文本将出现在选定外部策略的下方。

提示：
为新的 Name 元素创建和设置参数的一种简便方式是：复制现有 Name 元素并填入新策略的新值。

7.4.5 验证是否声明了外部策略文件

外部策略文件是在目标 RDBMS 的常规参数（SBO）文件的 Parameter 部分中声明的。例如，用于 Oracle 的外部策略文件是 oracle.stg。它在 oracle.sbo 文件中具有值 oracle，如下所示：

Parameter (10)

	Name	Abc Text
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 是 Oracle 的外部策略文件的名称。此名称是在 oracle.sbo 文件中声明的。

7.4.5.1 验证是否在 SBO 文件中声明了策略文件

验证是否正确声明了外部策略文件：

- 1

打开目标 RDBMS 的 SBO 文件。
- 2

确保参数“策略名称”设置为外部策略文件的名称。这是默认设置。
- 3

如果未正确设置名称，请输入外部策略文件的正确名称。
- 4

如果进行了修改，请保存并关闭文件。
- 或
- 5

如果未进行任何修改，不必保存即可关闭文件。

注意：

Universe 设计工具以前版本的外部策略在 PRM 文件中声明。这已不再适用于 Universe Designer 6.5。SBO 文件中的“策略文件”参数在默认情况下设置为目标 RDBMS 的外部策略文件的名称。有关将外部策略迁移到 Universe Designer 6.5 的完整信息，请参阅第 412 页上的[“什么是外部策略？”](#)一节。

7.4.6 使用示例外部策略

所有外部策略文件都包含一些 Business Objects 产品随附的现有策略。例如，文件可能包含一个对象策略、一个联接策略和一个表浏览器策略或每种类型的多个策略。

可以定制示例文件，或使用它作为基础来创建新的外部策略。可以定制现有策略，也可以创建您自己的策略。

在修改每个文件之前保存其副本。

7.4.7 策略文件（STG）具有什么样的结构？

每个支持的 RDBMS 都有一个 XML 格式的外部策略（STG）文件。可以将现有策略迁移到此文件，也可以将新的外部策略创建到此文件中。所有外部策略文件都使用以下目录中的策略 dtd（<RDBMS>.dtd）文件：

`$INSTALLDIR/dataAccess/RDBMS/connectionServer`

外部策略 XML 文件中的元素是在外部策略 DTD 文件中定义的。如果使用某些 XML 编辑器（例如 XML SPY），当您创建新的策略元素时，可用的参数将列在一个下拉列表中。

外部策略文件包含一个名为 Strategies 的主要部分。所有外部策略都是在此部分中定义的。Strategies 部分包含以下元素和参数：

文件元素	说明
Strategy	主要元素。所有外部策略都是在此元素内创建的。
Name	外部策略的名称。此名称出现在“策略”页面上的下拉列表中。默认元素。
Type	<p>“策略”页面上外部策略在其中出现的列表。共有三个值：</p> <ul style="list-style-type: none"> • JOIN：联接策略出现在“联接”列表中。 • OBJECT：类和对象策略出现在“类和对象”列表中。 • STRUCT：表检测策略出现在“表”列表中。
SQL	脚本的 SQL 代码。这是选中策略之后 Universe 设计工具运行的 SQL 脚本。SQL 脚本必须采用能使对象和联接创建以及表检测例程正常运行的特定输出格式。有关针对外部策略构建 SQL 脚本的信息，请参阅第 419 页上的 “对象策略 (OBJECT) 的输出格式” 一节。
Connection	指定数据库连接。连接类型必须为个人。
SkipMeasures	设置为 Y 时，将会跳过“快速设计向导”中处理度量创建的屏幕：
File	外部文本文件的文件路径，该文件包含按特定输出格式排列的数据（用于自动创建 Universe）。有关更多信息，请参阅第 424 页上的 “创建数据的文本文件” 一节。

示例：oracle.stg 中的类和对象外部策略

Oracle 的外部策略文件为 oracle.stg。该文件存储在 \$INSTALLDIR/dataAccess/RDBMS/connectionServer/oracle/oracle.stg 目录中。该文件中包含 Universe 设计工具随附的一些示例外部策略。可以定制这些策略，或使用它们作为新策略的模板。

oracle.stg 文件中的外部策略如下所示，该策略可自动将表与类以及列与对象关联。

```
<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),'|',
  '0','|',
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),'|',
  '0','|',
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>

```

7.4.8 策略的输出格式

可以在外部策略文件中的 <SQL> 标记内编写或复制 SQL 脚本。该 SQL 脚本返回的信息顺序和类型取决于创建的是对象、联接还是表策略。对于各种不同类型的策略，Universe 设计工具具有不同的信息需求。

在为策略创建 SQL 脚本时，必须确保为脚本生成的输出与下文描述的输出格式相匹配。

脚本输出被格式化为一系列的列。每个列都对应于一组生成的信息（用于创建对象、联接或表元素）。

本节呈现了以下策略的输出格式：

- 对象策略
- 联接策略
- 表浏览器策略。

7.4.8.1 对象策略（OBJECT）的输出格式

对象策略的输出格式包含 9 个列。必须确保输出包括所有这些列（即使它们包含空值）。所有返回的值必须用竖线 “|” 字符分隔。此竖线字符必须附加在返回值末尾。

列号	列包含...	说明
1	表	表名称格式为 [限定符.][所有者.]表。其中每个名称可以包含多达 35 个字符。如果将此列留空，则将从 Select（第五列）和 Where（第六列）中获取表。
2	列名称	列的名称。
3	类名	类的名称。按如下方式编写子类：类\子类格式。
4	对象名称	对象或条件的名称。如果对象名称为空，则创建类及其说明。
5	Select	Select 语句。
6	Where	如果将 Select 列留空但包括 Where 子句，则会创建预定义的条件及其说明。
7	类型	C（字符），N（数值），D（日期），T（长文本）。如果将该列留空，则默认值为 N。
8	说明	对象的说明。

列号	列包含...	说明
9	限定	D（维）、M（度量）或 I（明细）。如果将该列留空，则默认值为 D。

示例：用于将列说明复制到对象说明的外部对象策略

下面的示例不包含 Where 子句。Where 子句的输出列为空。

<Strategies>

<Strategy Name="Read Column descriptions">

<Type>OBJECT</Type>

<SQL>Select

	列	说明
Table_name, ' ',	1	表名称
Column_name, ' ',	2	列名称
Replace (Table_name, ' _ ', ' '), ' ',	3	将表名称中的下划线替换为类名称中的空白
Replace (Column_name, ' _ ', ' '), ' ',	4	将列名称中的下划线替换为对象名称中的空白。
Table_name ' . ' Column_name, ' ',	5	将表名称拼接到列名称（用句点分隔）。这是 Select 语句。
, ' ',	6	无 Where 子句
Column_type, ' ',	7	从系统表中获取列类型
Column_Desc, ' ',	8	从系统表中获取列说明

' ',' '	9	如果对象类型为空，则默认为维。
---------	---	-----------------

</SQL>

7.4.8.2 联接策略（JOIN）的输出格式

联接策略的输出格式包含以下列：

列号	列包含...	说明
1	表 1	联接中第一个表的名称
2	表 2	联接中第二个表的名称。
3	联接定义	联接的实际定义，采用表 1. 列 1 = 表 2. 列 2 的格式
4	外部类型	外部联接类型。L= 外部向左、R= 外部向右。如果将该列留空，则没有外部联接。
5	基数（可选）	有效值为 11、1N、N1。

7.4.8.3 表浏览器策略（STRUCT）的输出格式

表浏览器策略的输出格式包含以下列：

列号	列包含...	说明
1	限定符	取决于 RDBMS。表限定符为数据库名称或某些其他标识。
2	所有者	取决于 RDBMS
3	表	表、视图或同义词的名称。
4	列	列名称。
5	数据类型	C（字符），N（数值），D（日期），T（长文本）。如果将该列留空，则缺省值为 C。
6	可为空 Y（是）或 N（否）	指明列中是否可存在空值。

7.4.9 创建外部策略

可以使用两种方式创建外部策略：

通过以下方式创建外部策略...	XML 文件中的标记	说明
直接插入 SQL 脚本。	SQL	使用 SQL 标记将策略的 SQL 脚本直接插入外部策略文件。
引用外部文件中的数据	FILE	输入包含策略数据的外部文本文件的文件路径和名称。

以下过程中描述了这两种方法。

7.4.9.1 创建外部策略

直接创建外部策略：

- 1 在 XML 编辑器打开目标 RDBMS 的外部策略文件。目标 RDBMS 的策略文件位于此处：
`$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS>.stg`。
- 2 创建一个新的策略元素。

这是新的策略。如果使用的是 XML 编辑器（例如 XML Spy），则会自动创建策略的 Name、Type 和 SQL 元素。
- 3 输入策略名称。

可以在“Universe 参数”对话框和“快速设计”向导的“策略”选项卡中看到策略的名称。
- 4 输入 TYPE 参数：OBJECT、JOIN 或 STRUCT。

例如，TYPE=OBJECT。
- 5 输入策略的 SQL 语句。第 418 页上的[“策略的输出格式”](#)一节中对 SQL 格式进行了说明。

或

如果要引用包含数据的文本文件，请将 SQL 元素替换为 File 元素。输入数据文件的文件路径，例如，C:\Path\Filename.txt
- 6 如有必要，添加可选的元素并设置值。
- 7 检查 XML 文件的有效性，然后保存并关闭文件。
- 8 验证是否在目标 RDBMS 的常规数据访问文件（<RDBMS>.SBO）中声明了外部策略文件。按如下方式执行此操作：
 - 打开以下目录中的常规数据访问文件（SBO）：
`$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/`
 - 确保将 Strategies File 元素设置为外部策略文件的名称。这是默认值。
 - 如果修改了 SBO 文件，请保存并关闭文件。

外部策略即会出现在“参数”对话框“策略”页面上的“联接”、“对象”或“表”下拉列表中。必须关闭并重新启动 Universe 设计工具，才能看到新建的外部策略。

注意：

如果要添加在“策略”页面上选择外部策略时出现在策略下方的帮助文本，您应将此文本添加到一个单独的文件，即位于与外部策略文件相同目录中的外部 <RDBMS><语言>.STG 文件。第 413 页上的[“为外部策略创建帮助文本”](#)一节中说明了如何为外部策略文件添加帮助文本。

7.4.10 创建数据的文本文件

可以创建包含外部策略的数据的文本文件。在创建外部策略时，您可以输入文本文件的文件路径和名称（而不是直接插入 SQL）。可以在外部策略文件中插入 FILE 元素，并将值设置为文件路径和名称。

SQL 脚本的输出必须遵循策略、对象、联接或表的类型的正确格式。第 418 页上的 [“策略的输出格式”](#) 一节中对输出格式进行了说明。

所有格式均由用制表符分隔的信息列组成。

7.4.11 在 Universe 设计工具中应用外部策略

可以按如下方式应用外部策略：

1. 确保在“参数”对话框的“策略”页面中选择了要使用的外部策略。

例如，

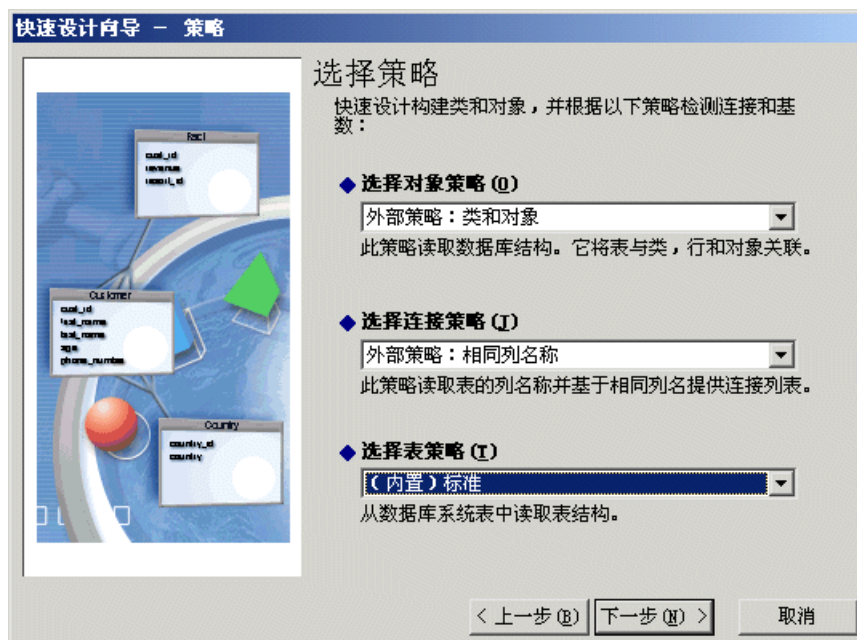
- 要插入使用对象策略抽取的对象，请从“插入”菜单中选择“候选对象”命令。
- 要插入从联接策略中派生的联接，请从“工具”菜单中选择“检测联接”命令。
- 要插入使用表浏览器策略抽取的表，请从“插入”菜单中选择“表”命令。

注意：

选择联接策略之后，Universe 设计工具将使用该策略来检测候选联接和基数。可以选择应用建议的联接或基数。如果要基于选定策略自动应用候选联接和基数，您必须在“选项”对话框的数据库页面（“工具”>“选项”>“数据库”）上选择对应的创建选项。有关更多信息，请参阅第 96 页上的 [“使用策略的自动创建功能”](#) 一节。

7.4.11.1 在“快速设计向导”中选择策略

可以从“快速设计向导”中选择您设置的外部策略。为此，必须从向导的欢迎窗口中单击选项“单击此处选择策略”。



7.5 使用分析函数

Universe 设计工具支持对特定 RDBMS 使用分析函数。分析函数在 RedBrick 中称为 RISQL 函数，在 Teradata 中称为 OLAP 函数。可使用 Universe 设计工具为 Universe 中的对象定义分析函数。

Web Intelligence 用户还可以使用分析函数来执行通常无法使用 InfoView 报表功能进行的数据分析。有关如何在 Web Intelligence 中使用这些函数以及其他函数的更多信息，请参阅文档《在 Web Intelligence 中使用函数、公式和计算》中的“使用智能度量计算值”一节。

本节说明如何针对以下 RDBMS 为 Universe 中的对象定义分析函数、RISQL 函数和 OLAP 函数：

- 第 428 页上的 [“IBM DB2 UDB 和 Oracle”](#)
- 第 432 页上的 [“RedBrick \(RISQL 函数\)”](#)
- 第 435 页上的 [“Teradata \(OLAP 函数\)”](#)

7.5.1 什么是分析函数？

分析函数是一种在结果集上执行分析任务的函数（可将结果集划分为有序的行组合或分区）。

在 Universe 设计工具中，可以利用分析函数定义对象，以计算一个或多个分区中的名次、累积聚合和比率。视 RDBMS 而定，也可以在分区中定义想在其上应用分析的行范围。

有关分析函数的完整说明，请参阅您的 RDBMS 文档。

7.5.2 使用分析函数有哪些优点？

在 Universe 设计工具中使用分析函数定义对象，可以为 Web Intelligence 用户带来以下好处：

- 减少了工作量。使用分析函数定义的对象可以执行通常需要在报表级别使用扩展语法的数据分析。
- 增加了功能。在 InfoView 中，通常无法开展许多数据分析任务，如计算移动平均和应用高级聚合处理。现在，使用分析函数的对象允许 Web Intelligence 用户执行以前不可能进行的高级数据分析。
- 提升了查询性能。计算在服务器上进行。

7.5.3 支持哪些分析函数族？

可以为以下函数族定义分析函数：

- 排名
- 累积聚合
- 比率、报告比率或报告聚合

7.5.4 如何在 Universe 设计工具中使用分析函数？

通过在对象的 SELECT 语句中定义分析函数来使用分析函数。

每个参数 (PRM) 文件中的 RDBMS 部分都列出了可在 SELECT 语句中使用的分析函数。此列表可能未包含在为分析函数支持的各个 RDBMS 中每个函数族的所有可用函数。

7.5.4.1 什么是 PRM 文件？

PRM 文件是一个参数文件，用于在 Web Intelligence 产品中配置 Universe 创建和 SQL 查询生成。每个支持的 RDBMS 都有一个 PRM 文件。PRM 文件位于以下文件夹中：

```
<INSTALLDIR>\dataAccess\RDBMS\connectionServer\<rdbms>\
```

有关修改参数文件的完整信息，请参阅《数据访问指南》。

在使用某个分析函数之前，应验证它是否在 PRM 文件中列出。如果未列出，可以向列表添加函数名称。Universe 设计工具随后将支持在 Select 语句中对对象使用该函数。有关更多信息，请参阅第 429 页上的 [“在 PRM 文件中验证和添加分析函数支持”](#) 一节。

7.5.4.2 将分析函数用于各个 RDBMS

将会说明将分析函数用于以下各个 RDBMS 的信息：

- 在 Select 语句中可以用于分析函数、RISQL 函数和 OLAP 函数的语法。
- 如何验证和修改 PRM 文件，以确保支持未列出的分析函数。
- 有关使用分析函数的特定于 RDBMS 的规则和限制。
- 在编辑 Select 语句时自动插入分析函数语法。

7.5.5 IBM DB2 UDB 和 Oracle

可以将相同的分析函数语法用于这两个 RDBMS。

7.5.5.1 为 DB2、UDB 和 Oracle RDBMS 定义 SELECT 语句

可以为对象在 SELECT 语句中定义分析函数。您需要在 Select 语句的其中一个编辑框中键入语法。

注意：

可以自动输入语法，方法是将分析函数添加到“编辑 Select 语句”对话框中的“函数”列表中。要使某个函数在“函数”列表中可用，需要将此分析函数添加到 .prm 文件的 [FUNCTIONS] 部分。有关更多信息，请参阅第 438 页上的 [“在 Select 语句中自动插入语法”](#) 一节。

分析函数由关键字 OVER 标识；例如：

```
RANK() OVER (PARTITION BY calender.cal_year ORDER BY SUM(telco_facts.total_billed_rev)DESC)
```

跟在 OVER 关键字后面的子句定义分区，以及在结果表中如何对行进行排序。

以下说明了用于每个分析函数族的语法：

函数族	语法	说明
排名	<code>RANK() OVER (PARTITION BY arg1 ORDER BY arg2 ASC/DESC)</code>	<ul style="list-style-type: none">• arg1 是可选的。如果未包含任何参数，则默认情况下分区为整个结果集。• arg2 是必需的。名次基于此参数值。• ASC/DESC 决定是按升序还是降序对值进行排序。ASC 是默认值。
Window 聚合	<code>SUM(arg1) OVER (PARTITION BY arg2 ORDER BY arg3)</code>	<ul style="list-style-type: none">• arg1 是累积聚合所基于的参数。• arg2 是 reset 子句。它是可选的。• arg3 是 group 子句。它是可选的。
报告 聚合	<code>RATIO_TO_REPORT(arg1) OVER (PARTITION BY arg2)</code>	<ul style="list-style-type: none">• arg1 是比率所基于的参数。• arg2 是 reset 子句。它是可选的。

使用 Window 子句

对于 Window 聚合函数族，也可以在 arg3 后面定义一个定义 window 大小范围的 <window 子句>。
例如：

```
<window frame units> ::= ROW | RANGE <window frame start> ::= UNBOUNDED PRECEDING | <window frame preceding> | CURRENT ROW <window frame between>
```

对于 BETWEEN 子句语法和其他 window 大小定义，请参阅您的 RDBMS 文档。

7.5.5.2 在 PRM 文件中验证和添加分析函数支持

已更新了 IBM DB2 UDB 和 Oracle 的 PRM 文件，以支持分析函数的使用。

但是，PRM 文件可能未包含在目标 RDBMS 中可用的所有分析函数。在使用某个分析函数之前，应验证它是否在 PRM 文件的 RDBMS 部分中列出，并且在必要时将其添加到列表中。

可以按如下方式执行此操作：

向 Oracle 或 IBM DB2 PRM 文件添加对分析函数的支持：

- 1 浏览到 Business Objects 路径中的 Data Access 目录。
- 2 在文本编辑器中打开 RDBMS 的 PRM 文件。
- 3 滚动到 PRM 文件的 RDBMS 部分。
- 4 验证是否存在以下参数和值：

PRM 中的参数和值	说明
OVER_CLAUSE = Y	生成适当的 SQL (OVER_CLAUSE)。
RISQL_FUNCTIONS = <所使用函数的列表>	可用的分析函数。

- 5 如果要使用未列出的分析函数，请在列表的结尾键入该函数的名称。例如，要使用 RATIO_TO_REPORT，必须按以下所示将其添加到列表中：

```
[RDBMS]
{GENERAL}
...
OVER_CLAUSE=Y
RISQL_FUNCTIONS=RANK,SUM,AVG,COUNT,MIN,MAX,
VARIANCE,STDDEV,RATIO_TO_REPORT
```

- 6 保存任何修改并关闭文件。

需要重新启动 Universe 设计工具，对 PRM 文件所做的任何更改才会生效。

7.5.5.3 DB2、UDB 和 Oracle RDBMS 分析函数的使用规则

将分析函数用于 DB2 UDB 和 Oracle 时，以下规则适用：

规则	说明
分析函数不能出现在 GROUP BY 子句中。	<p>GROUP BY 子句中使用在分析函数中定义的聚合函数（如 SUM），但将不会使用分析函数（如 RANK）。</p> <p>为了确保在 GROUP BY 子句中没有使用分析函数，它们被列在 PRM 文件中 RISQL FUNCTIONS 参数的后面。必须将其前面的 OVER_CLAUSE 设置为 Y。这是默认设置。</p>
分析函数必须不能生成 GROUP BY 子句。	<p>如果向 PRM 文件中的 Functions 部分添加分析函数（以填充“编辑 SQL”对话框中的函数列表），则必须确保将 GROUP CLAUSE 设为 N。这将阻止它生成 GROUP BY 子句。有关更多信息，请参阅第 438 页上的 “在 Select 语句中自动插入语法” 一节。</p>
如果某个分析函数使用聚合函数，则该分析函数使用的所有维将出现在 GROUP BY 子句中。	<p>例如，RANK() OVER (PARTITION BY 年度 ORDER BY SUM(销售))。GROUP BY 子句将包含年度维（即使在查询中只使用了名次函数）。</p>

7.5.5.4 在 Oracle 和 DB2 中使用分析函数的限制

在 IBM DB2 UDB v7.1 和 Oracle 8.1.6 中使用分析函数时，有以下限制：

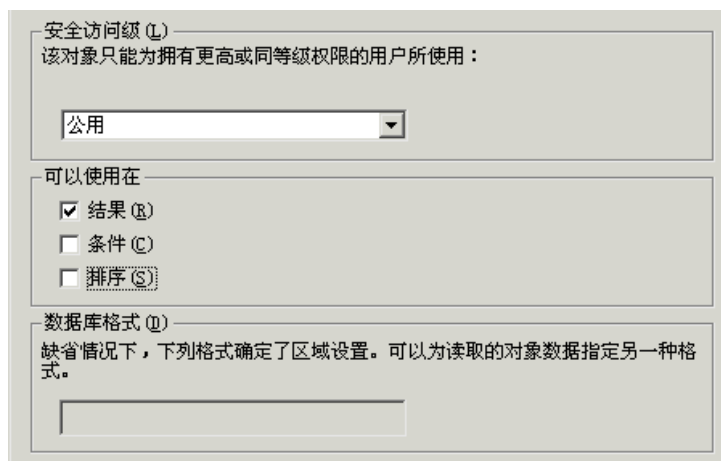
- 不能在也使用分析函数的对象的定义中使用 @Prompt 和 @Variable 函数。
- 不支持将分析函数用作用户对象。如果向 PRM 文件中的 Functions 部分添加分析函数（以填充“编辑 SQL”对话框中的函数列表），则必须确保将 IN MACRO 设为 N。
- 不能将使用分析函数的对象用作条件或用于排序中。如果最终用户尝试使用这些对象来定义条件，他们将收到 SQL 错误消息。可以按以下所述编辑对象属性，以防止最终用户在条件或排序中使用对象：

防止在条件或排序中使用分析对象

防止在条件或排序中使用分析函数：

1. 在 Universe 设计工具中右键单击对象。

- 2 从上下文菜单中选择“对象属性”。
将显示“编辑属性”对话框。
- 3 清除“可以使用在”组框中的“条件”和“排序”复选框。



- 4 单击“确定”。

7.5.6 RedBrick (RISQL 函数)

以下章节说明在 Universe 设计工具中如何使用 RISQL 函数。

7.5.6.1 为 RedBrick RISQL 函数定义 SELECT 语句

可以为对象在 SELECT 语句中定义分析函数。您需要在 Select 语句的其中一个编辑框中键入语法。

注意：

可以自动输入语法，方法是将 RISQL 函数添加到“编辑 Select 语句”对话框中的“函数”列表中。要使某个函数在“函数”列表中可用，需要将此 RISQL 函数添加到 PRM 文件的 [FUNCTIONS] 部分。有关更多信息，请参阅第 438 页上的[“在 Select 语句中自动插入语法”](#)一节。

以下说明了用于每个 RISQL 函数族的语法

函数族	语法	说明
求秩 (RANK)	RANK(arg1) 例如: RANK(SUM(telco_facts.total_billed_rev))	arg1 是必需的。名次基于此参数。
聚合函数族 (CUME、MOVINGAVG、MOVINGSUM)	MOVINGSUM(arg1, 数字) 例如: MOVINGSUM (COUNT(complaints.id), 2)	<ul style="list-style-type: none"> arg1 是必需的。累积聚合基于此参数。 数字是可选的。这是用于总和的前面行的数量。
比率 (RATIOTOREPORT)	RATIOTOREPORT(arg1) 例如: RATIOTOREPORT (SUM(telco_facts.total_billed_rev))	arg1 是必需的。比率基于此参数。

7.5.6.2 在 PRM 文件中验证和添加 RSQL 函数支持

PRM 文件可能未包含所有可用的 RSQL 函数。在使用某个 RSQL 函数之前，应验证它是否在 PRM 文件的 RDBMS 部分中列出，并在必要时将其添加到列表中。可以按如下方式执行此操作：

向 Redbrick PRM 文件添加对分析函数的支持：

- 1 浏览到 Business Objects 路径中的 Data Access 目录。
- 2 在文本编辑器中打开 RDBMS 的 PRM 文件。
- 3 滚动到 PRM 文件的 RDBMS 部分。
- 4 验证是否存在以下参数和值：

PRM 中的参数和值	说明
OLAP_CLAUSE = WHEN	应用条件。
RSQL_FUNCTIONS = <所使用函数的列表>	可用的分析函数。

下面显示了一个示例：

```
[RDBMS]
(GENERAL)
...
OVER_CLAUSE=WHEN
RISQL_FUNCTION= RANK, CUME, MOVINGSUM, MOVINGAVG, RATIOREPORT, TERTILE
```

- 5 如果要使用未列出的 RISQL 函数，请在列表的结尾键入该函数的名称。
- 6 保存任何修改并关闭文件。

需要重新启动 Universe 设计工具，对 PRM 文件所做的任何更改才会生效。

7.5.6.3 使用 RISQL 函数的规则

使用 RISQL 函数时，以下规则适用：

规则	说明
RISQL 函数不能出现在 GROUP BY 子句中。	<p>GROUP BY 子句中使用在 RISQL 函数中定义的聚合函数（如 SUM），但将不会使用分析函数（如 RANK）。</p> <p>为了确保在 GROUP BY 子句中没有使用 RISQL 函数，它们被列在 PRM 文件中 RISQL FUNCTIONS 参数的后面。必须将其前面的 OVER_CLAUSE 设为 WHEN。这是默认设置。</p>
RISQL 函数必须不能生成 GROUP BY 子句。	<p>如果向 PRM 文件中的 Functions 部分添加 RISQL 函数（以填充“编辑 SQL”对话框中的函数列表），则必须确保将 GROUP CLAUSE 设为 N。这将阻止它生成 GROUP BY 子句。有关更多信息，请参阅第 438 页上的“在 Select 语句中自动插入语法”一节。</p>
可以在条件中使用 RISQL 函数	会生成 WHEN 子句

7.5.6.4 在 RedBrick 中使用分析函数的限制

使用 RSQL 函数时，存在以下限制：

- 不支持 RESET BY 子句。
- 不支持 SORT BY 子句。有关如何通过编辑对象属性来避免最终用户使用该对象的过程，请参阅“在 Oracle 和 DB2 中使用分析函数的限制”一节。

7.5.7 Teradata (OLAP 函数)

以下章节说明在 Universe 设计工具中如何使用 OLAP 函数。

7.5.7.1 为 Teradata OLAP 函数定义 Select 语句

比率函数在 Teradata V2R3 中不可用。可以在对象的 Select 语句中定义 OLAP 函数。您需要在 Select 语句的其中一个编辑框中键入语法。

有关如何使用函数列表中的函数来自动执行语法输入的信息，请参阅“在 Oracle 和 DB2 中使用分析函数的限制”一节。

以下说明了用于每个 OLAP 函数族的语法：

函数族	语法	说明
求秩 (RANK)	RANK(arg1 DESC/ASC) 例如: RANK(invoice_line.nb_guests)	<ul style="list-style-type: none"> arg1 是必需的。名次基于此参数。参数可以是对象或对象列表。 <p>注意: 不能使用将聚合对象 (sum、avg、min、count) 用作 arg1 的对象。</p> <ul style="list-style-type: none"> DESC/ASC 指定名次顺序。ASC 是缺省顺序。
聚合函数族 (CSUM、MAVG、MDIFF、MLINREG、MSUM)	CSUM(arg1 DESC/ASC) 例如: CSUM(invoice_line.nb_guests)	<ul style="list-style-type: none"> arg1 是必需的。累积聚合基于此参数。参数可以是对象或对象列表。 DESC/ASC 指定结果行的顺序。ASC 是缺省顺序。

7.5.7.2 在 PRM 文件中验证和添加 OLAP 函数支持

已更新了 Teradata 的 PRM 文件，以支持 OLAP 函数的使用。但是，PRM 文件可能未包含所有可用的 OLAP 函数。在使用某个 OLAP 函数之前，应验证它是否在 PRM 文件的 RDBMS 部分中列出，并在必要时将其添加到列表中。可以按如下方式执行此操作：

向 Teradata PRM 文件添加对分析函数的支持

- 1 浏览到 Business Objects 路径中的 Data Access 目录。
- 2 在文本编辑器中打开 RDBMS 的 PRM 文件。
- 3 滚动到 PRM 文件的 RDBMS 部分。
- 4 验证是否存在以下参数和值：

PRM 中的参数和值	说明
OLAP_CLAUSE = QUALIFY	应用条件。

PRM 中的参数和值	说明
RISQL_FUNCTIONS = <所使用函数的列表>	可用的分析函数。

下面显示了一个示例：

```
[RDBMS]
(GENERAL)
...
OVER_CLAUSE=QUALIFY
RISQL_FUNCTION= RANK, CSUM, MAVG, MDIFF, MLINREG, MSUM, QUANTILE
```

- 5 如果要使用未列出的 RISQL 函数，请在列表的结尾键入该函数的名称。
- 6 保存任何修改并关闭文件。

需要重新启动 Universe 设计工具，对 PRM 文件所做的任何更改才会生效。

7.5.7.3 使用 OLAP 函数的规则

使用 OLAP 函数时，以下规则适用：

- OLAP 函数不能出现在 GROUP BY 子句中。为了确保在 GROUP BY 子句中没有使用 OLAP 函数，它们被列在 PRM 文件中 RISQL_FUNCTIONS 参数的后面。必须将其前面的 OVER_CLAUSE 设为 QUALIFY。这是默认设置。
- 不能在同一个查询中将使用 OLAP 函数的对象和使用聚合函数的对象结合使用。
- 可以在条件中使用 OLAP 函数。会生成 QUALIFY 子句。
- 可以在 SORT BY 子句中使用 OLAP 函数。

7.5.7.4 在 Teradata 中使用分析函数的限制

使用 OLAP 函数时，存在以下限制：

- 不支持 RESET BY 子句。
- 不能在子查询中使用 OLAP 函数。

- 不能在另一个函数所在的同一 Select 语句中使用 OLAP 函数。
- OLAP 函数不能基于另一个函数。
- 不支持将 OLAP 函数用作用户对象。

7.5.8 在 Select 语句中自动插入语法

通过将分析函数添加到“编辑 Select 语句”对话框中的“函数”列表框，可以自动插入分析函数语法。

通过将分析函数添加到目标 RDBMS 的相应 PRM 文件中 [FUNCTION] 部分下的函数列表，对“函数”列表框进行填充。

将函数添加到 PRM 文件后，函数即会在“编辑 Select 语句”对话框中的“函数”列表框内变得可用。双击函数语法时，即将定义的语法插入到编辑框中。

将分析函数添加到 PRM 文件中时，必须设置以下值：

参数	说明
GROUP = N	分析函数、RISQL 函数和 OLAP 函数不能生成 GROUP BY 子句。通过设置值 N，即可阻止在 GROUP BY 子句中使用分析函数。
只对于 IBM DB2 UDB v. 7.1 和 ORACLE 8.1.6: IN_MACRO = N	这会阻止在用户对象中使用 DB2 UDB 和 Oracle 的分析函数。对于 RedBrick 和 Teradata，可以将此值设为 Y。

可以按以下所述将分析函数添加到 PRM 文件中的 [FUNCTION] 部分：

向 PRM 文件添加分析函数：

- 1 浏览到 Business Objects 路径中的 Data Access 目录。
- 2 在文本编辑器中打开 RDBMS 的 PRM 文件。
- 3 滚动到 PRM 文件的 [FUNCTION] 部分。
- 4 复制一个现有函数并将其粘贴到列表的末端。
- 5 为新粘贴的函数键入一个唯一编号，然后为您想添加到列表的分析函数适当地修改值。
- 6 将 GROUP 值设为 N。

如果您使用的是 IBM DB2 UDB 或 ORACLE，则将 IN_MACRO 值设为 N。

例如：

```
(n)
NAME: RANK
TRAD:
HELP: Return the rank of
TYPE=N
IN_MACRO=N
GROUP=N
SQL=
```

7. 保存并关闭 PRM 文件。

需要重新启动 Universe 设计工具，才会应用更改。

注意：

重新启动 Universe 设计工具时，所添加的分析函数的语法将出现在适当的“类型”节点（“数字”、“字符”或“日期”）下。

7.6 使用 SQL Prefix 函数

SQL Prefix 函数通过使用 Universe 参数“Begin_SQL”，将 SQL 指令插入到 BusinessObjects 生成的 SQL 语句之前。这样可确保这些指令在所有生成的 SQL 语句之前执行。此函数适用于任何支持在 SELECT 语句之前传递参数的数据库。下面列出一些示例：

- Teradata：使用“QUERY_BAND”代表事务处理（请参阅 Teradata 文档）
- Oracle：开启只读事务处理
- Netezza：触发优化选项

要设置 SQL Prefix 函数，请为 Universe 设置 SQL_prefix 参数。

相关主题

- 第 106 页上的“[在用户界面中设置的 SQL 参数](#)”

7.6.1 使用 BEGIN_SQL Universe 参数为 SQL 语句添加前缀

数据库应支持在 SELECT 语句之前传递参数。

使用 BEGIN_SQL 参数可以在每次生成 SQL 语句时都使用相同参数为 SQL 语句添加前缀。

- 1 打开“Universe 参数”对话框。
- 2 单击“参数”选项卡。
- 3 在“参数”列表中，选择 BEGIN_SQL 参数，然后输入相应的前缀命令。
- 4 保存设置。
- 5 保存 Universe。

示例：

本示例使用 BEGIN_SQL 参数和 Teradata。查询绑定了用户 ID 和应用程序 ID 以便进行报告。在“Universe 参数”对话框的“参数”窗格中，BEGIN_SQL 参数设置如下：

```
BEGIN_SQL = SET QUERY_BAND = ' UserId=DG12234;AppId=TRM;' FOR TRANSACTION;
```

在运行查询时将执行以下两个 SQL 语句：

1) BEGIN_SQL 语句：

```
SET QUERY_BAND = 'UserId=DG12234;AppId=TRM;' FOR TRANSACTION;
```

2) 主查询 SQL 结果语句：

```
SELECT
  RESORT_COUNTRY.COUNTRY, sum(INVOICE_LINE.DAYS * INVOICE_LINE.NB_GUESTS
    * SERVICE.PRICE)
FROM
  COUNTRY RESORT_COUNTRY, INVOICE_LINE, RESORT_COUNTRY.COUNTRY
```

7.7 优化数组提取参数

CS.CFG 文件中的“数组提取”参数使您可以设置在 FETCH 过程中允许的最大行数。CFG 文件是一个 XML 文件，它指定了在依据数据库运行查询时 Business Objects 产品所使用的某些参数的默认值。

“数组提取”参数决定了网络上的包大小。例如，如果将“数组提取”设为 20，而您计划检索 100 行，则将会为了检索数据而执行五次提取。

某些数据源不允许修改 FETCH 大小。在这种情况下，将会在一次 FETCH 中返回所有行。如果想检索二进制长对象（BLOB），则应将“数组提取大小”设为 1。

如果您的网络允许您发送大型的数组提取，则可以设置较大的新值（可以在 1 到 999 的范围内设置值）。这将加速 FETCH 过程，并减少查询处理时间。

7.7.1 修改数组提取参数

修改“数组提取”参数：

- 1 在 XML 编辑器中打开 CS.CFG 文件。

CFG 文件存储在以下目录中：

<INSTALDIR>\dataAccess\RDBMS\connectionServer。

- 2 搜索“数组提取”参数。
- 3 设置参数值。保存并关闭 CFG 文件。
- 4 重新启动 Universe 设计工具。

7.8 分配表的权重

表的权重是反映表中有多少行的量度。较轻的表的行数少于较重的表。缺省情况下，BusinessObjects 从较轻的表到较重的表（即行数最少的表到行数最多的表）对表进行排序。这决定了 SQL 语句的 FROM 子句中的表顺序。

在数据库级别对表进行排序时所依从的顺序取决于数据库。例如，Sybase 使用与 BusinessObjects 相同的顺序，但 Oracle 使用相反的顺序。将会为大部分数据库优化 SQL，但不会为 Oracle 进行优化（在 Oracle 中，最小的表排在排序顺序中的第一位）。

因此，如果您使用了 Oracle 数据库，则可以通过倒转 BusinessObjects 对表进行排序的顺序来优化 SQL。要这样做，您必须更改数据库的相关 PRM 文件中的某个参数。

7.8.1 修改 PRM 文件以分配表的权重

修改 PRM 文件以分配表的权重：

- 1 在 XML 编辑器中打开数据库的 PRM 文件。

PRM 文件存储在以下目录中：

<INSTALDIR>\dataAccess\RDBMS\connectionServer\<rdbms>\

例如，用于 Oracle 的文件是以下目录中的 oracle.prm：

<INSTALDIR>\dataAccess\RDBMS\connectionServer\oracle\oracle.prm

- 2 在该文件的 Configuration 节中查找 REVERSE_TABLE_WEIGHT 参数。

- 3 将 Y 改为 N。
例如，此参数显示为 REVERSE_TABLE_WEIGHT=N。
如果文件中没有此行，则缺省值为 Y。
- 4 这将迫使 BusinessObjects 从行数最多的表到行数最少的表对表进行排序。
- 5 保存并关闭 .PRM 文件。
- 6 重新启动 Universe 设计工具，以便应用对 .PRM 文件的更改。

7.9 修改为某个表返回的行数

也可以在 Universe 设计工具中手动更改任何表的行数。要查看任何表中的行数，请选择“查看”>“表中的行数”。行数将显示在每个表符号的左下角。可以按以下所述修改此数字：

7.9.1 修改返回的行数

修改为某个表返回的行数：

- 1 在 Universe 设计工具中打开一个 Universe。
- 2 右键单击相关的表
- 3 从上下文菜单中选择“表中的行数”。
将出现“表行计数”对话框。
- 4 选择“手动修改表的行计数”单选按钮。
对话框左边将出现一个文本框。
- 5 在此文本框中键入一个数字。这是您想用于该表的行数。
- 6 单击“确定”，然后保存该 Universe。

7.10 使用快捷联接

快捷联接链接两个已在公共路径中联接的表。可以使用快捷联接减少在查询中使用的表数。有关更多信息，请参阅第 187 页上的[“快捷联接”](#)一节。

注意：

快捷联接将不会创建环路。

第 8 章 使用 OLAP Universe

8.1 关于 OLAP Universe

8.1.1 什么是 OLAP Universe?

OLAP Universe 是已依据 OLAP 多维数据集或查询生成的 BusinessObjects Universe。此 Universe 是通过与 OLAP 数据源的选定连接自动创建的。

一旦创建了 Universe，就可以将它导出到中央管理服务器（CMS）中作为任何其它 Universe。然后，Web Intelligence 用户可以使用此 Universe 运行查询和创建报表。

采用以下方式可以生成和维护 OLAP Universe：

- 要生成 OLAP Universe，请首先选择 OLAP 数据源。

注意：

要建立到 OLAP 数据源的安全连接，需要生成 Universe 或刷新其结构的用户需具有该连接的“下载”权限。此权限在 CMC 中由管理员设置。

- 使用“新建连接”向导定义数据源的连接，然后为新 Universe 选择连接。

Universe 设计工具将自动生成此 Universe。OLAP 结构被直接映射到 Universe 中的类、度量、维、详细信息以及过滤器。Universe 结构显示在 Universe 窗格中。

- 用户可以将 OLAP Universe 保存和导出到 CMS 中。
- 用户可以修改任何 OLAP Universe 元素。
- 使用“更新 OLAP Universe”向导可以管理 OLAP Universe 的生命周期。向导会自动刷新 Universe 结构以反映 OLAP 数据源中的变更。向导可以区分生成的对象与手动添加或修改的对象，使用户可以保留在 Universe 设计工具中手动进行的更改。

相关主题

- 第 445 页上的[“可用于创建 Universe 的 OLAP 数据源 ”](#)
- 第 451 页上的[“关于 OLAP 数据源的连接”](#)
- 第 462 页上的[“支持 OLAP Universe 的 Universe 设计工具功能”](#)
- 第 476 页上的[“关于 OLAP Universe 生命周期管理”](#)

8.1.2 可用于创建 Universe 的 OLAP 数据源

可以依据以下 OLAP 数据源自动创建 OLAP Universe:

- SAP NetWeaver Business Warehouse (BW)
- Microsoft Analysis Services (MSAS) 2000
- Microsoft Analysis Services (MSAS) 2005
- Hyperion Essbase

注意:

为了从 SAP BusinessObjects OLAP 产品（包括 Universe 设计工具、Web Intelligence 胖客户端和 Web Intelligence）连接到 Essbase OLAP 数据源，请确保在承载这些 SAP BusinessObjects OLAP 产品的计算机上正确安装并配置 Essbase 客户端中间件。也就是说，应确保创建 Essbase 客户端环境变量 ARBORPATH 和 ESSBASEPATH，并设置为 Windows 系统环境变量（而不是 Windows 用户环境变量）。

一个依据一个多维数据集或查询自动生成的 Universe。OLAP Universe 支持在 Universe 中使用单一多维数据集。

相关主题

- 第 445 页上的[“SAP NetWeaver Business Warehouse \(BW\) 数据源”](#)
- 第 511 页上的[“如何在 Universe 中映射和使用 SAP NetWeaver BW 对象”](#)
- 第 449 页上的[“OLAP Universe 支持的 MSAS 功能”](#)
- 第 520 页上的[“将 MSAS 多维数据集映射到 Universe 元素的方法 ”](#)
- 第 450 页上的[“OLAP Universe 支持的 Essbase 功能”](#)
- 第 519 页上的[“如何将 Essbase 多维数据集映射到 Universe 元素”](#)

8.1.2.1 SAP NetWeaver Business Warehouse (BW) 数据源

在基于 NetWeaver BEx 数据源创建 OLAP Universe 时，可以直接基于信息块/MultiCube 或基于在任何信息提供者之上启用的 BEx 查询构建 Universe。信息提供者可以是：

- 一个信息块
- 一个多个块或多信息提供者
- 一个操作数据存储（ODS）
- 一个 InfoSet

相关主题

- 第 446 页上的“[SAP NetWeaver Business Warehouse \(BW\) 信息块用作数据源](#)”
- 第 446 页上的“[SAP NetWeaver BW 查询作为数据源](#)”
- 第 447 页上的“[查询作为推荐的数据源](#)”

8.1.2.1.1 SAP NetWeaver Business Warehouse (BW) 信息块用作数据源

在构建 OLAP Universe 时，支持将以下类型的信息块作为数据源：

- 标准和事务处理信息块：数据和元数据实际存储在相同的 SAP NetWeaver Business Warehouse (BW) 系统中
- 远程信息块：数据实际存储在远程系统中

注意：

尽管完全支持，但对于使用 Ad-Hoc 查询、报表和分析的方案，建议不要基于远程信息块构建和部署 Universe。此类体系结构通常预计不会满足交互查询的查询性能要求。

- 多个块和多信息提供者

注意：

在基于多个块或多信息提供者构建和部署 Business Objects Universe 的方式与基于信息块构建和部署 Universe 的方式相同。

信息块中的所有特性、层次结构、关键值（包括时间和单位）在 Universe 中均为可视。

8.1.2.1.2 SAP NetWeaver BW 查询作为数据源

SAP NetWeaver BW 客户使用 BEx 查询来访问 SAP Business Explorer 的前端。

注意：

若要担当数据源并通过至 Business Objects Universe 的 OLAP 接口提供，必须为 OLE DB for OLAP 发布 BEx 查询。在“查询属性”对话框的“扩展模式”选项卡上，允许从外部访问 SAP NetWeaver BW 查询设计者中的 BEx 查询。

BEx 查询中所有选定为行、列和自由特性的 InfoObject 均在 Universe 中可见，其中包括特性、层次结构、关键值、结构和变量。

InfoSet 和操作数据存储 (ODS) 可通过 BEx 查询显示在 Universe 中。

基于 ODS 的查询

ODS 可通过 BEx 查询显示在 Universe 中。

在聚合到信息块之前，常常先使用 ODS 对象来管理事务处理级别的详细数据。将 ODS 对象包括在 NetWeaver 数据存储设计中，这样可以最小化信息块的大小，改进加载和查询性能。

注意：

ODS 通常是一个大型的明细关系结构。通过 OLAP BAPI 接口访问 ODS 并不会得到理想的查询性能。请考虑以下替代方式，以满足最终用户希望快速传送报表的要求：

- 创建通过 BAPI 调用直接访问 ODS 的权限
- 在 Web Intelligence 中使用直接 SQL 访问 ODS 表

基于 InfoSet 的查询

InfoSet 可通过 BEx 查询显示在 Universe 中。

SAP NetWeaver BW 中有时会定义 InfoSet 以报告主数据。

注意：

可以通过使 Universe 基于信息块的方式来报告主数据，从而避免使用 InfoSet 和 BEx 查询。这两种方法之间的主要区别在于，基于信息块报告主数据会将数据范围限定于有效事务。

相关主题

- 第 447 页上的“[查询作为推荐的数据源](#)”

8.1.2.1.3 查询作为推荐的数据源

基于以下原因，建议将 BEx 查询用作生成 Business Objects Universe 的数据源：

- 如下表汇总的信息显示，在信息块级别上并不能检索到所有的 SAP NetWeaver BW 元数据功能。

NetWeaver BW 元数据功能	SAP OLAP 业务应用程序编程接口 (BAPI) 支持等级
特性 (包括时间和单位)	信息块/BE _x 查询
层次结构	信息块/BE _x 查询
基本关键值	信息块/BE _x 查询
导航属性	仅限 BE _x 查询
显示属性	信息块/BE _x 查询
计算所得关键值/公式	仅限 BE _x 查询
受限关键值	仅限 BE _x 查询
自定义结构	仅限 BE _x 查询
变量	仅限 BE _x 查询

- BE_x 查询使数据建模环境得到灵活扩展。而更改信息块会费劲些。
- BE_x 查询提供了创建符合最终用户要求的自定义数据源所需的重要功能。

虽然将 BE_x 查询用作数据源有一定的优势，然而不必为每个报表提供一个 BE_x 查询，也不必为每个现有 BW 查询提供一个 Universe。为将维护成本降到最低，请将重点放在实施策略上，以限制满足所有 Ad-Hoc 查询和报告需求时所需的 BE_x 查询和 Universe 的最终数量。请记住以下几点，以降低所需的 Universe 数量：

- 当 Web Intelligence 是前端工具时，用户不受 BE_x 查询中的输出格式限制。
- 在使用根据大型 BE_x 查询创建的 OLAP Universe 时，性能不受直接影响。没有插入到 Web Intelligence 查询中的 OLAP Universe 对象不会对查询性能产生直接影响。

注意：

Business Objects 建议为 Ad-Hoc 查询和报告范围内的每个信息块或 MultiCube 提供一些 BE_x 查询（少则一个，多则数个）。然后在每个 BE_x 查询基础之上构建一个 Universe。

8.1.2.1.4 SAP NetWeaver BW 多语言 Universe

通过 Web Intelligence，用户可以利用 SAP NetWeaver BW 的多语言功能。要实施多语言环境，NetWeaver BW 系统必须包括多语言元数据和多语言数据。

必须为解决方案支持的每种语言创建一个 Universe。创建 Universe 连接时所用的语言确定生成 Universe 时所用的语言。

用户的 SAP 身份验证确定返回至查询的数据所用的语言。用户必须使用 SAP 身份验证登录到 InfoView，并为从 SAP 服务器返回的结果指定所需的语言。

结果集语言依赖于 SAP 的 Unicode 支持。如果 SAP 系统不包含以所需语言表示的数据，Web Intelligence 中不提供以此语言表示的数据。如果没有在 NetWeaver BW 中翻译说明，Web Intelligence 将恢复为显示技术名称，而不显示说明。

8.1.2.1.5 在 Universe 设计工具中使用 SAP NetWeaver BW 的先决条件

在根据 SAP NetWeaver BW 数据源创建 Universe 时，可以启用 SSO（单一登录）来查看时间。SSO 允许用户使用自己的 SAP 凭据登录到 SAP BusinessObjects Enterprise 并利用 SAP 身份验证。

要为基于 SAP 的 OLAP Universe 启用 SSO，必须安装 SAP 集成并配置 SAP 安全插件。

一旦配置了 SAP 安全集成，就可以使用 SAP 凭据启动 Universe 设计工具。请按使用 SAP 用户 ID 配置安全集成时的定义，将 SAP 系统 ID 和 SAP 客户端 ID 拼接在一起以生成 BusinessObjects Enterprise 用户名。

有关更多信息，请参阅《Business Objects XI Integration for SAP Solutions 安装指南》和《Business Objects XI Integration for SAP Solutions 用户指南》。

8.1.2.2 OLAP Universe 支持的 MSAS 功能

下表概要介绍了根据 MSAS 数据源生成的 Universe 的 MSAS 功能支持级别。

MSAS 元数据功能	OLAP Universe 支持级别
多维数据集	支持
本地多维数据集	支持
虚拟多维数据集 (MSAS 2000)	支持
透视 (MSAS 2005)	支持
维	支持
虚拟维 (MSAS 2000)	支持

MSAS 元数据功能	OLAP Universe 支持级别
层次结构	支持
级别	支持
级别属性	支持
特性 (MSAS 2005)	支持
度量	支持
度量组 (MSAS 2005)	支持
计算所得度量	支持
显示文件夹 (MSAS 2005)	支持
KPI (MSAS 2005)	不支持
操作	不支持
数据库排序顺序	必须在 Web Intelligence 中定义自定义排序顺序
回写	不支持

相关主题

- 第 520 页上的“[将 MSAS 多维数据集映射到 Universe 元素的方法](#)”

8.1.2.3 OLAP Universe 支持的 Essbase 功能

下表概要介绍了根据 Hyperion Essbase 数据源生成的 Universe 的 Essbase 功能支持级别。

Essbase 元数据功能	OLAP Universe 支持级别
块存储模式	支持
聚合存储模式	支持
混合模式	不支持
别名表	支持
维	支持
特性维	支持
重复成员	支持

Essbase 元数据功能	OLAP Universe 支持级别
生成	支持
级别	不支持
用户定义属性 (UDA)	不支持
动态时间序列 (DTS)	不支持
Essbase Integration Services (EIS) 贯穿钻取	不支持
替代变量	不支持
链接分区	不支持
链接报表对象 (LRO)	不支持
数据库排序顺序	必须在 Web Intelligence 中定义自定义排序顺序
回写	不支持

相关主题

- 第 519 页上的“[如何将 Essbase 多维数据集映射到 Universe 元素](#)”

8.2 定义 OLAP 数据源的连接

8.2.1 关于 OLAP 数据源的连接

要生成 OLAP Universe，必须首先定义 OLAP 数据源的连接。请为要用于创建 Universe 的每个多维数据集或查询定义一个连接。

请使用“新建连接”向导来定义连接。该向导会引导您完成下列创建连接的步骤：

- 在 Universe 设计工具中启动“新建连接”向导。
- 为连接命名，然后选择数据库中间件。
- 定义连接的登录参数。这些参数随所选的数据库中间件而有所不同。
- 选择用于创建 Universe 的多维数据集或查询。

- 定义连接期限。
- 定义定制参数。这些参数随所选的数据库中间件而有所不同。

定义连接是创建 OLAP Universe 流程的第一步。定义连接后，Universe 设计工具会自动生成 Universe。

注意：

如果从“工具”菜单的连接列表中定义连接，则必须在单独的步骤中创建 Universe。

相关主题

- 第 452 页上的[“启动“新建连接”向导”](#)
- 第 453 页上的[“为 OLAP 连接选择数据库中间件”](#)
- 第 454 页上的[“SAP BW OLAP 连接的登录参数”](#)
- 第 455 页上的[“MSAS OLAP 连接的登录参数”](#)
- 第 457 页上的[“为 Essbase 连接定义登录参数”](#)
- 第 457 页上的[“为 OLAP 连接选择源多维数据集或查询”](#)
- 第 458 页上的[“定义 OLAP 连接的配置参数”](#)
- 第 459 页上的[“为 Essbase 连接定义定制参数”](#)

8.2.2 启动“新建连接”向导

要启动“新建连接”向导，请执行以下操作之一：

启动位置	操作
“新建 Universe”图标	单击“新建 Universe”图标，然后在“Universe 参数”框的“定义”页上单击“新建...”。
“文件”菜单	从空会话中，选择“文件” > “参数”，然后在“Universe 参数”框的“定义”页上单击“新建...”。
快速设计向导	<p>如果“快速设计”向导已启用，将在启动 Universe 设计工具时自动启动。在“快速设计向导”的步骤 1 中单击“新建...”。</p> <p>注意： 如果已禁用此向导，请选择“文件” > “新建”。如果在选择“文件” > “新建”时向导未启动，请选择“工具” > “选项”。在“选项”对话框的“常规”页中，选择“文件 / 新建启动快速设计向导”复选框。单击“确定”，然后选择“文件” > “新建”。</p>
“工具”菜单	选择“工具” > “连接”。在“向导连接”对话框中单击“添加...”。

8.2.3 为 OLAP 连接选择数据库中间件

在“新建连接”向导的“数据库中间件选择”页中，输入连接的名称，选择类型以及连接的数据库中间件。

数据库中间件选择参数	说明
连接类型	<p>选择“安全”以控制对连接的访问（推荐）。</p> <p>选择“共享”，则不控制所有用户对连接的连接。</p> <p>选择“个人”，则只允许 Universe 创建者访问。可以使用个人连接以仅访问本地计算机上的个人数据。</p>
连接名称	输入连接名称。
过滤存储过程的网络层	“过滤存储过程的网络层”参数不用于 OLAP 连接。
可用数据访问驱动程序列表。	<p>此页列出对应于数据访问驱动程序注册表项的数据库和中间件。</p> <p>展开目标数据库的节点以显示该数据库的受支持的中间件。</p> <p>展开中间件节点以显示 OLAP 中间件的 Business Objects 数据访问驱动程序。</p> <p>选择此数据访问驱动程序。</p>

8.2.4 SAP BW OLAP 连接的登录参数

“新建连接”向导的“登录参数”对话框可包含以下参数：

登录参数	说明
身份验证模式	<ul style="list-style-type: none"> 使用指定的用户名和密码：使用登录详细信息作为身份验证。 使用 BusinessObjects 凭据映射：提示用户输入与其 BusinessObjects 帐户关联的数据库用户密码以刷新报表。这通过 dbuser 和 dbpass 参数设定。这些是在管理层设定的。有关设置此选项的信息，请参阅《BusinessObjects Enterprise 管理员指南》。 查看中刷新报表时使用单一登录：如果选定此项，用于访问 CMS 的用户名和密码将自动用作数据库登录参数。有关设置单一登录（SSO）的信息，请参阅《BusinessObjects Enterprise 管理员指南》。

登录参数	说明
使用 SNC（如果可用）	选择该复选框即使用 SNC。
客户端	用于在 SAP NetWeaver BW 系统上标识该客户端的数字（必需）。
用户名	如果“身份验证模式”是“使用指定的用户名和密码”，用于访问 OLAP 服务器的用户名。
密码	如果“身份验证模式”是“使用指定的用户名和密码”，用于访问 OLAP 服务器的密码。
语言	<p>将用于连接的语言。</p> <p>注意： 连接语言确定在生成 Universe 时所用的语言。</p>
保存语言	<p>将用于连接的语言：</p> <ul style="list-style-type: none"> 如果选中“保存语言”，将使用“语言”字段中的值。 如果清除“保存语言”，将使用用户会话中的值。
“登录模式”或“服务器类型”	<p>选择“应用程序服务器”，以便直接连接 SAP 服务器且不使用负载平衡。</p> <p>选择“消息服务器”以利用 SAP 的负载平衡功能。</p>
应用程序服务器	选择或者输入 SAP 应用程序服务器的名称或 IP 地址（在采用程序服务器登录模式时必需）。
系统编号	输入系统编号，例如 00（在采用程序服务器登录模式时必需）。
系统 ID	<p>在使用“消息服务器”登录模式时，输入“消息服务器”、“登录组”，如有需要还应输入“系统 ID”。</p>
登录组	
消息服务器	

8.2.5 MSAS OLAP 连接的登录参数

“新建连接”向导的“登录参数”对话框可包含以下参数：

登录参数	说明
身份验证模式	<ul style="list-style-type: none"> 使用指定的用户名和密码：使用登录详细信息作为身份验证。 使用 BusinessObjects 凭据映射：提示用户输入与其 BusinessObjects 帐户关联的数据库用户密码以刷新报表。这通过 dbuser 和 dbpass 参数设定。这些是在管理层设定的。有关设置此选项的信息，请参阅《SAP BusinessObjects Business Intelligence 平台管理员指南》。 查看中刷新报表时使用单一登录：如果选定此项，用于访问 CMS 的用户名和密码将自动用作数据库登录参数。有关设置单一登录（SSO）的信息，请参阅《SAP BusinessObjects Business Intelligence 平台管理员指南》。
服务器	<p>输入以下信息之一：</p> <ul style="list-style-type: none"> 在 MSAS 服务器中提供和配置的 MSAS 库的 URL。 MSAS 数据源的服务器名称。 MSAS 多维数据集文件的完整路径文件名。在双引号中输入完整路径文件名，例如： “Z:\All cubes\test.cub” <p>注意： 如果多维数据集文件所在的主机系统不是 SAP BusinessObjects 主机，则这两台计算机必须具有一个共享连接。用户必须在 SAP BusinessObjects 主机上直接创建到多维数据集文件的连接。</p>
用户名	如果“身份验证模式”是“使用指定的用户名和密码”，用于访问 OLAP 服务器的用户名。
密码	如果“身份验证模式”是“使用指定的用户名和密码”，用于访问 OLAP 服务器的密码。
语言	将用于连接的语言。

8.2.6 为 Essbase 连接定义登录参数

在“新建连接”向导的“登录参数”页中，指定用于连接到 Essbase 数据库的登录详细信息。

登录参数	说明
身份验证模式	<p>选择“使用指定的用户名和密码”，则需要用户在使用连接时输入登录信息。要使 Essbase 安全性与 BusinessObjects Enterprise 保持同步，请分别在“用户名”和“密码”中输入 Essbase DBuser 和 Essbase DBpass 对应的登录凭据。</p> <p>选择“使用 BusinessObjects 凭据映射”，则可以使用用户的 BusinessObjects Enterprise 登录凭据进行连接。</p> <p>注意： 不支持对 Essbase 连接使用“查看中刷新报表时使用单一登录”选项。</p>
用户名	输入 Essbase DBuser。
密码	输入 Essbase DBpass。
服务器	输入 Essbase 服务器名称。

8.2.7 为 OLAP 连接选择源多维数据集或查询

多维数据集浏览器显示可供目标服务器使用的 OLAP 多维数据集。

展开多维数据集节点以显示可用的多维数据集和查询。浏览器提供以下工具以帮助执行搜索：

多维数据集浏览器工具	说明
收藏夹	包含为快速访问而选择的多维数据集链接的文件夹。要将多维数据集添加到“收藏夹”，请在 OLAP 多维数据集浏览器中右击多维数据集，然后从上下文菜单中选择“添加到收藏夹”。
搜索	搜索可用的多维数据集或查询的名称以查找文本字符串。在文本框中键入字符串，然后单击“搜索”。查找到的每个实例均会突出显示。单击“搜索”可以继续进行搜索。
\$INFOCUBE 文件夹	对于 SAP NetWeaver BW 数据源，在名为“\$INFOCUBE”的文件夹中将信息块和多个块分组在一起。

选择用于创建 Universe 的多维数据集或查询。

8.2.8 定义 OLAP 连接的配置参数

在“新建连接”向导的“配置参数”页中，定义用于管理连接期限的连接参数。用户可以在创建连接时接受默认设置，然后以后进行修改。

配置参数	说明
连接池模式	使用“连接池模式”和“池超时”参数定义连接期限。
池超时	<p>默认情况下，“连接池模式”设置为“保持连接活动”，并且“池超时”的默认值为 10 分钟。</p> <p>注意： Business Objects 建议保留默认的连接期限。如果将“连接池模式”设置为在每次事务处理之后断开连接，Universe 构建流程将明显慢下来。如果在每次事务处理之后断开连接，还会影响关键的最终用户工作流，例如使用分层的值列表。</p> <p>在使用 SAP NetWeaver BW 时，连接期限会产生显著影响。</p> <p>然而，连接期限还会影响使用 BEx 查询中的变更更新现有 Universe 的操作。这是因为每次建立与 SAP NetWeaver BW 的连接时，OLAP BAPI 接口都会在客户端建立一个元数据缓存。只有当连接关闭时，才会清空缓存。</p> <p>要尽量降低元数据缓存与 SAP BEx 查询更新不同步的风险，可以将“池超时”从 10 分钟更改为 1 分钟。</p> <p>在并行执行 BW 查询编辑以及将新 Universe 映射到这些查询时，建议先关闭 Universe 设计工具（以便关闭 Universe 连接和清空元数据缓存），然后构建任何新 Universe，以反映刚才在 BEx 查询端执行的更改。</p>
数组提取大小	使用“数组提取大小”参数可设置 FETCH 过程中允许的最合适行数。
数组绑定大小	“数组绑定大小”和“登录超时”参数不能用于 OLAP 连接。
登录超时	

8.2.9 为 Essbase 连接定义定制参数

在“新建连接”向导的“定制参数”页中，指定别名表，并选择将哪个维用作生成 Universe 时的“度量”维。

登录参数	说明
别名表	要根据默认表之外的别名表生成 Universe，请从列表中选择别名表。
度量维	选择要用作“度量”维的那个维。Universe 设计工具生成用户选择作为 Universe 度量的维的成员。

8.3 自定义 OLAP Universe

8.3.1 使用附加参数创建 OLAP Universe

此功能只适用于 OLAP Universe，允许用户在使用 MSAS、SAP NetWeaver Business Warehouse (BW) 或 Essbas 创建 OLAP Universe 时，定义附加元数据参数。

创建 OLAP Universe 时，可定义以下参数：

通用 OLAP 选项	说明
将技术名称生成为明细	可设置应用程序，将技术名称生成为 Universe 中维的明细对象。当生成 Universe 时，可创建指向技术名称的明细对象。

SAP OLAP 选项	说明
将度量聚合委派给数据库	可设置应用程序，将度量的聚合函数委派给数据库。
替换前缀 L00、L01	Universe 级别前缀表示某一对象的层次结构级别。L00 级别是顶级或根级，而 L01 是下一级别。在“新 Universe 向导”中，可使用不同的前缀替换 OLAP Universe 级别前缀。例如，保留级别编号，不过可以用级别替换前缀“L”。在“新前缀”字段中键入自己的前缀。此前缀位于 OLAP Universe 中所有级别之前。
将级别 00 重命名为“所有”	如果将“生成级别 00”设置为“否”，将禁用此选项。下一次生成 Universe 时，可将顶级（根级）L00 重命名为“所有”。
生成级别 00	此选项仅适用于 SAP 特性。可针对特性和层次结构取消激活此选项。将始终为层次结构变量生成级别 00。生成或更新 Universe 时，可重新生成级别编号（L00、L01、L02...）。级别编号将附加到级别名称（例如“Monthly Sales_L01”）后。此操作对 Web Intelligence 报表颇有帮助，其中“所有”级别用于聚合查询结果。这样一来，便不必在 Web Intelligence 报表中创建聚合字段。

注意：

如果在创建 Universe 时禁用了“生成级别 00”，则不会为层次结构生成根级别。

8.3.2 针对 OLAP Universe 定义 OLAP 选项

使用 OLAP 选项可定义如何从 OLAP 源生成特定的 Universe 元数据。可以通过“选项”对话框的“OLAP”页面（“工具” > “选项” > “OLAP”）选择 OLAP 选项。根据选定选项，可在 Universe 中提取并创建所有 OLAP 源的内容。可选择以下 OLAP Universe 生成选项：

通用 OLAP 选项	说明
将技术名称生成为明细	可设置应用程序，将 Universe 的技术名称生成为属性。在生成 Universe 时，此设置将创建指向技术名称的对象。

SAP OLAP 选项	说明
将度量聚合委派给数据库	可设置应用程序，令其为使用聚合函数的度量生成委派的度量。生成 Universe 时，会将任何使用聚合函数的度量设置为“委派给数据库”。
替换前缀 L00、L01	Universe 级别前缀表示某一对象的层次结构级别。L00 级别是顶级或根级，而 L01 是下一级别。在“新 Universe 向导”中，可使用不同的前缀替换 OLAP Universe 级别前缀。例如，保留级别编号，不过可以用级别替换前缀“L”。在“新前缀”字段中键入自己的前缀。此前缀位于 OLAP Universe 中所有级别之前。
将级别 00 重命名为“所有”	如果将生成级别 00 设置为否，将禁用此选项。下一次生成 Universe 时，可将顶级（根级）L00 重命名为“所有”。此操作对 Web Intelligence 报表颇有帮助，其中“所有”级别用于聚合查询结果。这样一来，便不必在 Web Intelligence 报表中创建聚合字段。
生成级别 00	此选项仅适用于 SAP 特性。可仅针对特性取消激活此选项。针对层次结构及层次结构变量，将始终生成级别 00。 生成或更新 Universe 时，可重新生成级别编号（L00、L01、L02...）。级别编号附加到级别名称（例如“Monthly Sales_L01”）后

8.3.3 定义 OLAP Universe 中的对象

可以使用 SQL 编辑器为对象定义 SELECT 语句或 WHERE 子句，并为 OLAP Universe 对象插入 MDX 运算符和函数。在 SQL 编辑器中可用的选项和函数取决于基础数据库。

8.3.4 支持 OLAP Universe 的 Universe 设计工具功能

OLAP Universe 是自动创建的。一旦创建 OLAP Universe，就可以修改其中的任何 Universe 元素。

对于生成的 OLAP Universe，支持以下 Universe 设计工具功能：

- 隐藏、复制和重命名类及对象（维、明细和度量）
- 插入新类和对象（维、明细和度量）
- 编辑对象的格式
- 编辑对象的数据类型
- 定义主键和外键
- 解析维、明细和度量对象的 MDX 语法
- 检查 Universe 完整性
- 编辑层次结构
- 创建级联值列表
- 为值列表定义委派搜索，以允许用户限制在运行查询时的值列表加载操作
- 使用变量的默认值
- 定义包含委派投影函数的度量（智能度量）
- 刷新 Universe 结构

此外，以下功能只可用于 OLAP Universe：

- 创建计算所得的度量（仅限 SAP Netweaver BW 和 MSAS）
- 创建预定义条件
- 定义可选提示

所有基于 OLAP Universe 的对象均通过“索引感知”生成。如果对象的某个层次结构中存在重复值，通过“索引感知”可消除值列表中的不一致。例如，“Paris”（巴黎）在层次结构中出现两次，一次在父项“France”（法国）下面，另一次在父项“Texas”（德克萨斯）下面。如果用户选择“France”（法国）下的“Paris”（巴黎），则只返回与法国巴黎相关的行。

对于 OLAP Universe，不支持以下 Universe 设计工具功能：

- 用户不能在 OLAP Universe 中设置行级别安全授权。
- 用户不能编辑 OLAP Universe 中的值列表。
- 用户不能查看和编辑 Universe 实体关系模式，因为没有为 OLAP Universe 生成的模式。

相关主题

- 第 467 页上的[“OLAP Universe 中的计算所得度量”](#)
- 第 470 页上的[“OLAP Universe 中的预定义条件”](#)
- 第 475 页上的[“OLAP Universe 中的可选提示”](#)
- 第 463 页上的[“投影函数“委派给数据库””](#)

8.3.5 投影函数“委派给数据库”

在 Universe 中，任何度量均可以具有投影函数（Sum、Min、Max、Count 和 Avg）。当报表中显示的维数小于查询结果集中的维数时，投影函数用于在 Web Intelligence 中对度量进行本地聚合。

非叠加度量（例如比率、平均和权重）只能显示在与查询结果集相同的聚合级别上。因此，在 Universe 中非叠加度量的投影函数通常设置为“无”。

使用投影函数“委派给数据库”可以将非叠加度量的聚合委派给数据库服务器。在 Web Intelligence 中这些称为智能度量。在对象属性的属性页上，智能度量的投影函数设置为“委派给数据库”。有关如何在 Web Intelligence 中使用这些函数以及其他函数的更多信息，请参阅文档《在 Web Intelligence 中使用函数、公式和计算》中的“使用智能度量计算值”一节。

注意：

对于基于 MSAS 和 Essbase 数据源的 OLAP Universe，默认情况下通过设置为“委派给数据库”的投影函数创建 Universe 中的所有度量。

注意：

在使用基于已设置聚合感知的度量的智能度量时，应注意以下限制：强烈建议确保度量定义中使用的各聚合表具有一致的数据（就明细值而言聚合值是准确的），否则智能度量会生成不一致的数据。例如，如果为智能度量使用了一个年度聚合表和一个日聚合表，就全部年度而言，年度聚合表的数据与日聚合表的数据保持一致，然而对于当前年度，年度表可以为空，而日聚合表在每日基础上的数据是准确的。在这种情况下，使用基于当前年度和日聚合表的智能度量的报表会给出不连贯的结果。

示例：智能度量

在本例中，一个查询包含两个维（Country 和 Region）和三个度量（Order Amount、Amount Delivered，以及 % Delivered）。

L01 Region	Amount Delivered	Order Amount	% Delivered
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		总和：	299.992

“% Delivered”的总和不正确，因为它等于“% Delivered”列的合计。

如果在 Universe 中将此度量的投影函数设置为“委派给数据库”，当用户刷新报表时，Web Intelligence 会连接到数据库以计算正确的值。

L01 Region	Amount Delivered	Order Amount	% Delivered
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		总和:	299.992
		总计:	99.997

注意:

使用某些函数时必须特别谨慎，例如比率函数 (Average)。在计算列的平均值时，如果函数配置不正确，其行为可能会异常。

例如，SQL 函数 `sum(Shop_facts.Margin)/sum(Shop_facts.Quantity_sold)` 可能得到意外的结果。如果配置不正确，它会计算每个单元格的平均值，并返回所有平均值的总和。要更正结果，必须按以下所示配置函数的参数：

- 。 1 转到函数的“编辑属性”选项。
- 。 2 从“函数”下拉列表中为选项“选择在聚合时如何投影此度量”选择“委派给数据库”。
- 。 3 保存更改。

相关主题

- 第 466 页上的[“为度量设置聚合投影”](#)

8.3.6 为 OLAP Universe 设置委派度量

可设置应用程序，令其为使用聚合函数的度量生成委派的度量。生成 Universe 时，会将所有使用聚合函数的度量设置为“委派给数据库”。

相关主题

- 第 483 页上的[“替换 OLAP Universe 级别前缀”](#)
- 第 483 页上的[“为 OLAP Universe 重新生成级别 00”](#)
- 第 483 页上的[“将级别 L00 重命名为“所有””](#)

8.3.7 为度量设置聚合投影

在创建度量时，您必须指定将聚合函数投影到报表上的方式。

将在查询过程的以下两个级别聚合度量对象的返回值：

- 查询级别。使用推导出的 SELECT 语句对数据进行聚合。
- 微多维数据集到块级别。将数据从微多维数据集投影到报表中的块时。度量的这种投影功能允许在微多维数据集中以本地方式进行聚合。

注意：

微多维数据集是用于在将查询投影到报表中之前表示查询所返回数据的一种概念性方式。它表示由 Business Objects 报表产品保留在内存中的返回值。块级别是用户使用返回数据创建的二维报表。用户可以选择使用保留在微多维数据集中的全部数据或只使用部分数据来创建报表。用户还可以在微多维数据集中的返回值上执行聚合函数（本地聚合），以便在报表上创建新的值。

这两个聚合级别按如下方式放置在查询过程中：

- 用户在 Web Intelligence 中创建查询。
- Web Intelligence 从查询中推导出 SQL，然后将 SELECT 语句发送到目标数据库。
- 数据返回到微多维数据集。这是第一个聚合级别。
- 微多维数据集将聚合数据投影到报表上。在“查询”窗格中分解出需要聚合到更低级别的数据。这是第二个聚合级别。

当您开始进行查询时，Select 语句的结果集存储在微多维数据集中，并且微多维数据集中包含的所有数据将被投影到块中。由于数据是从微多维数据集中包含的最低级别投影的，因此不会发生投影聚合。

但是，当您使用“查询”窗格以便仅从微多维数据集中投影部分数据时，则必须进行聚合才能在较高级别显示度量值。

例如，在前一示例中，如果不将年度数据投影到块中，则需要将与“年度”相关的三个行缩减为一个行以显示该地区的总“销售收入”，因此使用了加总聚合。

可以在度量的“编辑属性”表单的“属性”页面上设置投影聚合（右键单击“对象”>“对象属性”>“属性”）。

投影聚合与 SELECT 聚合不同。

相关主题

- 第 463 页上的[“投影函数“委派给数据库”](#)”

8.3.8 OLAP Universe 中的计算所得度量

可以在 Universe 中创建计算所得的度量以限制查询。OLAP Universe 中的计算所得度量与非 OLAP Universe 中度量对象上的定义类似，但有一点不同，它使用嵌入在 XML 标记中的 MDX 函数，而不是使用 SQL 来定义限制。

计算所得度量可用于这些 OLAP 数据源：

- SAP NetWeaver Business Warehouse (BW)
- MSAS 2000 和 2005

计算所得度量可用在过滤器或 WHERE 子句中。

计算所得度量表达式的语法

计算所得度量的语法包括嵌入在 <EXPRESSION></EXPRESSION> 标记中的计算。

在计算所得度量的表达式中允许使用 Universe 设计工具函数，例如：

- @Select
- @Prompt
- @Variable
- @Where

注意：

计算所得度量的表达式不能包括 @Aggregate_Aware 函数。检查完整性函数将验证 XML 语法和以上列出的任一 @ 函数，包括已插入到 MDX 语句的函数。但是不会分析 MDX 语句。

表达式中可以使用常量，例如 “10” 或 “ABC”。

计算所得度量可以引用任何 OLAP 元数据：

- 度量
- 维
- 维级别
- MDX 表达式

计算所得度量表达式的建议

由于以下原因，请使用 @Select（度量名称），而不是度量定义：

- 在查询时解析 @Select。
- 计算所得度量可以引用 @Select 函数中的其它计算所得度量。
- 已检查 @Select 函数中的对象有效性。

为每个对象定义生成并设置索引感知。

使用对象或明细的引用，后者的定义中引用级别或特性的技术名称或唯一名称。

示例：计算所得度量的表达式

```
<EXPRESSION>@Select(Key Figures\Order Amount)*@Select(Key Figures\Order Quantity)</EXPRESSION>
```

相关主题

- 第 468 页上的[“在 OLAP Universe 中创建计算所得度量”](#)

8.3.8.1 在 OLAP Universe 中创建计算所得度量

在 OLAP Universe 中创建计算所得度量：

- 1 在 Universe 设计工具中，打开一个 OLAP Universe。
- 2 在此 Universe 中插入新度量对象。
- 3 在“Where:”框中，输入或粘贴对象定义作为 XML/MDX 表达式。
- 4 单击“分析”以检查对象定义，并修正任何错误。
- 5 单击“确定”以保存对象定义。
- 6 选择“工具” > “检查完整性”。

完整性检查会验证 XML 语法和所有 Universe 设计工具 @ 函数。

相关主题

- 第 467 页上的[“OLAP Universe 中的计算所得度量”](#)

8.3.9 关于多维数据集查询的 MDX 函数

使用 MDX 编辑器定义多维数据集查询。

向 OLAP Universe 添加新对象或预定义过滤器时，可使用针对特定数据源连接的一系列 MDX 表达式。

可用表达式库存储于 .prm 连接文件中。针对某对象打开“编辑属性”窗格以及针对某查询打开“编辑 SELECT 语句”窗格时，“函数”窗格中将显示可用的表达式。要在 SELECT 或 WHERE 语句中插入表达式，请在语句中单击需要插入该表达式的位置，然后双击适合的表达式。

OLAP Universe MDX 字典 — 函数列表 (PRM 文件)

向 OLAP Universe 添加新对象或预定义过滤器时，针对表达式中使用的正确 OLAP 连接（SAP 或 MSAS），可在对象和过滤器编辑器中使用 MDX 函数（主要是成员函数）和运算符。有关如何设置 SAP 或 MySQL（sap.prm、sqlsrv_as.prm）的说明，请参阅《数据访问指南》。可用的函数和运算符取决于 Universe 的连接类型。PRM 文件针对每个连接提供此函数列表。不提供所支持函数的完整列表，仅提供最常用的函数。

以下 MDX 运算符可用于查询。

- Equal
- NotEqual
- InList
- NotInList
- Greater
- GreaterOrEqual
- Less
- LessOrEqual
- Between
- NotBetween
- Like
- NotLike

下表列出了编辑条件时可用的部分 MDX 文件夹函数的示例。可用函数取决于基础数据库。

- 集合函数（ADDCALCULATEDMEMBERS、ALLMEMBERS……）
- 统计/数字函数（AGGREGATE、AVG……）
- 导航/成员函数（ANCESTOR、ASCENDANTS……）
- 元数据函数（AXIS、HIERARCHY……）

8.3.10 过滤器和 WHERE 语句的 XML 语法

本节介绍用于在 OLAP Universe 中定义 WHERE 子句或过滤器语句的 XML 语法。必须手动添加 FILTER 或 FILTER EXPRESSION 标记，然后手动或使用 Universe 设计工具的 MDX 编辑器在标记之间输入表达式。

- 使用单个对象定义时，请使用 <FILTER= "your_object_definition">。请在双引号内输入对象定义。
- 使用包含一个或多个对象的复杂 MDX 表达式时，请使用 <FILTER EXPRESSION= "yourcomplexMDX_expression">。请在双引号内输入表达式。

单个过滤器对象的语法如下：

```
<FILTER = "your_object_definition"><CONDITION OPERATORCONDITION="yourOperator"><CONSTANT
VALUE="your_Value"/></CONDITION></FILTER>
```

其中：

- yourMDX_expression 是单个对象定义，置于双引号内。
- CONSTANTVALUE 是 CONSTANT CAPTION 或 CONSTANT TECH_NAME
- yourOperator 是过滤器表达式运算符（equals、inlist……）。使用 InList 运算符时，必须为列表中的每个项目插入一个 CONSTANT CAPTION 或 CONSTANT TECH_NAME 元素。
- 使用 CONSTANT CAPTION 时，your_Value 是定义的过滤器值，而使用 CONSTANT TECH_NAME 时则是对象标识符。

列出三个国家/地区时，针对使用 InList 运算符的单个过滤器对象的语法如下：

```
<FILTER= "your_object_definition "><CONDITION OPERATORCONDITION="InList"><CONSTANT CAP
TION="England"/><CONSTANT CAPTION="France"/><CONSTANT CAPTION="Germany"/></CONDITION></FIL
TER>
```

针对复杂过滤器表达式以及过滤值的 TECH_NAME 的语法如下：

```
<FILTER EXPRESSION="yourComplex_MDX_Expression"><CONDITION OPERATORCONDITION="Equal"><CON
STANT TECH_NAME="1"/></CONDITION></FILTER>
```

示例：使用过滤器表达式中的计算所得成员进行过滤

```
<FILTER EXPRESSION="IIF([OCALYEAR].CurrentMember > "2000", 1, 0)"><CONDITION OPERATOR
CONDITION="Equal"><CONSTANT CAPTION="1"/></CONDITION></FILTER>
```

8.3.11 OLAP Universe 中的预定义条件

OLAP Universe 中的预定义条件与非 OLAP Universe 中的条件类似，但有一点不同，就是它使用 XML，而不是使用 SQL 来定义 WHERE 子句。可以手动声明过滤器，或者通过使用预定义过滤器编辑器来完成。

8.3.11.1 预定义过滤器选项的 XML 语法

预定义条件的语法

单个预定义条件可能包含多个用 AND 和 OR 运算符联接的过滤器。默认情况下，所有过滤器使用 AND 运算符联接。要用 OR 联接过滤器，必须使用 AND 和 OR 运算符标记。

预定义过滤器定义中允许使用函数 @Select、@Prompt 和 @Variable。

预定义过滤器可以包含一个或多个提示。提示可以是必需的，也可以是可选的。

示例：对预定义条件使用 AND 和 OR 标记

```
<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>
```

8.3.11.2 在 OLAP Universe 中手动创建预定义条件

创建预定义条件：

- 1 在 Universe 设计工具中，打开 OLAP Universe，然后单击“Universe”窗格底部的条件单选按钮。将出现“Universe”窗格的“条件”视图。它包含 Universe 中类的树形视图。
- 2 右键单击某个类，然后从上下文菜单中选择“条件...”。
- 3 在“Where: ”框中编辑 XML 模板过滤器。

模板过滤器的格式如下：

```
<FILTER "[Level Object definition]">
  <CONDITION OPERATORCONDITION="Operator">
    <CONSTANT Level Attribute="Value"/>
  </CONDITION>
</FILTER>
```

按如下所示替换模板中的元素：

模板元素：	可能的值：
Level Object definition	输入过滤器中隐含的维级别或度量。输入对象定义，而不是输入对象名称。
Operator	输入以下项之一： <ul style="list-style-type: none"> • Equal • NotEqual • Greater • Less • GreaterOrEqual • LessOrEqual • Between • NotBetween • InList • NotInList • Like • NotLike
Level Attribute	输入以下项之一： <ul style="list-style-type: none"> • NAME • CAPTION • TECH_NAME • DESCRIPTION
Value	输入值或提示。为每个 CONSTANT 标记定义一个值。

已编辑预定义条件的示例：

```
<FILTER KEY="[OD_DIV].[LEVEL01]">
  <CONDITION OPERATORCONDITION="InList">
    <CONSTANT CAPTION="Internal"/>
    <CONSTANT CAPTION="Service"/>
  </CONDITION>
</FILTER>
```

- 单击“分析”以检查语法，并修正任何错误。
- 单击“确定”以保存条件。

相关主题

- 第 470 页上的[“OLAP Universe 中的预定义条件”](#)
- 第 475 页上的[“OLAP Universe 中的可选提示”](#)

8.3.11.3 关于预定义过滤器编辑器

“预定义过滤器”编辑器用于在 OLAP Universe 中编辑预定义过滤器。可使用它选择对象、运算符、值列表、提示、函数，以及其他可用于为 OLAP Universe 定义过滤器的可选元素。

在过滤器的“条件属性”面板中，可手动键入过滤器表达式，或单击“>>”打开“预定义过滤器”编辑器。打开编辑器后，可在过滤器表达式中插入 @Prompt：在过滤器表达式的相应位置右键单击，然后从快捷菜单中选择“新建 @Prompt”。预定义过滤器编辑器将在查询/对象定义中插入过滤器表达式。

示例：在 Country 级别对 Customer 维加以限制，可以将国家\地区限制为 Canada

```
<FILTER KEY="[Customer].[Country].[Country]"> <CONDITION OPERATORCONDITION="等于">
<CONSTANT CAPTION="Canada" /> </CONDITION> </FILTER>
```

相关主题

- 第 473 页上的[“关于预定义过滤器编辑器的选项”](#)
- 第 474 页上的[“使用预定义过滤器编辑器编辑预定义过滤器”](#)
- 第 468 页上的[“关于多维数据集查询的 MDX 函数”](#)

8.3.11.4 关于预定义过滤器编辑器的选项

“预定义过滤器”编辑器可用来为 OLAP Universe 轻松定义 Universe 过滤器。可定义以下选项：

选项	说明
选择运算符	从可用列表中选择一个运算符。默认值 = 等于
过滤器依据	过滤器依据的对象为现有 Universe 对象或自由定义（例如：[Measures].[Internet Sales Amount]）。默认值 = Universe 对象。
选择值列表	如果过滤器是基于现有对象，则在当前 Universe 中选择对象列表。默认选择 = 对象列表中的根类。
比较值	定义用于比较对象/表达式的值。根据所选运算符，可输入一组或两组值。这些值可能是静态的，也可能是基于提示的。默认值 = 静态值。

选项	说明
插入提示	手动编辑提示，或使用 @Prompt 编辑器。单击 >> 打开 @Prompt 编辑器。
设置索引感知	启用索引感知功能。必须声明主键，才能正确使用此功能。在 Universe 设计工具中设置索引感知后，可使用主键和外键列加速数据检索，并帮助 Universe 设计工具生成更有效的 SQL 过滤器。默认值 = 未选定。
使用计算所得表达式	选定此选项后，会将过滤器表达式括入 <EXPRESSION> </EXPRESSION> 标记内。默认值 = 未选定。
可选	将当前过滤器表达式设置为可选。仅适用于过滤器编辑器中当前的过滤器表达式，而非整个预定义的条件对象。默认值 = 未选定。

注意：

“可选”标记无法用于 Web Intelligence 中的预定义过滤器。如果使用这些标记，它们将被视为查询的必需部分，进而阻止运行查询。

相关主题

- 第 474 页上的[“使用预定义过滤器编辑器编辑预定义过滤器”](#)

8.3.11.5 使用预定义过滤器编辑器编辑预定义过滤器

假设用户正在 OLAP Universe 中编辑过滤器。

在用户选择或输入值时，“预定义过滤器”编辑器会随之更新。可右键单击过滤器表达式，在过滤器表达式中插入 @Prompt 表达式。右键单击并选择“新建 @Prompt”，会打开“提示”编辑器。

- 1 在条件（过滤器）窗格的“属性”窗格中，单击“>>”。
将出现“预定义过滤器”编辑器。
- 2 要使此过滤器基于某个 Universe 对象，先选择“Universe 对象”，然后从“可用对象”窗格中选择一个对象。要使此预定义过滤器基于用户自己的表达式，先选择“自由定义”，然后在“可用对象”窗格中键入表达式。
- 3 从“运算符”列表中选择运算符。仅允许属于列表和不属于列表运算符具有多个值（右操作数）。
- 4 选择“静态值”定义一个或多个固定值，或选择“提示”插入提示表达式。
如果选择“提示”，将激活“编辑”按钮。单击“编辑”打开“@Prompt”编辑器，如果需要，还可定义提示表达式。
- 5 单击“确定”确认过滤器定义。
分析器会检查语法错误，其中包括完整性检查。如果发现错误，则显示带有错误消息的警告消息。如果未发现错误，会使用过滤器定义将新的条件对象添加到 Universe。

相关主题

- 第 473 页上的[“关于预定义过滤器编辑器的选项”](#)
- 第 473 页上的[“关于预定义过滤器编辑器”](#)

8.3.12 OLAP Universe 中的可选提示

从 OLAP 数据源生成的 Universe 支持可选提示。

对于 SAP NetWeaver BW 可选变量，含可选条件的过滤器是在 Universe 中自动生成的。

在预定义条件或在对象的 Where 子句中，若要使提示成为可选，请将 XML 过滤器表达式嵌入到两个 XML 标记 <OPTIONAL> 和 </OPTIONAL> 之间。

示例：预定义条件中的可选提示

```
<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>
```

相关主题

- 第 471 页上的[“在 OLAP Universe 中手动创建预定义条件”](#)

8.3.13 提高 SAP NetWeaver BW Universe 上某些查询的性能

如果是对只包含维的键明细对象和中间名明细对象的 SAP NetWeaver BW Universe 执行查询，可修改生成的对象语法来提高查询性能。

修改语法：

- 1 在 Universe 设计工具中打开该 Universe。
- 2 双击要修改的键明细对象。

- 3 在“编辑属性”对话框的“定义”选项卡上的“Select”文本框中，更改语法以引用 SAP 特性的 NAME 属性。

例如，对于对象“L01 Customer Key (L01 客户键)”，请更改生成的 Select 语法：

```
[Z_CUSTOM]. [LEVEL01]. [[2Z_CUSTOM]]. [Value]
```

以引用 NAME 属性：

```
[Z_CUSTOM]. [LEVEL01]. [NAME]
```

- 4 单击“确定”以保存更改。
 - 5 对名称对象执行相同步骤。更改语法以引用 SAP 特性的 DESCRIPTION 属性。
- 例如，对于对象“L01 Customer Medium Name (L01 客户中间名)”，请更改生成的 Select 语法：

```
[Z_CUSTOM]. [LEVEL01]. [[5Z_CUSTOM]]. [Value]
```

以引用 DESCRIPTION 属性：

```
[Z_CUSTOM]. [LEVEL01]. [DESCRIPTION]
```

8.4 OLAP Universe 生命周期管理

8.4.1 关于 OLAP Universe 生命周期管理

注意：

打开使用 XIR3.1 SP2 之前版本的 Universe Designer 创建的 Universe 时，必须先刷新并保存 Universe，然后才能在 Universe 或 OLAP 源中进行更改。

OLAP Universe 是从 OLAP 数据源自动生成的（例如，SAP NetWeaver BEx 查询或 MSAS 2005 多维数据集）。在 Universe 设计工具中，可以在现有的 OLAP Universe 中创建和更改对象。

使用“更新 OLAP Universe”向导可以自动刷新 OLAP Universe 的结构，以反映在 OLAP 数据源中所做的更改。向导会将 Universe 与更新后的数据源进行比较。向导可以区分生成的对象与手动添加或修改的对象，使用户可以保留在 Universe 设计工具中手动进行的更改。此向导不会更新在 Universe 设计工具中手动添加的对象。

可检测和更新的内容取决于项目和数据源，如下表所示。

向导可检测的内容	可检测新项目的位置	可检测已修改的项目的位置	可检测已删除的项目的位置
维	所有数据源	所有数据源	所有数据源
层次结构	仅限 SAP NetWeaver BW 和 MSAS	所有数据源	所有数据源
级别	所有数据源	所有数据源	所有数据源
属性	仅限 MSAS	仅限 MSAS	仅限 MSAS
度量	所有数据源	所有数据源	所有数据源
SAP NetWeaver BW 变量	仅限 SAP NetWeaver BW	仅限 SAP NetWeaver BW	仅限 SAP NetWeaver BW
子类	所有数据源	所有数据源	所有数据源

注意：

在更新由 XIR3.1 SP2 之前版本的 Universe Designer 创建的 Universe 时，如果维的名称已经在 SAP 多维数据集中进行更改，则不能对其进行刷新：该维在 Universe 中存在副本。必须手动更新 Universe 中的类。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 477 页上的[“概述：Universe 对象状态和 OLAP 对象状态之间的关系”](#)
- 第 484 页上的[“在 OLAP Universe 更新中管理维的方式”](#)
- 第 489 页上的[“在 OLAP Universe 更新中管理层次结构或特性的方式”](#)
- 第 496 页上的[“在 OLAP Universe 更新中管理级别的方式”](#)
- 第 500 页上的[“在 OLAP Universe 更新中管理 SAP 变量的方式”](#)
- 第 504 页上的[“在 OLAP Universe 更新中管理关键值或度量的方式”](#)
- 第 508 页上的[“在 OLAP Universe 更新中管理 SAP 关键日期的方式”](#)

8.4.2 概述：Universe 对象状态和 OLAP 对象状态之间的关系

下表对 SAP OLAP 对象状态和 Universe 对象状态之间的关系进行了简要概述。大多数操作都带有特定的注释，有关更多信息，请参阅本章中各节的详细内容。

OLAP 元数据		Universe 对象状态				
		未更改	已更新*	已删除	已移动	隐藏
维		Universe 等价物 = 类				
状态	未更改	NoC	Upd	NoC	NoC	NoC
	已更新*	Upd	Upd	NoC	Upd	Upd
	已删除	Del/Ob	Del/Ob	NoC	Del/Ob	NoC
	已移动	Move	NoC	NoC	NoC	Move
	已创建特性	CreS	CreS	N/A	CreS	CreS
	已创建	Cre	Cre	N/A	Cre	Cre
层次结构或特性		Universe 等价物 = 子类				
状态	未更改	NoC	Upd	NoC	NoC	NoC
	已更新*	Upd	Upd	NoC	Upd	Upd
	已更改	UpdMDX	UpdMDX	NoC	UpdMDX	UpdMDX
	显示属性	Cre	Cre	Cre	Cre	Cre
	导航属性	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	已删除	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	已移动	Move	Move	NoC	Move	Move
	新建	Cre	Cre	Cre	Cre	Cre
级别		Universe 等价物 = 级别				
状态	未更改	NoC	NoC	NoC	NoC	NoC
	已更新*	Upd	Upd	NoC	Upd	Upd
	已删除	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	已移动	Move	Move	NoC	Move	Move
	新建	Cre	Cre	Cre	Cre	Cre
变量		Universe 等价物 = 过滤器				

OLAP 元数据		Universe 对象状态				
		未更改	已更新*	已删除	已移动	隐藏
状态	未更改	NoC	NoC	NoC	NoC	NoC
	已更新*	Upd	Upd	Cre	Upd	Upd
	已删除	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	新建	Cre	Cre	Cre	Cre	Cre
关键值		Universe 等价物 = 度量				
状态	未更改	NoC	NoC	NoC	NoC	NoC
	已更新*	Upd	Upd	NoC	Upd	Upd
	已删除	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	已移动	Move	Move	NoC	Move	Move
	新建	Cre	Cre	Cre	Cre	Cre
关键日期		Universe 等价物 = 参数				
状态	未更改	NoC	N/A	Cre	不适用	不适用
	已删除	Del	不适用	不适用	不适用	不适用
	新建	Cre	N/A	Cre	不适用	不适用

说明：

- *：对象属性（名称、说明等）之一已被更改。
- Cre：创建等价对象
- CreS：创建等价子类对象
- Del/Ob：已删除或过期（隐藏名称前缀为 ## 的过期对象）
- Move：对象已移动
- N/A：不适用
- NoC：未更改
- Upd：已更新
- UpdMDX：更新 MDX 定义

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 484 页上的[“在 OLAP Universe 更新中管理维的方式”](#)
- 第 489 页上的[“在 OLAP Universe 更新中管理层次结构或特性的方式”](#)
- 第 496 页上的[“在 OLAP Universe 更新中管理级别的方式”](#)
- 第 500 页上的[“在 OLAP Universe 更新中管理 SAP 变量的方式”](#)

- 第 504 页上的“[在 OLAP Universe 更新中管理关键值或度量的方式](#)”
- 第 508 页上的“[在 OLAP Universe 更新中管理 SAP 关键日期的方式](#)”

8.4.3 刷新 OLAP Universe

刷新 OLAP Universe 的结构：

- 在 Universe 设计工具中，打开要更新的 Universe。
- 选择“查看” > “刷新结构”。

将出现“更新 OLAP Universe”向导。

- 单击“开始”。

显示“更新的元数据定义”页面。

如果要保留对 Universe 中对象进行的手动修改，请选择“保留”选项。默认情况下选定所有“保留”选项。可以选择保留以下属性：

选项	说明
保留业务名称	类、维、度量、明细和条件名称。
保留对象类型	如果对象已在 Universe 中进行更改（例如，将明细更改为维，则更新将不会再次使用初始对象类型）。涉及维、度量及明细。
保留对象说明	选择该选项时，如果说明在 OLAP 源中进行更新，则 Universe 将不再更新该信息。
保留对象的数据类型	字符、数字、日期及长文本类型的对象。
保留对象的值列表选项	可保留最初设置的选项： <ul style="list-style-type: none"> • 关联值列表 • 使用之前自动刷新 • 分层显示 • 随 Universe 导出 • 委派搜索

选项	说明
保留对象的高级选项	选项如下： 安全访问级别 对象用途 <ul style="list-style-type: none">• 用于结果• 用于条件• 用于排序
删除过期对象	数据源中不再存在的项目将从 Universe 中删除。
隐藏过期的对象	将在 Universe 中隐藏多维数据集中不再存在的项目，并对其附以前缀 /###/。

- 选择所需选项，然后单击“下一步”。

将出现“更改管理结果”页，其中显示已添加/删除/隐藏的对象。隐藏的对象被移到 Universe 的一个单独类中，以 /###/ 开头，并用斜体字体显示。

- 在添加的元数据选项窗格中，设置有关如何处理添加的元数据的选项

通用 OLAP 选项	说明
将技术名称生成为明细	可设置应用程序，将 Universe 的技术名称生成为属性。在生成 Universe 时，此设置将创建指向技术名称的对象。
重新生成所有手动删除的对象	将重新生成任何手动删除的 Universe 对象。

SAP OLAP 选项	说明
将度量聚合委派给数据库	可设置应用程序，将度量的聚合函数委派给数据库。
替换前缀 L00、L01	Universe 级别前缀表示某一对象的层次结构级别。L00 级别是顶级或根级，而 L01 是下一级别。在“新 Universe 向导”中，可使用不同的前缀替换 OLAP Universe 级别前缀。例如，保留级别编号，不过可以用级别替换前缀“L”。在“新前缀”字段中键入自己的前缀。此前缀位于 OLAP Universe 中所有级别之前。
将级别 00 重命名为“所有”	如果将“生成级别 00”设置为“否”，将禁用此选项。下一次生成 Universe 时，可将顶级（根级）L00 重命名为“所有”。
生成级别 00	此选项仅适用于 SAP 特性。可仅针对特性取消激活此选项。将始终为层次结构及层次结构变量生成级别 00。生成或更新 Universe 时，可重新生成级别编号（L00、L01、L02...）。级别编号将附加到级别名称（例如“Monthly Sales_L01”）后。此操作对 Web Intelligence 报表颇有帮助，其中“所有”级别用于聚合查询结果。这样一来，便不必在 Web Intelligence 报表中创建聚合字段。

- 在“更改管理结果”页上，选择以下项之一：

选项	说明
确定	如果对结果不满意，请单击“确定”，然后关闭 Universe，且不执行保存或导出。
导出	如果对更改满意，请单击“导出”以进行保存，并将更新后的 Universe 导出至 CMS。
检查完整性	单击“检查完整性”以执行完整性检查。它会检查结构、分析对象、分析联接、分析条件并检查基数。检查结束后会出现“完整性检查结果”页。从此页中可以打印检查结果。

如果没有看见预计对 Universe 进行的所有更改，则停止操作，并重新启动 Universe 设计工具，然后再次尝试更新。此操作会导致与数据源建立新连接，并清除缓存。

相关主题

- 第 484 页上的[“正在同步 Universe 和 OLAP 多维数据集”](#)
- 第 476 页上的[“关于 OLAP Universe 生命周期管理”](#)

8.4.4 为 OLAP Universe 重新生成级别 00

生成或更新 Universe 时，可重新生成级别编号（L00、L01、L02...）。级别编号附加到级别名称（例如“Monthly Sales_L01”）后

相关主题

- 第 465 页上的[“为 OLAP Universe 设置委派度量”](#)
- 第 483 页上的[“替换 OLAP Universe 级别前缀”](#)
- 第 483 页上的[“将级别 L00 重命名为“所有””](#)

8.4.5 将级别 L00 重命名为“所有”

下一次生成 Universe 时，可将顶级（根级）L00 重命名为“所有”。此操作对 SAP BusinessObjects Web Intelligence 报表颇有帮助，其中“所有”级别用于聚合查询结果。这样一来，便不必在 Web Intelligence 报表中创建聚合字段。

相关主题

- 第 465 页上的[“为 OLAP Universe 设置委派度量”](#)
- 第 483 页上的[“替换 OLAP Universe 级别前缀”](#)
- 第 483 页上的[“为 OLAP Universe 重新生成级别 00”](#)

8.4.6 替换 OLAP Universe 级别前缀

Universe 级别前缀表示某一对象的层次结构级别。L00 级别是顶级或根级，而 L01 是下一级别。在“新 Universe 向导”中，可使用不同的前缀替换 OLAP Universe 级别。例如，保留级别编号，不过可以用级别替换前缀“L”。在“新前缀”字段中键入自己的前缀。此前缀位于 OLAP Universe 中所有级别之前。

相关主题

- 第 465 页上的[“为 OLAP Universe 设置委派度量”](#)
- 第 483 页上的[“为 OLAP Universe 重新生成级别 00”](#)
- 第 483 页上的[“将级别 L00 重命名为“所有””](#)

8.4.7 正在同步 Universe 和 OLAP 多维数据集

更新 Universe 时，Universe 中的对象会与 OLAP 多维数据集中的对象进行比较。从而确保在多维数据集中所做的更改不会反过来影响 Universe。也就是说在 Universe 中所使用（甚至删除的）的任何对象都必须始终可用。Universe 可以使用 OLAP 多维数据集中的任何新对象。要查看更改对不同对象的影响方式，请点击以下链接。

更新对象属性时，只更新 Universe 中的部分属性，不应更改其他属性。具体情况参见下表。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 484 页上的[“在 OLAP Universe 更新中管理维的方式”](#)
- 第 489 页上的[“在 OLAP Universe 更新中管理层次结构或特性的方式”](#)
- 第 496 页上的[“在 OLAP Universe 更新中管理级别的方式”](#)
- 第 500 页上的[“在 OLAP Universe 更新中管理 SAP 变量的方式”](#)
- 第 504 页上的[“在 OLAP Universe 更新中管理关键值或度量的方式”](#)
- 第 508 页上的[“在 OLAP Universe 更新中管理 SAP 关键日期的方式”](#)

8.4.8 在 OLAP Universe 更新中管理维的方式

适用于 SAP、MSAS 和 Essbase 数据源。Universe 类相当于 OLAP 维。OLAP 对象相关的 Universe 对象的管理方式取决于更改类型。请参阅以下主题，了解特定的 OLAP 对象更改如何影响 Universe 对象。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 485 页上的[“未更改维所造成的影响”](#)
- 第 485 页上的[“更新维（名称、说明等）所造成的影响”](#)

- 第 486 页上的[“删除维所造成的影响”](#)
- 第 487 页上的[“移动维所造成的影响”](#)
- 第 488 页上的[“创建层次结构或特性所造成的影响”](#)
- 第 489 页上的[“新建维所造成的影响”](#)

8.4.8.1 未更改维所造成的影响

下表显示了当未更改维时，在不同情况下对等价 Universe 类所造成的影响：

对 Universe 类执行的操作	Universe 类受到的影响
未更改	Universe 类未更改
更新	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
删除	未更改 Universe 类。 如果选定选项“重新生成手动删除的对象”，则创建对象。不重新生成尚未删除的子级别。
移动	未更改 Universe 类。
隐藏	未更改 Universe 类。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 485 页上的[“更新维（名称、说明等）所造成的影响”](#)
- 第 486 页上的[“删除维所造成的影响”](#)
- 第 487 页上的[“移动维所造成的影响”](#)
- 第 488 页上的[“创建层次结构或特性所造成的影响”](#)
- 第 489 页上的[“新建维所造成的影响”](#)

8.4.8.2 更新维（名称、说明等）所造成的影响

下表显示了更新维的名称或说明时，在不同情况下对等价 Universe 类所造成的影响：

对 Universe 类执行的操作	Universe 类受到的影响
未更改	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
更新	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
删除	未更改 Universe 类。 如果选定选项“重新生成手动删除的对象”，则创建。 不重新生成尚未删除的子级别。
移动	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
隐藏	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 485 页上的[“未更改维所造成的影响”](#)
- 第 486 页上的[“删除维所造成的影响”](#)
- 第 487 页上的[“移动维所造成的影响”](#)
- 第 488 页上的[“创建层次结构或特性所造成的影响”](#)
- 第 489 页上的[“新建维所造成的影响”](#)

8.4.8.3 删除维所造成的影响

下表显示当删除维时，在不同情况下对等价 Universe 类所造成的影响：

对 Universe 类执行的操作	Universe 类受到的影响
未更改	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。如果类包含定制对象，则不删除
更新	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。如果类包含定制对象，则不删除
删除	未更改 Universe 类。
移动	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。如果类包含定制对象，则不删除
隐藏	未更改 Universe 类

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 485 页上的[“未更改维所造成的影响”](#)
- 第 485 页上的[“更新维（名称、说明等）所造成的影响 ”](#)
- 第 487 页上的[“移动维所造成的影响 ”](#)
- 第 488 页上的[“创建层次结构或特性所造成的影响 ”](#)
- 第 489 页上的[“新建维所造成的影响 ”](#)

8.4.8.4 移动维所造成的影响

下表显示了移动维时，在不同情况下对等价 Universe 类所造成的影响：

对 Universe 类执行的操作	Universe 类受到的影响
未更改	相应地移动类
更新	无更改
删除	无更改。 如果选项：“重新生成手动删除的对象” = “是”，则创建不重新生成尚未删除的子级别。

对 Universe 类执行的操作	Universe 类受到的影响
移动	无更改
隐藏	相应地移动类

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 485 页上的[“未更改维所造成的影响”](#)
- 第 485 页上的[“更新维（名称、说明等）所造成的影响 ”](#)
- 第 486 页上的[“删除维所造成的影响 ”](#)
- 第 488 页上的[“创建层次结构或特性所造成的影响 ”](#)
- 第 489 页上的[“新建维所造成的影响 ”](#)

8.4.8.5 创建层次结构或特性所造成的影响

层次结构适用于 MSAS 或 Essbase 数据源，特性适用于 SAP 数据源。下表显示了创建 SAP 特性时，在不同情况下对等价 Universe 类所造成的影响：

对 Universe 类执行的操作	Universe 类受到的影响
未更改	创建子类
更新	创建子类
删除	不适用
移动	创建子类
隐藏	创建子类

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 485 页上的[“未更改维所造成的影响”](#)
- 第 485 页上的[“更新维（名称、说明等）所造成的影响 ”](#)
- 第 486 页上的[“删除维所造成的影响 ”](#)
- 第 487 页上的[“移动维所造成的影响 ”](#)

- 第 489 页上的[“新建维所造成的影响”](#)

8.4.8.6 新建维所造成的影响

创建维时可创建 Universe 类。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 485 页上的[“未更改维所造成的影响”](#)
- 第 485 页上的[“更新维（名称、说明等）所造成的影响”](#)
- 第 486 页上的[“删除维所造成的影响”](#)
- 第 487 页上的[“移动维所造成的影响”](#)
- 第 488 页上的[“创建层次结构或特性所造成的影响”](#)

8.4.9 在 OLAP Universe 更新中管理层次结构或特性的方式

本节适用于 MSAS 和 Essbase 数据源的层次结构，以及 SAP 数据源的特性。Universe 子类相当于 OLAP 特性。OLAP 对象相关的 Universe 对象的管理方式取决于更改类型。请参阅以下主题，了解特定的 OLAP 对象更改如何影响 Universe 对象。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“未更改层次结构或特性所造成的影响”](#)
- 第 490 页上的[“更新特性业务名称或说明所造成的影响”](#)
- 第 491 页上的[“更改特性的活动层次结构所造成的影响”](#)
- 第 492 页上的[“特性显示属性更改为导航属性所造成的影响”](#)
- 第 494 页上的[“删除层次结构或特性后所造成的影响”](#)
- 第 495 页上的[“移动层次结构或特性后所造成的影响”](#)
- 第 496 页上的[“新建层次结构或特性所造成的影响”](#)

8.4.9.1 未更改层次结构或特性所造成的影响

下表显示了未更改层次结构或特性时，在不同情况下对等价 Universe 子类所造成的影响：

对 Universe 子类执行的操作	Universe 子类受到的影响
未更改	无更改
更新	无更改
删除	无更改。 如果选项：“重新生成手动删除的对象” = “是”，则执行创建。 不重新生成尚未删除的子级别。
移动	无更改
隐藏	无更改

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“更新特性业务名称或说明所造成的影响”](#)
- 第 491 页上的[“更改特性的活动层次结构所造成的影响”](#)
- 第 492 页上的[“特性显示属性更改为导航属性所造成的影响”](#)
- 第 494 页上的[“删除层次结构或特性后所造成的影响”](#)
- 第 495 页上的[“移动层次结构或特性后所造成的影响”](#)
- 第 496 页上的[“新建层次结构或特性所造成的影响”](#)

8.4.9.2 更新特性业务名称或说明所造成的影响

下表显示了更新特性时，在不同情况下对等价 Universe 子类所造成的影响：

对 Universe 子类执行的操作	Universe 子类受到的影响
未更改	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
更新	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
删除	无更改。 如果选项：“重新生成手动删除的对象” = “是”，则执行创建。 不重新生成未删除的子级别。
移动	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果未选定这些选项，则保持未更改状态。
隐藏	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“未更改层次结构或特性所造成的影响”](#)
- 第 491 页上的[“更改特性的活动层次结构所造成的影响”](#)
- 第 492 页上的[“特性显示属性更改为导航属性所造成的影响”](#)
- 第 494 页上的[“删除层次结构或特性后所造成的影响”](#)
- 第 495 页上的[“移动层次结构或特性后所造成的影响”](#)
- 第 496 页上的[“新建层次结构或特性所造成的影响”](#)

8.4.9.3 更改特性的活动层次结构所造成的影响

仅适用于 SAP 数据源。下表显示了当更改特性的活动层次结构时，在不同情况下对等价 Universe 子类所造成的影响：

对 Universe 子类执行的操作	Universe 子类受到的影响
未更改	在子类中更新现有对象的 MDX 定义，以引用新的活动层次结构。 刷新前构建的报表依然正常工作
更新	在子类中更新现有对象的 MDX 定义，以引用新的活动层次结构。 刷新前构建的报表依然正常工作
删除	无更改。 如果选项：“重新生成手动删除的对象” = “是”，则创建 不重新生成尚未删除的子级别。
移动	在子类中更新现有对象的 MDX 定义，以引用新的活动层次结构。 刷新前构建的报表依然正常工作。
隐藏	在子类中更新现有对象的 MDX 定义，以引用新的活动层次结构。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“未更改层次结构或特性所造成的影响”](#)
- 第 490 页上的[“更新特性业务名称或说明所造成的影响”](#)
- 第 492 页上的[“特性显示属性更改为导航属性所造成的影响”](#)
- 第 494 页上的[“删除层次结构或特性后所造成的影响”](#)
- 第 495 页上的[“移动层次结构或特性后所造成的影响”](#)
- 第 496 页上的[“新建层次结构或特性所造成的影响”](#)

8.4.9.4 特性显示属性更改为导航属性所造成的影响

仅适用于 SAP 数据源。下表显示了将特性显示属性更改为导航属性时，在不同情况下对等价 Universe 子类所造成的影响：

对 Universe 子类执行的操作	Universe 子类受到的影响
未更改	创建
更新	创建
删除	创建
移动	创建
隐藏	创建

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“未更改层次结构或特性所造成的影响”](#)
- 第 490 页上的[“更新特性业务名称或说明所造成的影响”](#)
- 第 491 页上的[“更改特性的活动层次结构所造成的影响”](#)
- 第 494 页上的[“删除层次结构或特性后所造成的影响”](#)
- 第 495 页上的[“移动层次结构或特性后所造成的影响”](#)
- 第 496 页上的[“新建层次结构或特性所造成的影响”](#)

8.4.9.5 特性导航属性更改为显示属性所造成的影响

仅适用于 SAP 数据源。下表显示了将层次结构或特性导航属性更改为显示属性时，在不同情况下对等价 Universe 子类所造成的影响：

对 Universe 子类执行的操作	Universe 子类受到的影响
未更改	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则隐藏子类。如果类包含定制对象，则不删除。
更新	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则隐藏子类。如果类包含定制对象，则不删除。
删除	无更改
移动	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则隐藏子类。如果类包含定制对象，则不删除。

对 Universe 子类执行的操作	Universe 子类受到的影响
隐藏	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则隐藏子类。如果类包含定制对象，则不删除。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“未更改层次结构或特性所造成的影响”](#)
- 第 490 页上的[“更新特性业务名称或说明所造成的影响”](#)
- 第 491 页上的[“更改特性的活动层次结构所造成的影响”](#)
- 第 494 页上的[“删除层次结构或特性后所造成的影响”](#)
- 第 495 页上的[“移动层次结构或特性后所造成的影响”](#)
- 第 496 页上的[“新建层次结构或特性所造成的影响”](#)

8.4.9.6 删除层次结构或特性后所造成的影响

下表显示了在删除层次结构或特性后，在不同情况下对等价 Universe 子类所造成的影响：

对 Universe 子类执行的操作	Universe 子类受到的影响
未更改	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。如果子类包含定制对象，则不删除。
更新	如果选定选项“删除过期对象”，则删除；如果选定选项“隐藏过期的对象”，则废弃子类。如果子类包含定制对象，则不删除。
删除	无更改
移动	如果选定选项“删除过期对象”，则删除；如果选定选项“隐藏过期的对象”，则废弃子类。如果子类包含定制对象，则不删除。
隐藏	如果选定选项“删除过期对象”，则删除；如果选定选项“隐藏过期的对象”，则废弃子类。如果子类包含定制对象，则不删除。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“未更改层次结构或特性所造成的影响”](#)

- 第 490 页上的[“更新特性业务名称或说明所造成的影响”](#)
- 第 491 页上的[“更改特性的活动层次结构所造成的影响”](#)
- 第 492 页上的[“特性显示属性更改为导航属性所造成的影响”](#)
- 第 495 页上的[“移动层次结构或特性后所造成的影响”](#)
- 第 496 页上的[“新建层次结构或特性所造成的影响”](#)

8.4.9.7 移动层次结构或特性后所造成的影响

如果在相同维中移动特性，则不进行更改：忽略下表。下表显示了将层次结构或特性移动到其他维时，在不同情况下对等价 Universe 子类所造成的影响：

对 Universe 子类执行的操作	Universe 子类受到的影响
未更改	相应地移动子类。
更新	相应地移动子类。
删除	无更改。 如果选项：“重新生成所有手动删除的对象” = “是”，则创建不重新生成尚未删除的子级别。
移动	无更改。
隐藏	相应地移动子类。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“未更改层次结构或特性所造成的影响”](#)
- 第 490 页上的[“更新特性业务名称或说明所造成的影响”](#)
- 第 491 页上的[“更改特性的活动层次结构所造成的影响”](#)
- 第 492 页上的[“特性显示属性更改为导航属性所造成的影响”](#)
- 第 494 页上的[“删除层次结构或特性后所造成的影响”](#)
- 第 496 页上的[“新建层次结构或特性所造成的影响”](#)

8.4.9.8 新建层次结构或特性所造成的影响

创建层次结构或特性时可创建 Universe 子类。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 490 页上的[“未更改层次结构或特性所造成的影响”](#)
- 第 490 页上的[“更新特性业务名称或说明所造成的影响 ”](#)
- 第 491 页上的[“更改特性的活动层次结构所造成的影响 ”](#)
- 第 492 页上的[“特性显示属性更改为导航属性所造成的影响”](#)
- 第 494 页上的[“删除层次结构或特性后所造成的影响 ”](#)
- 第 495 页上的[“移动层次结构或特性后所造成的影响 ”](#)

8.4.10 在 OLAP Universe 更新中管理级别的方式

注意：

在 Universe 中，请勿将级别移动至其他层次结构。如果需要移动级别，请将该级别复制并粘贴到新的层次结构中。

Universe 级别或维对象相当于 OLAP 级别。OLAP 对象相关的 Universe 对象的管理方式取决于更改类型。请参阅以下主题，了解特定的 OLAP 对象更改如何影响 Universe 对象。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 497 页上的[“未更改级别所造成的影响 ”](#)
- 第 497 页上的[“更新级别名称或说明所造成的影响 ”](#)
- 第 498 页上的[“删除级别所造成的影响 ”](#)
- 第 499 页上的[“移动级别所造成的影响 ”](#)
- 第 500 页上的[“新建级别所造成的影响 ”](#)

8.4.10.1 未更改级别所造成的影响

下表显示了未更改级别时，在不同情况下对 Universe 级别所造成的影响：

对 Universe 级别执行的操作	Universe 级别受到的影响
未更改	无更改
更新	无更改
删除	无更改。如果选项：“重新生成手动删除的对象” = “是”，则创建
移动	无更改
隐藏	无更改

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 497 页上的[“更新级别名称或说明所造成的影响 ”](#)
- 第 498 页上的[“删除级别所造成的影响 ”](#)
- 第 499 页上的[“移动级别所造成的影响 ”](#)
- 第 500 页上的[“新建级别所造成的影响 ”](#)

8.4.10.2 更新级别名称或说明所造成的影响

下表显示了更新级别的名称或说明时，在不同情况下对 Universe 级别所造成的影响：

对 Universe 级别执行的操作	Universe 级别受到的影响
未更改	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
更新	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
删除	无更改。如果选项：“重新生成所有手动删除的对象” = “是”，则创建
移动	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
隐藏	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 497 页上的[“未更改级别所造成的影响”](#)
- 第 498 页上的[“删除级别所造成的影响”](#)
- 第 499 页上的[“移动级别所造成的影响”](#)
- 第 500 页上的[“新建级别所造成的影响”](#)

8.4.10.3 删除级别所造成的影响

下表显示了删除级别时，在不同情况下对 Universe 级别所造成的影响：

对 Universe 级别执行的操作	Universe 级别受到的影响
未更改	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。
更新	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。
删除	无更改。
移动	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。
隐藏	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 497 页上的[“未更改级别所造成的影响”](#)
- 第 497 页上的[“更新级别名称或说明所造成的影响”](#)
- 第 499 页上的[“移动级别所造成的影响”](#)
- 第 500 页上的[“新建级别所造成的影响”](#)

8.4.10.4 移动级别所造成的影响

下表显示了移动级别时，在不同情况下对 Universe 级别所造成的影响：

对 Universe 级别执行的操作	Universe 级别受到的影响
未更改	相应地移动级别（在相同层次结构内）。
更新	相应地移动级别（在相同层次结构内）。
删除	无更改。如果选项“重新生成所有手动删除的对象” = “是”，则创建。
移动	无更改。如果选项“重新生成所有手动删除的对象” = “是”，则创建。
隐藏	相应地移动级别（在相同层次结构内）。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 497 页上的[“未更改级别所造成的影响 ”](#)
- 第 497 页上的[“更新级别名称或说明所造成的影响 ”](#)
- 第 498 页上的[“删除级别所造成的影响 ”](#)
- 第 500 页上的[“新建级别所造成的影响 ”](#)

8.4.10.5 新建级别所造成的影响

创建 OLAP 级别时将创建 Universe 级别。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 497 页上的[“未更改级别所造成的影响 ”](#)
- 第 497 页上的[“更新级别名称或说明所造成的影响 ”](#)
- 第 498 页上的[“删除级别所造成的影响 ”](#)
- 第 499 页上的[“移动级别所造成的影响 ”](#)

8.4.11 在 OLAP Universe 更新中管理 SAP 变量的方式

本节仅涉及 SAP 数据源。Universe 过滤器和值对象的相关列表相当于 OLAP 变量。OLAP 对象相关的 Universe 对象的管理方式取决于更改类型。请参阅以下主题，了解特定的 OLAP 对象更改如何影响 Universe 对象。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 501 页上的[“未更改 SAP 变量所造成的影响”](#)
- 第 501 页上的[“更新 SAP 变量的名称或说明所造成的影响 ”](#)
- 第 502 页上的[“删除 SAP 变量所造成的影响 ”](#)
- 第 503 页上的[“新建 SAP 变量所造成的影响 ”](#)

8.4.11.1 未更改 SAP 变量所造成的影响

下表显示了未更改 SAP 源变量时，在不同情况下对 Universe 过滤器的管理方式：

对 Universe 过滤器执行的操作	Universe 过滤器受到的影响
未更改	无更改
更新	无更改
删除	创建。如果变量中引用的特性不在 Universe 中，则也为该特性创建一个子类。
移动	无更改
隐藏	无更改

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 501 页上的[“更新 SAP 变量的名称或说明所造成的影响 ”](#)
- 第 502 页上的[“删除 SAP 变量所造成的影响 ”](#)
- 第 503 页上的[“新建 SAP 变量所造成的影响 ”](#)

8.4.11.2 更新 SAP 变量的名称或说明所造成的影响

下表显示了更新 SAP 源变量的名称或说明时，在不同情况下对 Universe 过滤器的管理方式：

对 Universe 过滤器执行的操作	Universe 过滤器受到的影响
未更改	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
更新	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
删除	创建。如果变量中引用的特性不在 Universe 中，则也为该特性创建一个子类。
移动	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。
隐藏	如果未选定选项“保留业务名称”，则更新业务名称。 如果未选定选项“保留对象的说明”，则更新说明。 如果选定这些选项，则保持未更改状态。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 501 页上的[“未更改 SAP 变量所造成的影响”](#)
- 第 502 页上的[“删除 SAP 变量所造成的影响 ”](#)
- 第 503 页上的[“新建 SAP 变量所造成的影响 ”](#)

8.4.11.3 删除 SAP 变量所造成的影响

下表显示了删除 SAP 变量时，在不同情况下对等价 Universe 过滤器的管理方式：

对 Universe 过滤器执行的操作	Universe 过滤器受到的影响
未更改	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则隐藏子类。当过滤器被废弃时，也从“强制”更改为“可选”，以避免在查询中自动应用
更新	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则隐藏子类。当过滤器被废弃时，也从“强制”更改为“可选”，以避免在查询中自动应用
删除	无更改。
移动	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则隐藏子类。当过滤器被废弃时，也从“强制”更改为“可选”，以避免在查询中自动应用
隐藏	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则隐藏子类。当过滤器被废弃时，也从“强制”更改为“可选”，以避免在查询中自动应用

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 501 页上的[“未更改 SAP 变量所造成的影响”](#)
- 第 501 页上的[“更新 SAP 变量的名称或说明所造成的影响”](#)
- 第 503 页上的[“新建 SAP 变量所造成的影响”](#)

8.4.11.4 新建 SAP 变量所造成的影响

下表显示了新建 SAP 变量时，在不同情况下对 Universe 过滤器的管理方式：

对 Universe 过滤器执行的操作	Universe 过滤器受到的影响
未更改	创建。如果变量中引用的特性不在 Universe 中，则也为该特性创建一个子类。
更新	创建。如果变量中引用的特性不在 Universe 中，则也为该特性创建一个子类。

对 Universe 过滤器执行的操作	Universe 过滤器受到的影响
删除	创建。如果变量中引用的特性不在 Universe 中，则也为该特性创建一个子类。
移动	创建。如果变量中引用的特性不在 Universe 中，则也为该特性创建一个子类。
隐藏	创建。如果变量中引用的特性不在 Universe 中，则也为该特性创建一个子类。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 501 页上的[“未更改 SAP 变量所造成的影响”](#)
- 第 501 页上的[“更新 SAP 变量的名称或说明所造成的影响”](#)
- 第 502 页上的[“删除 SAP 变量所造成的影响”](#)

8.4.12 在 OLAP Universe 更新中管理关键值或度量的方式

SAP 数据源采用关键值，MSAS 和 Essbase 数据源采用度量。Universe 度量相当于 OLAP 关键值。OLAP 对象相关的 Universe 对象的管理方式取决于更改类型。请参阅以下主题，了解特定的 OLAP 对象更改如何影响 Universe 对象。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 504 页上的[“未更改关键值或度量所造成的影响”](#)
- 第 505 页上的[“更新关键值或度量的名称、说明或数据类型所造成的影响”](#)
- 第 506 页上的[“删除关键值或度量所造成的影响”](#)
- 第 507 页上的[“移动关键值或度量所造成的影响”](#)
- 第 508 页上的[“新建关键值或度量所造成的影响”](#)

8.4.12.1 未更改关键值或度量所造成的影响

下表显示了未更改 SAP 关键值或 MSAS/Essbase 度量时，在不同情况下对等价 Universe 度量所造成的影响：

对 Universe 度量执行的操作	Universe 度量受到的影响
未更改	无更改
更新	无更改
删除	无更改。如果选项：“重新生成手动删除的对象” = “是”，则创建
移动	无更改
隐藏	无更改

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 505 页上的[“更新关键值或度量的名称、说明或数据类型所造成的影响”](#)
- 第 506 页上的[“删除关键值或度量所造成的影响”](#)
- 第 507 页上的[“移动关键值或度量所造成的影响”](#)
- 第 508 页上的[“新建关键值或度量所造成的影响”](#)

8.4.12.2 更新关键值或度量的名称、说明或数据类型所造成的影响

下表显示了更新 SAP 关键值或 MSAS/Essbase 度量时，在不同情况下对 Universe 度量所造成的影响：

对 Universe 度量执行的操作	Universe 度量受到的影响
未更改	<p>如果未选定选项“保留业务名称”，则更新业务名称。</p> <p>如果未选定选项“保留对象的说明”，则更新说明。</p> <p>如果未选定选项“保留对象的数据类型”，则更新数据类型。</p> <p>如果选定这些选项，则保持未更改状态。</p>

对 Universe 度量执行的操作	Universe 度量受到的影响
更新	<p>如果未选定选项“保留业务名称”，则更新业务名称。</p> <p>如果未选定选项“保留对象的说明”，则更新说明。</p> <p>如果未选定选项“保留对象的数据类型”，则更新数据类型。</p> <p>如果选定这些选项，则保持未更改状态。</p>
删除	<p>无更改。如果选项：“重新生成手动删除的对象” = “是”，则创建</p>
移动	<p>如果未选定选项“保留业务名称”，则更新业务名称。</p> <p>如果未选定选项“保留对象的说明”，则更新说明。</p> <p>如果未选定选项“保留对象的数据类型”，则更新数据类型。</p> <p>如果选定这些选项，则保持未更改状态。</p>
隐藏	<p>如果未选定选项“保留业务名称”，则更新业务名称。</p> <p>如果未选定选项“保留对象的说明”，则更新说明。</p> <p>如果未选定选项“保留对象的数据类型”，则更新数据类型。</p> <p>如果选定这些选项，则保持未更改状态。</p>

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 504 页上的[“未更改关键值或度量所造成的影响 ”](#)
- 第 506 页上的[“删除关键值或度量所造成的影响 ”](#)
- 第 507 页上的[“移动关键值或度量所造成的影响 ”](#)
- 第 508 页上的[“新建关键值或度量所造成的影响 ”](#)

8.4.12.3 删除关键值或度量所造成的影响

下表显示了删除 SAP 关键值或 MSAS/Essbase 度量时，在不同情况下对等价 Universe 度量所造成的影响：

对 Universe 度量执行的操作	Universe 度量受到的影响
未更改	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。
更新	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。
删除	无更改。
移动	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。
隐藏	如果选定选项“删除过期对象”，则删除。如果选定选项“隐藏过期的对象”，则废弃子类。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 504 页上的[“未更改关键值或度量所造成的影响 ”](#)
- 第 505 页上的[“更新关键值或度量的名称、说明或数据类型所造成的影响 ”](#)
- 第 507 页上的[“移动关键值或度量所造成的影响 ”](#)
- 第 508 页上的[“新建关键值或度量所造成的影响 ”](#)

8.4.12.4 移动关键值或度量所造成的影响

下表显示了移动 SAP 关键值或 MSAS/Essbase 度量时，在不同情况下对 Universe 度量所造成的影响：

对 Universe 度量执行的操作	Universe 度量受到的影响
未更改	相应地移动对象。
更新	相应地移动对象。
删除	无更改。如果选项：“重新生成手动删除的对象” = “是”，则创建。
移动	无更改。
隐藏	相应地移动对象。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 504 页上的[“未更改关键值或度量所造成的影响 ”](#)
- 第 505 页上的[“更新关键值或度量的名称、说明或数据类型所造成的影响 ”](#)
- 第 506 页上的[“删除关键值或度量所造成的影响 ”](#)
- 第 508 页上的[“新建关键值或度量所造成的影响 ”](#)

8.4.12.5 新建关键值或度量所造成的影响

创建 OLAP 关键值或度量时将创建 Universe 度量。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 504 页上的[“未更改关键值或度量所造成的影响 ”](#)
- 第 505 页上的[“更新关键值或度量的名称、说明或数据类型所造成的影响 ”](#)
- 第 506 页上的[“删除关键值或度量所造成的影响 ”](#)
- 第 507 页上的[“移动关键值或度量所造成的影响 ”](#)

8.4.13 在 OLAP Universe 更新中管理 SAP 关键日期的方式

本节仅适用于 SAP 数据源。Universe 参数相当于 OLAP 关键日期。OLAP 对象相关的 Universe 对象的管理方式取决于更改类型。请参阅以下主题，了解特定的 OLAP 对象更改如何影响 Universe 对象。

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 509 页上的[“未更改 SAP 关键日期所造成的影响 ”](#)
- 第 509 页上的[“删除 SAP 关键日期所造成的影响 ”](#)
- 第 510 页上的[“新建 SAP 关键日期所造成的影响 ”](#)

8.4.13.1 未更改 SAP 关键日期所造成的影响

Universe 参数相当于 OLAP 关键日期。下表显示了未更改 SAP 关键日期时，在不同情况下对 Universe 参数所造成的影响：

对 Universe 参数执行的操作	Universe 参数受到的影响
未更改	无更改
更新	不适用
删除	不适用
移动	不适用
隐藏	不适用

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 509 页上的[“删除 SAP 关键日期所造成的影响”](#)
- 第 510 页上的[“新建 SAP 关键日期所造成的影响”](#)

8.4.13.2 删除 SAP 关键日期所造成的影响

Universe 参数相当于 OLAP 关键日期。下表显示了删除 SAP 关键日期时，在不同情况下对等价 Universe 参数所造成的影响：

对 Universe 参数执行的操作	Universe 参数受到的影响
未更改	删除
更新	不适用
删除	不适用

对 Universe 参数执行的操作	Universe 参数受到的影响
移动	不适用
隐藏	不适用

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 509 页上的[“未更改 SAP 关键日期所造成的影响 ”](#)
- 第 510 页上的[“新建 SAP 关键日期所造成的影响”](#)

8.4.13.3 新建 SAP 关键日期所造成的影响

Universe 参数相当于 OLAP 关键日期。下表显示了新建 SAP 关键日期时，在不同情况下对等价 Universe 参数所造成的影响：

对 Universe 参数执行的操作	Universe 参数受到的影响
未更改	创建
更新	不适用
删除	创建
移动	不适用
隐藏	不适用

相关主题

- 第 480 页上的[“刷新 OLAP Universe”](#)
- 第 509 页上的[“未更改 SAP 关键日期所造成的影响 ”](#)
- 第 509 页上的[“删除 SAP 关键日期所造成的影响”](#)

8.5 不同 OLAP 多维数据集映射到 Universe 的方式

8.5.1 如何在 Universe 中映射和使用 SAP NetWeaver BW 对象

在根据信息块或 BEx 查询创建 Universe 时，Universe 设计工具会将 SAP NetWeaver BW OLAP 结构映射至 Universe 中等价的类和对象中。

BEx 查询中所有设置为行、列、自由特性和过滤器的 InfoObject 都会显示在 Universe 中。其中包括特性、层次结构、关键值、结构和变量。

层次结构被映射，以便 Web Intelligence 用户可以根据 BW 层次结构向下钻取。

对于信息块，所有维、关键值和层次结构都被映射。

下表显示为每个 BW 对象创建的 Universe 对象。

NetWeaver BW 对象:	创建的 Universe 对象:
维组	类
特性	带维和明细对象的子类
带层次结构的特性	<p>如果数据源是 BEx 查询：包含当前定义的层次结构中每个层次结构级别的维和明细对象的子类</p> <p>如果数据源是信息块：包含为此特性定义的所有层次结构中每个层次结构级别的维和明细对象的子类</p>
基于特性的结构（仅限 BEx 查询）	结构的带单个维对象的类
导航属性	带维和明细对象的子类（与特性相同）
显示属性	维的明细对象
关键值结构	类
关键值	关键值结构（带表示单位/货币的维对象）的类中的度量对象

NetWeaver BW 对象:	创建的 Universe 对象:
计算所得关键值 (仅限 BEx 查询)	度和维对象 (与关键值相同)
受限关键值 (仅限 BEx 查询)	度和维对象 (与关键值相同)
变量 (仅限 BEx 查询)	查询中的必备过滤器 在变量适用的维的类中, 有两个支持值列表的维对象, 一个用于标题, 另一个用于说明。
关键日期变量 (仅限 BEx 查询)	用于在 Universe 中定义关键日期变量的 Universe 参数

不映射 BEx 查询的“过滤器”节中的特性。但对 Universe 应用过滤。如果过滤器具有固定值, 则在运行 Web Intelligence 查询时以透明方式应用过滤器。如果为特性定义了变量, 则映射变量, 但受以下限制:

- 变量的行为始终与必需变量类似
- 支持层次结构和层次结构节点变量, 但层次结构版本变量除外

要回避这些限制, 请将特性从“过滤器”节移至 BEx 查询的“自由”节。

相关主题

- 第 512 页上的[“特性的映射和使用方式”](#)
- 第 513 页上的[“关键值的映射和使用方式”](#)
- 第 513 页上的[“层次结构的映射和使用方式”](#)
- 第 514 页上的[“Universe 中支持变量的方式”](#)
- 第 516 页上的[“变量映射到 Universe 的方式”](#)

8.5.1.1 特性的映射和使用方式

如果没有在 BEx 查询或信息块中根据特性定义层次结构, Universe 设计工具会创建一个类, 类中将特性作为两个维对象包括在内: 级别 00 和级别 01。“级别 00”维表示在选定所有成员时 (从 NetWeaver 返回的成员为“所有成员”) 特性的聚合。“级别 01”维将特性的所有成员作为一个简单值列表包括在内。

对于每个维对象，Universe 设计工具会为键创建一个明细对象、为说明（简短说明、适中说明和长说明）创建多达三个明细对象，并为每个显示属性创建一个明细对象。

使用特性的技术名称来定义 SELECT 子句。

BW 查询中定义的导航属性将按照映射特性的相同方式映射到父对象类中。

注意：

如果在 Universe 中定义大量导航属性，将对 Web Intelligence 的查询性能产生负面影响。

BEx 查询中定义的基于特性的结构以单维对象形式包括在 Universe 中，并且结构元素用作维成员。

8.5.1.2 关键值的映射和使用方式

I 信息块中或在 BEx 查询中定义的所有关键值均包括在 Universe 中一个称为“关键值”的单一对象类中。

大部分关键值在 NetWeaver BW 中定义，并包含货币或单位特性。对于每个关键值，Universe 设计工具将创建：

- 数字格式的度量对象，与不带单位的关键值相对应。
- 字符格式的维对象，包含单位或货币。例如，“美元”、“英镑”、“千米”。
- 字符格式的维对象，包含关键值和单位（带格式的值），基于 SAP 服务器上配置的用户首选项。例如，“200 美元”、“345 英镑”、“25 千米”。

“关键值”类包括计算所得的关键值以及 BEx 查询中定义的受限关键值。原始计算和限制应用于此查询，但不显示在 Universe 中。

8.5.1.3 层次结构的映射和使用方式

映射层次结构之后，Web Intelligence 用户就可以按与自定义 Universe 层次结构相同的方式根据 SAP NetWeaver BW 层次结构向下钻取。

注意：

Web Intelligence “文档属性”对话框中的“使用查询钻取”选项可显著提高向下钻取的性能。

当在 BEx 查询中根据特性定义层次结构时，Universe 设计工具会在 Universe 中创建一个分层结构，层次中的每个级别对应一个子类。此结构取决于当前的 BEx 查询定义：

- 如果在 BEx 查询中定义了层次结构，Universe 设计工具会在 Universe 中创建此分层结构。

- 如果在 BEx 查询中定义了一个层次结构变量以便用户在运行时选择层次结构，Universe 设计工具将在 Universe 中创建一个一般层次结构。此结构具有为该特性的任何可用分层结构定义的最高级别数。

在信息块之上构建 Universe 时，所有根据该特性定义的层次结构都显示在生成的 Universe 中。Universe 设计工具会为每个分层结构创建子类，每一个结构均包含对应于该层次级别的子类。

在 Universe 中，层次结构的“级别 00”表示此结构的顶部节点。当分层结构存在多个顶点时，“级别 00”维则按值列表形式包含所有顶部节点。当层次结构特性设置为不过滤未分配的节点时，必需使用未分配成员的顶部节点来包括“级别 00”。未分配成员在层次结构的最低级别上分组。

注意：

大多数情况下，SAP NetWeaver BW 层次结构只有一个顶部节点。用户可以从默认 Universe 中删除“级别 00”对象，以简化 Universe 的使用。通常情况下，只有在必需查询/报告未分配成员时才需要保留“级别 00”。

如果 BEx 查询中的层次结构级别数发生变化，必须更新此 Universe。

相关主题

- 第 476 页上的“[关于 OLAP Universe 生命周期管理](#)”

8.5.1.4 Universe 中支持变量的方式

SAP 变量可以解释为在 BW 查询中定义的用户提示。变量可以是必需的，也可以是可选的，并且可以包含默认值。

特性变量用于过滤特性的值。变量的值在执行查询时填入。变量可以存储特性值、层次结构、层次结构节点、文本和公式元素。

NetWeaver BW 变量仅应用于 BEx 查询。

注意：

在查询设计器中定义变量时，必须在 SAP NetWeaver BW 变量向导的“其他设置”对话框上选定“Ready for Input”（准备就绪可以输入）选项。

Universe 中支持以下类型的 SAP NetWeaver BW 变量：

- 特性变量
- 层次结构变量，层次结构版本变量除外
- 层次结构节点变量
- 货币变量
- 公式变量
- 文本变量（作为置换路径）
- 关键日期变量

下表显示 Universe 对需用户输入的 BW 变量的支持。用户录入的变量可以是必需的，也可以是可选的，并且可以包含默认值。

变量类型		支持级别
特性（包括关键日期和货币）	单一值提示	支持
	多单一值提示	支持
	间隔提示	支持 对于属于单值变量的关键日期变量不支持
	选择选项提示	做为间隔提示支持 对于属于单值变量的关键日期变量，不支持做为间隔提示
	预先计算的值集	不支持
文本		支持
公式		支持价格、配额和数值
层次结构		支持，但版本变量除外
层次结构节点		支持

下表显示 Universe 对其他处理类型的 BW 变量的支持。

变量类型	处理类型			
	置换路径	授权	客户出口	SAP 出口
特性	支持	支持	支持，不在 Universe 中创建提示	支持
文本	支持	不提供	支持	不提供
公式	支持	不提供	支持	支持，无用户输入
层次结构	不适用	不适用	支持	支持
层次结构节点	不适用	不适用	支持	支持，无用户输入

支持“排除”运算符，但是 Web Intelligence 未指定要从查询中排除选定值。其它运算符（例如“小于”和“大于”）只与“选择”选项录入类型一起使用。“选择”选项类型会转为 Web Intelligence 提示的间隔。

注意：

要在 Web Intelligence 中处理 BW 变量，必须在 Web Intelligence 查询中至少包括一个度量。

相关主题

- 第 516 页上的[“变量映射到 Universe 的方式”](#)
- 第 518 页上的[“Universe 中支持关键日期变量的方式”](#)
- 第 519 页上的[“Universe 中支持层次结构和层次结构节点变量的方式”](#)

8.5.1.4.1 变量映射到 Universe 的方式

即使没有在结果集中使用维，仍然需要向用户提示所有可选变量和必需变量的信息，以便用户限制结果集。因而即使相应特性不在 BEx 查询中，也会映射在此查询中定义的变量。

用户必须了解变量是必需的还是可选的，并可以忽略可选变量。可选变量在 Universe 中被定义为可选，并转为 Web Intelligence 中的可选提示。必需变量转为 Web Intelligence 中的必需提示。

对于特性变量，Universe 设计工具会在 Universe 中创建一个必需过滤器。必需过滤器是一个预定义的查询过滤器对象，它对 Web Intelligence 用户隐藏，但以系统透明的方式应用于在 Universe 上构建的所有 Web Intelligence 查询。

变量类型	映射至
特性变量，包括货币和公式变量	Universe 必需过滤器
层次结构变量	Universe 必需过滤器
层次结构节点变量	类必需过滤器
关键日期变量	Universe 参数

对于每个必需过滤器，将创建两个维对象作为 @Prompt 函数的引用对象，以显示预计的值列表。在 Universe 中值列表维是隐藏的。这些维是提示正常发挥作用所必需的，因而不能被删除，在移动和修改时，请务必小心。

变量的默认值是在过滤器的 @Prompt 函数中使用主键、持久/非持久以及默认值参数定义的。在 Universe 的过滤器的“属性”页中可以查看 @Prompt 函数的语法。

为避免 Web Intelligence 用户定义的 BW 变量和过滤器之间发生冲突，在生成 SAP 变量定义中使用的对象时，请取消选中对象属性的“高级”页上的“可在条件中使用”选项，从而禁止 Web Intelligence 用户在“过滤器”窗格中包括 SAP 变量中用到的维。

示例：为 SAP BW 变量生成的 Where 子句

本示例显示为基于维对象“Customer2”的 BW 变量生成的 Where 子句。用户可以在过滤器的“属性”页上查看为变量生成的 Where 子句的语法。

```
<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>
```

提示文本根据 BW 变量名生成。用户可以编辑此文本，以增强其可读性。

“Customer2\LovCustomer Variable Single Value MandatoryBase”是在构建值列表时使用的隐藏的 Universe 对象名称。

注意：

如果重命名此类，或将“值列表”对象移至其它文件夹，则必须在过滤器键中更新此语法。

8.5.1.4.2 支持变量和值列表的方式

一个 BEx 查询可以包含超过 10 个变量，也就是说可以加载 10 个或更多的值列表。加载和刷新值列表会对性能产生重要影响。以下选项可用于改善带变量的查询的查询性能：

- 在生成 Universe 时，所有 SAP BW 变量（关键日期除外）都被映射至必需过滤器。默认情况下，过滤器对象与值列表不关联（层次结构节点变量除外）。用户必须在对象属性页中显式关联值列表。
- 可选变量作为可选提示生成。可选提示不会在运行查询时自动加载值列表。
- 值列表属性上的委派搜索选项表示用户在运行查询时具有一个空值列表。输入搜索条件可以限制值列表中返回的值数量。

要激活值列表的委派搜索选项，请在值列表适用的对象的对象属性页上编辑值列表属性。

注意：

级联值列表不支持委派搜索。

相关主题

- 第 475 页上的[“OLAP Universe 中的可选提示”](#)

8.5.1.4.3 Universe 中支持关键日期变量的方式

使用 BEx 查询中的关键日期变量可以为依赖于时间的数据指定日期。关键日期会影响针对维检索的数据，例如，产品说明可能在一段时间内发生变化。关键日期会影响分层结构，例如，某个成本中心在某一年可能处于级别 01，而在另一年处于级别 02。

关键日期变量是特殊的 SAP BW 变量，因为用户输入的日期值未包括在 BW 查询的任何维中。关键日期是查询的一个属性。

在 BEx 查询中，可以为以下两种用途定义关键日期变量：

- 指定特定层次结构的有效日期，且仅影响该层次结构。
- 指定完成查询的日期。在此情况中，查询中设置的关键日期会影响以下各项：
 - 依赖于时间的主数据
 - 货币汇率
 - 层次结构列表
 - 依赖于时间的分层结构

注意：

在 Universe 中，关键日期的使用范围限定为整个 Universe。因此，在 Universe 中生成的关键日期会影响所有其它 SAP 变量和数据。

SAP NetWeaver BW 仅支持为每个 BW 查询使用一个关键日期变量，因此 Universe 仅包含一个关键日期变量。

关键日期变量可以是必需的，也可以是可选的，并且可以包含默认值。如果未定义默认值，而且用户也没有输入值，查询则使用当前的系统日期。

查询的关键日期变量属性被映射到 5 个 Universe 参数，如下表所述。

参数	说明
KEYDATE_ENABLED	如果在 Universe 上启用关键日期，则设置为“是”。
KEYDATE_NAME	关键日期变量的技术名称。
KEYDATE_CAPTION	在向用户提示输入值时显示的关键日期变量的标题。
KEYDATE_DEFAULT_VALUE	关键日期的默认值（如果有）。

参数	说明
KEYDATE_MANDATORY	如果用户必须输入值或使用默认值，则设置为“是”。

在查询运行时，Web Intelligence 会向所有查询建议相同的关键日期。用户可以修改关键日期。“关键日期属性”对话框可用于管理使用哪个关键日期。在提示任何其它任何类型的变量之前，系统会先向用户提示关键日期。

8.5.1.4.4 Universe 中支持层次结构和层次结构节点变量的方式

层次结构变量用于向用户提示关于将在查询中使用的层次结构信息。Web Intelligence 用户可以创建查询和报表，以检索和显示任何层次结构的成员。

如果层次结构变量是可选的，并且用户将提示留空，报表中则不使用层次结构。

报表包含最大层次结构级别数，且独立于所选的层次结构。结果集中未返回的层次结构级别在报表中为空白。

层次结构节点变量用于向用户提示关于要定义为查询中层次结构顶部节点的节点信息。

当查询中同时包含层次结构和层次结构节点变量时，Web Intelligence 用户必须首先在可用层次结构的列表选择一个层次结构。然后选择层次结构节点。可用层次结构节点列表显示所有层次结构的层次结构节点。此列表没有基于选定的层次结构进行过滤。用户负责从正确的层次结构选择节点。从不同层次结构选择层次结构节点会导致报表为空。

相关主题

- 第 513 页上的[“层次结构的映射和使用方式”](#)

8.5.2 如何将 Essbase 多维数据集映射到 Universe 元素

Universe 设计工具通过将 Essbase 大纲映射到对等类和对象的方式，依据 Essbase 多维数据集创建 Universe。用户可以在创建连接时确定多维数据集数据源。

Essbase 别名表为大纲中的维、级别和成员定义了一组别名。Universe 设计工具根据用户在创建 Essbase 数据源的连接时选择的别名表，使用这些名称生成 Universe。

在 Essbase 大纲中，度量被定义为维。请选择要在创建 Essbase 数据源的连接时用作度量维的那个维。Universe 设计工具生成该维的成员，充当 Universe 中的度量。

任何维均支持多层的层次结构。最多只能为每个维定义一个层次结构。

下表显示在 Universe 中为每个 Essbase 大纲元素创建的对象。

Essbase 对象	创建的 Universe 对象:
维	包含维生成的类。
生成	维类中的对象，包含两个明细对象：一个用于标题，一个用于名称。
度量维	根据选作 Universe 连接中的度量维的那个维命名的类（通常为“度量”类或“帐户”类）。
度量	度量类或子类中的度量对象。这些度量使用类和子类组成的结构（与 Essbase 大纲中的结构匹配）进行创建。

使用聚合投影函数（默认设置为“委派给数据库”）生成度量。在刷新 Web Intelligence 报表时，度量的聚合操作委派给数据库服务器运行。

相关主题

- 第 451 页上的[“关于 OLAP 数据源的连接”](#)
- 第 463 页上的[“投影函数“委派给数据库””](#)

8.5.3 将 MSAS 多维数据集映射到 Universe 元素的方法

Universe 设计工具通过将 MSAS 结构映射到对等类和对象的方式，依据 MSAS 多维数据集创建 Universe。用户可以在创建连接时指定多维数据集数据源。

下表显示在 Universe 结构中为每个 MSAS 对象创建的对象。此映射应用于 MSAS 虚拟多维数据集、本地多维数据集（.cub 文件）以及 MSAS 标准多维数据集。

MSAS 对象:	创建的 Universe 对象:
维	包含维对象的类。

MSAS 对象:	创建的 Universe 对象:
显示文件夹 (MSAS 2005)	维类中的子类。
层次结构	相应维类中的子类，或相应的显示文件夹类中的子类。
特性 (MSAS 2005)	相应维类中的子类，或相应的显示文件夹类中的子类。
度量	包含所有度量对象的度量类。度量对象在度量类或度量组的子类中创建。
度量组 (MSAS 2005)	度量类中的子类。
级别	维类或子类中的对象，以及表示所有子级别聚合的“所有级别”对象。
级别属性	适用于级别对象的明细。

使用聚合投影函数（默认设置为“委派给数据库”）生成度量。在刷新 Web Intelligence 报表时，度量的聚合操作委派给数据库服务器运行。

相关主题

- 第 451 页上的“[关于 OLAP 数据源的连接](#)”
- 第 463 页上的“[投影函数“委派给数据库”](#)”

第 9 章 使用来自元数据源的 Universe

9.1 有关从元数据源生成 Universe 的介绍

Universe 设计工具中的“元数据交换”允许从其他数据仓库产品生成的 XML 文件中创建 Universe。它将分析 XML 文件中的内容以提取元数据信息，并将其转换成 BusinessObjects 元数据，包括类、对象、表、列、定制层次结构和联接。然后，此应用程序将创建一个新的 BusinessObjects Universe。也可以从其他元数据源中创建 Universe。

可使用 Universe 设计工具从以下元数据源创建 Universe：

元数据源	名称
符合以下标准的 XML 文件	<ul style="list-style-type: none">• 公共仓库模型 (CWM 1.0)• 公共仓库模型 OLAP (CWM OLAP)• Oracle Warehouse Builder• BusinessObjects Data Integrator• IBM DB2 Data Warehouse Center• IBM DB2 Cube Views
数据库视图	Oracle 分析工作区

也可以使用 Universe 设计工具更新使用某些 XML 元数据源的 Universe，并将 Universe 导出成 DB2 Cube Views (DB2CV) XML 格式。

9.2 概述

使用“元数据交换”面板（“文件”>“元数据交换”）选择元数据格式。这是目标元数据源文件所使用的格式。有关信息，请参阅“选择元数据源”一节。

一旦选定了格式，可以通过 Universe 构建器向导的指引来选择目标数据库，并选择要用于生成 Universe 的结构。然后，选择目标连接，并生成 Universe。

所有 XML 元数据源的 Universe 创建过程都是一样的。第 524 页上的 [“依据 XML 源创建 Universe”](#) 一节中描述了如何依据 XML 元数据源创建 Universe。

Oracle 分析工作区数据源的 Universe 创建过程与上述过程不同。一旦选择了连接，特定于选定的元数据源的 Universe 创建面板将会出现。每个受支持的元数据源的特定小节中详细描述了受支持的元数据源。

一旦从元数据源创建了 Universe，就可以像修改任何其他 Universe 那样去修改此 Universe 的任何组成部分。

可以保存 Universe 并将其导出到中央管理系统（CMS）。导出到 CMS 后，Web Intelligence 用户可以使用此 Universe 创建查询和报表。

9.3 Universe 创建概述

使用“元数据交换”面板（“文件”>“元数据交换”）选择元数据格式。这是目标元数据源文件所使用的格式。有关信息，请参阅第 523 页上的 [“选择元数据源”](#) 一节。

一旦选定了格式，可以通过 Universe 构建器向导的指引来选择目标数据库，并选择要用于生成 Universe 的结构。然后，选择目标连接，并生成 Universe。

所有 XML 元数据源的 Universe 创建过程都是一样的。第 524 页上的 [“依据 XML 源创建 Universe”](#) 一节中描述了如何依据 XML 元数据源创建 Universe。

Oracle 分析工作区数据源的 Universe 创建过程与上述过程不同。一旦选择了连接，特定于 Oracle 分析工作区的 Universe 创建面板将会出现。依据数据库创建视图，然后从视图中创建 Universe。

依据元数据源创建了 Universe 后，就可以像修改任何其他 Universe 一样修改此 Universe 的任何组成部分。

可以保存 Universe 并将其导出到中央管理系统（CMS）。导出到 CMS 后，Web Intelligence 用户可以使用此 Universe 创建查询和报表。

9.4 选择元数据源

选择元数据源以从“元数据交换”面板（“文件”>“元数据交换”）中创建或更新 Universe。也可以选择 Universe 以导出成 DB2CV XML 格式。

在“元数据交换”面板中可以使用以下选项：

“元数据交换”选项	说明
从视图创建 Universe	从下拉列表中选择元数据源格式。这是用于构建 Universe 的源 XML 文件或数据库视图。Universe 创建向导可引导逐步完成选择元数据源连接，选择要映射到 Universe 的元数据元素，直至最终生成 Universe。
依据以下项目更新 Universe	选择已更新的元数据源。这是已用于创建 Universe 的元数据源。此源已更新，现在需要用同样的修改来更新 Universe。Universe 更新向导将指引您完成更新 Universe 所需的步骤。
将 Universe 导出至	选择一种元数据格式，可将 Universe 导出为该格式。例如，可以选择 DB2CV XML 标准，然后用该格式保存 Universe。

9.5 依据 XML 源创建 Universe

通过“元数据交换”（“文件”>“元数据交换”）中的可用 OLAP Universe 构建器向导指引，从 XML 元数据源中创建 Universe。在生成 Universe 前，可设置 Universe 连接和生成选项。

相关主题

- 第 524 页上的[“XML 元数据源”](#)

9.5.1 XML 元数据源

可以依据符合以下数据源标准的 XML 文件创建 Universe：

- 公共仓库模型（CWM Relational 1.0）

- 公共仓库模型 OLAP (CWM OLAP)
- Oracle Warehouse Builder (Oracle WB)
- Data Integrator
- IBM DB2 Data Warehouse Center (IBM DB2 DWC)
- IBM DB2 Cube Views

通过“元数据交换”（“文件”>“元数据交换”）中的可用 OLAP Universe 构建器向导指引，从 XML 元数据源中创建 Universe。

9.5.2 依据 XML 元数据源生成 Universe

- 1 选择“文件”>“元数据交换”。

此时将出现“元数据交换”面板。

- 2 从“依据以下项创建 Universe:”下拉列表框中选择元数据格式。

单击“确定”。

Universe 构建器向导将启动。

单击“下一步”。

此时将出现 XML 文件源页面。

- 3 单击“浏览”按钮并选择 XML 源文件。这是要用于生成 Universe 的文件。

单击“下一步”。

此时将出现“选择数据库”页面。

- 4 单击源数据库。

单击“下一步”。

此时将出现 Universe 元素页面。在左窗格中列出了可用的数据库表和列。

- 5 选择一个或多个表和列，然后单击右箭头以填充右窗格。右窗格中的表和列是那些出现在生成的 Universe 中的表和列。可以根据需要使用箭头按钮从 Universe 窗格中添加和删除表。

单击“下一步”。

此时出现连接和 Universe 属性页面。其中列出了可供 Universe 设计工具使用的连接。

- 6 在连接列表中单击一个连接。这是 Universe 用于检索数据的数据源连接。

键入 Universe 的名称。

选中或清除选项复选框。单击“高级”按钮以设置跟踪日志文件和 XML 源文件选项。

单击“下一步”。

此时将出现 Universe 生成摘要页面。它列出了在执行向导过程中选定的选项的摘要。

单击“完成”。

将在 Universe 设计工具的“Universe”和“结构”窗格中显示生成的 Universe。

9.5.3 选择连接和 Universe 选项

元数据 Universe 构建器向导的连接和 Universe 生成页面上具有以下选项：

向导页面	Universe 选项	说明
生成 Universe 设置	选择连接	列出的连接即为可供 Universe 设计工具使用的连接。这是指向目标 RDBMS 的连接。
	Universe 名称	即将生成的 Universe 名称。
	自动保存 Universe	选中时，在创建中保存 Universe。
	替换现有 Universe	选中时，如果存在同名的 Universe，并且选择了“自动保存 Universe”，则新的 Universe 将替换现有的 Universe。
高级设置	“常规”选项卡 跟踪	跟踪文件夹路径。这是在生成 Universes 时保存日志文件的文件夹。可以浏览到某个文件夹并选择此文件夹。

向导页面	Universe 选项	说明
	“文件位置”选项卡 默认 XML 源文件文件夹	存储用于创建 Universe 的 XML 文件的默认文件夹路径。可以浏览到某个文件夹并选择此文件夹。
	“文件位置”选项卡 参数文件	存储参数文件的默认文件夹路径。这些是创建 Universe 时创建的文件。这些文件存储并引用选定的元数据，以便能够再次使用选定的元数据来创建或更新其他 Universe。参数文件并不存储选定的元数据。它是一个过滤器，可通过原始的 XML 文件将元数据桥指向选定的元数据。可以浏览到某个文件夹并选择此文件夹。

9.5.4 依据 XML 元数据源更新 Universe

- 1 选择“文件”>“元数据交换”。此时将出现“元数据交换”面板。
- 2 从“依据以下产品更新 Universe”下拉列表框中选择一种元数据格式。单击“确定”。Universe 构建器向导将启动。单击“下一步”。此时将出现 XML 文件源页面。
- 3 单击“浏览”按钮并选择 XML 源文件。这是要用于更新 Universe 的文件。单击“下一步”。此时将出现“选择数据库”页面。
- 4 单击源数据库。单击“下一步”。此时将出现 Universe 文件页面。单击“浏览”按钮并选择 Universe。这是要从选定的 XML 元数据源中更新的 Universe。单击“下一步”。此时将出现 Universe 元素页面。在左窗格中列出了可用的数据库表和列。已添加或修改的表标有一红色复选标记。
- 5 选择一个或多个表和列，然后单击右箭头，以将已修改的表填充至右窗格。右窗格中的表和列是那些出现在生成的 Universe 中的表和列。可以根据需要使用箭头按钮从 Universe 窗格中添加和删除表。单击“下一步”。此时出现连接和 Universe 属性页面。其中列出了可供 Universe 设计工具使用的连接。第 526 页上的 [“选择连接和 Universe 选项”](#)一节中对这些连接进行了说明。
- 6 在连接列表中单击一个连接。这是 Universe 用于检索数据的数据源连接。键入 Universe 的名称。选中或清除选项复选框。单击“高级”按钮以设置跟踪日志文件和 XML 源文件选项。单击“下一步”。此时将出现 Universe 生成摘要页面。它列出了在执行向导过程中选定的选项的摘要。单击“完成”。将在 Universe 设计工具的 Universe 和结构窗格中显示更新后的 Universe。

9.6 将 Universe 导出到 DB2CV

可以将 Universe 导出为 IBM DB2 Cube Views XML 格式文件。

将 Universe 的定义导出到与 IBM DB2 Cube Views XML 格式兼容的 XML 文件。通过使用 API 或“OLAP 中心”工具，可以在 IBM DB2 Cube Views 中加载此文件。IBM DB2 Cube Views 从 XML 文件中读取元数据，并推荐合适的自动汇总表（AST）以优化以后的查询。

相关主题

- 第 528 页上的[“针对导出的 Universe 先决条件”](#)
- 第 529 页上的[“标识 Universe 元数据”](#)

9.6.1 针对导出的 Universe 先决条件

以下列表描述了要将 Universe 成功导出为 XML 文件所必需的 Universe 先决条件：

Universe 级别限制

- 每个 Universe 都会被导出到多维数据集模型。
- Universe 必须将单一雪花型模式与单一事实表相匹配。
- Universe 至少必须包含一个度量。
- 不支持 Universe 之间的链接。
- 不会考虑上下文，因此也不会导出上下文。
- 定制层次结构：必须将定制层次结构的各个级别分组在同一个类中。

类和对象

- 唯一支持的 @ 函数是 @Select 函数。导出过程中不会映射所有其他 @ 函数。
- 不会导出对象定义的“Where”字段中的条件。注意：DB2 Cube Views 对象中不支持条件，因为它们不用于优化。
- 不会导出多参数聚合函数。
- 每个类都必须包含依据相同维表构建的对象。

- 必须将 IBMDB2CV 中由同一个维引用的所有对象（IBMDB2CV 中的特性）分组在 Universe 的同一个类中。其他类中的任何其他度量将被自动添加到 IBMDB2CV 中的 Fact 对象。

联接

如果联接左边或右边的列与 Universe 中的对象不匹配，则会自动为此列创建一个特性，并将其添加到包含该列的表的维（或事实）。

9.6.2 标识 Universe 元数据

本节描述在将 Universe 定义导出为 XML 文件的过程中，如何标识和处理在 IBM DB2 Cube Views 中没有多维匹配项的 Universe 中的对象。

Universe 包含关系元数据

Universe 基于没有多维设计约束的关系元数据。Universe 中的所有对象不必匹配对等的 IBM DB2 Cube Views 对象，并且不会遵循 IBM DB2 Cube Views 多维规则。

要正确地匹配关系结构，BusinessObjects UMB 必须运行某些自动检测过程，以便为 IBM DB2 Cube Views 标识并定义必需且适当的元数据。下面描述了受影响的多维对象。

Fact

IBM DB2 Cube Views 的 Fact 对象是依据在 Universe 中找到的一组度量自动构建的。

维

未标识为事实的表被假定为维表。IBM DB2 Cube Views 维对象是直接依据 BusinessObjects 的类推导出的。

类中的所有对象确定了 IBM DB2 Cube Views 维的特性。依据类中的 BusinessObjects 对象推导出的表可通过分析对象的“选择”字段进行检测。

特性

特性不是直接依据 Universe 中的表列推导出的。通过以下信息检测和标识候选特性：

- 类中的 BusinessObjects 对象。
- 由 @Select 语句在其他 BusinessObjects 对象的“选择”字段中引用的 BusinessObjects 对象。
- 联接中涉及的列。

特性关系

Universe 中的明细-维关系将转换为 IBM DB2 Cube Views 中类型为“Functional Dependency（功能相关性）”的特性关系。

联接

联接及其属性是直接从 Universe 结构中读取的。

度量

将在所有类中搜索度量对象。如果某个度量不是依据相同事实表构建的，则会忽略该度量。

层次结构

DB2 Cube Views 中的层次结构将链接到维对象，并且该层次结构的所有级别将成为此相同维的成员。在 Universe 中则不是这样，其中的定制层次结构可包含来自不同 Business Objects 类的级别。按以下所述来对待层次结构：

- 如果 Universe 只使用默认层次结构，则会使用每个类中的对象顺序依据对象推导出用于导出到 IBM DB2 Cube Views 的层次结构。
- 如果 Universe 具有定制层次结构，则无需进行任何修改即可导出它们。

9.6.3 将 Universe 导出为 DB2CV XML 文件

按如下方式将 BusinessObjects Universe 导出为 IBM DB2 Cube Views XML 文件：

9.6.4 Universe 到 DB2CV 元数据的映射

本节详细描述了 Universe 结构和 IBM DB2 Cube Views 结构之间的映射。

以下各节详细说明将 Universe 导出为 XML 文件时从 Universe 映射到 IBM DB2 Cube Views 的结构。

Universe 到多维数据集模型

下表描述了 Universe 到多维数据集模型的映射

Universe 属性	多维数据集属性
简短名称（文件名称）	多维数据集文件名

Universe 属性	多维数据集属性
Universe 名称（长名称）	业务名称 默认情况下，名称为 Universe 短名称（<Universe 短名称>）
说明	注释
事实表名称	factsRef
类列表	dimensionRef
涉及事实表的联接的列表。	joinRef

类到维

下表描述了类到维的映射：

类属性	维属性
名称	名称和业务名称
说明	注释
维和明细对象的列表有关度量映射，请参阅下面的“度量到度量”表。	attributeRef
由类推导出的维表之间的联接。	joinRef
层次结构	如果层次结构是定制层次结构，则会按照 IBM DB2 Cube Views 的要求修改维以获取相同维中的所有层次结构级别。 这些层次结构放在 hierarchyRef 属性中。

事实表到事实

下表描述了事实表到事实的映射：

事实表属性	事实属性
事实表名称 在“导出 Universe”面板的事实框中手动输入此名称。也可以接受默认名称 Facts_<Universe 名称>。	名称和业务名称
表说明	注释
Universe 中所有度量的列表	measureRef
度量中引用的列和对象的列表	attributeRef

度量到度量

下表描述了度量到度量的映射：

度量属性	度量属性
名称	名称和业务名称
说明	注释
由 Select 语句推导出的列和对象。	sqlExpression 列
Select 语句公式	sqlExpression 模板
聚合函数	聚合函数

维和明细对象到特性

下表描述了维和明细到特性的映射：

维和明细对象	特性
名称	名称和业务名称

维和明细对象	特性
说明	注释
Select 语句中引用的列和对象	sqlExpression 列
Select 语句公式	sqlExpression 模板

维和明细关系到特性关系

下表描述了维/明细关系到特性关系的映射

维/明细关系	特性关系
维名称 + 明细名称 级联字符为 “_”	名称和业务名称
维	左特性
明细	右特性

默认层次结构到层次结构

下表描述了默认层次结构到层次结构的映射：

默认层次结构	层次结构
名称	名称和业务名称
对象的列表。 明细对象一定不能是层次结构的一部分。	AttributeRef

注意：

如果没有定制层次结构，则会使用类作为层次结构。

定制层次结构到层次结构

下表描述了定制层次结构到层次结构的映射：

定制层次结构	层次结构
名称	名称和业务名称
对象的列表	attributeRef

联接到联接

下表描述了联接到联接的映射：

联接	联接
左侧表名称 + 右侧表名称。 级联字符为 “_”	名称和业务名称
左列	左特性
右列	右特性
复杂表达式： 对于每个简单表达式，将确定左列和右列。	每个简单表达式都映射到一个特性对。

9.6.5 映射特定 SQL 表达式

某些 SQL 表达式是由导出过程按特定方式映射的。本节将介绍以下几种 SQL 表达式：

- 度量的 SELECT 表达式
- @AggregateAware 函数
- 复合联接表达式
- Theta 联接
- 快捷联接

度量的 SELECT 表达式

BusinessObjects UMB 从度量的 SELECT 中获取以下信息：

- 检测度量中涉及的表和列，并将它们映射到 sqlExpression:column
- 确定聚合函数
- 确定公式表达式，并将它映射到 sqlExpression:template。

@AggregateAware 函数

如果对象包含 @AggregateAware 函数，将只会考虑 @AggregateAware 函数的最后一个参数。这是包含函数所使用最低聚合级别的表达式。例如：

Universe 中度量表达式的 @AggregateAware 表达式如下所示：

```
@Aggregate_Aware(
sum(AggregatedTable1.Sales_revenue),
sum(AggregatedTable2.Sales_revenue),
sum(Fact_Table.Amount_sold))
```

映射到 IBM DB2 Cube Views 的表达式为：

```
sum(Fact_Table.Amount_sold))
```

复合联接表达式

Universe 中复合联接的表达式可包含如下类型的表达式：

```
LeftTable.Column=RightTable.Column
```

在复合联接中，可使用 AND 运算符将这些类型的表达式链接在一起。BusinessObjects UMB 在同一联接内将复合联接中的每个表达式映射到 IBM DB2 Cube Views 的一个特性对。

Theta 联接

一个 Theta 联接将分割为两个 IBM DB2 Cube Views 联接，其中运算符 BETWEEN 被替换为运算符 <= 和 >=。例如：

Universe 中的联接具有以下表达式：

```
Customer.age between Age_group.age_min and Age_group.age_max
```

此联接将分割为具有以下表达式的两个联接：

```
Join1: Customer.age >= Age_group.age_min
Join2: Customer.age <= Age_group.age_max
```

快捷联接

不导出到 IBM DB2 Cube Views。Universe 中的快捷联接不会考虑中间表，从而提供了另一种提升查询性能的途径。由于快捷联接会在多维数据集模型中创建环路，因此不会导出它们。

9.7 Oracle 分析工作区

使用 Oracle OLAP Universe 构建器向导可引导用户完成创建 Universe 的步骤。从“元数据交换”面板（“文件”>“元数据交换”）中连接到 Oracle OLAP Universe 构建器向导。

如何使用 Oracle Universe 构建器向导创建 Universe 的概述如下：

启动“元数据交换”，并从“依据以下产品创建 Universe”下拉列表中选择“Oracle OLAP”。

Oracle OLAP Universe 构建器向导将启动。完成以下阶段：

- 连接到要用于构建 Universe 的信息提供者。
- 选择一个数据库。
- 选择充当目标元数据源的多维数据集。
- 基于多维数据集元数据创建一个视图。
- 基于此视图生成 Universe。

也可以从现有视图中创建 Universe。

9.7.1 如何从 OLAP 多维数据集中生成 Universe？

使用 Oracle OLAP Universe 构建器创建 Universe 时，它会自动设置为可通过 SQL 访问 Oracle 分析工作区。BusinessObjects Oracle OLAP Universe 构建器将执行以下主要任务：

- 将关系事实视图作为实际视图或派生表插入 Universe 中。
- 添加用于表示维级别和层次结构的别名
- 利用常规联接和快捷联接将关系视图联接到维表。联接的表达式特定于此解决方法。
- 为每个多维数据集维创建一个对象类，并为维的每个级别创建一个对象。
- 如果维具有多个层次结构，则为每个层次结构创建一个子类。在视图定义和 Universe 中均支持多层次结构的维。
- 定义聚合导航，以解决由于多层次结构维而导致的对象不兼容问题。
- 使用 AggregateAware 函数来定义对象表达式，以处理聚合导航
- 将映射实际维成员（标识符）的对象转换为表示成员说明的对象的明细。
- 创建度量对象

9.7.2 将 Oracle OLAP 结构映射到 Universe 组成部分

本节说明如何从 Oracle OLAP 多维数据集结构中创建 Universe。它说明了生成的 Universe 结构，并回答了一些有关映射过程的一般问题。

9.7.3 分析关系视图

BusinessObjects Oracle OLAP Universe 构建器生成一些视图，这些视图调用 OLAP_TABLE 函数，将视图列映射到多维数据集的维和度量的层次结构。生成的脚本具有以下形式：

```
CREATE VIEW BOBJ_FK_UNITS_CUBE_VIEW AS SELECT * FROM TABLE(OLAP_TABLE('GLOBAL_AW2.TEST DURATION session','','','&LIMIT_MAP'
```

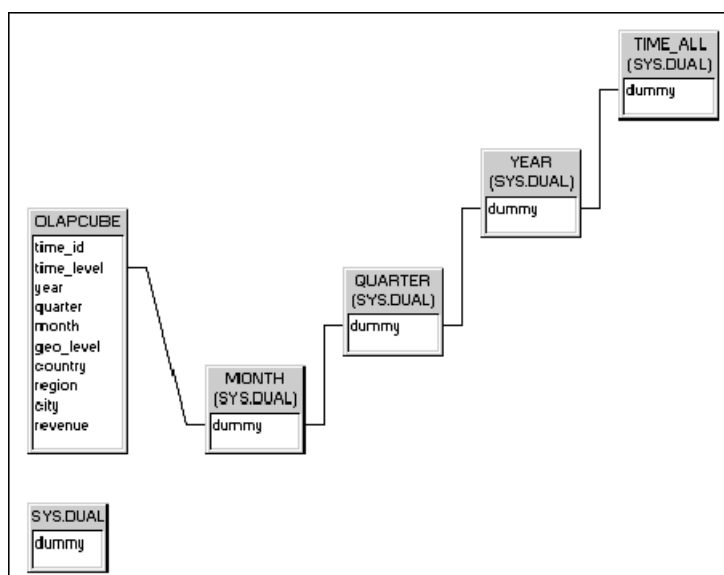
LIMIT_MAP 是一个变量，用于存储 OLAP_TABLE 的 limit_map 参数的文本。此文本是由 Oracle OLAP Universe 构建器生成的。以下是 limit_map 参数的示例：

```
DIMENSION GLOBAL_AW2.TEST!FK_TIME WITH
HIERARCHY GLOBAL_AW2.TEST!FK_TIME_PARENTREL (FK_TIME_HIERLIST \'CALENDAR\')
LEVELREL FK_TIME_YEAR,FK_TIME_QUARTER,FK_TIME_MONTH
FROM GLOBAL_AW2.TEST!FK_TIME_FAMILYREL USING GLOBAL_AW2.TEST!FK_TIME_LEVLIST
LEVELREL FK_TIME_YEAR_DESC,FK_TIME_QUARTER_DESC,FK_TIME_MONTH_DESC
FROM GLOBAL_AW2.TEST!FK_TIME_FAMILYREL USING GLOBAL_AW2.TEST!FK_TIME_LEVLIST
LABEL GLOBAL_AW2.TEST!FK_TIME_LONG_DESCRIPTION
ATTRIBUTE FK_TIME_LEVEL FROM GLOBAL_AW2.TEST!FK_TIME_LEVELREL
DIMENSION GLOBAL_AW2.TEST!FK_CUSTOMER WITH
HIERARCHY GLOBAL_AW2.TEST!FK_CUSTOMER_PARENTREL (FK_CUSTOMER_HIERLIST \'MARKET_SEGMENT\')
INHIERARCHY GLOBAL_AW2.TEST!FK_CUSTOMER_INHIER
LEVELREL null,null,null,FK_CUSTOMER_TOTAL_MARKET,FK_CUSTOMER_MARKET_SEGMENT,
FK_CUSTOMER_ACCOUNT,FK_CUSTOMER_SHIP_TO
FROM GLOBAL_AW2.TEST!FK_CUSTOMER_FAMILYREL USING GLOBAL_AW2.TEST!FK_CUSTOMER_LEVLIST
LEVELREL null,null,null,FK_CUSTOMER_TOTAL_MARKET_DESC,FK_CUSTOMER_MARKET_SEGMENT_D01,
FK_CUSTOMER_ACCOUNT_DESC,FK_CUSTOMER_SHIP_TO_DESC
FROM GLOBAL_AW2.TEST!FK_CUSTOMER_FAMILYREL USING GLOBAL_AW2.TEST!FK_CUSTOMER_LEVLIST
LABEL GLOBAL_AW2.TEST!FK_CUSTOMER_LONG_DESCRIPTION
HIERARCHY GLOBAL_AW2.TEST!FK_CUSTOMER_PARENTREL (FK_CUSTOMER_HIERLIST \'SHIPMENTS\')
INHIERARCHY GLOBAL_AW2.TEST!FK_CUSTOMER_INHIER
LEVELREL null,null,null,FK_CUSTOMER_ALL_CUSTOMERS,
FK_CUSTOMER_REGION,FK_CUSTOMER_WAREHOUSE,null
FROM GLOBAL_AW2.TEST!FK_CUSTOMER_FAMILYREL USING GLOBAL_AW2.TEST!FK_CUSTOMER_LEVLIST
LEVELREL null,null,null,FK_CUSTOMER_ALL_CUSTOMERS_DESC,
FK_CUSTOMER_REGION_DESC,FK_CUSTOMER_WAREHOUSE_DESC,null
FROM GLOBAL_AW2.TEST!FK_CUSTOMER_FAMILYREL USING GLOBAL_AW2.TEST!FK_CUSTOMER_LEVLIST
LABEL GLOBAL_AW2.TEST!FK_CUSTOMER_LONG_DESCRIPTION
ATTRIBUTE FK_CUSTOMER_LEVEL FROM GLOBAL_AW2.TEST!FK_CUSTOMER_LEVELREL
MEASURE FK_UNITS_CUBE_UNITS AS NUMBER FROM GLOBAL_AW2.TEST!FK_UNITS_CUBE_UNITS
ROW2CELL OLAP_CALC
```

9.7.4 Universe 中的快捷链接有何具体用途？

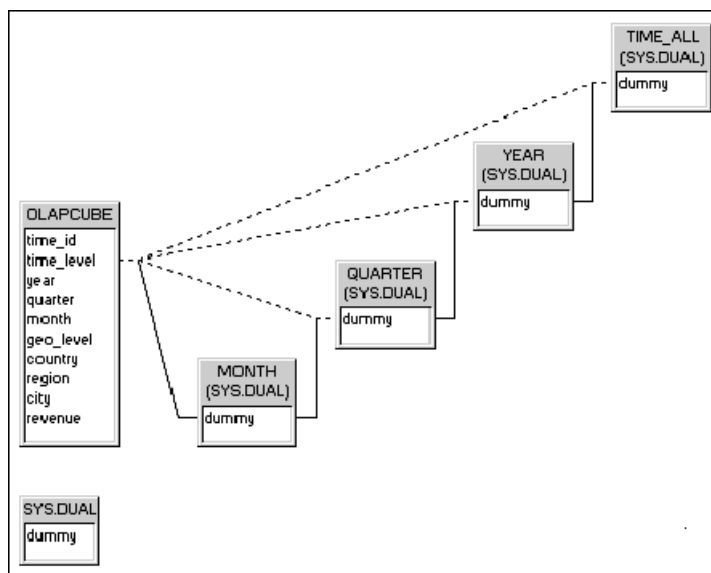
快捷链接确保 BusinessObjects 为每个对象组合而不是为每个对象生成 SQL。

BusinessObjects 在以下情况中使用快捷链接：可以忽略查询中的表，并且可以在两个在层次结构中并未直接链接的表之间建立“捷径”。例如，根据以下模式：



如果在 QUARTER 和 OLAPCUBE 表之间定义快捷链接，则 BusinessObjects 不需要通过 MONTH 表进行联接就能按季度检索收入。

必须通过快捷链接将时间层次结构中的每个表（最低级别的表除外）联接到 OLAPCUBE.time_level，如下所示：



联接表达式必须包含用于限制从 OLAPCUBE 中返回的行的表达式；对 QUARTER 而言，此表达式为 OLAPCUBE.time_level = 'QTR'。要确保 Universe 设计工具允许此联接，表达式还必须引用 MONTH 表，而此表应出现在注释中（因为它在所关心的实际联接表达式的生成中不起作用）。因此，完整的联接表达式为：

```
/* QUARTER.DUMMY */ OLAPCUBE.time_level = 'QTR'
```

此示例时间层次结构的快捷联接表达式的完整列表如下：

联接的表	表达式
MONTH、OLAPCUBE	/* MONTH.DUMMY */ OLAPCUBE.time_level = 'MONTH'
QUARTER、OLAPCUBE	/* QUARTER.DUMMY */ OLAPCUBE.time_level = 'QTR'
YEAR、OLAPCUBE	/* YEAR.DUMMY */ OLAPCUBE.time_level = 'YEAR'
TIME_ALL、OLAPCUBE	/* TIME_ALL.DUMMY */ OLAPCUBE.time_level = 'ALL'

9.7.5 Oracle OLAP 结构如何映射到 Universe 组成部分？

为了获得和设置预期的 Universe，Oracle OLAP Universe 构建器将按如下方式添加和配置 Universe 对象：

视图

Oracle OLAP Universe 构建器将关系视图作为表插入 Universe 和 oracle 表 sys.dual 中。如果选择使用派生表，则随视图的定义一起插入派生表（利用 OLAP_TABLE 函数选择部件）。

层次结构表

对于在关系视图中呈现的每个层次结构，将为层次结构的每个级别创建别名 sys.dual。别名的名称是级别名称。例如：如果有一个包含 4 个级别（ALL、YEAR、MONTH、QUARTER）的维度 TIME，则创建 4 个别名：ALL、YEAR、MONTH、QUARTER。

多层次结构的表

注意：

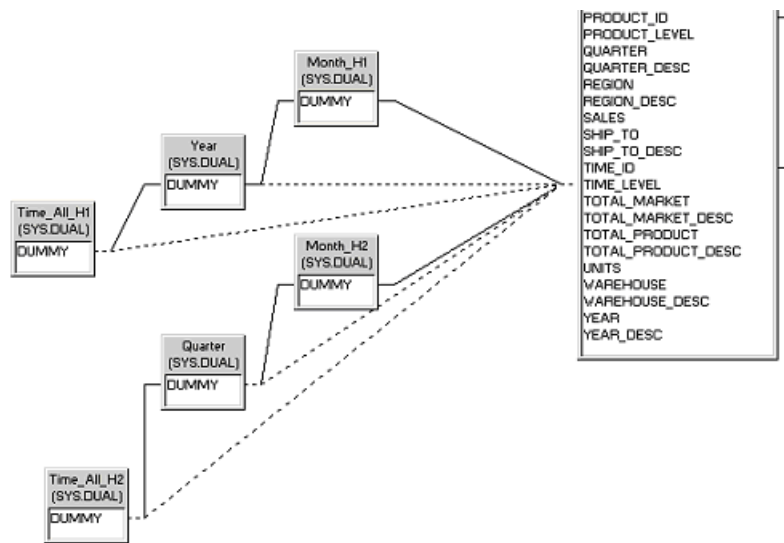
多层次结构是一种特殊情况。有关更多信息，请参阅本章的最后一节。

如果维具有多个层次结构，则将为每个层次结构创建一组不同的表，即使一些层次结构共享相同的级别也是如此。这意味着，对于共享的级别，层次结构有多少个，就将创建多少个别名。通过连接级别名称和层次结构名称对此类别名进行命名。例如：

time 维具有两个层次结构：H1 (All_Time, Year, Month) 和 H2 (All_Time, Quarter, Month)

All_Time 和 Month 是两个层次结构共有的，因此 All_Time 将有两个别名：All_Time_H1 和 All_Time_H2

Month 也有两个别名：Month_H1 和 Month_H2



维联接

- 表示级别的每个表被联接到同一层次结构中直接位于其下的级别。联接表达式为：

```
/* Alias1.DUMMY=Alias2.DUMMY */ 1=1
```

其中 Alias1 表示一个级别，而 Alias2 表示层次结构中直接位于其上的级别。示例：

```
/* Quarter.DUMMY=Year.DUMMY */ 1=1
```

- 每个表均使用快捷联接类型联接到视图（最低级别除外，最低级别使用常规联接）。联接表达式定义一个值，用于过滤从视图中返回的行，而且此联接表达式的形式如下：

```
/* Alias.DUMMY */
```

```
VIEW.levelColumn = 'level_value'
```

其中 Alias 是别名的名称，levelColumn 是表示视图中的级别的列，而 level_value 是该列的值（与级别名称匹配）。

示例：MYVIEW 是表示 OLAP 多维数据集的视图，而包含级别的列是 time_level，级别的值是：ALL、YEAR、QTR、MONTH

联接的表的表达式

```
MONTH, MYVIEW /* MONTH.DUMMY */ MYVIEW.time_level = 'MONTH'
QUARTER, MYVIEW /* QUARTER.DUMMY */ MYVIEW.time_level = 'QTR'
YEAR, MYVIEW /* YEAR.DUMMY */ MYVIEW.time_level = 'YEAR'
TIME_ALL, MYVIEW /* TIME_ALL.DUMMY */ MYVIEW.time_level = 'ALL'
```

映射类和对象

元数据桥必须为每个 OLAP 维创建一个类，并为每个级别创建一个对象。类和对象属性按如下方式映射：

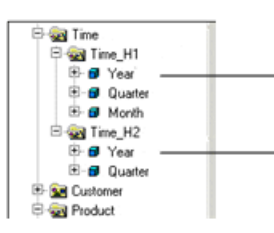
Universe 项目	属性	映射自 OLAP 项目...
类 另请参阅本章末尾介绍的特殊情况。	名称	维名称
	对象	级别和特性。
维	名称	由视图中的字段名称标识的当前级别名称。
	Select	视图字段，例如 MYVIEW.YEAR。
	表	用于确保使用适当联接的其他表： <ul style="list-style-type: none"> • 当前视图的表 • 视图 • 所有维的最高级别的表
明细（可选）	父维 注意：就上面的维而言，所有其他属性都相同。	依据与“ID”字段相关的“说明”字段创建的维对象。例如，“ID”字段为 YEAR，“说明”字段为 YEAR_DESC。“YEAR”对象是 YEAR_DESC 对象的明细。
度量	名称	多维数据集中的度量名称。
	选择（无贯穿钻取）	视图字段，例如 MYVIEW.SALES
	表	用于确保使用适当联接的其他表： <ul style="list-style-type: none"> • 视图 • 所有维的最高级别的表
	聚合函数	无。

多层次结构特殊情况：（多层次结构维支持）

为了支持分析型数据仓库的多层次结构维，将在 Universe 中执行以下操作：

- 按本节开头的说明为每个层次结构创建一组表。
- 为维创建一个类，并为每个层次结构创建一个子类。通过连接维度名称和层次结构名称对子类进行命名。
- 对于与层次结构内的级别对应的每个对象，向 Select 表达式中添加聚合函数。它将以注释的形式引用所有高级别的别名（当前层次结构的别名除外）。例如：

```
@Aggregate_Aware (glb_dnorm_fact_mktseg_view.YEAR/*Year_H1.DUMMY
Channel_All.dummy Customer_All.dummy Product_All.dummy Time_All_H1.dummy
glb_dnorm_fact_mkseg_view.dummy*/)
```



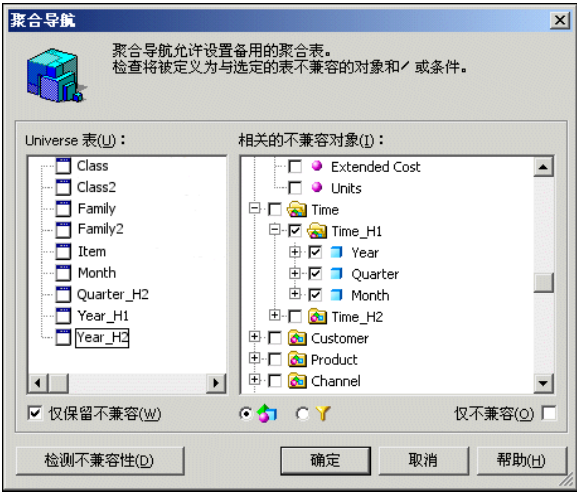
```
@Aggregate_Aware (glb_dnorm_fact_mktseg_view.YEAR/*Year_H2.DUMMY
Channel_All.dummy Customer_All.dummy Product_All.dummy Product_All2.dummy
glb_dnorm_fact_mkseg_view.dummy*/)
```

- 聚合导航被设置为使子类（层次结构）的对象与对应于另一个层次结构的表不兼容。这可防止最终用户使用代表着属于不同层次结构的级别的报表对象。

例如，表 Year_H1（来自层次结构 H1）与来自层次结构 H2 的对象不兼容：



而表 Year_H2（来自层次结构 H2）与来自层次结构 H1 的对象不兼容：



9.7.6 创建视图并生成 Universe

生成 Oracle OLAP Universe 的方法是：首先使用分析工作区多维数据集元数据来定义视图，然后设置 Universe 创建选项，生成新的 Universe。

9.7.7 用于从 Oracle 分析工作区创建 Universe 和视图的选项

可按如下方式创建 Universe 和视图：

Universe 创建选项	说明
创建视图并生成 Universe。	创建视图，然后选择映射哪些结构以创建 Universe。
仅从 Oracle 分析工作区创建视图。	如果不想创建 Universe，可以创建并保存视图。可以随时在视图列表中访问该视图并使用它创建 Universe。

Universe 创建选项	说明
从现有视图中生成 Universe。	选择一个现有的视图，然后直接从该视图生成 Universe。

相关主题

- 第 544 页上的[“创建视图并生成 Universe”](#)
- 第 546 页上的[“仅从 Oracle 分析工作区创建视图”](#)
- 第 547 页上的[“从现有的视图中生成 Universe”](#)

9.7.8 创建视图并生成 Universe

生成 Oracle OLAP Universe 的方法是：首先使用分析工作区多维数据集元数据来定义视图，然后设置 Universe 创建选项，生成新的 Universe。

创建视图并生成 Universe：

- 1 选择“文件”>“元数据交换”。

此时将出现“元数据桥”面板。

从“依据以下项创建 Universe”下拉列表中选择“Oracle OLAP”。

启动 Oracle OLAP Universe 构建器向导。

- 2 选择“创建视图并生成 Universe”然后单击“下一步”。
- 3 选择一个连接，输入用户名和密码，然后单击“完成”。

“分析工作区 (AW) 多维数据集”面板将会出现，显示可用于该连接的分析工作区多维数据集。

- 4 单击“多维数据集”节点。

将会显示可用在该连接中的分析工作区 (AW)。

- 5 展开一个 AW 节点以显示可用于该 AW 的多维数据集。
- 6 选择一个多维数据集，然后单击“下一步”。
- 7 一个状态框将显示从所选的多维数据集中加载元数据的进度。

将出现“视图创建”页面。它列出了该多维数据集可以使用的维和度量。

- 8 如果需要，修改数据类型和长度值。按如下方式执行此操作：

* 双击数据类型或长度值。

* 从下拉列表框中选择一个数据类型。

9. 单击“下一步”。

将出现“层次结构级别”页面。此页列出了包含数据类型和值的层次结构级别。

10. 如有必要，编辑层次结构值，然后单击“下一步”。

此时将出现“视图和 Universe 属性”页面。

11. 键入视图的名称，然后选择视图和 Universe 选项。视图属性和 Universe 选项如下所示：

查看视图的名称：可以编辑此字段。

为 OLAP_EXPRESSION 创建列：选择此选项后，会在视图中另外创建一个类型为 Raw(32) 的列，以允许在 Universe 中使用 OLAP_EXPRESSION 函数。

为标识符创建列：选择此选项后，将创建代表维成员（标识符）的列。

替换现有数据库对象：选择此选项后，将替换数据库中现有的类型和视图结构。

使用派生表：选择此选项后，不会从数据库中实际创建的视图中生成 Universe，而是通过派生表引用多维数据集结构来生成 Universe。派生表是一个只在引用了数据库结构的 Universe 中存在的虚拟表。它并不是在数据库中创建的。这在以下情况中很有用：用户没有“创建视图”权限，或者不希望增加数据库中的视图数量。有关使用派生表的信息，请参阅《Universe design tool User's Guide》（Universe 设计工具用户指南）。

将对象 ID 转换为明细：只有在为视图选择了“为标识符创建列”选项后才处于活动状态。选择此选项后，对象 ID 在生成的 Universe 中被转换为明细对象。

12. 单击“下一步”。

将出现“SQL 验证”页面。

13. 验证视图的 SQL，然后单击“下一步”。

将出现 Universe 信息摘要页面。

14. 验证 Universe 信息，然后单击“完成”。

Universe 设计工具将启动并打开至新生成的 Universe。

9.7.9 仅从 Oracle 分析工作区创建视图

可以使用分析工作区多维数据集元数据来创建视图。保存的视图将出现在视图列表中。创建了视图后，就可以稍后选择视图并生成 Universe。要只创建视图，请按照第 544 页上的[“创建视图并生成 Universe”](#)一节中相同的过程进行操作，但在启动“Oracle OLAP Universe 构建器向导”时，选择“只创建视图”单选按钮。

将会在目标数据库中创建视图。可以随时连接到此视图以创建 Universe。有关使用视图创建 Universe 的过程，请参阅第 547 页上的 [“从现有的视图中生成 Universe”](#) 一节。

9.7.10 从现有的视图中生成 Universe

可以从现有视图中生成 Universe。现有的视图出现在列表中。从列表中选择一個视图，然后生成 Universe。

9.7.11 从现有的视图中生成 Universe

- 1 选择“文件”>“元数据交换”。此时将出现“元数据交换”面板。从“依据以下产品创建 Universe”下拉列表中选择“Oracle OLAP”，然后单击“确定”。Oracle OLAP Universe 构建器向导将启动。
- 2 从 Oracle OLAP Universe 构建器向导的开始页中，选择“依据视图生成 Universe”单选按钮。单击“下一步”。此时将出现连接参数框。
- 3 选择一个连接并输入用户名和密码，然后单击“下一步”。“多维数据集”面板将出现，并显示可用于该连接的分析工作区多维数据集。
- 4 单击“多维数据集”节点。将会显示可用在该连接中的分析工作区（AW）。
- 5 展开一个 AW 节点以显示可用于该 AW 的多维数据集。选择一个多维数据集，然后单击“下一步”。此时将显示在多维数据集上定义的可用视图列表。
- 6 单击列表中的一个视图名称，然后单击“下一步”。一个状态框将显示从所选的多维数据集中加载元数据的进度。此时将出现“Universe 创建”页面。它列出了在可用于创建 Universe 的视图中定义的维、度量和层次结构级别。
- 7 如果需要，修改列名称或层次结构级别。请按以下所述进行修改：双击一个列名称或级别值。根据需要选择或键入名称。
- 8 单击“完成”。Universe 设计工具将启动并打开至新生成的 Universe。

第 10 章 部署 Universe

10.1 概述

本章与 Universe 的部署和管理有关。它描述了以下内容：

- 第 548 页上的 [“Universe 的部署方法”](#)
- 第 550 页上的 [“针对 Universe 设置访问限制”](#)
- 第 559 页上的 [“管理用户和登录”](#)

10.2 Universe 的部署方法

部署 Universe 是指将 Universe 提供给 Web Intelligence 用户或其他设计员使用。可以通过将 Universe 导出到中央管理服务器 (CMS) 资源库来部署 Universe。

可以在导出到生产资源库之前测试 Universe，方法如下：将 Universe 导出到测试资源库，并在 Web Intelligence 中运行测试。

仅当 Universe 完成设计、构建和测试阶段后，才能将 Universe 部署给 Web Intelligence 用户。

有关通过在资源库中导入和导出 Universe 来部署 Universe 的信息，请参阅以下各节：

- 第 44 页上的 [“导入 Universe”](#)
- 第 46 页上的 [“导出 Universe”](#)

10.2.1 在资源库中标识 Universe

Universe 由以下参数标识：

标识符	说明
文件名	最长 100 个字符和一个 .unv 扩展名。
长名称	最多包括 35 个字符。这是最终用户在 Web Intelligence 中用于识别 Universe 的名称。因此，该名称应说明 Universe 的用途。
唯一的系统标识符	CMS 分配的标识符。

10.2.1.1 Universe 标识符规则

以下规则适用于存储在资源库 Universe 文件夹中的 Universe 的 Universe 标识符。

- Universe 标识符在 CMS 中是唯一的。
- 文件名和文件夹位置的（路径）的组合。Universe 在父文件夹中是唯一的。

10.3 使所有用户都能访问 Universe

通过以工作组模式保存 Universe，可以使所有 Universe 设计工具用户可同时使用工作组模式和企业模式访问该 Universe。Universe 的连接不能是安全连接。如果希望某个 Universe 可供所有用户使用，则在保存该 Universe 时它必须具有不安全连接。

要使 Universe 可供所有 Universe 设计工具用户访问：

1. 验证希望可供所有用户使用的 Universe 并未具有安全连接。

- 2 必须具有安全连接才能将 Universe 导出到 CMS。如果 Universe 具有安全连接，请选择或创建新的共享连接。有关更多信息，请参阅第 85 页上的 [“定义和编辑连接”](#) 一节。
- 3 选择“文件” > “另存为”。
将出现“另存为”框。
- 4 选择“为所有用户保存”复选框。
- 5 单击“确定”。

10.4 针对 Universe 设置访问限制

可以将限制应用于使用 Universe 的已定义用户和组。

可在以下两个级别管理 Universe 安全性：

安全级别	说明
CMS	您可以通过中央管理控制台设置应用于存储在 CMS 中的 Universe 的限制。您可以设置 Universe 用户能访问的内容，并可以限制查看、编辑、删除和 Universe 中的其他操作，具体取决于为用户组定义的权限。本指南未介绍关于在 CMS 级别设置限制的信息。若要了解关于使用中央管理系统的信息，请参阅《BusinessObjects Enterprise 管理员指南》。
Universe	可以为获准使用 Universe 的用户定义限制。限制可以包含对象访问、行访问、查询和 SQL 生成控制以及连接控制。本指南描述了可针对 Universe 定义的限制的类型。

10.4.1 什么是限制？

限制是一个应用于 Universe 的限制的命名组。可以将限制应用于 Universe 的选定组或用户帐户。当用户连接到 Universe 时，他们在 Universe 中使用的对象、行、查询类型和连接取决于所应用的限制。

将限制分配给 BusinessObjects 用户或组。这根据用户组的配置文件限制了对 Universe 对象或资源的访问。

10.4.2 可以在 Universe 中应用什么限制？

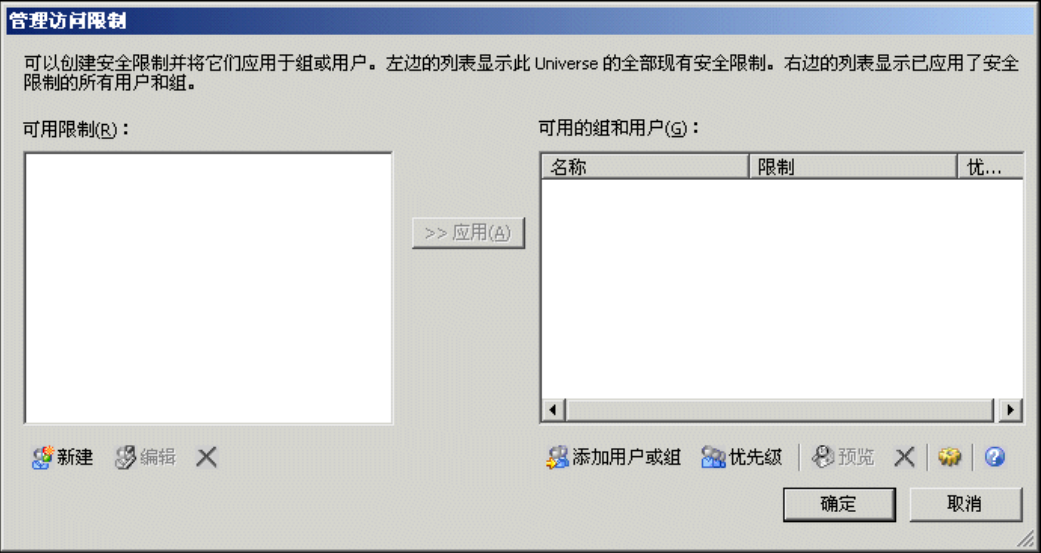
适用于用户组的访问限制是在限制中定义的。可以为 Universe 定义多个限制。可以随时编辑或删除限制。

限制可以定义以下类型的限制条件，以应用于 Universe 的选定用户或组：

限制类型	说明
连接	指向数据源的 Universe 连接。可以为 Universe 选择一个替代连接。第 85 页上的 “定义和编辑连接” 一节中描述了创建和编辑连接的过程。
查询控制	用于限制结果集大小和查询执行时间的选项。有关更多信息，请参阅第 100 页上的 “指明 SQL 限制” 一节。
SQL 生成选项	用于控制为查询生成的 SQL 的选项。有关更多信息，请参阅第 103 页上的 “设置 SQL 生成参数” 一节。
对象访问	可以选择不可用于 Universe 的对象。
行访问	可以定义限制行访问和查询所返回结果集的 WHERE 子句。
替代表访问	可以将 Universe 中引用的表替换为数据库中的另一个表。

10. 4. 3 如何管理访问限制？

访问限制是在“管理访问限制”对话框中管理的。通过选择“工具”>“管理安全性”>“管理访问限制”来访问此对话框。对话框如下所示。



“可用限制”窗格中列出了当前可用于 Universe 的限制。

为每个限制定义的用户和用户组出现在“可用的组和用户”窗格中。

下面描述了可用于管理访问限制的选项。

限制选项	说明
新建	定义新限制。
编辑	修改现有限制。
删除选定限制 ✕	从列表中删除限制。

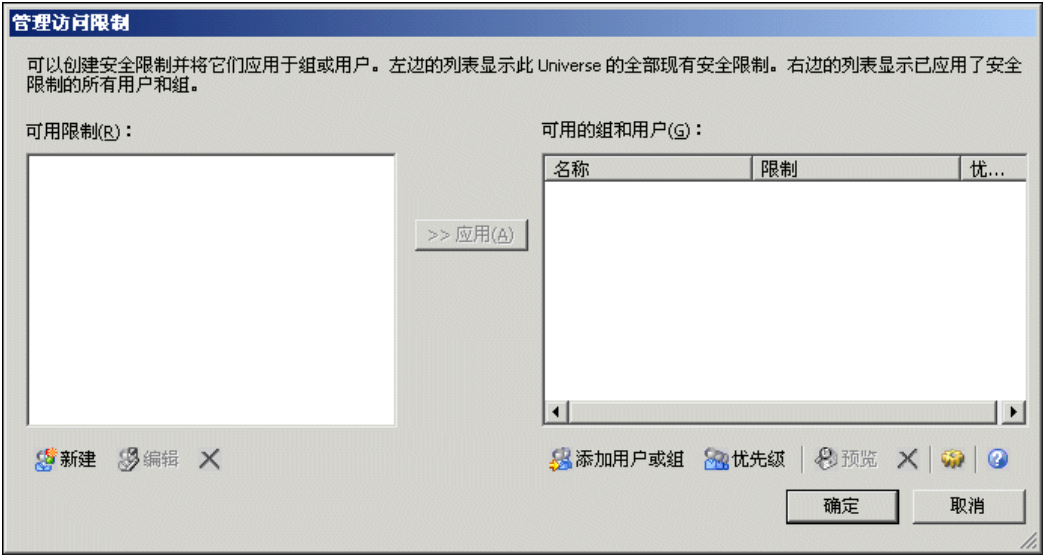
限制选项	说明
添加用户或组	通过为 BusinessObjects 系统定义的 BusinessObjects 用户和组的列表添加用户或组。
优先级	允许为一个或多个用户组设置优先级。
预览	允许查看为 BusinessObjects 系统定义的所有用户和组。
从选定的用户或组中删除安全选项 	删除对选定的用户或组设置的所有限制。
限制选项 	允许选择是用 AND 还是用 OR 运算符来执行行限制。

10. 4. 4 创建限制

将 Universe 导出到 CMS 后，可以随时创建、编辑和删除限制。
取决于目标用户组的查询需求，可以创建多项限制。


10. 4. 4. 1 创建限制

- 1 选择“工具”>“管理安全性”>“管理访问限制”。
将出现“管理访问限制”框。



- 2 单击“新建”。
- 将出现“编辑限制”对话框。可为 Universe 连接选择替换连接，或者通过使用“连接”对话框旁边的按钮编辑连接属性。
- 3 执行以下一项或多项操作：

要设置...	执行此操作...
新连接	从“连接”列表框中选择一个连接名称。
查询控制	<ul style="list-style-type: none">单击“控制”选项卡。选择查询选项，并为每个选项输入值。
SQL 生成选项	<ul style="list-style-type: none">单击“SQL”选项卡。根据情况选择“查询”、“多重路径”或“笛卡儿积”选项的复选框。
对象访问限制	<ul style="list-style-type: none">单击“对象”选项卡。单击“添加”。将出现“受限对象”框。单击“选择”。将出现“对象浏览器”。选择一个要限制的对象。单击“确定”关闭对象浏览器和“受限对象”框。

要设置...	执行此操作...
行访问限制	<ul style="list-style-type: none">单击“行”选项卡。单击“添加”。单击“表”框旁边的“浏览”按钮。单击一个表名称，并单击“确定”。单击“Where 子句”框旁边的“浏览”按钮。在定义框中键入 WHERE 子句。或者 通过从 SQL 编辑器中选择列、运算符和函数来构建 WHERE 子句。有关使用此编辑器的信息，请参阅第 169 页上的 “使用联接 SQL 编辑器” 一节。
指向替代表的引用	<ul style="list-style-type: none">单击“表映射”选项卡。单击“添加”。将出现“新表映射”框。将光标放在“原始表”框中并单击“选择”。将出现一个表浏览器。选择表并单击“确定”。将光标放在“替换表”框中并单击“选择”。在表浏览器中选择一个表，然后单击“确定”。 

- 4 单击“确定”。
- 新限制将出现在列表中。
- 5 单击“确定”。

10.4.5 应用 Universe 访问限制

通过将限制应用于一个或多个用户或用户组，从而在 Universe 上设置访问限制。

10.4.5.1 对 Universe 用户组应用限制

可以为 Universe 连接选择替换连接。

- 1 选择“工具”>“管理安全性”>“管理访问限制”。

将出现“管理访问限制”对话框。

- 2 单击“可用限制”窗格中的限制。
- 3 单击“可用的用户和组”窗格中的用户或组。

或

如果要选择多个用户或组，请按住 Ctrl 键的同时单击多个用户或组。

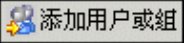
- 4 单击“应用”。
- 5 单击“确定”。

10.4.6 将用户组添加到 Universe 的可用用户列表

将限制应用于为 BusinessObjects 系统定义的用户组。这些用户是在 BusinessObjects 管理控制台的“管理组 and 用户帐户”部分中定义的。有关为 BusinessObjects 系统设置用户和组的信息，请参阅《BusinessObjects Enterprise XI 3.0 管理员指南》。

如果需要将限制应用于不在“可用的组和用户”窗格中的用户组，可按如下方式将该用户组添加到列表：

10.4.6.1 将用户组添加到“可用的组和用户”窗格中。

- 1 从“管理访问限制”框中（“工具”>“管理安全性”>“管理访问限制”），单击“添加用户或组”图标。

将出现“选择用户和组”对话框。它列出了在 BusinessObjects 管理控制台中定义的有权访问 BusinessObjects 系统的所有用户组。如果用户列表太大而无法轻松找到目标用户或组，则可按如下方式搜索列表：

- 从下拉列表中选择“名称”或“说明”。
- 在“搜索文本”框中键入文本字符串，以在上面选择的“名称”或“说明”字段中搜索文本字符串列表。
- 单击“搜索”图标开始搜索。

要过滤列表，也可以选择“组或用户”复选框以在列表中只显示组或用户。

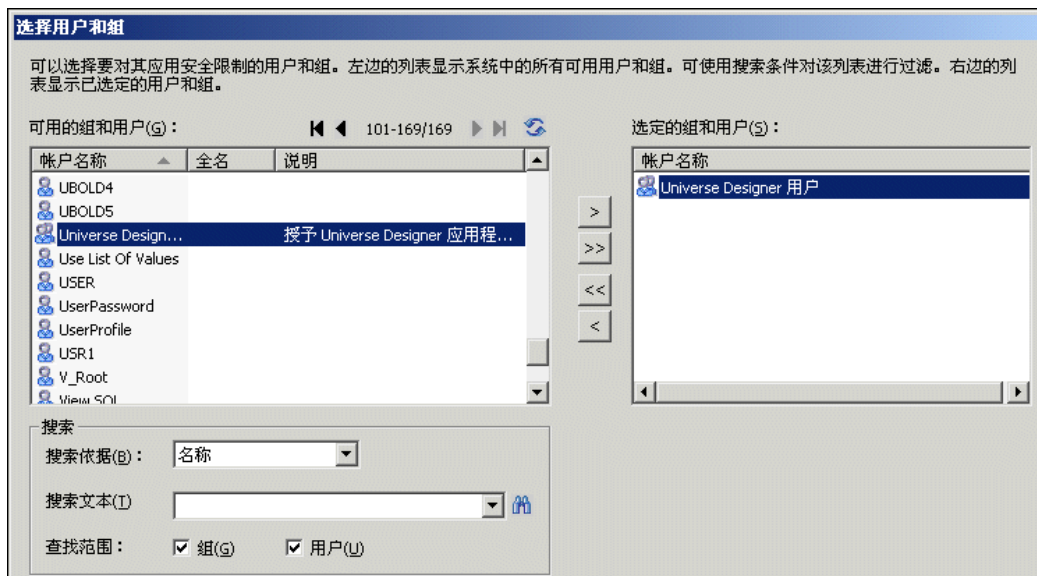
- 2 单击组或用户。

或

按住 Ctrl 键的同时单击多个用户或组。

- 3 单击向右箭头。

用户或组将出现在对话框右边的“选定的组 and 用户列表”窗格中。



- 4 单击“确定”。

该用户或组现在将出现在“管理访问限制”对话框中的“可用的组和用户”列表中。

10.4.7 设置限制组优先级

可以指定对属于多个组（这些组使用 Universe）的用户应用哪项限制。例如，某个用户属于两个组：限制为只能查看 5000 行数据的“销售”组，以及限制为只能查看 10000 行数据的“营销”组。当此用户刷新报表时，将应用与最低级别的组相关的限制。在上例中，如果“销售”组的次序为 1，而“营销”组的次序为 2，则会使用营销组的限制（10000）。

可以按次序排列用户组。将会使用在列出的次序中次序最低的组的限制。

注意：

这只适用于独有的限制，比如连接、表映射或 SQL 控制。如果在两个组上都设置了对象限制，则将应用所有这些限制。

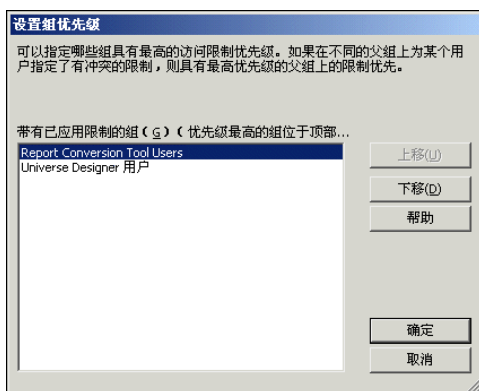
10.4.7.1 为使用多个限制的情况设置用户组优先级

- 1 选择“工具”>“管理安全性”>“管理访问限制”。

将出现“管理访问限制”对话框。

- 2 单击“可用的用户和组”窗格中的用户或组。
- 3 单击“优先级”图标。

将出现“设置组优先级”框。



- 4 选择用户或组，然后单击“上移”或“下移”按钮以更改优先级。
- 5 单击“确定”。

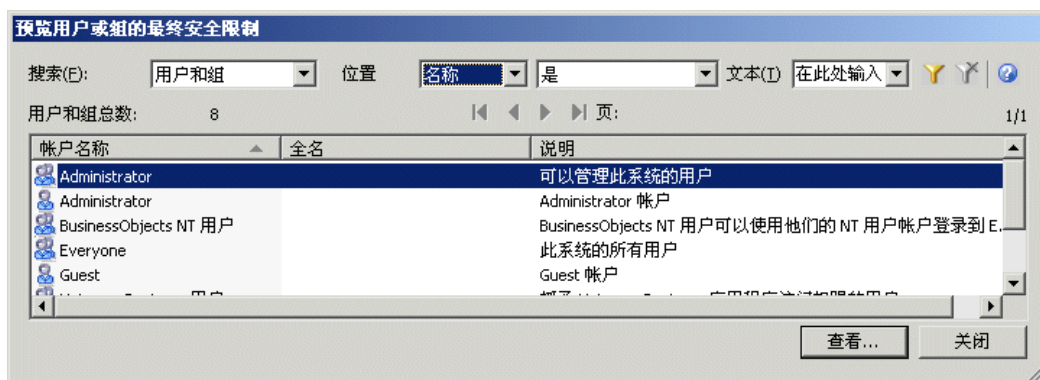
10.4.8 查看用户和组的安全限制

可以查看应用于所有用户和组的限制。

10.4.8.1 查看所有 Universe 用户和组的限制

1. 选择“工具”>“预览安全限制”。

将出现“预览用户和组”对话框。



2. 单击列表中的某个用户帐户名。
3. 单击“预览”。

将出现应用于该用户帐户的安全限制。显示为红色的参数和选项已经过修改，并专门应用于限制。

4. 单击“确定”关闭该框。

10.5 管理用户和登录

可以作为另一个用户登录 Universe 设计工具，也可更改自己的登录。用户帐户必须是对于目标资源库有效的帐户。

还可以在独立模式下登录 Universe 设计工具。可以使用 Universe 设计工具创建 Universe、个人和共享连接，但不能访问 CMS 中的连接和 Universe。

10.5.1 管理登录

也可以在不退出工作会话的情况下作为另一个用户登录 Universe 设计工具。用户帐户对于目标资源库必须有效。只有在您知道对应的用户名和密码时才能作为另一个用户登录。

10.5.1.1 作为另一个用户登录

1. 选择“工具”>“登录为”。

如果存在已打开的 Universe，Universe 设计工具将自动关闭这些 Universe。将出现“用户标识”对话框。

2. 在“用户名”框中键入有效的用户名。

- 3 键入新用户名和密码。
- 4 单击“确定”。

作为另一个用户登录时，会自动授予该用户的所有权限；但是，由于在用户配置文件上设置的限制的缘故，可能会禁止执行某些操作。

10.5.2 管理密码

在会话期间，可以更改在登录时所用的密码（只要该用户帐户有权限这样做）。但是，您不能更改用户名。

10.5.2.1 更改您的密码

- 1 选择“工具”>“更改密码”。
 - 将出现“更改密码”对话框。
 - 2 在“输入旧密码”框中键入现有的密码。
 - 3 在“输入新密码”框中键入新密码。
 - 4 通过在“确认新密码”框中再次键入新密码，对其进行确认。
 - 5 单击“确定”。
- 密码即被更改。

第 11 章 使用示例资料

11.1 使用示例资料 — 概览

本附录提供了有关 Club 数据库（使用 Microsoft Access 构建）的结构的信息。本指南中的大多数示例和图例都是从此数据库中衍生出的。

可以在 Business Objects 路径中的 \Samples\<language>\Databases 子文件夹下找到数据库文件 Club.mdb。在此文件夹中，您还将找到 efashion 演示数据库。

11.2 Club 数据库

本指南中提供的大多数示例内都使用了 Club 数据库。

11.2.1 表的结构

Island Resorts（一家专门提供成套度假服务的虚拟公司）的销售经理使用 Club 数据库。销售经理可以根据此数据库中的信息来进行销售和营销分析。此数据库由以下各表组成：

- Age_group
- City
- Country
- Customer

- Invoice_Line
- Region
- Region_Sline
- Reservation_Line
- Reservations
- Resort
- Sales
- Sales_Person
- Service
- Service_Line

以下各节描述上面的每个表以及它们的列。

11.2.1.1 Age_group 表

Age_group 表存储有关客户年龄段的信息。

列名称	说明
age_min	年龄段的下限
age_max	年龄段的上限
age_range	客户的年龄段

11.2.1.2 City 表

City 表存储有关客户居住的城市的信息。

列名称	说明
city_id	系统生成的城市编号
city	客户居住的城市（Albertville（阿尔贝维尔）、Amsterdam（阿姆斯特丹）、Augsburg（奥格斯堡）...Versailles（凡尔赛）、Washington D.C（华盛顿特区）、Yokohama（横滨））
region_id	系统生成的地区编号

11.2.1.3 Country 表

Country 表与客户居住的国家/地区相关。

列名称	说明
country_id	系统生成的国家/地区编号
country	客户居住的国家/地区的名称（Australia（澳大利亚）、France（法国）、Germany（德国）、Holland（荷兰）、Japan（日本）、UK（英国）、US（美国）。）

11.2.1.4 Customer 表

Customer 表包含与诸如姓名和地址等客户身份证明相关的信息。

列名称	说明
cust_id	系统生成的客户编号
first_name	客户的名字
last_name	客户的姓氏
age	客户的年龄
phone_number	客户的电话号码
address	客户完整地址中说明城市内具体地址的部分 (西方地址书写格式的第一行)
city_id	系统生成的城市编号
sales_id	系统生成的销售人员编号 (销售成套度假服务的人员)。
sponsor_id	系统生成的赞助商编号 (可选)

11.2.1.5 Invoice_Line 表

这个表包含发票信息；它用于向客户开出帐单。

列名称	说明
inv_id	系统生成的发票编号
service_id	系统生成的服务编号
days	代表在休假地停留时间长度的数字（3-15），以天数为单位。出于开帐单的目的，一次停留至多可以为 15 天。超过 15 天后，系统将剩余的天数视为一次新的停留。
nb_guests	为其开具发票的宾客的数量

11.2.1.6 Region 表

Region 表存储有关客户居住的地区的信息。

列名称	说明
region_id	系统生成的地区编号
region	客户居住的地区（Bavaria（巴伐利亚）、East Coast（美国东海岸）、East Germany（德国东部）...Wales（威尔士）、West（美国西部）、West Japan（日本西部））
country_id	系统生成的国家/地区编号

11.2.1.7 Region_Sline 表

这个表在 Universe 中启用销售收入聚合计算。本指南第 5 章中论述了聚合感知。

列名称	说明
sl_id	系统生成的服务系列编号（服务系列信息在 Service_Line 表中提供）
region_id	系统生成的地区编号
sales_revenue	按地区列出的总销售收入。

11.2.1.8 Reservation_Line 表

与客户预订相关的信息存储在 Reservation_Line 表中。

列名称	说明
res_id	系统生成的预订编号
service_id	系统生成的服务编号
res_days	在一周中的第几天预订（1 - 7）
future_guests	将来的宾客数（1 - 5）

11.2.1.9 Reservations 表

Reservation 表包含有关客户预订的日期的信息。

列名称	说明
res_id	系统生成的预订编号
cust_id	系统生成的客户编号
res_date	客户进行预订时的日期

11.2.1.10 Resort 表

Resort 表包含有关每个度假地的信息。

列名称	说明
resort_id	系统生成的度假地编号
resort	度假地的名称：Australian Reef（澳大利亚珊瑚礁）、Bahamas Beach（巴哈马海滩）、French Riviera（法国利维拉）、Hawaiian Club（夏威夷俱乐部）、Royal Caribbean（皇家加勒比海）
country_id	系统生成的国家/地区编号

11.2.1.11 Sales 表

Sales 表包含销售信息。

列名称	说明
inv_id	系统生成的发票编号
cust_id	系统生成的客户编号
invoice_date	发票的日期

11.2.1.12 Sales_Person 表

Sales_Person 表存储有关 Island Resorts 公司的销售人员的信息。

列名称	说明
sales_id	系统生成的销售人员编号
sales_person	销售人员姓名 (Andersen, Barrot, Bauman... Moore, Nagata, Schmidt)

11.2.1.13 Service 表

Service 表包含有关指定休假地中所提供服务的价格和类型的信息。

列名称	说明
service_id	系统生成的服务编号
service	休假地中提供的服务（请参阅下面的查询结果）
sl_id	系统生成的服务系列编号（服务系列信息在下一个表中提供）
price	服务的价格

11.2.1.14 Service_Line 表

Service_Line 表存储有关休假地的服务系列的信息。服务系列就是指服务所属的类别。

列名称	说明
sl_id	系统生成的服务系列编号
service_line	服务系列包括：膳宿、食品和饮料、娱乐
resort_id	系统生成的休假地编号（值从 1 到 5）

附录 A 更多信息

信息资源	位置
SAP BusinessObjects 产品信息	http://www.sap.com
SAP Help Portal	<p>导航到 http://help.sap.com/businessobjects/ 并在 “SAP BusinessObjects Overview” (SAP BusinessObjects 概述) 侧面板上单击 “All Products” (所有产品)。</p> <p>用户可以从 SAP Help Portal 访问涵盖所有 SAP BusinessObjects 产品及其开发信息的最新文档。用户可以下载 PDF 版本或可安装的 HTML 库。</p> <p>某些指南存储在 SAP Service Marketplace 中, 并且无法从 SAP Help Portal 获得。这些指南将在 Help Portal 上列出, 并附有指向 SAP Service Marketplace 的链接。拥有维护协议的客户有访问此站点的授权用户 ID。如要获得 ID, 请联系客户支持代表。</p>
SAP Service Marketplace	<p>http://service.sap.com/bosap-support > 文档</p> <ul style="list-style-type: none"> • 安装指南: https://service.sap.com/bosap-instguides • 发行说明: http://service.sap.com/releasenotes <p>SAP Service Marketplace 用于存储某些安装指南、升级和迁移指南、部署指南、发行说明以及所支持平台的文档。拥有维护协议的客户有访问此站点的授权用户 ID。请联系客户支持代表以获得 ID。如果是从 SAP Help Portal 重定向到 SAP Service Marketplace, 请使用左侧导航窗格中的菜单找到包含要访问的文档的类别。</p>
Docupedia	<p>https://cw.sdn.sap.com/cw/community/docupedia</p> <p>Docupedia 提供更多文档资源、协作编写环境, 以及交互式反馈渠道。</p>
开发人员资源	<p>https://boc.sdn.sap.com/</p> <p>https://www.sdn.sap.com/irj/sdn/businessobjects-sdklibrary</p>

信息资源	位置
SAP Community Network 上的 SAP BusinessObjects 文章	https://www.sdn.sap.com/irj/boc/businessobjects-articles 这些文章以前称为技术论文。
技术说明	https://service.sap.com/notes 这些技术说明以前称为知识库文章。
SAP Community Network 上的论坛	https://www.sdn.sap.com/irj/scn/forums
培训	http://www.sap.com/services/education 从传统的课堂学习到有针对性的电子学习讲座，我们都可以提供一套培训方案，满足您的学习需要，适合您喜欢的学习方式。
联机客户支持	http://service.sap.com/bosap-support SAP Support Portal 包含关于客户支持计划和服务的信息。它还包含指向各种技术信息和下载内容的链接。拥有维护协议的客户有访问此站点的授权用户 ID。如要获得 ID，请联系客户支持代表。
咨询	http://www.sap.com/services/bysubject/businessobjectsconsulting 从最初的分析阶段到交付部署项目为止，顾问将始终与您协同工作。我们提供各种主题（例如，关系数据库和多维数据库、连通性、数据库设计工具以及自定义嵌入技术）的专业技能。

索引

符號

- @ 函数 383
- @Aggregate_Aware 373, 385
- @Prompt 386
- @Select 400
- @Variable 402
- @Variable 区域设置 405
- @Where 408
- \$INSTALLDIR
 - 指南中使用的变量 13

A

- ANSI 92
 - Universe 参数 106
 - 创建完全外部联接 185
 - 定义联接 174
 - 对联接的支持 158, 172
- ANSI SQL92 语法 132
- AUTO_UPDATE_QUERY
 - Universe 参数 106
- 安全
 - 对象 292
 - 对象访问 272
 - 连接 88
- 安全性
 - CMS 级别 550
 - Universe 级别 550
 - 限制集 550

B

- BACK_QUOTE_SUPPORTED
 - Universe 参数 107
- Beach Universe 31
- BEGIN_SQL
 - Universe 参数 107
- BLOB_COMPARISON
 - Universe 参数 108
- BOUNDARY_WEIGHT_TABLE
 - Universe 参数 109
- 帮助
 - 为外部策略创建 413
- 保存
 - Universe 48
 - 默认 50
 - 为 PDF 49
 - 修改默认 50
- 必需过滤器
 - 变量 (SAP NetWeaver BW) 516

- 必需过滤器 (续)
 - 定义 315
 - 值列表 317
- 编号
 - Universe 修订 52
- 编辑
 - SQL 编辑器 297
 - 层次结构 326
 - 动态 SQL 参数 104
 - 对象 286
 - 类 282
 - 连接 85
 - 联接 167, 168, 169
 - 上下文 218
 - 使用公式栏 171
 - 值列表 339
- 编辑器
 - SQL 169
- 变量
 - 在 OLAP Universe 中 (NetWeaver BW) 514
- 标识
 - Universe 83, 549
 - 层次结构 325
 - 断层陷阱 254
 - 环路 234
 - 聚合级别 372
 - 扇形陷阱 259
- 标准化 246
- 表
 - 包含聚合表的环路 380
 - 策略 95
 - 插入 145, 146
 - 插入联接 165
 - 查看值 69
 - 查找 208
 - 抽取联接 32
 - 处理 62
 - 创建默认类和对象 32
 - 更改显示 65
 - 聚合 370
 - 浏览器, 请参阅表浏览器 145
 - 排列 149
 - 派生的 150
 - 删除 62
 - 事实定义 208
 - 添加 146
 - 填充模式 145
 - 图形显示 62
 - 推导出多个表 320
 - 显示行数 72

- 表 (续)
 - 修改返回的行数 442
 - 选择 62
 - 优化插入 148
 - 自动排列 65
 - 组织 61
 - 组织显示 149
 - “表”按钮 286
- 表的权重
 - 分配 441
- 表浏览器 52
 - 查看数据 147
- 激活 146
- 使用 145
- 优化 148
- 别名
 - Essbase 表 519
 - 不恰当使用 249
 - 创建 211, 236, 241
 - 定义 210
 - 多个 237
 - 检测 235, 236
 - 解决环路 229
 - 解决扇形陷阱 260
 - 名称 211, 212
 - 模式中的角色 210
 - 删除 214
- 并
 - 启用 100
 - 允许 32
- 不兼容的对象 375
- 不正确的结果
 - 断层陷阱 251
 - 环路 227
 - 扇形陷阱 257

C

- CASE_SENSITIVE 127
- CHECK_OWNER_STATE 127
- CHECK_QUALIFIER_STATE 127
- Club 数据库 31, 562
 - Age_group 表 563
 - City 表 563
 - Country 表 564
 - Customer 表 565
 - Invoice 表 565
 - Region 表 566
 - Region_Sline 表 567
 - Reservation_Line 表 567
 - Resort 表 568

- Club 数据库 (续)
 - Sales 表 569
 - Sales_Person 表 569
 - Service 表 569
 - Service_Line 表 570
 - 表的结构 562
 - COLUMNS_SORT
 - Universe 参数 110
 - COMBINE_WITHOUT_PARENTHESES
 - Universe 参数 110
 - COMBINED_WITH_SYNCHRO
 - Universe 参数 111
 - COMMA 128
 - COMPARE_CONTEXTS_WITH_JOINS
 - Universe 参数 111
 - CONCAT 129
 - CORE_ORDER_PRIORITY
 - Universe 参数 112
 - CORRECT_AGGREGATED_CONDITIONS_IF_DRILL
 - Universe 参数 113
 - CUMULATIVE_OBJECT_WHERE
 - Universe 参数 113
 - 参数
 - 设置资源控制 98
 - 参数文件
 - 定义 427
 - 操作
 - 撤消 57
 - 测试
 - Universe 367
 - 策略
 - 表 95
 - 对象 95
 - 联接 95
 - 输出格式 418
 - 外部参阅外部策略 410
 - 选择 94
 - 在“快速设计”向导中选择 425
 - 层次结构
 - 编辑器 326
 - 变量 (NetWeaver BW) 519
 - 标识 325
 - 创建 324, 326, 327
 - 更改对象的顺序 328
 - 设置 326, 327
 - 使默认层次结构可见 326
 - 在 OLAP Universe 中映射 (NetWeaver BW) 513
 - 值列表 341
 - 自定义 326
 - 钻取 325
 - 层次结构表
 - 在 Oracle 分析工作区 Universe 中使用 540
 - 插入
 - @ 函数 383
 - 插入 (续)
 - 表 145, 146
 - 优化 148
 - 查看
 - 从表浏览器查看数据 147
 - 连接 91
 - 数据库表 145
 - 行数 72
 - 值列表 338
 - 组安全限制 559
 - 查看语言 29
 - 查询
 - 并 100
 - 不兼容 222
 - 不明确 222
 - 复合条件 100
 - 合并条件对象 319
 - 环路 227
 - 设置控制 99, 100
 - 条件对象使用 319
 - 推导式 222
 - 限制执行时间 100
 - 相交 100
 - 允许子查询 100
 - 查询控制
 - 限制 438
 - 查找
 - Universe 中的快速搜索 61
 - 模式中的环路 234
 - 在 Universe 中搜索 58
 - 查找表
 - 定义 208
 - 共享 245
 - 灵活 246
 - 产品语言 29
 - 超级链接
 - 读取方式 288, 289
 - 对象创建 288, 289
 - 对象格式 288, 289
 - 撤消
 - 操作 57
 - 抽取
 - 表的联接 32
 - 窗口
 - 使用 54
 - 创建
 - Theta 联接 180
 - Universe 79, 80
 - Universe (依据元数据源) 524
 - 别名 211, 236
 - 层次结构 324, 326
 - 从 OLAP 源生成的 Universe 444
 - 动态 SQL 参数 104
 - 度量 303
 - 对象 283, 284
 - 复合联接 179
 - 创建 (续)
 - 级联值列表 329
 - 类 279, 281
 - 连接 85, 91
 - 连接 OLAP 451
 - 联接 160, 161, 162, 163
 - 链接 355
 - 明细 303
 - 默认类和对象 32
 - 上下文 216, 238
 - 条件对象 314
 - 外部策略 422
 - 维 302
 - 文件中的值列表 346
 - 限制集 553
 - 相等联接 177
 - 值列表 338
 - 值列表的层次 341
 - 子类 282
 - 自联接 188
 - 创建 Universe
 - 根据存储过程 363
 - 创建嵌套派生表 154
 - 存储过程
 - 动态值 365
 - 多个结果集 366
 - 类和对象 363
 - 存储过程中的动态值 365
 - 存储过程中的对象 363
 - 错误
 - 检查完整性 202, 268
- ## D
- DATE_WITHOUT_QUOTE 129
 - DECIMAL_COMMA
 - Universe 参数 114
 - DELIMIT_IDENTIFIERS 141
 - DELIMIT_LOWERCASE 129
 - DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION
 - Universe 参数 115
 - DISTINCT_VALUES
 - Universe 参数 115
 - 打开
 - Universe 46
 - 打印
 - Universe 75
 - “检查完整性”结果 203, 269
 - 设置选项 76
 - 页面设置 77
 - 预览 78
 - 单一登录 (SSO)
 - MSAS 2005 456
 - SAP NetWeaver BW 449
 - 导出
 - Universe 46

导出 (续)

- Universe 到 XML 元数据文件 528
- 锁定 Universe 52
- 先决条件 528
- 限制 528
- 值列表 343

导入

- Universe 44
- 锁定 Universe 52

登录

- 管理 560
- 密码 33
- 身份验证 33
- 系统 33
- 用户名 33
- 作为另一个用户 560

笛卡儿积

- 警告 101
- 阻止 101

定义 251

- .PRM 文件 427
- @ 函数 383
- OLAP Universe 中的条件对象 470
- Theta 联接 180
- Universe 参数 79
- Where 子句 310
- 动态 SQL 参数 103
- 度量 303
- 断层陷阱 251
- 对象 283
- 复合相等联接 179
- 环路 226
- 基数 191
- 聚合感知对象 373
- 快捷联接 187
- 类 279
- 连接 91
- 连接 OLAP 451
- 明细 303
- 扇形陷阱 257
- 上下文 214
- 条件对象 314
- 外部策略 412
- 维 302
- 值列表 334
- 自联接 188

定制

- 值列表 349

动态

- SQL 参数 104

独立

- 不与 CMS 一起使用 560

度量

- Group 子句 305
- 创建 303
- 定义 303

度量 (续)

- 动态特性 304
- 多个语句 101
- 计算 467
- 聚合函数 304
- 聚合投影 306, 466
- 投影函数“委派给数据库” 307, 463

断层陷阱 251

- 标识 254
- 检测 254
- 解决 251, 255
- 使用多个 SQL 256
- 使用上下文 255
- 直观检测 263

对象 15, 272, 274

- Select 语句 272
- Where 子句 272
- 安全 292
- 安全访问 272
- 编辑 286
- “表”按钮 286
- 不兼容 375
- 策略 95
- 层次结构 324
- 创建 283, 284
- 创建默认 32
- 定义 283
- 定义聚合感知 373
- 度量 275
- 高级选项 291
- 格式 300
- 更改层次顺序 328
- 关联值列表 289
- 键感知选项 292
- 角色概述 274
- “解析”按钮 286

类型 272, 275

- 名称 272
- 明细 275
- 拼接 323
- 日期格式 272
- 删除格式 301
- 生成 SQL 概述 19
- 属性 285
- 说明 272
- 推导出的 SQL 的概述 16
- 维 275
- 显示 17, 276
- 限定 16, 289
- 修改 286
- 移动 277
- 隐藏 278
- 用户访问 292
- 源信息选项 297
- 在结果中 272

对象 (续)

- 在条件中 272
- 指定限定 290
- 对象访问
 - 限制 438
- 多层次结构的表
 - 在 Oracle 分析工作区 Universe 中使用 540
- 多个 SQL
 - 断层陷阱 256
 - 扇形陷阱 263
 - 用于解决断层陷阱 256
- 多个别名 237
- 多维分析 324
 - 创建层次 327
 - 类型属于 324
- 多语言 Universe 27
- 多语种 Universe
 - SAP NetWeaver BW 448

E

eFashion

- Universe 31
- 数据库 562

END_SQL

- Universe 参数 116

Essbase 多维数据集

- Universe 中支持的功能 450
- 定义连接 451
- 将对象映射到 Universe 519

EVAL_WITHOUT_PARENTHESES

- Universe 参数 117

EXT_JOIN_INVERT 141

EXTERN_SORT_EXCLUDE_DISTINCT 130

F

FILTER_IN_FROM

- Universe 参数 117

FIRST_LOCAL_CLASS_PRIORITY

- Universe 参数 118

FORCE_SORTED_LOV

- Universe 参数 118

翻译管理工具 30

方法

- Universe 设计 23

访问

- 所有用户都能访问 Universe 549
- 外部策略 413

分发

- Universe 548

分配

- 表的权重 441
- 密码 89

- 分析
 - 联接 167
- 分析函数 425
 - IBM DB2 428
 - Oracle 428
 - RedBrick 432
 - Teradata 435
 - 优点 426
 - 在“函数”列表中可用 438
 - 支持的类型 426
- 附加 OLAP 参数 460
- 复合联接
 - 创建 179
- 复合条件
 - 启用 100
- 复制 277
- G**
 - Group 子句
 - 度量推导 305
 - GROUPBY_WITH_ALIAS 130
 - 高级
 - 对象选项 291
 - 格式
 - 对象 300
 - 删除 301
 - 显示数据类型 69
 - 个人
 - 连接 88
 - 更改
 - 表显示 65
 - 密码 561
 - 模式显示 67
 - 更新
 - OLAP Universe 476
 - 上下文 221
 - 工具栏
 - 使用 55
 - 移动 55
 - 工作组
 - Universe 设计 51
 - 公式栏
 - 编辑联接 171
 - 显示 171
 - 共享
 - 连接 88
 - 故障排除
 - 检查完整性 203, 269
 - 关闭
 - Universe 51
 - 关键日期变量
 - 在 OLAP Universe 中 518
 - 关键值
 - OLAP Universe 中的映射 513
 - 管理
 - 值列表 348
 - 过度膨胀的结果
 - 断层陷阱 251
 - 扇形陷阱 257
 - 过滤器
 - 查询中必需 315
 - 类和条件 276
- H**
 - 函数
 - 添加到 PRM 文件 438
 - 在“函数”列表中可用 438
 - 行
 - 设置查看的最大行数 98
 - 显示数目 72
 - 修改返回的数目 71
 - 修改行计数 72, 74
 - 行访问
 - 限制 438
 - 行计数
 - 查询优化 32
 - 适应数据容量 32
 - 显示 69
 - 合并查询
 - 允许 32
 - 环路
 - 标识 234
 - 定义 226
 - 检测 235, 239
 - 解决 226, 234
 - 使用别名解决 229
 - 使用上下文解决 231
 - 示例 243
 - 影响查询 227
- 基数 (续)
 - 检测 32, 196
 - 键 194
 - 解决数据库限制 200
 - 手动设置 194
 - 为联接设置 193
 - 显示 192
 - 优化 198
 - 优化检测 198
 - 在 Designer 中使用 191
- 基于派生表的派生表 153
- 激活
 - 表浏览器 146
 - 列表模式 63
- 级联值列表
 - 创建 329
- 计算所得度量
 - 在 OLAP Universe 中 467
- 减
 - 允许 32
- 剪切 277
- 剪贴板
 - 操作 277
- 检测
 - Universe 错误 202, 268
 - 别名 235, 236
 - 断层陷阱 254
 - 环路 235, 239
 - 基数 196
 - 联接 163
 - 联接路径问题 263
 - 联接中的基数 32
 - 扇形陷阱 259
 - 上下文 235, 238
 - 完整性错误 202, 268
 - 优化基数 198
- 检测上下文时的问题 221
- 检查
 - Universe 200, 201, 202, 266, 267, 268
- 检查完整性 367
 - 错误类型 202, 268
 - 打印结果 203, 269
 - 发送选项 78, 205
 - 启动 Designer 时 78, 205
 - 数据库中的更改 205, 271
 - 自动解析 78, 205
- 键
 - 感知 292
 - 基数 194
 - 显示 159
 - 主键 159
- 键感知
 - 设置外键感知 295
 - 设置主键感知 294
- I**
 - IBM DB2
 - 分析函数 428
 - IDENTIFIER_DELIMITER 131
 - IF_NULL 132
 - INNERJOIN_IN_WHERE
 - Universe 参数 119
- J**
 - Java Bean Universe 361, 362
 - Java Bean Universe 中的存储过程
 - 361, 362
 - JOIN_BY_SQL
 - Universe 参数 120
 - 基数 221
 - 定义 191

- 键选项卡
 - 键感知选项 292
 - 结构
 - STG 文件 416
 - “结构”窗格 52
 - 刷新 205, 271
 - 显示选项 68
 - 解决
 - 断层陷阱 251, 255
 - 环路 226, 234
 - 环路（使用别名） 229
 - 环路（使用上下文） 231
 - 联接路径问题 209
 - 扇形陷阱 257, 260
 - “解析”按钮 286
 - 界面
 - 构成项 54
 - 界面语言 27, 29
 - 聚合
 - 表 370
 - 为度量设置投影 306, 466
 - 聚合感知 370
 - 标识对象 372
 - 测试 Universe 382
 - 导航表 378
 - 导航不兼容的对象 378
 - 定义对象 373
 - 设置 371
 - 数据仓库 370
 - 指定不兼容的对象 375
- K**
- KEY_INFO_SUPPORTED 142
 - 可选提示
 - 在 OLAP Universe 中 475
 - 控制
 - Universe 550
 - 快捷联接
 - 创建 187
 - 定义 78
 - 在 Oracle 分析工作区 Universe 中使用 538
 - 快速设计
 - 取消激活向导 35
 - 显示选项 37
 - 向导 36
 - 扩展 PRM 131
 - CASE_SENSITIVE 127
 - CHECK_OWNER_STATE 127
 - CHECK_QUALIFIER_STATE 127
 - COMMA 128
 - CONCAT 129
 - DATE_WITHOUT_QUOTE 129
 - DELIMIT_IDENTIFIERS 141
 - DELIMIT_LOWERCASE 129
 - 扩展 PRM（续）
 - EXT_JOIN_INVERT 141
 - EXTERN_SORT_EXCLUDE_DISTINCT 130
 - GROUPBY_WITH_ALIAS 130
 - IF_NULL 132
 - KEY_INFO_SUPPORTED 142
 - ORDER_BY_STRINGS 142
 - OUTERJOINS_COMPLEX 132
 - OUTERJOINS_GENERATION 132
 - OVER_CLAUSE 135
 - OWNER 135
 - PREFIX_SYS_TABLE 136
 - QUALIFIER 136
 - QUOTE_OWNER 137
 - REFRESH_COLUMNS_TYPE 137
 - REMOVE_SEMICOLONS 138
 - REVERSE_TABLE_WEIGHT 138
 - UNICODE_PATTERN 139
 - USER_INPUT_DATE_FORMAT 139
 - USER_INPUT_NUMERIC_SEPARATOR 140
- L**
- LOV
 - 存储过程
 - 使用值列表 366
 - LOV, 请参阅“值列表” 289
 - 类 15, 275
 - 编辑 282
 - 创建 279, 281
 - 创建默认 32
 - 定义 279
 - 属性 281
 - 修改 282
 - 移动 277
 - 隐藏 278
 - 子类 282
 - 类和对象
 - 在 OLAP Universe 中修改 462
 - 在 Oracle 分析工作区 Universe 中 540
 - 在存储过程中 363
 - 类和对象映射
 - Essbase OLAP Universe 519
 - MSAS OLAP Universe 520
 - NetWeaver BW OLAP Universe 511
 - 类型
 - 对象 272
 - 连接 86
 - OLAP Universe 451
 - Universe 参数 83
 - XI R2, 使用 35
 - 安全 88
 - 参数概述 OLAP 451
 - 连接（续）
 - 查看可用 91
 - 创建新的 91
 - 定义 85
 - 个人 88
 - 共享 88
 - 密码 87, 89
 - 名称 87
 - 删除 92
 - 数据库引擎 87
 - 添加 93
 - 限制 438
 - 新 91
 - 修改 84, 85
 - 连接面板 86
 - 联接
 - ANSI 92 支持 158, 172
 - Theta 联接 176, 180
 - 编辑 167, 168, 169
 - 策略 95
 - 插入表之间 165
 - 创建 160, 161, 162
 - 定义 157
 - 分析 167
 - 检测 163
 - 检测基数 32
 - 检索链接的表 32
 - 快捷联接 176, 187
 - 删除 175
 - 设置基数 193
 - 使用 ANSI 92 语法定义 174
 - 使用公式栏编辑 171
 - 属性 165
 - 外部联接 176, 183
 - 外键 159
 - 相等联接 176
 - 修改 168
 - 运算符 167
 - 支持的类型 176
 - 主键 159
 - 自联接 176, 188
 - 联接路径
 - 别名定义 210
 - 不正确的结果 208
 - 查找表 208
 - 断层陷阱 209, 251
 - 环路 209
 - 检测问题 209, 263
 - 解决问题 209
 - 扇形陷阱 209
 - 事实表角色 208
 - 问题概述 207
 - 链接
 - Universe 102
 - 创建 355
 - 动态 355

链接的 Universe 350
 CORE_ORDER_PRIORITY 360
 动态链接 355
 对象顺序 360
 链接方法 352
 内核方法 352
 删除链接 358
 设置 355
 限制 354
 要求 354
 优点 354
 在一个中包含另一个 360
 主方法 353
 组合方法 353

列

查看值 69

列表模式

激活 63

灵活查找表 246

浏览器

表 52

M

MAX_INLIST_VALUES

 Universe 参数 120

MDX 函数 297, 299, 468

MSAS 多维数据集

 Universe 中支持的功能 449

 定义连接 451

 将对象映射到 Universe 520

密码

登录 33

更改 561

连接 87, 89

名称

Universe 83

别名 211, 212

对象 272

连接 87

明细

创建 303

定义 303

模式

别名使用 210

定义 143

基数的使用 191

检测联接路径问题 263

上下文使用 215

设计阶段 145

刷新 205, 271

显示 67

显示数据类型 69

显示行计数 69

用表填充 145

默认

保存选项 50

类和对象 32

修改保存选项 50

默认层次结构

使用自定义层次结构 326

默认值

提示语法 386

N

内核 Universe

更改 359

删除链接 358

内核方法

到链接的 Universe 352

O

OLAP

将 Oracle OLAP 映射到 Universe
537

OLAP Universe

MDX 函数 299, 468

创建 444

定义连接 451

修改 462

选项 461

映射 Essbase 对象 519

映射 MSAS 对象 520

映射 SAP NetWeaver BW 对象 511

预定义过滤器编辑器 473

支持的数据源 445

OLAP 附加参数 460

olap 函数 425

Teradata 435

OLAP 预定义过滤器编辑器 473

OLAP_UNIVERSE

 Universe 参数 121

Oracle

分析函数 428

Oracle 分析工作区

层次结构表 540

创建 Universe 536, 544

创建视图并生成 Universe 544

多层次结构的表 540

将 OLAP 映射到 Universe 537

类和对象映射 540

视图 540

维联接 540

依据视图生成 Universe 547

在 Universe 中使用快捷联接 538

ORDER_BY_STRINGS 142

OUTERJOINS_COMPLEX 132

OUTERJOINS_GENERATION 132

OVER_CLAUSE 135

OWNER 135

P

PATH_FINDER_OFF

 Universe 参数 121

PDF

另存为 49

PREFIX_SYS_TABLE 136

PRM 文件 427

 CASE_SENSITIVE 127

 CHECK_OWNER_STATE 127

 CHECK_QUALIFIER_STATE 127

 COMMA 128

 CONCAT 129

 DATE_WITHOUT_QUOTE 129

 DELIMIT_IDENTIFIERS 141

 DELIMIT_LOWERCASE 129

 EXT_JOIN_INVERT 141

 EXTERN_SORT_EXCLUDE_DISTINCT
130

 GROUPBY_WITH_ALIAS 130

 IDENTIFIER_DELIMITER 131

 IF_NULL 132

 KEY_INFO_SUPPORTED 142

 ORDER_BY_STRINGS 142

 OUTERJOINS_COMPLEX 132

 OUTERJOINS_GENERATION 132

 OVER_CLAUSE 135

 OWNER 135

 PREFIX_SYS_TABLE 136

 QUALIFIER 136

 QUOTE_OWNER 137

 REFRESH_COLUMNS_TYPE 137

 REMOVE_SEMICOLONS 138

 REVERSE_TABLE_WEIGHT 138

 UNICODE_PATTERN 139

 USER_INPUT_DATE_FORMAT 139

 USER_INPUT_NUMERIC_SEPARATOR
140

 添加函数 438

@Prompt 编辑器 387

排列

自动排列表 149

排列表 65

派生 Universe

创建链接 355

对象顺序 360

派生表

使用 150

派生表编辑器 154

配置选项

在 OLAP Universe 中 461

拼接对象 323

创建 323

语法 323

Q

QUALIFIER 136
 QUOTE_OWNER 137
 启动
 Universe 设计工具 33, 34
 迁移
 外部策略帮助文本 413
 外部策略到 XI 411
 嵌套派生表 153, 154
 创建 154
 清除
 值列表 272
 区域设置 29, 405
 区域设置定义 28
 取消激活
 设计向导 35

R

RedBrick
 risql 函数 432
 REFRESH_COLUMNS_TYPE 137
 REMOVE_SEMICOLONS 138
 REPLACE_COMMA_BY_CONCAT
 Universe 参数 122
 REVERSE_TABLE_WEIGHT 138
 risql 函数 425
 RedBrick 432
 日期
 数据库格式 272

S

SAP NetWeaver BW
 OLAP Universe 的数据源 446
 Universe 中支持的功能 447
 定义连接 451
 多语言 Universe 448
 映射到 OLAP Universe 511
 Select 语句 272
 SELFJOINS_IN_WHERE
 Universe 参数 122
 SHORTCUT_BEHAVIOR
 Universe 参数 123
 SMART_AGGREGATE
 Universe 参数 124
 SQL
 编辑器 297
 多个语句 101
 设置查询控制 100
 为外部策略创建文本文件 424
 SQL 编辑器
 编辑联接 169
 SQL 参数
 动态 103

SQL 生成选项
 限制 438
 STG
 文件参数 416
 STORED_PROC_UNIVERSE
 Universe 参数 124
 删除
 SQL 参数 104
 表 62
 别名 214
 对象格式 301
 连接 92
 联接 175
 上下文 220
 扇形陷阱
 标识 259
 定义 257
 过度膨胀的结果 257
 检测 259
 解决 257, 260
 使用别名和上下文 260
 使用多个 SQL 263
 直观检测 263
 上下文
 编辑 218
 不兼容查询 222
 不明确查询 222
 创建 216, 238, 241
 定义 214
 多个 SQL 语句 101
 更新 221
 检测 235, 238
 检测问题 221
 解决断层陷阱 255
 解决环路 231
 解决扇形陷阱 260
 模式中的角色 215
 删除 220
 修改 218
 上下文推导式查询 222
 设计
 模式 145
 设计向导
 取消激活 35
 设置
 保存默认值 50
 保存选项 50
 层次结构 327
 查看的最大行数 98
 动态 SQL 参数 103
 行计数 72
 基数 193, 194
 链接的 Universe 355
 模式显示选项 68
 摘要参数 93
 资源控制 99

身份验证
 Essbase 连接 457
 MSAS 连接 456
 SAP NetWeaver BW 连接 454
 登录 33
 生成
 层次结构 326
 动态 SQL 参数 104
 声明
 外部策略 415
 示例
 Universe 和数据库 31
 事实表
 定义 208
 视图
 “Universe”窗格中的条件 276
 对象 276
 条件 276
 在 Oracle 分析工作区 Universe
 中使用 540
 手动
 Universe 检查 202, 268
 对象创建 283
 设置基数 194
 输出
 外部策略的格式 418
 输入列 155
 鼠标
 操作 56
 属性
 Universe 79
 数据
 查看 147
 返回空集 319
 值列表文件 346
 钻取 325
 数据库
 查看表 145
 支持的模式 19
 值列表的委派搜索 335
 数据库引擎
 连接 87
 数据类型
 显示 69
 数组提取
 优化 440
 刷新
 OLAP Universe 476
 结构 205, 271
 值列表 272, 346
 说明
 Universe 83
 对象 272
 修改 84
 搜索
 在 Universe 中 58

索引感知

- 定义 292
- 设置外键索引 295
- 设置主键索引 294
- 使用 292
- 在 OLAP Universe 中 462

锁定

- Universe 52

T

Teradata

- olap 函数 435

Theta 联接

- 创建 180
- 定义 78

THOROUGH_PARSE

- Universe 参数 125

TRUST_CARDINALITIES

- Universe 参数 125

特性

- OLAP Universe 中的映射 512

提示

- 可选 (OLAP Universe) 475
- 默认值语法 386

提示编辑器 387

替代表访问

- 限制 438

替换

- 字符串或字符 58

替换语言 29

添加

- 表 146
- 连接 93
- 连接 OLAP 451

条件

- 对象, 请参阅“条件对象” 314
- 推导出多个表 320
- 显示 276
- 应用于值列表 339

条件对象

- 查询中必需 315
- 创建 314
- 定义 314
- 相互冲突的 Where 子句 315
- 移动 277
- 隐藏 278
- 在 OLAP Universe 中定义 470
- 在查询中使用 319

统计信息

- Universe 82

图形

- 标识环路 234
- 表 62
- 创建联接 161
- 检测联接路径问题 263

脱机

- 独立模式 560

U

UNICODE_PATTERN 139

UNICODE_STRINGS

- Universe 参数 126

Universe

- .unv 文件扩展名 48
- OLAP 444
- 保存 48
- 保存选项 50
- 编辑连接 85
- 标识 83, 549
- 标识符 549
- 测试 367
- 长名称 48, 549
- 窗格 52, 276
- 窗口概述 17
- 创建 79, 80
- 创建概述 18
- 创建连接 85
- 创建默认的类型和对象 32
- 打开 46
- 打印 75
- 导出 46
- 导出限制 528
- 导入 44
- 定义参数 79, 83
- 定义连接 85
- 动态链接 355
- 分发 548
- 概述 14
- 根据存储过程 363
- 更新 OLAP 476
- 工作组设计 51
- 关闭 51
- 检查完整性 200, 266
- 开发周期 23, 25
- 快速设计向导 36
- 利用概述 20
- 连接 83
- 链接 Universe 102
- 名称 83, 549
- 派生 Universe 中的对象顺序 360
- 设计员个人特征 21
- 设置资源控制 98
- 刷新 OLAP 476
- 说明 83, 93
- 所有用户都能访问 549
- 锁定 52
- 统计信息 82
- 文件名 549
- 显示条件 276
- 修订版号 52

Universe (续)

- 修改名称 84
- 在另一个中包含 360
- 摘要参数 93
- 摘要信息 82
- 注释 82, 93
- 资源控制 99
- 作用 14
- Universe 安全性
 - 设置控制 550
- Universe 检查完整性 367
- Universe 设计
 - 方法 23
 - 规划阶段 24
 - 开发周期 25
- Universe 设计工具
 - Universe 窗口 52
 - “Universe”窗格 52
 - “结构”窗格 52
 - 界面构成项 54
 - 启动 33, 34
 - 用户界面 52
- Universe 生命周期 (OLAP) 476
- Universe 文件
 - 使用信息设计工具处理 32
- USER_INPUT_DATE_FORMAT 139
- USER_INPUT_NUMERIC_SEPARATOR 140

W

Where 子句

- 不返回数据 319
- 冲突 319
- 定义 310
- 对象 272
- 问题 313
- 相互冲突 315
- 修改 310
- 外部策略 410
 - STG 文件参数 416
 - 表浏览器策略输出格式 421
 - 创建 422
 - 创建 SQL 文本文件 424
 - 创建帮助文本 413
 - 定义 412
 - 对象策略输出格式 419
 - 联接策略输出格式 421
 - 迁移帮助文本 413
 - 迁移到 XI 411
 - 设置查看的行数 98
 - 声明外部策略文件 415
 - 使用 410
 - 使用示例 416
 - 输出格式 418
 - 文件和过程概述 411
 - 选择 94

外部策略 (续)
 引用文本文件 422
 在 Designer 中访问 413
 在 Designer 中应用 424
 直接插入 SQL 422

外部联接

ANSI 92 185
 创建 183
 定义 78
 完全 185
 限制 186

外键 159

设置感知 295
 索引感知 292

完全外部联接

创建 185

完整性

检查 Universe 200, 266
 手动检查 202, 268
 自动检查 201, 267

维

创建 302
 定义 302

维联接

在 Oracle 分析工作区 Universe
 中使用 540

委派搜索

值列表的 335

文档语言 29

文件

创建值列表 346

文件夹

资源库文件系统 47

X

系统

登录 33

显示

表的行数 72
 对象 17, 276
 更改表 65
 公式栏 171
 行计数 69
 基数 192
 键 159
 列表模式 63
 模式 68
 模式选项 67
 数据类型 69
 组织表 61

限定

对象 289, 290

限制

SQL 生成选项 438
 Where 子句 310

限制 (续)

Where 子句问题 313
 查询控制 438
 查询执行时间 100
 对象访问 438
 多个表 320
 行访问 438
 连接 438
 使用准则 322
 使用自联接 320
 替代表访问 438

限制集

创建 553
 定义 551
 设置组优先级 558
 应用于用户组 556

相等联接

创建 177
 定义 78
 复合 179

相交

启用 100
 允许 32

向导

快速设计 36

信息设计工具

使用 .unv Universe 32

修订版号 52

修复

断层陷阱 255
 环路 226
 扇形陷阱 257

修改

Universe 定义参数 84
 Universe 名称 84
 Where 子句 310
 表显示 65
 对象 286
 对象格式 300
 返回的行数 71, 442
 行计数 72, 74
 类 282
 连接 84, 85
 联接 168, 169
 模式显示 67
 默认保存选项 50
 上下文 218
 数组提取 440
 说明 84
 值列表 339

选项

关联值列表 272
 使用之前自动刷新 272
 随 Universe 导出 272
 允许用户编辑此值列表 272

选择

表 62
 策略 94
 模式显示选项 67

Y

演示

Universe 31
 数据库 31

验证

Universe 200, 266

页面

指定设置 77

移动

对象 277
 工具栏 55
 类 277

隐藏

对象 278
 类 278
 条件对象 278

应用

外部策略 424

用参数表示的列 155

用户

登录 33, 560
 访问 Universe 549
 访问对象 292

用户界面语言 29

用户组

设置限制优先级 558
 应用限制集 556

优化

表浏览器 148
 值列表 349

语法

拼接对象 323
 在 SELECT 中的自动插入 438

语言 29

语言定义 28, 29

预定义过滤器编辑器 473

预览

Universe 78

元数据

Universe 选项 526
 连接选项 526

元数据 Universe

创建 524
 从 Oracle 分析工作区中创建
 536, 544
 将 Universe 导出为 XML 格式 528
 选择源 523

源信息

对象选项 297

源语言 29

运算符

 联接 167

Z

摘要

 Universe 信息 82

摘要参数 93

粘贴 277

正在同步 OLAP Universe 484

值

 表视图 69

 列视图 69

值列表 332

 编辑 272, 339

 查看 338

 创建 338

 创建层次 341

 导出 343

 定义 334

 定制 349

 个人数据文件 346

 关联对象 289

 管理 348

 和 SAP NetWeaver BW 变量 517

 级联 329

值列表 (续)

 清除 272

 属性 335

 刷新 272, 346

 显示 272

 修改 339

 选项 272, 289

 应用条件 339

 用在报表中 333

 优化 349

 指定属性 290

智能度量 307, 463

主方法

 到链接的 Universe 353

主键 159

 设置感知 294

 索引感知 292

注释

 Universe 82

资源

 为 Universe 设置 98

资源库

 部署 Universe 548

 导出 Universe 46

 文件夹结构 47

子类

 创建 282

字符

 查找或替换 58

字符串

 查找并替换 58

自动

 Universe 检查 201, 267

 表排列 65

 创建别名 241

 创建上下文 241

 对象创建 284

 环路检测 239

 基数检测 196

 类创建 281

 联接插入 165

自联接

 创建 188

 定义 78

 限制数据 320

组安全性

 查看限制 559

组合方法

 到链接的 Universe 353

组织

 表显示 61, 149

钻取 325