SAP BusinessObjects Business Intelligence 플랫폼
문서 버전: 4.1 Support Pack 6 – 2015-06-11

유니버스 디자인 도구 사용자 가이드

내용

1 유니버스 디자인 도구 소개 . . . 13
1.1 문서 기록 . . . 13
1.2 개요 . . . 13
1.3 유니버스 디자인 도구 및 유니버스 기본 사항 . . . 14
유니버스 개요 . . . 14
유니버스의 역할 . . . 14
유니버스의 구성 요소 . . . 14
유니버스 창 정보 . . . 16
유니버스 디자인 도구 설치 루트 경로 . . . 17
1.4 유니버스 디자인 도구를 사용하여 유니버스를 만드는 방법 . . . 17
개체에서 SQL 을 생성하는 방법 . . . 18
지원되는 데이터베이스 스키마 형식 . . . 18
유니버스 사용 방식 . . . 19
1.5 유니버스 디자이너 . . . 20
필요한 기술과 지식 . . . 20
유니버스 디자이너의 역할 . . . 20
1.6 기본적인 유니버스 만들기 단계 . . . 21
1.7 유니버스 개발 과정 소개 . . . 21
유니버스 디자인 방법 . . . 21
유니버스 개발 사이클 . . . 23
유니버스 계획 및 구현 시간 최적화 . . . 24
1.8 다국어 유니버스 . . . 24
1.9 언어 및 로캘 정의 . . . 25
1.10 다른 로캘 . . . 26
1.11 유니버스 디자인 도구 사용자 인터페이스에 대한 제품 언어 설정 . . . 27
1.12 다국어 유니버스 사용 . . . 27
1.13 연결된 유니버스의 대체 로캘 결정 . . . 27
1.14 번역 관리 도구 . . . 27
1.15 다국어 데이터 . . . 28
1.16 유니버스 디자인 도구 예제 자료 . . . 28
데모 데이터베이스 . . . 28
데모 유니버스 . . . 28
1.17 정보 디자인 도구에서 유니버스 사용 . . . 29

2 기본 작업 수행 . . . 30
2.1 개요 . . . 30

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.2 유니버스 디자인 도구 시작. 30
유니버스 디자인 도구 시작. 31
빠른 디자인 마법사 사용 31
2.3 Designer XI R3 에서 XI R2 연결 및 유니버스 사용. 32
2.4 빠른 디자인 마법사를 사용하여 기본 유니버스 만들기. 32
빠른 디자인 마법사를 사용하는 이유. 32
빠른 디자인 마법사 사용. 33
빠른 디자인 마법사로 만든 유니버스에 대한 추가 작업. 40
2.5 유니버스 가져오기. 40
리포지토리에서 유니버스 가져오기. 40
열기 및 가져오기의 차이. 41
2.6 유니버스 열기. 41
유니버스를 직접 열려면. 41
2.7 유니버스 내보내기. 41
리포지토리 파일 시스템에서의 유니버스 구성 방식. 42
리포지토리로 유니버스 내보내기. 42
내보내기 및 저장의 차이. 43
2.8 유니버스 저장. 43
식별자로서의 유니버스 파일 이름. 44
유니버스 저장. 44
유니버스 정의를 PDF 로 저장. 44
2.9 유니버스 닫기. 45
2.10 여러 디자이너와 작업. 46
유니버스 잠금. 46
수정 번호. 46
2.11 유니버스 디자인 도구 사용자 인터페이스 사용. 46
사용자 인터페이스의 주 구성 요소. 46
유니버스 디자인 도구 사용자 인터페이스. 47
창 조작 48
도구 모음 사용. 48
유니버스 디자인 도구에서 작업 수행. 49
2.12 찾기 및 바꾸기 사용. 51
찾기 사용. 51
빠른 찾기 사용. 53
2.13 테이블 디스플레이 구성. 54
테이블 표현 방식. 54
테이블 조작. 54
목록 모드 사용. 55
테이블 자동 정렬. 56
테이블 디스플레이 변경. 56

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.14 스키마 표시 옵션 선택. 58
구조 창 표시 그래픽 옵션 설정. 59
테이블 및 열 값 보기. 59
데이터베이스 테이블의 행 수 보기. 62
2.15 유니버스 인쇄. 65
인쇄 옵션 설정. 65

3 유니버스 만들기 및 유니버스 매개 변수 설정. 68
3.1 유니버스 매개 변수 개요. 68
3.2 새 유니버스 만들기. 69
처음부터 새로 유니버스 만들기. 69
3.3 요약 정보 보기 및 입력. 70
3.4 유니버스 매개 변수 설정. 71
유니버스 식별 71
연결 정의 및 편집. 72
유니버스 요약 매개 변수 설정. 80
전략 선택. 81
리소스 제어 지정. 84
사용 가능한 시스템 리소스 옵션. 85
리소스 제어 정보를 입력하려면. 85
여러 SQL 문을 생성하는 쿼리의 실행 시간 제한 86
SQL 제한 지정. 86
연결된 유니버스의 옵션 지정. 88
SQL 생성 매개 변수 설정. 88
SQL 생성 매개 변수 정보. 90
동적 SQL 생성 매개 변수 편집. 90
사용자 인터페이스에서 설정한 SQL 매개 변수. 91
PRM 파일에서 설정한 SQL 매개 변수. 106

4 테이블과 조인을 사용하여 스키마 만들기. 120
4.1 개요. 120
4.2 스키마 정의. 120
성공적인 유니버스의 기반이 되는 스키마 디자인. 121
스키마 디자인 및 유니버스 생성 프로세스. 121
스키마 디자인의 단계. 121
4.3 테이블 삽입. 121
테이블 탐색기 사용. 122
구조 창에서 테이블 정렬. 125
4.4 파생 테이블 사용. 125
파생 테이블 추가, 편집 및 삭제. 126
4.5 중첩된 파생 테이블. 128

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

파생 테이블 편집기 사용. 128
중첩된 파생 테이블 만들기. 129
중첩된 파생 테이블 이름 바꾸기. 129
4.6 입력 열이 있는 테이블 사용. 129
하드 코딩된 값 목록 정의. 130
사용자가 입력하거나 선택하는 값 목록을 정의하려면. 131
4.7 조인 정의. 131
조인 정의. 131
스키마에서 조인을 사용하는 이유. 132
조인 암시를 수행하는 SQL. 132
조인되지 않아야 하는 테이블. 132
기본 및 외래 키 조인. 133
조인의 카디널리티 이해. 134
조인 만들기. 134
조인 속성. 139
조인 편집. 141
유니버스에서 ANSI 92 조인 형식 지원. 144
조인 삭제. 147
4.8 특정한 유형의 조인 정의. 147
동일 조인 만들기. 148
테타 조인. 152
외부 조인. 155
바로 가기 조인. 159
자체 제한 조인. 160
4.9 카디널리티 사용. 162
유니버스 디자인 도구에서 카디널리티를 사용하는 방법. 163
수동으로 카디널리티 설정. 165
4.10 유니버스 검사. 172
자동으로 유니버스 무결성 검사. 172

5 스키마의 조인 문제 해결. 179
5.1 개요. 179
5.2 조인 경로 문제란?. 179
조회 테이블이란?. 179
팩트 테이블이란?. 179
잘못된 결과를 반환하는 조인 경로 유형. 180
조인 문제 검색 및 해결. 180
5.3 별칭 정의. 181
스키마에서 별칭을 사용하는 방법. 181
별칭 만들기. 182
5.4 컨텍스트 정의. 185

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

스키마에서 컨텍스트를 사용하는 방법 . . . 185
컨텍스트 만들기 . . . 186
컨텍스트 편집 . . . 189
컨텍스트 삭제 . . . 190
컨텍스트 업데이트 . . . 191
컨텍스트 검색을 수행하지 못하게 하는 조인 경로 . . . 191
컨텍스트가 쿼리에 미치는 영향 . . . 192
5.5 루프 해결 . . . 195
루프란? . . . 196
루프가 쿼리에 미치는 영향 . . . 197
시각적으로 루프 식별 . . . 204
자동으로 루프 식별 및 해결 . . . 204
루프를 검색하고 해결하는 도구 기능 . . . 204
루프 해결 예제 . . . 212
5.6 캐즘 트랩 해결 . . . 220
캐즘 트랩이란? . . . 221
캐즘 트랩이 결과를 확장하는 방법 . . . 222
캐즘 트랩 검색 . . . 224
캐즘 트랩 해결 . . . 224
5.7 팬 트랩 해결 . . . 226
팬 트랩이란? . . . 226
팬 트랩 해결 방법 . . . 229
팬 트랩 해결 방법 . . . 229
5.8 그래픽적으로 조인 문제 검색 . . . 233
잠재적인 캐즘 트랩 . . . 233
잠재적인 팬 트랩 . . . 234
5.9 유니버스 검사 . . . 236
자동으로 유니버스 무결성 검사 . . . 236
수동으로 유니버스 무결성 검사 . . . 237
유니버스 구조 새로 고치기 . . . 240

6 유니버스 만들기 . . . 242
6.1 개요 . . . 242
6.2 유니버스 작성 소개 . . . 242
개체란 . . . 242
유니버스에 사용되는 개체 유형 . . . 243
클래스 및 개체 사용 . . . 244
클래스란? . . . 244
6.3 유니버스 창 사용 . . . 244
클래스와 개체 또는 조건 표시 . . . 245
6.4 클래스, 개체 및 조건에 대한 기본 작업 . . . 246

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

잘라내기, 복사, 붙여넣기. 246
클래스, 개체 또는 조건 이동. 246
클래스, 개체 및 조건 표시 또는 숨기기. 246
6.5 클래스 정의. 247
클래스 만들기. 247
클래스 속성. 249
클래스 수정. 249
하위 클래스 사용. 249
6.6 개체 정의. 250
개체 만들기. 250
개체 속성. 252
개체 수정. 253
개체 정의. 253
속성. 256
고급. 257
인덱스 인식 정의. 258
소스 정보. 262
SQL 편집기를 사용하여 개체 정의. 263
개체 서식 정의. 265
개체 정의에 사용된 테이블 보기. 266
차원 정의. 267
설명 정의. 267
계수 정의. 268
개체에 대한 제한 정의. 273
조건 개체 정의. 277
자체 제한 조인을 사용하여 제한 적용. 283
여러 테이블을 유추하여 제한 적용. 283
연결 개체. 285
6.7 계층 정의. 287
다차원 분석이란?. 287
계층구조 식별 방법. 288
계층 설정. 288
6.8 계층에 대한 계단식 값 목록 사용. 292
계단식 값 목록 만들기. 292
6.9 값 목록 사용. 295
값 목록이 사용되는 방법. 295
개체에 값 목록을 사용하는 방법 정의. 296
값 목록 속성 및 옵션. 297
값 목록 편집. 301
값 목록 내보내기. 305

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

값 목록의 값 새로 고치기. 307
개인 데이터 파일의 데이터 사용. 307
유니버스의 값 목록 관리. 309
LOV 파일 최적화 및 사용자 지정. 310
6.10 유니버스 연결. 310
연결된 유니버스란?. 310
유니버스를 연결하는 여러 가지 방법. 312
유니버스 링크 이점. 313
유니버스 링크 요구 사항. 314
유니버스 링크 제한 사항. 314
두 유니버스 사이에 링크 만들기. 314
파생 유니버스 편집. 318
링크 제거. 318
코어 유니버스 위치 변경. 319
파생 유니버스 및 값 목록. 319
코어 유니버스의 순서로 개체 표시. 319
6.11 다른 유니버스 안에 유니버스 포함. 320
코어 유니버스를 파생 유니버스에 복사. 320
6.12 저장 프로시저 유니버스 만들기. 320
Java bean 유니버스의 저장 프로시저. 321
저장 프로시저를 기반으로 유니버스 만들기. 322
6.13 유니버스 테스트. 325
쿼리 패널에서 개체 테스트. 325
유니버스의 무결성 테스트. 326
Web Intelligence 로 유니버스 테스트. 326

7 유니버스 최적화. 327
7.1 개요. 327
7.2 집계 테이블 사용. 327
집계 인식 개요. 327
데이터 웨어하우스에 집계 인식 적용. 328
집계 인식 설정. 328
개체 작성. 329
집계 개체의 모든 조합 식별. 329
집계 순서에 따라 개체 정렬. 330
@Aggregate_Aware 함수를 사용하여 집계 개체 정의. 330
호환되지 않는 개체 지정. 332
호환되지 않는 개체 지정. 335
집계 테이블에 관련된 루프 해결. 337
집계 인식 테스트. 339
7.3 개체의 SQL 에서 @함수 사용. 339

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

개체에 @Function 삽입. 340
@Aggregate_Aware. 342
@Prompt. 343
@Script. 355
@Select. 356
@Variable. 357
@Where. 364
7.4 외부 전략을 사용하여 유니버스 만들기 사용자 지정. 365
유니버스 디자인 도구로 외부 전략 마이그레이션. 366
외부 전략 개요. 366
외부 전략이란?. 367
외부 전략에 대한 도움말 텍스트 만들기. 368
외부 전략 파일이 선언되었는지 확인. 369
외부 전략 예제 사용. 370
전략 파일(STG)의 구조. 370
전략 출력 형식. 372
외부 전략 만들기. 375
데이터 텍스트 파일 만들기. 376
유니버스 디자인 도구에서 외부 전략 적용. 376
7.5 분석 함수 사용. 377
분석 함수 소개. 378
분석 함수 사용 이점. 378
지원되는 분석 함수 계열. 378
유니버스 디자인 도구에서 분석 함수를 사용하는 방법. 379
IBM DB2 UDB 및 Oracle. 379
RedBrick(RISQL 함수). 383
Teradata(OLAP 함수). 385
Select 문에 자동으로 구문 삽입. 387
7.6 SQL 접두사 함수 사용. 388
BEGIN_SQL 유니버스 매개 변수를 사용하여 SQL 문에 접두사를 사용하려면. 389
7.7 배열 반입 매개 변수 최적화. 390
배열 반입 매개 변수 수정. 390
7.8 테이블 가중치 할당. 390
PRM 파일을 수정하여 테이블 가중치 할당 391
7.9 테이블에 대해 반환되는 행 수 수정 391
7.10 바로 가기 조인 사용. 391

8 OLAP 유니버스 작업. 392
8.1 OLAP 유니버스 정보. 392
OLAP 유니버스란?. 392
유니버스를 만드는 데 사용할 수 있는 OLAP 데이터 소스. 392

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.2 OLAP 데이터 소스에 대한 연결 정의. 398
OLAP 데이터 소스 연결 정보. 398
새 연결 마법사 시작. 399
OLAP 연결에 대한 데이터베이스 미들웨어를 선택하려면. 400
SAP BW OLAP 연결에 대한 로그인 매개변수. 401
MSAS OLAP 연결에 대한 로그인 매개 변수 402
Essbase 연결에 대한 로그인 매개 변수 정의. 402
OLAP 연결에 대한 소스 큐브 또는 쿼리를 선택하려면. 403
OLAP 연결에 대한 구성 매개 변수를 정의하려면. 404
Essbase 연결에 대한 사용자 지정 매개 변수 정의. 404
8.3 OLAP 유니버스 사용자 지정. 405
추가 매개 변수로 OLAP 유니버스 만들기. 405
OLAP 유니버스에 대한 OLAP 옵션 정의. 406
OLAP 유니버스의 개체 정의. 406
OLAP 유니버스에 대해 지원되는 유니버스 디자인 도구 기능. 407
데이터베이스 위임 프로젝션 함수. 408
OLAP 유니버스에 대해 위임된 계수 설정. 409
계수의 집계 프로젝션 설정. 410
OLAP 유니버스의 계산된 계수. 411
큐브 쿼리의 MDX 함수. 412
WHERE 절 및 필터에 대한 XML 구문. 413
OLAP 유니버스의 미리 정의된 조건. 414
OLAP 유니버스의 선택적 프롬프트. 418
SAP BW 유니버스에 대한 특정 쿼리의 성능 향상. 419
8.4 OLAP 유니버스 수명 주기 관리. 420
OLAP 유니버스 수명 주기 관리 정보. 420
개요: 유니버스 개체 상태 및 OLAP 개체 상태 사이의 관계 421
OLAP 유니버스 새로 고침. 423
OLAP 유니버스에 대해 수준 00 다시 생성. 425
수준 00 을 '모두'로 이름 바꾸기. 425
OLAP 유니버스 수준 접두사 바꾸기. 425
유니버스와 OLAP 큐브 동기화. 426
OLAP 유니버스 업데이트 시 차원 관리. 426
OLAP 유니버스 업데이트 시 계층구조 또는 특성 관리 431
OLAP 유니버스 업데이트 시 수준 관리 437
OLAP 유니버스 업데이트 시 SAP 변수 관리 440
OLAP 유니버스 업데이트 시 주요 수치 또는 계수 관리 443
OLAP 유니버스 업데이트 시 SAP 주요 날짜 관리 방식 447
8.5 다양한 OLAP 큐브가 유니버스에 매핑되는 방식. 449
유니버스에서 SAP BW 개체가 매핑 및 사용되는 방식. 449

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

Essbase 큐브가 유니버스 구성 요소에 매핑되는 방식. 457
MSAS 큐브가 유니버스 구성 요소에 매핑되는 방식 458

9 메타데이터 소스의 유니버스 작업. 459
9.1 메타데이터 소스에서 유니버스 생성 소개. 459
9.2 개요. 459
9.3 유니버스 작성 개요. 460
9.4 메타데이터 소스 선택. 460
9.5 메타데이터 소스 옵션을 선택하려면. 461
9.6 XML 소스에서 유니버스 만들기. 461
XML 메타데이터 소스. 462
XML 메타데이터 소스에서 유니버스를 생성하려면 462
연결 및 유니버스 옵션 선택. 463
XML 메타데이터 소스에서 유니버스를 업데이트하려면. 464
9.7 DB2CV 로 유니버스 내보내기. 464
내보내는 데 필요한 유니버스 사전 조건. 464
유니버스 메타데이터 확인. 465
DB2CV XML 파일로 유니버스 내보내기 467
유니버스와 DB2CV 메타데이터 간의 매핑. 467
특정 SQL 식 매핑. 471
9.8 Oracle Analytic Workspaces. 473
OLAP 큐브에서 유니버스를 생성하는 방법. 473
Oracle OLAP 구조를 유니버스 구성 요소에 매핑. 473
관계형 뷰 분석. 474
유니버스의 바로 가기 조인의 용도. 474
유니버스 구성 요소에 Oracle OLAP 구조를 매핑하는 방법. 476
뷰 만들기 및 유니버스 생성. 481
Oracle Analytic Workspace 에서 유니버스와 뷰를 만드는 옵션. 481
뷰 만들기 및 유니버스 생성. 481
Oracle Analytical Workspace 에 뷰만 만들기. 483
기존 뷰에서 유니버스 생성. 483

10 유니버스 배포. 484
10.1 개요. 484
10.2 유니버스 배포 방법. 484
리포지토리에서 유니버스 식별. 484
10.3 모든 사용자에게 유니버스에 대한 액세스 권한 부여. 485
10.4 유니버스에 액세스 제한 설정. 485
제한이란?. 486
유니버스에서 적용할 수 있는 제한. 486
액세스 제한을 관리하는 방법. 487

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

제한 만들기. 488
유니버스 액세스 제한 적용. 490
유니버스에 대해 사용 가능한 사용자 목록에 사용자 그룹 추가. 491
제한 그룹 우선 순위 설정. 492
사용자 및 그룹 보안 제한 보기. 493
10.5 사용자 및 로그인 관리. 494
로그인 관리. 495
암호 관리. 495

11 샘플 자료 사용. 497
11.1 개요. 497
11.2 Club 데이터베이스. 497
테이블 구조. 497

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1 유니버스 디자인 도구 소개

1.1 문서 기록

다음 표에 중요한 문서 변경 사항이 간략하게 나와 있습니다.

버전	날짜	설명
SAP BusinessObjects 유니버스 디자인 도구 4.0	2010 년 11 월 30 일	이 문서의 첫 번째 릴리스입니다. Universe Designer 가 '유니버스 디자인 도구'로 바뀌었습니다.
SAP BusinessObjects 유니버스 디자인 도구 4.0 서비스 팩 1	2011 년 2 월 25 일	
SAP BusinessObjects 유니버스 디자인 도구 4.0 서비스 팩 2	2011 년 6 월 15 일	연결 개체에 "로컬에 연결 다운로드"라는 관리자 정의 보안 권한이 추가되었습니다.
SAP BusinessObjects 유니버스 디자인 도구 4.0 기능 팩 3	2012 년 2 월 20 일	
SAP BusinessObjects 유니버스 디자인 도구 4.1 지원 팩 5	2014 년 10 월	• "외부 조인" 단원의 "절차"에서 네 번째 단계가 수정되고 "참고"가 추가되었습니다. • "카디널리티 표시" 단원에서 "연결선" 단어로 수정되었습니다.

1.2 개요

이 장에서는 유니버스 작성에 사용되는 도구인 유니버스 디자인 도구에 대한 일반적인 내용을 소개합니다. 이 장에서는 유니버스의 정의, 유니버스에 포함되는 내용, 유니버스를 만드는 방법, 비즈니스 환경에서 유니버스가 수행하는 역할에 대해 설명합니다.

또한, 일반적인 유니버스 개발 사이클을 최고의 디자인 방법과 함께 설명합니다. 이 릴리스와 함께 제공되는 데모 데이터베이스 및 유니버스에 대해서도 설명합니다.

이 장에서는 유니버스 디자인 도구, 개발 프로세스 및 유니버스에 사용 가능한 다양한 언어를 소개합니다. 이 장은 다음 항목으로 구성되어 있습니다.

관련 정보

유니버스 디자인 도구 및 유니버스 기본 사항 [페이지 14]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스 디자인 도구를 사용하여 유니버스를 만드는 방법 [페이지 17]
유니버스 디자이너 [페이지 20]
유니버스 개발 과정 소개 [페이지 21]
다국어 유니버스 [페이지 24]

1.3 유니버스 디자인 도구 및 유니버스 기본 사항

Business Objects 유니버스 디자인 도구는 Web Intelligence 및 Desktop Intelligence 사용자를 위해 유니버스를 만드는 데 활용할 수 있는 소프트웨어 도구입니다.

1.3.1 유니버스 개요

유니버스는 다음과 같은 항목을 포함하고 있는 파일입니다.

- 하나 이상의 데이터베이스 미들웨어에 대한 연결 매개 변수.
- 개체라고 부르는 SQL 구조. 개체는 데이터베이스에 있는 열, 테이블 및 데이터베이스 함수와 같은 실제 SQL 구조에 매핑됩니다. 개체는 클래스에 그룹화되어 있습니다. 개체와 클래스는 Web Intelligence 사용자가 모두 볼 수 있습니다.
- 데이터베이스에서 사용되는 조인 및 테이블의 스키마. 개체는 스키마에 포함된 데이터베이스 구조로부터 생성됩니다. 스키마는 유니버스 디자인 도구 사용자에게만 제공되며, Web Intelligence 및 Desktop Intelligence 사용자는 볼 수 없습니다.

Web Intelligence 사용자는 유니버스에 연결하여 데이터베이스에 대해 쿼리를 실행합니다. 이러한 사용자는 데이터베이스의 기본 데이터 구조를 보지 않고 기본 데이터 구조를 알지 못해도 유니버스의 개체를 사용하여 데이터를 분석하고 보고서를 만들 수 있습니다.

1.3.2 유니버스의 역할

유니버스의 역할은 전문적인 기술을 갖추지 않은 Web Intelligence 사용자가 데이터베이스에 대해 쿼리를 실행하여 보고서를 만들고 데이터 분석을 수행할 수 있도록 사용하기 간편하고 이해하기 쉬운 인터페이스를 제공하는 데 있습니다.

유니버스 디자이너는 유니버스 디자인 도구를 사용하여 사용자가 업무 요구 사항을 충족시키는 데 필요한 정보를 얻기 위해 액세스하고 쿼리해야 할 데이터베이스 구조(예: 열 및 데이터베이스 함수)를 나타내는 개체를 만듭니다.

유니버스에서 생성되는 개체는 최종 사용자의 업무 환경 및 용어와 연관성이 있어야 합니다. 이러한 개체의 역할은 데이터베이스의 SQL 구조에 대한 업무 중심의 프런트 엔드를 나타내는 데 있습니다.

1.3.3 유니버스의 구성 요소

유니버스는 다음과 같은 구조를 포함합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 클래스
- 개체

1.3.3.1 클래스

클래스는 유니버스 내에 포함된 개체의 논리적 그룹입니다. 이는 개체의 범주를 나타냅니다. 클래스의 이름은 여기에 포함된 개체의 범주를 나타내도록 지정해야 합니다. 클래스는 계층적으로 하위 클래스로 나뉠 수 있습니다.

1.3.3.2 개체

개체는 데이터베이스의 파생 데이터 또는 데이터에 매핑되는 명명된 구성 요소입니다. 개체의 이름은 대상으로 지정된 사용자 그룹의 업무 관련 용어에서 파생된 것이어야 합니다. 예를 들어, 제품 관리자가 사용할 유니버스에 사용되는 개체는 제품, 수명 주기 또는 릴리스 날짜 등일 수 있습니다. 재무 분석가가 사용할 유니버스에는 이윤 또는 투자 수익 같은 개체가 포함될 수 있습니다.

1.3.3.3 개체 유형

유니버스 디자인 도구에서 개체는 세 가지 유형(차원, 설명 또는 계수) 중 하나로 정규화됩니다.

표 1:

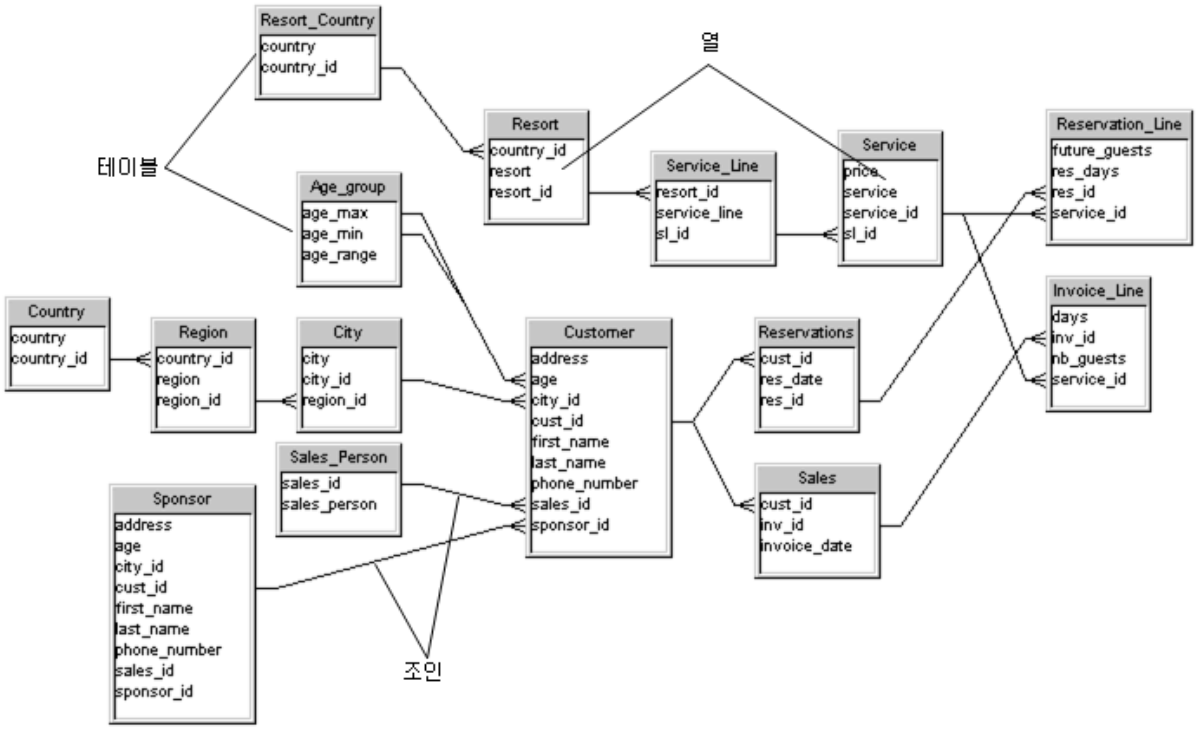
개체 유형	설명
차원	분석을 위한 매개 변수입니다. 차원은 일반적으로 지리, 제품 또는 시간 같은 계층에 연결됩니다. 성 및 City_Id 를 예로 들 수 있습니다.
설명	차원에 대한 설명을 제공하지만 분석에 무게를 둔 것은 아닙니다. 전화 번호를 예로 들 수 있습니다.
계수	차원 개체를 수량화하는 데 사용되는 숫자 정보를 전달합니다. 판매 수익을 예로 들 수 있습니다.

1.3.3.4 개체 - 스키마에 표시되는 SQL 구조 추정

Web Intelligence 사용자는 데이터베이스 스키마에 삽입한 SQL 구조를 유니버스에 표시되는 개체를 통해 추정할 수 있습니다. 유니버스 디자이너는 사용자의 분석 및 보고서 작성에 필요한 데이터를 반환하는 테이블 및 조인을 기반으로 이 스키마를 만듭니다.

스키마는 유니버스 파일의 일부이지만 유니버스 디자인 도구에만 표시되며 이 도구에서만 액세스할 수 있습니다. 스키마를 만들려면 유니버스 창의 구조 창을 사용합니다. 아래에는 예제 유니버스인 Beach.unv 의 스키마가 나와 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



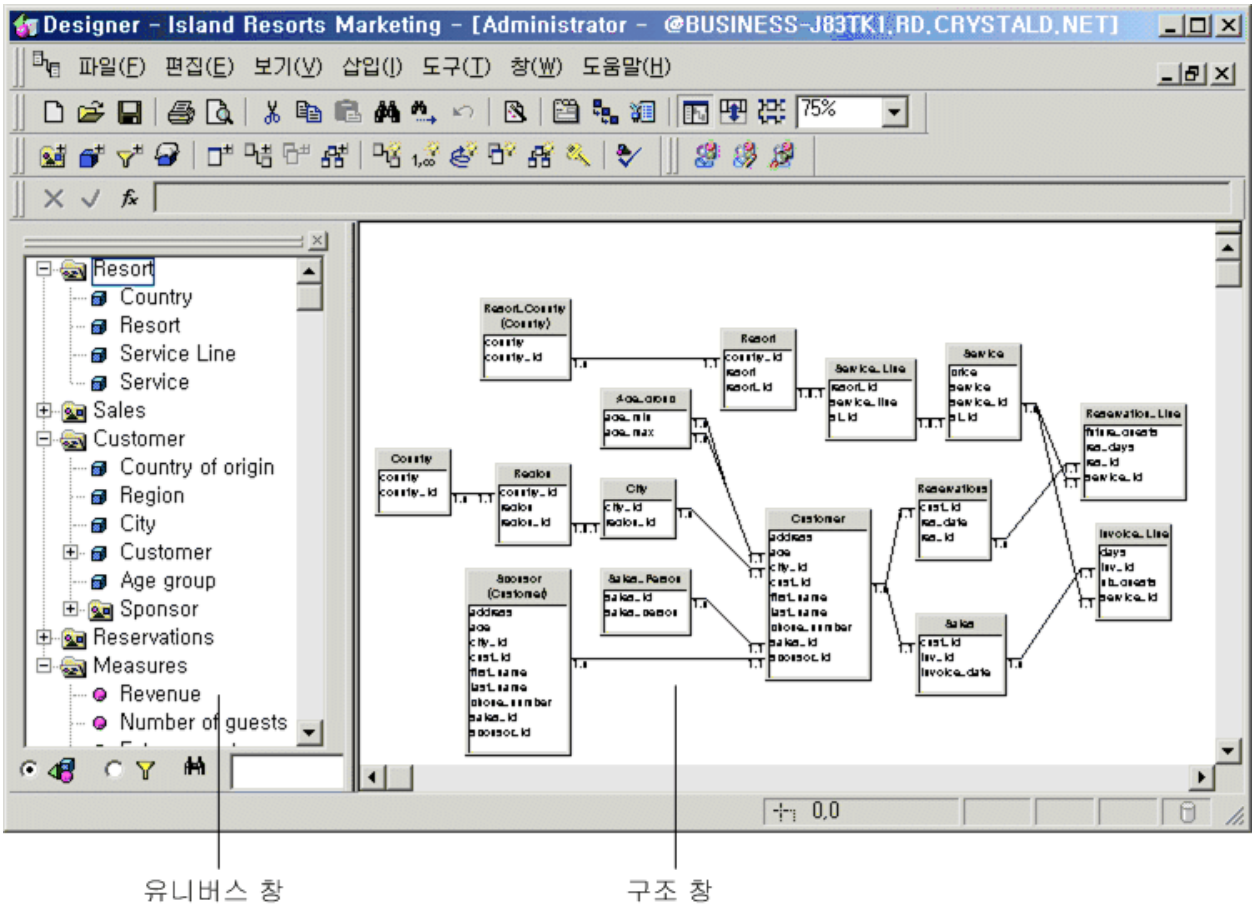
1.3.3.5 유니버스에서 개체가 표시되는 방식

개체는 유니버스 창에서 트리 탐색기 뷰의 노드로 표시됩니다. 개체 탐색기를 사용하면 클래스와 개체를 작성, 삭제, 복사, 표시 및 이동할 수 있습니다.

1.3.4 유니버스 창 정보

다음 그림에 유니버스 디자인 도구의 유니버스 창이 나와 있습니다. 여기에는 유니버스 창(Web Intelligence 에도 표시됨)과 구조 창(유니버스 디자인 도구에만 표시됨)이 모두 포함되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



1.3.5 유니버스 디자인 도구 설치 루트 경로

이 가이드에서 $INSTALLDIR 변수는 유니버스 디자인 도구와 Web Intelligence 에 사용되는 데이터 액세스 파일의 설치 루트 경로입니다. 이 경로는 유니버스 디자인 도구 실행 파일과 데이터 액세스 드라이버가 포함된 운영 체제 하위 디렉터리에 따른 Business Objects 설치 경로입니다.

Windows 의 경우 $INSTALLDIR = `\\...\Business Objects\BusinessObjects Enterprise 12.0\win32_x86` 입니다.

예: `C:\Program Files\Business Objects\BusinessObjects Enterprise 12.0\win32_x86`

1.4 유니버스 디자인 도구를 사용하여 유니버스를 만드는 방법

유니버스 디자인 도구에서는 데이터베이스 미들웨어에 연결하는 데 사용할 수 있는 연결 마법사를 제공합니다. 이 도구를 사용하면 여러 개의 연결을 만들 수 있지만, 각 유니버스에 대해 연결을 하나씩만 정의할 수 있습니다. 이 데이터베이스 연결은 유니버스와 함께 저장됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

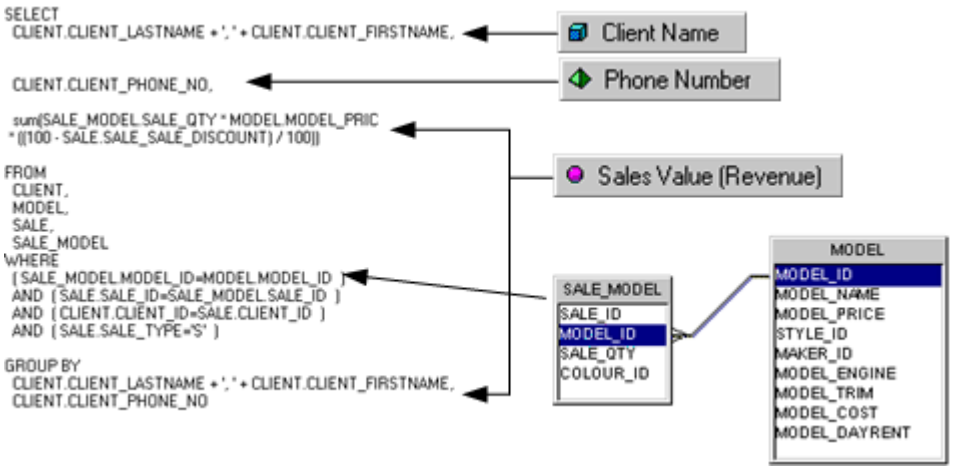
유니버스 디자인 도구에서는 데이터베이스의 테이블을 선택하고 확인하는 데 사용할 수 있는 그래픽 인터페이스를 제공합니다. 데이터베이스 테이블은 스키마 다이어그램에서 테이블 기호로 표시됩니다. 이 인터페이스를 사용하면 테이블을 조작하고, 테이블을 연결하는 조인을 만들며, 별칭 테이블과 컨텍스트를 만들고, 스키마의 루프를 해결할 수 있습니다. Web Intelligence 사용자에게는 이 스키마가 표시되지 않습니다.

유니버스 디자인 도구에서는 개체 탐색기 뷰인 트리 탐색기를 제공합니다. 탐색기 트리를 사용하면 스키마 뷰에 표시되는 SQL 구조와 열에 매핑되는 개체를 만들 수 있습니다. Web Intelligence 사용자는 이러한 개체를 조작하여 데이터베이스에 대해 쿼리를 실행할 수 있습니다.

유니버스 디자인 도구를 사용하면 중앙 관리 시스템(CMS) 리포지토리로 유니버스를 가져오거나 내보내는 방식으로 유니버스를 배포할 수 있습니다.

1.4.1 개체에서 SQL 을 생성하는 방법

Web Intelligence 사용자는 쿼리 작업 영역으로 개체를 끌어 놓아 쿼리를 만듭니다. 각 개체의 정의에서는 SELECT 문을 추정합니다. 쿼리를 실행하면 대상 데이터베이스에 대해 모든 개체와 관련하여 SELECT 문과 선택적으로 WHERE 절이 실행됩니다.



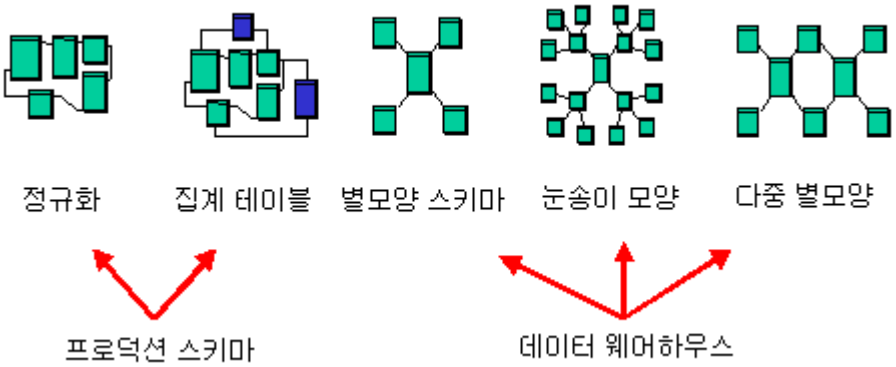
사용자가 쿼리 작업 영역에 계수 개체와 함께 차원 또는 설명 개체가 포함되도록 선택하면 그러한 차원 및 설명 개체의 내용이 포함된 GROUP BY 절이 SELECT 문에 자동으로 추가됩니다.

WHERE 절의 조인 및 FROM 절에 포함되는 테이블은 구조 창에서 작성한 테이블 스키마로부터 추정됩니다.

1.4.2 지원되는 데이터베이스 스키마 형식

유니버스 디자인 도구에서는 아래에 나와 있는 모든 형식을 포함하여 대부분의 데이터베이스 스키마 형식을 지원할 수 있습니다. 유니버스 디자인 도구를 사용하기 전에 데이터베이스를 다시 정의하거나 최적화할 필요가 없습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



1.4.3 유니버스 사용 방식

유니버스는 Web Intelligence 사용자가 사용합니다. 유니버스는 중앙 관리 시스템(CMS) 리포지토리에 저장됩니다. 최종 사용자는 웹 브라우저를 통해 유니버스에 연결됩니다.

데이터베이스에 대한 연결이 유니버스에 정의되어 있으므로 유니버스에 연결된 최종 사용자는 데이터에 대한 액세스 권한을 자동으로 받습니다. 한편, 데이터에 대한 액세스는 유니버스에서 사용 가능한 개체로만 제한됩니다. 이러한 개체는 정의된 사용자 그룹의 사용자 요구 사항 프로필을 기반으로 유니버스 디자이너가 만든 것입니다.

1.4.3.1 대상으로 지정된 데이터 요구 사항 표현

유니버스는 특정 응용 프로그램, 시스템 또는 사용자 그룹의 데이터 요구 사항을 나타낼 수 있습니다. 예를 들어, 유니버스에는 한 기업의 마케팅 또는 회계 부서의 데이터 요구 사항을 나타내는 개체가 포함될 수 있습니다.

유니버스는 급여 지급 또는 재고 시스템 같은 조직화된 절차 또는 부서 내 하위 팀의 데이터 요구 사항을 나타낼 수도 있습니다.

클래스의 예로는 직원 정보, 참석자 정보 및 부서 정보를 들 수 있습니다.

1.4.3.2 유니버스 및 데이터베이스 스키마

데이터베이스 스키마는 인사(PERSONNEL), 재고(INVENTORY) 및 판매(SALES)라는 세 가지 유니버스를 만드는 데 사용합니다. 각 유니버스에는 클래스와 개체가 포함되어 있습니다. 각 개체는 데이터베이스 구조의 한 부분에 매핑됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1.4.3.3 유니버스의 사용자

Web Intelligence 사용자는 보고서를 작성하거나 분석하는 데 유니버스를 사용합니다. 유니버스는 이러한 사용자의 업무 영역에 관련된 클래스와 개체를 제공해야 합니다.

1.5 유니버스 디자이너

유니버스는 유니버스 디자이너가 유니버스 디자인 도구를 사용하여 만듭니다. 유니버스 디자이너에 대한 표준 프로필은 없습니다. 기업 내에서 데이터베이스 관리자, 응용 프로그램 관리자 또는 개발자, 프로젝트 관리자 또는 다른 사용자를 위해 유니버스를 만드는 데 필요한 기술을 충분히 습득한 보고서 작성자를 유니버스 디자이너로 임명할 수 있습니다.

회사에 유니버스 디자이너가 여러 명 있을 수도 있습니다. 유니버스 디자이너의 수는 기업의 데이터 요구 사항에 따라 달라집니다. 예를 들어, 응용 프로그램, 프로젝트, 부서 또는 기능 영역별로 유니버스 디자이너를 한 명씩 지정할 수 있습니다.

여러 사람이 유니버스를 만드는 경우 일련의 규칙이나 용어 지침을 정의하여 개체를 일관성 있게 나타내는 것이 중요합니다.

1.5.1 필요한 기술과 지식

유니버스 디자이너는 다음과 같은 기술 및 지식을 일정 수준 이상 습득해야 합니다.

표 2:

기술/지식	설명
사용자 요구 사항 분석 능력	유니버스는 사용자의 데이터 요구 사항에 맞도록 만들어집니다. 유니버스 디자이너에게는 사용자 요구 사항을 분석하여 사용자의 용어에 관련된 클래스와 개체를 만들고 사용자 조직의 요구 사항을 충족시키는 유니버스를 개발하는 데 필요한 기술이 있어야 합니다. 이러한 요구 사항에는 분석에 적합한 보고서 작성 및 쿼리 결과가 포함됩니다.
데이터베이스 지식	유니버스 디자이너는 기업의 DBMS(데이터베이스 관리 시스템), 데이터베이스 배포 방식, 논리적 데이터베이스 구조 및 기업 데이터베이스에 저장된 데이터 형식을 잘 알고 있어야 합니다.
SQL(Structured Query Language:구조적 쿼리 언어)	SQL 에 대한 실무 지식이 필요합니다.

1.5.2 유니버스 디자이너의 역할

유니버스 디자이너는 일반적으로 다음과 같은 역할을 담당합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 사용자 요구 사항 분석
- 유니버스 디자인 및 작성
- 유니버스 배포
- 유니버스 관리

1.6 기본적인 유니버스 만들기 단계

유니버스는 보고 도구로 보고서를 만드는 최종 사용자가 사용하는 .unv 파일의 일부이며, .unv 파일 중 최종 사용자에게 표시되는 유일한 부분입니다. 유니버스를 만들 때는 다음 단계를 수행하십시오.

1. 빈 유니버스 파일을 새로 만들고 연결 및 매개 변수 세부 정보를 설정합니다.
2. 사용할 데이터베이스 테이블의 스키마를 만들고 테이블 간의 조인을 정의합니다. 이 스키마는 보고 도구를 사용하여 보고서를 만드는 최종 사용자에게 표시되지 않습니다.
3. 스키마의 조인 문제를 해결합니다.
4. 유니버스를 구성할 클래스와 개체를 만듭니다. 이 유니버스는 보고 도구로 보고서를 만드는 최종 사용자에게 제공됩니다.

관련 정보

기본 작업 수행 [페이지 30]
유니버스 만들기 및 유니버스 매개 변수 설정 [페이지 68]
테이블과 조인을 사용하여 스키마 만들기 [페이지 120]
스키마의 조인 문제 해결 [페이지 179]
유니버스 만들기 [페이지 242]
유니버스 최적화 [페이지 327]

1.7 유니버스 개발 과정 소개

다음 단원에서는 유니버스를 직접 만드는 방법에 대한 개요를 제공하고 일반적인 유니버스 개발 사이클에 맞도록 유니버스를 만드는 방법을 설명합니다.

1.7.1 유니버스 디자인 방법

이 설명서에 나와 있는 유니버스 디자인 방법은 계획 단계 하나와 구현 단계 세 개로 이루어져 있습니다.

- 비즈니스 문제 분석 및 유니버스 솔루션 계획
- 스키마 디자인

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 유니버스 작성
- 사용자에게 유니버스 배포

각 구현 단계는 초기 계획 단계를 완료했다는 가정 하에 진행됩니다. 계획 단계는 유니버스 디자인 도구를 사용하지 않고 수행할 수 있으며, 이는 유니버스의 성패를 판가름하는 중요한 단계입니다. 사용자의 보고서 작성 요구 사항을 면밀히 검토하지 않은 채 불완전하게 계획한 유니버스는 디자인, 구현 및 관리가 어렵고 정작 대상 사용자에게는 쓸모 없는 것이 될 수 있습니다.

아래에는 이러한 각 단계에 대한 설명이 나와 있습니다.

1.7.1.1 유니버스 디자인 도구 사용 전 유니버스 계획

첫 번째 단계를 시작하기 전에 유니버스 작성 프로젝트에 할당된 시간의 최대 80% 정도를 유니버스 계획에 할애해야 합니다. 특히 다음과 같은 사항을 염두에 두어야 합니다.

- 유니버스를 사용할 대상 사용자의 데이터 분석 및 보고서 작성 요구 사항을 분석해야 합니다. 스키마를 만드는 데 사용되는 구조는 해당 테이블과 열에 포함된 데이터에 대한 액세스를 필요로 하는 명확하게 정의된 사용자 요구 사항을 기반으로 해야 합니다.
- 유니버스 디자인 도구를 사용하기 전에 앞으로 만들어야 할 개체에 대한 분명한 개념이 서 있어야 합니다. 데이터베이스에서 사용할 수 있는 열을 조회하여 개체를 만드는 것이 아니라 사용자의 요구 사항 분석을 통해 이미 확인된 개체에 일치하는 열을 식별해야 합니다.

1.7.1.2 스키마 디자인

유니버스의 기본 데이터베이스 구조에 대한 스키마를 만듭니다. 이 스키마에는 대상 데이터베이스의 테이블 및 열과 이를 연결하는 데 사용되는 조인이 포함됩니다. 별칭이나 컨텍스트를 사용하는 과정에서 구조에 발생할 수 있는 루프, 캐즘 트랩 및 팬 트랩 같은 조인 문제를 해결해야 할 수도 있습니다. 전체 구조의 무결성을 테스트합니다. 스키마 디자인 단계에 대한 자세한 내용은 이 설명서의 테이블과 조인을 사용하여 스키마 만들기 [페이지 120] 및 스키마의 조인 문제 해결 [페이지 179] 장을 참조하십시오.

1.7.1.3 유니버스 작성

스키마의 구성 요소를 기반으로 SELECT 문을 추정하는 개체를 만듭니다. 이러한 개체를 클래스로 구성합니다. 이는 사용자 보고서 요구 사항에 대한 분석을 통해 확인된 개체입니다. 사용자의 보고서 작성 및 다차원 분석 기능을 향상시키고 쿼리 성능을 최적화하기 위한 여러 가지 형식의 개체를 만들 수 있습니다.

유니버스 구조의 무결성을 테스트합니다. 또한 Web Intelligence 에서 보고서를 실행하여 테스트를 수행해야 합니다.

이러한 작성 단계에 대한 내용은 유니버스 만들기 [페이지 242] 장을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1.7.1.4 유니버스 배포

유니버스를 중앙 관리 시스템(CMS) 리포지토리에 내보내어 테스트 또는 실제 운용을 위해 유니버스를 사용자에게 배포할 수 있습니다. 이 단계에 대한 자세한 내용은 유니버스 배포 [페이지 484] 장을 참조하십시오.

1.7.2 유니버스 개발 사이클

유니버스 개발은 계획, 디자인, 작성, 배포 및 관리 단계로 구성된 순환 과정입니다. 유니버스 디자인 도구를 사용하여 유니버스를 디자인하고 작성할 수 있지만, 유니버스의 실용성은 개발 주기의 여러 단계가 상호 간에 얼마나 잘 작용하는지에 따라 달라집니다.

이 단원에서는 유니버스 개발 프로젝트를 계획하고 구현하는 데 사용할 수 있는 유니버스 디자인 방법의 개요를 제공합니다.

아래 표에는 일반적인 유니버스 개발 사이클의 주요 단계가 간략하게 소개되어 있습니다.

표 3:

개발 단계	설명
준비	• 대상 데이터 소스를 확인하고 그 구조를 숙지합니다. • 각 대상 데이터베이스의 각 테이블에 어떠한 데이터가 포함되어 있는지 확인합니다. • 조인을 살펴봅니다. • 카디널리티를 확인합니다. • 가능한 사항을 점검합니다.
분석	• 사용자 수와 구성 방식을 확인합니다. 예를 들어, 사용자 그룹이 부서별로 조직화되어 있는지 업무 분야에 따라 구성되어 있는지 확인합니다. • 사용자가 필요로 하는 정보가 무엇인지 확인합니다. • 사용자가 필요로 하는 표준 보고서를 확인합니다. • 개체 이름을 적절하게 지정할 수 있도록 사용자의 업무에 사용되는 용어를 숙지합니다.
계획	프로젝트 전략을 확인합니다. 예를 들어, 얼마나 많은 유니버스를 만들어야 하고 어떠한 유니버스에 어느 정도의 기능을 어떠한 수준까지 연결해야 하는지 확인합니다.
구현	• 유니버스 디자인 도구를 사용하여 유니버스를 작성합니다. 이 설명서에서는 유니버스 개발 주기 중 유니버스 디자인 도구를 실제로 사용하는 단계에 대해 다룹니다. • 추정된 SQL 의 유효성과 신뢰성을 작성 과정에서 자주 테스트합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

개발 단계	설명
테스트	유니버스를 통해 얻으려는 정보를 어느 정도 이해하고 있는 소규모의 Web Intelligence 파워 유저 그룹을 구성합니다. 유니버스가 실제로 사용되는 상황을 시뮬레이트하는 철저한 테스트를 수행하도록 이 그룹의 사용자에게 요청합니다.
배포	최종 사용자가 액세스할 수 있는 중앙 관리 시스템(CMS) 리포지토리로 유니버스를 내보내어 배포합니다.
발전	데이터 소스와 사용자 요구 사항이 변경되고 확장되면 그에 맞춰 유니버스를 업데이트하고 유지 관리합니다.

노트

유니버스 디자인은 항상 데이터 소스 구조가 아니라 사용자 요구 사항을 중심으로 이루어져야 합니다.

1.7.3 유니버스 계획 및 구현 시간 최적화

사용자 요구 사항 분석 및 디자인은 전체 과정에서 가장 중요한 단계입니다. 사용자는 액세스 가능한 데이터 및 개체 이름 지정에 사용되는 업무 관련 용어를 통해 자신의 요구 사항을 충족시키는 데 사용될 유니버스의 개발 과정에 깊이 관여해야 합니다.

처음 세 단계를 올바르게 수행하면 구현 과정은 매우 빠르고 쉽게 진행됩니다.

유니버스 개발에 할당된 시간의 최대 80% 정도를 다음과 같은 초기 세 단계에 할애할 수 있습니다.

- 준비
- 분석
- 계획

유니버스의 기반을 다지는 데 이 정도의 시간을 사용하면 유니버스 디자인 도구를 실제로 사용하여 유니버스를 작성할 때 계획 및 분석에 필요한 시간을 충분히 할애하지 않은 경우보다 나머지 20% 정도의 시간을 훨씬 더 생산적으로 활용할 수 있습니다.

1.8 다국어 유니버스

SAP BusinessObjects Enterprise XI4 의 주요 기능 중 하나는 동일한 유니버스에서 다국어 메타데이터 및 보고서를 생성할 수 있다는 것입니다. 이 기능은 단일 메타데이터 유니버스 모델에서 지원되고 유니코드를 완전하게 지원하며 한 단계로 진행되는 로캘 구분형 다국어 보고 솔루션을 구현할 수 있도록 합니다. 이를 통해 사용자는 동일한 유니버스에서 보고서를 작성한 후 사용자 기본 설정에 따라 여러 언어로 보고서를 표시할 수 있습니다.

번역 가능한 유니버스 메타데이터는 다음과 같습니다.

- 유니버스 이름

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 유니버스 설명
- 클래스 이름
- 개체 이름
- 개체 이름, 설명 및 형식
- 사용자 지정 계층구조 이름
- 프롬프트 및 입력 열 질문

> **노트**
>
> 유니버스 메타데이터에 정의된 프롬프트만 번역할 수 있으며, @Prompt 함수를 사용하여 정의된 프롬프트는 번역할 수 없습니다.

유니버스에는 여러 로캘의 번역이 포함될 수 있습니다. 유니버스를 기반으로 보고서를 만들면 사용자의 기본 설정 보기 로캘에 따라 결정된 로캘로 메타데이터가 표시됩니다.

또한 유니버스는 사용 가능한 로캘이 없을 경우 사용되는 로캘인 대체 로캘을 정의합니다.

상태가 사용 준비로 설정된 로캘만 유니버스를 기반으로 보고서를 만든 사용자에게 표시됩니다.

이와 같이 표시되는 로캘에서 상태가 번역 표시 범주에 속한 메타데이터만 유니버스를 기반으로 보고서를 만든 사용자에게 표시됩니다. 번역 표시 범주에는 다음 상태의 메타데이터가 포함됩니다.

- 번역 검토 필요
- 로컬리제이션 검토 필요
- 각색 검토 필요
- 번역 완료
- 최종
- 사인오프

유니버스 메타데이터 번역, 유니버스 로캘 및 메타데이터 상태 설정은 번역 관리 도구를 통해 수행됩니다. 번역 및 로캘 매개 변수는 추가 XML 스트림에 .unv 파일 형식으로 저장됩니다.

유니버스 디자인 도구 사용자 인터페이스를 여러 언어로 표시할 수도 있습니다. 다음은 유니버스 디자인 도구의 다국어 유니버스 기능에 대한 설명입니다.

1.9 언어 및 로캘 정의

일부 언어는 몇 개의 국가와 관련되어 있습니다. 예를 들어 프랑스어(fr)는 프랑스(FR), 벨기에(BE) 및 스위스(CH)에서 사용되는 언어입니다. 이 예에서 fr-FR, fr-BE, fr-CH 는 각각 프랑스(FR), 벨기에(BE) 및 스위스(CH)에서 사용되는 프랑스어(fr)를 의미합니다.

표 4:

언어	국가
프랑스어	프랑스
프랑스어	벨기에
프랑스어	스위스

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

동시에 한 국가가 여러 언어와 관련될 수도 있습니다(fr-CH, de-CH, it-CH). 예를 들어 스위스에서는 독일어, 프랑스어 및 이탈리어어가 모두 사용됩니다.

표 5:

언어	국가
프랑스어	스위스어
독일어	스위스어
이탈리아어	스위스어

로캘은 언어 및 지리적 영역의 조합과 데이터의 정렬 방식을 정의합니다. 일반적으로 날짜와 시간은 특정 형식으로 지정됩니다. 언어와 국가를 조합하면(예: 프랑스의 프랑스어) 로캘은 운영 체제 또는 응용 프로그램에 따라 다음과 같이 표시됩니다.

표 6:

운영 체제	로캘 형식
Windows	프랑스어(프랑스) 로캘은 시스템 설정(국가)에서 가져옵니다.
Java	fr_FR
Sun Solaris	fr_FR.ISO8859-1

설명서 및 응용 프로그램에서 간단히 표현하기 위해 "언어"라는 용어에 언어와 로캘의 의미가 모두 포함되는 경우도 있습니다.

1.10 다른 로캘

표 7:

용어	정의
제품 언어	유니버스 디자인 도구 사용자 인터페이스 언어입니다. 메뉴와 메시지가 이 언어로 표시됩니다.
PVL(기본 설정 보기 로캘)	기본 설정 보기 언어 설정입니다. 이 언어는 리소스(문서 또는 유니버스) 컨텐트 또는 특성 목록에 포함되는 문자열, 텍스트 및 형식이 InfoView 나 Web Intelligence Rich Client 의 응용 프로그램에 표시될 때 사용되는 로캘을 정의합니다.
대체 로캘	기본 설정 보기 로캘을 사용할 수 없을 경우 사용되는 로캘입니다.
소스 언어	문서를 만들 때 사용된 로캘입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1.11 유니버스 디자인 도구 사용자 인터페이스에 대한 제품 언어 설정

유니버스 디자인 도구의 도구 > 옵션 설정에 있는 일반 탭의 사용 가능한 언어[UIL(사용자 인터페이스 언어)] 목록에서 언어를 선택합니다. 유니버스 메타데이터에는 아무 영향도 미치지 않습니다. 개체 이름, 컨텍스트 이름 및 클래스는 데이터베이스 요소의 원래 언어로 표시됩니다. 유니버스 메타데이터를 번역하려면 번역 관리 도구를 사용합니다.

1.12 다국어 유니버스 사용

사용자가 Web Intelligence 등에서 다국어 유니버스를 기반으로 보고서를 만드는 경우 메타데이터 표시에 사용되는 로캘은 로캘의 가용성/상태 및 메타데이터 범주에 따라 결정됩니다.

- 상태가 표시 범주에 속하며 이 로캘이 "사용 준비"로 정의된 경우에는 로캘로 번역된 메타데이터가 표시됩니다.
- 번역 메타데이터를 표시할 로캘은 다음과 같은 우선 순위로 사용됩니다.
  - 사용자의 기본 설정 보기 로캘
  - 기본 설정 보기 로캘을 사용할 수 없을 경우 이 유니버스에 정의된 대체 로캘
  - 이 유니버스에 정의된 대체 로캘이 없을 경우 사용자 기본 설정 보기 로캘의 주요 로캘
  - 주요 로캘을 사용할 수 없을 경우 원본 컨텐트가 표시됩니다. 이 원본 컨텐트는 유니버스 디자인 도구에 정의되어 있으므로 메타데이터입니다.

1.13 연결된 유니버스의 대체 로캘 결정

파생된 유니버스는 여러 주요 유니버스의 메타데이터를 재사용할 수 있습니다. 파생된 유니버스와 주요 유니버스에 여러 대체 로캘이 정의된 경우 다음과 같이 사용됩니다.

- 대체 로캘이 파생된 유니버스 수준에서 정의된 경우 이 대체 로캘이 사용됩니다.
- 파생된 유니버스 수준에서 정의된 대체 로캘이 없을 경우 파생된 유니버스에 정의된 첫 번째 주요 유니버스의 대체 로캘(있을 경우)이 사용됩니다.
- 주요 유니버스에 정의된 대체 로캘이 없을 경우 유니버스에 대체 로캘이 없는 것입니다.

1.14 번역 관리 도구

유니버스 디자인 도구를 통해서는 유니버스 메타데이터를 번역하거나 메타데이터 번역을 표시할 수 없습니다. 유니버스 디자인 도구에는 원본 유니버스 컨텐트만 표시됩니다. BusinessObjects Enterprise 제품군에서는 번역 작업을 위해 번역 관리 도구가 제공됩니다. 이 도구는 Windows 플랫폼에서만 사용할 수 있는 독립 실행형 응용 프로그램입니다.

번역 관리 도구를 사용하여 유니버스 디자이너는 다음 작업을 수행할 수 있습니다.

- 유니버스에서 새 로캘을 추가하여 사용 준비로 설정합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 유니버스의 대체 로캘을 정의합니다.
- 인터페이스를 통해 추가된 로캘로 유니버스 메타데이터를 번역합니다.
- 여러 로캘로 번역된 메타데이터의 번역 상태를 설정합니다.
- 외부 번역을 위해 유니버스 메타데이터를 XLIFF 파일로 내보낸 후 응용 프로그램으로 다시 가져옵니다.

유니버스 메타데이터가 번역된 후에는 다시 저장할 수 있으며 이러한 번역을 통해 다국어 보고서를 작성할 수도 있습니다. 자세한 내용은 번역 관리 도구 사용자 가이드를 참조하십시오.

1.15 다국어 데이터

다국어 데이터를 필터링하고 쿼리 시 사용자 기본 설정 보기 로캘의 데이터만 검색하려는 디자이너는 PREFERRED_VIEWING_LOCALE 및 DOMINANT_PREFERRED_VIEWING_LOCALE 변수를 통해 유니버스를 사용자 지정할 수 있습니다. @Variable 함수를 사용하면 됩니다.

1.16 유니버스 디자인 도구 예제 자료

다음과 같은 샘플이 유니버스 디자인 도구와 함께 제공됩니다.

1.16.1 데모 데이터베이스

이 설명서에 나와 있는 예제의 대부분은 Microsoft Access 2000 에서 만든 Club 데이터베이스를 기반으로 합니다. 이 데이터베이스는 가상의 기업인 Island Resorts 의 영업 부장이 영업 및 마케팅 분석을 수행하는 데 사용됩니다. 데이터베이스 파일인 Club.mdb 는 Business Objects 설치 경로의 Databases 하위 폴더에 있습니다.

이 데이터베이스의 구조에 대한 자세한 내용은 이 설명서의 뒷부분에 있는 부록을 참조하십시오.

이 릴리스에는 efashion 데이터베이스도 함께 제공됩니다. 이 MS Access 2000 데이터베이스에는 지난 3 년간 13 개 상점(미국 12 곳, 캐나다 1 곳)에서 판매된 211 가지 제품(색상별 차이를 고려할 경우 663 가지 제품)에 대한 이력이 기록되어 있습니다.

이 데이터베이스에 포함된 내용은 다음과 같습니다.

- 주별 판매 정보가 기록된 89,000 개의 행으로 이루어진 기본 팩트 테이블
- 프로모션이 포함된 두 번째 팩트 테이블
- 집계 탐색 기능이 설정된 두 개의 집계 테이블

1.16.2 데모 유니버스

Business Objects 설치 경로에 있는 Samples 폴더의 Universes 하위 폴더에는 beach.unv 라는 완전한 데모 유니버스가 제공됩니다. 이 유니버스는 위에서 설명한 Club 데이터베이스를 사용하여 작성된 것입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

이 유니버스를 사용하면 유니버스 디자인 도구로 특정 개체와 클래스를 작성하는 방법을 배울 수 있습니다.

efashion 데이터베이스를 사용하여 작성된 efashion 유니버스도 유니버스 디자인 도구와 함께 제공됩니다.

1.17 정보 디자인 도구에서 유니버스 사용

정보 디자인 도구를 사용하여 유니버스 디자인 도구, 유니버스 디자인 도구 데스크톱 에디션, Universe Designer 또는 Universe Designer Personal 에서 만들어진 .unv 형식의 유니버스로 작업할 수 있습니다. 이 파일은 정보 디자인 도구에서 직접 사용할 수 없으므로 파일 버전에 따라 먼저 변환하거나, 업그레이드한 후 변환해야 합니다. 여러 다양한 버전의 .unv 유니버스 파일을 사용하기 위해 필요한 단계 및 파일 변환 후 지원되는 기능에 대한 설명은 정보 디자인 도구를 참조하십시오.

> **노트**
>
> 정보 디자인 도구에서 사용하기 위해 .unv 파일을 변환한 후에는 처음에 해당 파일을 만들 때 사용된 도구에서는 열 수 없습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2 기본 작업 수행

2.1 개요

이 장에서는 유니버스를 만들고, 수정하고, 업데이트하기 위해 유니버스 디자인 도구에서 수행하는 기본적인 작업에 대해 설명합니다. 이 장은 다음 항목으로 구성되어 있습니다.

- 유니버스 디자인 도구 시작 [페이지 30]
- 유니버스 가져오기 [페이지 40]
- 유니버스 열기 [페이지 41]
- 유니버스 내보내기 [페이지 41]
- 유니버스 저장 [페이지 43]
- 유니버스 만들기 및 유니버스 매개 변수 설정 [페이지 68]
- 유니버스 디자인 도구 사용자 인터페이스 사용 [페이지 46]
- 찾기 및 바꾸기 사용 [페이지 51]
- 테이블 디스플레이 구성 [페이지 54]
- 스키마 표시 옵션 선택 [페이지 58]
- 유니버스 인쇄 [페이지 65]

2.2 유니버스 디자인 도구 시작

유니버스 디자인 도구는 중앙 관리 시스템(CMS) 리포지토리와 함께만 사용할 수 있습니다. 이 도구를 시작하기 전에 리포지토리에 로그인해야 합니다.

이 도구를 처음으로 시작하는 경우 기존 유니버스를 사용하려면 먼저 유니버스를 직접 열고 보안 연결을 통해 유니버스를 저장하여 리포지토리로 내보내야 합니다. 그런 다음 유니버스를 가져와 업데이트하고 업데이트된 버전을 내보냅니다. 이렇게 하면 CMS 와 로컬 유니버스 버전을 동기화할 수 있습니다.

유니버스 디자인 도구를 시작한 후에는 다음 방법 중 하나로 유니버스를 열 수 있습니다.

- 새 유니버스 만들기
- CMS 리포지토리에서 유니버스 가져오기
- 파일 시스템에서 직접 유니버스 열기

유니버스를 리포지토리에 내보낸 경우 Web Intelligence 사용자만 유니버스를 사용할 수 있습니다. 유니버스 디자인 도구에서의 가장 일반적인 작업 방식은 유니버스를 가져와서 필요에 따라 변경한 다음 업데이트된 유니버스를 내보내는 것입니다. 이렇게 하면 CMS(리포지토리) 버전을 파일 버전과 동기화할 수 있습니다.

> **노트**
>
> 유니버스를 파일 시스템에 저장할 수 있습니다. 대상 CMS 에 대한 연결 권한이 없는 다른 사용자와 유니버스를 공유하려는 경우에 유니버스를 파일 시스템에 저장합니다. 자세한 내용은 유니버스 저장 [페이지 43] 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

작업 표시줄에서 유니버스 디자인 도구를 시작하려면 현재 릴리스에 대해 설치된 Business Objects 제품 그룹에서 유니버스 디자인 도구 아이콘을 클릭합니다. 그러면 이 도구를 시작하기 전에 CMS 에 로그인하라는 메시지가 표시됩니다.

2.2.1 유니버스 디자인 도구 시작

유니버스 디자인 도구를 시작하려면

1. 작업 표시줄에서 시작 단추를 클릭합니다.
2. 프로그램 메뉴를 가리킵니다.
3. *BusinessObjects* 명령에서 유니버스 디자인 도구 프로그램을 클릭합니다.
   CMS 에 대한 로그인 상자가 나타납니다.
4. 다음 정보를 입력합니다. 이 정보는 일반적으로 BusinessObjects 관리자가 제공합니다.

   표 8:

로그인 정보	설명
시스템	CMS 서버의 이름입니다.
사용자 이름	리포지토리 사용자 이름입니다.
암호	리포지토리 암호입니다.
인증	보안 수준입니다.

5. 확인을 클릭합니다.
   유니버스 디자인 도구 시작 화면이 나타나고 빈 세션이 열립니다.
   제목 표시줄에서 사용자 이름과 CMS 이름을 확인할 수 있습니다.

유니버스 디자인 도구에 대해 설정된 옵션에 따라 유니버스 디자인 도구를 시작할 때 빠른 시작 유니버스 디자인 마법사를 자동으로 시작할 수 있습니다. 취소를 클릭하여 마법사를 닫습니다. 마법사의 다른 옵션을 비활성화하는 방법은 빠른 디자인 마법사 비활성화 [페이지 39] 단원을 참조하십시오. 빠른 디자인 마법사를 사용하려면 빠른 디자인 마법사 사용 [페이지 31] 단원을 참조하십시오.

2.2.2 빠른 디자인 마법사 사용

세션을 처음 시작하면 기본적으로 빠른 디자인 마법사가 나타납니다. 마법사를 사용하여 간편하게 유니버스를 만들거나 유니버스 디자인 도구의 사용법을 익힐 수는 있지만, 이 마법사는 최종 사용자의 보고 요구 사항을 충족시키는 완전한 유니버스를 만드는 데는 적합하지 않습니다.

빠른 디자인 마법사를 비활성화하여 유니버스 디자인 도구의 사용법을 익히는 수단으로만 활용하고, 실제 유니버스를 디자인하는 데는 사용하지 않는 것이 좋습니다. 빠른 디자인 마법사를 사용하는 방법에 대해 설명하는 빠른 디자인 마법사 사용 [페이지 31] 장을 제외하고 이 설명서에 나와 있는 모든 유니버스 디자인, 작성 및 관리 정보와 절차는 사용자가 이 마법사를 비활성화한 것을 전제로 진행됩니다. 빠른 디자인 마법사의 다른 옵션을 비활성화하는 방법은 빠른 디자인 마법사 비활성화 [페이지 39] 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.3 Designer XI R3 에서 XI R2 연결 및 유니버스 사용

유니버스 디자인 도구의 이번 릴리스에서는 연결에 액세스하고 XI R2™ CMS 에 저장된 유니버스를 열거나 가져올 수 있습니다. XI R2™ 유니버스 및 연결을 사용할 경우 다음 사항에 유의해야 합니다.

- DeskTop Intelligence XI R2™ 사용자는 XI 3.1™ 유니버스 및 XI R2™ 연결을 기반으로 Desktop Intelligence XI 3.1™ 에서 만든 문서를 새로 고칠 수 있습니다.
- Desktop Intelligence XI R2™ 사용자는 XI 3.1™ 유니버스 및 XI R2™ 연결을 기반으로 문서를 만들 수 있습니다.
- XI R2™ 연결을 편집하고 저장하려는 경우 연결이 저장될 경우 XI 3.1™ 연결로 저장되며 해당 연결을 사용하는 XI R2 ™ 보고서를 새로 고칠 수 없음을 나타내는 경고 메시지가 표시됩니다.
- XI 3.1 Universe Designer™에서는 XI R2™ 유니버스를 열 수 있지만 이전 버전의 Designer™에서는 XI 3.1™ 유니버스를 열 수 없습니다.

Desktop Intelligence XI R2™와 XI 3.1™ 간의 이러한 상호 연결 기능을 통해 관리자는 DeskTop Intelligence XI R2™ 및 XI 3.1 ™ 클라이언트와 업그레이드된 XI 3.1™ 서버 간의 연결을 계속 유지하면서 서버를 업그레이드할 수 있습니다. 이 과정은 대규모 클라이언트 모집단의 업그레이드가 관리되는 일시적인 단계입니다.

2.4 빠른 디자인 마법사를 사용하여 기본 유니버스 만들기

단순한 관계형 스키마를 기반으로 데모 또는 간단한 테스트 유니버스를 만들려면 기본적이지만 완전한 유니버스를 만들 수 있는 빠른 디자인 마법사를 사용하십시오. 생성된 유니버스는 즉시 사용하거나 개체를 수정하고 복잡한 새 유니버스를 만들 수 있습니다. 이와 같은 방법으로 유니버스의 품질과 구조를 점차적으로 재정의할 수 있습니다.

프로덕션 유니버스를 디자인할 경우에는 수동으로 유니버스를 만들어야 합니다. 이 가이드의 다른 모든 장은 유니버스를 수동으로 만드는 방법을 설명하기 위한 목적으로 작성되었습니다. 자동 유니버스를 만드는 방법은 이 단원에서만 설명합니다.

2.4.1 빠른 디자인 마법사를 사용하는 이유

빠른 디자인 마법사는 유니버스를 만드는 전 과정을 안내합니다. 이 마법사는 데이터베이스에 대한 연결 설정을 안내하고 간단한 클래스와 개체를 생성할 수 있도록 합니다. 또한 마법사에서는 개체, 조인 및 테이블의 자동 생성을 위한 기본 전략을 제공합니다.

빠른 디자인을 사용하면 다음과 같은 이점이 있습니다.

- 유니버스 디자인 도구를 처음 사용하는 경우 이 마법사를 통해 사용자 인터페이스와 유니버스 디자인에 대한 기본 사항을 익힐 수 있습니다.
- 데모 유니버스를 만들 경우 대부분의 디자인 프로세스를 자동화함으로써 시간을 단축할 수 있습니다. 마법사를 사용하면 유니버스의 작업 모델을 빠르게 설정한 다음, 대상 사용자의 요구에 맞게 유니버스를 사용자 지정할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

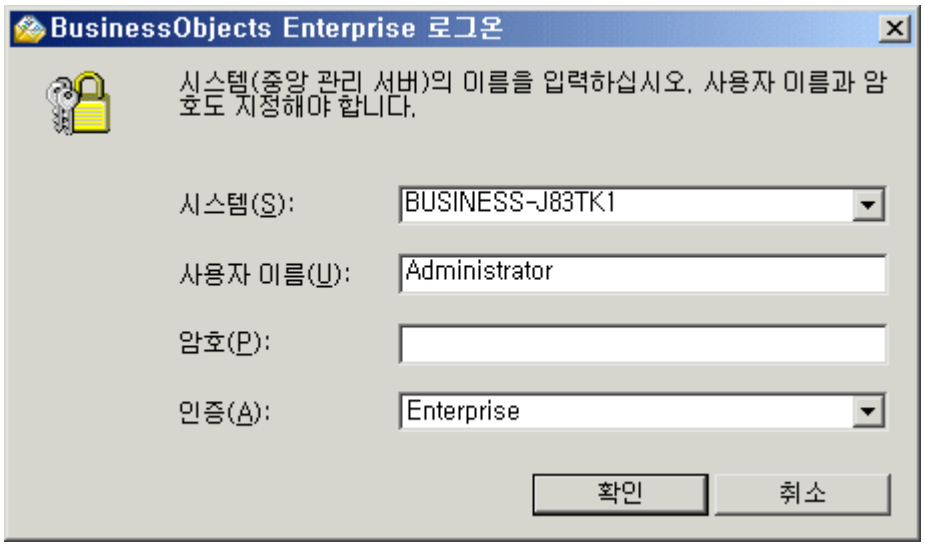
2.4.2 빠른 디자인 마법사 사용

빠른 디자인은 유니버스를 자동으로 생성하는 데 사용하는 마법사의 이름입니다. 마법사의 각 단계는 다음 각 단원에서 설명합니다.

2.4.2.1 빠른 디자인 마법사 시작

빠른 디자인 마법사를 시작하려면:

1. 유니버스 디자인 도구를 시작합니다.
   사용자 확인 대화 상자가 표시됩니다.



2. 사용자 확인 대화 상자에 사용자 이름과 암호를 입력합니다.
3. 확인을 클릭합니다.
   빠른 디자인 마법사의 시작 화면이 나타납니다.

> **노트**
>
> 다음 번에 세션을 시작할 때 마법사가 나타나지 않도록 하려면 시작할 때 이 마법사 실행 확인란을 선택하지 마십시오. 또한 옵션 대화 상자의 일반 탭에는 마법사 표시와 관련하여 시작 마법사 표시 및 파일/새로 만들기를 선택할 때 빠른 디자인 마법사 시작(도구 메뉴, 옵션 명령)의 두 가지 옵션이 있습니다.

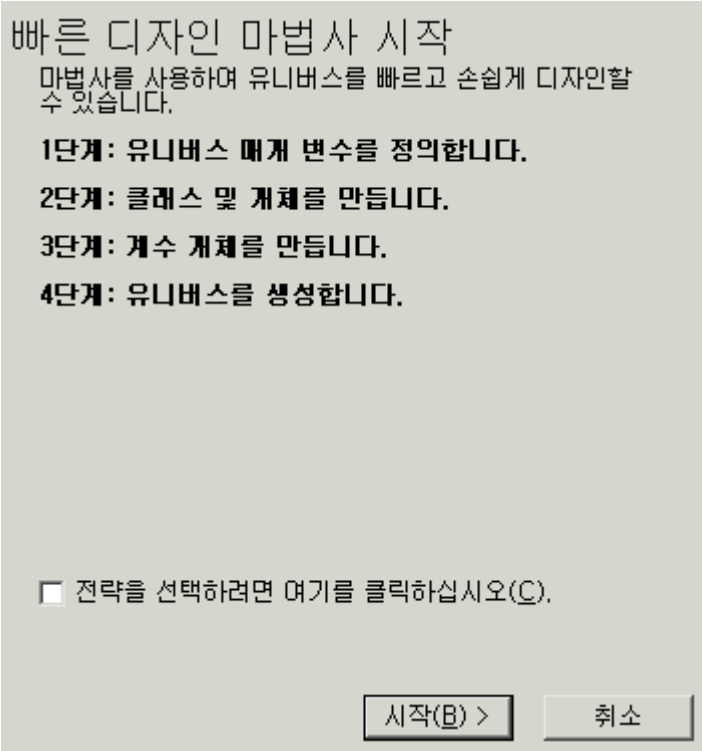
2.4.2.2 시작 화면

시작 화면에는 기본 유니버스를 만드는 데 필요한 4 단계에 대한 개요가 표시됩니다. 또한 전략을 선택하려면 여기를 클릭하십시오라는 확인란이 있습니다. 이 확인란을 클릭할 경우 유니버스 생성을 위한 전략을 선택할 수 있으며, 그렇지 않을 경우 유니버스 디자인 도구에서 기본 제공 전략이 적용됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

이후에 나타나는 빠른 디자인의 각 대화 상자에서는 작업을 수행하는 데 필요한 정보를 제공해야 합니다.

현재 대화 상자에서 다음 대화 상자로 이동하려면 다음을 클릭합니다. 뒤로를 클릭하여 이전 대화 상자로 돌아갈 수 있습니다. 그리고 언제라도 취소 단추를 클릭하면 프로세스를 끝내고 빠른 디자인을 종료할 수 있습니다.



"전략을 선택하려면 여기를 클릭하십시오." 확인란을 선택하면 전략이 나열된 대화 상자가 열립니다. 이 대화 상자는 전략 선택 [페이지 35]에서 설명합니다. 전략을 선택하거나 기본 전략을 사용할 수 있습니다.

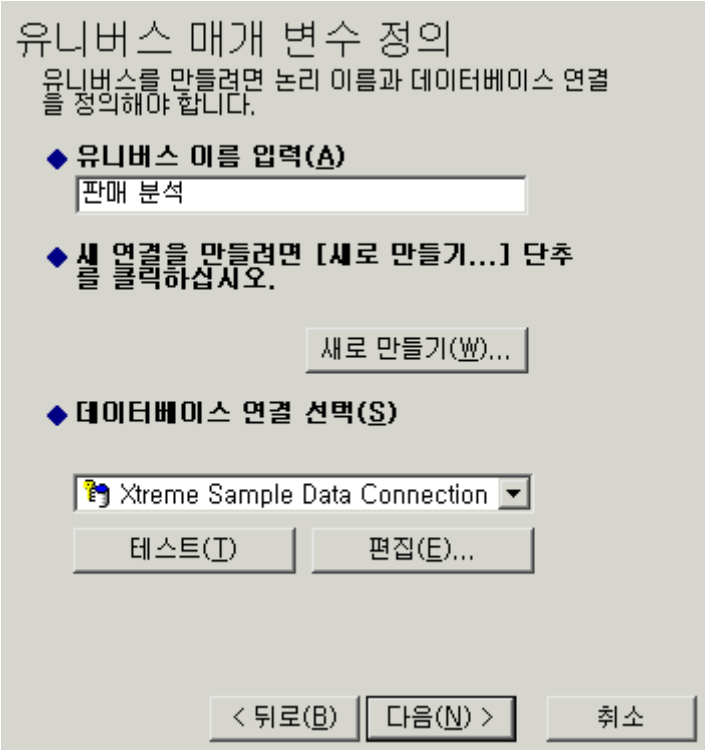
시작을 클릭하면 생성 프로세스가 시작됩니다.

2.4.2.3 유니버스 매개 변수 정의

이 단계에서는 유니버스 이름과 데이터베이스 연결 등의 유니버스 매개 변수를 정의합니다.

유니버스에 대해 최대 35 자의 영숫자를 사용하여 긴 이름을 입력할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



연결을 새로 만들거나 기존 연결을 선택할 수 있습니다. 연결을 만들려면 새로 만들기 단추를 클릭하고 나타나는 대화 상자에 필요한 매개 변수를 지정합니다. 이러한 대화 상자에 대한 자세한 내용은 유니버스 식별 매개 변수 수정 [페이지 72] 단원을 참조하십시오.

연결이 유효한지 확인하려면 테스트 단추를 클릭합니다. 편집 단추를 클릭하면 연결 매개 변수를 수정할 수 있습니다.

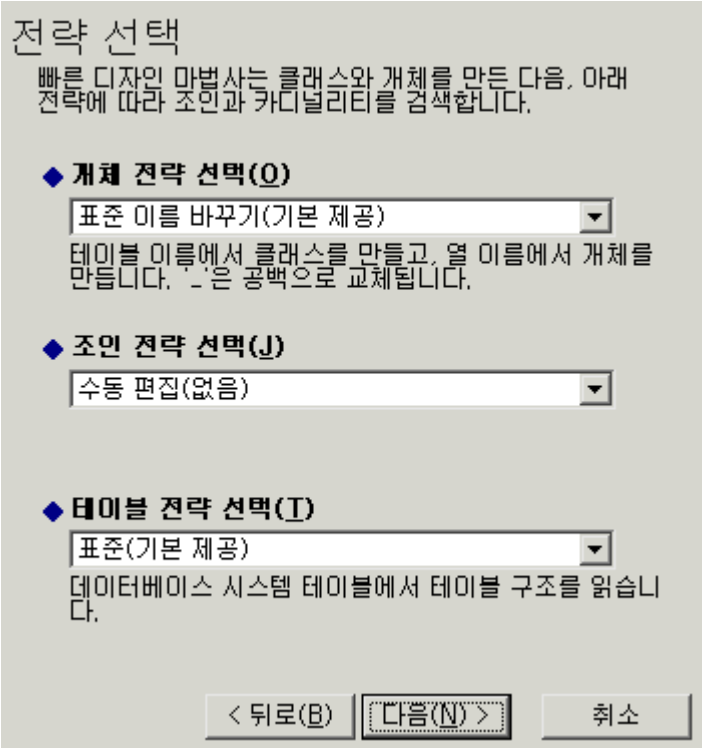
다음 단추를 클릭하여 다음 단계로 이동합니다.

2.4.2.4 전략 선택

시작 화면에서 전략 선택 확인란을 클릭한 경우에는 빠른 디자인 마법사에서 개체, 조인 및 테이블을 만드는 전략을 지정할 수 있는 옵션이 제공됩니다.

전략은 데이터베이스 또는 플랫 파일에서 구조 정보를 읽어오는 스크립트입니다. 유니버스 디자인 도구에서는 이러한 스크립트를 사용하여 개체, 조인 및 테이블을 자동으로 만듭니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



목록 상자에서 다른 전략을 선택하거나 전략을 선택하지 않을 수 있습니다. 현재 전략에 대한 간략한 설명이 목록 상자 바로 아래 나타납니다.

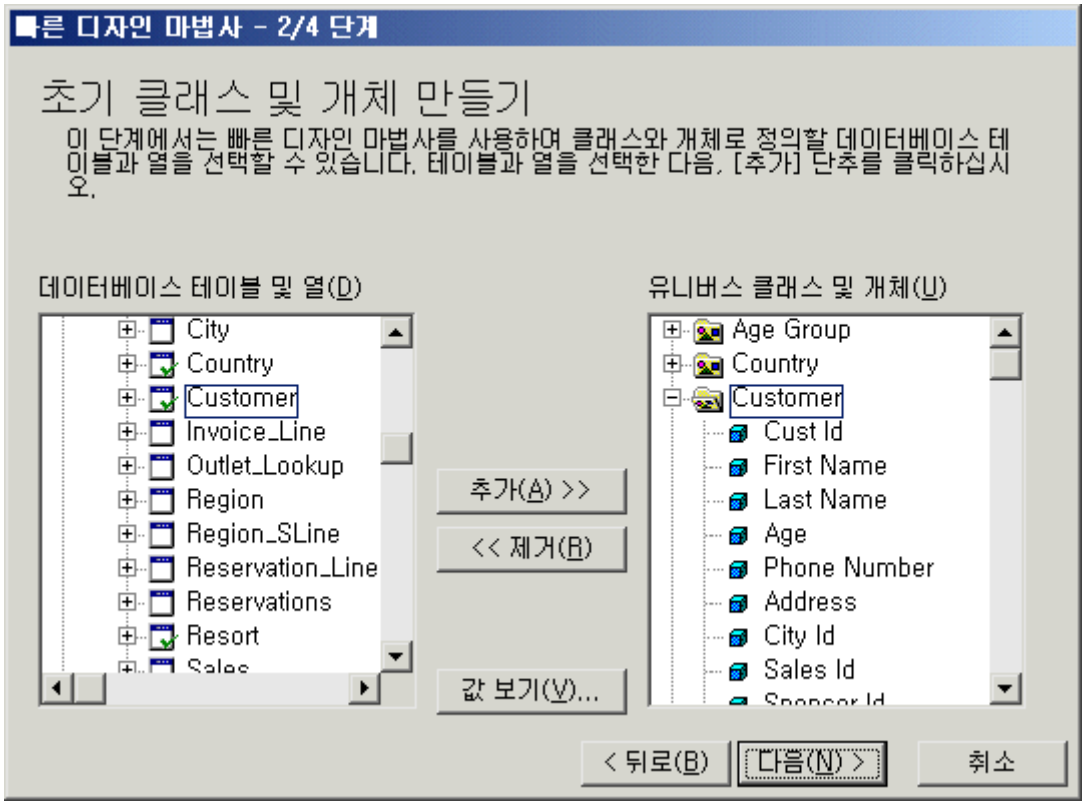
유니버스 디자인 도구에서 기본적으로 제공하는 내부 전략 이외에 사용자 고유의 외부 전략을 만들 수도 있습니다. 외부 전략을 사용하여 유니버스 만들기 사용자 지정 [페이지 365] 단원을 참조하십시오.

다음 단추를 클릭하여 다음 단계로 이동합니다.

2.4.2.5 초기 클래스 및 개체 만들기

데이터베이스 연결 매개 변수를 기준으로 마법사에서는 데이터베이스 테이블과 열 목록을 표시합니다. 왼쪽 창에서 테이블과 열을 선택한 후 오른쪽의 유니버스 클래스 및 개체 창에 추가하여 초기 클래스와 개체를 만들 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



기본적으로 왼쪽 창에는 테이블의 이름만 표시됩니다. 다음 방법을 사용하면 파일 트리 전체를 탐색하고 오른쪽 창에 클래스와 개체를 추가할 수 있습니다.

- 테이블의 열을 보려면 테이블 이름 옆에 있는 플러스 기호(+)를 클릭합니다.
- 테이블 또는 열의 데이터 값을 보려면 해당 값을 클릭한 다음 값 보기 단추를 클릭합니다.
- 하나의 테이블을 선택하려면 해당 테이블을 클릭한 다음 추가 단추를 클릭합니다.
- 여러 개의 연속된 테이블을 선택하려면 Shift 키를 누른 상태에서 처음 테이블과 마지막 테이블을 클릭합니다. 선택한 두 테이블 사이의 모든 테이블이 선택됩니다. 그런 다음 추가 단추를 클릭합니다.
- 연속되지 않은 여러 개의 테이블을 선택하려면 Ctrl 키를 누른 상태에서 각 테이블을 클릭합니다. 추가 단추를 클릭합니다.
- 테이블을 선택하는 또 다른 방법은 왼쪽 창에서 오른쪽 창으로 끌어 놓는 것입니다.
  테이블을 삽입하면 유니버스 디자인 도구에서 열을 모두 포함시킵니다.

오른쪽 창에는 클래스의 이름이 폴더 아이콘 옆에 표시됩니다. 클래스 이름 옆에 있는 플러스 기호(+)를 클릭하면 개체를 볼 수 있습니다. 클래스 또는 개체를 두 번 클릭하고 대화 상자에 새 이름을 입력하여 이름을 변경할 수 있습니다.

기본적으로 개체는 차원 개체가 될 수 있습니다. 차원 개체는 개체 이름 앞에 큐브 기호가 표시됩니다.

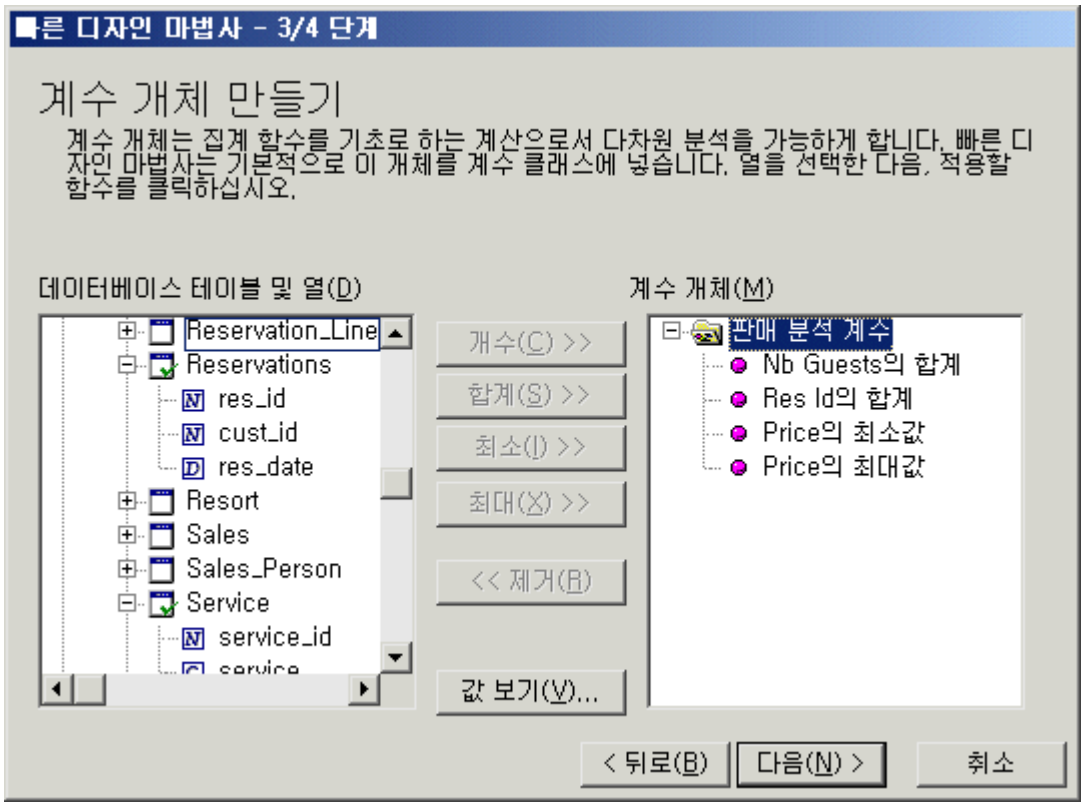
클래스 또는 개체를 제거하려면 해당 클래스나 개체를 클릭한 다음 제거 단추를 클릭합니다.

다음을 클릭하여 다음 단계로 이동합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.4.2.6 계수 개체 만들기

계수 개체는 Count, Sum, Minimum, Maximum 등의 집계 함수에서 파생됩니다. 이 개체 유형은 숫자 정보를 제공합니다. 계수 개체의 예는 아래 대화 상자의 오른쪽 창에 표시됩니다.



개체와 연결된 데이터 값을 보려면 해당 값을 클릭한 다음 값 보기 단추를 클릭합니다.

계수 개체를 만들려면 왼쪽 창에서 적절한 개체를 클릭한 다음 집계 단추를 클릭합니다. 직접 만드는 계수 개체의 이름은 원하는 대로 변경할 수 있습니다.

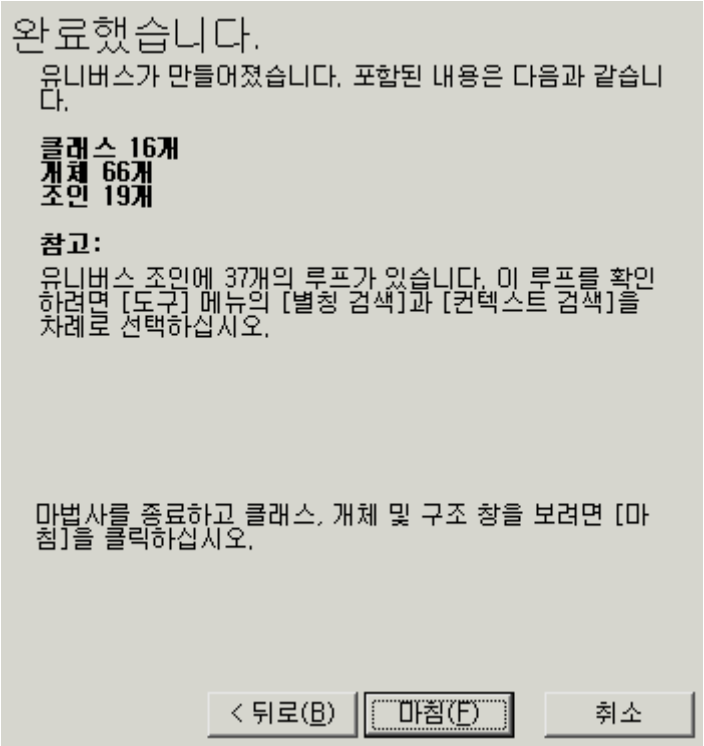
계수 개체를 하나 이상의 계수 클래스로 그룹화하면 유니버스의 구조가 향상됩니다. 또한 최종 사용자가 탐색을 쉽게 할 수 있게 됩니다. 계수 개체에 대한 자세한 내용은 계수 정의 [페이지 268] 단원을 참조하십시오.

다음을 클릭하면 빠른 디자인에서 유니버스 생성이 시작됩니다.

2.4.2.7 유니버스 생성

빠른 디자인 마법사는 사용자가 지정한 매개 변수를 기준으로 새로운 유니버스를 자동으로 생성합니다. 그리고 유니버스에서 만들어진 클래스, 개체 및 조인의 수를 나타냅니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



위 대화 상자에는 유니버스의 조인에 루프가 있음을 알리는 메시지가 표시됩니다. 유니버스 디자인 도구를 통해 별칭 및 컨텍스트를 사용하여 루프를 해결할 수 있습니다. 자세한 내용은 스키마 디자인 장을 참조하십시오.

마침 단추를 클릭하면 새로운 유니버스의 유니버스 창과 구조 창이 나타납니다.

2.4.2.8 작업 세션 종료

파일 > 다른 이름으로 저장을 선택하여 유니버스를 종료한 다음 파일 > 닫기를 선택하여 유니버스를 닫습니다.

유니버스를 저장하면 유니버스 디자인 도구에서 파일 이름을 입력하라는 메시지를 표시합니다. 유니버스 파일 이름에는 운영 체제에서 허용하는 최대 수의 문자가 포함될 수 있습니다. 이 파일의 확장자는 .unv 입니다. 기본적으로 유니버스 디자인 도구에서는 BusinessObjects 폴더의 하위 폴더인 Universe 에 이러한 파일을 저장합니다. Windows 2000 의 경우 이 폴더는 해당 사용자 프로필의 Local Data 폴더 아래 나타납니다.

유니버스 디자인 도구를 끝내려면 파일 > 끝내기를 선택합니다.

2.4.2.9 빠른 디자인 마법사 비활성화

세션을 처음 시작하면 기본적으로 빠른 디자인 마법사가 나타납니다. 새 유니버스를 만들 때 이 마법사가 자동으로 나타나지 않도록 하려면 다음과 같이 합니다.

빠른 디자인 마법사를 시작하려면

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자의 일반 페이지가 열립니다.
2. 시작 마법사 표시 확인란을 선택 취소합니다. (시작 마법사 시작 페이지에서 시작할 때 이 마법사 실행 확인란을 선택 취소한 경우에는 이 확인란이 이미 선택 취소되어 있습니다.
3. 파일에서 새로 만들기를 선택하면 빠른 디자인 마법사 시작 확인란을 선택 취소합니다.
4. 확인을 클릭합니다.

옵션 대화 상자의 일반 페이지에서 이 확인란을 선택하여 언제든지 빠른 디자인 마법사를 실행할 수 있습니다. 빠른 디자인 마법사를 사용하는 방법에 대한 자세한 내용은 빠른 디자인 마법사 사용 [페이지 31] 단원을 참조하십시오.

2.4.3 빠른 디자인 마법사로 만든 유니버스에 대한 추가 작업

빠른 디자인 마법사로 기본 유니버스를 만든 후에는 이를 기반으로 조인을 편집하고 별칭 또는 컨텍스트를 사용하여 모든 루프를 해결할 수 있습니다. 다양한 유니버스 디자인 도구 기능을 사용하여 더욱 복잡한 구성 요소로 유니버스를 향상시킬 수도 있습니다. 이러한 정보는 이 가이드의 관련 단원을 참조하십시오.

2.5 유니버스 가져오기

리포지토리의 유니버스 폴더에 저장된 하나 이상의 유니버스를 가져올 수 있습니다. 이미 리포지토리로 내보낸 유니버스만 가져올 수 있습니다.

유니버스를 가져올 때 CMS 는 리포지토리 파일 시스템에서 유니버스 버전을 확인합니다. 버전이 동일하면 유니버스 디자인 도구에서 유니버스를 사용할 수 있습니다. 리포지토리 파일 시스템의 유니버스 버전이 CMS 버전보다 최신 것이면 폴더의 유니버스를 교체할지 확인하는 메시지가 나타납니다. 예로 대답하면 리포지토리 파일 시스템의 유니버스가 CMS 의 버전으로 대체됩니다.

2.5.1 리포지토리에서 유니버스 가져오기

2.5.1.1 리포지토리에서 유니버스를 가져오려면

1. 파일 > 가져오기 를 선택합니다.

   유니버스 가져오기 대화 상자가 나타납니다.
2. 드롭다운 목록 상자에서 유니버스 폴더를 선택합니다.

   또는

   찾아보기 단추를 클릭하고 폴더 브라우저를 사용하여 유니버스를 선택합니다.

   유니버스를 이 폴더에서 가져오려고 합니다.
3. 유니버스를 잠그려면 유니버스 이름을 두 번 클릭합니다.

   잠긴 유니버스에는 자물쇠 기호가 나타납니다. 유니버스 잠금을 해제하려면 다시 두 번 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4. 유니버스 이름을 클릭합니다.

   가져오려는 유니버스의 이름입니다.
5. 가져오기 폴더 상자에서 가져오기 폴더의 파일 경로를 확인합니다.

   유니버스를 이 폴더로 가져옵니다.
6. 확인을 클릭합니다.

2.5.2 열기 및 가져오기의 차이

파일 시스템에서 직접 유니버스를 열 수 있습니다. 이 유니버스를 저장하면 유니버스가 파일 시스템에만 저장되고 CMS 에서는 업데이트되지 않습니다. 이 유니버스의 업데이트 내용은 Web Intelligence 사용자에게 표시되지 않습니다.

유니버스를 가져오면 리포지토리에서 사용 가능한 현재 버전을 유니버스 디자인 도구에서 사용할 수 있습니다. 유니버스 수정을 마친 다음 유니버스를 리포지토리에 내보냅니다. 최신 변경 사항을 반영하여 CMS 가 업데이트됩니다.

2.6 유니버스 열기

메뉴 명령을 사용하거나 열기를 클릭하여 유니버스를 엽니다. 유니버스를 가져오지 않고 직접 열면 로컬 파일 시스템의 버전이 열립니다. 이 버전은 CMS 의 최신 버전과 다를 수 있습니다.

2.6.1 유니버스를 직접 열려면

1. 파일 > 열기 를 선택합니다.

   열기 상자가 열리고 유니버스 파일의 기본 저장 위치로 지정되어 있는 디렉터리가 표시됩니다. 이 디렉터리는 옵션 대화 상자의 저장 페이지(도구 > 옵션 > 저장)에서 설정할 수 있습니다.
2. 필요한 경우 유니버스 파일(.UNV)이 있는 디렉터리로 이동합니다.
3. 유니버스 파일을 선택하고 열기를 클릭합니다.

   또는

   유니버스 파일을 두 번 클릭합니다.

현재 유니버스 디자인 도구 창에 유니버스가 열립니다.

2.7 유니버스 내보내기

유니버스를 리포지토리에 내보내면 Web Intelligence 사용자와 다른 디자이너가 유니버스를 사용할 수 있도록 만들 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스를 내보내는 경우 다음과 같은 결과가 발생합니다.

- 리포지토리 파일 시스템의 선택된 유니버스 폴더로 유니버스가 이동합니다.
  및
- 중앙 관리 시스템(CMS)에서 유니버스가 작성됩니다.

유니버스를 리포지토리에 내보낼 때마다 CMS 의 유니버스 버전이 업데이트됩니다. 이 버전이 Web Intelligence 사용자에게 제공됩니다.

> **노트**
>
> 유니버스를 저장하는 경우에는 유니버스를 내보내는 경우와 결과가 다릅니다. 유니버스를 저장하면 리포지토리 로컬 파일 시스템의 유니버스가 업데이트되지만 유니버스의 CMS 리포지토리 버전은 업데이트되지 않습니다. 자세한 내용은 내보내기 및 저장의 차이 [페이지 43] 단원을 참조하십시오.

2.7.1 리포지토리 파일 시스템에서의 유니버스 구성 방식

리포지토리는 CMS 서버와 로컬 파일 시스템에 유니버스를 저장합니다. 사용자는 로컬 파일 시스템의 유니버스 버전을 사용하여 작업합니다. 로컬 파일 시스템은 유니버스 디자인 도구가 설치된 서버입니다. 사용자가 작업한 유니버스는 기본적으로 다음과 같은 사용자 프로필 경로의 유니버스 폴더에 저장됩니다.

표 9:

\\Documents and Settings\<**사용자**>\Application Data\Business Objects\Business Objects 12.0\universes\@<**리포지토리 이름**>\universe folder\<**유니버스**>.unv

CMS 서버에 저장되는 유니버스는 버전 제어에 사용됩니다. 업데이트된 유니버스를 리포지토리에 내보내면 업데이트된 유니버스가 CMS 서버에 복사됩니다.

2.7.2 리포지토리로 유니버스 내보내기

2.7.2.1 리포지토리로 유니버스를 내보내려면

1. 파일 > 내보내기 를 선택합니다.

   유니버스 내보내기 대화 상자가 나타납니다.
2. 폴더 드롭다운 목록 상자에서 유니버스 폴더를 선택합니다.

   또는

   찾아보기 단추를 클릭하고 폴더 브라우저에서 유니버스 폴더를 선택합니다.

   유니버스를 이 폴더로 내보내려고 합니다.
3. 유니버스를 잠그려면 유니버스 이름을 두 번 클릭합니다.

   잠긴 유니버스에는 자물쇠 기호가 나타납니다. 유니버스 잠금을 해제하려면 다시 두 번 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4. 그룹 목록 상자에서 그룹을 클릭합니다. 이 그룹은 내보낸 유니버스를 사용할 사용자 그룹입니다.
5. 유니버스 목록 상자에서 유니버스를 클릭합니다. 유니버스 목록 상자에 활성 유니버스의 이름이 표시됩니다.
6. 열리지 않은 다른 유니버스를 내보내려면 유니버스 추가 단추를 클릭한 후 브라우저를 사용하여 다른 유니버스를 선택합니다.
7. 확인을 클릭합니다.

2.7.3 내보내기 및 저장의 차이

유니버스를 저장하면 리포지토리 파일 시스템의 버전이 업데이트됩니다. 이 경우 CMS 버전은 업데이트되지 않습니다.

유니버스를 내보내면 리포지토리 파일 시스템의 업데이트된 버전이 CMS 의 업데이트된 유니버스와 동기화됩니다.

업데이트된 버전을 내보내지 않고 유니버스를 저장하면 CMS 가 업데이트되지 않습니다. 저장된 유니버스는 다른 사용자가 사용할 수 없습니다.

리포지토리의 각 유니버스에는 시스템 식별자가 지정됩니다. 식별자에 대한 자세한 내용은 리포지토리에서 유니버스 식별 [페이지 484] 단원을 참조하십시오.

다른 디자이너가 리포지토리에서 유니버스를 잠근 경우에는 유니버스를 내보낼 수 없습니다.

유니버스는 보안 연결로 정의된 경우에만 내보낼 수 있습니다.

2.8 유니버스 저장

작업 세션 동안 정기적으로 유니버스를 저장해야 합니다. 유니버스를 저장하면 유니버스 디자인 도구에서는 해당 유니버스를 로컬 파일 시스템에 `.unv` 확장자의 파일로 저장합니다.

Web Intelligence 에서 사용자는 유니버스 이름(긴 이름)으로 유니버스를 식별합니다.

유니버스를 저장하는 경우 CMS 에는 변경 내용이 저장되지 않습니다. 유니버스 업데이트를 마친 후에 유니버스를 CMS 에 내보내야 합니다.

유니버스 이름(긴 이름)과 `.unv` 파일 이름에 사용할 수 있는 문자의 최대 길이는 다음과 같습니다.

표 10:

이름 종류	최대 문자 수
유니버스 이름	100
.unv 이름	운영 체제 최대값

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.8.1 식별자로서의 유니버스 파일 이름

유니버스를 기반으로 보고서를 만든 후에는 유니버스 파일 이름 `.unv` 를 변경하지 말아야 합니다. 파일 이름을 변경하면 이전 이름의 유니버스에 기반하여 작성된 보고서가 이름이 변경된 후의 유니버스를 가리키지 않게 됩니다.

2.8.2 유니버스 저장

유니버스 이름은 `.unv` 이름과 다를 수 있습니다.

다른 이름으로 저장을 사용하여 유니버스를 새 이름으로 저장하는 경우 새 유니버스는 CMS 에 연결되지 않습니다. 새 유니버스의 버전을 만들려면 새 유니버스를 CMS 에 내보내야 합니다.

다음과 같은 방법을 사용하여 유니버스를 저장할 수 있습니다.

- 메뉴 모음에서 파일 > 저장 을 선택합니다.
- 저장 아이콘을 클릭합니다.
- 키보드에서 *Ctrl+S* 를 누릅니다.

2.8.3 유니버스 정의를 PDF 로 저장

유니버스 정보를 Adobe PDF 파일로 저장합니다. 유니버스에 대해 출력할 수 있는 것과 동일한 정보를 저장할 수 있습니다. 이러한 정보에는 다음 사항이 포함됩니다.

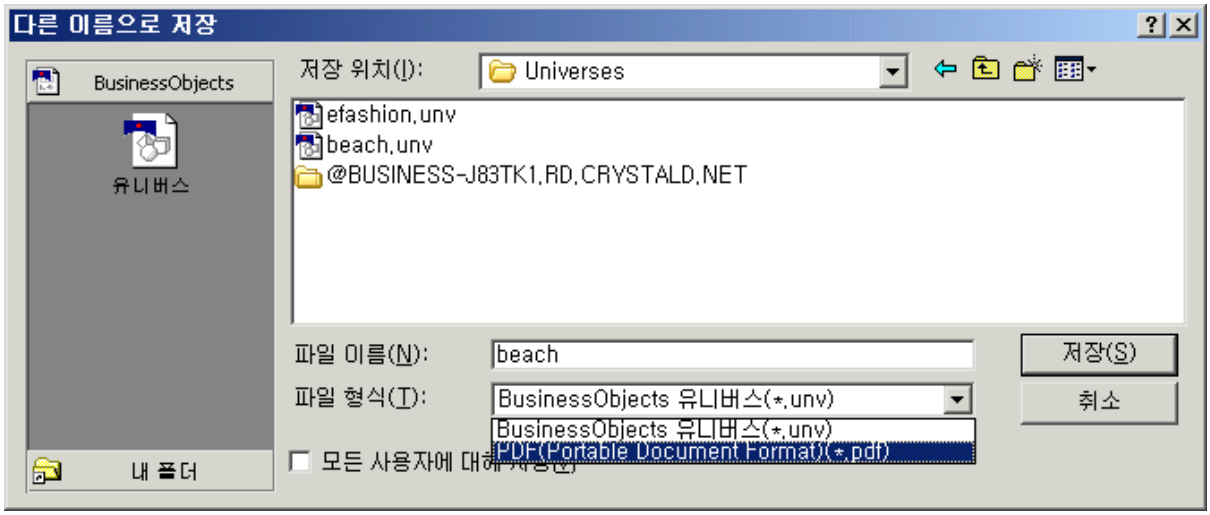
- 일반 정보: 매개 변수, 연결된 유니버스 및 그래픽 테이블 스키마
- 구성 요소 목록: 개체, 조건, 계층, 테이블, 조인 및 컨텍스트를 포함한 유니버스의 구성 요소 목록
- 구성 요소 설명: 유니버스의 개체, 조건, 계층, 테이블, 조인 및 컨텍스트에 대한 설명

인쇄 옵션 대화 상자(도구 > 옵션 > 인쇄)에서 PDF 에 표시할 구성 요소를 선택할 수 있습니다. 이러한 옵션은 유니버스 인쇄 [페이지 65] 단원에서 설명합니다.

유니버스 정보를 PDF 파일로 저장하려면

1. 파일 > 다른 이름으로 저장 을 선택합니다.
2. 파일 형식 드롭다운 목록 상자에서 *PDF(Portable Document Format)(*.pdf)*을 선택합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



3. 저장을 클릭합니다.

2.8.3.1 기본 저장 옵션 설정

기본적으로 유니버스 디자인 도구에서는 파일을 Business Objects 경로의 하위 폴더인 Universe 에 저장합니다. 다음과 같이 기본 저장 폴더를 다른 위치로 지정할 수 있습니다.

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자가 나타납니다.
2. 저장 탭을 클릭합니다.
   저장 페이지가 나타납니다.
3. 기본 유니버스 폴더 텍스트 상자에 파일 경로를 입력합니다.
   또는
4. `.unv` 파일이 포함된 폴더를 찾습니다.
5. 자동 저장 시간을 지정하려면 자동 저장 확인란을 선택하고 분 값 선택 상자에서 저장 간격 시간을 선택하거나 입력합니다.
6. 확인을 클릭합니다.

2.9 유니버스 닫기

다음과 같은 방법을 사용하여 유니버스를 닫을 수 있습니다.

유니버스를 닫으려면

- 메뉴 모음에서 파일 > 닫기 를 선택합니다.
- 유니버스 창의 오른쪽 맨 위에 있는 창 닫기 단추를 클릭합니다.
- 키보드에서 *Ctrl+W* 를 누릅니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.10 여러 디자이너와 작업

여러 명의 디자이너가 버전 간에 충돌을 일으키지 않고 동일한 유니버스를 사용할 수 있는 다중 사용자 환경에서 유니버스 디자인 도구를 사용할 수 있습니다.

한 번에 한 명의 디자이너만 유니버스에 대한 수정 작업을 할 수 있도록 유니버스를 잠그고, 변경 내용 추적을 위해 유니버스에 버전 번호를 할당할 수 있습니다.

2.10.1 유니버스 잠금

유니버스 폴더에 저장된 유니버스는 필요한 사용자 권한이 부여된 여러 디자이너가 공유할 수 있습니다.

한 번에 한 명의 디자이너만 한 유니버스에서 작업을 수행할 수 있습니다. 유니버스 작업을 원하는 디자이너는 다른 디자이너가 유니버스를 잠그지 않은 경우에만 작업을 수행할 수 있습니다.

> **노트**
>
> 유니버스는 가져오기 또는 내보내기 대화 상자에서 잠급니다. 유니버스가 잠기면 자물쇠 기호가 유니버스 이름 옆에 나타납니다. 다른 디자이너가 유니버스를 잠근 경우에는 자물쇠 기호가 흐리게 나타납니다.

2.10.2 수정 번호

유니버스 폴더로 유니버스를 내보낼 때마다 유니버스 디자인 도구에서는 유니버스의 수정 번호를 증분합니다. 따라서 유니버스의 최신 버전을 알 수 있습니다. 유니버스 매개 변수의 요약 탭(파일 > 유니버스 매개 변수 > 요약)에는 리비전 번호가 표시됩니다.

2.11 유니버스 디자인 도구 사용자 인터페이스 사용

유니버스 디자인 도구 사용자 인터페이스는 Microsoft Windows 표준을 따릅니다. 이 사용자 인터페이스에는 창, 메뉴, 도구 모음, 바로 가기 키 및 온라인 도움말 기능이 있습니다.

2.11.1 사용자 인터페이스의 주 구성 요소

각 유니버스는 하나의 유니버스 창에 포함되고 각 창은 주 창에 포함됩니다.

테이블 탐색기라는 별도의 독립된 창을 사용할 수도 있습니다. 이 창에는 연결된 데이터베이스에서 사용할 수 있는 테이블이 모두 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.11.1.1 유니버스 창

유니버스 창은 다음과 같은 두 개의 창으로 나뉘어 있습니다.

표 11:

창	표시
구조	유니버스의 기본 대상 데이터베이스에 대한 그래픽 표현이 표시됩니다. 여기에는 최종 사용자가 쿼리를 실행하기 위해 사용하는 개체에 매핑된 테이블과 조인이 포함됩니다.
유니버스	유니버스에 정의된 클래스와 개체가 표시됩니다. 이러한 유니버스 구성 요소는 Web Intelligence 사용자가 쿼리를 만들기 위해 보거나 사용할 수 있습니다.

2.11.1.2 테이블 탐색기

테이블 탐색기는 연결된 데이터베이스에서 사용할 수 있는 테이블을 표시하는 창입니다. 테이블을 선택한 다음 구조 창으로 끌어 놓거나 테이블 탐색기에서 적절한 테이블을 두 번 클릭하여 구조 창에 테이블을 삽입할 수 있습니다.

다음 방법 중 하나를 사용하여 테이블 탐색기를 표시할 수 있습니다.

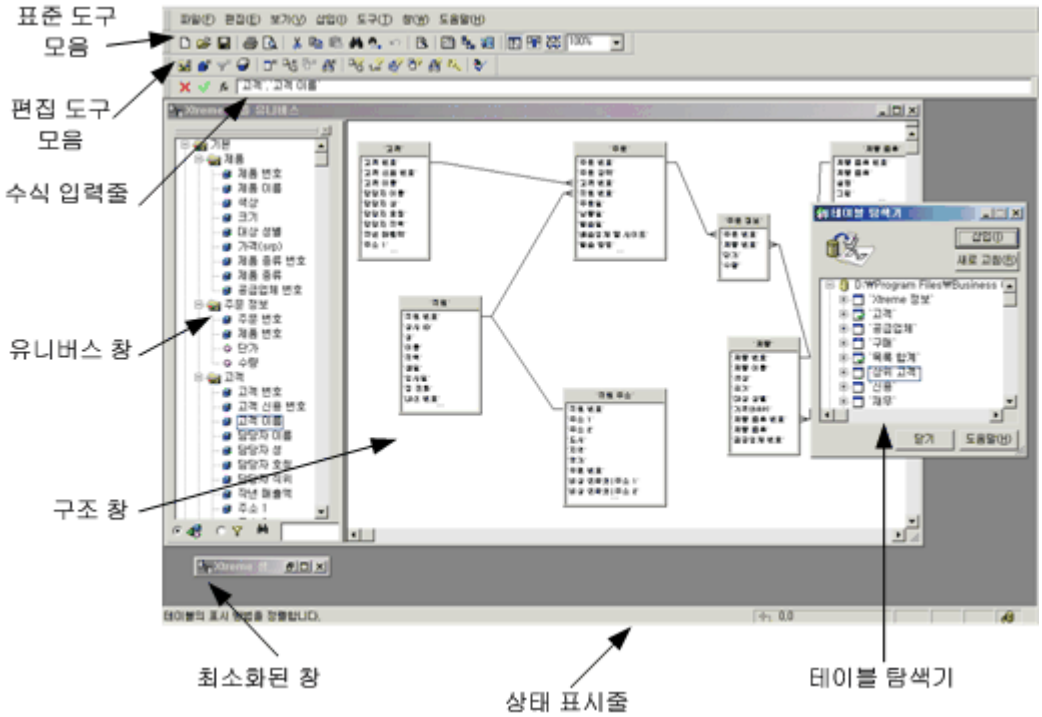
- 구조 창의 배경을 두 번 클릭합니다.
- 구조 창의 배경을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 테이블 삽입을 선택합니다.
- 삽입 > 테이블 을 선택합니다.

> 노트
>
> 테이블 탐색기를 사용하는 방법에 대한 자세한 내용은 "스키마 디자인" 장을 참조하십시오.

2.11.2 유니버스 디자인 도구 사용자 인터페이스

아래 그림에는 인터페이스의 주 구성 요소에 대한 이름이 나와 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



2.11.3 창 조작

다음과 같은 방법으로 사용자 인터페이스의 창을 사용할 수 있습니다.

- 작업 세션에서 한 번에 여러 개의 유니버스를 대상으로 작업할 수 있습니다. 하나의 구조 창과 하나의 유니버스 창에 각 유니버스가 표시됩니다.
- 최근에 열었던 유니버스는 파일 메뉴의 아래쪽에 나열됩니다. 이곳에 나열되는 유니버스의 수를 변경하려면 ▶ 도구 > 옵션 > 일반 ◀을 선택한 다음 최근에 사용한 파일 목록에서 유니버스의 수를 지정합니다.
- 유니버스 디자인 도구 창에서 창을 이동하거나 창 크기를 조정하거나 창을 최소화할 수 있습니다.
- ▶ 창 > 정렬 ◀을 선택한 다음 계단식 배열, 가로 바둑판식 배열 또는 세로 바둑판식 배열을 선택하여 자신에게 가장 편리한 방식으로 창을 배치할 수 있습니다.
- ▶ 창 > 아이콘 정렬 ◀을 선택하여 유니버스 디자인 도구 창에서 최소화된 모든 창을 나란히 배열할 수 있습니다.

2.11.4 도구 모음 사용

유니버스 디자인 도구 창에는 표준 도구 모음과 편집 도구 모음이라는 두 가지 도구 모음이 포함되어 있습니다.

유니버스 창 또는 구조 창 중 어느 창이 활성화되어 있는지에 따라 각 도구 모음에서 선택할 수 있는 단추가 달라집니다. 사용할 수 없는 단추는 흐리게 표시됩니다.

도구 모음은 도킹할 수 있습니다. 도구 모음을 유니버스 창의 아무 위치로나 끌어 놓아 배치할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.11.4.1 도구 모음 이동

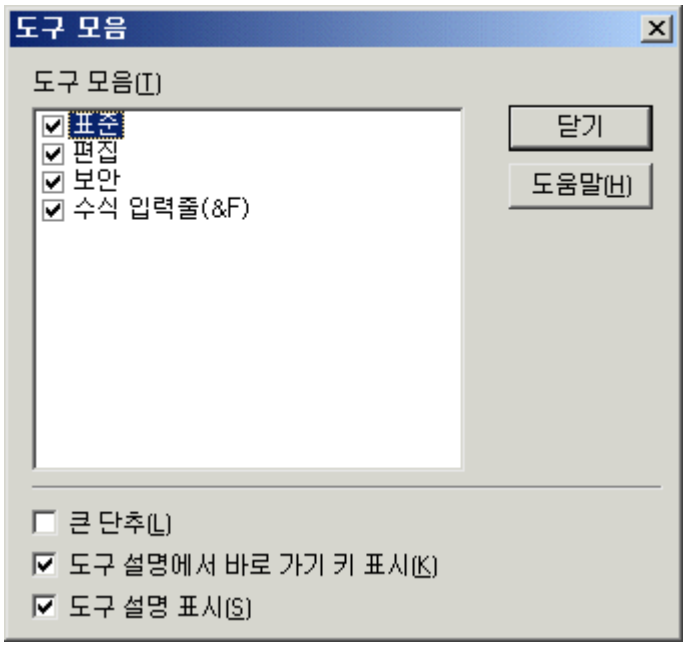
도구 모음을 이동하려면

1. 도구 모음을 둘러싸고 있는 사각형 내부를 클릭합니다.
   두 도구 모음에 대한 영역은 위 그림에 나와 있습니다.
2. 마우스 왼쪽 단추를 누른 채로 도구 모음을 원하는 위치로 끌어 옵니다.
3. 마우스 단추를 놓습니다.
   도구 모음이 따로 표시됩니다.

2.11.4.2 도구 모음 숨기기 및 표시

도구 모음을 표시하거나 숨기려면

1. 보기 > 도구 모음 을 선택합니다.
   도구 모음 대화 상자가 나타납니다.



2. 도구 모음에 해당하는 확인란을 선택하거나 해제합니다.
3. 대화 상자의 아래쪽에 나열되어 있는 도구 모음 단추, 도구 설명 및 바로 가기 키의 디스플레이에 대한 옵션을 선택하거나 해제합니다.
4. 확인을 클릭합니다.

2.11.5 유니버스 디자인 도구에서 작업 수행

유니버스 디자인 도구에서는 다음과 같은 방법으로 작업을 수행합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 메뉴에서 명령을 선택합니다.
- 키보드에서 Alt 키와 함께 바로 가기 키를 누릅니다.
- 도구 모음에서 단추를 클릭합니다.

2.11.5.1 마우스 사용

다음과 같이 마우스 단추를 한 번 또는 두 번 클릭할 수 있습니다.

한 번 클릭

다음 작업을 수행하려면 마우스 단추를 한 번 클릭합니다.

- 표준 작업 수행(명령 선택 또는 단추 클릭)
- 유니버스 창, 구조 창 또는테이블 탐색기에서 요소 선택
- 유니버스 디자인 도구 창에서 구성 요소를 하나 이상 선택할 경우 마우스 오른쪽 단추를 한 번 클릭하면 팝업 메뉴가 표시됩니다. 이 메뉴에는 선택된 구성 요소와 관련된 명령이 포함되어 있습니다.

두 번 클릭

다음과 같은 유니버스 구조를 두 번 클릭하면 디스플레이 변경에 영향을 미치거나 속성을 수정할 수 있습니다.

표 12:

두 번 클릭	결과
구조 창의 빈 공간	테이블 탐색기가 열립니다.
구조 창의 테이블	테이블 디스플레이가 수정됩니다. 테이블과 열을 세 가지 보기 중 하나로 표시할 수 있습니다. 자세한 내용은 목록 모드 사용 [페이지 55] 단원을 참조하십시오.
구조 창의 조인	조인에 대한 조인 편집 대화 상자가 나타납니다. 이 대화 상자에서 조인 속성을 수정할 수 있습니다.
유니버스 창의 클래스	클래스에 대한 속성 편집 대화 상자가 나타납니다. 이 대화 상자에서 클래스 속성을 수정할 수 있습니다.
유니버스 창의 개체	개체에 대한 속성 편집 대화 상자가 나타납니다. 이 대화 상자에서 개체 속성을 수정할 수 있습니다.
유니버스 창의 조건 보기에 있는 조건 개체	조건 개체에 대한 속성 편집 대화 상자가 나타납니다. 이 대화 상자에서 개체 속성을 수정할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.11.5.2 작업 실행 취소

이전에 수행한 작업을 다음과 같은 두 가지 방법으로 취소할 수 있습니다.

- 편집 > 실행 취소 를 선택합니다.
- 실행 취소를 클릭합니다.

2.12 찾기 및 바꾸기 사용

찾기 기능을 사용하여 유니버스 창과 구조 창에서 문자 또는 텍스트 문자열을 찾을 수 있습니다. 찾기 및 바꾸기를 사용하면 유니버스에서 구조의 이름과 설명에 있는 문자 또는 텍스트를 찾고 바꿀 수 있습니다.

2.12.1 찾기 사용

유니버스 및 구조 창에서 유니버스 구조에 포함된 텍스트를 검색할 수 있습니다.

2.12.1.1 찾기 옵션 설정

사용 가능한 찾기 옵션은 유니버스 창과 구조 창 중 어느 것이 활성화되어 있는지에 따라 달라집니다.

문자열을 찾기 위한 다음과 같은 검색 옵션을 설정할 수 있습니다.

표 13:

옵션	사용할 수 있는 옵션	설명
찾을 내용	유니버스 또는 구조 창이 활성화된 경우	검색할 텍스트 문자열을 지정합니다.
대/소문자 구분	유니버스 또는 구조 창이 활성화된 경우	검색 조건에 대/소문자 일치 여부를 포함시킵니다.
단어 단위로	유니버스 또는 구조 창이 활성화된 경우	전체 문자열이 일치하도록 지정합니다.
이름에서도 조회	유니버스 창이 활성화된 경우	이 옵션을 선택한 경우 클래스 이름, 개체 이름 또는 미리 정의된 조건 이름만 검색됩니다. 이 옵션을 해제하면 클래스, 개체 또는 미리 정의된 조건 이름은 검색 대상에 포함되지 않습니다.

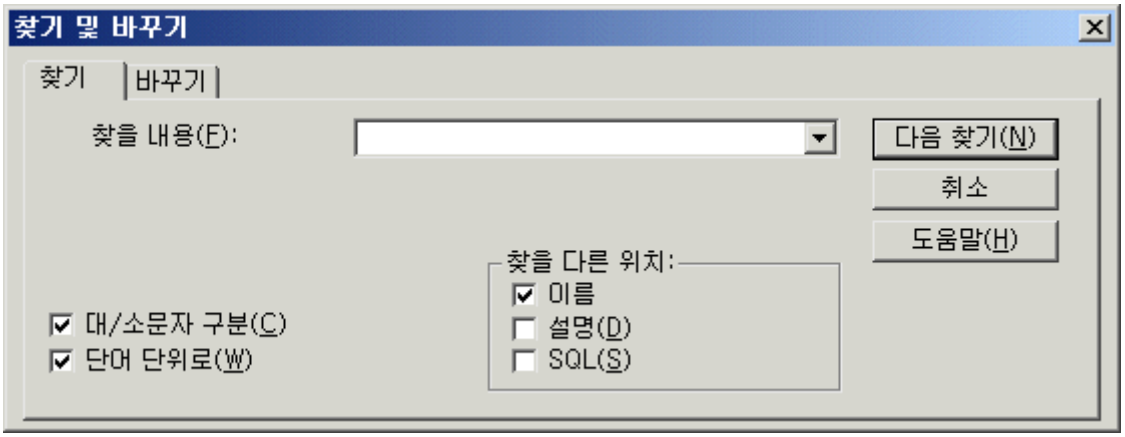
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

옵션	사용할 수 있는 옵션	설명
설명에서도 조회	유니버스 창이 활성화된 경우	이 옵션을 선택하면 유니버스 구조의 모든 설명이 검색 대상에 포함됩니다.
SQL 에서도 조회	유니버스 창이 활성화된 경우	이 옵션을 선택하면 개체, 조인 및 기타 유니버스 구조에 대한 SQL 정의가 검색 대상에 포함됩니다.

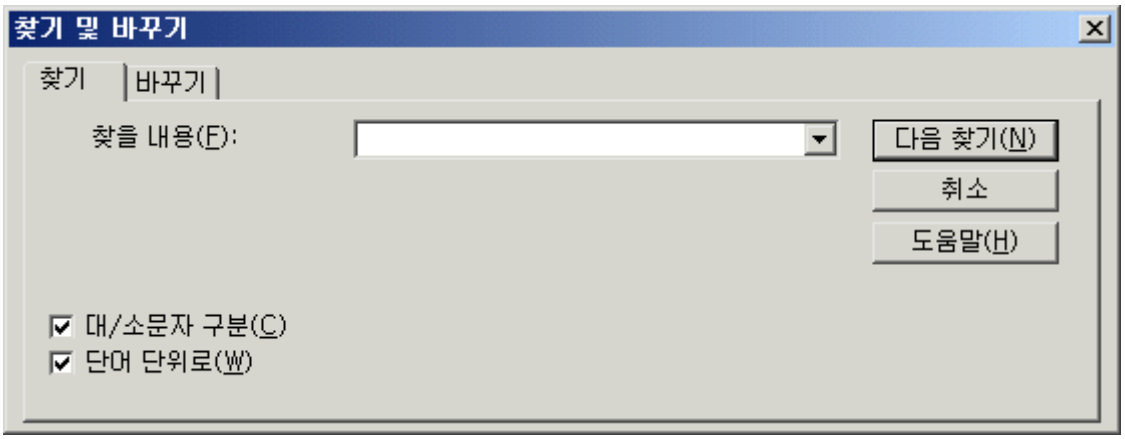
2.12.1.2 유니버스에서 검색

유니버스에서 검색하려면

1. 유니버스 또는 구조 창을 클릭합니다.
   이 창에서 문자열을 찾습니다.
2. 편집 > 찾기 를 선택합니다.
   찾기/바꾸기 대화 상자가 나타납니다. 활성화된 유니버스 창에 대한 대화 상자는 아래와 같습니다.



활성화된 구조 창에 대한 대화 상자는 아래와 같습니다.



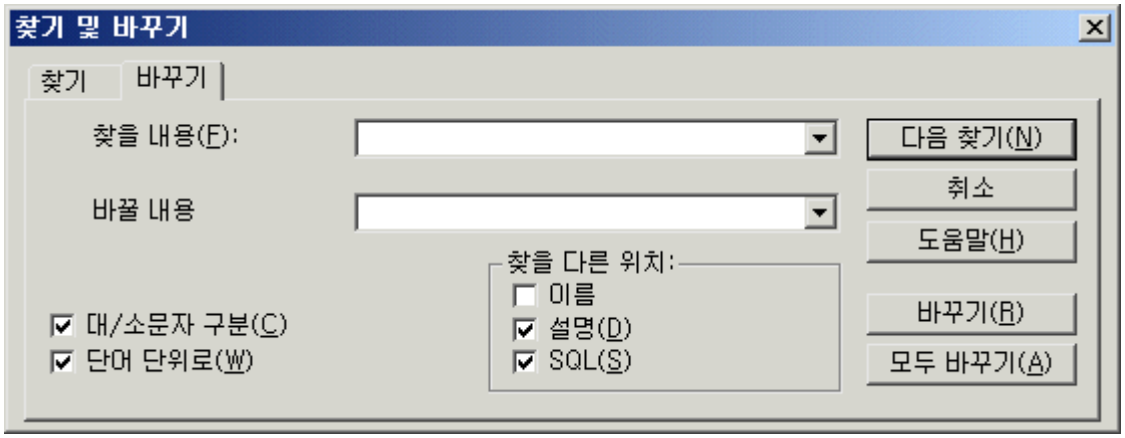
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. 찾을 내용 텍스트 상자에 문자 또는 문자열을 입력합니다.
4. 검색 옵션 확인란을 선택하거나 해제합니다.
5. 다음 찾기를 클릭합니다.
   유니버스 창에서 문자나 문자열을 찾으면 개체가 강조 표시됩니다. SQL 정의 또는 개체 설명에서 인스턴스를 찾으면 개체 속성 대화 상자가 자동으로 열리고 문자 또는 문자열이 강조 표시됩니다.
6. 검색 문자열과 일치하는 다른 인스턴스를 검색하려면 다음 찾기를 클릭합니다.
7. 찾기/바꾸기 대화 상자를 닫으려면 취소를 클릭합니다.

2.12.1.3 유니버스에서 찾기 및 바꾸기

유니버스에서 문자 또는 문자열을 찾고 바꾸려면

1. 편집 > 다음 바꾸기 를 선택합니다.
   찾기/바꾸기 대화 상자가 나타납니다.
2. 찾을 내용 텍스트 상자에 문자 또는 문자열을 입력합니다.



3. 바꾸기 텍스트 상자에 문자 또는 문자열을 입력합니다. 찾을 내용 텍스트 상자에 입력한 내용과 일치하는 인스턴스는 여기 입력한 텍스트 항목으로 바뀝니다.
4. 검색 옵션 확인란을 선택하거나 해제합니다.
5. 인스턴스가 발견될 때마다 텍스트 항목을 바꾸려면 바꾸기를 클릭합니다.
   또는
   유니버스의 모든 인스턴스를 자동으로 바꾸려면 모두 바꾸기를 클릭합니다.
   찾은 항목을 개별적으로 바꾸는 경우 개체 설명에서 항목이 발견되면 개체 속성 대화 상자가 자동으로 열리고 이 대화 상자가 활성화됩니다. 검색을 계속 진행하려면 찾기/바꾸기 상자를 클릭해야 합니다.

2.12.2 빠른 찾기 사용

유니버스 창 아래쪽에 있는 검색 상자에 검색 문자열의 첫 번째 문자를 입력하여 활성 창을 검색할 수 있습니다.

유니버스 창이 활성화되어 있는 경우 클래스와 개체 이름에 대한 검색이 수행됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

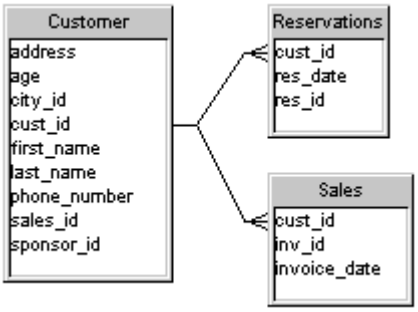
구조 창이 활성화되어 있는 경우 테이블 이름에 대한 검색이 수행됩니다.

2.13 테이블 디스플레이 구성

이 단원에서는 구조 창에서 테이블을 구성하고 조작하는 데 사용할 수 있는 그래픽 기능에 대해 설명합니다. 스키마를 디자인하는데 사용되는 디자인 방법과 구조 창에서 스키마를 성공적으로 만들기 위해 알아야 할 사항에 대한 자세한 내용은 테이블과 조인을 사용하여 스키마 만들기 [페이지 120] 장을 참조하십시오.

2.13.1 테이블 표현 방식

구조 창에서 테이블은 사각형 기호로 표시됩니다. 테이블의 이름은 사각형의 위쪽 부분에 있는 구획 안에 표시됩니다. 사각형 안에 있는 항목 목록은 테이블의 열을 나타냅니다. 테이블을 연결하는 선은 조인입니다.



2.13.2 테이블 조작

구조 창에서 다음과 같은 작업을 수행하여 테이블을 조작할 수 있습니다.

2.13.2.1 테이블 선택

다음과 같이 테이블을 선택할 수 있습니다.

표 14:

선택 대상	방법
테이블 하나	테이블을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

선택 대상	방법
여러 테이블	• 마우스 왼쪽 단추를 누른 채로 끌어 테이블 주위에 선택 영역을 그립니다. • Shift 키를 누른 채로 여러 테이블을 클릭합니다.
모든 테이블을 한 번에	편집 > 모두 선택 을 선택합니다.

선택을 취소하려면 포인터를 테이블 바깥쪽에 놓고 마우스 단추를 다시 클릭합니다.

2.13.2.2 테이블 삭제

테이블을 삭제하려면

1. 테이블을 선택합니다.
2. 다음 작업 중 하나를 수행합니다.
   - 표준 도구 모음에서 잘라내기 단추를 클릭합니다.
   - 편집 > 잘라내기 를 선택합니다.
   - Delete 키를 누릅니다.

2.13.3 목록 모드 사용

목록 모드를 사용하여 활성 유니버스에 사용되는 테이블, 조인 및 컨텍스트를 나열할 수 있습니다. 유니버스 디자인 도구의 목록 모드에서는 구조 창 위에 세 개의 창이 추가됩니다. 이러한 창의 이름은 테이블, 조인, 컨텍스트입니다.

다음과 같은 방식으로 목록 모드를 사용할 수 있습니다.

표 15:

작업	결과
목록 모드 창에 나열된 구성 요소를 클릭합니다.	구조 창에서 구성 요소가 강조 표시됩니다.
구조 창에서 테이블, 조인 또는 컨텍스트를 선택합니다.	목록 창에 나열된 구성 요소가 강조 표시됩니다.
테이블 창에서 테이블 이름을 두 번 클릭합니다.	테이블 이름 바꾸기 대화 상자가 열립니다. 테이블 이름을 바꿀 수 있고 데이터베이스에 따라 테이블 소유자와 한정자를 편집할 수 있습니다.
조인 창에서 조인 이름을 두 번 클릭합니다.	조인에 대한 조인 편집 대화 상자가 나타납니다. 조인 속성을 편집할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

작업	결과
컨텍스트 창에서 컨텍스트 이름을 두 번 클릭합니다.	컨텍스트 편집 대화 상자가 나타납니다. `Ctrl` 키를 누른 채로 목록에서 조인을 클릭하면 선택된 컨텍스트에 조인을 추가할 수 있습니다.
구성 요소를 클릭한 다음 두 목록 창 사이의 삼각형을 클릭합니다.	원래 구성 요소에 관련된 구성 요소가 인접한 목록 창에 표시됩니다. 관련이 없는 구성 요소는 모두 필터링을 통해 제거됩니다.
목록 창과 구조 창 사이의 구분선을 클릭한 다음 위쪽 또는 아래쪽으로 끌어 옵니다.	구분선을 끄는 방향에 따라 목록 창의 크기가 늘어나거나 줄어듭니다.

2.13.3.1 창 사이의 삼각형을 사용하여 나열된 구성 요소 필터링

창 사이에 표시되는 작은 삼각형은 구성 요소 표시에 대한 필터 역할을 수행합니다. 예:

- 테이블 창에서 테이블 이름을 클릭한 다음 조인 창을 가리키는 삼각형을 클릭합니다. 그러면 조인 창에는 선택된 테이블의 조인만 표시됩니다.
- 조인 창에서 조인 이름을 클릭한 다음 테이블 창을 가리키는 삼각형을 클릭합니다. 그러면 조인으로 연결된 테이블만 테이블 창에 표시됩니다.

2.13.3.2 목록 모드에서 일반 보기로 돌아가기

다음과 같은 두 가지 방법으로 목록 보기를 취소하고 일반 보기로 돌아갈 수 있습니다.

- 목록 모드에서 보기 > 목록 모드 를 선택합니다.
- 목록 모드에서 목록 모드 단추를 클릭합니다.

2.13.4 테이블 자동 정렬

다음 두 가지 방법으로 구조 창에서 자동으로 테이블을 정렬할 수 있습니다.

- 보기 > 테이블 정렬 을 선택합니다.
- 정렬 단추를 클릭합니다.

2.13.5 테이블 디스플레이 변경

서로 다른 세 가지 보기로 테이블을 표시할 수 있습니다. 각 유형의 보기는 테이블 기호에 표시되는 정보의 양에 대한 필터 역할을 합니다.

각 보기에 대한 설명은 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 16:

테이블 보기	설명
기본값	각 테이블이 최대 8 개의 열까지 표시됩니다. 이 값은 수정할 수 있습니다. 자세한 내용은 스키마 표시 옵션 선택 [페이지 58] 단원을 참조하십시오.
이름만 보기	테이블 기호에 테이블 이름만 표시됩니다. 이렇게 하면 테이블 수가 많을 때 구조 창에서 테이블이 잘리는 경우를 줄일 수 있습니다.
조인 열	테이블 사이의 조인에 관련된 열만 각 테이블 기호에 표시됩니다. 이는 일반적으로 키 열입니다.

각 테이블 보기는 다음과 같습니다.

2.13.5.1 기본 테이블 보기

아래에는 처음 8 개의 열이 포함된 테이블 기호가 나와 있습니다.

테이블에 기본값보다 많은 열이 있는 경우 마지막 열 뒤에 줄임표(...)가 표시됩니다. 테이블을 한 번 클릭하면 스크롤 막대가 나타납니다. 테이블의 아래쪽 테두리를 아래로 끌어서 테이블의 크기를 늘릴 수 있습니다.

2.13.5.2 테이블 이름만 보기

다음과 같이 테이블 기호에 테이블 이름만 표시할 수 있습니다.

- 테이블을 두 번 클릭합니다.

테이블의 이름만 표시됩니다.

2.13.5.3 조인 열 테이블 보기

다음과 같이 테이블 기호에 조인 열만 표시할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 구조에서 이미 이름 전용 뷰에 포함된 테이블을 두 번 클릭합니다. 테이블에는 조인 열만 표시됩니다.

2.13.5.4 모든 테이블의 표시 변경

선택된 모든 테이블의 보기를 동시에 변경하려면

- 보기 > 테이블 표시 변경 을 선택합니다.

2.14 스키마 표시 옵션 선택

구조 창에서 테이블, 열, 조인 및 카디널리티의 모양을 사용자 지정할 수 있습니다.

구조 창에 구성 요소를 표시할 때 다음과 같은 그래픽 옵션을 사용할 수 있습니다.

표 17:

옵션	설명
조인 모양	조인은 서로 다른 형식의 간단한 선으로 표현하거나 까마귀 발톱 또는 카디널리티 비율 같은 카디널리티 표시기가 포함된 선으로 나타낼 수 있습니다.
최적 측면	두 테이블을 연결하도록 선택된 조인은 한 테이블의 왼쪽 또는 오른쪽에서 다른 테이블의 왼쪽 또는 오른쪽으로 최단 거리로 끝나도록 잘 표시되는 측면이 자동으로 평가됩니다.
테이블	테이블에는 3 차원 효과를 줄 수 있으며 별칭 이름을 표시하거나 행 수를 표시할 수 있습니다. 각 테이블의 행 수를 표시하려면 보기 > 테이블의 행 수를 선택하여 행 수를 다시 계산해야 할 수도 있습니다. 자세한 내용은 반환되는 행의 수에 대한 기본값 수정 [페이지 61] 단원에서 설명합니다.
열	열 옆에 열 데이터 형식을 표시할 수 있습니다. 키 열에 밑줄을 표시하거나 열을 테이블 기호의 왼쪽 또는 가운데에 정렬하여 표시할 수도 있습니다.
기본 열 수	테이블 기호에 표시되는 열의 수에 대한 기본값을 입력할 수 있습니다. 테이블에 기본값보다 많은 열이 있으면 테이블 기호에서 열 목록의 끝에 줄임표(...)가 표시됩니다. 테이블을 한 번 클릭하면 테이블의 측면에 스크롤 막대가 나타납니다.
선택 중심	구조 창의 보기가 계산된 중심점을 기준으로 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.14.1 구조 창 표시 그래픽 옵션 설정

다음과 같이 구조 창의 구성 요소에 대한 그래픽 옵션을 설정할 수 있습니다.

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자가 나타납니다.
2. 그래픽 탭을 클릭합니다.
   그래픽 페이지가 나타납니다. 이 페이지에는 구조 창의 구성 요소에 대한 그래픽 옵션이 나열됩니다.
3. 그래픽 표시 옵션을 선택하거나 입력합니다.
4. 확인을 클릭합니다.

2.14.1.1 그래픽 옵션 예제

다음은 옵션 대화 상자에서 제공되는 그래픽 옵션(도구 > 옵션 > 그래픽)을 사용하여 구조 창의 구성 요소를 나타낼 수 있는 그래픽 표현의 몇 가지 예제입니다.

별칭 이름

구조 창에서 별칭 테이블을 선택하면 별칭 테이블의 이름이 표시되고 이를 파생시킨 테이블의 이름이 괄호 안에 표시됩니다.

행 수 표시 및 형식 표시

행 수 표시를 선택하면 각 테이블 기호의 아래쪽에 각 테이블의 행 수가 표시됩니다. 행 수를 표시하기 전에 보기 > 테이블 행 수 를 선택하여 모든 테이블의 행 수를 새로 고쳐야 합니다.

서식 표시를 선택하면 열 형식을 나타내는 문자가 열 이름 옆에 표시됩니다. 열 형식은 다음과 같습니다.

- C 는 문자를 나타냅니다.
- D 는 날짜를 나타냅니다.
- N 은 숫자를 나타냅니다.
- T 는 긴 텍스트를 나타냅니다.
- L 은 BLOB(Binary Large OBject)를 나타냅니다.

구조 창에서 숫자는 테이블 왼쪽 아래 가장자리 밑에 표시되어 있고 데이터 유형은 열 이름 옆에 표시되어 있습니다.

2.14.2 테이블 및 열 값 보기

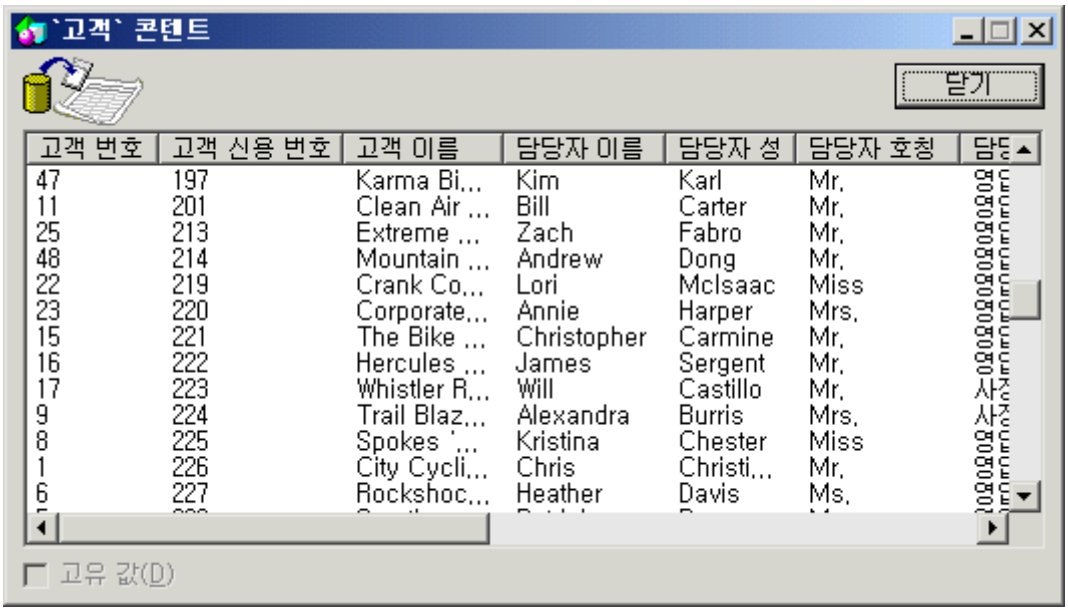
특정 테이블이나 열의 데이터 값을 볼 수 있습니다. 테이블에 대해 볼 수 있는 행 수는 기본적으로 100 개입니다. 필요에 따라 행이 더 많이 또는 더 적게 반환되도록 이 값을 변경할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2.14.2.1 테이블의 값 보기

테이블의 값을 보려면

1. 구조 창에서 테이블을 클릭합니다.
2. 보기 > 테이블 값 을 선택합니다.
   테이블의 내용을 표시하는 대화 상자가 나타나고 테이블의 각 열에 대한 값이 나열됩니다.



3. 고유 값만 표시하려면 고유 값 확인란을 선택합니다.
4. 닫기를 클릭합니다.

2.14.2.2 열의 값 보기

열 값을 볼 때 보기 > 확대 를 선택하여 열의 보기를 확대할 수 있습니다. 이렇게 하면 열을 더 쉽게 선택할 수 있습니다.

다음과 같이 개별 열의 값을 볼 수 있습니다.

1. 구조 창에서 테이블 열 위에 마우스 포인터를 놓습니다.
   포인터가 손 모양의 기호로 바뀝니다.
2. 열을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 열 값 보기를 선택합니다.
   열의 내용을 표시하는 대화 상자가 나타나고 열 값이 나열됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. 고유 값만 표시하려면 고유 값 확인란을 선택합니다.
4. 닫기를 클릭합니다.

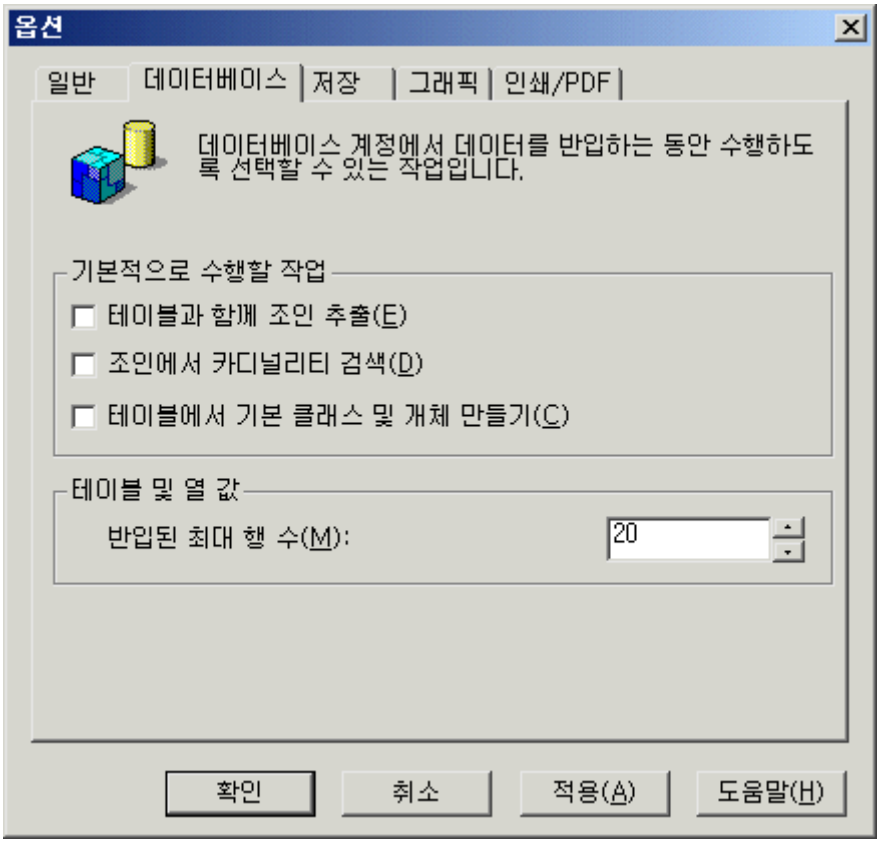
2.14.2.3 반환되는 행의 수에 대한 기본값 수정

테이블 또는 열 값을 볼 때 반환되는 행의 수에 대한 기본값을 수정할 수 있습니다. 이렇게 하면 반환되는 값의 수를 줄일 수 있으므로 이 기능은 테이블에서 값의 일부 샘플만 확인하려는 경우에 유용합니다.

테이블에 대해 반입되는 행의 수를 수정하려면

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자가 나타납니다.
2. 데이터베이스 탭을 클릭합니다.
   데이터베이스 페이지가 나타납니다.
3. 테이블 및 열 값 목록 상자의 위쪽 및 아래쪽 화살표를 사용하여 숫자를 선택하거나 직접 입력합니다.
   아래의 데이터베이스 페이지에서는 테이블 또는 열의 값을 볼 때 20 개의 행이 반환되도록 지정되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



4. 확인을 클릭합니다.

2.14.3 데이터베이스 테이블의 행 수 보기

각 테이블의 행 수를 표시할 수 있습니다. 이 작업은 다음 두 단계로 수행할 수 있습니다.

- 그래픽 옵션 행 수 표시(▶ 도구 > 옵션 > 그래픽 ▶)를 활성화합니다.
- ▶ 보기 > 테이블 행 수 ▶를 선택하여 모든 테이블의 행 수를 새로 고칩니다.

데이터베이스의 각 테이블에 있는 행의 수를 표시하거나 선택된 테이블에 대한 고정된 행 수를 설정하여 쿼리 성능을 최적화할 수 있습니다. 이렇게 하면 FROM 절에서 테이블 가중치를 기준으로 테이블 순서를 제어할 수 있습니다. 자세한 내용은 테이블의 행 수 수정 [페이지 64] 단원에서 설명합니다.

> **i 노트**
>
> 테이블의 행 수를 표시하는 작업은 테이블 또는 열 값 보기에 반환되는 행 수를 설정하는 것과는 다릅니다.

2.14.3.1 테이블의 행 수 표시

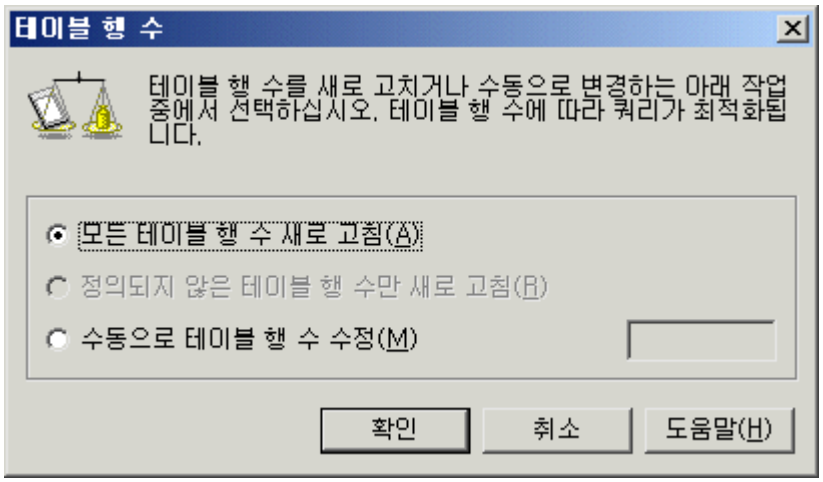
각 테이블의 행 수를 표시하려면

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자가 나타납니다.
2. 그래픽 탭을 클릭합니다.
   그래픽 페이지가 나타납니다.
3. 행 수 표시 확인란을 선택합니다.
4. 확인을 클릭합니다.
5. 하나 이상의 테이블을 선택합니다.
   또는
   구조 창의 아무 곳이나 클릭하고 편집 > 모두 선택 을 선택하여 구조 창의 테이블을 모두 선택합니다.

> **노트**
>
> 구조 창을 클릭하면 구조 창의 구성 요소와 관련된 메뉴 항목이 활성화됩니다. 메뉴 항목을 선택하기 전에 구조 창을 클릭하지 않으면 유니버스 창에 적용되는 메뉴 항목만 사용할 수 있습니다.

6. 보기 > 테이블 행 수 를 선택합니다.
   테이블 행 수 대화 상자가 나타납니다.



이 대화 상자의 옵션에 대한 설명은 다음과 같습니다.

표 18:

옵션	설명
모든 테이블의 행 수 새로 고침	구조 창의 모든 테이블 또는 선택된 테이블에 대한 행 수의 표시를 새로 고칩니다.
정의되지 않은 행 수만 새로 고침	이전에 선택하지 않은 테이블의 행 수를 표시합니다. 그 결과로 구조 창에서 모든 테이블이 행 수와 함께 표시됩니다.
테이블 행 수 직접 수정	구조 창의 모든 테이블 또는 선택된 테이블에 대한 행 수를 사용자가 직접 수정할 수 있습니다. 옵션 옆에 있는 텍스트 상자에 새 값을 입력합니다. 이 옵션은 쿼리를 최적화하는 데 사용됩니다. 자세한 내용은 다음 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7. 모든 테이블 행 수 새로 고침 라디오 단추를 선택합니다.
8. 확인을 클릭합니다.
   선택된 각 테이블의 행 수가 구조 창에 있는 각 테이블 기호의 왼쪽 맨 아래에 표시됩니다.

2.14.3.2 테이블의 행 수 수정

테이블의 행 수를 수정할 수 있습니다. 이 작업은 다음과 같은 두 가지 목적에 사용됩니다.

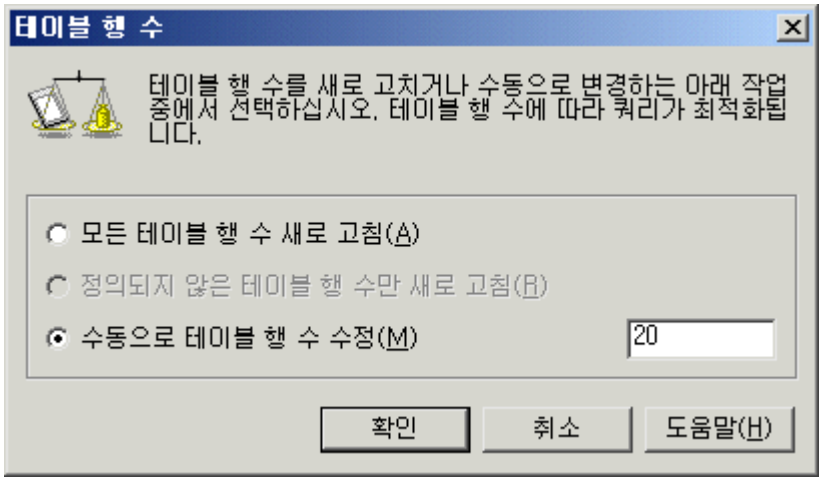
표 19:

행 수를 수정하는 목적	설명
쿼리 최적화	쿼리 최적화는 생성된 SQL 의 FROM 절에서 테이블의 순서를 기반으로 합니다. 행 수가 많은 테이블은 행 수가 적은 테이블보다 앞에 표시됩니다. 이 순서는 RDBMS 에 최적화 기능이 없는 경우에 특히 중요할 수 있습니다. 테이블의 행 수를 수정하면 FROM 절에서 테이블의 순서를 변경할 수 있습니다.
이후의 데이터 용량 변경에 행 수 맞추기	테이블에 저장될 행의 수가 테이블 행 수에 제대로 반영되지 않은 경우 테이블 행 수를 수정할 수 있습니다. 예를 들어, 작업 중인 테스트 테이블의 현재 행 수는 100 개지만 실제로는 행을 50,000 개까지 포함할 수 있는 경우가 있습니다.

하나 이상의 테이블에 대해 행 수를 수정하려면

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자가 나타납니다.
2. 그래픽 탭을 클릭합니다.
   그래픽 페이지가 나타납니다.
3. 행 수 표시 확인란을 선택합니다.
4. 확인을 클릭합니다.
5. 하나 이상의 테이블을 선택합니다.
   또는
   구조 창의 아무 곳이나 클릭하고 편집 > 모두 선택 을 선택하여 구조 창의 테이블을 모두 선택합니다.
6. 보기 > 테이블 행 수 를 선택합니다.
   테이블 행 수 대화 상자가 나타납니다.
7. 수동으로 테이블 행 수 수정 라디오 단추를 선택합니다.
8. 테이블에 대해 표시하려는 행의 수를 입력합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



9. 확인을 클릭합니다.
   선택된 각 테이블의 행 수가 구조 창에 있는 각 테이블 기호의 왼쪽 맨 아래에 표시됩니다.

2.15 유니버스 인쇄

유니버스 디자인 도구는 표준 Windows 인쇄 기능을 모두 제공합니다. 구조 창의 테이블, 열 및 조인 목록을 비롯하여 스키마를 출력할 수 있습니다. 인쇄된 페이지에 구성 요소와 정보가 표시되는 방식을 제어할 수도 있습니다.

> **노트**
>
> 유니버스를 PDF 파일로 저장한 다음 PDF 파일을 인쇄하면 유니버스 정의와 스키마를 PDF 버전으로 인쇄할 수 있습니다. 자세한 내용은 유니버스 정의를 PDF 로 저장 [페이지 44] 단원을 참조하십시오.

2.15.1 인쇄 옵션 설정

옵션 대화 상자의 인쇄 페이지(▶ 도구 > 옵션 > 인쇄 ◀)에서 인쇄 옵션을 선택할 수 있습니다. 설정된 인쇄 옵션은 유니버스 정의를 PDF 로 저장할 때 PDF 파일에 대해 저장되는 옵션에도 적용됩니다. 선택할 수 있는 인쇄 및 PDF 옵션은 다음과 같습니다.

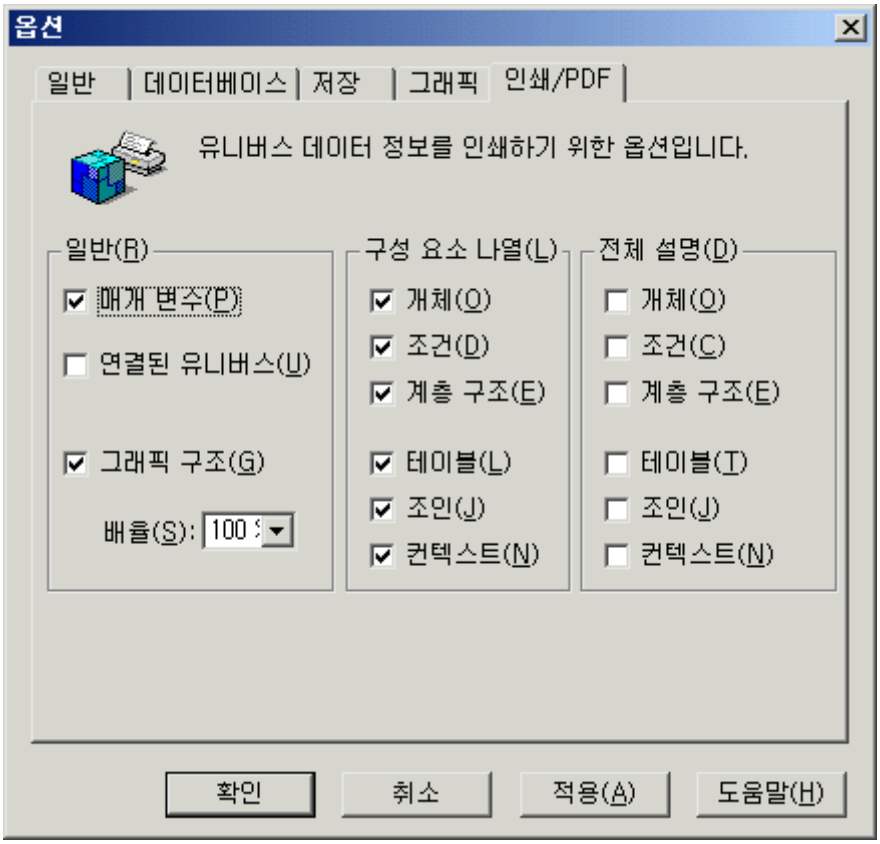
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 20:

인쇄 옵션	출력 내용
일반 정보	다음과 같은 정보가 출력됩니다. • 유니버스 매개 변수 • 연결된 유니버스 구조 창의 스키마에 대한 그래픽 구조가 출력됩니다. 이 그래픽의 배율을 선택할 수 있습니다.
구성 요소 목록	개체, 조건, 계층, 테이블, 조인 및 컨텍스트 같은 하나 이상의 유형으로 그룹화된 유니버스의 구성 요소 목록
구성 요소 설명	개체, 조건, 계층, 테이블, 조인 및 컨텍스트 구성 요소에 대한 설명 설명에는 구성 요소의 속성에 대한 자세한 정보가 포함됩니다. 개체의 경우 이 정보에는 SQL 정의, 자격 및 보안 액세스 수준이 포함될 수 있습니다.

유니버스의 인쇄 옵션을 설정하려면

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자가 나타납니다.
2. 인쇄/*PDF* 탭을 클릭합니다.
   인쇄 페이지가 나타납니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. 필요한 인쇄 옵션 확인란을 선택합니다.
4. 확인을 클릭합니다.

2.15.1.1 페이지 설정 지정

페이지 설정 옵션을 지정하려면

1. 파일 > 페이지 설정 을 선택합니다.
   페이지 설정 시트가 나타납니다.
2. 페이지 설정 옵션을 선택하거나 입력합니다.
3. 확인을 클릭합니다.

2.15.1.2 인쇄 미리 보기 사용

인쇄하기 전에 다음 두 가지 방법으로 유니버스를 미리 볼 수 있습니다.

- 파일 > 인쇄 미리 보기 를 선택합니다.
- 인쇄 미리 보기 단추를 클릭합니다.

2.15.1.3 유니버스 인쇄

두 가지 방법으로 유니버스를 인쇄할 수 있습니다.

- 파일 > 인쇄 를 선택합니다.
- 인쇄 단추를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3 유니버스 만들기 및 유니버스 매개 변수 설정

유니버스를 작성하려면 먼저 새 유니버스 파일을 만들어야 합니다.

새 유니버스 파일을 만들려면 유니버스에서 데이터베이스 미들웨어에 액세스할 수 있도록 연결 매개 변수를 정의해야 합니다. 유니버스 디자인 도구의 개체 생성, 유니버스 간 연결 및 쿼리 제한 방식을 결정하는 다른 매개 변수를 정의할 수도 있습니다.

새 유니버스는 `.unv` 파일로 저장됩니다. 새 유니버스에는 클래스와 개체가 포함되어 있지 않습니다. 테이블 스키마를 디자인한 다음 데이터베이스 구조에 매핑되는 개체를 만들어 유니버스 개발 과정에서 이러한 항목을 만듭니다.

3.1 유니버스 매개 변수 개요

유니버스 매개 변수는 유니버스와 데이터베이스 연결을 식별하고 유니버스를 사용하여 실행할 수 있는 쿼리 유형을 지정하고 시스템 리소스에 대한 사용을 제어하기 위해 유니버스에 대해 지정하는 정의 및 제한 사항입니다.

유니버스를 만들 때 유니버스 매개 변수 대화 상자(파일 > 매개 변수)에서 유니버스 매개 변수를 정의합니다. 새 유니버스를 만들 때 사용자가 직접 선택하거나 생성해야 하는 유일한 항목은 데이터베이스 연결 매개 변수입니다.

이러한 매개 변수는 언제든지 수정할 수 있습니다. 사용자가 정의할 수 있는 유니버스 매개 변수는 다음과 같습니다.

표 21:

매개 변수	설명
정의	유니버스 이름, 설명 및 연결 매개 변수와 정보입니다. 이러한 매개 변수는 유니버스를 식별하는 데 사용됩니다. 이 매개 변수 정의 및 수정에 대한 내용은 유니버스 식별 [페이지 71] 단원을 참조하십시오.
요약 정보	버전 및 수정 정보, 디자이너 주석 및 유니버스 통계입니다. 이 매개 변수 정의 및 수정에 대한 내용은 요약 정보 보기 및 입력 [페이지 70] 단원을 참조하십시오.
전략	유니버스에 사용되는 전략을 나타냅니다. 전략은 데이터베이스에서 구조 정보를 추출하는 데 사용되는 스크립트입니다. 이 매개 변수 정의 및 수정에 대한 내용은 전략 선택 [페이지 81] 단원을 참조하십시오.
제어	시스템 리소스를 사용하는 데 적용되는 제한 사항 집합을 나타냅니다. 이 매개 변수 정의 및 수정에 대한 내용은 리소스 제어 지정 [페이지 84] 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

매개 변수	설명
SQL	최종 사용자가 쿼리 창에서 실행할 수 있는 쿼리의 유형을 나타냅니다. 이 매개 변수 정의 및 수정에 대한 내용은 리소스 제어 지정 [페이지 84] 단원을 참조하십시오.
링크	연결된 유니버스에 대해 정의된 설정을 나타냅니다. 이 매개 변수 정의 및 수정에 대한 내용은 SQL 제한 옵션 입력 [페이지 87] 단원을 참조하십시오.

3.2 새 유니버스 만들기

다음 절차에서는 유니버스 매개 변수를 정의한 다음 유니버스를 저장하는 방식으로 유니버스를 처음부터 새로 만드는 방법을 설명합니다. 이 절차에서는 매개 변수 대화 상자에서 사용할 수 있는 모든 페이지에 대한 개요를 제공합니다.

각 단계에 대한 자세한 내용은 이 장에서 매개 변수에 대한 단원을 참조하십시오.

유니버스를 만들 때 모든 매개 변수를 정의할 필요는 없습니다. 연결은 반드시 선택해야 하지만 다른 매개 변수에 대해서는 기본값을 적용한 다음 필요할 때 언제든지 적절하게 수정할 수 있습니다.

3.2.1 처음부터 새로 유니버스 만들기

유니버스를 처음부터 새로 만들려면

1. 파일 > 새로 만들기 를 선택합니다.
   유니버스 매개 변수 대화 상자의 정의 페이지가 열립니다. 이 페이지에 대한 내용은 유니버스 식별 [페이지 71] 단원을 참조하십시오.

   > **노트**
   >
   > 유니버스 매개 변수에 대한 옵션을 선택할 경우 옵션 저장 프로시저 유니버스를 선택하려면 여기를 클릭하십시오는 회색으로 표시됩니다. 따라서 선택하거나 선택 취소할 수 없습니다. 만들려는 유니버스의 유형을 변경하려면 취소를 클릭하고 다시 시작하십시오.

   - 유니버스의 이름과 설명을 입력합니다.
   - 연결 드롭다운 목록 상자에서 연결을 선택합니다.

   또는

   - 드롭다운 목록에 나열되어 있지 않은 새 연결을 정의하려면 새로 만들기 단추를 클릭합니다. 새 연결 정의에 대한 내용은 유니버스 식별 매개 변수 수정 [페이지 72] 단원을 참조하십시오.
2. 요약 탭을 클릭합니다.
   요약 페이지가 나타납니다. 이 페이지에 대한 내용은 요약 정보 보기 및 입력 [페이지 70] 단원을 참조하십시오.
   - 주석 상자에 유니버스 정보를 입력합니다.
3. 전략 탭을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

전략 페이지가 나타납니다. 이 페이지에는 연결된 데이터 소스에 사용할 수 있는 전략이 표시됩니다. 이 페이지에 대한 내용은 전략 선택 [페이지 81] 단원을 참조하십시오.
   - 개체, 조인 및 테이블 드롭다운 목록 상자에서 전략을 선택합니다.

   연결에 대한 RDBMS 에 따라서는 각 드롭다운 목록 상자에서 여러 개의 전략이 제공될 수도 있습니다.
4. 제어 탭을 클릭합니다.
   제어 페이지가 나타납니다. 이 페이지에 대한 내용은 리소스 제어 지정 [페이지 84] 단원을 참조하십시오.
   - 쿼리 제한 그룹 상자의 각 확인란을 선택하거나 해제합니다.
   - 선택한 확인란에 대해 값을 입력합니다.
5. *SQL* 탭을 클릭합니다.
   SQL 페이지가 나타납니다. 이 페이지에 대한 내용은 SQL 제한 지정 [페이지 86]을 참조하십시오.
   - 필요에 따라 확인란을 선택하거나 해제합니다.
6. 기존의 유니버스에 새 유니버스를 연결하려면 링크 탭을 클릭합니다.
   연결 페이지가 나타납니다. 이 페이지에 대한 내용은 SQL 제한 옵션 입력 [페이지 87] 단원을 참조하십시오.
   - 링크 추가 단추를 클릭하고 새 유니버스를 연결할 유니버스를 선택합니다.
7. 매개 변수 탭을 클릭합니다.
   매개 변수 페이지가 나타납니다. 이 페이지에는 SQL 생성을 최적화하기 위해 설정할 수 있는 SQL 매개 변수가 나열됩니다. 이 페이지에 대한 내용은 SQL 생성 매개 변수 설정 [페이지 88] 단원을 참조하십시오.
8. 확인을 클릭합니다.
   유니버스 디자인 도구에서 유니버스 및 구조 창이 열립니다.
9. 파일 > 저장 을 선택합니다.
   - 유니버스 파일의 이름을 입력합니다.
   - 저장을 클릭합니다.

3.3 요약 정보 보기 및 입력

요약 페이지에는 유니버스 관리 정보가 표시됩니다. 이 정보는 활성 유니버스의 개발 과정을 추적하는 데 도움이 됩니다.

요약 페이지에는 다음과 같은 정보가 표시됩니다.

표 22:

정보	설명
작성	유니버스 작성 날짜와 작성자의 이름입니다.
수정	마지막 수정 날짜와 수정한 사용자의 이름입니다.
수정	유니버스를 CMS 로 내보낸 횟수를 나타내는 수정 번호입니다.
주석	자신과 다른 디자이너의 이해를 돕기 위한 유니버스 관련 정보입니다. 이 정보는 유니버스 디자인 도구에서만 제공됩니다. 확인 페이지의 설명 필드에 사용자를 위한 유니버스 관련 정보를 포함시켜야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

정보	설명
통계	유니버스에 포함된 클래스, 개체, 테이블, 별칭, 조인, 컨텍스트 및 계층의 수가 나열됩니다.

3.4 유니버스 매개 변수 설정

다음과 같은 작업을 수행하기 위한 유니버스 매개 변수를 설정할 수 있습니다.

- 유니버스 식별 [페이지 71]
- 유니버스 식별 매개 변수 수정 [페이지 72]
- 요약 정보 보기 및 입력 [페이지 70]
- 전략 선택 [페이지 81]
- SQL 제한 지정 [페이지 86]
- SQL 제한 옵션 입력 [페이지 87]
- SQL 생성 매개 변수 설정 [페이지 88]

각 유형의 매개 변수는 매개 변수 대화 상자(파일 > 매개 변수)의 페이지별로 제공됩니다. 각 매개 변수 그룹에 대한 설명은 아래의 해당 단원에 나와 있습니다.

3.4.1 유니버스 식별

각 유니버스는 다음 매개 변수를 통해 식별됩니다.

표 23:

식별자	사용 주체
파일 이름 (8 자)	파일 시스템 및 Web Intelligence 에서 유니버스를 참조하는 데 사용
긴 이름 (35 자)	Web Intelligence 사용자
설명	Web Intelligence 사용자
고유한 숫자 ID	CMS 에서 유니버스를 식별하는 데 사용하는 번호로서, 유니버스를 CMS 에 처음 내보낼 때 할당됩니다.

이름 및 설명 매개 변수는 유니버스 매개 변수 대화 상자의 정의 페이지에서 유니버스를 만들 때 정의됩니다. 유니버스 식별 매개 변수는 언제든지 수정할 수 있습니다.

이 페이지에서 데이터베이스 연결을 정의할 수도 있습니다.

새 연결 정의에 대한 내용은 유니버스 식별 매개 변수 수정 [페이지 72] 단원을 참조하십시오.

유니버스에 대해 다음과 같은 식별 매개 변수를 정의할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 24:

식별 매개 변수	설명
이름	유니버스 이름입니다. Web Intelligence 사용자가 유니버스를 식별하는 데 사용합니다. 레지스트리에서 지원하는 이름 문자는 총감독자가 정의합니다. 문자 지원은 RDBMS 에 따라 다릅니다.
설명	유니버스 목적과 내용에 대한 설명입니다. 이 필드는 선택 사항입니다. 이 설명은 Web Intelligence 사용자가 볼 수 있으므로 이 필드의 정보는 유니버스의 역할에 대한 유용한 정보를 제공할 수 있도록 지정해야 합니다.
연결	Web Intelligence 에서 데이터베이스 파일의 데이터에 액세스하는 방법을 정의하는 매개 변수의 명명된 집합입니다. 연결 드롭다운 목록 상자에는 사용 가능한 모든 연결이 표시됩니다. 새 연결을 만들 수도 있습니다.

3.4.1.1 유니버스 식별 매개 변수 수정

유니버스 식별 매개 변수를 수정하려면

1. 파일 > 매개 변수 를 선택합니다.
   또는
   도구 모음에서 유니버스 매개 변수 단추를 클릭합니다.
   유니버스 매개 변수 대화 상자의 정의 페이지가 열립니다.
2. 이름과 설명을 입력합니다.
3. 연결 드롭다운 목록 상자에서 연결을 선택합니다.
4. 테스트를 클릭하여 연결이 올바른지 확인합니다.
   서버가 응답하지 않는다는 메시지가 표시되면 연결이 유효하지 않은 것입니다. 편집을 클릭하고 연결 속성을 편집하여 연결 매개 변수를 수정할 수 있습니다. 오류가 해결되지 않으면 RDBMS 설명서에서 오류 메시지에 관련된 단원을 참조하십시오.
5. 확인을 클릭합니다.

3.4.2 연결 정의 및 편집

연결은 Business Objects 응용 프로그램이 데이터베이스 파일의 데이터에 액세스하는 방식을 정의하는 명명된 매개 변수 집합입니다. 연결을 통해 Web Intelligence 가 미들웨어에 연결됩니다. 데이터에 액세스하려면 연결이 있어야 합니다.

유니버스를 만들 때 연결을 선택하거나 만들어야 합니다. 연결은 언제든지 수정, 삭제 또는 교체할 수 있습니다.

> **노트**
>
> 연결 개체에는 "로컬에 연결 다운로드"라는 관리자가 정의한 추가 보안 권한이 있습니다. 관리자는 (중요한) 연결 정보를 로컬로 다운로드할 수 있는 대상을 정의하는 등 연결과 관련된 보안을 정의해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

> **노트**
>
> 연결을 만들고 수정하고 최적화하는 방법에 대한 자세한 내용은 데이터 액세스 가이드를 참조하십시오.

다음과 같은 방법으로 연결을 관리할 수 있습니다.

연결 관리 방법	설명
연결 패널에서	도구 > 연결 메뉴의 명령을 통해 실행합니다. 이 패널에는 사용자가 로그온한 CMS 에서 액세스할 수 있는 연결 목록(개인, 공유, 보안)이 표시됩니다. 유니버스 디자인 도구가 독립 실행형 모드에서 시작된 경우에는 개인 및 공유 연결만 표시됩니다. 연결은 단순 목록으로 표시하거나 연결의 하위 폴더가 나타나는 계층구조로 표시할 수 있습니다. 이 페이지에서 새 연결을 삭제, 편집 및 작성할 수 있습니다.
유니버스 매개 변수 대화 상자에서	유니버스 매개 변수 대화 상자의 정의 페이지(파일 > 매개 변수 > 정의)에서 기존의 연결이 현재 유니버스에 적절하지 않은 경우 새 연결을 만듭니다. 연결 편집도 가능합니다.

연결에는 다음과 같은 요소가 포함됩니다.

- 데이터 액세스 드라이버
- 연결 및 로그인 매개 변수
- 연결 유형

각 요소에 대한 설명은 다음 단원에 나와 있습니다.

3.4.2.1 Connections 폴더 관리

유니버스 디자인 도구를 사용하여 Connections 폴더를 관리할 수 있습니다. 보안 연결이 표시되거나 관리되는 곳에서는 Connections 폴더가 표시됩니다. 다음과 같은 방법으로 연결을 관리할 수 있습니다.

연결 관리 방법	설명
연결 패널	도구 > 연결 메뉴의 명령을 통해 실행합니다. 이 패널에는 사용자가 로그온한 CMS 에서 액세스할 수 있는 연결 목록(개인, 공유, 보안)이 표시됩니다. 유니버스 디자인 도구가 독립 실행형 모드에서 시작된 경우에는 개인 및 공유 연결만 표시됩니다. 연결은 단순 목록으로 표시하거나 연결의 하위 폴더가 나타나는 계층구조로 표시할 수 있습니다. 이 페이지에서 새 연결을 삭제, 편집 및 작성할 수 있습니다. 연결 패널에서 실행되는 작업은 유효성이 확인되면 자동으로 CMS 에서 커밋됩니다.
유니버스 매개 변수 대화 상자	유니버스 매개 변수 대화 상자의 정의 페이지(파일 > 매개 변수 > 정의)에서 액세스합니다. 기존의 연결이 현재 유니버스에 적절하지 않은 경우 새 연결을 만듭니다. 연결 편집도 가능합니다.

수행할 수 있는 작업은 사용자 권한에 따라 다르지만, 사용 가능한 작업은 다음과 같습니다.

- 새 연결 만들기

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 새 Connections 폴더 만들기
- 연결 또는 폴더 편집
- 연결 또는 빈 폴더 삭제
- 연결 또는 폴더 이름 바꾸기
- 연결 설명 편집
- 연결 속성 확인
- 잘라내기, 복사, 붙여넣기(편집 모드)

3.4.2.2 데이터 액세스 드라이버

데이터 액세스 드라이버는 유니버스를 미들웨어에 연결하는 소프트웨어 계층입니다.

데이터 액세스 드라이버는 Business Objects 제품과 함께 제공됩니다. 지원되는 각 미들웨어별로 데이터 액세스 드라이버가 있습니다. 유니버스 디자인 도구를 설치하는 경우 데이터 액세스 키에 따라 설치되는 데이터 액세스 드라이버가 결정됩니다.

새 연결을 만들 때는 대상 RDBMS 에 연결하는 데 사용되는 RDBMS 미들웨어에 적합한 데이터 액세스 드라이버를 선택합니다.

3.4.2.3 연결 및 로그인 매개 변수

다음과 같은 연결 및 로그인 매개 변수를 지정하여 데이터 액세스 드라이버를 구성합니다.

표 25:

매개 변수	설명
유형	개인, 공유 또는 보안 연결 형식입니다.
이름	연결에 대한 이름을 식별합니다.
사용자 이름	데이터베이스 사용자 이름입니다. 일반적으로 데이터베이스 관리자가 사용자에게 지정합니다.
암호	데이터베이스 암호입니다. 일반적으로 데이터베이스 관리자가 사용자에게 지정합니다.
뷰 타임에 보고서를 새로 고칠 때 SSO(Single Sign-On) 사용	이 확인란을 선택하면 CMS 에 액세스하는 데 사용되는 사용자 이름 및 암호가 자동으로 데이터베이스 로그인 매개 변수로 사용됩니다. 단일 로그온 설정에 대한 내용은 BusinessObjects Enterprise 관리자 가이드를 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

매개 변수	설명
Business Objects 사용자 계정과 연관된 데이터베이스 자격 증명 사용	이 확인란을 선택하면 보고서를 새로 고치기 위해 사용자가 BusinessObjects 계정과 관련된 데이터베이스 사용자 암호를 입력해야 합니다. 이 옵션은 중앙 관리 콘솔 수준에서 설정됩니다. 이 옵션 설정에 대한 내용은 BusinessObjects Enterprise 관리자 가이드를 참조하십시오.
데이터 소스/서비스	데이터 소스 또는 데이터베이스 이름입니다. ODBC 드라이버를 사용하는 경우 데이터 소스 이름이 대상 데이터베이스를 식별합니다. 기본 드라이버를 사용하는 경우 데이터베이스 이름으로 대상 데이터베이스를 확인합니다.

노트

ODBC 를 통해 Excel 파일 및 .csv 형식의 텍스트 파일에 대한 연결을 만들 수 있습니다. ODCB 를 통해 액세스한 텍스트 파일 또는 Excel 파일 기반의 유니버스를 Web Intelligence 에서 사용할 수 있도록 하려면 해당 연결에 대한 msjet.prm 파일을 편집해야 합니다.

이 파일은 `$INSTALLDIR$/BusinessObjects Enterprise 12.0/win32_x86/dataAccess/connectionserver/odbc` 폴더에 있습니다. 여기서 `$INSTALLDIR$`은 Business Objects 응용 프로그램이 설치된 디렉터리입니다. `msjet.prm` 파일에서 `DB_TYPE` 매개 변수를 다음과 같이 변경합니다.

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

대상: `<Parameter Name='DB_TYPE'>MS Jet</Parameter>`

이와 같이 변경한 후에는 Business Objects Enterprise 서버를 중지했다가 다시 시작해야 합니다. 참고: Web Intelligence 서버와 동일한 컴퓨터에서 유니버스 디자인 도구를 실행하고 있는 경우, 이 값을 변경한 후 텍스트 또는 Excel 파일을 기반으로 추가 유니버스를 만들려면 값을 `<Parameter Name='DB_TYPE'>MS Jet Engine</Parameter>`로 재설정해야 합니다.

3.4.2.4 연결 유형

연결 유형에 따라 연결을 사용하여 데이터에 액세스할 수 있는 사용자가 결정됩니다. 유니버스 디자인 도구에서는 작업 세션 도중 만들어진 연결을 모두 자동으로 저장합니다. 다음에 세션을 시작할 때 이러한 연결을 사용할 수 있습니다.

유니버스 디자인 도구를 사용하여 다음과 같은 세 가지 유형의 연결을 만들 수 있습니다.

- 개인
- 공유
- 보안

다음은 각 연결 유형에 대한 설명입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

개인 연결

데이터에 대한 액세스를 유니버스 작성자 및 유니버스가 만들어진 컴퓨터로 제한합니다.

연결 매개 변수는 사용자 프로필 디렉터리의 Business Objects 12.0 폴더 아래의 LSI 폴더에 있는 PDAC.LSI 파일에 저장됩니다. 예를 들면 다음과 같습니다.

```
C:\Documents and Settings\<user name>\Application Data\Business Objects\Business
Objects 12.0\lsi
```

이러한 매개 변수는 정적이며 업데이트할 수 없습니다.

개인 연결은 Business Objects 제품 보안 측면에서 안전하지 않습니다.

개인 연결을 사용하여 유니버스를 배포하지 마십시오. 개인 연결을 사용하여 로컬 컴퓨터의 개인 데이터에 액세스할 수 있습니다.

공유 연결

모든 사용자가 데이터에 액세스할 수 있습니다. 이러한 연결은 Business Objects 제품 보안 측면에서 안전하지 않습니다.

연결 매개 변수는 사용자 프로필 디렉터리의 Business Objects 12.0 폴더 아래의 `lsi` 폴더에 있는 SDAC.LSI 파일에 저장됩니다. 예를 들면 다음과 같습니다.

```
C:\Documents and Settings\<user name>\Application Data\Business Objects\Business
Objects 12.0\lsi
```

보안 연결

- 데이터에 대한 액세스를 중앙화하고 통제합니다. 가장 안전한 연결 유형이며, 민감한 데이터에 대한 액세스를 보호하려면 이 연결 유형을 사용해야 합니다.
- 유니버스 디자인 도구를 사용하여 보안 연결을 만들 수 있습니다.
- CMS 를 통해 유니버스를 배포하려면 보안 연결을 사용해야 합니다.
- 보안 연결은 언제라도 사용 및 업데이트가 가능합니다.

3.4.2.5 개인 및 공유 연결에 암호 설정

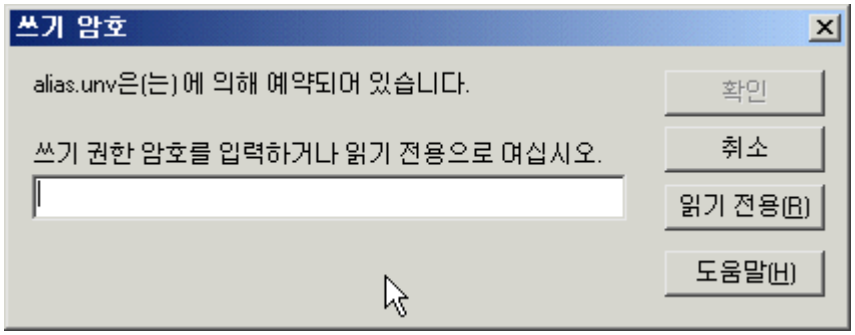
개인 또는 공유 연결 유형의 유니버스에 암호를 설정할 수 있습니다. 암호를 사용하면 리포지토리가 없는 환경에서 승인되지 않은 사용자가 유니버스에 접근할 수 없도록 보호할 수 있습니다.

> **노트**
>
> 암호를 잊어버린 경우에는 유니버스 파일을 복구할 수 없습니다. 따라서 유니버스 암호를 별도로 백업해 두는 것이 좋습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

암호를 설정하는 데는 두 가지 옵션을 사용할 수 있습니다.

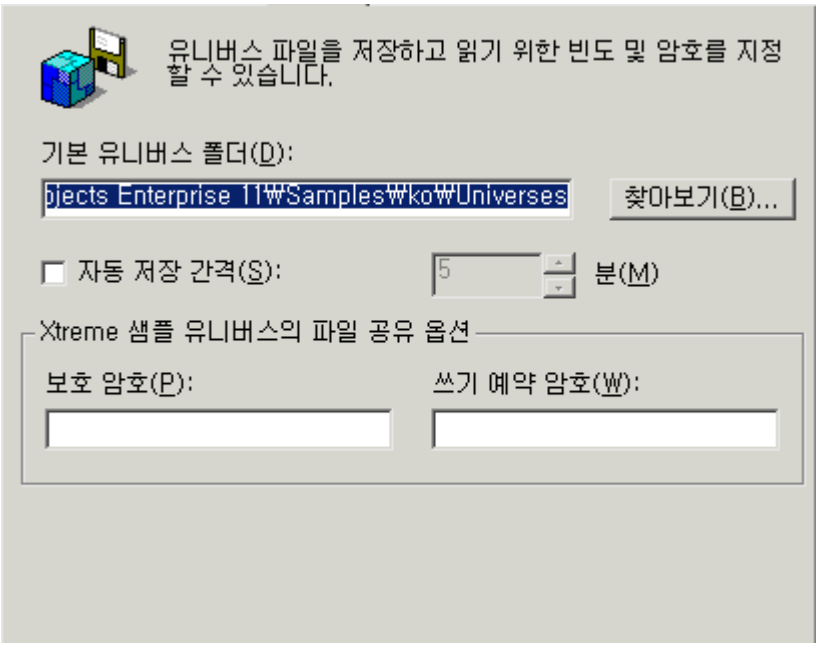
- 보호 암호를 선택하면 암호를 입력할 수 있는 간단한 대화 상자가 열립니다. 입력한 암호가 올바르면 유니버스가 열립니다.
- 쓰기 제한 암호를 선택하면 다음과 같은 대화 상자가 열립니다.



사용자는 읽기 전용 모드로 유니버스를 열 수 있고, 올바른 암호를 입력하면 읽기-쓰기 모드로 유니버스를 열 수 있습니다.

개인 또는 공유 연결을 사용할 때 암호를 설정하려면

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자가 나타납니다.
2. 저장 탭을 클릭합니다.
   저장 페이지가 나타납니다.



3. 보호 암호 또는 쓰기 예약 암호 텍스트 상자에 암호를 입력합니다. 암호는 영숫자로 최대 40 자까지 입력할 수 있습니다.
4. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.2.6 사용자 DBPass 가 변경된 후의 데이터베이스 액세스

BusinessObjects 관리자는 데이터베이스 사용자 암호가 변경된 후에 BusinessObjects 사용자 로그인(이름 및 암호)으로 데이터에 계속 액세스하도록 할 수 있습니다.

다음 매개 변수가 설정되면 데이터베이스 암호가 변경되어도 BusinessObjects 사용자가 BusinessObjects 관리자에게 문의할 필요 없이 계속해서 데이터에 액세스할 수 있습니다.

- 중앙 관리 콘솔에서 로그온 시 사용자의 데이터 소스 자격 증명 활성화 및 업데이트 확인란을 선택해야 합니다.
- 유니버스 디자인 도구의 새 연결 마법사에 있는 새 연결 정의 페이지에서 *BusinessObjects 사용자 계정과 연관된 데이터베이스 자격 증명 사용* 및 보는 도중 보고서를 새로 고칠 때 단일 로그온 사용 확인란을 선택해야 합니다.

위에서 언급한 확인란을 선택하면 업데이트된 `DBUser` 및 `DBPass` 매개 변수가 BusinessObjects 사용자 계정과 자동으로 연결됩니다.

> **노트**
>
> `DBUser` 및 `DBPass` 는 정적 매개 변수로, 중앙 관리 콘솔에서 업데이트되어야 합니다. 데이터베이스 암호가 변경된 경우 중앙 관리 콘솔에서 각 사용자 계정의 암호를 업데이트하는 것이 좋습니다.

3.4.2.7 새 연결 정의

새 연결 마법사를 사용하여 새 데이터베이스 연결을 정의할 수 있습니다. 다음과 같은 방법으로 이 마법사에 액세스할 수 있습니다.

- 유니버스 매개 변수 대화 상자의 정의 페이지(파일 > 매개 변수 > 정의)에 액세스합니다. 일반적으로 유니버스에서 액세스해야 하는 데이터에 사용 가능한 기존의 연결이 없는 경우에 새 연결을 정의합니다.
- 연결 목록(도구 > 연결)에서 정의합니다.

연결 마법사를 사용하여 연결에 대한 고급 매개 변수와 사용자 지정 매개 변수를 설정할 수 있습니다. 연결을 만들고 편집하고 최적화하는 방법에 대한 자세한 내용은 Data Access 가이드를 참조하십시오.

유니버스 매개 변수 대화 상자에서 연결을 만드는 경우 유니버스 매개 변수 대화 상자가 나타나고 연결 상자에 새 연결이 나열됩니다.

연결 대화 상자에서 연결을 만드는 경우 만들어진 연결이 목록에 나타납니다.

관련 정보

새 연결 마법사 시작 [페이지 399]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.2.8 사용 가능한 연결 보기

저장되어 사용 가능한 모든 연결을 연결 목록에서 확인할 수 있습니다. 기존의 연결을 편집하거나 새 연결을 만들 수도 있습니다.

사용 가능한 연결을 보려면

1. 도구 > 연결 을 선택합니다.
   연결 목록이 나타납니다. 이 목록에는 현재 유니버스에 사용할 수 있는 모든 연결이 표시됩니다.
2. 취소를 클릭하여 대화 상자를 닫습니다.

연결 대화 상자에서 연결을 편집할 수 있습니다.

보안 연결은 온라인 모드에서 작업하는 경우에만 편집할 수 있습니다. 개인 및 공유 연결은 작업 모드에 상관 없이 수정할 수 있습니다.

기존 연결의 이름은 수정할 수 없습니다.

3.4.2.9 연결 편집

연결을 편집하려면

1. 도구 > 연결 을 선택합니다.
   연결 목록이 나타납니다.
2. 사용 가능한 연결의 목록에서 연결 이름을 클릭합니다.
3. 편집을 클릭합니다.
   연결에 대한 로그인 페이지가 나타납니다.
4. 필요한 경우 데이터 소스 또는 서비스 상자에 새 데이터 소스 또는 데이터베이스 이름을 입력합니다.
5. 필요에 따라 로그인 매개 변수의 수정된 값을 입력합니다.
6. 다음을 클릭합니다.
   테스트 수행 페이지가 나타납니다.
7. 데이터 소스 테스트를 클릭하여 수정된 연결을 확인합니다.
8. 다음을 클릭하여 고급 및 사용자 지정 페이지로 이동합니다. 필요에 따라 매개 변수를 수정할 수 있습니다. 기본값이나 기존의 값을 적용할 수도 있습니다.
9. 사용자 지정 페이지에서 마침을 클릭하여 연결에 대해 변경된 내용을 적용합니다.

3.4.2.10 연결 삭제

연결 목록에서 연결을 삭제할 수 있습니다. 보안 연결은 온라인 모드에서 작업하는 경우에만 삭제할 수 있습니다. 개인 및 공유 연결은 작업 모드에 상관 없이 삭제할 수 있습니다.

연결을 삭제하려면

1. 도구 > 연결 을 선택합니다.
   연결 목록이 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 목록에서 연결 이름을 선택합니다.
3. 제거를 클릭합니다.
   확인 상자가 나타납니다.
4. 예를 클릭합니다.
   연결이 목록에서 제거됩니다.

3.4.2.11 새 연결 추가

도구 > 연결 을 선택하고추가를 클릭한 다음 새 연결 정의 마법사의 지침에 따라 연결 페이지에서 새 연결을 추가할 수 있습니다. 연결 마법사를 사용하여 작업하는 방법에 대한 자세한 내용은 연결 편집 [페이지 79] 단원을 참조하십시오.

3.4.3 유니버스 요약 매개 변수 설정

요약 페이지에는 유니버스 관리 정보가 표시됩니다. 이 정보는 활성 유니버스의 개발 과정을 추적하는 데 도움이 됩니다.

요약 페이지에는 다음과 같은 정보가 표시됩니다.

표 26:

정보	설명
작성	유니버스 작성 날짜와 작성자의 이름입니다.
수정	마지막 수정 날짜와 수정한 사용자의 이름입니다.
수정	유니버스를 리포지토리로 내보낸 횟수를 나타내는 수정 번호입니다.
주석	자신과 다른 디자이너의 이해를 돕기 위한 유니버스 관련 정보입니다.
통계	유니버스에 포함된 클래스, 개체, 테이블, 별칭, 조인, 컨텍스트 및 계층의 수가 나열됩니다.

3.4.3.1 요약 정보를 보거나 입력하려면

1. 파일 > 매개 변수를 선택합니다.

   또는

   매개 변수 도구를 클릭합니다.

   유니버스 매개 변수 대화 상자가 열립니다.
2. 요약 탭을 클릭합니다.

   요약 페이지가 나타납니다.
3. 주석 텍스트 상자에 주석을 입력합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4. 확인을 클릭합니다.

3.4.4 전략 선택

전략은 플랫 파일 또는 데이터베이스에서 구조 정보를 자동으로 추출하는 스크립트입니다. 전략의 두 가지 기본 역할은 다음과 같습니다.

- 자동 조인 및 카디널리티 검색(조인 전략)
- 자동 클래스, 개체 및 조인 작성(개체 및 조인 전략)

데이터베이스의 SQL 구조를 기반으로 유니버스의 구조를 검색 및 작성하는 과정을 자동화하려는 경우에 전략을 유용하게 사용할 수 있습니다.

> **노트**
>
> 유니버스 구조 작성을 자동화하는 전략을 유니버스 디자인 및 작성의 필수 부분으로 포함시킬 필요는 없습니다. 이러한 전략을 사용하면 데이터베이스 또는 데이터베이스 디자인 도구에 이미 들어 있는 메타데이터 정보를 활용할 수 있으므로 유니버스를 신속하게 만들려는 경우에 유용할 수 있습니다. 그러나 사용자의 요구 사항 분석을 통해 직접 도출된 관계를 기반으로 개체와 조인을 만들어서 유니버스를 작성하는 경우에는 전략에서 제공하는 자동 작성 기능을 사용하지 않는 것이 더 좋을 수도 있습니다.

유니버스 디자인 도구에서는 다음과 같은 두 가지 유형의 전략을 지정할 수 있습니다.

표 27:

전략	설명
기본 제공 전략	유니버스 디자인 도구와 함께 제공된 기본 전략입니다. 기본 제공 전략은 사용자 지정할 수 없습니다.
외부 전략	기본 제공 전략과 동일한 유형의 정보를 포함하는 사용자 정의 스크립트이지만, 데이터베이스에서 정보를 최적으로 검색하기 위해 사용자 지정할 수 있습니다.

3.4.4.1 전략 선택

전략을 선택하려면

1. 파일 > 매개 변수 를 선택합니다.
   또는
   매개 변수 도구를 클릭합니다.
   유니버스 매개 변수 대화 상자가 나타납니다.
2. 전략 탭을 클릭합니다.
   전략 페이지가 나타납니다.
3. 개체, 조인 또는 테이블 드롭다운 목록 상자에서 전략을 선택합니다.
4. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.4.2 기본 제공 전략 사용

기본 제공 전략은 유니버스 디자인 도구와 함께 제공되는 기본 전략입니다. 지원되는 모든 데이터베이스에 대해 기본 전략이 제공됩니다. 이러한 기본 전략은 수정할 수 없습니다. 기본 전략은 전략 드롭다운 목록에서 외부 전략 앞에 표시됩니다.

기본 제공 전략은 다음과 같은 용도로 사용할 수 있습니다.

표 28:

전략	사용 목적
개체	테이블 스키마에서 테이블을 만들 때 기본 클래스 및 개체를 자동으로 작성합니다.*
조인	• 테이블 스키마에서 테이블을 만들 때 기본 조인을 자동으로 추출합니다.* • 조인 작성 시 카디널리티를 자동으로 삽입합니다.* • 테이블 스키마에서 조인을 자동으로 검색합니다. 도구 > 자동화된 검색 > 조인 검색을 선택하면 유니버스 디자인 도구가 전략을 사용하여 후보 조인을 자동으로 검색합니다. 조인을 구현하거나 구현하지 않도록 선택할 수 있습니다. • 테이블 스키마에서 기존의 조인을 자동으로 검색하고 카디널리티를 삽입합니다. 도구 > 자동화된 검색 > 카디널리티 검색을 사용하면 유니버스 디자인 도구가 전략을 사용하여 테이블 스키마에서 선택된 조인의 카디널리티를 자동으로 검색합니다.
테이블	테이블 브라우저에서 테이블에 사용할 수 있는 정보를 필터링합니다.

* 전략에 이러한 자동 작성 기능을 사용하려면 옵션 대화 상자의 데이터베이스 페이지에서 이 기능을 활성화해야 합니다.

개체 전략 사용

개체 전략은 테이블 스키마에 테이블을 추가할 때 클래스와 개체를 자동으로 만들려는 경우에만 사용합니다. 이 전략은 옵션 대화 상자의 데이터베이스 페이지에서 활성화해야 사용할 수 있습니다. 자세한 내용은 전략의 자동 작성 기능 사용 [페이지 83] 단원을 참조하십시오.

조인 전략 사용

선택된 조인 전략에 따라 유니버스 디자인 도구가 테이블 스키마에서 카디널리티와 조인을 자동으로 검색하는 방식이 결정됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

데이터베이스에 따라서는 목록에 하나 이상의 조인 전략이 있을 수 있습니다. 예를 들어, Oracle 데이터베이스를 사용하는 경우 일치하는 열 이름을 기준으로 또는 일치하는 열 번호 이름을 기준으로 조인을 자동 검색하도록 조인 전략을 지정할 수 있습니다.

전략을 선택하지 않을 경우 유니버스 디자인 도구에서는 열 이름과 일치하는 기본 조인 전략을 사용하여 조인을 검색합니다. 조인을 검색하기 위해 선택된 조인 전략을 사용하도록 활성화할 필요는 없습니다. 테이블 스키마에서 조인이나 카디널리티를 검색하도록 선택하면 항상 전략이 사용됩니다.

조인 전략은 조인을 자동으로 만들고 조인이 만들어졌을 때 카디널리티를 구현하기 위해 사용할 수도 있습니다. 이 전략의 자동 기본 작성 기능을 사용하려면 옵션 대화 상자의 데이터베이스 페이지에서 이를 활성화해야 합니다. 자세한 내용은 전략의 자동 작성 기능 사용 [페이지 83] 단원을 참조하십시오.

테이블 전략 사용

선택된 테이블 전략은 데이터베이스 테이블의 구조를 읽습니다. 전략 종류에 따라 테이블 브라우저에 표시되는 정보의 종류를 전략에서 결정할 수도 있습니다. 예를 들어, 열 데이터 유형과 설명이 표시되도록 결정할 수 있습니다.

3.4.4.3 전략의 자동 작성 기능 사용

전략의 자동 작성 및 삽입 기능은 기본적으로 활성화되어 있지 않습니다. 이러한 기능을 사용하려면 개체 또는 조인 작성에 적용할 전략에 해당하는 기본 작성 확인란을 선택해야 합니다. 이 확인란은 옵션 대화 상자의 데이터베이스 페이지(도구 > 옵션 > 데이터베이스)에 나열되어 있습니다.

다음은 데이터베이스 페이지에 있는 각 기본 작성 옵션에 대한 설명입니다.

표 29:

옵션	해제한 경우	선택한 경우
테이블과 함께 조인 추출	조인을 직접 만들어야 합니다. 도구 > 자동화된 검색 > 조인 검색을 선택하면 유니버스 디자인 도구가 전략을 사용하여 조인을 검색하고 후보 조인을 제안합니다. 후보 조인을 구현하거나 구현하지 않도록 선택할 수 있습니다.	선택된 조인 전략에 따라 테이블을 연결하는 조인이 포함된 테이블을 검색합니다.
조인에서 카디널리티 검색	카디널리티를 직접 정의해야 합니다. 도구 > 자동화된 검색 > 카디널리티 검색을 선택하면 유니버스 디자인 도구가 전략을 사용하여 선택된 조인의 카디널리티를 검색하고 구현합니다.	조인 작성 시 조인에 내재된 카디널리티를 검색하고 구현합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

옵션	해제한 경우	선택한 경우
테이블에서 기본 클래스 및 개체 만들기	클래스와 개체를 유니버스 창에서 직접 만들거나 구조 창에서 유니버스 창으로 테이블이나 열을 끌어 놓아 클래스와 개체를 만들어야 합니다.	테이블을 구조 창에 추가하면 유니버스 창에 기본 클래스와 개체가 자동으로 생성됩니다. 클래스는 테이블 이름에 해당하고 개체는 열 이름에 해당합니다. 이 경우 밑줄 문자(_)는 모두 공백으로 바뀝니다.

전략의 기본 작성 옵션을 선택하려면

1. 도구 > 옵션 을 선택합니다.
   옵션 대화 상자가 나타납니다.
2. 데이터베이스 탭을 클릭합니다.
   데이터베이스 페이지가 나타납니다.
3. 전략을 사용하려는 기본 작성 기능에 해당하는 확인란을 선택합니다.
4. 확인을 클릭합니다.

3.4.4.4 표시할 행의 수 설정

데이터베이스 옵션 대화 상자를 사용하면 데이터베이스의 각 테이블에서 표시할 행의 최대 수를 지정할 수도 있습니다. 데이터베이스에서 검색된 실제 행의 수를 제한할 수 없지만, 테이블 또는 열 값을 표시하는 경우 동시에 표시할 수 있는 행의 수에 대한 기본값을 수정합니다. 이 설정은 유니버스 디자인 도구에서 반환된 행에만 적용되며 Web Intelligence에서 실행된 쿼리에 대해서는 적용되지 않습니다.

표시할 수 있는 행의 최대 수를 설정하려면

- 반입된 행의 최대 수 옵션의 텍스트 상자에 값을 입력합니다. 위쪽 또는 아래쪽 화살표를 한 번 이상 클릭하여 기본값(100)을 늘리거나 줄일 수도 있습니다.

3.4.4.5 외부 전략 사용

외부 전략은 사용자 지정된 자동 유니버스 작성 작업을 수행하기 위해 정의된 출력 구조에 따라 사용자 정의된 SQL 스크립트입니다. 외부 전략은 외부 XML 전략 파일(<RDBMS>.STG)에 저장됩니다. 이 파일의 SQL 스크립트는 다른 전략과 함께 전략 페이지의 드롭다운 목록에 표시됩니다.

외부 전략에는 기본 제공 전략과 동일한 유형의 정보가 포함되지만, 유니버스 디자인 도구가 특정 유형의 데이터베이스 정보를 검색하거나 데이터베이스에서 정보를 검색하는 방식을 최적화할 수 있도록 사용자 지정되는 경우가 많습니다.

외부 전략을 정의하는 방법에 대한 자세한 내용은 외부 전략을 사용하여 유니버스 만들기 사용자 지정 [페이지 365] 단원을 참조하십시오.

3.4.5 리소스 제어 지정

유니버스 디자인 도구에서는 시스템 리소스에 대한 사용을 제어할 수 있는 여러 가지 옵션을 제공합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

> **노트**
>
> 제한 미리 보기 대화 상자에서 이 탭을 보는 경우에는 제한에 적용되도록 수정된 매개 변수가 빨간색으로 표시됩니다.

3.4.6 사용 가능한 시스템 리소스 옵션

시스템 리소스에 대해 다음과 같은 제한을 지정할 수 있습니다.

표 30:

쿼리 제한	설명
결과 집합의 크기를 지정된 값으로 제한	쿼리에서 반환되는 행의 수를 지정한 값으로 제한합니다. 이 제한은 반환되는 행의 수에만 적용되며, RDBMS 에서 쿼리의 모든 행을 처리하지 못하게 하지는 않습니다. 이 제한은 RDBMS 에서 행을 보내기 시작한 이후의 행의 수에만 적용됩니다.
실행 시간을 지정된 값으로 제한	쿼리 실행 시간이 분 단위의 지정된 값으로 제한됩니다. 이 옵션은 Web Intelligence 에 데이터가 전달되는 시간을 제한하지만 데이터베이스에서 프로세스를 중지하지는 않습니다.
긴 텍스트 개체의 길이를 지정된 값으로 제한	긴 텍스트 개체의 최대 문자 수를 지정합니다. 이 확인란을 선택하지 않으면 매개 변수가 활성화되지 않습니다. 이 옵션은 기본 최대값(1000)으로 자동 설정됩니다. 기본값보다 큰 결과를 허용하려면 이 확인란을 선택하고 값을 입력합니다.

3.4.7 리소스 제어 정보를 입력하려면

1. 파일 > 매개 변수를 선택합니다.

   또는

   매개 변수 도구를 클릭합니다.

   유니버스 매개 변수 대화 상자가 열립니다.
2. 제어 탭을 클릭합니다.

   제어 페이지가 나타납니다.
3. 쿼리 제한 그룹 상자에서 확인란을 선택합니다.

   선택된 쿼리 제한 옵션에 해당하는 텍스트 상자에 값을 입력합니다. 텍스트 상자의 끝에 있는 위쪽 및 아래쪽 화살표를 클릭하여 입력된 값을 늘리거나 줄일 수 있습니다.
4. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.8 여러 SQL 문을 생성하는 쿼리의 실행 시간 제한

쿼리 실행에 대해 지정하는 시간 제한은 쿼리의 총 실행 시간입니다. 쿼리에 여러 개의 SQL 문이 포함되어 있는 경우 각 문에는 총 쿼리 실행 시간을 문의 수로 나눈 시간이 할당되므로 쿼리의 각 문에 동일한 실행 시간이 부여됩니다.

다른 문에 비해 실행 시간이 훨씬 많이 필요한 문의 경우 실행 시간이 쿼리 내에서 그 문에 할당된 실행 시간보다 길어 문의 실행이 완료되지 않을 수도 있습니다.

여러 SQL 문에 대한 실행 시간 제한값을 지정할 때는 실행 시간이 가장 긴 문 하나의 정상 실행 시간을 고려하고 이 값에 쿼리의 문 수를 곱해야 합니다.

3.4.9 SQL 제한 지정

최종 사용자가 Web Intelligence 의 쿼리 창에서 수식화할 수 있는 쿼리 유형에 대한 제어를 설정할 수 있습니다.

쿼리 생성과 관련된 다음과 같은 영역에 대한 제어를 지정할 수 있습니다.

- 개별 쿼리의 복합 피연산자, 연산자 및 하위 쿼리 사용
- 여러 SQL 문 생성
- 카티전 곱 발생 방지 또는 경고

이러한 각 제어 집합은 다음 단원에서 설명합니다.

3.4.9.1 쿼리 제어

개별 쿼리에 대해 다음과 같은 제어를 설정할 수 있습니다.

표 31:

옵션	설명
UNION, INTERSECT 및 MINUS 연산자 허용	최종 사용자가 데이터 집합 연산자(UNION, INTERSECT 및 MINUS)를 사용하여 쿼리를 조합하여 하나의 결과 집합을 얻을 수 있도록 합니다.

3.4.9.2 여러 SQL 문 제어

여러 SQL 문의 처리 방법을 결정하기 위한 다음과 같은 제어를 설정할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 32:

옵션	설명
각 컨텍스트에 여러 SQL 문 사용	최종 사용자가 컨텍스트를 사용할 때 여러 SQL 문이 포함된 쿼리를 만들 수 있도록 합니다. 유니버스에 컨텍스트가 있는 경우 이 옵션을 선택합니다.
각 계수에 여러 SQL 문	서로 다른 테이블의 열에서 파생된 계수 개체가 쿼리에 포함될 때마다 SQL 을 여러 개의 문으로 분할합니다. 이 옵션의 사용 방법에 대한 자세한 내용은 각 계수에 여러 SQL 문 사용 옵션 사용 [페이지 225] 단원을 참조하십시오. 계수 개체가 동일한 테이블의 열을 기반으로 한 경우 이 옵션을 선택해도 SQL 이 분할되지 않습니다.
다중 컨텍스트 선택 허용	최종 사용자가 여러 컨텍스트의 개체에 대한 쿼리를 만들고 다중 컨텍스트에서 하나의 결과 집합을 생성할 수 있도록 합니다. 컨텍스트를 사용하여 루프, 캐즘 트랩, 팬 트랩 또는 기타 조인 경로 문제를 해결하는 경우 이 확인란을 해제해야 합니다.

3.4.9.3 카티전 곱 제어

카티전 곱은 쿼리에 포함된 각 테이블의 각 행에 대한 가능한 조합이 모두 포함된 결과 집합입니다. 카티전 곱의 결과는 거의 항상 부정확합니다.

카티전 곱의 생성에 대한 다음과 같은 제어를 설정할 수 있습니다.

표 33:

옵션	설명
사용 못함	이 옵션을 선택하면 카티전 곱이 생성되는 쿼리가 실행되지 않습니다.
경고	이 옵션을 선택하면 쿼리의 결과로 카티전 곱이 생성될 수 있음을 알리는 경고 메시지가 최종 사용자에게 표시됩니다.

3.4.9.4 SQL 제한 옵션 입력

SQL 제한 옵션을 입력하려면

1. 파일 > 매개 변수 를 선택합니다.
   또는
   매개 변수 도구를 클릭합니다.
   유니버스 매개 변수 대화 상자가 나타납니다.
2. *SQL* 탭을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

SQL 페이지가 나타납니다.
3. 쿼리 및 여러 경로 그룹 상자에서 옵션을 선택하거나 해제합니다.
4. 카티전 곱 그룹 상자에서 라디오 단추를 선택합니다.
5. 확인을 클릭합니다.

3.4.10 연결된 유니버스의 옵션 지정

링크 탭은 동적으로 연결된 유니버스와 함께 사용됩니다. 자세한 내용은 유니버스 배포 [페이지 484] 장을 참조하십시오.

3.4.11 SQL 생성 매개 변수 설정

유니버스 디자인 도구에서는 대부분의 RDBMS 에 공통되는 특정 SQL 매개 변수를 동적으로 구성하여 유니버스를 사용하는 Web Intelligence 제품에서 생성된 SQL 을 최적화할 수 있습니다.

3.4.11.1 이전 버전의 유니버스 디자인 도구에서 매개 변수(PRM) 파일 사용

Designer 6.5 이전 버전에서는 유니버스에 의해 사용되는 SQL 생성 매개 변수가 매개 변수(PRM) 파일이라는 별도의 파일에서 유지 관리 및 편집되었습니다. PRM 파일에서 설정된 값은 연결에 대해 정의된 해당 데이터 액세스 드라이버를 사용하는 모든 유니버스에 적용되었습니다.

쿼리 생성을 최적화하는 데 사용되는 많은 SQL 매개 변수가 이제는 개별 유니버스 파일 내에서 제어됩니다. 더 이상 쿼리 생성 매개 변수에 대해 PRM 파일이 사용되지 않습니다. 이제 유니버스 디자인 도구에서 매개 변수를 설정할 수 있습니다. 단, 데이터베이스 관련 매개 변수에 대해서는 PRM 파일이 계속 사용됩니다.

> **노트**
>
> 데이터 액세스 드라이버와 관련하여 PRM 파일에 대한 자세한 내용은 데이터 액세스 가이드를 참조하십시오. 도움말 > 데이터 액세스 가이드 를 선택하면 이 가이드에 액세스할 수 있습니다.

3.4.11.2 유니버스 디자인 도구에서 동적으로 SQL 매개 변수 설정

지원되는 대부분의 RDBMS 미들웨어에 공통적으로 존재하는 많은 매개 변수는 유니버스 매개 변수 대화 상자의 매개 변수 탭(파일 > 매개 변수 > 매개 변수)에서 편집할 수 있습니다.

이러한 매개 변수는 활성 유니버스에만 적용되며 UNV 파일에 저장됩니다. 유니버스 디자인 도구에서 유니버스에 대한 SQL 매개 변수를 수정할 경우, 연결에 대한 데이터 액세스 드라이버와 연관된 PRM 파일에서 정의된 값이 아닌 유니버스 디자인 도구에서 정의된 값이 사용됩니다.

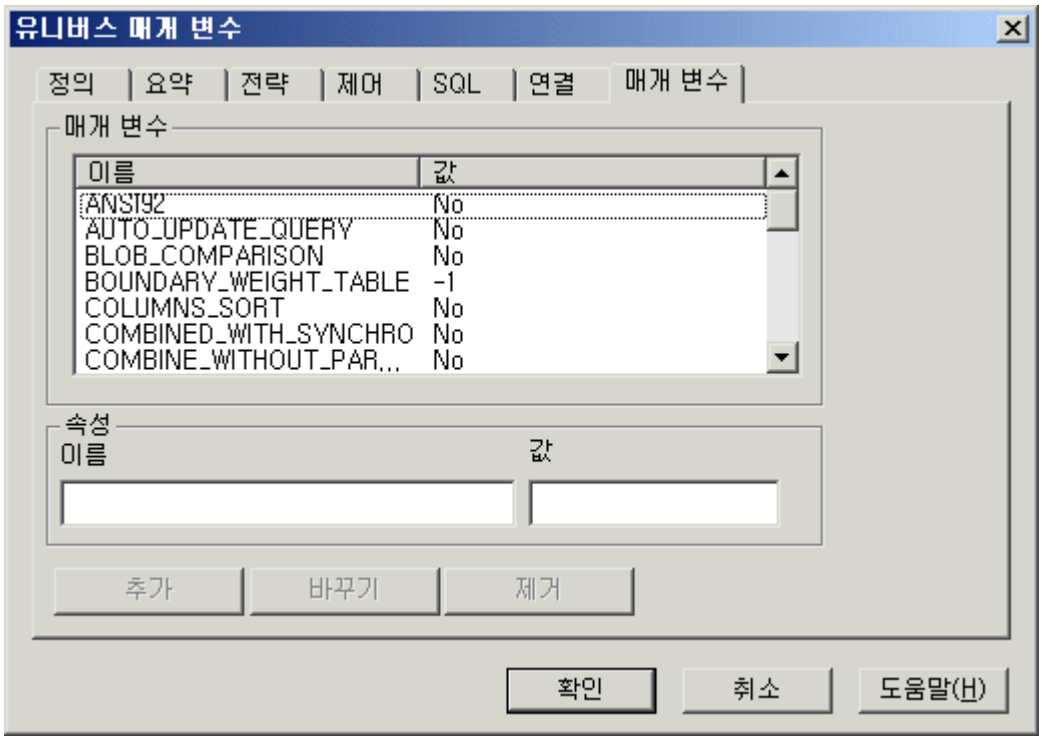
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.11.3 SQL 생성 매개 변수 편집

유니버스를 사용하는 제품에서 SQL 생성을 결정하는 SQL 매개 변수에 대한 값을 수정할 수 있습니다.

SQL 생성 매개 변수를 편집하려면

1. 파일 > 매개 변수 를 선택합니다.
   매개 변수 대화 상자가 나타납니다.
2. 매개 변수 탭을 클릭합니다.
   매개 변수 페이지가 나타납니다.



3. 다음과 같이 매개 변수를 편집, 추가 또는 제거합니다.

표 34:

작업	단계
새 매개 변수 추가	○ 목록에서 매개 변수를 클릭합니다. ○ 이름 상자에 이름을 입력합니다. ○ 값 상자에 값을 입력합니다. ○ 추가를 클릭합니다. ○ 새 값이 목록 맨 아래 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

작업	단계
이름 또는 값 변경	○ 목록에서 매개 변수를 클릭합니다. ○ 이름 상자에 새 이름을 입력합니다. ○ 값 상자에 새 값을 입력합니다. ○ 바꾸기를 클릭합니다. 값이 새 정의로 바뀝니다.
매개 변수 삭제	○ 목록에서 제거할 매개 변수를 클릭합니다. ○ 삭제를 클릭합니다.

4. 확인을 클릭합니다.

> **노트**
>
> 유니버스에서 설정한 SQL 생성 매개 변수 값은 해당 유니버스를 사용하는 제품에서만 사용할 수 있습니다.

3.4.12 SQL 생성 매개 변수 정보

다음은 쿼리 스크립트 생성에 영향을 미치는 매개 변수에 대한 설명입니다. 매개 변수는 다음 두 개 그룹으로 나뉘어 알파벳 순으로 나열됩니다.

- 유니버스 디자인 도구 또는 정보 디자인 도구의 사용자 인터페이스에서 설정한 SQL 매개 변수. 이는 대부분의 데이터 액세스 드라이버에 공통적인 SQL 매개 변수입니다. 각 매개 변수는 이를 설정한 유니버스에 대해서만 유효합니다.
- PRM 파일에서 설정한 SQL 매개 변수. 이는 대상 데이터 액세스 드라이버에 대한 데이터 액세스 매개 변수(PRM) 파일에 나열된 연결 전용 매개 변수입니다.

관련 정보

사용자 인터페이스에서 설정한 SQL 매개 변수 [페이지 91]
PRM 파일에서 설정한 SQL 매개 변수 [페이지 106]

3.4.13 동적 SQL 생성 매개 변수 편집

1. 파일 > 매개 변수를 선택합니다.

   매개 변수 대화 상자가 나타납니다.
2. 매개 변수 탭을 클릭합니다.

   매개 변수 페이지가 나타납니다.
3. 다음과 같이 매개 변수를 편집, 추가 또는 제거합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 35:

작업	방법
새 매개 변수 추가	목록에서 매개 변수를 클릭합니다. 이름 상자에 이름을 입력합니다. 값 상자에 값을 입력합니다. 추가를 클릭합니다. 새 값이 목록 맨 아래 나타납니다.
이름 또는 값 변경. 목록에서 매개 변수를 클릭합니다.	이름 상자에 새 이름을 입력합니다. 값 상자에 새 값을 입력합니다. 바꾸기를 클릭합니다. 값이 새 정의로 바뀝니다.
매개 변수 삭제	목록에서 제거할 매개 변수를 클릭합니다. 제거를 클릭합니다.

4. 확인을 클릭합니다.

유니버스에서 설정한 SQL 생성 매개 변수 값은 해당 유니버스를 사용하는 제품에서만 사용할 수 있습니다.

동적 SQL 매개 변수 및 데이터 액세스 드라이버와 관련한 PRM 파일 편집에 대한 자세한 내용은 데이터 액세스 가이드를 참조하십시오. 도움말 > 데이터 액세스 가이드를 선택하면 이 가이드에 액세스할 수 있습니다.

3.4.14 사용자 인터페이스에서 설정한 SQL 매개 변수

3.4.14.1 ANSI92

`ANSI92` = Yes|No

표 36:

값	`Yes/No`
기본값	`No`
설명	생성된 SQL 을 `ANSI92` 표준으로 컴파일할지 지정합니다. `Yes`: SQL 생성이 `ANSI92` 표준과 호환되도록 합니다. `No`: SQL 생성이 `PRM` 매개 변수 `OUTER_JOIN_GENERATION` 에 따라 동작합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.14.2 AUTO_UPDATE_QUERY

`AUTO_UPDATE_QUERY` = Yes|No

표 37:

값	`Yes/No`
기본값	`No`
설명	쿼리의 개체를 사용자 프로필에 사용할 수 없는 경우 어떠한 작업을 수행할지 결정합니다. `Yes`: 쿼리를 업데이트하고 쿼리에서 개체를 제거합니다. `No`: 개체를 쿼리에 유지합니다.

3.4.14.3 BACK_QUOTE_SUPPORTED

`BACK_QUOTE_SUPPORTED` = Yes|No

표 38:

값	`YES`: SQL 에 역따옴표를 사용합니다. `NO`: SQL 에 역따옴표를 사용하지 않습니다.
기본값	`YES` `No`(OpenAccess 데이터베이스의 경우)
설명	SQL 에 공백이나 특수 문자가 포함된 테이블 또는 열 이름을 묶을 역따옴표를 사용할지 여부를 지정합니다.
결과	Table name=`My Table`

3.4.14.4 BEGIN_SQL

`BEGIN_SQL` = <String>

표 39:

값	문자열
기본값	빈 문자열

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

설명	이 매개 변수는 계정, 우선 순위 및 작업 부하 관리에 사용할 SQL 문 접두사를 추가하는 데 사용됩니다. 이 매개 변수는 문서 생성 및 LOV 쿼리를 비롯한 모든 SQL 생성에 적용됩니다.
	
	이 매개 변수는 Web Intelligence, LiveOffice 및 QaaWS 에서 지원됩니다. 그러나 Desktop Intelligence 및 Crystal Reports 에서는 무시됩니다.
	
	Teradata 예:
	
	`BEGIN_SQL=SET QUERY_BAND='string' for transaction;`
	
	이 매개 변수에는 하나 이상의 이름-값 쌍이 포함되어야 하며, 이름-값 쌍은 세미콜론으로 구분되고 작은 따옴표로 둘러싸야 합니다. 모든 SQL 문에는 BEGIN_SQL 뒤에 오는 매개 변수가 접두사로 추가됩니다. 이 매개 변수에 입력한 이름-값 쌍은 GetQueryBandPairs 시스템 테이블에 기록됩니다.
	
	세 개의 이름-값 쌍에 대한 예:
	
	`BEGIN_SQL=SET QUERY_BAND='UserID=Jones;JobID=980;AppID=TRM' for transaction;`
	
	또한 이름-값 쌍에서 @Variable 함수를 값으로 사용할 수 있으며 반환된 값은 작은 따옴표로 묶습니다. `BEGIN_SQL=SET QUERY_BAND='USER='@Variable('BOUSER');Document='@Variable('DPNAME')';' for transaction;`

3.4.14.5 BLOB_COMPARISON

`BLOB_COMPARISON` = Yes|No

표 40:

값	`Yes/No`
기본값	`No`
편집 가능 여부	No
설명	BLOB 파일이 `SELECT` 문에 사용되는 경우 `DISTINCT` 문을 사용하여 쿼리를 생성할 수 있는지 지정합니다. 이는 쿼리 속성의 `No Duplicate Row` 설정과 관련이 있습니다. `Yes`: `DISTINCT` 문을 쿼리 내에 사용할 수 있습니다. `No`: `No Duplicate Row` 쿼리 설정을 사용하는 경우라 해도 쿼리 내에 `DISTINCT` 문을 사용할 수 없습니다.

3.4.14.6 BOUNDARY_WEIGHT_TABLE

`BOUNDARY_WEIGHT_TABLE` = Integer 32bits [0-9]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 41:

값	32 비트 정수[0-9 또는 음의 정수]
기본값	-1
설명	테이블에 행이 여러 개인 경우 FROM 절을 최적화할 수 있습니다. 입력된 값보다 테이블 크기(행 수)가 더 크면 테이블이 하위 쿼리로 선언됩니다. FROM (SELECT **col1, col2,....., coln, ,....,** FROM Table_Name WHERE **단순 조건**). 단순 조건은 하위 쿼리가 없는 조건으로 정의됩니다. -1, 0 또는 모든 음수에 이 최적화가 사용되지 않는다는 의미입니다.
제한	다음과 같은 경우에는 최적화가 구현되지 않습니다. • 쿼리 조건에 OR 연산자가 있는 경우 • SQL 에 관련된 테이블이 하나뿐인 경우 • 쿼리에 외부 조인이 포함된 경우 • 최적화하려는 테이블에 대한 조건이 정의되지 않은 경우 • 최적화하려는 테이블이 파생된 테이블인 경우

3.4.14.7 COLUMNS_SORT

COLUMNS_SORT = Yes|No

표 42:

값	Yes/No
기본값	No
설명	구조 창의 테이블에 열이 표시되는 순서를 결정합니다. Yes: 열을 알파벳 순으로 표시합니다. No: 데이터베이스에서 가져온 순서대로 열을 표시합니다.

3.4.14.8 COMBINE_WITHOUT_PARENTHESIS

COMBINE_WITHOUT_PARENTHESIS= Yes|No

표 43:

값	Yes/No

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

기본값	No
설명	UNION, INTERSECT 또는 MINUS 연산자가 포함되어 있을 때 쿼리를 괄호로 묶어야 하는지 여부를 지정합니다. RedBrick 에서 사용됩니다. Yes: 괄호를 제거합니다. No: 괄호를 그대로 둡니다.

3.4.14.9 COMBINED_WITH_SYNCHRO

COMBINED_WITH_SYNCHRO = Yes|No

표 44:

값	Yes|No
기본값	No
설명	UNION, INTERSECTION 또는 EXCEPT 연산자를 포함하고 있으며 각 하위 쿼리에 호환되지 않는 개체를 가진 쿼리를 실행할 수 있는지 여부를 지정합니다. Yes: UNION, INTERSECTION 및 EXCEPT 연산자를 포함하고 있으며 각 하위 쿼리에 호환되지 않는 개체를 가진 쿼리를 실행할 수 있도록 지정합니다. 이 유형의 쿼리에서는 동기화(보고서의 두 블록)가 생성됩니다. No: UNION, INTERSECTION 및 EXCEPT 연산자를 포함하고 있으며 각 하위 쿼리에 호환되지 않는 개체를 가진 쿼리를 실행할 수 없도록 지정합니다. 쿼리를 실행하면 다음 오류 메시지가 나타납니다. "이 쿼리는 너무 복잡합니다. 하위 쿼리 중 하나에 호환되지 않는 개체가 들어 있습니다." 이 값이 기본값입니다.

3.4.14.10 COMPARE_CONTEXTS_WITH_JOINS

COMPARE_CONTEXTS_WITH_JOINS = Yes|No

표 45:

값	Yes|No
기본값	Yes
설명	컨텍스트 비교 방법을 지정합니다. Yes: 시스템은 컨텍스트가 동일한 조인을 제공하는지 확인합니다. No: 시스템은 컨텍스트가 동일한 테이블 집합을 제공하는지 확인합니다. 기본값입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.14.11 CORE_ORDER_PRIORITY

CORE_ORDER_PRIORITY = Yes|No

표 46:

값	Yes|No
기본값	No
설명	이 매개 변수는 연결된 파생 유니버스에 추가하는 클래스나 개체에 적용됩니다. 주요 유니버스나 원본 파생 유니버스의 클래스나 개체에는 이 매개 변수가 적용되지 않습니다. 이 매개 변수는 유니버스 디자인 도구에서 새 클래스와 개체를 구성하는 방식을 지정합니다. FIRST_LOCAL_CLASS_PRIORITY 매개 변수를 참조하십시오. Yes: 클래스와 개체가 다음과 같이 구성되도록 지정합니다. • 첫 번째 주요 유니버스 클래스 주요 유니버스 개체 첫 번째 주요 유니버스 클래스에 속한 모든 파생 유니버스 개체 • 두 번째 주요 유니버스 클래스 주요 유니버스 개체 두 번째 주요 유니버스 클래스에 속한 모든 파생 유니버스 개체 • 기타 주요 유니버스 클래스... • 파생 유니버스 클래스 및 개체 No: 클래스와 개체가 파생 유니버스에 정의된 원래 순서를 따르도록 지정합니다. 기본값입니다.

3.4.14.12 CORRECT_AGGREGATED_CONDITIONS_IF_DRILL

CORRECT_AGGREGATED_CONDITIONS_IF_DRILL = Yes|No

표 47:

값	Yes|No
기본값	No
설명	Desktop Intelligence 에만 적용됩니다. Desktop Intelligence 가 쿼리 및 조건에서 계수를 집계할 수 있는지 지정합니다. Yes: 쿼리를 드릴할 수 있는 경우 Desktop Intelligence 는 기본 쿼리 및 조건에서 개별적으로 계수를 집계할 수 있습니다. No: 쿼리를 드릴할 수 있는 경우 Desktop Intelligence 는 기본 쿼리 및 조건에서 개별적으로 계수를 집계할 수 없습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.14.13 CUMULATIVE_OBJECT_WHERE

`CUMULATIVE_OBJECT_WHERE` = Yes|No

표 48:

값	`Yes|No`
기본값	`No`

설명	이 매개 변수는 필터링된 개체에만 적용됩니다. 개체의 `WHERE` 절과 해당 개체의 쿼리 조건을 결합할 방법을 지정합니다.
	`Yes`: `WHERE` 절이 `AND` 연산자를 사용한 기본 쿼리와 결합되도록 지정합니다.
	`No`: 개체의 `WHERE` 절이 이 개체에 대한 조건과 결합되도록 지정합니다.
	예제:
	John 이 아닌 모든 프랑스인 고객을 찾거나 뉴욕 이외의 미국 도시를 찾는 조건이 필요한 경우 SQL 은 다음과 같습니다.
	`Yes`:
	`(customer.first_name <> 'John') OR (city.city <> 'New York AND customer_country.country = 'France' AND city_country.country = 'USA'`
	`No`:
	`(customer.first_name <> 'John' AND customer_country.country = 'France' ) OR (city.city <> 'New York' AND city_country.country = 'USA' )`

3.4.14.14 DECIMAL_COMMA

`DECIMAL_COMMA` = Yes|No

표 49:

값	`Yes|No`
기본값	`No`

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

설명	Business Objects 제품에서 필요한 경우 소수 구분 기호로 쉼표를 삽입하도록 지정합니다. `Yes`: Business Objects 제품에서 필요한 경우 소수 구분 기호로 쉼표를 삽입합니다. `No`: Business Objects 제품에서 소수 구분 기호로 쉼표를 삽입하지 않습니다. 기본값입니다.

3.4.14.15 DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION

`DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION` = Yes|No

표 50:

값	`Yes/No`
기본값	`No`
설명	기본 설정을 사용하는 대신 반환된 배열의 크기를 최적화하는 데 사용할 수 있는 최적화 알고리즘입니다. `No`: 유니버스에서 실행되는 모든 쿼리가 이 최적화의 혜택을 받을 수 있습니다. `Yes`: 쿼리에서 기본 설정을 사용합니다. 이 매개 변수는 OLAP 연결에도 적용됩니다.

3.4.14.16 DISTINCT_VALUES

`DISTINCT_VALUES` = GROUPBY|DISTINCT

표 51:

값	`GROUPBY` | `DISTINCT`
기본값	`DISTINCT`
설명	"중복 행을 검색하지 않음" 옵션을 사용하는 경우 쿼리 패널과 값 목록에서 `DISTINCT` 또는 `GROUPBY` 절을 사용하여 SQL 을 생성하도록 지정합니다. `DISTINCT`: `DISTINCT` 절을 사용하여 SQL 을 생성합니다. 예를 들면 다음과 같습니다. `SELECT DISTINCT` cust_name `FROM Customers` `GROUPBY`: `GROUP BY` 절을 사용하여 SQL 을 생성합니다. 예를 들면 다음과 같습니다. `SELECT` cust_name `FROM` Customers `GROUP BY` cust_name

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.14.17 END_SQL

END_SQL = String

표 52:

값	문자열
기본값	<빈 문자열>
설명	이 매개 변수에 지정된 문이 각 SQL 문의 끝에 추가됩니다.
예제	IBM DB2 데이터베이스의 경우 다음과 같이 사용할 수 있습니다. END_SQL=FOR SELECT ONLY 서버에서 데이터 블록을 훨씬 빠르게 읽을 수 있습니다. 다른 예는 다음과 같습니다. END_SQL='write ' UNVID To Usage_Audit.Querieded_universe 유니버스 ID 를 감사 테이블에 기록하여 쿼리 대상 사용자 및 테이블과 같은 기타 데이터를 기록할 수 있습니다.

3.4.14.18 EVAL_WITHOUT_PARENTHESIS

EVAL_WITHOUT_PARENTHESIS = Yes|No

표 53:

값	Yes|No
기본값	No
설명	기본적으로 @Select(클래스\개체) 함수는 괄호 안에 포함된 <클래스\개체> 개체에 대한 SELECT 문으로 바뀝니다. 예를 들어, @Select(개체 1) * @Select(개체 2) 같이 두 개의 @Select 문을 결합하는 경우 SQL(개체 1) = A-B 이고 SQL(개체 2) =C 이면 연산은 (A-B) * (C)가 됩니다. EVAL_WITHOUT_PARENTHESIS = Yes 로 설정하면 괄호가 기본적으로 추가되지 않도록 할 수 있습니다. 이 경우 연산은 A - B * C 입니다. Yes: 함수 @Select(클래스\개체)에 대해 SELECT 문에서 괄호를 제거합니다. No: @Select(클래스\개체)에 대해 Select 문 앞뒤에 괄호를 추가합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.14.19 FILTER_IN_FROM

FILTER_IN_FROM = Yes|No

표 54:

값	Yes|No
기본값	No
설명	FROM 절에 쿼리 조건이 포함되는지 결정합니다. 이 설정은 다른 유니버스 매개 변수 설정 ANSI92 를 Yes. 로 설정한 경우에만 적용됩니다. Yes: 외부 조인을 편집할 때 유니버스 디자인 도구의 고급 조인 속성 대화 상자에 있는 드롭다운 목록 상자에서 선택된 기본 동작 속성이 "FROM 절에 모든 개체"로 설정됩니다. No: 외부 조인을 편집할 때 유니버스 디자인 도구의 고급 조인 속성 대화 상자에 있는 드롭다운 목록 상자에서 선택된 기본 동작 속성이 "FROM 절의 모든 개체 제외"로 설정됩니다.

3.4.14.20 FIRST_LOCAL_CLASS_PRIORITY

FIRST_LOCAL_CLASS_PRIORITY = Yes|No

표 55:

값	Yes|No
기본값	No
설명	이 매개 변수는 Desktop Intelligence 에만 적용됩니다. CORE_ORDER_PRIORITY=Yes 인 경우만 고려합니다. Yes: 파생 유니버스의 클래스가 먼저 나열됩니다. No: 파생 유니버스의 개체와 하위 클래스가 주요 유니버스의 개체와 하위 클래스 뒤에 나타납니다.

3.4.14.21 FORCE_SORTED_LOV

FORCE_SORTED_LOV = Yes|No

표 56:

값	Yes|No
기본값	No

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

설명	정렬된 값 목록을 검색합니다. Yes: 값 목록을 정렬하도록 지정합니다. No: 값 목록을 정렬하지 않도록 지정합니다.

3.4.14.22 INNERJOIN_IN_WHERE

INNERJOIN_IN_WHERE = Yes|No

표 57:

값	Yes|No
기본값	No. 매개 변수를 직접 추가하여 활성화해야 합니다.
설명	ANSI92 가 yes 로 설정된 경우 시스템에서 WHERE 절에 모든 내부 조인이 있는 SQL 구문을 생성합니다. 이는 쿼리에 FULL OUTER, RIGHT OUTER 또는 LEFT OUTER 조인이 아닌 내부 조인만 들어 있는 경우에만 가능합니다. Yes: ANSI92 가 yes 로 설정되면 시스템은 쿼리에 내부 조인만 포함된 경우를 제외하고 FROM 절에 ANSI92 조인 구문을 생성합니다. 이 경우 내부 조인은 WHERE 절에 포함됩니다. No: ANSI92 가 Yes 로 설정되면 시스템은 FROM 절에 ANSI 92 조인 구문을 생성합니다.

3.4.14.23 JOIN_BY_SQL

JOIN_BY_SQL = Yes|No

표 58:

값	Yes|No
기본값	No
설명	여러 SQL 문이 처리되는 방식을 지정합니다. 여러 문을 결합할 수 있습니다(데이터베이스가 허용할 경우). Yes: 여러 SQL 문을 결합하도록 지정합니다. No: 여러 SQL 문을 결합하지 않도록 지정합니다. 기본값입니다.

3.4.14.24 MAX_INLIST_VALUES

MAX_INLIST_VALUES = [0-99]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 59:

값	정수: 최소 -1, 최대는 DB 에 따라 다름
기본값	-1
설명	`IN LIST` 연산자를 사용할 때 조건에 입력할 수 있는 값의 최대 수를 설정할 수 있도록 합니다. 99: `IN LIST` 연산자를 사용하여 조건을 만들 때 값을 최대 99 개까지 입력할 수 있도록 지정합니다. 입력할 수 있도록 허용된 최대 값은 데이터베이스에 따라 다릅니다. -1 값은 데이터베이스에 의해 발생하는 제한을 제외하면 반환되는 값 수에 제한이 없다는 의미입니다.

3.4.14.25 OLAP_UNIVERSE

`OLAP_UNIVERSE` = Yes|No

표 60:

값	`Yes` | `No`
기본값	기본값 없음
설명	OLAP 유니버스의 사용 여부를 나타냅니다. 유니버스 디자인 도구에서 OLAP 유니버스를 사용하는 경우 이 값이 `Yes` 로 설정되고 SQL 매개 변수 목록에 이 매개 변수가 표시됩니다. OLAP 유니버스가 아닌 경우 SQL 매개 변수 목록에 이 매개 변수가 표시되지 않습니다. `Yes`: 유니버스가 OLAP 유니버스입니다. `No`: 유니버스가 OLAP 유니버스가 아닙니다.

3.4.14.26 PATH_FINDER_OFF

이 매개 변수는 기본적으로 나열되지 않습니다. 매개 변수를 목록에 직접 추가하고 값을 설정해야 합니다.

`PATH_FINDER_OFF`= Yes|No

표 61:

값	`Yes` | `No`
기본값	기본값이 없습니다. 매개 변수를 직접 입력해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

설명	조인 생성이 데이터베이스에서 수행되므로 HPIW 에 사용됩니다. Yes: 쿼리에서 조인을 생성하지 않습니다. No: 쿼리에서 조인을 생성합니다. 기본 동작입니다.

3.4.14.27 REPLACE_COMMA_BY_CONCAT

REPLACE_COMMA_BY_CONCAT= Yes|No

표 62:

값	Yes | No
기본값	No
설명	이전 버전의 유니버스 디자인 도구에서는 쉼표를 사용하여 개체 Select 문의 여러 필드를 구분할 수 있었습니다. 쉼표는 연결 연산자로 취급되었습니다. 이와 같은 방식으로 이미 쉼표가 사용되고 있는 유니버스에 대해 REPLACE_COMMA_BY_CONCAT 를 No 로 설정하면 이 동작을 유지할 수 있습니다. 현재 버전의 유니버스 디자인 도구에서는 기본적으로 이 매개 변수가 Yes 로 설정되어 있으므로 이와 같은 방식으로 쉼표를 사용하는 식이 연결 구문을 사용하도록 자동 변경됩니다. Yes: 여러 필드 개체가 있는 경우 쉼표가 연결 식으로 바뀝니다. No: 쉼표를 그대로 유지합니다.

3.4.14.28 SELFJOINS_IN_WHERE

SELFJOINS_IN_WHERE = Yes|No

표 63:

값	Yes | No
기본값	No
설명	자체 조인은 일반적으로 FROM 절에 포함됩니다. 따라서 강제로 시스템이 WHERE 절에 모든 자체 조인 조건이 있는 SQL 구문을 생성하도록 할 수 있습니다. 이 매개 변수를 적용하려면 ANSI92 매개 변수를 Yes 로 설정해야 합니다. 목록에 수동으로 매개 변수를 추가하여 활성화해야 합니다. Yes: 자체 조인의 조건은 SQL 쿼리의 WHERE 절에 포함됩니다. No: 자체 조인 구문은 ANSI 규칙에 따라 생성되고 자체 조인 조건은 SQL 쿼리의 FROM 절에 있는 테이블 조인 정의의 ON 절에 포함됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.14.29 SHORTCUT_BEHAVIOR

SHORTCUT_BEHAVIOR = Global|Successive

표 64:

값	Global|Successive
기본값	Successive
설명	바로 가기 조인이 적용되는 방식을 지정합니다. 이전에는 이 매개 변수가 PRM 파일의 GLOBAL_SHORTCUTS 에 나열되었습니다. Yes 에 해당하는 값은 Global 로 변경되었고 No. 에 해당하는 값은 Successive 로 변경되었습니다. Global: 바로 가기 조인을 하나씩 적용하도록 지정합니다. 바로 가기 조인은 다음 바로 가기 조인에 사용되는 조인 경로에서 테이블을 제거하지 않고 하나 이상의 테이블을 실제로 무시하는 경우에만 적용됩니다. Successive: 바로 가기 조인이 모두 적용되도록 지정합니다. 참고: 카티전 곱이 생성되는 경우에는 바로 가기 조인이 적용되지 않습니다.

3.4.14.30 SMART_AGGREGATE

SMART_AGGREGATE = Yes|No

표 65:

값	Yes|No
기본값	No
설명	집계 테이블에 기반을 둔 스마트 측정에 집계 테이블이 사용되는 방식을 결정합니다. 따라서 비율에 기반을 둔 유니버스 개체를 정확하게 집계할 수 있습니다. 기본적으로 시스템은 집계된 테이블로부터 미리 계산된 값을 이용합니다. 이러한 테이블이 일관되지 않고 각 시기마다 다를 경우 이 매개 변수를 사용하여 가장 세부적인 집계 테이블이 사용되도록 해야 합니다. 이 매개 변수는 유니버스 매개 변수 목록에 표시되지 않으며 기본적으로 활성화되지 않습니다. 유니버스 설계자는 매개 변수 목록에 이 매개 변수를 수동으로 삽입한 후 Yes 값을 사용하여 활성화해야 합니다. Yes: 모든 추가 그룹화 집합 쿼리는 집계 테이블에 기초한 스마트 계수에 대한 초기 쿼리의 집계 테이블에 기초해야 합니다. No: 시스템에서 가장 적절한 집계 테이블을 가져옵니다.

3.4.14.31 STORED_PROC_UNIVERSE

STORED_PROC_UNIVERSE = Yes|No

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 66:

값	Yes|No
기본값	No
설명	저장 프로시저를 포함하는 유니버스를 만드는 경우 이 값이 자동으로 Yes 로 설정됩니다. 이 값을 직접 변경하지 마십시오. Yes: 만들거나 편집 중인 유니버스에 저장 프로시저가 포함되어 있습니다. No: 유니버스에 저장 프로시저가 포함되어 있지 않습니다.

3.4.14.32 THOROUGH_PARSE

THOROUGH_PARSE = Yes|No

표 67:

값	Yes|No
기본값	No
설명	개별 개체 구문 분석과 쿼리 창의 기본 구문 분석에 사용되는 방법을 지정합니다. Yes: PREPARE, DESCRIBE 및 EXECUTE 문이 개체의 SQL 구문 분석에 사용됩니다. Prepare+DescribeCol+Execute No: PREPARE 및 DESCRIBE 문이 개체의 SQL 구문 분석에 사용됩니다.

3.4.14.33 TRUST_CARDINALITIES

TRUST_CARDINALITIES = Yes|No

표 68:

값	Yes|No
기본값	No
설명	필요 이상의 결과가 반환되는 경우 SQL 을 최적화할 수 있도록 합니다. Yes: 계수가 포함된 쿼리의 경우 계수를 확장하고 결과 개체에 나타나지 않는 모든 조건은 계수에 대한 잘못된 결과를 반환할 수 있는 테이블이 쿼리에 포함되지 않도록 하위 쿼리로 변환됩니다. No: 최적화가 구현되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.14.34 UNICODE_STRINGS

UNICODE_STRINGS = Yes|No

표 69:

값	Yes|No
기본값	No
설명	현재 유니버스에서 유니코드 문자열을 조작할 수 있는지 여부를 지정합니다. Microsoft SQL Server 및 Oracle 9 에만 적용됩니다. SBO 파일의 데이터베이스 문자 집합이 유니코드로 설정된 경우 NCHAR 및 NVARCHAR 같은 특정 유니코드 열 형식을 처리할 수 있도록 SQL 생성을 수정해야 합니다. Yes: 문자열을 기반으로 하는 조건이 PRM 파일의 UNICODE_PATTERN 매개 변수 값에 따라 SQL 로 형식이 지정됩니다. 예를 들어 MS SQL Server (sqlsrv.prm) 의 경우 UNICODE_PATTERN=N$입니다. Customer_name='Arai' 조건은 Customer_name=N'Arai'가 됩니다. 참고: 유니코드 값에 따라 @Prompt 구문으로 프롬프트를 만들 경우 데이터 형식은 'C'가 아니라 'U'여야 합니다. No: 문자열을 기반으로 하는 모든 조건이 표준 SQL 로 형식이 지정됩니다. 예를 들어, Customer_name='Arai' 조건은 Customer_name='Arai'로 계속 유지됩니다.

3.4.15 PRM 파일에서 설정한 SQL 매개 변수

3.4.15.1 CASE_SENSITIVE

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

표 70:

설명	데이터베이스에서 대소문자 구분 여부를 지정합니다. 이 매개 변수는 Oracle 에서 사용됩니다.
값	YES: 데이터베이스가 대소문자를 구분합니다. NO: 데이터베이스가 대소문자를 구분하지 않습니다.
기본값	NO

3.4.15.2 CHECK_OWNER_STATE

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


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 71:

설명	데이터베이스에서 소유자 이름별 테이블 분류를 지원하는지에 대한 SQL 의 확인 여부를 지정합니다.
값	YES: SQL 에서 데이터베이스의 소유자 이름별 테이블 분류 지원 여부를 확인합니다. NO: SQL 에서 데이터베이스의 소유자 이름별 테이블 분류 지원 여부를 확인하지 않습니다.
기본값	YES

3.4.15.3 CHECK_QUALIFIER_STATE

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

표 72:

설명	데이터베이스에서 한정자별 테이블 분류를 지원하는지에 대한 SQL 의 확인 여부를 지정합니다.
값	YES: SQL 에서 데이터베이스의 한정자별 테이블 분류 지원 여부를 확인합니다. NO: SQL 에서 데이터베이스의 한정자별 테이블 분류 지원 여부를 확인하지 않습니다.
기본값	YES

3.4.15.4 COMMA

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

표 73:

설명	다음과 같은 구문의 개체에 쉼표 대신 사용해야 하는 데이터베이스 연결 연산자를 지정합니다. Tab.Col1, Tab.Col2. 모든 데이터 액세스 드라이버에서 사용되는 매개 변수입니다.
값	||' '|| +' '+
기본값	||' '||
결과	Tab.Col1||' '||Tab.Col2

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.15.5 CONCAT

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

표 74:

설명	연결 연산자를 지정합니다. 모든 데이터 액세스 드라이버에서 사용되는 매개 변수입니다.
값	이중 파이프 (||) 또는 더하기 기호 (+)
기본값	||

3.4.15.6 DATE_WITHOUT_QUOTE

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

표 75:

설명	SQL 구문에서 날짜를 작은 따옴표로 묶을지 여부를 지정합니다. 이 매개 변수는 MS Access 에서 사용됩니다.
값	YES: 날짜를 작은 따옴표로 묶지 않습니다. NO: 날짜를 작은 따옴표로 묶습니다.
기본값	YES

3.4.15.7 DELIMIT_LOWERCASE

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

표 76:

설명	소문자 식별자가 따옴표로 구분되는지 여부를 지정합니다.
값	YES: 소문자 식별자가 따옴표로 구분됩니다. NO: 소문자 식별자가 따옴표로 구분되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.15.8 EXTERN_SORT_EXCLUDE_DISTINCT

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

표 77:

설명	쿼리에 ORDER BY 절이 포함되어 있을 때 응용 프로그램의 SELECT DISTINCT 생성 여부를 지정합니다.
값	YES: 쿼리에 ORDER BY 절이 포함되어 있을 때 SELECT DISTINCT 가 생성되지 않습니다. NO: 쿼리에 ORDER BY 절이 포함되어 있을 때 DISTICT 가 생성됩니다.
기본값	YES

3.4.15.9 GROUPBY_WITH_ALIAS

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

표 78:

설명	데이터베이스에서 SELECT 문에 별칭을 포함한 GROUP BY 절을 작성할 수 있는지 여부를 지정합니다.
값	YES: SELECT 문에 별칭을 포함하는 GROUP BY 절을 작성할 수 있습니다. NO: SELECT 문에서 별칭이 있는 GROUP BY 절을 작성할 수 없습니다.
기본값	YES

3.4.15.10 IDENTIFIER_DELIMITER

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

표 79:

설명	다음 기능을 지정합니다. • BACK_QUOTE_SUPPORTED 매개 변수가 활성화된 경우 공백이나 특수 문자가 포함된 테이블 또는 열 이름을 따옴표로 묶습니다. • DELIMIT_IDENTIFIERS 매개 변수가 활성화된 경우 문자에 관계없이 테이블 또는 열 이름을 따옴표로 묶습니다. 이 매개 변수를 사용하려면 BACK_QUOTE_SUPPORTED 또는 DELIMIT_IDENTIFIERS 를 YES 로 설정해야 합니다. 이 값이 두 매개 변수의 기본값입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

값	"(큰 따옴표): 공백이나 특수 문자가 포함된 테이블 또는 열 이름을 큰 따옴표로 묶습니다. '(작은 따옴표): 공백이나 특수 문자가 포함된 테이블 또는 열 이름을 작은 따옴표로 묶습니다. 이 값은 Microsoft Access 에서만 사용할 수 있습니다.
기본값	"
결과	Table name="My Table"

3.4.15.11 IF_NULL

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

표 80:

설명	매개 변수 두 개를 사용하는 함수를 지정합니다. 첫 번째 매개 변수가 NULL 을 반환하면 두 번째 매개 변수 값이 사용됩니다.
값	데이터베이스에 따라 다릅니다.
기본값	데이터베이스에 따라 다릅니다.

3.4.15.12 OUTERJOINS_COMPLEX

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

표 81:

설명	이 매개 변수는 OUTERJOINS_GENERATION 과 함께 외부 조인 쿼리를 제어합니다.
값	YES NO

3.4.15.13 OUTERJOINS_GENERATION

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

이 매개 변수는 기본 외부 조인 생성 동작을 제어합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 외부 조인 생성이 ANSI92 사양을 준수합니다.
- 외부 조인 생성이 이전 버전의 유니버스 디자인 도구와 동일하게 유지됩니다.

노트

PRM 파일 `OUTERJOINS_GENERATION` 매개 변수는 다음과 같은 방법으로 유니버스 `ANSI92` 설정과 관련됩니다.

- PRM 파일 `OUTERJOINS_GENERATION` 매개 변수가 `ANSI92` 로 설정되고 유니버스 `ANSI92` 설정이 `NO` 로 설정된 경우, PRM 매개 변수가 유니버스 설정을 무시하고 외부 조인은 ANSI92 동작을 준수합니다.
- PRM 파일 `OUTERJOINS_GENERATION` 매개 변수가 `USUAL` 로 설정되면 유니버스 `ANSI92` 설정의 우선 순위가 높으며, 외부 조인의 ANSI92 준수 여부는 유니버스 `ANSI92` 설정이 `YES` 인지 `NO` 인지에 따라 달라집니다.

기억할 사항

`ANSI92` 값은 SQL 생성 최적화에는 유용하지 않은 `REVERSE_TABLE_WEIGHT` 매개 변수를 만듭니다. ANSI92 동작을 준수하는 외부 조인은 SQL 문에서 테이블의 순서를 이끕니다.

표 82:

설명	외부 조인에 대한 SQL 구문을 지정합니다. `ANSI 92` 값을 설정하면 FROM 절에서 외부 조인이 생성됩니다. 다른 값을 설정하면 WHERE 절에서 외부 조인이 생성됩니다. 이 설정을 수정할 경우에는 조인 속성을 검사하여 외부 조인 식이 유효하고 카디널리티가 정확한지 확인해야 합니다. ANSI92 는 조인 구문에서 수동 사용자 지정을 지원하지 않습니다.
값	`OUTERJOINS_GENERATION` 의 기본 값은 다음과 같습니다. • `ANSI92`: 유니버스 디자인 도구의 `ANSI92` 매개 변수 값과 관계없이 기본 외부 조인 동작이 ANSI92 표준을 따릅니다. • `No`: 외부 조인이 지원되지 않습니다. • `USUAL`: 기본 외부 조인 동작이 이전 버전의 유니버스 디자인 도구와 동일합니다. 유니버스 디자인 도구의 `ANSI92` 매개 변수가 `Yes` 로 설정된 경우 이 동작은 무시됩니다. 데이터베이스에 따라 다른 설정을 사용할 수 있습니다. 아래 기본값을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

기본값	ANSI_92: Oracle, MS SQL Server 2005 및 Sybase 의 기본값입니다. DB2: IBM DB2 의 기본값입니다. FULL_ODBC: Microsoft SQL Server 의 기본값입니다. INFORMIX: IBM Informix 의 기본값입니다. INGRES: Teradata 의 기본값입니다. NO: ODBC 의 기본값입니다. USUAL: HP Neoview, Netezza, IBM Red Brick, MS SQL Server 2000 의 기본값입니다.

OUTERJOINS_GENERATION 매개 변수 설정 예제

설정 = USUAL:

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

설정 = DB2:

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

설정 = ODBC:

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

설정 = INFORMIX

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

설정 = FULL-ODBC

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

설정 = ANSI_92:

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


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

Oracle 과 함께 OUTERJOINS 사용

기본 OUTERJOINS_GENERATION 설정은 ANSI92 매개 변수의 유니버스 수준 설정과 상관없이 기존 유니버스 동작에 영향을 줄 수 있습니다.

기존 Oracle 유니버스의 동작이 이전 유니버스 디자인 도구 버전과 일치하도록 설정하려면

1. PRM 파일에서 OUTERJOINS_GENERATION 매개 변수가 USUAL 로 설정되어 있는지 확인합니다.
2. PRM 파일에서 LEFT_OUTER 및 RIGHT_OUTER 매개 변수를 $(+) 로 설정합니다.

이전 버전의 유니버스 디자인 도구에 있는 유니버스 SQL 매개 변수 및 PRM 파일에 대한 자세한 내용은 *Designer* 가이드를 참조하십시오.

3.4.15.14 OVER_CLAUSE

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

표 83:

설명	SAP BusinessObjects 응용 프로그램에서 SQL 을 생성할 때 RISQL 함수를 포함하도록 허용합니다. 데이터베이스에 대해 지원되는 RISQL 함수는 ANALYTIC_FUNCTIONS 매개 변수에 나열됩니다.
값	YES: 응용 프로그램에서 SQL 을 생성할 때 RISQL 함수를 포함할 수 있습니다. NO: 응용 프로그램에서 SQL 을 생성할 때 RISQL 함수를 포함할 수 없습니다.
기본값	YES

3.4.15.15 OWNER

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

표 84:

설명	데이터베이스에서 소유자 이름을 테이블의 접두사로 지원하는지 여부를 지정합니다.
값	YES: 데이터베이스에서 소유자 이름을 테이블 접두사로 사용할 수 있습니다. NO: 데이터베이스에서 소유자 이름을 테이블 접두사로 사용할 수 없습니다.
기본값	YES

3.4.15.16 PREFIX_SYS_TABLE

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


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

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

표 85:

설명	시스템 테이블이 유니버스 디자인 도구에 표시되는지 여부를 지정합니다.
값	`MSys`: MS Access 시스템 테이블이 유니버스 디자인 도구 테이블 탐색기에 표시되지 않습니다. `RBW_`: IBM Red Brick 시스템 테이블이 Universe Designer 테이블 탐색기에 표시되지 않습니다. 값 없음: 데이터베이스 시스템 테이블이 데이터베이스 디자인 도구 테이블 탐색기에 표시됩니다.
기본값	`MSys`: MS Access 의 기본값입니다. `RBW_`: IBM Red Brick 의 기본값입니다.

3.4.15.17 QUALIFIER

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

표 86:

설명	데이터베이스에서 한정자 이름을 테이블의 접두사로 지원하는지 여부를 지정합니다.
값	`YES`: 데이터베이스에서 한정자 이름을 테이블 접두사로 사용할 수 있습니다. `NO`: 데이터베이스에서 한정자 이름을 테이블 접두사로 사용할 수 없습니다.
기본값	RDBMS 종속

3.4.15.18 QUOTE_OWNER

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

표 87:

설명	소유자 이름을 작은 따옴표로 묶어야 하는지 여부를 지정합니다. IBM Informix 에서만 사용됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

값	YES: 작은 따옴표로 묶은 소유자 이름이 테이블 이름 접두사로 사용됩니다. 이 설정은 ANSI 호환 IBM Informix 데이터베이스의 필수 설정입니다. 그렇지 않으면 IBM Informix 에서 소유자 이름이 대문자로 변환됩니다. NO: 작은 따옴표로 묶은 소유자 이름이 테이블 이름 접두사로 사용되지 않습니다.
기본값	YES
결과	SELECT Alias.col(<Alias>는 로컬 별칭) FROM 'Owner'.table.col Alias

3.4.15.19 REFRESH_COLUMNS_TYPE

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

표 88:

설명	열이 새로 고쳐지는 방식을 지정합니다.
값	O: 열이 소유자 이름별로 새로 고쳐집니다. Q: 열이 한정자 이름별로 새로 고쳐집니다. T: 열이 테이블 이름별로 새로 고쳐집니다.
기본값	O: Oracle 의 기본값입니다. Q: IBM Red Brick, Sybase, MS SQL Server 및 MS Access 의 기본값입니다.

3.4.15.20 REMOVE_SEMICOLONS

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


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 89:

설명	SAP BusinessObjects 응용 프로그램의 쿼리 패널이 자유 SQL 의 세미콜론을 제거할지 여부를 지정합니다.
값	YES: 쿼리 패널이 세미콜론을 제거합니다. NO: 쿼리 패널이 세미콜론을 제거하지 않습니다.

3.4.15.21 REVERSE_TABLE_WEIGHT

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

표 90:

설명	테이블이 생성되는 순서를 지정합니다. 이 매개 변수는 Oracle 에서 사용됩니다. 이 매개 변수는 일부 다른 데이터베이스에서도 사용될 수 있으며, YES 값과 NO 값을 바꿀 수도 있습니다. **노트** 이 매개 변수는 Teradata 에서 지원되지 않습니다. **기억할 사항** OUTERJOINS_GENERATION 매개 변수가 ANSI92 로 설정되거나 유니버스 ANSI92 설정이 YES 로 설정된 경우 REVERSE_TABLE_WEIGHT 매개 변수는 SQL 생성 최적화에 영향을 미치지 않습니다.
값	YES: 가장 작은 테이블에서 가장 큰 테이블 순으로 테이블이 생성됩니다. NO: 가장 큰 테이블에서 가장 작은 테이블 순으로 테이블이 생성됩니다.
기본값	YES

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.15.22 UNICODE_PATTERN

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

표 91:

설명	유니버스 SQL 생성 매개 변수 `UNICODE_STRINGS` 가 `YES` 로 설정된 경우에만 적용됩니다. 그런 다음 문자열을 기반으로 하는 모든 조건이 문자열 값으로 형식이 지정됩니다. MS SQL Server 와 Oracle 에서만 사용됩니다.
값	`N$`: MS SQL Server 용 `UNISTR($)` : Oracle 용

3.4.15.23 USER_INPUT_DATE_FORMAT

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

표 92:

설명	SQL 문의 WHERE 절에서 생성되는 기본 날짜 및 시간 형식을 지정합니다.
값	`{\d 'yyyy-mm-dd'}`: ODBC 의 기본 날짜 형식. `'DD-MM-YYYY HH:MM:SS'`: Oracle 의 기본 날짜 및 시간 형식. `'MM/DD/YYYY'`: IBM Informix 의 기본 날짜 형식. `'yyyy-mm-dd HH:mm:ss'`: MS SQL Server 및 대부분의 IBM DB2 서버의 기본 날짜 및 시간 형식. `'mm/dd/yyyy hh:m:s am/pm'`: Sybase 의 기본 날짜 및 시간 형식. `'yyyy-mm-dd'`: Sybase 게이트웨이의 기본 날짜 형식. **노트** ODBC 에서 시간 또는 타임스탬프 변수를 사용해야 하는 경우 `odbc.sbo` 파일에서 기본 날짜 형식 값을 `{\t 'hh:mm:ss'}` 또는 `{\t\s 'yyyy-mm-dd hh:mm:ss'}`로 바꿔야 합니다.
기본값	위의 값 참조

3.4.15.24 USER_INPUT_NUMERIC_SEPARATOR

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


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 93:

설명	생성된 SQL 스크립트에서 사용되는 기본 소수 구분 기호를 지정합니다.
값	'.'(마침표)
기본값	'.'

3.4.15.25 DELIMIT_IDENTIFIERS

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

표 94:

설명	데이터베이스 식별자의 인용 가능 여부를 지정합니다. 식별자가 IDENTIFIER_DELIMITER 매개 변수에 지정된 구분 기호를 사용하여 인용됩니다.
값	YES: 식별자를 인용할 수 있습니다. NO: 식별자를 인용할 수 없습니다.
기본값	YES
결과	Table name="my_table"

3.4.15.26 EXT_JOIN_INVERT

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

표 95:

설명	조인 식에서 외부 조인 기호를 표시할 방식을 지정합니다. IBM DB2, IBM Informix, Oracle 및 Teradata 에서 사용되는 매개 변수입니다.
값	YES: Universe Designer 의 조인 편집 대화 상자에서 외부 조인 확인란을 클릭하면 외부 조인 기호가 조인 식에서 반대 위치에 나타납니다. NO: Universe Designer 의 조인 편집 대화 상자에서 외부 조인 확인란을 클릭하면 외부 조인 기호가 외부 조인을 만든 동일한 위치에 나타납니다.
기본값	YES

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3.4.15.27 KEY_INFO_SUPPORTED

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

표 96:

설명	데이터베이스에서 기본 키 및 보조 키 정의를 검색할 수 있는지 여부를 지정합니다.
값	YES: 데이터베이스에서 기본 키 및 보조 키 정의를 검색할 수 있습니다. 이 매개 변수를 설정하면 Universe Designer 에서 구조 창에 키를 표시할 수 있습니다. NO: 데이터베이스에서 기본 키 및 보조 키 정의를 검색할 수 없습니다.
기본값	YES

3.4.15.28 ORDER_BY_STRINGS

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

표 97:

설명	데이터베이스가 문자열 열을 기반으로 ORDER BY 절을 올바르게 처리할 수 있는지 여부를 지정합니다. 이 매개 변수는 SAP BusinessObjects Data Federator 의 ORDERBYSTRINGS 기능에 해당합니다. 데이터베이스에서 처리할 수 없는 경우 Data Federator 쿼리 서버에서 정렬을 수행합니다.
값	YES: 데이터베이스에서 정렬 처리를 수행할 수 있습니다. NO: 데이터베이스에서 정렬 처리를 수행할 수 없습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4 테이블과 조인을 사용하여 스키마 만들기

4.1 개요

이 장에서는 Web Intelligence 사용자가 보고서 작성을 위해 사용하는 개체를 만드는 데 필요한 모든 SQL 구조가 포함된 스키마를 만들 수 있는 방법에 대해 설명합니다. 이러한 SQL 구조에는 테이블, 열, 조인 및 데이터베이스 함수가 포함됩니다. 정확한 스키마 작성은 모든 최종 사용자의 보고서 요구 사항을 충족하는 유니버스를 만들기 위한 기본입니다.

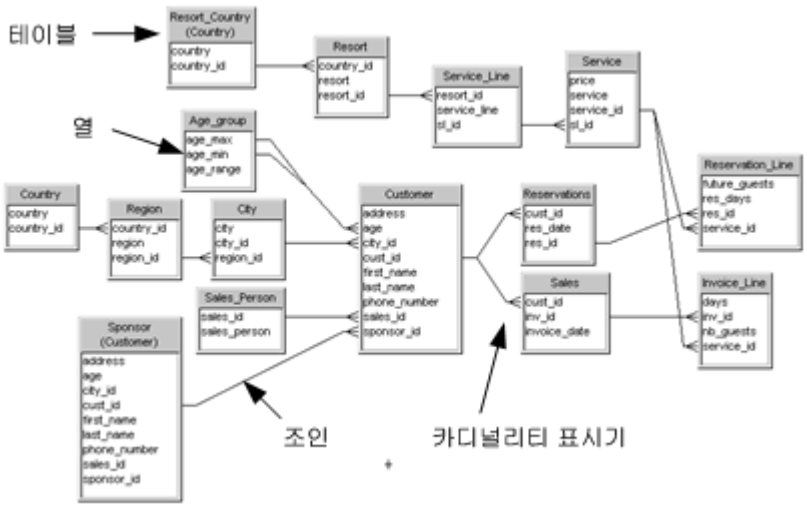
4.2 스키마 정의

스키마는 데이터베이스 구조를 그래픽으로 표현한 것입니다. 유니버스 디자인 도구에서는 유니버스가 나타내는 일부 데이터베이스에 대한 스키마를 만들 수 있습니다.

스키마에는 테이블과 조인이 포함됩니다. 테이블에는 최종 사용자가 보고서를 작성하는 데 사용하는 개체에 매핑되는 열이 포함됩니다. 조인은 테이블을 연결하여 둘 이상의 테이블을 대상으로 실행되는 쿼리에 대하여 정확한 데이터가 반환되도록 합니다.

스키마를 디자인하려면 구조 창에서 테이블 탐색기를 사용하여 대상 데이터베이스의 테이블을 선택합니다. 테이블을 연결하는 조인도 만듭니다. 유니버스에 대한 스키마를 정의한 다음에는 자동 무결성 검사를 사용하여 스키마를 검증할 수 있습니다.

Beach 유니버스에 대한 스키마는 다음과 같이 나타납니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.2.1 성공적인 유니버스의 기반이 되는 스키마 디자인

올바른 스키마 디자인은 성공적인 유니버스 디자인의 필수 요소입니다. 스키마는 최종 사용자들이 보고서를 작성하는 데 필요한 개체와 일치하는 열을 기반으로 하는 테이블로 채워집니다. 이러한 개체는 사용자의 요구 사항에 대한 분석을 통하여 정의되어야 합니다. 데이터베이스에서 사용자가 필요로 하는 개체를 만들 수 있는 테이블을 찾아야 합니다.

4.2.2 스키마 디자인 및 유니버스 생성 프로세스

스키마를 만드는 작업은 유니버스 개발 주기의 구현 단계 중 첫 번째 단계입니다. 사용자 분석 단계와 계획 단계는 모두 유니버스 디자인 도구를 사용하지 않고도 가능하지만, 이 도구를 사용하여 유니버스를 작성하려면 먼저 스키마를 만들어야 합니다.

다음 목록은 일반적인 유니버스 개발 주기에서 스키마 디자인 단계의 위치(구현, 1 단계)를 나타냅니다.

- 준비

1. 사용자 요구 사항 분석
2. 계획

- 유니버스 디자인 도구를 사용하여 구현

1. 스키마 디자인 및 테스트
2. 유니버스 개체 작성 및 테스트
3. 리포지토리를 사용하여 유니버스 배포

- 유지 관리

1. 사용자 요구 사항 또는 데이터 소스의 변경 내용을 기반으로 유니버스 업데이트 및 유지 관리

4.2.3 스키마 디자인의 단계

이 장에서는 스키마 디자인의 다음 단계를 설명합니다.

- 테이블 삽입 및 구성
- 조인 만들기 및 카디널리티 설정
- 루프, 캐즘 트랩 및 팬 트랩과 같은 조인 문제 해결
- 스키마의 무결성 테스트

4.3 테이블 삽입

대상 데이터베이스에서 테이블을 선택하고 구조 창에서 테이블을 나타내는 기호를 삽입하여 스키마 디자인을 시작합니다. 유니버스 디자인 도구에서는 테이블 기호를 간단히 테이블이라고 합니다.

테이블 탐색기를 사용하여 스키마에 삽입할 테이블을 선택합니다. 테이블 탐색기는 대상 데이터베이스에서 사용 가능한 테이블에 대한 트리 보기를 보여주는 독립된 창입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

> **노트**
>
> 테이블을 선택하기 전에 유니버스 생성에 사용할 전략을 지정할 수 있습니다. 이 항목에 대한 자세한 내용은 전략 선택 [페이지 81]을 참조하십시오.

4.3.1 테이블 탐색기 사용

테이블 탐색기는 대상 데이터베이스의 테이블 및 열에 대한 트리 보기를 보여주는 독립된 창입니다. 테이블 탐색기를 사용하여 스키마에 삽입할 테이블을 데이터베이스에서 확인하고 선택합니다. 테이블 이름 옆에 있는 노드를 확장하면 테이블의 열이 표시됩니다.

4.3.1.1 테이블 탐색기 활성화

테이블 탐색기는 기본적으로 보이지 않습니다. 구조 창에 테이블을 추가하고자 할 때 테이블 탐색기를 활성화해야 합니다. 아래 나열된 방법 중 하나를 사용하여 테이블 탐색기를 활성화할 수 있습니다.

테이블 탐색기를 활성화하려면

- 삽입 > 테이블 을 선택합니다.
  또는
- 구조창의 빈 공간을 두 번 클릭합니다.
  또는
- 테이블 탐색기 단추를 클릭합니다.
  테이블 탐색기 창이 구조 창에 나타납니다.

4.3.1.2 테이블 탐색기에서 테이블 삽입

다음 방법 중 하나로 테이블 탐색기를 사용하여 하나 또는 여러 개의 테이블을 삽입할 수 있습니다.

하나의 테이블 삽입

하나의 테이블을 삽입하려면

- 테이블을 클릭하고 삽입 단추를 클릭합니다.
  또는
- 테이블을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 삽입을 선택합니다.
  또는
- 테이블을 두 번 클릭합니다.
  또는

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 테이블을 클릭하고 구조 창으로 끌어 놓습니다.
  테이블이 구조 창에 나타납니다.

여러 개의 테이블 삽입

여러 개의 테이블을 삽입하려면

1. Ctrl 키를 누른 상태에서 각 테이블을 클릭합니다.
   또는
2. 첫 번째 테이블을 클릭하고 Shift 키를 누른 상태에서 마지막 테이블을 클릭하여 연속적으로 테이블 블록을 선택합니다.
   여러 개의 테이블이 선택됩니다.
3. 삽입 단추를 클릭합니다.
   또는
   테이블을 구조 창으로 끌어 놓습니다.
   또는
   선택된 테이블을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 삽입을 선택합니다.
   해당하는 모든 열이 포함된 각 테이블이 구조창에 나타납니다. 유니버스에 삽입한 모든 테이블은 테이블 탐색기에서 테이블 이름 옆에 확인 표시가 함께 나타납니다.

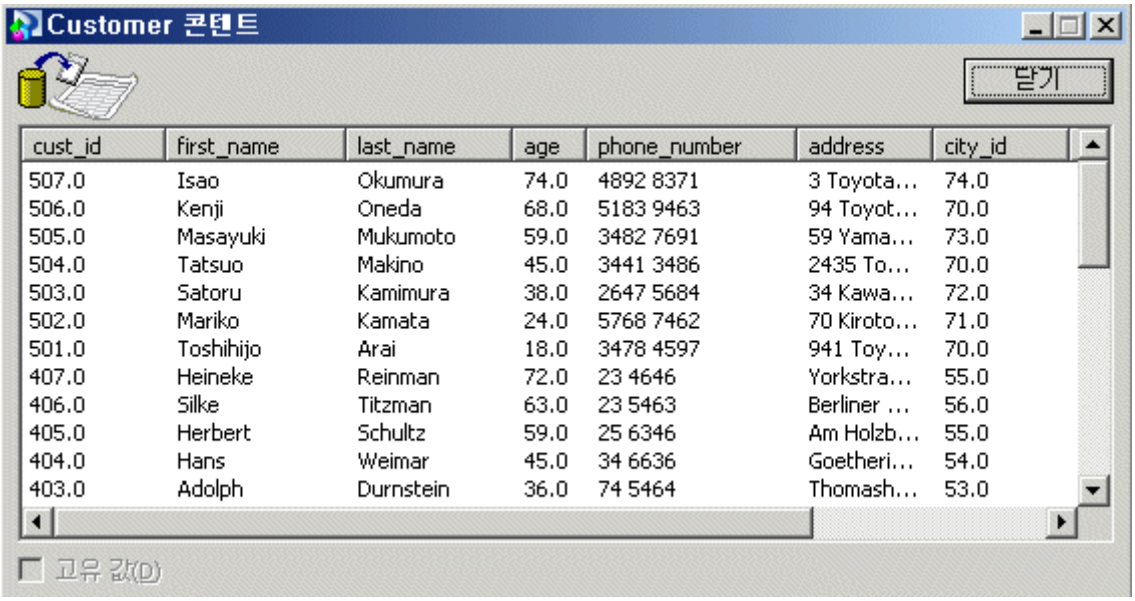
4.3.1.3 테이블 탐색기에서 데이터 보기

테이블 탐색기를 사용하여 테이블 또는 개별 열에 포함되어 있는 데이터를 볼 수 있습니다.

테이블 탐색기에서 데이터를 보려면

1. 테이블 탐색기에 있는 테이블을 마우스 오른쪽 단추로 클릭합니다.
   또는
   테이블 탐색기에서 테이블 노드를 확장하고 테이블 열을 마우스 오른쪽 단추로 클릭합니다.
2. 상황에 맞는 메뉴에서 테이블 값 보기를 선택합니다.
   또는
   상황에 맞는 메뉴에서 열 값 보기를 선택합니다.
   테이블 또는 열에 포함된 데이터가 나열된 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



cust_id	first_name	last_name	age	phone_number	address	city_id
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 Holzb...	55.0
404.0	Hans	Weimar	45.0	34 6636	Goetheri...	54.0
403.0	Adolph	Durnstein	36.0	74 5464	Thomash...	53.0

➡ 팁

열이 너무 좁아서 전체 행 값을 볼 수 없는 경우, `Ctrl+Shift` 키를 누른 상태에서 '`+`' 키를 눌러 열 너비를 늘릴 수 있습니다.

4.3.1.4 테이블 탐색기 성능 최적화

테이블 탐색기에서 구조 창에 테이블을 삽입하는 데 걸리는 시간은 다음 요인에 따라 달라질 수 있습니다.

표 98:

테이블 삽입 시간 지연 요인	테이블 삽입 최적화 방법
데이터베이스에 많은 수의 테이블이 있습니다. 유니버스 디자인 도구에서 시스템 카탈로그를 쿼리하는 경우, 카탈로그가 매우 크면 테이블 검색 작업이 느려질 수 있습니다.	삽입하고자 하는 테이블을 사용하여 별도의 데이터베이스 계정으로 데이터 웨어하우스를 만듭니다. 새 웨어하우스에 대한 연결을 만듭니다.
조인을 자동으로 삽입하고 삽입하는 테이블의 카디널리티를 검사합니다.	테이블만 삽입합니다. 다음과 같이 할 수 있습니다. 1. 도구 > 옵션을 선택합니다. 옵션 대화 상자가 나타납니다. 2. 데이터베이스 탭을 클릭합니다. 데이터베이스 페이지가 나타납니다. 3. 테이블과 함께 조인 추출 및 조인에서 카디널리티 검색 확인란의 선택을 취소합니다. 4. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

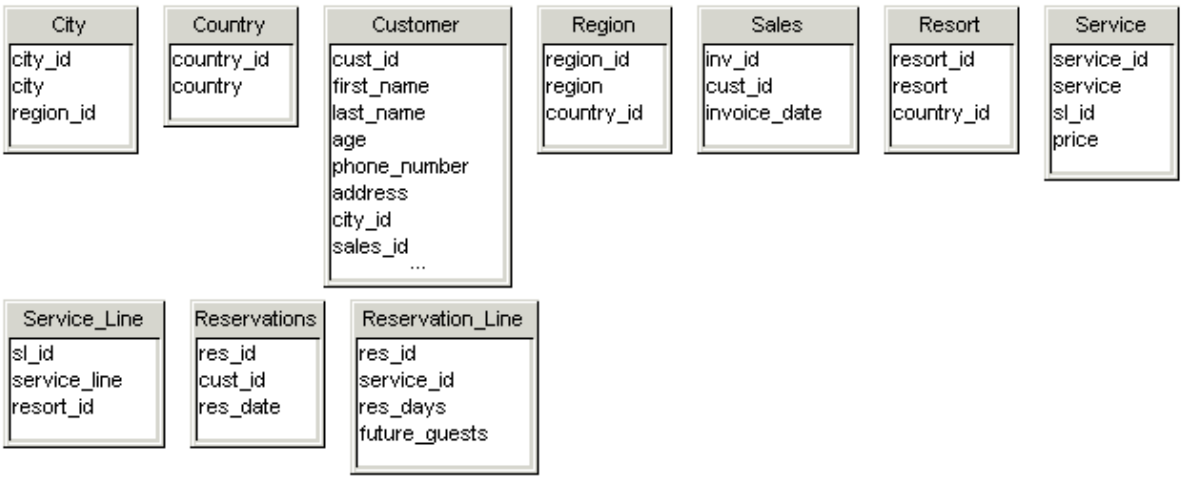
4.3.2 구조 창에서 테이블 정렬

테이블을 수동으로 다시 정렬하여 조인 생성을 시작하기 전에 구조 창에서 테이블을 자동으로 정렬하여 초기 스키마를 정돈할 수 있습니다.

4.3.2.1 구조 창에서 자동으로 테이블 정렬

테이블을 자동으로 정렬하려면

- 보기 > 테이블 정렬을 선택합니다.
  테이블이 순서대로 정렬됩니다.



4.4 파생 테이블 사용

파생 테이블은 유니버스 스키마에서 정의된 테이블입니다. 파생 테이블의 개체를 만드는 방식은 다른 테이블과 같습니다. 파생 테이블은 유니버스 수준에서 SQL 쿼리로 정의되어 유니버스 디자인 도구에서 논리 테이블로 사용될 수 있습니다.

파생 테이블을 사용할 경우 다음과 같은 이점이 있습니다.

- 분석용 문서에 반환되는 데이터의 양이 줄어듭니다.
  파생 테이블에 복합 계산 및 함수를 포함시킬 수 있습니다. 이러한 작업은 결과 집합이 문서에 반환되기 전에 수행되므로 시간이 절약되며 보고서 수준에서 많은 양의 데이터에 대하여 복잡한 분석을 수행할 필요가 줄어듭니다.
- 데이터베이스 요약 테이블에 대한 유지 관리가 줄어듭니다.
  파생 테이블은 경우에 따라 집계 인식을 사용하여 유니버스로 통합되며 복잡한 계산 결과를 포함하는 통계 테이블을 대체할 수 있습니다. 이러한 집계 테이블은 유지 관리 비용이 높고 자주 새로 고쳐야 합니다. 파생 테이블은 그와 동일한 데이터를 반환할 뿐만 아니라 실시간 데이터 분석을 제공할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

파생 테이블은 데이터베이스 뷰와 비슷하지만 파생 테이블의 SQL 에는 프롬프트가 포함될 수 있다는 이점이 있습니다.

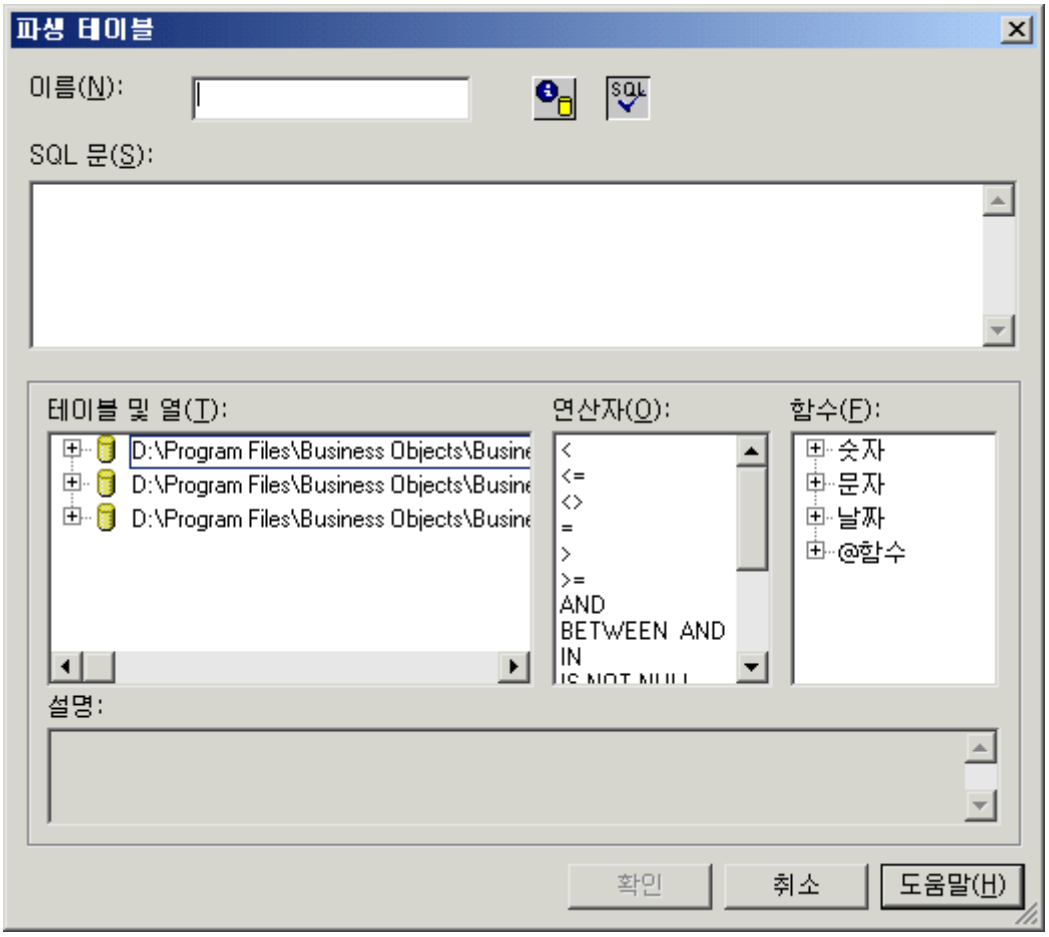
4.4.1 파생 테이블 추가, 편집 및 삭제

파생 테이블은 일반 데이터베이스 테이블과 완전히 동일한 방식으로 유니버스 디자인 도구 스키마에 나타나지만, 테이블을 만드는 워크플로는 다릅니다. 다음 단원에서는 파생 테이블을 추가, 편집 및 삭제하는 방법에 대해 설명합니다.

4.4.1.1 파생 테이블 추가

파생 테이블을 추가하려면

1. 삽입 메뉴에서 파생 테이블을 클릭합니다.
   파생 테이블 대화 상자가 나타납니다.



2. 테이블 이름 상자에 테이블 이름을 입력합니다.
3. 테이블 이름 상자 아래에 있는 상자에 테이블 SQL 을 작성합니다.
   SQL 은 직접 입력하거나 테이블 및 열, 연산자 및 함수 상자를 사용하여 작성할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4. 확인을 클릭합니다.
   파생 테이블이 실제 데이터베이스 테이블과 함께 스키마에 나타납니다.
5. 일반 테이블과 동일하게 파생 테이블 열을 기준으로 개체를 만듭니다.

4.4.1.2 파생 테이블 편집

파생 테이블을 편집하려면

1. 유니버스 디자인 도구 스키마에서 테이블을 마우스 오른쪽 단추로 클릭하고 바로 가기 메뉴에서 파생 테이블 편집을 선택합니다.
2. 파생 테이블을 편집한 다음 확인을 클릭합니다.

4.4.1.3 파생 테이블을 삭제하려면

1. 스키마 창에서 삭제할 파생 테이블을 선택합니다.
2. Delete 키를 누릅니다.

4.4.1.4 예: 파생 테이블 만들기

예

서버 정보를 반환하는 파생 테이블 만들기

이 예제에서는 사용자가 보고서에 데이터베이스 서버에 대한 정보를 추가할 수 있도록 하는 개체를 만듭니다. servername 및 version 이라는 두 개체를 만드는데, 이들 개체는 SQL Server 데이터베이스에서 실행되는 유니버스의 기본 변수인 @@SERVERNAME 및 @VERSION 값을 반환합니다.

다음을 수행합니다.

1. 삽입 메뉴에서 파생 테이블을 선택합니다.
   파생 테이블 대화 상자가 나타납니다.
2. 테이블 이름 상자에 **<serverinfo>** 를 입력합니다.
3. SQL 상자에 `Select @@SERVERNAME as servername, @@VERSION as version` 이라는 SQL 을 입력합니다.

   노트

   SQL 에서 모든 파생 열에 별칭을 제공해야 합니다. 유니버스 디자인 도구는 이러한 별칭을 사용하여 파생 테이블의 열 이름을 지정합니다.

4. 확인을 클릭합니다.
   파생 테이블 serverinfo 가 유니버스 디자인 도구 스키마에 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5. 서버 정보라는 클래스를 만들고, serverinfo 파생 테이블의 servername 및 version 열과 테이블의 열을 기준으로 클래스 아래 두 개의 차원 개체를 추가합니다. serverinfo 테이블은 다른 일반 데이터베이스 테이블과 같이 테이블 목록에 나타나고, 해당 열은 일반 테이블 열과 같이 열 목록에 나타납니다.

이제 사용자는 보고서에 servername 및 version 개체를 포함시킬 수 있습니다.

예

각 국가의 지역 수 표시

이 예제에서는 각 국가의 지역 수를 표시하는 테이블을 만듭니다. SQL 은 다음과 같습니다.

```
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
```

이 경우 계산을 포함하는 열 별칭을 지정하는 것이 중요합니다. 유니버스 디자인 도구는 이러한 별칭을 파생 테이블의 열 이름으로 사용합니다. 이 경우 테이블에는 country 와 number_of_regions 라는 두 개의 열이 있습니다.

4.5 중첩된 파생 테이블

중첩된 파생 테이블('파생 테이블의 파생 테이블'이라고도 함)은 적어도 하나 이상의 기존 파생 테이블에서 파생된 테이블입니다. 중첩된 파생 테이블은 데이터베이스 테이블을 참조할 수도 있습니다.

파생 테이블 편집기를 사용하여 SQL 식을 입력하고 파생 테이블과 필요한 경우 데이터베이스의 실제 테이블을 선택하여 중첩된 파생 테이블을 만들 수 있습니다. 파생 테이블에 대한 SQL 식은 보고서가 생성될 때 중첩된 파생 테이블에 대한 SQL 식에 삽입됩니다.

4.5.1 파생 테이블 편집기 사용

파생 테이블 편집기를 사용하면 파생 테이블 또는 중첩된 파생 테이블을 정의할 수 있습니다. `SQL` 식을 입력하고 편집기에서 개체(테이블, 파생 테이블, 열, 함수)를 두 번 클릭하면 파생 테이블이나 중첩된 파생 테이블에 대한 `SQL` 식을 만들 수 있습니다.

중첩된 파생 테이블에서 파생 테이블을 참조하려면 `@DerivedTable` 함수를 사용합니다.

- 함수 `@DerivedTable` (Derived_table_name) 은 파생 테이블 편집기의 함수 카탈로그에 포함됩니다.
- 파생 테이블 편집기 창의 아래쪽 가운데에 기존의 파생 테이블 및 중첩된 파생 테이블이 표시됩니다. 이 창은 유니버스에 파생 테이블이 있을 때만 표시됩니다.

무결성 검사를 클릭하면 파생 테이블 및 중첩된 파생 테이블에 대해 다음과 같은 수행할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 참조된 파생 테이블이 제거될 때 파생 테이블에 미치는 영향을 감지합니다.
- 순환 참조를 확인합니다.
- 개체 정의(`SELECT` 및 `WHERE`) 내에 `@DerivedTable()` 이 있는지 확인합니다. 이것은 허용되지 않습니다.

4.5.2 중첩된 파생 테이블 만들기

파생 테이블 만들 때와 같은 방법으로 중첩된 파생 테이블을 만듭니다. 파생 테이블을 추가하고 이름을 바꿀 때와 같은 방법으로 중첩된 파생 테이블을 추가하고 이름을 바꿀 수 있습니다.

중첩된 파생 테이블을 만들려면 다음을 수행하십시오.

1. 샘플 디렉터리(`Business Objects\BusinessObjects Enterprise 12\Samples\en\UniverseSamples`)에서 유니버스(*.unv)를 엽니다.
2. 유니버스 구조 창에서 마우스 오른쪽 단추를 클릭하고 상황에 맞는 메뉴에서 파생 테이블을 선택합니다.
   파생 테이블 편집기가 열리고 파생 테이블 편집기 아래쪽의 가운데 창에 사용 가능한 파생 테이블이 나열됩니다.
3. 중첩된 파생 테이블의 이름을 입력합니다.
4. `SQL` 식을 입력합니다. 전체 텍스트를 입력하거나 편집기 도우미를 사용할 수 있습니다.
5. 개체(테이블, 파생 테이블, 열, 함수)를 두 번 클릭합니다.
6. `@DerivedTable` 함수의 `@DerivedTable`(Derived_table_name) 구문을 사용하여 파생 테이블을 선택합니다.
7. 구문 검사를 클릭하여 파생 테이블의 구문을 검사하고 가능한 오류를 수정한 후 중첩된 파생 테이블의 유효성을 검사합니다.
   중첩된 파생 테이블이 유니버스에 추가됩니다.
8. 확인을 클릭하여 중첩된 파생 테이블의 유효성을 검사합니다.
   구조 창에 중첩된 파생 테이블이 나타납니다. 파생 테이블과 중첩된 파생 테이블은 실제 데이터베이스 테이블을 나타내는 테이블보다 더 옅은 색으로 표시됩니다.

> **노트**
>
> 테이블 값을 표시하려면 다른 테이블을 마우스 오른쪽 단추로 클릭합니다.

유니버스에 중첩된 파생 테이블이 생성되었습니다.

4.5.3 중첩된 파생 테이블 이름 바꾸기

파생 테이블의 이름을 바꾸면 해당 이름을 참조하는 다른 모든 파생 테이블로 새 이름이 전파되고 업데이트됩니다.

4.6 입력 열이 있는 테이블 사용

입력 열을 포함하는 테이블을 유니버스에 삽입할 경우 Web Intelligence 또는 Query as a Web Service 사용자가 입력 열을 계산하는 데 필요한 값을 선택하거나 입력해야 합니다. 입력 열은 값에 바인딩됩니다. 입력 열의 데이터가 모두 원본 데이터베이스에서 사용할 수 있는 것은 아니며, 사용 가능한 데이터는 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 유니버스를 만들 때 하드 코딩한 값
- 최종 사용자가 프롬프트 다음에 입력했거나 목록에서 선택한 값
- 다른 테이블과의 조인을 통해 제공되는 값

입력 열이 있는 테이블은 데이터베이스 연결이 Business Objects Data Federator 서버인 경우에만 지원됩니다.

입력 열 계산에 사용할 수 있는 조인을 분석할 경우 다음 사항에 주의하십시오.

- 계산 알고리즘에서는 단순 조인만 처리됩니다.
- 왼쪽 열이나 오른쪽 열이 여러 개인 조인과 같은 복합 조인은 허용되지 않습니다.
- Equal(=) 또는 IN(INLIST) 연산자가 있는 조인만 처리됩니다. Between 같은 연산자는 입력 열 계산에 사용할 수 없습니다.

입력 열이 있는 테이블은 구조 창에서 입력 열 옆에 화살표가 나타나고, 테이블 탐색기 창에서 입력 열을 특정 아이콘으로 식별합니다.

입력 열이 있는 테이블을 삽입할 때 입력 열 편집기를 사용하여 설정을 편집합니다.

> **노트**
>
> 테이블의 입력 열에 대한 기본값은 유니버스에 테이블을 추가할 때 지정해야 합니다.

이 기능은 다음 제품과 구성 요소에서 사용할 수 있습니다.

- Web Intelligence
- Query as a Web Service

관련 정보

하드 코딩된 값 목록 정의 [페이지 130]
사용자가 입력하거나 선택하는 값 목록을 정의하려면 [페이지 131]

4.6.1 하드 코딩된 값 목록 정의

데이터베이스에는 하나 이상의 입력 열이 있는 테이블이 최소한 하나 포함됩니다.

하드 코딩된 값 목록은 입력 열의 값을 결정하는 테이블의 입력으로 사용됩니다. 최종 사용자는 어떤 값도 입력하지 않습니다. 값 목록을 정의하려면 다음 단계를 따르십시오.

1. 데이터베이스에서 테이블을 선택하여 유니버스 디자인 도구의 구조 창에 추가합니다.
   입력 열 편집기가 나타납니다.
2. 입력 열 편집기에서 매개 변수를 클릭합니다.
3. 값 필드에 값이나 값 목록을 입력합니다. 각 값은 큰따옴표로 묶고 세미콜론(;)으로 값을 구분합니다.
   값이 값 열에 나타납니다.
4. 다음 실행 목록에서 이 값 사용을 선택했는지 확인합니다.
   이 값 사용이 다음 실행 열에 나타납니다.
5. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

테이블이 유니버스 디자인 도구의 구조 창에 나타납니다. 입력 열은 화살표로 식별됩니다.

4.6.2 사용자가 입력하거나 선택하는 값 목록을 정의하려면

데이터베이스에는 하나 이상의 입력 열이 있는 테이블이 최소한 하나 포함됩니다.

사용자는 입력 열의 값을 결정하는 테이블에 사용할 값을 입력하거나 값 목록에서 선택할 수 있습니다. 스키마의 입력 열 테이블에 사용할 값을 정의하려면 다음 단계를 따르십시오.

1. 데이터베이스에서 테이블을 선택하여 유니버스 디자인 도구의 구조 창에 추가합니다.
2. 입력 열 편집기에서 매개 변수를 클릭합니다.
3. 다음 실행 목록에서 값에 대한 프롬프트를 표시합니다.를 클릭합니다.

   Web Intelligence 또는 Query as a Web Service 쿼리가 실행될 때 사용자에게 연관된 값 목록에서 값을 선택하라는 메시지가 나타납니다.
4. 프롬프트 레이블 편집 필드에서 최종 사용자에게 나타나는 기본 프롬프트를 편집합니다.
5. 유니버스 개체 찾아보기를 클릭하고 유니버스에서 값 목록을 선택합니다.

   설정에 추가한 값 목록에서 개체를 제거하려면 선택한 개체 창에서 개체를 클릭하고 지우기를 클릭합니다.
6. 확인을 클릭합니다.

테이블이 유니버스 디자인 도구의 구조 창에 나타납니다. 입력 열은 화살표로 식별됩니다. 테이블 탐색기에서 입력 열은 빨간색 아이콘으로 식별됩니다.

4.7 조인 정의

스키마에 둘 이상의 테이블을 삽입한 다음에는 관련 테이블 사이에 조인을 만들어야 합니다. 조인은 여러 테이블의 데이터를 의미 있게 결합시키는 역할을 하므로 스키마에서 테이블만큼 중요합니다.

4.7.1 조인 정의

조인은 서로 별개이지만 연관성이 있는 테이블의 데이터를 연결하는 조건입니다. 테이블은 대개 상위-하위 관계를 갖습니다. 쿼리에 조인이 포함되지 않을 경우 데이터베이스는 쿼리 테이블에 가능한 모든 행 조합이 포함된 결과 집합을 반환합니다. 이러한 결과 집합을 카티전 곱이라고 하며 거의 유용하지 않습니다.

예를 들어, 행 수가 각각 100 개와 50 개인 두 테이블을 참조하는 쿼리의 카티전 곱에는 5000 개의 행이 있습니다. 많은 테이블이 포함된 큰 데이터베이스나 쿼리의 경우 카티전 곱은 다루기 어려울 정도로 커집니다. 유니버스 디자인 도구에서 조인은 스키마의 테이블을 연결하는 선으로 표현됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.7.2 스키마에서 조인을 사용하는 이유

조인을 사용하면 여러 테이블에서 데이터를 반환하는 쿼리가 부정확한 결과를 반환하지 않도록 할 수 있습니다. 두 테이블 사이의 조인은 두 테이블이 모두 쿼리에 포함될 때 데이터가 반환되는 방식을 정의합니다.

스키마의 각 테이블은 사용자의 요구 사항과 일치하는 하나 이상의 열 데이터를 포함합니다. 프로덕션 유니버스에서 Web Intelligence 사용자는 테이블 간 가능한 모든 조합을 통해 데이터를 반환하는 서로 다른 많은 개체(각각 하나의 열에 해당)를 결합하여 쿼리를 실행하고자 할 수 있습니다.

스키마의 모든 테이블을 조인으로 연결하면 여러 테이블의 열 데이터가 쿼리에서 조합 가능한 수를 제한할 수 있습니다. 조인은 테이블 사이의 열 조합을 대응 열 또는 공통 열로 제한합니다. 이렇게 하면 대응되는 의미가 없는 열의 정보를 포함하는 결과 데이터가 반환되지 않습니다.

> **노트**
>
> 조인은 항상 구조 창에서 만들어야 합니다. 개체에 대해 Where 절에서 수동으로 정의된 조인과 같이 구조 창에서 만들어지지 않은 조인은 런타임에 생성되므로 유니버스 디자인 도구에서 무결성 검사 및 컨텍스트 검색 대상으로 간주되지 않습니다. 디자인 타임에는 이러한 프로세스를 위한 정보가 필요합니다. 컨텍스트와 유니버스 무결성은 이 장의 뒷부분에서 다룹니다.

4.7.3 조인 암시를 수행하는 SQL

기본적으로 유니버스 디자인 도구는 테이블의 일치하는 열 또는 공통 열에 대한 참조를 통해 WHERE 절에서 암시적으로 조인을 지정합니다.

일반적으로 조인되는 각 테이블 쌍마다 하나의 WHERE 절이 존재합니다. 따라서 4 개의 테이블이 조인으로 연결되는 경우 3 개의 WHERE 조건이 필요합니다.

조인으로 연결된 두 테이블이 포함된 쿼리를 실행하여 얻는 결과는 조합된 모든 테이블의 열이 있는 단일 테이블입니다. 이 테이블의 각 행은 공통 열에 대해 대응하는 값이 있는 서로 다른 입력 테이블의 행 데이터를 포함합니다.

4.7.3.1 ANSI 92 지원

대상 RDBMS 가 ANSI 92 를 지원하는 경우 유니버스 매개 변수(파일 > 매개 변수 > 매개 변수) ANSI92 를 Yes 로 설정하여 스키마에서 작성된 조인에 대한 ANSI 92 지원을 활성화할 수 있습니다. 유니버스가 조인에 대한 ANSI 92 표준을 지원하는 경우 새로 만든 조인이 FROM 절에서 지정됩니다. 열에서 유추하여 FROM 절에 포함시킬 개체를 선택할 수도 있습니다. ANSI 92 지원은 유니버스에서 ANSI 92 조인 형식 지원 [페이지 144] 단원에서 설명합니다.

4.7.4 조인되지 않아야 하는 테이블

유니버스에 대해 실행된 Web Intelligence 쿼리의 개체에 의해 생성된 SQL 에 유추된 스키마의 모든 테이블을 조인해야 합니다. 여기에서 유일한 예외는 다음과 같은 유형의 테이블입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 각 용도별로 별칭이 지정된 스키마의 기본 테이블. 이러한 테이블은 이름 변경 또는 조인 문제 해결 목적을 위해 별칭을 만들어 둔 원본 테이블입니다. 이러한 기본 테이블은 일반적으로 개체 정의에서 사용되지 않습니다.
- 집계 인식 구문의 대상인 테이블(상황에 따라 고려할 필요가 있음). 예를 들어, 예제 efashion 유니버스(이름이 "Agg_"로 시작)의 집계 테이블은 스키마의 어느 테이블과도 조인되지 않습니다.

4.7.5 기본 및 외래 키 조인

일반적으로 한 테이블의 기본 키와 다른 테이블의 외래 키 사이에 조인을 만듭니다. 두 기본 키 사이에서도 조인을 만들 수 있습니다. 적어도 조인의 한쪽에 테이블의 기본 키가 포함되지 않는 조인은 매우 특이한 경우입니다.

데이터베이스에서 각 키가 구성되는 방식을 이해할 필요가 있습니다. 여러 열로 구성된 키는 조인에 대한 카디널리티를 설정하는 방법에 영향을 줄 수 있으며, 이에 따라 스키마의 컨텍스트를 설정하는 방식이 영향을 받을 수 있습니다.

컨텍스트 검색 및 사용은 조인 문제 검색 및 해결 [페이지 180]에서 설명합니다.

4.7.5.1 키 표시

구조 창에서 모든 테이블의 기본 키와 외래 키를 표시할 수 있습니다. 키 열은 키가 포함된 각 테이블에 밑줄과 함께 나타납니다. 키를 표시할 옵션을 선택할 때는 구조를 새로 고쳐야만 키가 밑줄과 함께 나타납니다.

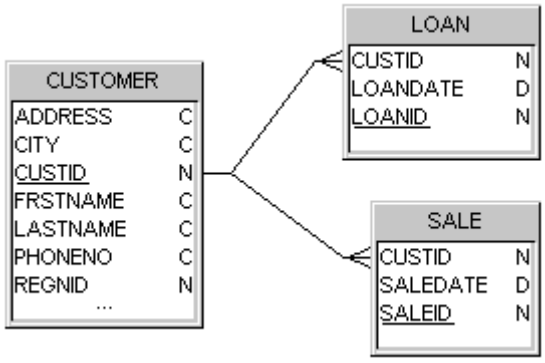
키 열을 밑줄과 함께 표시하는 기능은 대상 데이터베이스에서 정의된 기본 키에 따라 결정됩니다.

> **노트**
>
> 밑줄이 그어진 키 열을 표시할 때 정보는 .UNV 파일에 저장됩니다. 이 정보는 유니버스를 중앙 관리 서버(CMS) 리포지토리로 내보낼 경우 손실됩니다. 따라서 유니버스를 가져올 때마다 키를 다시 표시해야 합니다.

키를 표시하려면

1. 도구 > 옵션을 선택합니다.
   옵션 대화 상자의 일반 페이지가 열립니다.
2. 그래픽 탭을 클릭합니다.
   그래픽 페이지가 나타납니다.
3. 열 그룹 상자에서 키에 밑줄 표시 확인란을 선택합니다.
4. 확인을 클릭합니다.
   키 열이 밑줄과 함께 나타나도록 하려면 구조를 새로 고쳐야 합니다.
5. 보기 > 구조 새로 고침을 선택합니다.
   데이터베이스 구조가 새로 고쳐집니다. 스키마에서 키 열은 다음과 같이 밑줄로 그어져 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



4.7.6 조인의 카디널리티 이해

카디널리티는 한 테이블에서 몇 개의 행이 다른 테이블의 행과 대응하는지 알려줌으로써 두 테이블 사이의 조인을 설명합니다. 이 정보는 조인 문제를 검색하고 대상 RDBMS 구조의 제한 사항을 해결하기 위한 컨텍스트를 만드는 데 매우 중요합니다.

스키마에서 각 조인에 대한 카디널리티를 설정해야 합니다. 유니버스 디자인 도구에서는 카디널리티를 자동으로 검색하고 설정할 수 있지만, 조인되는 키의 특성을 고려하여 항상 카디널리티를 수동으로 확인해야 합니다.

카디널리티 설정 및 사용은 카디널리티 사용 [페이지 162] 단원에서 설명합니다.

4.7.7 조인 만들기

다음과 같은 여러 가지 방법으로 유니버스 디자인 도구에서 조인을 만들 수 있습니다.

- 스키마에서 수동으로 조인 연결
- 조인 속성 직접 정의
- 자동으로 검색된 조인 선택
- 테이블 삽입 시 자동으로 조인 생성

각각의 자세한 방법은 다음 설명을 참조하십시오.

4.7.7.1 스키마에서 수동으로 조인 연결

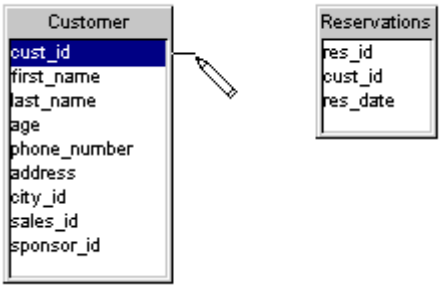
마우스를 사용하여 한 테이블의 열에서 다른 테이블의 대응하는 열로 선을 연결함으로써 테이블 사이에서 각각의 조인을 그래픽 방식으로 만들 수 있습니다.

수동으로 연결하여 조인을 만들려면

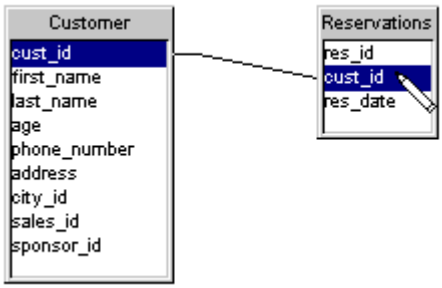
1. 포인터를 조인의 한쪽 끝이 될 열 위로 가져갑니다.
   포인터가 손 모양으로 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 마우스 왼쪽 단추를 클릭한 상태로 유지합니다.
   열이 강조 표시됩니다.
3. 마우스를 조인의 다른쪽 끝이 될 다른 테이블의 열로 끕니다.
   끌 때 포인터는 연필 모양으로 바뀝니다.



4. 연필 모양을 대상 열 위로 가져갑니다.
   대상 열이 강조 표시됩니다.



5. 마우스 단추를 놓습니다.
   두 테이블 사이에 조인이 만들어집니다.
6. 새 조인을 두 번 클릭합니다.
   조인 편집 대화 상자가 나타납니다. 조인 속성이 나열됩니다. 카디널리티 및 조인 유형을 포함하여 조인에 대해 설정할 수 있는 속성은 조인 속성 [페이지 139] 단원에서 설명합니다.
7. 조인에 대한 속성을 입력하고 선택합니다.
8. 확인을 클릭합니다.

4.7.7.2 조인 속성 직접 정의

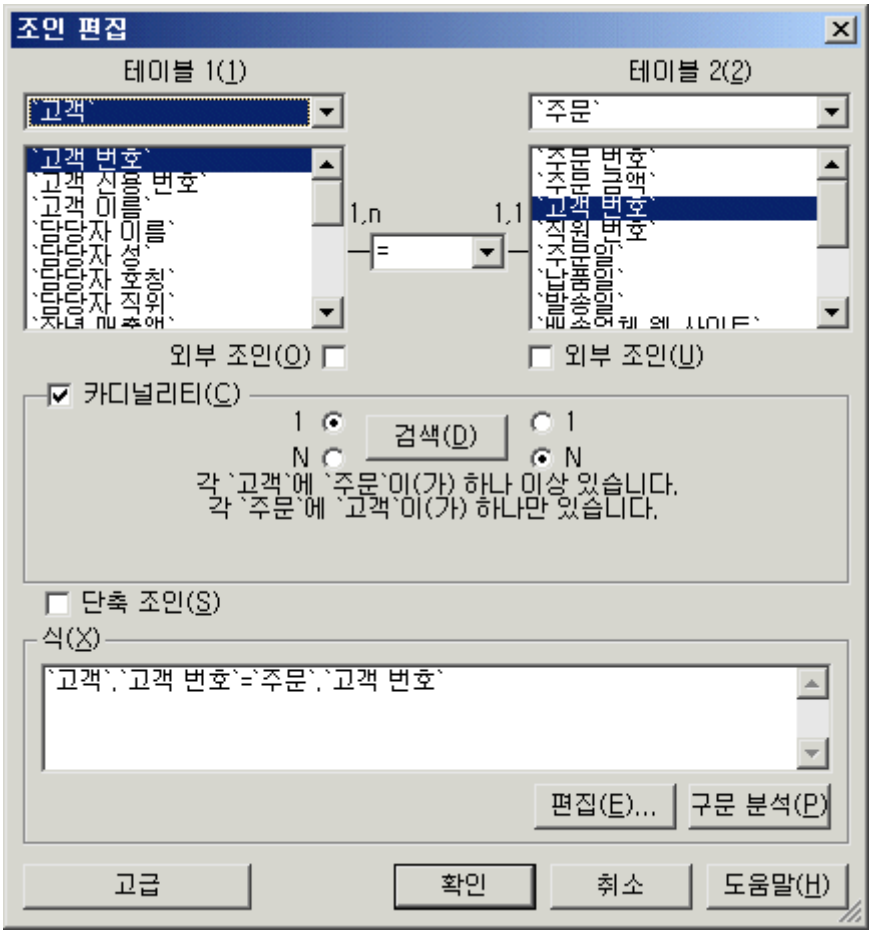
조인 편집 대화 상자에서 조인 속성을 직접 정의하여 조인을 만들 수 있습니다.

조인을 직접 만들려면

1. 삽입 > 조인을 선택합니다.
   또는
   조인 삽입 단추를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조인 편집 대화 상자가 나타납니다.



2. Table1 드롭다운 목록에서 테이블을 선택합니다.
   선택한 테이블의 열이 테이블 이름 아래 목록 상자에 나타납니다.
3. 새 조인의 한쪽 끝이 될 열의 이름을 클릭합니다.
4. Table2 드롭다운 목록에서 테이블을 선택합니다.
   선택한 테이블의 열이 테이블 이름 아래 목록 상자에 나타납니다.
5. 새 조인의 다른쪽 끝이 될 열의 이름을 클릭합니다.
   조인 연산자, 카디널리티 및 조인 유형을 포함하여 조인에 대해 설정할 수 있는 속성은 조인 속성 [페이지 139] 단원에서 설명합니다.
6. 조인에 대한 속성을 입력하고 선택합니다.
7. 확인을 클릭합니다.
   조인 편집 대화 상자에서 정의한 두 테이블을 연결하는 새 조인이 스키마에 나타납니다.

4.7.7.3 자동으로 검색된 조인 선택

유니버스 디자인 도구의 조인 검색 기능을 사용하여 스키마에서 선택된 조인을 자동으로 검색할 수 있습니다. 이 도구는 대상 데이터베이스의 테이블에서 열 이름을 식별하고 스키마의 테이블에 대한 후보 조인을 제안합니다. 그러면 제안된 모든 조인을 승인하거나 만들려는 조인만 선택할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조인이 자동으로 검색되는 방식

조인은 매개 변수 대화 상자의 전략 페이지(파일 > 매개 변수 > 전략 탭)에 나타나는 조인 전략을 기준으로 검색됩니다.

전략은 데이터베이스에서 구조 정보를 자동으로 추출하는 스크립트 파일입니다. 유니버스 디자인 도구에는 여러 가지 기본 전략이 제공됩니다. 이러한 전략은 매개 변수 대화 상자 전략 페이지의 드롭다운 목록에 나열됩니다.

기본 자동 조인 검색 전략은 키 정보를 제외하고 대응하는 열 이름을 기준으로 조인을 검색합니다. 자동 조인 검색 기능을 사용할 때 적용하고자 하는 조인 전략을 선택할 수 있습니다.

> **노트**
>
> 전략 사용에 대한 자세한 내용은 전략 선택 [페이지 81]을 참조하십시오.

자동 조인 검색 기능의 적절한 사용

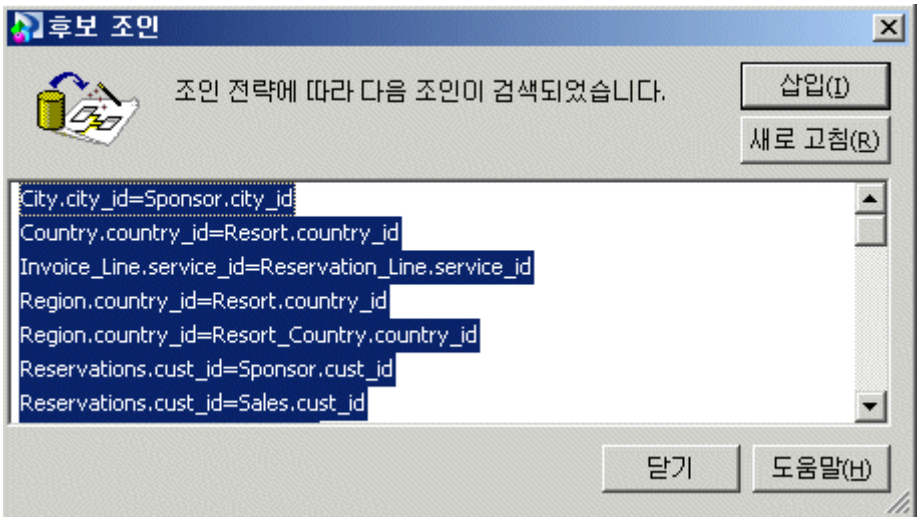
자동 조인 검색 기능을 잘 활용하면 스키마에서 조인을 빠르게 만들 수 있습니다. 하지만 스키마를 디자인할 때 자동 조인 검색 기능의 제한 사항을 알아둘 필요가 있습니다.

후보 조인을 검색하는 데 사용되는 조인 전략은 데이터베이스에서 열 이름을 대응시키는 것입니다. 대상 데이터베이스의 여러 테이블 사이에서 기본 키, 외래 키 및 기타 조인 열의 이름이 동일하지 않은 경우가 많은데 유니버스 디자인 도구는 이러한 열을 선택하지 않습니다. 따라서 만들 수 있도록 승인한, 자동으로 검색된 각 조인을 항상 수동으로 확인해야 합니다. 검색되지 않은 다른 조인 중에 필요한 조인이 있을 수 있습니다.

자동 검색을 사용하여 조인을 만들려면

1. 조인을 검색하는 데 사용할 조인 전략이 매개 변수 대화 상자의 조인 드롭다운 목록 상자에서 선택되었는지 확인합니다. 다음과 같이 확인할 수 있습니다.
   - 파일 > 매개 변수를 선택하고 전략 탭을 클릭합니다.
   - 조인을 검색하기 위해 사용할 전략을 조인 드롭다운 목록 상자에서 선택하고 확인을 클릭합니다.
2. 구조 창에서 여러 개의 테이블을 선택합니다.
   SHIFT 키를 누른 상태에서 각 테이블을 클릭하여 여러 개의 테이블을 선택하거나, 빈 공간에서 마우스를 클릭하고 커서를 끌어서 여러 개의 테이블이 포함된 사각형 영역을 만들어 해당 영역 안의 모든 테이블을 선택할 수 있습니다.
3. 도구 > 자동화된 검색 > 조인 검색을 선택합니다.
   또는
   조인 검색 단추를 클릭합니다.
   후보 조인 대화 상자가 나타납니다. 여기에는 선택된 테이블에 대한 후보 또는 제안된 조인이 나열됩니다. 후보 조인은 구조 창에서 선택된 테이블 사이에 파란색 선으로도 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



4. 모든 후보 조인을 만들려면 삽입을 클릭합니다.
5. 또는
   하나 이상의 조인을 선택하고 삽입을 클릭합니다.
   CTRL 키를 누른 상태에서 개별 조인을 선택하거나 SHIFT 키를 누른 상태에서 연속적인 블록의 처음과 마지막 조인을 클릭하여 하나 이상의 조인을 선택할 수 있습니다.
   조인이 스키마에 삽입됩니다.
6. 닫기를 클릭합니다.

4.7.7.4 연관된 테이블과 함께 자동으로 조인 삽입

조인을 사용하는 테이블이 구조 창에 삽입될 때 조인도 함께 자동으로 스키마에 삽입되도록 선택할 수 있습니다. 자동 조인 생성은 다음 두 가지 프로세스에 의해 결정됩니다.

- 활성 조인 전략이 조인을 검색하는 데 사용되는 열 정보를 결정합니다.
- 기본 생성 옵션인 테이블과 함께 조인 추출을 선택하여 해당 연관 테이블과 함께 자동으로 조인이 생성되도록 해야 합니다. 이 옵션은 옵션 대화 상자의 데이터베이스 페이지에 있습니다.

조인을 자동으로 삽입할 경우의 제한 사항

연관된 테이블과 함께 스키마에 조인을 자동으로 삽입하면 스키마에 조인을 빠르게 삽입할 수 있지만, 스키마에 심각한 디자인 문제가 발생할 수도 있습니다. 조인은 데이터베이스 구조를 기준으로 삽입되므로 데이터베이스에서 이름이 변경된 둘 이상의 테이블에 공통으로 포함된 열은 검색되지 않습니다.

프로덕션 유니버스에서 조인을 만들 때는 이 방법을 사용하면 안 됩니다. 대신 데모용이나 유니버스를 빠르게 만들려는 용도로만 사용하고, 이러한 경우에도 삽입 후 각 조인의 유효성을 검사해야 합니다.

연관된 테이블과 함께 조인을 자동으로 만들려면

1. 조인을 검색하기 위해 사용할 조인 전략이 매개 변수 대화 상자의 전략 페이지에서 선택되었는지 확인합니다.
2. 도구 > 옵션을 선택합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

옵션 대화 상자가 열립니다.

3. 데이터베이스 탭을 클릭합니다.
   데이터베이스 페이지가 나타납니다.
4. 테이블과 함께 조인 추출 확인란을 선택합니다.
5. 확인을 클릭합니다.
   이제 이미 구조 창에 삽입된 테이블의 다른 열을 참조하는 열이 포함된 테이블을 삽입하면 해당 테이블 사이의 참조가 자동으로 조인에 삽입됩니다.

4.7.8 조인 속성

조인 속성은 조인 편집 대화 상자에서 정의합니다. 조인에 대해 다음 속성을 정의할 수 있습니다.

표 99:

속성	설명
테이블 1	조인의 왼쪽 끝에 있는 테이블입니다. 드롭다운 목록 상자에서 선택된 테이블에 대한 열이 나열됩니다.
테이블 2	조인의 오른쪽 끝에 있는 테이블입니다. 드롭다운 목록 상자에서 선택된 테이블에 대한 열이 나열됩니다.
연산자	테이블이 조인되는 방식을 정의하는 연산자입니다. 조인에 사용할 수 있는 연산자는 조인 연산자 [페이지 140] 단원에서 설명합니다.
외부 조인	이 옵션을 선택하면 외부 조인 관계에서 대응되지 않는 데이터를 포함하는 테이블이 확인됩니다. 외부 조인은 테타 조인 만들기 [페이지 153] 단원에서 자세히 설명합니다.
카디널리티	이 옵션을 선택하면 조인에 대한 카디널리티를 정의할 수 있습니다. 카디널리티 정의 및 사용은 카디널리티 사용 [페이지 162] 단원에서 설명합니다.
바로 가기 조인	조인을 바로 가기 조인으로 정의합니다. 바로 가기 조인은 외부 조인 사용에 대한 제한 사항 [페이지 158] 단원에서 설명합니다.
식	두 조인된 테이블이 쿼리에 포함될 때 반환되는 데이터를 제한하는 데 사용되는 WHERE 절입니다.
고급	유니버스에 대한 ANSI 92 지원이 활성화된 경우에 사용할 수 있습니다. 이 옵션을 클릭하면 두 번째 조인 속성 상자가 열립니다. 이 상자에는 조인의 두 테이블에 대한 열을 기반으로 작성된 개체가 나열됩니다. FROM 절에 포함시킬 개체를 선택할 수 있습니다. 조인 구문에 대한 ANSI 92 지원을 활성화하는 방법은 유니버스에서 ANSI 92 조인 형식 지원 [페이지 144] 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.7.8.1 조인 연산자

Table1 과 Table2 상자 사이에 있는 드롭다운 목록 상자에서 조인에 대한 연산자를 선택할 수 있습니다. 연산자를 사용하면 조인된 열 사이의 데이터를 대응시키기 위하여 조인에서 사용할 제한 사항을 정의할 수 있습니다.

조인에 대해 다음 연산자를 선택할 수 있습니다.

표 100:

연산자	설명
=	같음
!=	같지 않음
>	보다 큼
<	보다 작음
>=	보다 크거나 같음
<=	보다 작거나 같음
사이에 있음	사이에 있음(테타 조인)
Complex	복합 관계

4.7.8.2 편집 및 구문 분석

조인 편집 대화 상자에는 조인 구문을 편집하고 확인하는 데 사용할 수 있는 다음 두 가지 기능이 포함되어 있습니다.

편집

편집 단추를 클릭하면 SQL 편집기가 열립니다. 이 그래픽 편집기를 사용하여 조인에서 사용된 테이블, 열, 연산자 및 함수에 대한 구문을 수정할 수 있습니다. 이 편집기 사용에 대한 자세한 내용은 조인 SQL 편집기 사용 [페이지 142] 단원을 참조하십시오.

구문 분석

구문 분석 단추를 클릭하면 조인 식의 SQL 구문을 확인하는 구문 분석 작업이 시작됩니다. 구문 분석이 성공하면 결과가 정상이라는 메시지가 나타납니다. 유니버스 디자인 도구에서 오류가 발생하면 문제의 원인을 나타내는 오류 메시지가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.7.9 조인 편집

다음 방법 중 하나를 사용하여 조인을 편집할 수 있습니다.

- 조인 편집 대화 상자에서 조인 속성 수정
- 조인 SQL 편집기를 사용하여 조인 SQL 구문 직접 수정
- 수식 입력줄을 사용하여 조인 SQL 구문 직접 수정

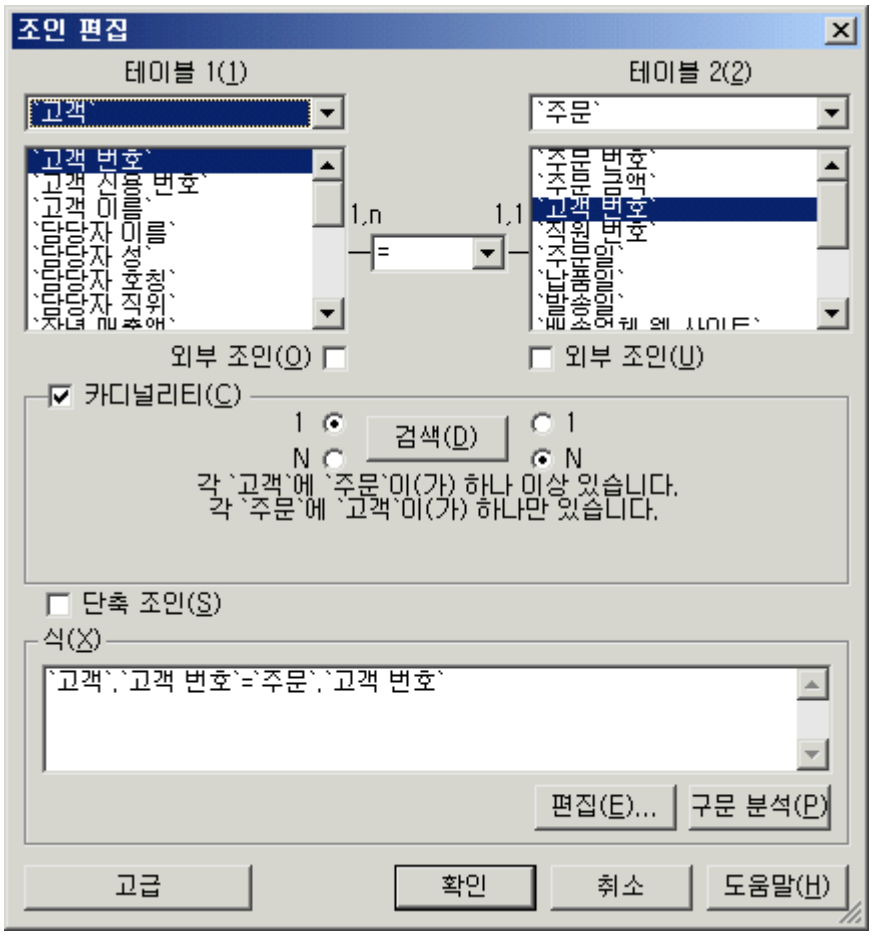
이러한 각 방법은 이 단원에서 설명합니다.

4.7.9.1 조인 편집 대화 상자 사용

조인 편집 대화 상자를 사용하여 조인 속성을 정의하고 편집할 수 있습니다. 또한 조인 SQL 편집기를 사용하여 이 대화 상자에서 조인 구문을 직접 편집할 수도 있습니다. 조인 속성은 조인 속성 [페이지 139] 단원에서 설명합니다.

조인 편집 대화 상자를 사용하여 조인을 편집하려면

1. 구조 창에서 조인을 두 번 클릭합니다.
   또는
   조인을 클릭하고 편집 > 조인을 선택합니다.
   조인 편집 대화 상자가 나타납니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 테이블 사이에 있는 드롭다운 목록 상자에서 연산자를 선택합니다.
3. 필요에 따라 다른 속성을 선택합니다.
4. ANSI 92 구문으로 조인을 정의하는 경우 고급 단추를 클릭합니다.
5. 확인을 클릭합니다.

➡ 팁

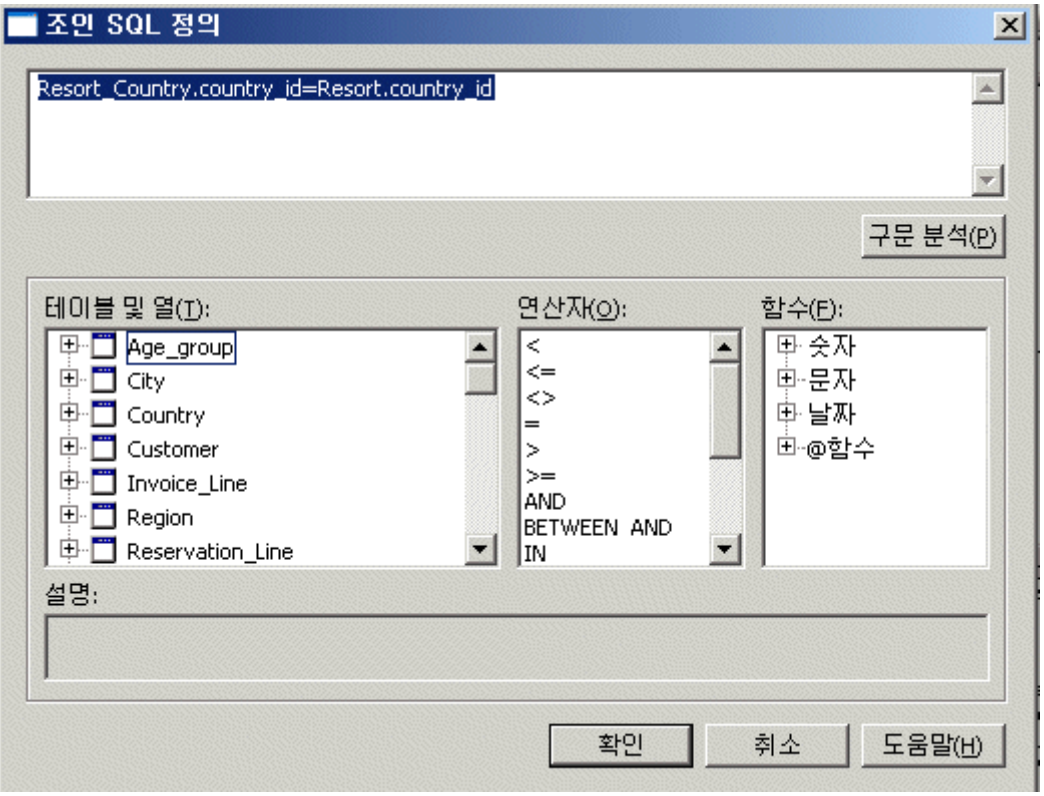
편집 단추를 누르고 조인 SQL 편집기를 사용하여 조인에 대한 SQL 을 직접 편집할 수도 있습니다. 자세한 내용은 조인 SQL 편집기 사용 [페이지 142]을 참조하십시오.

4.7.9.2 조인 SQL 편집기 사용

그래픽 편집기를 사용하여 조인에 대한 SQL 식을 직접 수정할 수 있습니다. 이 편집기를 조인 편집 대화 상자에서 액세스할 수 있습니다.

조인 SQL 편집기를 사용하여 조인을 수정하려면

1. 구조 창에서 조인을 두 번 클릭합니다.
   또는
   조인을 클릭하고 편집 > 조인을 선택합니다.
   조인 편집 대화 상자가 나타납니다.
2. 편집 단추를 클릭합니다.
   조인 SQL 정의 상자가 나타납니다. 조인에 대한 SQL 식이 텍스트 상자에 나타납니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. SQL 구문을 추가하거나 수정하고자 하는 위치에서 편집 상자에 있는 조인 식을 클릭합니다.
   다음과 같이 편집 기능을 사용하여 SQL 구문을 수정하거나 추가할 수 있습니다.

   표 101:

작업	수행 방법
조인 한쪽 끝에 있는 열 변경	○ 테이블 및 열 상자에서 테이블 노드를 확장합니다. ○ 열 이름을 두 번 클릭합니다.
조인에서 사용된 연산자 변경	연산자 상자에서 연산자를 두 번 클릭합니다.
조인에서 함수 사용	○ 함수 계열 노드를 확장합니다. ○ 함수를 두 번 클릭합니다.

   열, 연산자 또는 함수가 조인 정의에 나타납니다.

4. 확인을 클릭합니다.

4.7.9.3 수식 입력줄 사용

수식 입력줄은 유니버스 창 위에 있는 텍스트 상자로 구조 창에서 선택된 조인 또는 유니버스 창에서 선택된 개체의 수식이나 식을 보여줍니다. 수식 입력줄의 왼쪽에 있는 다음 세 가지 편집 단추를 사용할 수 있습니다.

표 102:

편집 단추	설명
	유효성이 검사되지 않은 마지막 수정 내용을 취소합니다. 변경 내용의 유효성을 검사하지 않고 조인 식을 변경한 경우 취소 단추를 클릭하면 식이 원래 상태로 돌아갑니다. 개별 수정 내용을 취소하려면 편집 > 실행 취소 옵션을 사용하거나 실행 취소를 클릭해야 합니다.
	식의 유효성을 검사합니다. 이 작업은 조인 식의 모든 변경 내용에 적용됩니다. 유효성 검사 후 편집 > 실행 취소 옵션을 사용하거나 실행 취소를 클릭하여 변경 내용을 취소할 수 있습니다.
	선택된 조인에 대한 조인 편집 대화 상자를 엽니다.

수식 입력줄을 표시하려면

- 보기 > 수식 입력줄 을 선택합니다.
  수식 입력줄이 유니버스 창 위에 나타납니다.

수식 입력줄을 사용하여 조인을 수정하려면

1. 편집하려는 조인을 클릭합니다.
   조인에 대한 수식이 수식 입력줄에 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 구문을 수정하고자 하는 위치에서 수식 입력줄에 있는 조인 식을 클릭합니다.
3. 필요에 따라 식을 수정합니다.
4. 유효성 검사를 클릭하여 변경 내용을 적용합니다.
5. 돌아가기 키를 눌러 수식 입력줄을 종료합니다.
   또는
   수식 입력줄의 바깥쪽을 아무 곳이나 클릭합니다.

4.7.10 유니버스에서 ANSI 92 조인 형식 지원

유니버스 디자인 도구는 조인에 대해 ANSI 92 구문을 지원합니다. ANSI 92 는 기본적으로 지원되지 않습니다. SQL 유니버스 매개 변수 ANSI92 를 YES 로 설정하여 지원을 활성화해야 합니다. 이 매개 변수는 유니버스 매개 변수 대화 상자의 매개 변수 페이지(파일 > 매개 변수 > 매개 변수)에 나열되어 있습니다. 지원을 활성화하면 유니버스의 조인에 대해 ANSI 92 구문을 사용하도록 선택할 수 있습니다.

> **노트**
>
> ANSI 92 설정 또한 .prm 파일에 선언됩니다. .prm 설정이 'usual'이면 유니버스 디자인 도구 설정이 우선적으로 적용됩니다. .prm 설정이 'ANSI92'이면 유니버스 디자인 도구 수준의 설정이 무시됩니다. .prm 파일 및 ANSI 92 설정에 대한 자세한 내용은 *Data Access 가이드*를 참조하십시오. 나타나는 결과는 데이터베이스 버전에 따라 달라질 수 있습니다. 자세한 내용은 데이터베이스 기술 세부 정보를 참조하십시오.

조인에서 이 구문을 사용하려면 먼저 대상 RDBMS 가 ANSI 92 를 지원하는지 확인해야 합니다.

유니버스에서의 ANSI 92 지원 활성화 및 ANSI 92 구문을 사용한 조인 정의에 대한 자세한 내용은 이후의 설명을 참조하십시오.

4.7.10.1 예제: 기본 조인 구문 및 ANSI 92 구문 비교

아래에는 두 조인에 대한 조인 구문이 나와 있습니다. 기본 동작을 보여 주는 첫 번째 구문에서는 조인이 WHERE 절에 정의되어 있고, 두 번째 구문에서는 동일한 조인이 ANSI 92 표준을 사용하여 FROM 절에 정의되어 있습니다.

기본 조인 구문

```
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  )
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

```
  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')
```

ANSI 92 표준을 사용한 동일한 조인

```
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 유니버스에서 ANSI 92 지원 활성화

조인에 대한 ANSI 92 지원을 활성화하려면

1. 파일 > 매개 변수를 선택합니다.
   유니버스 매개 변수 대화 상자가 나타납니다.
2. 매개 변수 탭을 클릭합니다.
   매개 변수 페이지가 나타납니다. 이 페이지에는 현재 유니버스에 대한 SQL 생성을 최적화하기 위해 유니버스 수준에서 설정할 수 있는 특정 SQL 생성 매개 변수가 나열됩니다. 이전 버전의 Business Objects 제품에서는 이러한 매개 변수가 대상 RDBMS 의 PRM 파일에 포함되었습니다. 특정 RDBMS 별 매개 변수는 여전히 PRM 파일에 포함되지만, 대부분의 표준 SQL 매개 변수는 이제 매개 변수 페이지에 나열됩니다. 사용 가능한 매개 변수의 전체 목록은 SQL 생성 매개 변수 설정 [페이지 88] 장을 참조하십시오.
3. 목록에서 ANSI92 매개 변수를 클릭합니다.
4. 값 상자에 YES 를 입력합니다.
5. 바꾸기를 클릭합니다.
6. 확인을 클릭합니다.
   이제 현재 유니버스에 대한 조인 정의에 ANSI 92 표준을 적용할 수 있습니다. 조인 편집 대화 상자에서 고급 단추를 클릭하면 고급 조인 상자가 나타납니다. 필터를 정의하여 조인의 FROM 절에 포함될 차원을 결정할 수 있습니다.

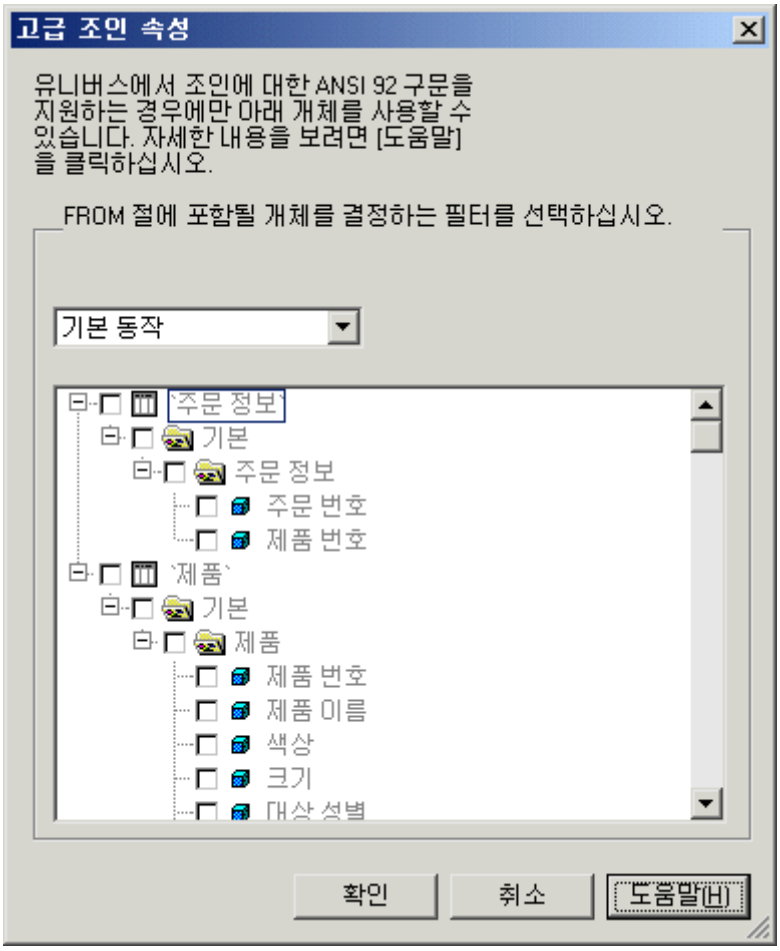
4.7.10.3 ANSI 92 구문으로 조인 정의

ANSI 92 구문을 사용하여 조인 편집 대화 상자에서 조인을 정의할 수 있습니다. 이렇게 하려면 조인 정의에 포함시킬 개체를 선택하는 데 사용할 수 있는 고급 편집 대화 상자에서 작업해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

ANSI 92 구문을 사용하여 조인을 정의하려면

1. 유니버스에 대한 ANSI 92 지원을 활성화합니다. 자세한 내용은 유니버스에서 ANSI 92 지원 활성화 [페이지 145] 단원을 참조하십시오.
2. 스키마에서 조인을 두 번 클릭합니다.
   조인에 대한 조인 편집 대화 상자가 열립니다.
3. 고급 단추를 클릭합니다.
   고급 조인 속성 대화 상자가 나타납니다.



4. 드롭다운 목록에서 다음과 같은 FROM 절 필터 중 하나를 선택합니다.

표 103:

FROM 옵션	설명
기본 동작	조인에 대한 기본 구문이 적용됩니다. 조인이 WHERE 절에 정의됩니다.
FROM 절의 모든 개체	조인의 오른쪽과 왼쪽에 있는 테이블의 열에 정의된 모든 개체가 FROM 절에 포함됩니다.
FROM 절의 모든 개체 제외	어떠한 개체도 FROM 절에 포함되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

FROM 옵션	설명
FROM 절의 선택된 개체	조인 테이블의 고급 조인 속성 트리 뷰에서 선택한 개체만 FROM 절에 포함됩니다.

5. FROM 절의 선택된 개체 필터를 선택한 경우 FROM 절에 포함시킬 개체를 선택합니다.
6. 확인을 클릭합니다.
7. 조인 편집 상자에 다른 모든 조인 매개 변수를 입력합니다.
8. 확인을 클릭합니다.

4.7.11 조인 삭제

조인을 삭제하려면

1. 조인을 클릭합니다.
   조인이 선택됩니다.
2. 다음 중 하나를 수행합니다.
   - 키보드의 Backspace 키를 누릅니다.
   - 키보드의 Delete 키를 누릅니다.
   - 테이블을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 지우기를 선택합니다.
     조인 삭제를 확인하는 확인 상자가 나타납니다.
3. 예를 클릭합니다.
   조인이 삭제됩니다.

> **노트**
>
> 조인을 삭제할 경우 스키마와 유니버스 모두에서 발생할 수 있는 결과를 고려하십시오. 조인을 삭제해도 컨텍스트에는 영향이 없는지 확인하십시오. 조인을 삭제하려고 하면 유니버스 디자인 도구에서 조인이 하나 이상의 컨텍스트에서 사용되는지 확인하도록 경고를 표시합니다. 어떤 컨텍스트가 영향을 받는지에 대해서는 직접 확인해야 하고, 조인 삭제로 인해 컨텍스트가 영향을 받을 경우 유니버스에 대한 영향을 평가해야 합니다.

4.8 특정한 유형의 조인 정의

유니버스 디자인 도구에서 다음과 같은 형식의 조인을 정의할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 104:

조인 유형	설명
동일 조인 (복합 동일 조인 포함)	한 테이블의 열 값과 다른 테이블의 열 값 사이에서 동가를 기준으로 테이블을 연결합니다. 두 테이블에 동일한 열이 존재하기 때문에 조인에 의하여 두 테이블이 동기화됩니다. 두 테이블 사이에서 한 조인이 여러 열을 연결하는 복합 동일 조인도 만들 수 있습니다.
테타 조인(조건부 조인)	두 열 사이에서 동가 이외의 관계를 기준으로 테이블을 연결합니다.
외부 조인	두 테이블을 연결하는데, 두 테이블 중 하나에는 다른 테이블의 공통 열에 있는 행과 대응하지 않는 행이 있습니다.
바로 가기 조인	중간 테이블을 거치지 않고 어떤 방향으로든지 동일한 결과가 나타나도록 두 테이블 사이에 대체 경로를 제공하는 조인입니다. 긴 조인 경로를 가능한 짧게 단축하여 쿼리 시간을 최적화합니다.
자체 제한 조인	테이블에 대한 제한 사항을 설정하는 데 사용되는 단일 테이블 조인입니다.

각 조인 유형은 이 장에서 각각 별도의 단원을 통해 자세히 설명합니다. 동일한 방법을 사용하여 각 유형의 조인을 만들 수 있지만, 조인을 만들 때 조인 편집 상자에서 각 조인에 대해 서로 다른 속성을 정의해야 합니다.

4.8.1 동일 조인 만들기

동일 조인은 테이블 1 의 열과 테이블 2 의 열에 공통 값이 있는 두 테이블을 연결합니다. 제한 사항은 다음 구문과 같습니다.

Table1.column_a = Table2.column_a

정규화된 데이터베이스에서 동일 조인에 사용되는 열은 일반적으로 한 테이블의 기본 키와 다른 테이블의 외래 키입니다. 키에 대한 내용은 조인되지 않아야 하는 테이블 [페이지 132] 단원을 참조하십시오.

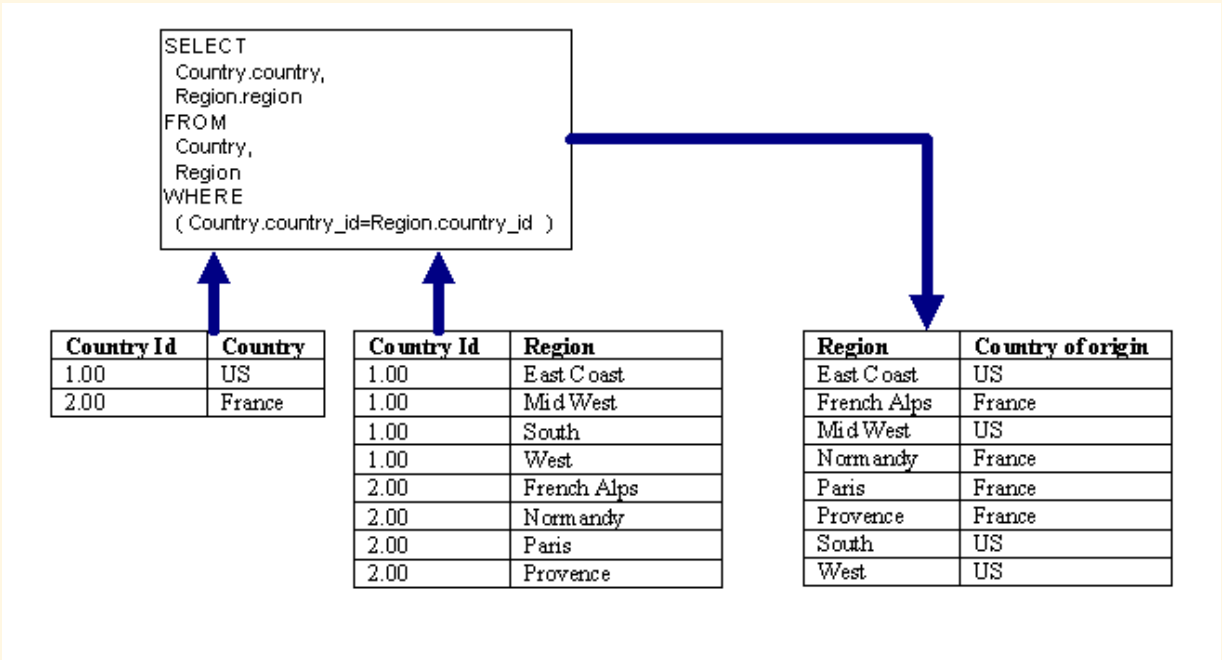
새 조인을 만들면 기본적으로 동일 조인이 됩니다. 스키마에서 대부분의 조인은 동일 조인이어야 합니다.

예

데이터를 제한하는 동일 조인

아래 예제에서 Select 문이 실행될 때 Select 및 From 절에 의하여 카티전 곱이 생성됩니다. 하지만 데이터가 반환되기 전에 Where 절에 의하여 제한 사항이 적용되므로 두 테이블의 Country ID 열 사이에서 대응하는 행만 반환됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



4.8.1.1 새 동일 조인 만들기

새 동일 조인을 만들려면

- 두 테이블 사이에 조인을 만듭니다.
  기본적으로 새 조인은 동일 조인입니다.

➔ 팁

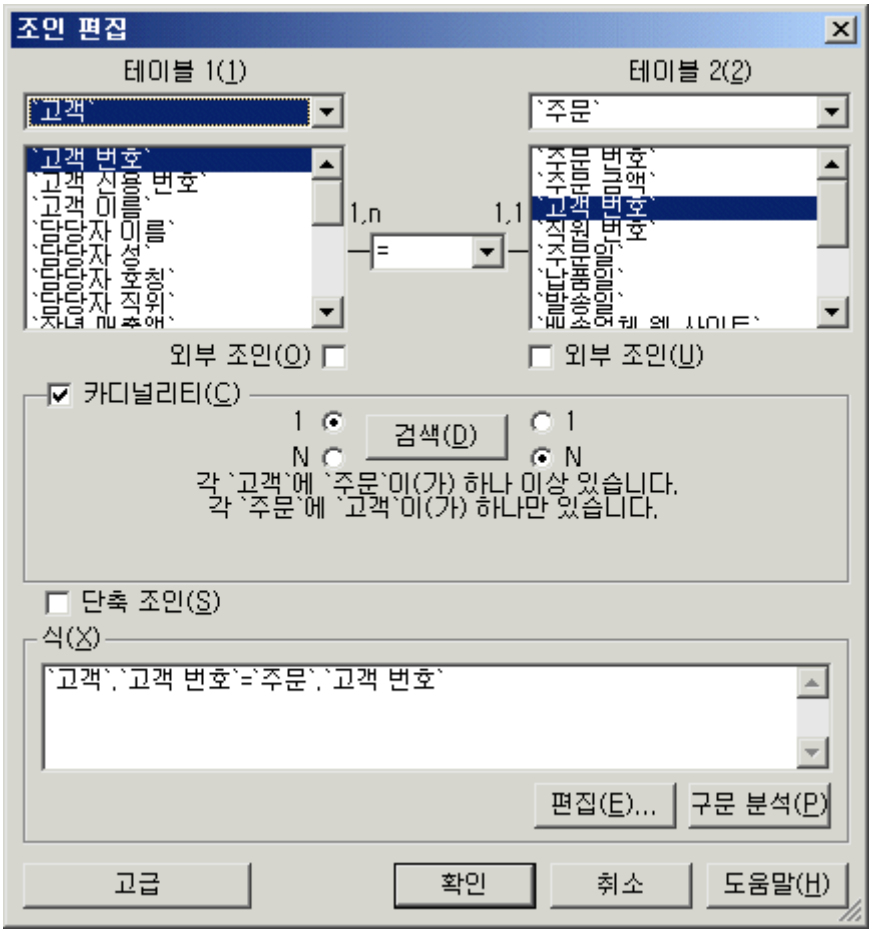
조인 만들기에 사용할 수 있는 다른 방법은 키 표시 [페이지 133] 단원에서 설명합니다.

4.8.1.2 기존 조인에서 동일 조인 만들기

기존 조인에서 동일 조인을 만들려면

1. 기존 조인을 두 번 클릭합니다.
   조인 편집 상자가 나타납니다.
2. Table1 목록 상자에서 열을 선택합니다.
3. Table2 목록 상자에서 대응하는 열을 선택합니다.
4. 연산자 드롭다운 목록 상자에서 =를 선택합니다.
   다음과 같이 조인 편집 상자에 Customer 테이블과 Reservations 테이블 사이의 동일 조인이 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



> **노트**
>
> 공통 열의 이름이 언제나 동일한 것은 아닙니다. 데이터베이스에서 기본 키와 외래 키의 열 이름을 확인해야 합니다. 서로 다른 테이블이 동일한 키 열을 사용하지만, 데이터베이스에서 테이블의 역할에 따라 각 테이블에 대한 열 이름이 변경된 경우도 있을 수 있습니다.

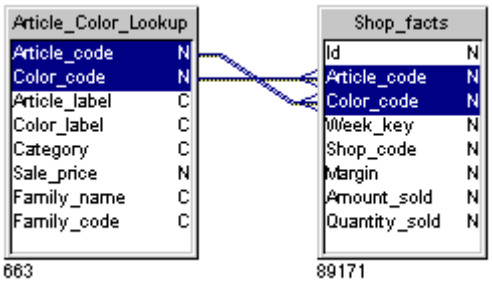
5. 구문 분석 단추를 클릭하여 조인 구문을 검사합니다.
   오류 메시지가 나타나면 열이 두 테이블 모두에 공통인지 확인합니다.
6. 확인을 클릭합니다.

4.8.1.3 복합 동일 조인 만들기

복합 동일 조인을 만들 수도 있습니다. 복합 동일 조인은 두 테이블 사이에서 여러 열을 연결하는 단일 조인입니다. 조인에 대한 속성 편집 시트에서 조인에 대해 Complex 연산자를 사용하여 복합 동일 조인을 만들 수 있습니다.

eFashion 유니버스 예제에는 다음과 같은 복합 조인이 포함되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



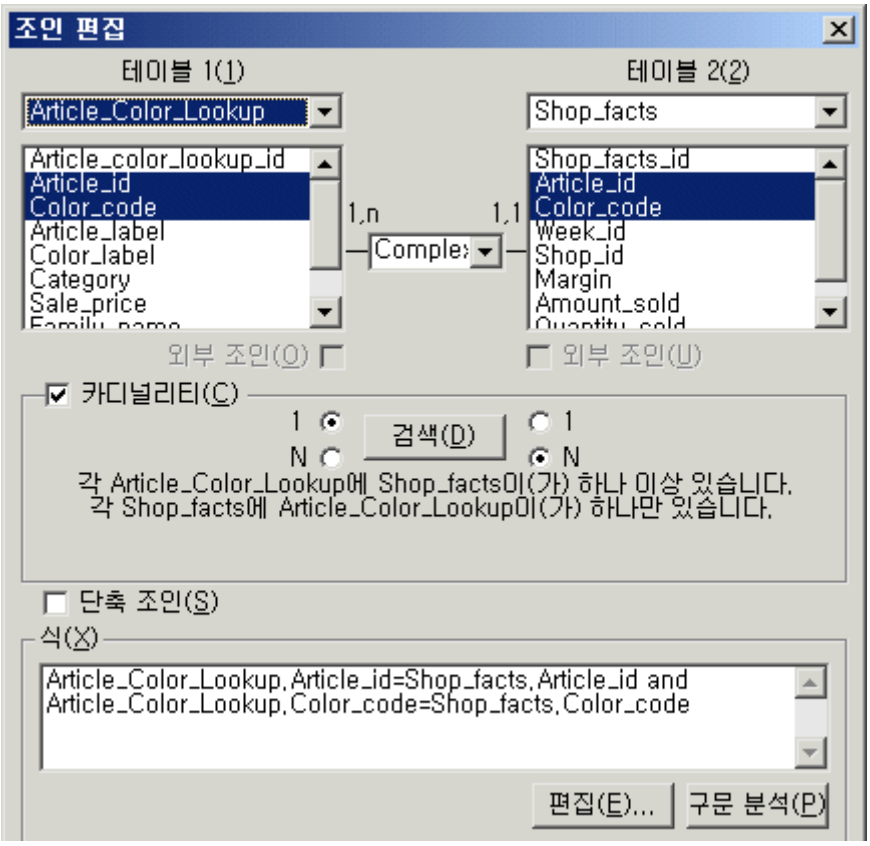
조인된 열 사이에서 여러 단일 동일 조인 대신 복합 동일 조인을 사용하면 다음과 같은 이점이 있습니다.

- 하나의 카디널리티만 검색하면 됩니다. 카디널리티를 검색할 때 시간이 단축되고, 스키마가 정돈되어 읽기가 쉽습니다.
- 조인에 대한 속성 편집 상자의 식 테스트 상자에서 두 테이블 사이의 모든 조인에 대한 SQL 을 볼 수 있습니다. 두 테이블 사이에서 여러 단일 동일 조인을 사용하면 각 조인에 대해 식이 하나씩만 존재합니다.

복합 동일 조인을 만들려면

1. 기존 조인을 두 번 클릭합니다.
   조인 편집 상자가 나타납니다.
2. Table1 목록 상자에서 여러 개의 열을 선택합니다.
3. Table2 목록 상자에서 대응하는 열을 선택합니다.
4. 연산자 드롭다운 목록 상자에서 "Complex"를 선택합니다.
   다음과 같이 조인 편집 상자에 Article_Color_Lookup 테이블과 Shop_facts 테이블 사이의 복합 동일 조인이 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



5. 구문 분석 단추를 클릭하여 조인 구문을 검사합니다.
   오류 메시지가 나타나면 열이 두 테이블 모두에 공통인지 확인합니다.
6. 확인을 클릭합니다.

4.8.2 테타 조인

테타 조인은 두 열 사이에 동가 이외의 관계를 기준으로 테이블을 연결하는 조인입니다. 테타 조인은 "=" 연산자 이외의 모든 연산자를 사용할 수 있습니다.

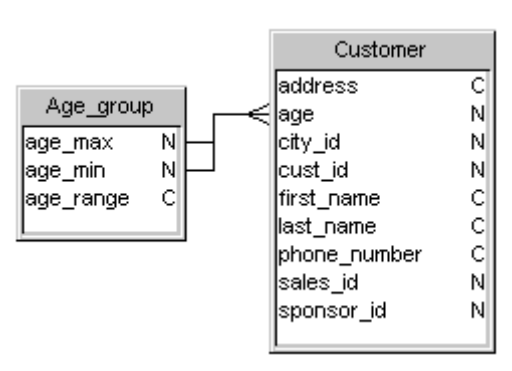
다음 예제와 절차는 "Between" 연산자를 사용하는 테타 조인을 만드는 방법을 보여 줍니다.

예

테타 조인

다음의 Age_Group 테이블은 고객의 나이에 대한 데이터를 분석하는 데 사용할 수 있는 연령대 정보를 포함하고 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

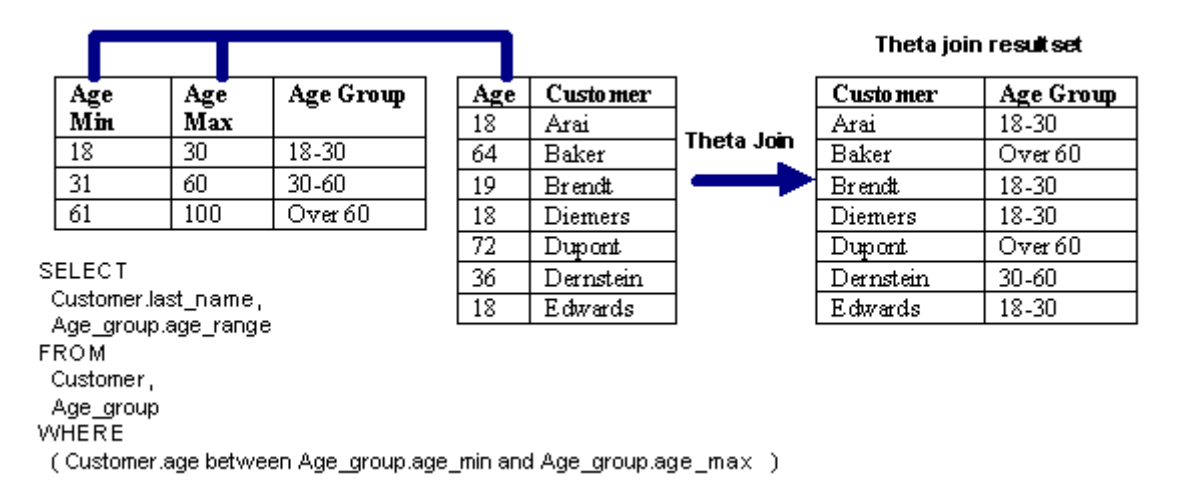


이 테이블을 유니버스에 포함시켜야 하지만 Customer 테이블과 Age_Group 테이블 사이에는 공통 열이 없으므로 동일 조인을 사용할 수 없습니다.

이 경우에는 최고 연령대와 최저 연령대에 대해 "Between" 연산자를 사용하는 테타 조인을 만듭니다. 테타 조인을 사용하면 Customer 테이블에 있는 Age 열의 행 값이 Age_Group 테이블에 있는 Age_Min 및 Age_Max 열의 행 값 사이에 있는 조인이 존재한다는 것을 추론할 수 있습니다. 조인은 다음 식으로 정의됩니다.

Customer.age between Age_group.age_min and Age_group.age_max

다음 다이어그램은 Age max, Age min 및 Age 사이의 조인을 나타내고, Age_Group 테이블과 Customer 테이블 모두에 대하여 쿼리를 실행할 때 테타 조인이 사용될 경우 반환되는 결과 집합을 보여줍니다.



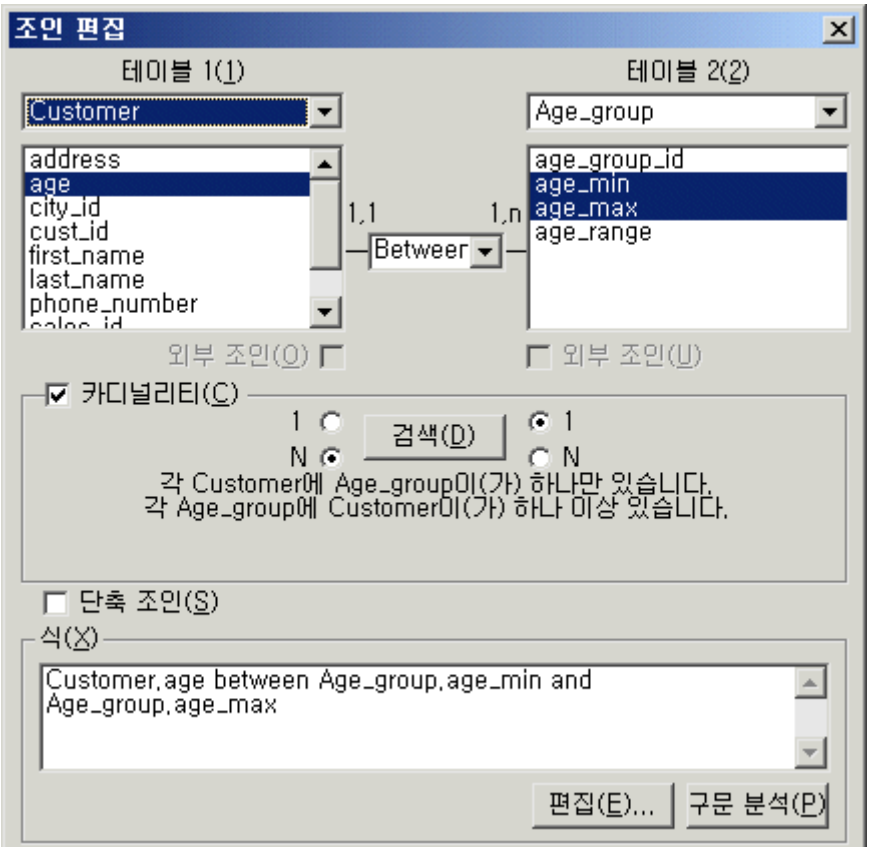
4.8.2.1 테타 조인 만들기

범위 열을 사용하여 테타 조인을 만들려면

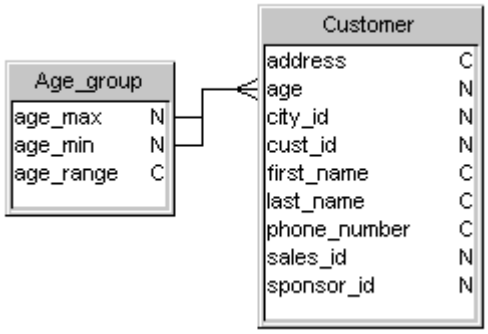
1. 두 테이블 사이에 조인을 만듭니다.
   기본적으로 동일 조인이 만들어집니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 조인을 두 번 클릭합니다.
   조인 편집 대화 상자가 나타납니다.
3. Table1 열 목록 상자에서 열을 클릭합니다.
4. CTRL 키를 누른 상태에서 Table2 열 목록 상자의 두 열을 클릭합니다.
   아래 예제에서는 age_min 및 age_max 의 두 열이 선택되었습니다. 연산자 드롭다운 목록에 Between 연산자가 자동으로 나타납니다.



5. 구문 분석 단추를 클릭하여 조인에 대한 유효성을 테스트합니다.
   오류 메시지가 나타나면 열을 정확하게 선택했는지 확인합니다.
6. 확인을 클릭합니다.
   구조 창에서 조인이 만들어집니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.8.3 외부 조인

외부 조인은 두 테이블을 연결하는 조인으로, 두 테이블 중 하나에는 다른 테이블의 공통 열에 있는 행과 대응하지 않는 행이 있습니다.

원래 동일 조인에서 외부 테이블이 될 테이블을 지정하여 외부 조인을 정의합니다. 외부 테이블에는 대응되지 않더라도 모든 값을 반환하고자 하는 열이 포함됩니다. 선택한 조인에 대해 조인 편집 대화 상자에서 외부 테이블을 지정합니다.

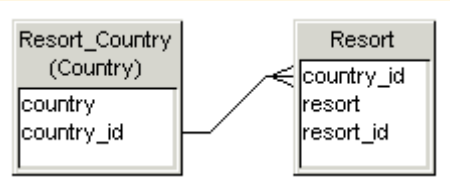
4.8.3.1 완전 외부 조인

기본적으로 조인의 어느 쪽에 외부 테이블이 지정되는가에 따라 왼쪽 외부 조인이나 오른쪽 외부 조인을 만들 수 있습니다. 유니버스에서 조인에 대한 ANSI 92 지원을 활성화하여 완전 외부 조인을 만들 수도 있습니다. ANSI 92 지원을 활성화하려면 유니버스 SQL 매개 변수 ANSI 92 를 YES 로 설정합니다(파일 > 매개 변수 > 매개 변수). 이렇게 하면 유니버스에서 조인에 대한 ANSI 92 구문을 지원할 수 있고 조인의 임의의 측면에 있는 테이블을 외부 테이블로 선택할 수 있습니다. 완전 외부 조인을 만드는 방법은 완전 외부 조인 정의 [페이지 158] 단원을 참조하십시오.

예

외부 조인

아래의 Resort_Country 테이블과 Resort 테이블은 동일 조인으로 연결되어 있습니다.



각 휴양지(Resort)는 국가(Country)에 속하지만, 모든 국가에 휴양지가 있는 것은 아닙니다. 외부 조인을 사용하면 쿼리 결과 집합에는 휴양지가 있는 국가, 즉 오스트레일리아, 프랑스, 미국의 정보만 표시됩니다.

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

하지만 Resort 테이블의 외래 키 값에 상관 없이 모든 국가를 표시하고자 하는 경우도 있습니다. 이렇게 하려면 외부 조인을 정의하여 Resort 열에서 대응하는 값이 없더라도 모든 국가가 반환되도록 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

외부 조인에 대한 구문(Microsoft Access)은 다음과 같습니다.

```
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 }
```

노트

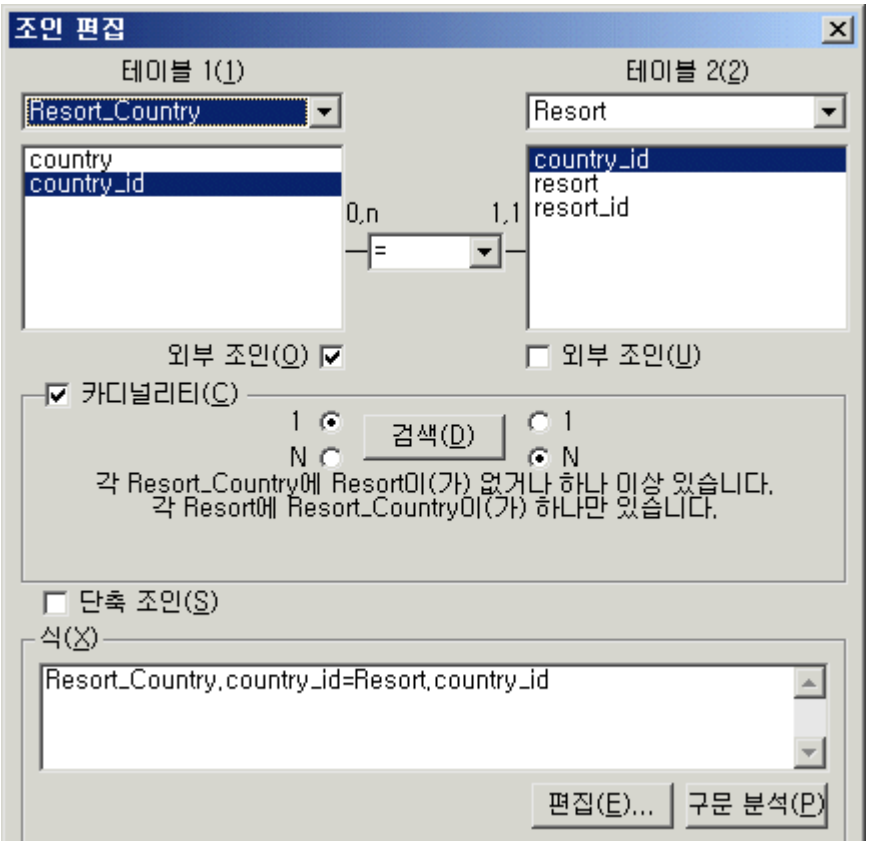
위의 예제는 Microsoft Access 를 사용하므로 Resort 테이블 뒤의 모든 일대다 조인도 다른 외부 조인을 사용해야 합니다. 그렇지 않으면 다음 조인에 의해 반환된 대응 NULL 이 없는 경우 원래 외부 조인에 의해 반환된 NULL 이 고려되지 않습니다. 외부 조인에 대한 처리는 RDBMS 에 따라 다르므로 자세한 내용은 해당 RDBMS 설명서를 참조하십시오. 외부 조인을 사용하는 제한 사항에 대한 자세한 내용은 외부 조인 사용에 대한 제한 사항 [페이지 158] 단원을 참조하십시오.

4.8.3.2 외부 조인 만들기

외부 조인을 만들려면

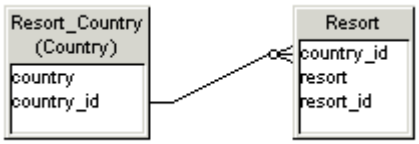
1. 기존의 동일 조인을 두 번 클릭합니다.
   조인 편집 대화 상자가 나타납니다.
2. 쿼리에서 모든 값을 반환하는 테이블에 대해 외부 조인 확인란을 선택합니다.
   다음 예제에서는 Resort_Country 에 대한 모든 값을 반환하고자 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



3. 구문 분석 단추를 클릭하여 조인 구문의 유효성을 검사합니다.
   오류 메시지가 나타나면 열을 정확하게 선택했는지 확인합니다.
4. 확인을 클릭합니다.
   유니버스 디자인 도구의 구조 창에 조인이 표시됩니다. 연결선이라는 그래픽 옵션을 사용하여 외부 조인이 만들어진 경우, 대응되지 않는 값을 반환하는 테이블의 조인 반대쪽에 작은 원이 표시됩니다.

> **노트**
>
> 연결선 외의 그래픽 옵션을 사용하여 외부 조인을 만드는 경우에는 작은 원이 표시되지 않습니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.8.3.3 완전 외부 조인 정의

외부 조인을 정의하기 위한 ANSI 92 표준을 사용하여 외부 조인을 정의할 수 있습니다. 이렇게 하면 완전 외부 조인을 지정할 수 있습니다. 외부 조인에 대해 ANSI 92 표준을 사용하려면 ANSI 92 매개 변수를 YES 로 설정해야 합니다. 이 매개 변수는 매개 변수 페이지(파일 > 매개 변수 > 매개 변수)에서 사용할 수 있습니다.

> **노트**
>
> 이 매개 변수와 유니버스의 기타 SQL 생성 매개 변수 설정에 대한 내용은 SQL 생성 매개 변수 설정 [페이지 88] 단원을 참조하십시오.

ANSI 92 매개 변수를 YES 로 설정하면 조인의 양쪽에 있는 테이블을 외부 테이블로 선택할 수 있습니다. 이 매개 변수를 설정하려면 먼저 대상 RDBMS 가 외부 조인에 대해 ANSI 92 표준을 지원하는지 확인해야 합니다.

완전 외부 조인을 정의하는 과정은 다음 두 단계로 이루어집니다.

- 유니버스의 외부 조인에 대한 ANSI 92 지원을 활성화합니다. 자세한 내용은 유니버스에서 ANSI 92 지원 활성화 [페이지 145] 단원을 참조하십시오.
- 조인 편집 대화 상자를 사용하여 완전 외부 조인을 정의합니다.

완전 외부 조인을 정의하려면

1. 유니버스에 대한 ANSI 92 지원을 활성화합니다.
2. 스키마에서 조인을 두 번 클릭합니다.
   조인 편집 대화 상자가 나타납니다.
3. 조인에 포함된 두 테이블 모두에 대해 외부 조인 확인란을 선택합니다.
4. 확인을 클릭합니다.
   유니버스 디자인 도구의 구조 창에 조인이 표시됩니다. 완전 외부 조인은 두 테이블 사이의 조인 연결에서 두 개의 원으로 표시됩니다.

4.8.3.4 외부 조인 사용에 대한 제한 사항

외부 조인은 매우 유용할 수 있지만 다음과 같은 성능 및 구현 문제에 대해 알아둘 필요가 있습니다.

- 성능이 저하될 수 있습니다. 외부 조인이 사용되면 더 많은 행이 반환되고 일부 데이터베이스에서는 인덱스를 사용하지 않으므로 데이터 양이 많을 경우 쿼리 속도가 느려질 수 있습니다.
- 외부 조인 사용에 대한 데이터베이스 제한 모든 데이터베이스가 WHERE 절에서 외부 조인에 대한 제어를 허용하지는 않습니다. 이러한 제한은 자체 제한 조인을 사용할 때 필요합니다. 예를 들어 Type Code 가 NULL 일 때 TYPE=10 은 true 가 될 수 없고 NULL 값은 외부 조인에 의해 생성되므로, TYPE=10 이거나 Type 이 NULL 인 경우 자체 제한 조인 'TYPE_CODE=10'은 모든 행을 반환할 수 있습니다.
- 원래 외부 조인 이후의 불완전한 쿼리 경로를 피하려면 대상 RDBMS 가 외부 조인을 처리하는 방식을 확인해야 합니다. 예를 들어, Microsoft Access 예제 Club.mdb 데이터베이스에서 조인 경로의 외부 조인 이후의 모든 일대다 조인은 다른 조인으로도 정의되어야 합니다. 그렇지 않으면 원래 외부 조인이 결과 쿼리에서 무시됩니다. 아래 예에서 Resort 와 Service_Line 사이의 조인은 Resort_Country 와 Resort 사이의 외부 조인에 의해 반환된 NULL 값을 무시합니다. 3 개의 테이블에 대하여 쿼리를 실행할 경우, 첫 번째 조인을 수행하는 별도의 쿼리를 만든 다음 그 쿼리를 SQL 문에 포함시킬 것을 권장하는 데이터베이스 오류 메시지가 반환됩니다. 이러한 유형의 오류는 많은 사용자에게 혼동을 줄 수 있으므로 이 경우 외부 조인을 사용하지 않거나 외부 조인으로 경로를 완전하게 만드는 것이 좋습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.8.4 바로 가기 조인

바로 가기 조인은 두 테이블 사이에서 대체 경로를 제공하는 조인입니다. 바로 가기 조인은 중간 테이블을 거치지 않으므로 일반적으로 긴 조인 경로를 단축하여 쿼리 성능을 향상시킵니다.

바로 가기 조인의 일반적인 용도는 공유 조회 테이블을 조인 경로를 따라 다른 테이블로 연결하는 것입니다. 조인 경로는 동일한 컨텍스트의 여러 다른 테이블로 구성됩니다.

이러한 경우에는 조회되는 값이 테이블 계층구조에서 낮은 수준으로 비정규화되어, 조인되는 모든 수준에서 동일한 값이 존재할 때만 바로 가기 조인이 효과적입니다.

바로 가기 조인은 특정 컨텍스트에 대한 조인 경로를 "가로지르지" 않을 경우 무시됩니다. 관련 Web Intelligence 쿼리에 대해 생성된 SQL 에서는 비효율적인 바로 가기 조인을 사용하지 않습니다.

> **노트**
>
> 유니버스 디자인 도구에서는 자동 루프 및 컨텍스트 검색 중 바로 가기 조인을 고려하지 않습니다. 하지만 바로 가기 조인에 대한 카디널리티를 설정하면 컨텍스트를 검색할 때 '일부 카디널리티가 설정되지 않음' 메시지를 피할 수 있습니다.

4.8.4.1 바로 가기 조인 만들기

바로 가기 조인을 만들려면

1. 조인 경로에서 직접 연결할 수 있는 두 테이블을 찾습니다.
2. 두 테이블 사이에 조인을 만듭니다.
3. 새 조인을 두 번 클릭합니다.
   조인 편집 대화 상자가 나타납니다.
4. 바로 가기 조인 확인란을 선택합니다.
5. 필요에 따라 다른 조인 속성을 선택하거나 입력합니다.
6. 확인을 클릭합니다.
   두 테이블을 연결하는 바로 가기 조인이 나타납니다. 바로 가기 조인이 구조 창에서 점선으로 표시됩니다.

> **노트**
>
> 바로 가기 조인의 카디널리티는 바로 가기 조인이 대체하는 조인 경로와 동일한 카디널리티로 설정해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

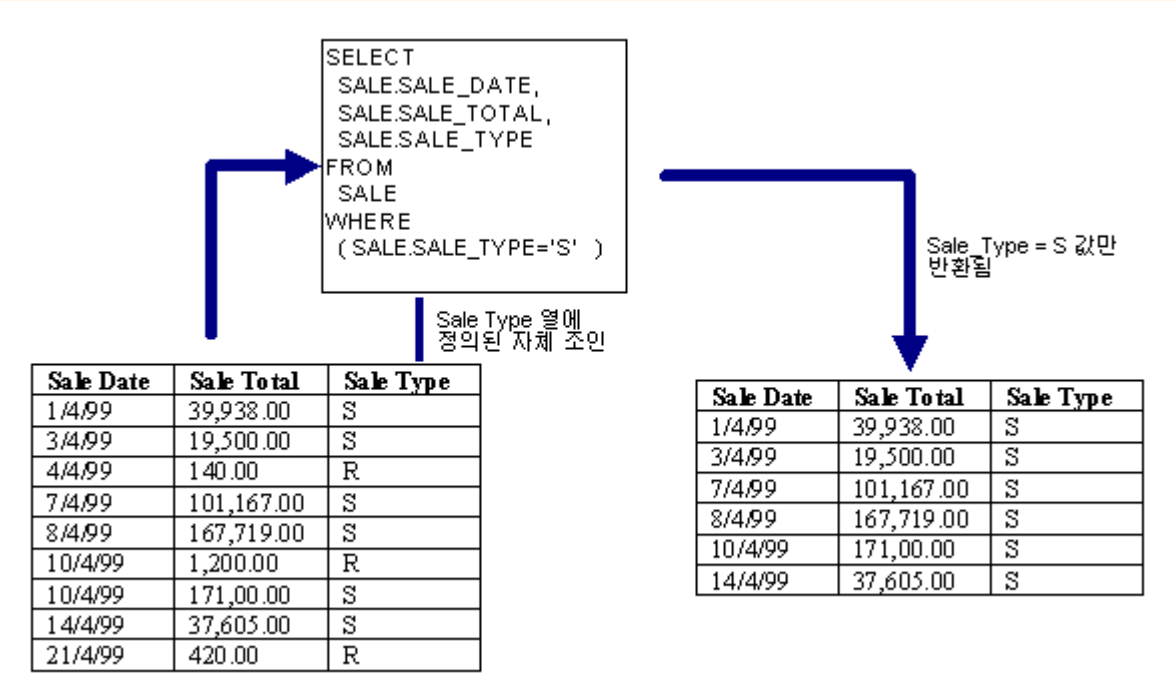
4.8.5 자체 제한 조인

자체 제한 조인은 실제 조인이 아니라, 단일 테이블에 대한 자체 제한입니다. 자체 제한 조인을 사용하면 고정 값을 사용하여 테이블 값에 의해 반환되는 결과를 제한할 수 있습니다.

예

자체 제한 조인

다음 Sales 테이블에는 판매된 자동차와 임대된 자동차 모두에 대한 데이터 행이 포함되어 있습니다. Sale_Type 열은 거래 유형(S = 자동차 판매, R = 자동차 임대)을 나타내는 플래그로 사용됩니다. 자체 제한 조인은 Sales 에서 반환되는 데이터를 Sale_Type = S 로 제한합니다. 이렇게 하면 Sales 테이블을 기준으로 하는 모든 개체 또는 이 테이블을 통과하는 조인은 자동차 판매와 관련된 쿼리 결과만 반환합니다.



자체 제한 조인을 사용하지 않을 경우 쿼리에 대한 결과 집합은 Sale_Type 열이 'S' 또는 'R'인 행이 됩니다.

팁

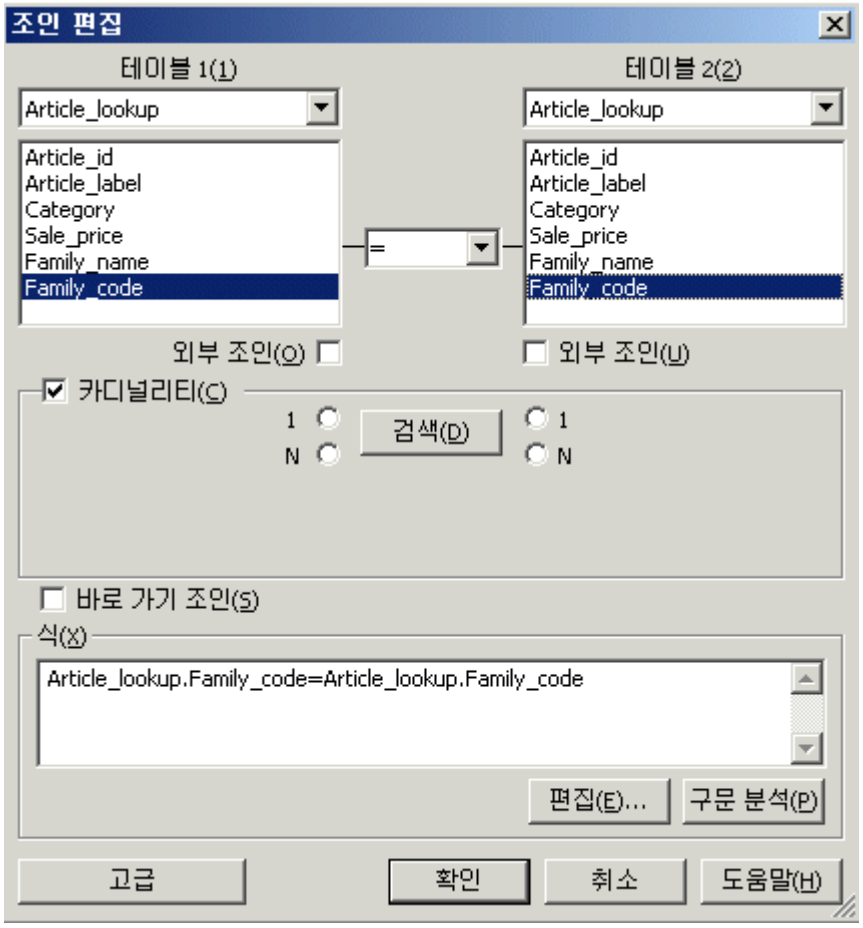
자체 제한 조인에 대한 카디널리티를 설정하면 컨텍스트를 검색할 때 '일부 카디널리티가 설정되지 않음' 메시지를 피할 수 있습니다. 카디널리티를 설정할 경우에는 실제 설정이 중요하지 않더라도 일대일로 일관성 있게 설정해야 합니다.

4.8.5.1 자체 제한 조인 만들기

자체 제한 조인을 만들려면

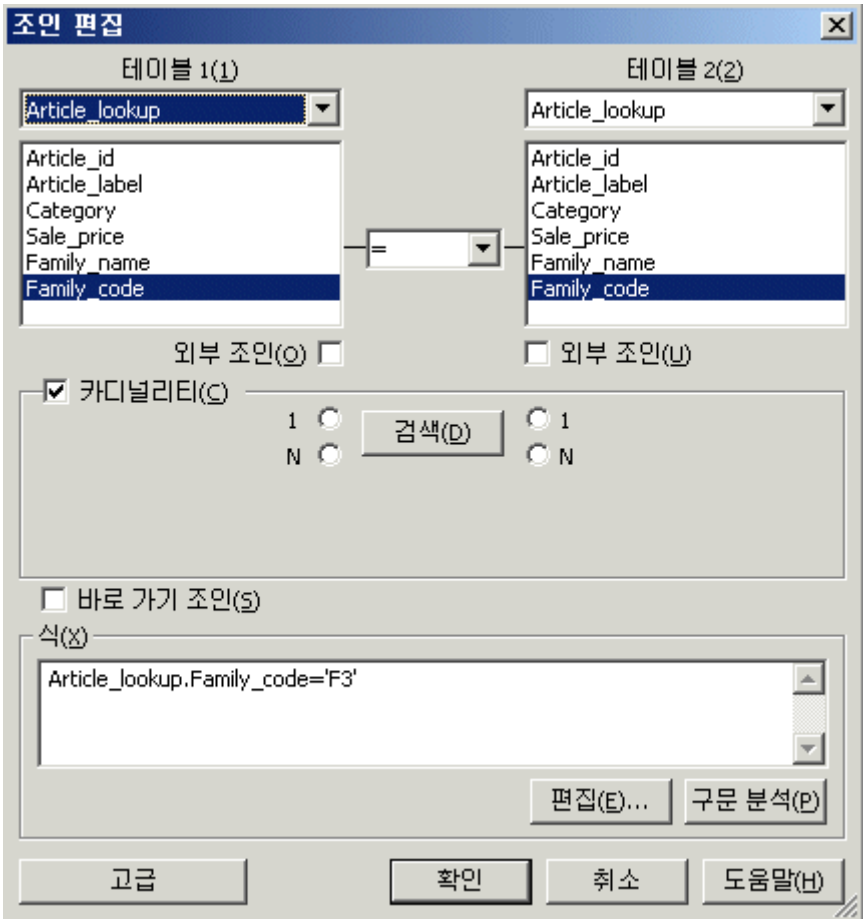
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1. 삽입 > 조인을 선택합니다.
   조인 편집 대화 상자가 나타납니다.
2. 자체 제한 조인을 설정할 테이블을 Table1 드롭다운 목록 상자에서 선택합니다.
   선택된 테이블의 열이 테이블 열 목록에 나타납니다.
3. 제한 사항을 정의하는 데 사용할 열을 열 드롭다운 목록 상자에서 클릭합니다.
4. Table1 드롭다운 목록 상자에서 선택한 동일한 테이블을 선택합니다.
5. Table1 열 목록 상자에서 선택한 동일한 열을 클릭합니다.
   조인에 대한 SQL 식이 식 텍스트 상자에 나타납니다.

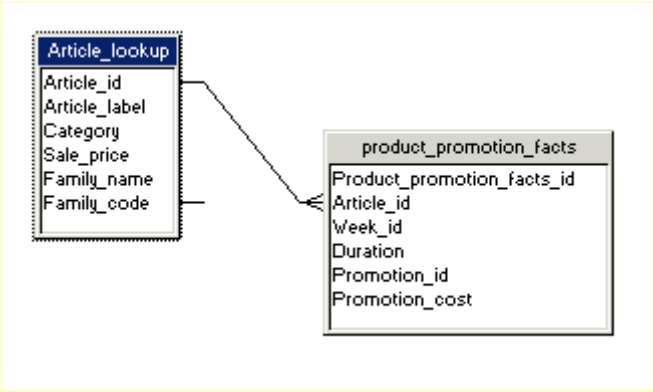


6. 조인 식의 피연산자 값을 조인 열에 대해 설정한 제한 사항 값으로 바꿉니다.
   예를 들어, Family_code 열에서 반환되는 값을 'F3'으로 제한하려는 경우 다음과 같이 = 기호 다음의 Article_lookup.Family_code 를 'F3'으로 바꿉니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



7. 구문 분석 단추를 클릭하여 구문을 검사합니다.
8. 확인을 클릭합니다.
   자체 제한 조인이 정의된 열에 대해 짧은 선으로 자체 제한 조인이 나타납니다.



4.9 카디널리티 사용

카디널리티는 한 테이블에서 얼마나 많은 행이 다른 테이블의 행과 대응하는지 설명하는 조인 속성입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

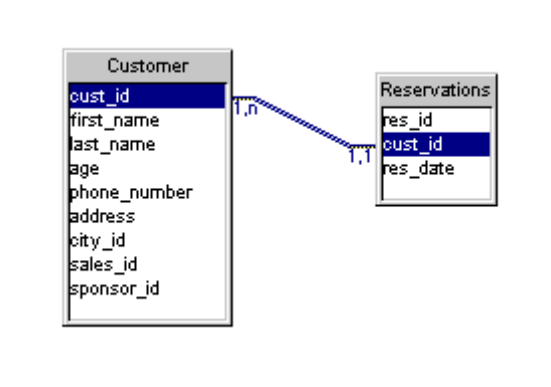
카디널리티는 조인의 다른쪽 열에 대응하는 행을 갖고 있는, 조인의 한쪽 열에 있는 행의 최소 수와 최대 수로 표현됩니다.

최소 및 최대 대응 행 수는 0, 1 또는 N 이 될 수 있습니다. 조인은 양방향 관계를 나타내므로 조인의 각 끝에 하나씩 언제나 두 개의 카디널리티가 있어야 합니다.

예

조인의 카디널리티

Customer 와 Reservations 의 두 테이블이 조인으로 연결되어 있습니다.



위의 조인에서 카디널리티는 다음과 같이 표현될 수 있습니다.

표 105:

설명	표시
각 고객에 대해 하나 이상의 예약이 있을 수 있습니다.	(1,N)
각 예약에 대해 오직 하나의 고객만 있을 수 있습니다.	(1,1)

4.9.1 유니버스 디자인 도구에서 카디널리티를 사용하는 방법

조인의 카디널리티는 쿼리를 실행할 때 생성된 SQL 에서 역할을 갖지 않습니다. 하지만 유니버스 디자인 도구에서는 카디널리티를 사용하여 컨텍스트 및 유효한 쿼리 경로를 확인합니다.

컨텍스트는 유효한 쿼리 경로를 제공하는 조인의 집합입니다. 컨텍스트를 사용하면 대상 데이터베이스에서 테이블이 연결된 방식 때문에 너무 많거나 너무 적은 행이 반환되는 조인 문제를 해결할 수 있습니다. 컨텍스트에 대한 자세한 내용은 조인 문제 검색 및 해결 [페이지 180]을 참조하십시오.

컨텍스트는 최종 사용자가 특정 조인 경로를 사용하도록 하거나 조인 경로 문제를 해결함으로써 쿼리를 위하여 생성된 SQL 에 영향을 줍니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 106:

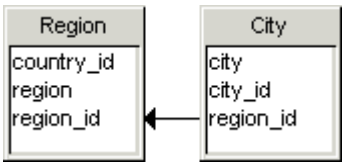
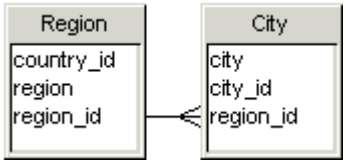
컨텍스트가 정확하고 조인 경로가 유효한지 확인하려면 스키마의 모든 조인에 대한 카디널리티가 정확하게 설정되었는지 확인해야 합니다.

카디널리티를 설정하면 테이블이 데이터베이스에서 어떻게 연관되어 있는지 이해하고, 스키마에서 발생 가능한 조인 경로 문제를 그래픽을 통해 파악하는 데도 도움이 됩니다.

4.9.1.1 카디널리티 표시

구조 창에서 다음 기호를 사용하여 카디널리티를 표시할 수 있습니다.

표 107:

카디널리티 기호	예	설명
화살표		화살표는 조인의 "한쪽"을 나타냅니다. 카디널리티가 1,1 이면 화살표의 머리가 각 조인 끝에 표시됩니다.
연결선		까마귀 발 모양은 조인의 "여러" 끝을 나타냅니다. 카디널리티가 1,1 이면 일직선이 표시됩니다.
1,N		카디널리티가 조인의 각 끝에 비율로 표시됩니다.

카디널리티를 표시하려면

1. 도구 > 옵션을 선택합니다.
   옵션 대화 상자의 일반 페이지가 열립니다.
2. 그래픽 탭을 클릭합니다.
   그래픽 페이지가 나타납니다.
3. 화살표, 까마귀 발 또는 1,n 라디오 단추를 클릭합니다.
4. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.9.2 수동으로 카디널리티 설정

조인 편집 상자에서 조인에 대한 카디널리티를 정의하여 카디널리티를 수동으로 설정할 수 있습니다.

카디널리티를 수동으로 설정하는 이유

카디널리티를 수동으로 설정할 경우에는 각 개별 조인을 고려해야 합니다. 이 방법은 스키마에서 발생할 수 있는 조인 경로 문제를 파악하는 데 도움이 됩니다. 자동으로 검색된 카디널리티만 선택할 경우에는 이러한 문제(조인 경로 끝에서 격리된 일대일 조인, 모든 열이 고유할 필요가 없는 상황에서 과도하게 많은 기본 키 등)를 찾아내지 못할 수 있습니다.

키 이해

각 테이블의 기본 키와 외래 키를 평가하여 대부분의 조인에 대한 카디널리티를 결정합니다. 기본 키와 외래 키에 대한 설명은 다음과 같습니다.

표 108:

키	설명
기본 키	테이블에서 각 행을 식별하는 값을 가진 테이블의 단일 열 또는 열 조합입니다. 기본 키는 테이블에서 행의 고유성을 보장합니다. 각 테이블에는 기본 키가 하나씩만 있습니다.
외래 키	다른 테이블의 기본 키 또는 다른 고유 키와 대응하는 데 필요한 값을 가진 열 또는 열 조합입니다. 외래 키는 '고객이 아직 생성되지 않은 경우 해당 고객에 대한 매출액을 생성할 수 없다'와 같은 제약 조건을 구현합니다. 각 테이블에는 여러 개의 외래 키가 있을 수 있습니다.

카디널리티 설정을 위한 기준

기본 키와 외래 키 사이의 관계를 평가하여 다음과 같이 조인에 대한 카디널리티를 결정합니다.

표 109:

조인의 연결 대상	카디널리티
테이블 1 의 전체 기본 키와 테이블 2 의 전체 기본 키. 예:	일대일(1,1). 각 기본 키 값에 대해 각 테이블에서 하나의 행만 반환됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 110:

조인의 연결 대상	카디널리티
테이블 1 의 전체 기본 키와 테이블 2 의 해당하는 외래 키. 예:	일대다(1,N). 테이블의 외래 키 값에 대하여 고유성이 보장되지 않으므로 테이블 1 의 기본 키 단일 값에 대해 대응하는 많은 값이 반환될 수 있습니다.

표 111:

조인의 연결 대상	카디널리티
테이블 1 의 전체 기본 키와 테이블 2 의 일부 기본 키. 예:	일대다(1,N). 불완전한 기본 키 대응의 경우 테이블 1 의 기본 키 단일 값에 대해 대응하는 많은 값이 반환될 수 있습니다.

4.9.2.1 수동으로 카디널리티 설정

조인 편집 상자에서 조인에 대한 카디널리티를 정의하여 카디널리티를 수동으로 설정할 수 있습니다.

카디널리티를 수동으로 설정하는 이유

카디널리티를 수동으로 설정할 경우에는 각 개별 조인을 고려해야 합니다. 이 방법은 스키마에서 발생할 수 있는 조인 경로 문제를 파악하는 데 도움이 됩니다. 자동으로 검색된 카디널리티만 선택할 경우에는 이러한 문제(조인 경로 끝에서 격리된 일대일 조인, 모든 열이 고유할 필요가 없는 상황에서 과도하게 많은 기본 키 등)를 찾아내지 못할 수 있습니다.

키 이해

각 테이블의 기본 키와 외래 키를 평가하여 대부분의 조인에 대한 카디널리티를 결정합니다. 기본 키와 외래 키에 대한 설명은 다음과 같습니다.

표 112:

키	설명
기본 키	테이블에서 각 행을 식별하는 값을 가진 테이블의 단일 열 또는 열 조합입니다. 기본 키는 테이블에서 행의 고유성을 보장합니다. 각 테이블에는 기본 키가 하나씩만 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

키	설명
외래 키	다른 테이블의 기본 키 또는 다른 고유 키와 대응하는 데 필요한 값을 가진 열 또는 열 조합입니다. 외래 키는 '고객이 아직 생성되지 않은 경우 해당 고객에 대한 매출액을 생성할 수 없다'와 같은 제약 조건을 구현합니다. 각 테이블에는 여러 개의 외래 키가 있을 수 있습니다.

카디널리티 설정을 위한 기준

기본 키와 외래 키 사이의 관계를 평가하여 다음과 같이 조인에 대한 카디널리티를 결정합니다.

표 113:

조인의 연결 대상	카디널리티
테이블 1 의 전체 기본 키와 테이블 2 의 전체 기본 키. 예:	일대일(1,1). 각 기본 키 값에 대해 각 테이블에서 하나의 행만 반환됩니다.

표 114:

조인의 연결 대상	카디널리티
테이블 1 의 전체 기본 키와 테이블 2 의 해당하는 외래 키. 예:	일대다(1,N). 테이블의 외래 키 값에 대하여 고유성이 보장되지 않으므로 테이블 1 의 기본 키 단일 값에 대해 대응하는 많은 값이 반환될 수 있습니다.

표 115:

조인의 연결 대상	카디널리티
테이블 1 의 전체 기본 키와 테이블 2 의 일부 기본 키. 예:	일대다(1,N). 불완전한 기본 키 대응의 경우 테이블 1 의 기본 키 단일 값에 대해 대응하는 많은 값이 반환될 수 있습니다.

4.9.2.2 자동으로 카디널리티 검색

유니버스 디자인 도구의 카디널리티 검색 기능을 사용하여 다음에 대해 카디널리티를 자동으로 검색할 수 있습니다.

- 선택된 조인
- 모든 조인
- 조인 생성 시
- 조인 편집 상자에서

자동 카디널리티 검색 기능을 사용하면 검색 시 카디널리티가 자동으로 구현됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

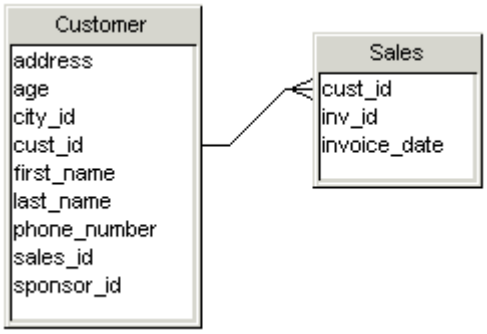
> **노트**
>
> 자동 카디널리티 기능은 적절하게 사용해야 합니다. 이 기능은 스키마에서 모든 카디널리티를 신속하게 검색하는 데 매우 유용하지만, 많은 관계형 데이터베이스에는 부정확한 카디널리티 검색 결과를 반환할 수 있는 여러 가지 구조적인 문제가 내재되어 있습니다. 여기에는 불완전한 기본 조인 및 과도하게 엔지니어링된 기본 키가 포함됩니다. 자세한 내용은 자동 카디널리티 검색 최적화 [페이지 170] 단원에서 설명합니다.

선택된 조인에 대해 자동으로 카디널리티 검색

선택된 조인에 대해 자동으로 카디널리티를 검색하려면

- 조인을 클릭하고 도구 > 카디널리티 검색을 선택합니다.
- 조인을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 카디널리티 검색을 선택합니다.

카디널리티가 끝이 여러 개인 까마귀 발 모양으로 표시됩니다.



조인을 선택하지 않고 바로 도구 > 카디널리티 검색을 선택하면 조인이 선택되지 않았음을 알리고 모든 조인에 대해 카디널리티를 검색할 것인지 여부를 묻는 메시지가 나타납니다.

모든 조인에 대해 자동으로 카디널리티 검색

모든 조인에 대해 자동으로 카디널리티를 검색하려면

1. 도구 > 자동화된 검색 > 카디널리티 검색을 선택합니다.
   또는

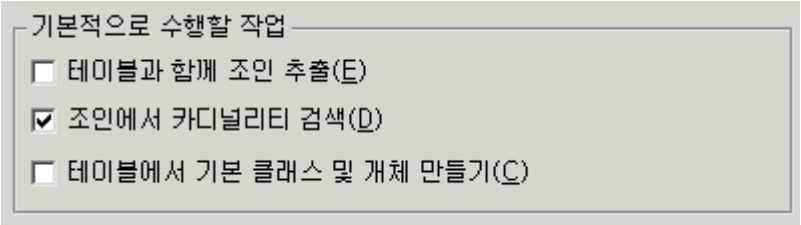
   카디널리티 검색 단추를 클릭합니다.
   모든 조인에 대해 카디널리티를 검색할 것인지 여부를 묻는 메시지 상자가 나타납니다.
2. 예를 클릭합니다.
   구조 창에 모든 조인이 카디널리티와 함께 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조인 생성 시 자동으로 카디널리티 검색

조인 생성 시 자동으로 카디널리티를 검색하려면

1. 도구 > 옵션을 선택합니다.
   옵션 대화 상자의 일반 페이지가 열립니다.
2. 데이터베이스 탭을 클릭합니다.
   데이터베이스 페이지가 나타납니다.
3. 조인에서 카디널리티 검색 확인란을 선택합니다.



4. 확인을 클릭합니다.
5. 새 조인을 만들 때 카디널리티가 자동으로 검색되고 조인에 표시됩니다.

조인 편집 상자에서 자동으로 카디널리티 검색

조인 편집 상자에서 자동으로 카디널리티를 검색하려면

1. 조인을 두 번 클릭합니다.
   조인 편집 대화 상자가 나타납니다.
2. 카디널리티 확인란을 선택합니다.
3. 검색 단추를 클릭합니다.
   검색된 카디널리티에 대해 카디널리티 라디오 단추가 자동으로 선택됩니다. 두 개의 카디널리티는 문장 형식으로 표현됩니다.

4. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4.9.2.3 자동 카디널리티 검색 최적화

대상 RDBMS 의 PRM 파일에서 매개 변수를 수정하여 카디널리티 검색 응답 시간을 단축할 수 있습니다. 이것은 검색 알고리즘에서 3 개의 SQL 문 대신 2 개만 읽도록 하여 알고리즘의 성능을 개선하는 방법입니다.

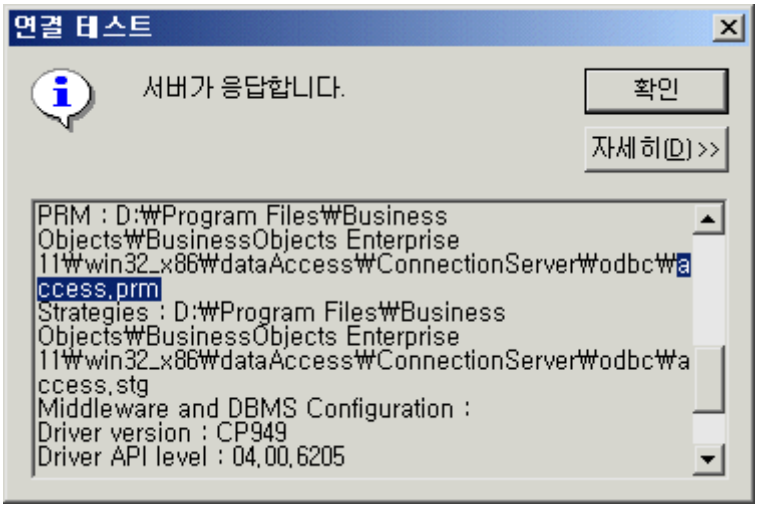
PRM 파일은 Web Intelligence 에서 유니버스 생성 및 SQL 쿼리 생성을 구성하는 데 사용되는 매개 변수가 포함된 텍스트 파일입니다. 지원되는 각 RDBMS 에 대해 하나의 PRM 파일이 존재합니다.

PRM 파일은 <INSTALLDIR>\win32_x86\dataAccess\ConnectionServer\ 아래의 데이터베이스 폴더에 있습니다.

연결에서 사용되는 **PRM** 파일 확인

유니버스 연결에서 사용되는 PRM 파일을 확인하려면

1. 파일 > 매개 변수를 선택합니다.
   매개 변수 대화 상자가 나타납니다.
2. 테스트 단추를 클릭합니다.
   연결 테스트 메시지 상자가 나타납니다.
3. 자세히 단추를 클릭합니다.
   유니버스에 대한 세부 정보가 드롭다운 메시지 상자에 나타납니다.
4. 메시지 상자를 아래로 스크롤하여 PRM 으로 시작하는 줄을 찾습니다.
   이 줄은 활성 유니버스에서 현재 사용되는 PRM 파일의 경로와 이름을 나타냅니다.



5. 확인을 클릭합니다.
   매개 변수 대화 상자로 돌아옵니다.
6. 취소를 클릭합니다.

PRM 파일을 사용하여 카디널리티 검색 최적화

PRM 파일을 사용하여 카디널리티 검색을 최적화하려면

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1. 텍스트 편집기에서 대상 데이터베이스에 대한 PRM 파일을 엽니다.
   PRM 파일은 Business Objects 경로의 Data Access 폴더에 있습니다.
2. LIGHT_DETECT_CARDINALITY 매개 변수를 YES 로 설정합니다.
3. PRM 파일을 저장하고 닫습니다.
   다음에 유니버스를 열 때 자동 카디널리티 검색이 최적화됩니다.

4.9.2.4 자동 카디널리티 검색 최적화

대상 RDBMS 의 PRM 파일에서 매개 변수를 수정하여 카디널리티 검색 응답 시간을 단축할 수 있습니다. 이것은 검색 알고리즘에서 3 개의 SQL 문 대신 2 개만 읽도록 하여 알고리즘의 성능을 개선하는 방법입니다.

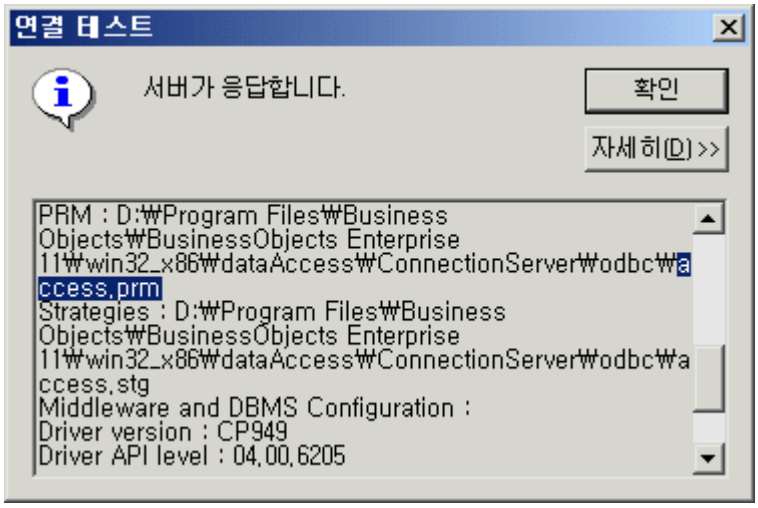
PRM 파일은 Web Intelligence 에서 유니버스 생성 및 SQL 쿼리 생성을 구성하는 데 사용되는 매개 변수가 포함된 텍스트 파일입니다. 지원되는 각 RDBMS 에 대해 하나의 PRM 파일이 존재합니다.

PRM 파일은 <INSTALLDIR>\win32_x86\dataAccess\ConnectionServer\ 아래의 데이터베이스 폴더에 있습니다.

연결에서 사용되는 **PRM** 파일 확인

유니버스 연결에서 사용되는 PRM 파일을 확인하려면

1. 파일 > 매개 변수를 선택합니다.
   매개 변수 대화 상자가 나타납니다.
2. 테스트 단추를 클릭합니다.
   연결 테스트 메시지 상자가 나타납니다.
3. 자세히 단추를 클릭합니다.
   유니버스에 대한 세부 정보가 드롭다운 메시지 상자에 나타납니다.
4. 메시지 상자를 아래로 스크롤하여 PRM 으로 시작하는 줄을 찾습니다.
   이 줄은 활성 유니버스에서 현재 사용되는 PRM 파일의 경로와 이름을 나타냅니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5. 확인을 클릭합니다.
   매개 변수 대화 상자로 돌아옵니다.
6. 취소를 클릭합니다.

PRM 파일을 사용하여 카디널리티 검색 최적화

PRM 파일을 사용하여 카디널리티 검색을 최적화하려면

1. 텍스트 편집기에서 대상 데이터베이스에 대한 PRM 파일을 엽니다.
   PRM 파일은 Business Objects 경로의 Data Access 폴더에 있습니다.
2. LIGHT_DETECT_CARDINALITY 매개 변수를 YES 로 설정합니다.
3. PRM 파일을 저장하고 닫습니다.
   다음에 유니버스를 열 때 자동 카디널리티 검색이 최적화됩니다.

4.10 유니버스 검사

유니버스를 디자인할 때는 무결성을 주기적으로 테스트해야 합니다. 다음과 같이 유니버스의 무결성을 확인할 수 있습니다.

표 116:

유니버스 검사	설명
자동	유니버스를 만들거나 유니버스를 내보내거나 유니버스를 열 때 유니버스 구조의 SQL 구문을 검사하도록 유니버스 디자인 도구 옵션을 설정할 수 있습니다.
수동	선택한 유니버스 구조에 대해 무결성 검사를 직접 실행할 수 있습니다.

4.10.1 자동으로 유니버스 무결성 검사

유니버스 디자인 도구에서는 유니버스를 만들거나 유니버스를 내보내거나 유니버스를 열 때 SQL 의 구문을 분석하도록 다음과 같이 무결성 검사 옵션을 설정할 수 있습니다.

표 117:

자동 검사 옵션	설명
정의할 때 자동으로 구문 분석	모든 개체, 조건 및 조인을 만들 때 해당 SQL 정의를 유니버스 디자인 도구에서 자동으로 검사합니다. 이 옵션은 구조 만들기 확인 메시지에서 확인을 클릭했을 때 적용됩니다.

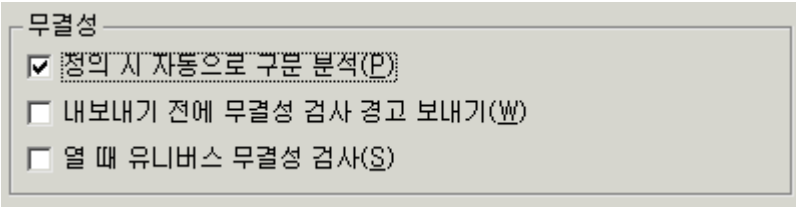
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

자동 검사 옵션	설명
무결성 검사 경고 보내기	검사되지 않은 유니버스를 내보내려고 할 때마다 유니버스 디자인 도구에서 경고를 표시합니다.
열 때 유니버스 무결성 검사	유니버스를 열 때마다 자동으로 검사합니다.

4.10.1.1 유니버스 자동 검사 옵션 설정

유니버스 자동 검사 옵션을 설정하려면

1. 도구 > 옵션을 선택합니다.
   옵션 대화 상자의 일반 페이지가 열립니다.
2. 무결성 그룹 상자에서 적절한 유니버스 자동 검사 옵션의 확인란을 선택하거나 선택 취소합니다.



3. 확인을 클릭합니다.

4.10.1.2 수동으로 유니버스 무결성 검사

무결성 검사를 사용하면 활성 유니버스의 디자인이 정확하고 최신 상태인지 테스트할 수 있습니다.

무결성 검사는 다음 사항을 검색합니다.

- 유니버스 내의 개체, 조인, 조건, 카디널리티 등의 오류
- 조인 경로의 루프
- 필요한 컨텍스트
- 대상 데이터베이스에 대한 변경 사항

이 기능은 데이터베이스 요소를 기준으로 유니버스의 요소를 검사하기 전에 데이터베이스에 대한 연결이 유효한지 확인합니다. 연결이 유효하지 않으면 무결성 검사가 중지되고 오류 메시지가 반환됩니다.

4.10.1.3 무결성 검사로 검색할 수 있는 오류 유형

무결성 검사로는 다음 사항을 검색할 수 있습니다.

- 개체, 조건 또는 조인의 SQL 정의에 있는 잘못된 구문
- 루프

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 끊어진 테이블
- 끊어진 조인
- 컨텍스트 내의 루프
- 누락되거나 잘못된 카디널리티

연결된 데이터베이스의 변경 사항을 무결성 검사로 확인하는 방법

무결성 검사 기능은 데이터베이스에 테이블 목록 요청을 보냅니다. 그런 다음 이 목록을 유니버스 내의 테이블과 비교합니다. 열에 대해서도 동일한 작업이 수행됩니다.

데이터베이스의 테이블 목록과 일치하지 않는 모든 테이블이나 열은 존재하지 않는 항목으로 구조 창에 표시됩니다. 이들 테이블 또는 열은 데이터베이스에서 삭제되었거나 이름이 바뀌었을 수 있습니다. 무결성 검사를 사용하여 유니버스의 무결성 확인 [페이지 174] 단원을 참조하십시오.

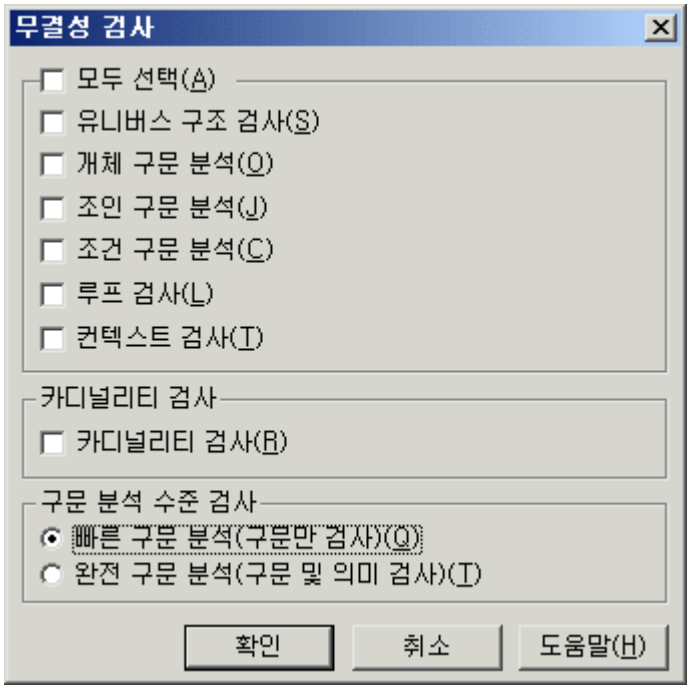
> **노트**
>
> 데이터 양이 많으면 카디널리티 검사를 실행하는 데 시간이 많이 걸릴 수 있습니다. 또한 데이터가 정확하지 않거나 손실되었을 때는 결과가 정확하지 않을 수 있습니다. 따라서, 데이터베이스 크기가 크고 완전하지 않은 데이터 항목이 포함되어 있는 경우에는 카디널리티 검사 옵션을 선택하지 않는 것이 좋습니다. 그래도 이 옵션을 사용하려면 PRM 파일을 수정하여 카디널리티 검색을 최적화할 수 있습니다. 자세한 내용은 자동 카디널리티 검색 최적화 [페이지 170] 단원을 참조하십시오.

4.10.1.4 무결성 검사를 사용하여 유니버스의 무결성 확인

유니버스 무결성을 검사하려면

1. 도구 > 무결성 검사를 선택합니다.
   또는
   무결성 검사 단추를 클릭합니다.
2. 무결성 검사 대화 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

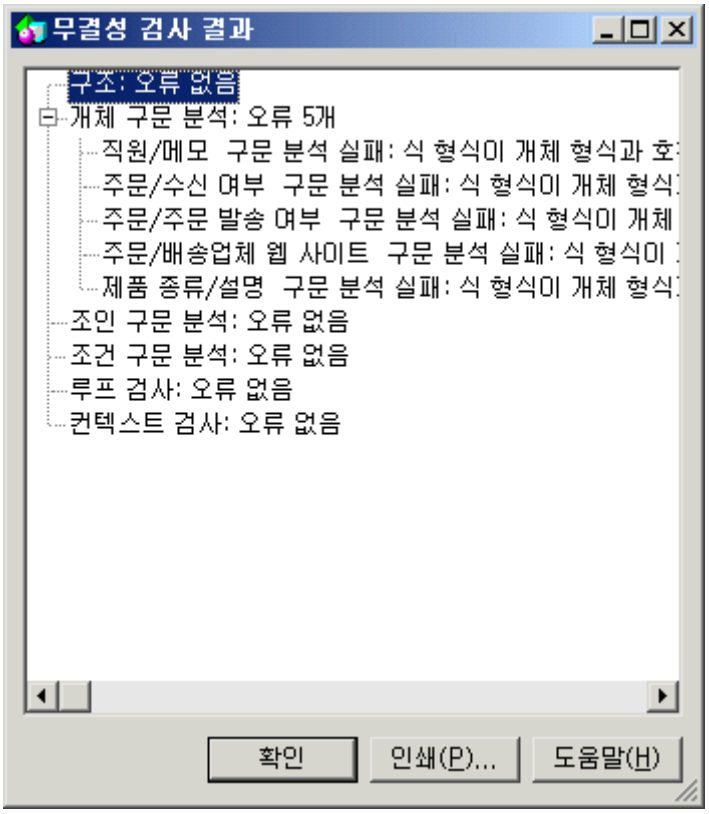


3. 검사할 구성 요소에 해당하는 확인란을 선택합니다.

> **노트**
>
> 카디널리티 검사는 모두 선택 옵션과 무관하게 선택할 수 있습니다. 이를 통해 데이터베이스에 따라 시간이 오래 걸릴 수 있는 카디널리티 검사 없이 유니버스 구조를 확인할 수 있습니다.

4. 검사하지 않을 구성 요소에 해당하는 확인란을 선택 취소합니다.
5. 빠른 구문 분석 확인란을 선택하여 구성 요소의 구문만 확인합니다.
   또는
   자세한 구문 분석 확인란을 선택하여 구성 요소의 구문과 의미를 모두 확인합니다.
6. 확인을 클릭합니다.
   유니버스 검사 진행률을 보여 주는 메시지 상자가 나타납니다.

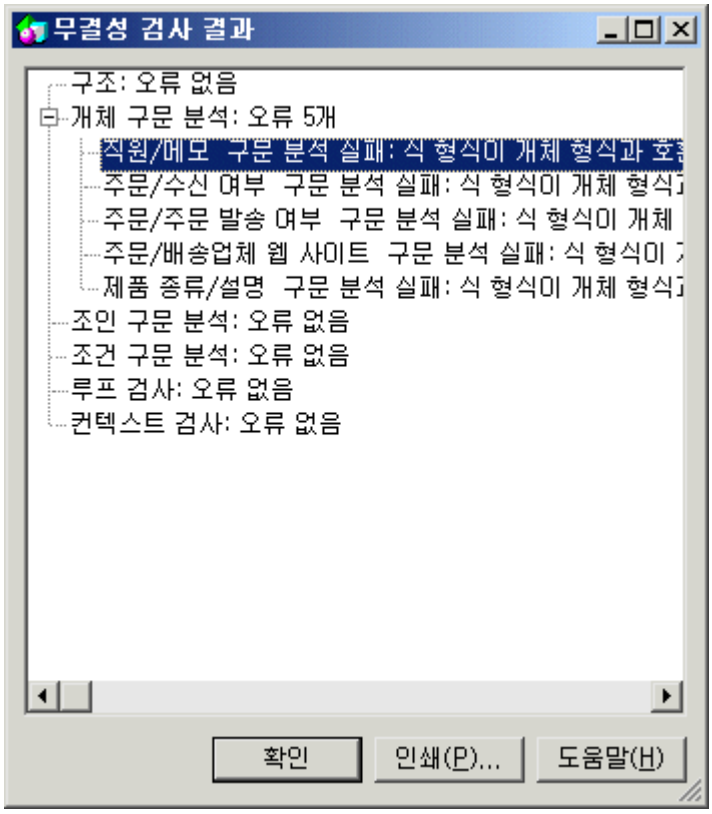
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



오류가 발견되지 않으면 각 오류 유형 옆에 "확인"이 표시됩니다.

7. 오류 유형 옆에 있는 더하기 기호(+)를 클릭하여 오류가 발생한 구성 요소 목록을 표시합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



목록에서 항목을 두 번 클릭하면 구조 창에 해당 구성 요소가 강조 표시됩니다.

8. 인쇄 단추를 클릭하여 창 내용을 인쇄합니다.
9. 확인을 클릭합니다.

> **노트**
>
> 루프 검사 확인란을 선택하려면 먼저 조인의 카디널리티를 검색해야 합니다. 그렇지 않으면 조인에서 루프를 식별할 때 오류가 발생합니다.

4.10.1.5 유니버스 구조 새로 고치기

무결성 검사를 실행한 결과 유니버스에 연결되어 있는 데이터베이스가 수정된 것으로 확인되면 구조 새로 고침을 사용하여 구조 창의 내용을 업데이트할 수 있습니다.

구조 새로 고침을 사용하면 다음과 같은 데이터베이스 변경 사항에 따라 유니버스 구조를 수정할 수 있습니다.

표 118:

조건	수행되는 작업
테이블에 열 추가	해당하는 유니버스 테이블에 열을 추가합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조건	수행되는 작업
테이블에서 열 제거	삭제해야 할 열과 관련 조인을 알려주는 경고 메시지를 표시합니다.
데이터베이스에서 테이블 제거	삭제해야 할 테이블과 관련 조인을 알려주는 경고 메시지를 표시합니다.
데이터베이스에서 테이블 이름 변경	해당하는 유니버스 테이블을 인식할 수 없다는 메시지를 표시합니다. 데이터베이스의 테이블과 일치하도록 이들 테이블의 이름을 바꿔야 합니다. 바꾼 이름도 일치하지 않으면 유니버스 디자인 도구에서는 이름이 바뀐 테이블이 데이터베이스에 없음을 알리는 메시지를 반환합니다.
데이터베이스 변경 사항 없음	유니버스를 업데이트하지 않아도 된다는 메시지를 표시합니다.

유니버스 구조를 새로 고치려면

- 보기 > 구조 새로 고침을 선택합니다.
- 데이터베이스 변경 사항을 알려주거나 변경 사항이 없는 경우 업데이트하지 않아도 된다는 사실을 알려주는 메시지 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5 스키마의 조인 문제 해결

5.1 개요

이 장에서는 스키마의 테이블 간에 조인을 만들었을 때 발생할 수 있는 여러 가지 문제에 대해 설명합니다. 여기에서는 조인 경로를 사용하는 쿼리를 유니버스에서 실행했을 때 올바른 결과가 반환될 수 있도록 조인 문제를 검색하고 해결하는 방법을 알려 줍니다. 유니버스를 작성하기 전에 조인 문제를 해결해야 합니다.

5.2 조인 경로 문제란?

조인 경로란 조인으로 연결된 테이블의 데이터에 액세스하기 위해 쿼리에서 사용할 수 있는 일련의 조인을 말합니다.

조인 경로 문제는 관계형 데이터베이스에서 조회 테이블과 팩트 테이블을 연결하는 방식에 대한 제한으로 인해 발생할 수 있습니다. 스키마를 디자인할 때 발생할 수 있는 세 가지 조인 경로 문제는 다음과 같습니다.

- 루프
- 캐즘 트랩
- 팬 트랩

별칭(기본 테이블의 복사본) 및 컨텍스트(정의된 조인 경로)를 만들고 유니버스 디자인 도구에서 제공하는 기능을 사용하여 계수 또는 컨텍스트에 대한 쿼리를 구분함으로써 이러한 문제를 모두 해결할 수 있습니다.

이 단원에서는 조회 테이블과 팩트 테이블을 간략하게 정의하고 이들 테이블을 사용할 때 발생할 수 있는 조인 경로 문제의 유형에 대해 설명합니다. 여기에는 별칭, 컨텍스트 및 기타 유니버스 디자인 도구 기능을 사용하여 유니버스 스키마에 발생한 조인 경로 문제를 해결할 수 있는 방법에 대해 설명합니다.

일반적으로 유니버스 디자인 도구에서는 조회 테이블과 팩트 테이블 간의 조인을 만듭니다.

5.2.1 조회 테이블이란?

조회(또는 차원) 테이블에는 특정 엔터티나 주제와 관련된 정보가 들어 있습니다. 예를 들어, 조회 테이블에는 고객의 이름, 전화 번호 및 고객이 거주하고 있는 도시와 국가 등의 지리적인 정보가 포함될 수 있습니다.

일반적으로 유니버스 디자인 도구에서 차원 개체와 설명 개체는 조회 테이블에서 파생됩니다.

5.2.2 팩트 테이블이란?

팩트 테이블에는 트랜잭션에 대한 통계 정보가 들어 있습니다. 예를 들어, 이 테이블에는 판매 수익 또는 이익과 같은 값이 포함될 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스에서 대부분의 계수는 팩트 테이블에서 정의됩니다.

5.2.3 잘못된 결과를 반환하는 조인 경로 유형

관계형 데이터베이스에서는 조인을 수행할 수 있는 방법의 제한으로 인해 쿼리가 잘못된 결과를 반환할 수 있습니다. 테이블 스키마에서 조회 테이블과 팩트 테이블이 연관된 방식에 따라 조인 경로에서 쿼리의 결과 행이 너무 적거나 너무 많은 인스턴스가 작성될 수 있습니다.

다음과 같은 조인 경로는 잘못된 결과를 가져올 수 있습니다.

표 119:

조인 경로 유형	반환값	설명
루프	행 수가 너무 적음	조회 테이블 사이에 조인 경로가 여러 개 있습니다.
다대일 조인 수렴	행 수가 너무 많음	두 개의 팩트 테이블에서 시작된 다대일(many to one) 조인이 단일 조회 테이블에 수렴됩니다. 이 유형의 조인 수렴은 캐즘 트랩이라는 조인 경로 문제의 원인이 될 수 있습니다.
연속 다대일 조인	행 수가 너무 많음	일대다(one to many) 조인으로 연결된 테이블에 일대다 조인으로 연결되어 있습니다. 이러한 방식의 일대다 조인은 팬 트랩이라는 조인 경로 문제의 원인이 될 수 있습니다.

5.2.4 조인 문제 검색 및 해결

유니버스 디자인 도구에서는 여러 가지 방법으로 조인 문제를 검색하고 해결할 수 있습니다. 각 방법은 해당 단원에서 자세하게 설명됩니다.

다음과 같은 방법으로 조인 경로 문제를 검색하고 해결할 수 있습니다.

표 120:

조인 문제	검색 방법	해결 방법
루프	• 별칭 검색 • 컨텍스트 검색 • 루프 검색 • 무결성 검사 • 시각적으로 스키마 분석	별칭 및 컨텍스트를 만들어 루프를 중단합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조인 문제	검색 방법	해결 방법
캐즘 트랩(다대일 조인 수렴)	시각적으로 테이블 스키마 분석	• 컨텍스트를 만듭니다. • 각 계수에 여러 SQL 문 사용 옵션을 사용합니다. • 다중 유니버스를 만듭니다(Web Intelligence 만 해당).
팬 트랩(연속 다대일 조인)	시각적으로 테이블 스키마 분석	• 별칭을 만들고 별칭을 사용하여 컨텍스트를 만든 다음 영향을 받는 계수 개체를 별칭에서 만듭니다. • 각 계수에 여러 SQL 문 사용 옵션을 사용합니다.

대부분의 조인 경로 문제는 별칭을 만들거나 컨텍스트를 구현하여 해결할 수 있습니다. 유니버스 디자인 도구에서는 루프 자동 검색 도구를 사용하여 스키마에서 루프를 식별하고 컨텍스트 자동 검색 도구를 사용하여 캐즘 트랩이 발생한 위치를 식별할 수 있습니다. 그러나 팬 트랩을 해결하려면 스키마를 시각적으로 분석하고 별칭을 만들 수 있어야 하며, 필요한 경우에는 컨텍스트를 수동으로 만들어야 합니다.

5.3 별칭 정의

스키마에서 별칭은 기존 테이블에 대한 참조입니다. 별칭이란 원본 테이블(기본 테이블)과 이름만 다른 복제본입니다. 이 테이블의 데이터는 원본 테이블의 데이터와 동일하지만 이름이 다르기 때문에 쿼리 SQL 에서는 두 개의 서로 다른 테이블을 사용하는 것으로 "착각"하여 인식합니다.

Beach 유니버스 스키마에는 두 개의 별칭 테이블 Resort_Country(Country 테이블의 별칭) 및 Sponsor(Customer 테이블의 별칭)가 있습니다. 각 별칭 테이블의 원래 테이블 이름은 괄호로 표시됩니다.

5.3.1 스키마에서 별칭을 사용하는 방법

크게 다음과 같은 두 가지 이유로 별칭을 사용합니다.

- 쿼리에서 테이블을 두 번 이상 사용하기 위해 별칭을 사용합니다. 주로 이런 용도로 별칭을 사용하며 루프 및 팬 트랩을 해결하기 위해 별칭을 사용하는 것도 여기에 포함됩니다. Beach 유니버스 예제에는 두 개의 별칭, 즉 Country 의 별칭인 Resort_Country 와 Customer 의 별칭인 Sponsor 가 들어 있습니다.
- SQL 을 직접 작성할 때 테이블 이름을 간단하게 입력하기 위해 별칭을 사용합니다.

➔ 팁

스키마에 테이블을 삽입할 때 각 테이블에 대해 별칭을 만들 수도 있습니다. 그런 다음 원본 기본 테이블이 아니라 별칭 테이블을 사용하여 스키마를 만듭니다. 이 경우 주 유니버스 구조와는 별도로 기본 테이블들을 함께 배치합니다. 이렇게 하면 테이블에 의미 있는 이름을 사용할 수 있으며, 이후 단계에서 기본 테이블에 별칭을 사용해야 하는 경우가 생기더라도 유니버스 구조의 주요 섹션을 다시 구성하지 않아도 됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.3.1.1 별칭을 사용하여 루프 해결

유니버스를 개발할 때 별칭은 주로 일반 테이블을 사용할 때 발생할 수 있는 루프를 해결하는 데 사용됩니다. 루프는 스키마에 포함된 테이블 집합 사이의 닫힌 경로를 정의하는 조인 집합입니다. 루프는 조회 테이블 사이에 여러 개의 조인 경로가 만들어졌을 때 발생합니다.

별칭을 사용하면 여러 개의 쿼리 경로에 사용되는 원본 조회 테이블에 대해 대체 테이블을 만들어 루프를 중단할 수 있습니다. 이와 같은 별칭 사용은 루프 해결 [페이지 195] 단원에서 설명합니다.

5.3.1.2 별칭을 사용하여 팬 트랩 해결

별칭은 잠재적인 팬 트랩을 해결하는 데도 사용됩니다. 팬 트랩은 조인에서 "다"(many)쪽에 집계가 누적되는 경우에 필요 이상의 결과가 반환되는 연속 일대다 조인 경로에서 발생할 수 있습니다. 이와 같은 별칭 사용은 캐즘 트랩 해결 [페이지 220] 단원에서 설명합니다.

5.3.2 별칭 만들기

별칭을 수동으로 만들 수도 있고, 유니버스 디자인 도구에서 조인 경로 루프를 해결할 잠재적인 별칭을 자동으로 검색하도록 할 수도 있습니다.

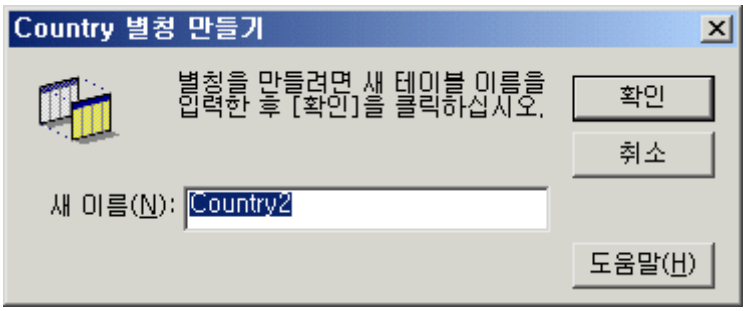
팬 트랩을 해결할 때는 별칭을 수동으로 만들어야 합니다. 기본 테이블을 사용하지 않고 별칭만 사용하여 스키마를 만들 때도 별칭을 수동으로 만들어야 합니다.

루프 해결을 위한 별칭 자동 검색 및 만들기는 별칭 검색 및 만들기 [페이지 206] 단원에서 설명합니다.

5.3.2.1 수동으로 별칭 만들기

수동으로 별칭을 만들려면

1. 별칭을 만드는 데 사용할 테이블을 클릭합니다.
2. 삽입 > 별칭을 선택합니다.
   또는
   별칭 삽입 단추를 클릭합니다.
   별칭 만들기 상자가 나타납니다. 이 상자에는 새 별칭 이름을 입력하라는 메시지가 표시됩니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. 별칭을 사용할 테이블의 새 이름을 입력하거나 자동으로 입력된 이름을 그대로 사용합니다.

> **노트**
>
> 별칭으로 지정하는 이름은 기본 테이블과 구별할 수 있도록 별칭의 역할과 관련된 것으로 지정해야 합니다. 예를 들어, Resort_Country 는 Country 의 별칭입니다. Resort_Country 는 휴양지 국가에 대한 데이터를 반환하는 쿼리에 사용되며 기본 테이블인 Country 는 고객 국가에 대한 데이터를 반환하는 쿼리에 사용됩니다.

4. 확인을 클릭합니다.
   별칭으로 지정한 테이블이 구조 창에 표시됩니다.
5. 스키마에서 별칭과 기타 테이블 사이에 필요한 조인을 만듭니다.

> **팁**
>
> 도구 > 옵션 > 그래픽을 선택한 다음 별칭 이름 확인란을 선택하여 테이블 제목에서 해당 기본 테이블의 이름과 함께 별칭을 표시하면 기본 테이블과 별칭이 혼동되지 않도록 만들 수 있습니다.

5.3.2.2 별칭 이름 바꾸기

별칭 이름은 언제든지 바꿀 수 있습니다. 별칭 및 테이블 이름 지정 규칙은 RDBMS 에 따라 달라집니다. 별칭 이름을 바꿀 때는 테이블 이름을 직접 바꾸거나 유니버스의 별칭 목록에서 선택할 수 있습니다.

직접 별칭 이름 바꾸기

별칭 이름을 직접 바꾸려면

1. 테이블을 클릭한 다음 편집 > 테이블 이름 바꾸기를 선택합니다.
   또는
   테이블을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 테이블 이름 바꾸기를 선택합니다.
   테이블 이름 바꾸기 대화 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 테이블 이름 상자에 새 이름을 입력합니다.
   소유자 및 자격 필드를 사용할 수 있는지 여부는 데이터베이스에 따라 달라집니다. 이들 필드가 활성화되어 있는 경우에는 필요에 따라 값을 수정할 수 있습니다.
3. 별칭 이름을 대문자로만 표시하려면 대문자 확인란을 선택합니다.
   또는
   별칭 이름을 소문자로만 표시하려면 소문자 확인란을 선택합니다.
4. 확인을 클릭합니다.

목록에서 별칭 이름 바꾸기

목록에서 별칭 이름을 바꾸려면

1. 도구 > 별칭 목록을 선택합니다.
2. 별칭 목록이 나타납니다. 이 목록에는 활성 유니버스의 별칭 전체가 표시됩니다.
3. 목록에서 별칭 이름을 클릭합니다.
4. 선택한 별칭의 새 이름을 새 이름 텍스트 상자에 입력합니다.
5. 적용을 클릭합니다.
6. 확인을 클릭합니다.

5.3.2.3 별칭 삭제

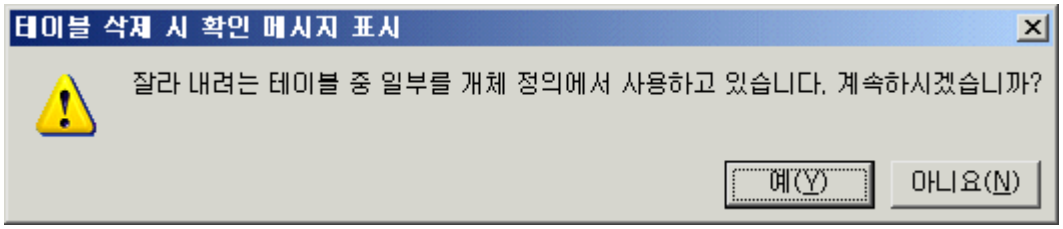
테이블을 삭제할 때와 같은 방법으로 별칭을 삭제합니다. 별칭을 사용하여 개체를 정의한 경우에는 다른 테이블을 사용하도록 개체를 수정한 후에 별칭을 삭제하거나, 더 이상 필요하지 않은 개체인 경우에는 별칭과 함께 삭제합니다.

Web Intelligence 에서는 삭제한 별칭을 사용하는 개체를 수정하거나 제거하지 않으면 해당 개체를 사용하는 쿼리에서 오류가 발생합니다.

별칭을 삭제하려면

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1. 별칭을 클릭한 다음 편집 > 지우기를 선택합니다.
   또는
   별칭을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 지우기를 선택합니다
   또는
   별칭을 클릭한 다음 Delete 키를 누릅니다.
   별칭을 사용하는 개체가 있으면 다음 메시지가 표시됩니다.



   별칭을 사용하는 개체가 없으면 확인 상자가 표시되지 않고 별칭이 즉시 삭제됩니다.
2. 예를 클릭합니다.
   구조 창에서 별칭이 삭제됩니다.

5.4 컨텍스트 정의

컨텍스트는 Web Intelligence 에서 SQL 을 생성할 수 있도록 유효한 쿼리 경로를 제공하는 조인 모음입니다.

5.4.1 스키마에서 컨텍스트를 사용하는 방법

다음과 같은 용도로 유니버스 스키마에서 컨텍스트를 사용할 수 있습니다.

- 루프를 해결합니다.
- 캐즘 트랩을 해결합니다.
- 팬 트랩을 해결하는 데 도움을 줍니다.
- 집계 인식을 사용하여 호환되지 않는 개체를 검색하는 데 도움을 줍니다.

5.4.1.1 컨텍스트를 사용하여 루프 해결

일반적으로 컨텍스트는 두 개의 쿼리 경로를 구분하여 하나는 특정 팩트 테이블에 대한 데이터를 반환하고 다른 하나는 다른 팩트 테이블에 대한 데이터를 반환하도록 하는 데 사용됩니다. 컨텍스트를 사용하면 여러 개의 팩트 테이블이 들어 있는 스키마에서 조인 경로를 지정할 수 있습니다. 이러한 스키마에서는 별칭을 사용하는 것이 적절하지 않습니다. 이와 같은 컨텍스트 사용은 루프 해결 [페이지 195] 단원에서 설명합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.4.1.2 컨텍스트를 사용하여 캐즘 트랩 및 팬 트랩 해결

컨텍스트는 잠재적인 캐즘 트랩을 해결하는 데도 사용됩니다. 캐즘 트랩은 다대일 조인 경로 두 개가 단일 테이블에 수렴될 때 발생합니다. 이 경우 단일 차원에 대해 여러 개의 행이 반환되어 필요 이상의 결과가 반환될 수 있습니다. 컨텍스트는 차원에 대해 정확한 수의 행이 반환되도록 쿼리를 분리할 수 있습니다. 컨텍스트를 별칭과 함께 사용하면 팬 트랩을 해결할 수도 있습니다. 이와 같은 컨텍스트 사용은 캐즘 트랩 해결 [페이지 220] 단원에서 설명합니다.

5.4.1.3 컨텍스트를 사용하여 **AggregateAwareness** 호환성 확인

컨텍스트를 사용하면 정의에서 @AggregateAware 함수를 사용하는 개체와 호환되지 않는 개체를 제외시킴으로써 집계를 인식하는 개체와 쿼리에 함께 사용되지 못하도록 할 수 있습니다.

5.4.2 컨텍스트 만들기

유니버스 디자인 도구에서 컨텍스트를 자동으로 검색하도록 할 수도 있고, 컨텍스트를 수동으로 만들 수도 있습니다.

컨텍스트를 사용하여 루프나 캐즘 트랩을 해결하는 경우 항상 유니버스 디자인 도구가 컨텍스트를 자동으로 검색하도록 해야 합니다. 그러나 다른 조인 경로 문제인 팬 트랩을 해결할 때는 컨텍스트를 수동으로 만들어야 할 수 있습니다.

루프 해결을 위한 컨텍스트 자동 검색은 루프 해결 [페이지 195] 단원에서 설명합니다.

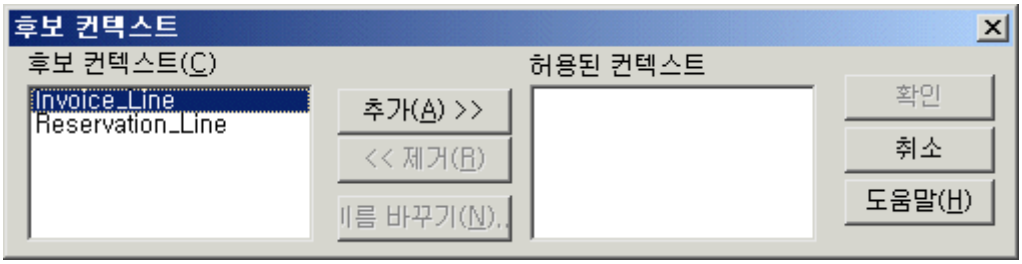
> **노트**
>
> 컨텍스트를 하나 이상 만들 때는 모든 조인이 하나 또는 여러 개의 컨텍스트에 포함되어 있어야 합니다. 컨텍스트에 포함되지 않은 조인으로 테이블이 연결되어 있으면 쿼리를 실행할 때 해당 조인이 제외됩니다.

다음 절차는 컨텍스트를 자동 및 수동으로 만드는 방법에 대해 설명합니다.

5.4.2.1 자동으로 컨텍스트 만들기

컨텍스트를 자동으로 만들려면

1. 도구 > 자동화된 검색 > 컨텍스트 검색을 선택합니다.
   후보 컨텍스트 상자가 나타납니다. 여기에는 스키마에 대한 후보 컨텍스트가 제안됩니다. 캐즘 트랩은 두 컨텍스트가 만나는 분기에서 발생하므로 후보 컨텍스트는 루프나 캐즘 트랩 중 하나를 해결하는 데 필요할 수 있습니다.

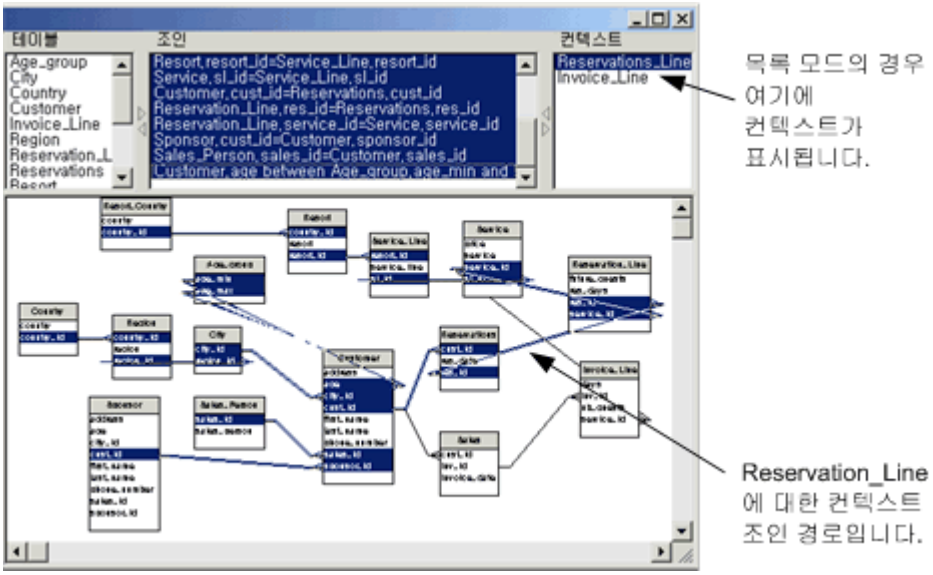


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 후보 컨텍스트 목록에서 컨텍스트를 클릭한 다음 추가 단추를 클릭합니다.
3. 목록에 있는 후보 컨텍스트 각각에 대해 2 단계를 반복합니다.

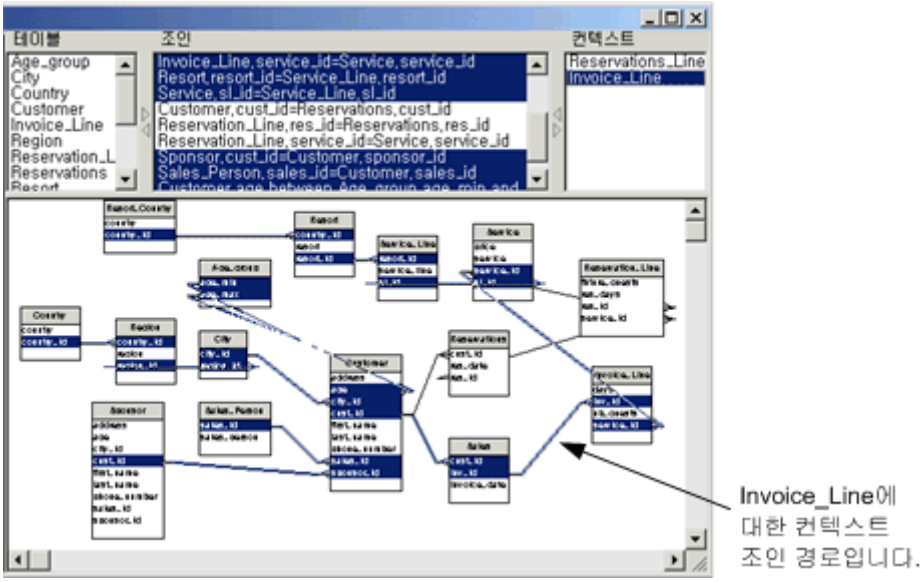
> **노트**
>
> 허용된 컨텍스트 목록에 후보 컨텍스트를 추가한 후에는 컨텍스트를 클릭하고 이름 바꾸기 단추를 클릭하여 컨텍스트의 이름을 변경할 수 있습니다. 그러면 편집 상자가 나타납니다. 이 상자에 새 이름을 입력한 후 확인을 클릭합니다.

4. 확인을 클릭합니다.
   목록 모드(보기 > 목록 모드)를 활성화하면 컨텍스트 창에 컨텍스트가 나열됩니다. 다음은 Invoice_Line 의 컨텍스트입니다.



5. 다음은 Invoice_Line 에 대한 컨텍스트입니다.

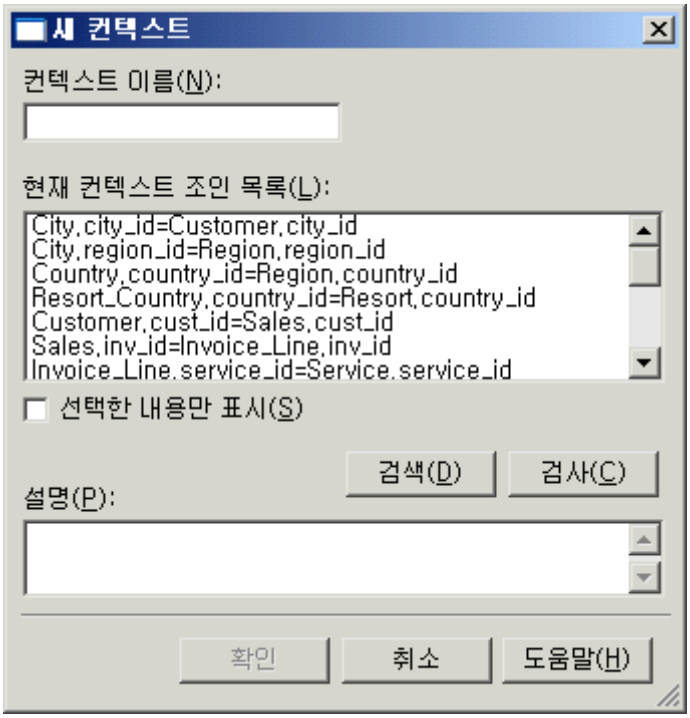
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



5.4.2.2 수동으로 컨텍스트 만들기

컨텍스트를 수동으로 만들려면

1. 삽입 > 컨텍스트를 선택합니다.
   또는
   컨텍스트 삽입 단추를 클릭합니다.
   새 컨텍스트 상자가 나타납니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 컨텍스트 이름 텍스트 상자에 컨텍스트 이름을 입력합니다.
3. 컨텍스트를 정의하는 모든 조인을 현재 컨텍스트 조인 목록에서 선택합니다.
   컨텍스트를 만들 때 다음과 같은 옵션을 사용할 수 있습니다.
4. 검색 단추를 클릭하여 제안된 컨텍스트를 구성하는 조인과 해당 컨텍스트 이름을 표시합니다.
5. 선택한 항목만 표시 확인란을 선택하여 선택한 조인만 표시합니다.
6. 검사 단추를 클릭합니다.
   유니버스 디자인 도구가 선택한 조인에서 루프가 있는지 검사합니다.
7. 컨텍스트에서 반환하는 데이터에 대한 설명을 입력합니다. 이 도움말 텍스트는 Web Intelligence 사용자가 컨텍스트 경로를 사용하는 쿼리를 실행했을 때 표시됩니다. 이 텍스트는 최종 사용자에게 도움이 되는 내용이어야 합니다.
8. 확인을 클릭합니다.
   컨텍스트가 만들어졌습니다.

5.4.3 컨텍스트 편집

컨텍스트 편집기를 사용하면 다음과 같이 컨텍스트의 속성을 수정할 수 있습니다.

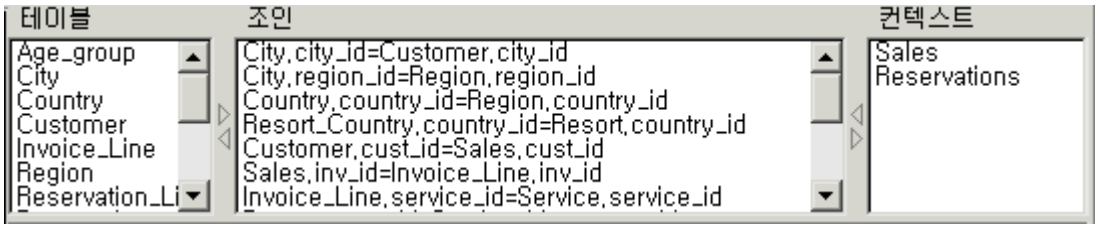
- 이름
- 컨텍스트에 포함된 조인
- 설명

컨텍스트에 해결되지 않은 루프가 있는지 검사할 수도 있습니다.

5.4.3.1 컨텍스트 속성 편집

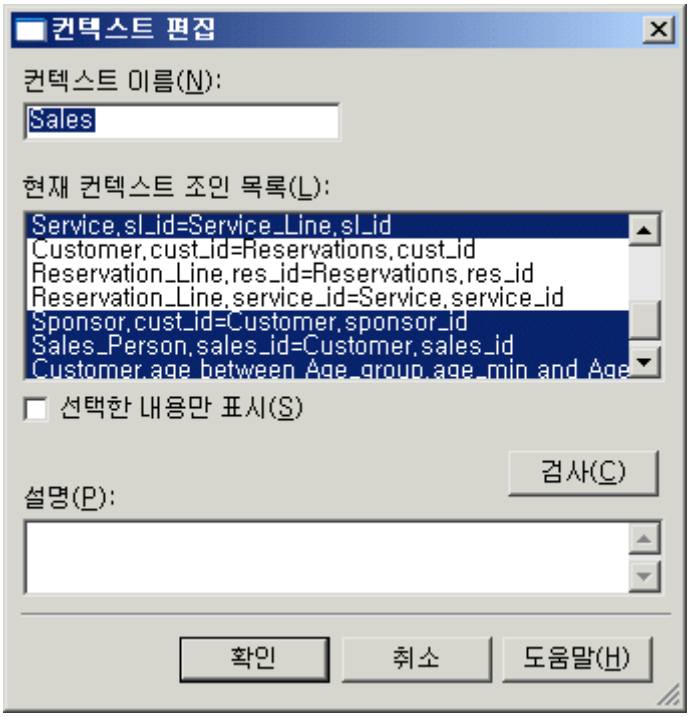
컨텍스트 속성을 편집하려면

1. 보기 > 목록 모드를 선택합니다.
   구조 창 위쪽에 목록 창이 나타납니다. 이 창에는 구조 창 내의 모든 테이블, 조인 및 컨텍스트에 대한 목록 상자가 들어 있습니다.



2. 컨텍스트 목록 창에서 컨텍스트 이름을 두 번 클릭합니다.
   컨텍스트 편집 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



3. 컨텍스트 이름을 변경하려면 컨텍스트 이름 상자에 새 이름을 입력합니다.
4. 강조 표시된 조인을 클릭하여 컨텍스트에서 제거합니다.
   또는
   강조 표시되지 않은 조인을 클릭하여 컨텍스트에 추가합니다.
5. 컨텍스트에 대한 설명을 입력합니다.
6. 확인을 클릭합니다.
   컨텍스트에 수정 사항이 표시됩니다.

5.4.4 컨텍스트 삭제

목록 창의 컨텍스트 목록에서 컨텍스트를 언제든지 삭제할 수 있습니다. 컨텍스트 내에서 테이블이나 조인을 추가하거나 삭제하는 경우에는 해당 컨텍스트를 삭제한 후에 테이블이나 조인을 수정해야 합니다.

수정 작업을 완료한 후에는 컨텍스트를 수동으로 다시 만들거나(컨텍스트로 캐즘 트랩을 해결해야 하는 경우) 컨텍스트 검색을 사용하여 새 컨텍스트를 자동으로 검색(컨텍스트로 루프를 해결해야 하는 경우)할 수 있습니다. 컨텍스트 검색에 대한 내용은 컨텍스트 검색 및 만들기 [페이지 207] 단원을 참조하십시오.

5.4.4.1 컨텍스트 목록에서 컨텍스트 삭제

컨텍스트 목록에서 컨텍스트를 삭제하려면

1. 보기 > 목록 모드를 선택하여 목록 모드를 활성화합니다.
2. 컨텍스트 목록 상자에서 컨텍스트 이름을 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 지우기를 선택합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

또는
컨텍스트 목록 상자에서 컨텍스트 이름을 클릭한 다음 편집 > 지우기를 클릭합니다.
컨텍스트가 목록에서 제거됩니다.

5.4.5 컨텍스트 업데이트

유니버스 구조가 변경되어도 컨텍스트는 자동으로 업데이트되지 않습니다. 구조에서 테이블 및 조인을 추가하거나 제거하면 모든 컨텍스트를 업데이트해야 합니다.

구조를 조금만 변경한 경우에는 컨텍스트 편집 상자 또는 목록 창을 사용하여 각 컨텍스트에 포함된 조인을 수동으로 업데이트할 수 있습니다. 그러나 유니버스 구조에 대한 변경 사항이 많은 경우에는 현재 컨텍스트를 삭제한 후 다시 만들어야 합니다.

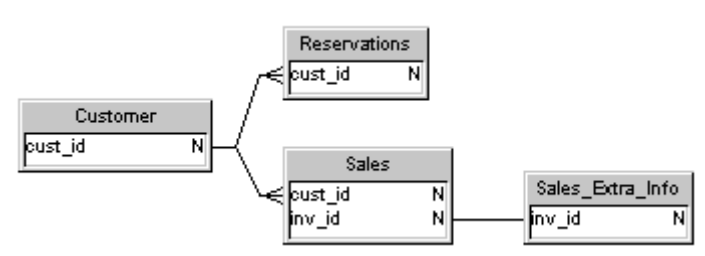
5.4.6 컨텍스트 검색을 수행하지 못하게 하는 조인 경로

조인 경로 끝에 일대일 카디널리티가 있으면 유니버스 디자인 도구에서 컨텍스트 검색 기능을 사용하여 컨텍스트를 검색하지 못할 수 있습니다. 조인 경로의 끝에 있는 테이블의 카디널리티를 일대다로 변경하면 이 문제가 해결됩니다.

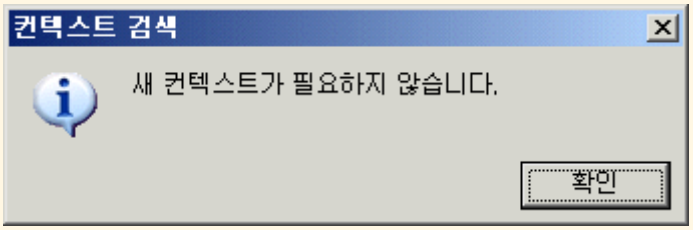
예

일대일 카디널리티로 인해 컨텍스트를 검색할 수 없는 경우

아래 스키마에는 각 판매에 대한 정보를 포함하는 Sales_Extra_Info 테이블이 있습니다. 이 테이블은 Sales 테이블에 일대일 조인으로 조인되어 있습니다.



조인 경로를 살펴보면 이 스키마에는 두 개의 컨텍스트, 즉 Reservations 컨텍스트와 Sales 컨텍스트가 있습니다. 그러나 도구 > 자동화된 검색 >컨텍스트 검색을 선택하여 이런 조인 경로 유형에 대해 컨텍스트를 자동으로 검색하면 다음과 같은 메시지가 표시됩니다.

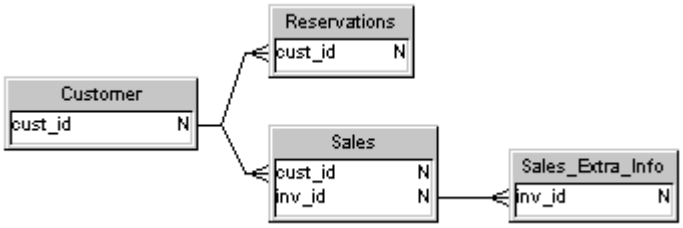


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

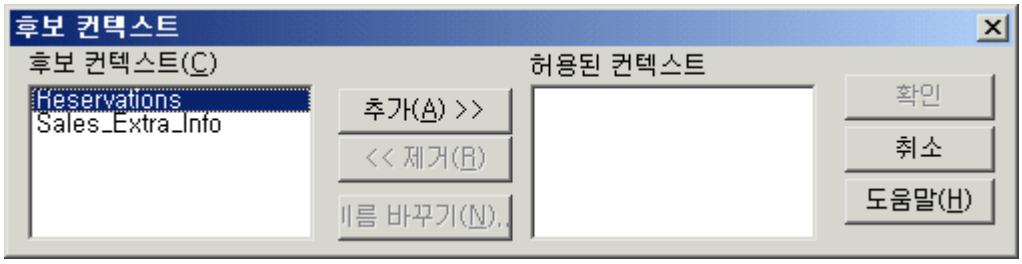
유니버스 디자인 도구에서는 컨텍스트를 검색할 때 조인 경로 끝에 있는 일대일 조인을 고려하지 않으므로 컨텍스트가 두 개 있다고 인식하지 않습니다.

5.4.6.1 카디널리티를 변경하여 컨텍스트 검색

이 문제를 해결하려면 Sale_Extra_Info 와 Sales 를 연결하는 조인의 카디널리티를 일대다로 설정하면 됩니다. 여기서는 조인 경로 끝에 일대일 조인이 있으면 안된다는 것이 중요하므로 카디널리티를 다대일로 변경해도 상관없습니다. 아래 스키마에는 이제 조인 경로 끝에 일대다 조인이 있습니다.



컨텍스트 검색을 실행하면 다음과 같이 두 컨텍스트가 검색됩니다.



5.4.7 컨텍스트가 쿼리에 미치는 영향

Web Intelligence 사용자가 스키마 구조에 정의된 개체를 사용할 수 있도록 허용한 방법에 따라 컨텍스트에서 다음과 같은 세 가지 쿼리 유형이 실행될 수 있습니다.

- 정확하지 않은 쿼리
- 유추된 쿼리
- 호환되지 않는 쿼리

Web Intelligence 에서 이러한 유형의 쿼리를 실행하면 컨텍스트에서 생성된 SQL 을 테스트할 수 있습니다. 이들 쿼리에서 오류가 발생하거나 올바르지 않은 데이터가 반환되면 관련된 조인 경로를 분석해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.4.7.1 정확하지 않은 쿼리

두 개의 쿼리 경로 중 하나를 선택하라는 메시지가 최종 사용자에게 표시됩니다. 이는 쿼리에 포함된 개체를 함께 사용해도 정보가 충분하지 않아 특정 컨텍스트 또는 다른 컨텍스트를 결정할 수 없는 경우에 발생합니다.

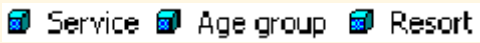
쿼리가 정확하지 않은 경우 Web Intelligence 에는 두 개 컨텍스트 중 하나를 선택하라는 대화 상자가 표시됩니다. 사용자가 컨텍스트를 선택하면 해당 테이블과 조인이 SQL 쿼리에 삽입됩니다.

예

정확하지 않은 쿼리 실행

Web Intelligence 사용자가 다음 쿼리를 실행합니다.

각 휴양지에 대해 각 연령층의 방문객이 사용하는 서비스를 검색합니다.



쿼리가 실행될 때 컨텍스트(이 경우 Reservations 또는 Sales 컨텍스트)를 선택하라는 메시지가 대화 상자에 표시됩니다.

사용자는 예약된 서비스에 대한 정보를 연령층별로 볼지 아니면 지불된 서비스에 대한 정보를 연령층별로 볼지 선택해야 합니다. 사용자가 Reservations 컨텍스트를 선택하면 다음 SQL 이 생성됩니다.

```
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  )
```

다른 컨텍스트(Sales)에서 참조하는 조인은 SQL 에 표시되지 않습니다.

5.4.7.2 유추된 쿼리

최종 사용자에게 컨텍스트를 선택하라는 메시지가 표시되지 않고 Web Intelligence 쿼리가 실행됩니다. 이 경우에는 올바른 컨텍스트를 유추할 수 있을 정도로 쿼리의 정보가 충분합니다. 예를 들어, 사용자가 다음 쿼리를 실행합니다.

사용할 수 있는 각 서비스에 대해 향후 손님 수를 연령층별로 검색합니다.

쿼리를 실행하면 컨텍스트를 선택하라는 메시지가 표시되지 않고 데이터가 반환됩니다. `Future Guests` 개체는 Reservations 컨텍스트에 속한 Reservation_Line 테이블의 합계입니다. Web Intelligence 에서는 Reservation 컨텍스트가 쿼리에 사용할 컨텍스트임을 유추합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.4.7.3 호환되지 않는 쿼리

서로 다른 두 개의 컨텍스트에서 가져온 개체를 쿼리에 조합합니다. 두 개의 Select 문이 동기화되어 반환 데이터가 개별 테이블 두 개에 표시됩니다.

예

호환되지 않는 쿼리 실행

Web Intelligence 사용자가 다음 쿼리를 실행합니다.

회사별로 총 손님 수를 연령층별로 찾고 예약한 달도 검색합니다.

쿼리를 실행하면 Web Intelligence 에서는 Sales 및 Reservations 컨텍스트를 모두 사용해야 한다고 유추하므로 메시지가 표시되지 않습니다. 두 컨텍스트에 대한 Select 문이 다음과 같이 동기화됩니다.

```
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
```

쿼리는 다음과 같이 두 부분으로 나뉩니다.

- 연령층 및 향후 손님
- 예약 달

두 개 쿼리의 결과를 검색할 때 Web Intelligence 에서는 **연령층**을 사용하여 결과를 조합합니다. 다음과 같이 하나의 보고서에서 테이블 두 개로 결과를 표시합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

18-30

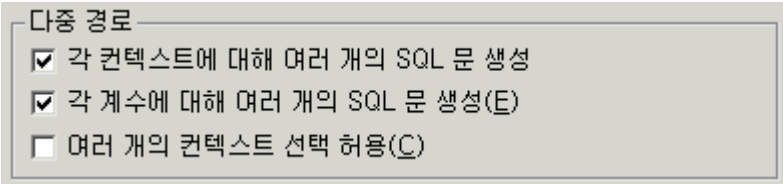
Number of guests	Reservation Month
451.00	Apr
	Aug
	Feb
	Jan
	Jun
	May
	Nov
	Oct
	Sep

호환되지 않는 쿼리를 Web Intelligence 에서 실행하려면 유니버스 디자인 도구에서 각 컨텍스트 옵션에 대해 여러 SQL 문을 선택해야 합니다. 이 작업은 다음 단원에서 설명합니다.

5.4.7.4 각 컨텍스트에 여러 SQL 문 사용 선택

각 컨텍스트에 여러 SQL 문 사용 옵션을 선택하려면

1. 파일 > 매개 변수를 선택합니다.
   유니버스 매개 변수 대화 상자가 나타납니다.
2. SQL 탭을 클릭합니다.
   SQL 페이지가 나타납니다.
3. 각 컨텍스트에 여러 SQL 문 사용 확인란을 선택합니다.



4. 확인을 클릭합니다.

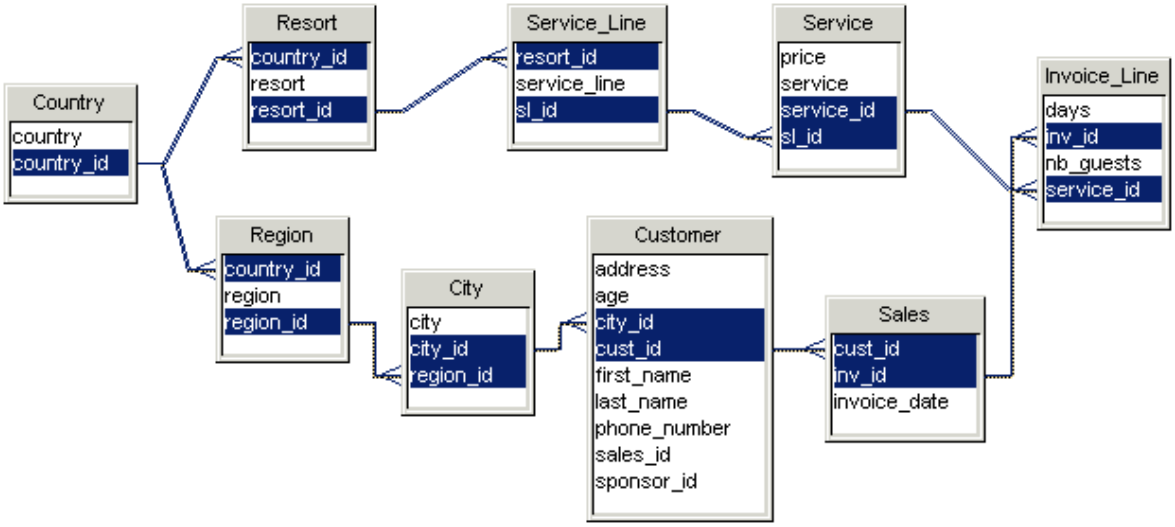
5.5 루프 해결

관계형 데이터베이스 스키마에서 너무 적은 수의 행을 반환하는 일반적인 조인 경로 유형을 루프라고 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.5.1 루프란?

루프는 스키마에 포함된 테이블 집합 사이의 닫힌 경로를 정의하는 조인 집합입니다. 루프는 조회 테이블 사이에 여러 개의 조인 경로가 만들어졌을 때 발생합니다. 다음은 루프 예제입니다.



이 스키마에는 두 개의 정보 집합이 연결되어 있습니다.

표 121:

테이블	연결된 정보
Resort	사용할 수 있는 서비스 목록, 각 서비스 목록에 포함된 서비스, 각 서비스에 대한 송장 정보 및 휴양지가 위치한 국가
Customer	고객이 거주하는 도시, 지역 및 국가, 고객의 판매, 각 판매에 대한 송장 정보

이들 두 정보 집합은 루프를 구성하는 일반 조인 경로로 연결되어 있습니다. Country 조회 테이블은 휴양지가 위치한 국가 또는 고객이 거주하는 국가가 될 수 있습니다.

5.5.1.1 루프가 데이터베이스에서는 발생하지 않고 유니버스 스키마에서 발생하는 이유

데이터베이스에서는 특정 사용자 요구 사항을 충족하기 위해 테이블 사이에 여러 개의 경로를 아무런 문제 없이 구현할 수 있습니다. 각 경로를 쿼리에 개별적으로 사용하면 고유한 결과 집합이 반환됩니다.

단, 유니버스 디자인 도구에서 디자인한 스키마에는 경우에 따라 둘 이상의 경로가 포함된 쿼리(관계형 데이터베이스에서는 처리할 수 없음)를 사용해야 하므로 반환된 결과가 올바르지 않을 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

이 경우, 각 경로에 대한 결과의 교집합만 반환되므로 예상했던 것보다 적은 수의 행이 반환됩니다. 또한 결과를 검토하여 문제점을 정확하게 파악하기 힘듭니다.

5.5.2 루프가 쿼리에 미치는 영향

위의 구조에 기반하여 유니버스를 만든 경우 루프에 포함된 테이블에 대해 실행하는 모든 쿼리는 휴양지의 국가 값과 고객 국적의 국가 값이 동일한 행만 반환합니다. 공유 조회 테이블인 Country 에 대한 이와 같은 이중 제한으로 인해 예상보다 적은 수의 행만 반환됩니다.

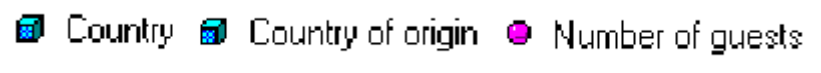
예

잘못된 결과를 반환하는 루프

위에서 언급한 루프가 포함된 스키마를 사용하여 다음 개체를 만듭니다.

Web Intelligence 에서 다음 쿼리를 실행합니다.

휴양지 국가 각각에 대해 각 휴양지에 숙박한 손님 수를 손님의 국적별로 검색합니다.



다음과 같이 결과가 반환되어야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

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

프랑스와 미국에 있는 휴양지 각각에 숙박하고 있는 독일, 일본 및 미국 방문객의 수가 표시됩니다.

그러나 루프가 포함된 유니버스를 사용하여 이 쿼리를 실행하면 다음과 같은 결과가 반환됩니다.

Country	Country of origin	Number of guests
US	US	431.00

이 결과에 따르면 미국에 있는 휴양지에는 미국 방문객만 숙박한 것으로 나타납니다. 그리고 다른 국적의 방문객은 없습니다.

5.5.2.1 루프가 쿼리에 미치는 영향

구조 내의 조인은 쿼리의 유추된 SQL 에서 사용하는 Where 절을 만드는 데 사용됩니다. 조인은 쿼리에서 반환하는 데이터를 제한하기 위한 목적으로 사용됩니다. 그러나 루프가 있으면 조인으로 인해 예상보다 더 많은 제한이 적용되며, 따라서 반환되는 데이터도 올바르지 않습니다.

루프로 인해 다음과 같이 Where 절이 만들어집니다.

```
 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'  )
```

아래의 두 조인 모두 Country 테이블에 제한을 적용합니다.

- Country.country_id=Resort.country_id
- Country.country_id=Region.country_id

Country 는 두 가지 목적으로 사용됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 휴양지 국가 조회
- 고객의 국적 조회

이러한 제한 때문에 휴양지 국가와 고객의 국적이 동일한 경우에만 데이터가 반환됩니다. 따라서 결과 보고서에는 미국 휴양지에 방문한 미국인 방문객 수만 표시됩니다.

루프의 특성에 따라 유니버스 디자인 도구에서는 두 가지 방법으로 루프를 해결할 수 있습니다. 하나는 별칭을 사용하여 조인 경로를 중단하는 방법이고, 다른 하나는 컨텍스트를 사용하여 두 개의 조인 경로를 구분함으로써 쿼리에서 둘 중 하나만 사용할 수 있도록 하는 방법입니다.

5.5.2.2 별칭을 사용하여 루프를 중단하는 방법

별칭은 같은 쿼리에서 동일한 테이블을 서로 다른 목적으로 두 번 사용하여 루프를 중단합니다. 별칭은 기본 테이블과 동일하며 이름만 다릅니다. 별칭의 데이터는 원본 테이블의 데이터와 동일하지만 이름이 다르기 때문에 SQL 에서는 두 개의 서로 다른 테이블을 사용하는 것으로 "착각"하여 인식합니다.

노트

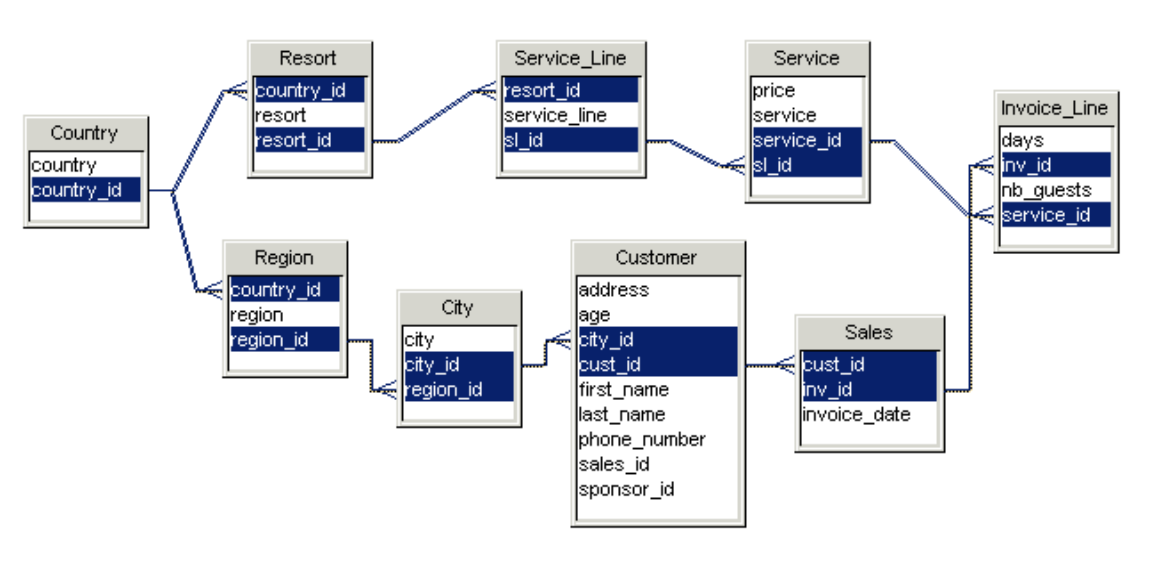
앞의 예제에서는 별칭 테이블을 하나만 만들면 루프를 성공적으로 해결할 수 있습니다. Region 조인은 원본 Country 테이블을 사용하고 Showroom 조인은 테이블의 별칭을 사용합니다. 그러나 원본 테이블의 각 조인에 대해 별칭 테이블을 하나씩 만들 수도 있습니다. 일부 관계형 데이터베이스 시스템에서는 두 번째 방법을 사용해야 할 수 있습니다.

예

별칭을 사용하여 루프 중단

아래 스키마는 앞 단원에서 사용한 것으로, 루프가 포함되어 있습니다. 이 스키마에 사용된 조인 경로의 경우 Country 조회 테이블은 두 개의 조인에서 "일(one)"에 해당되기 때문에 다음과 같이 두 가지 목적으로 사용될 수 있습니다.

- 휴양지 국가
- 고객의 국적

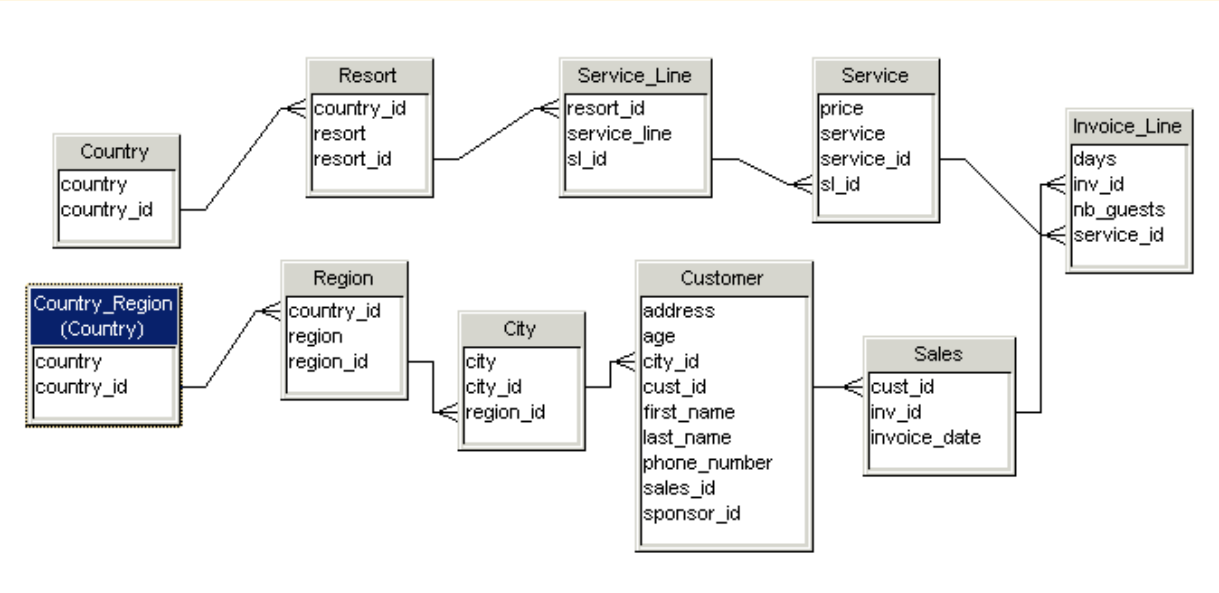


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

Country 의 별칭을 만들어 이름을 Country_Region 으로 바꿉니다. 그러면 기존의 조인이 다음과 같이 구분됩니다.

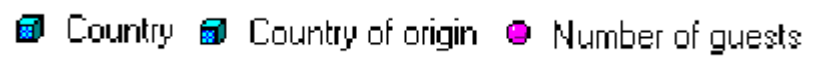
- Country 는 Resort 테이블에 조인되어 있습니다.
- Country_Region 은 Region 테이블에 조인됩니다.

이제 스키마가 다음과 같이 표시됩니다.



앞의 예제에서 너무 적은 수의 행을 반환했던 쿼리를 다시 실행합니다.

휴양지 국가 각각에 대해 각 휴양지에 숙박한 손님 수를 손님의 국적별로 검색합니다.



이 쿼리의 Where 절은 이제 다음과 같습니다.

```
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'  )
```

이제는 Country 테이블에 제한을 적용하는 조인과 Resort_Country 테이블에 제한을 적용하는 다른 조인이 있습니다. 즉, 루프가 중단되었습니다.

쿼리를 실행하면 다음 테이블이 반환됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

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 컨텍스트를 사용하여 루프를 해결하는 방법

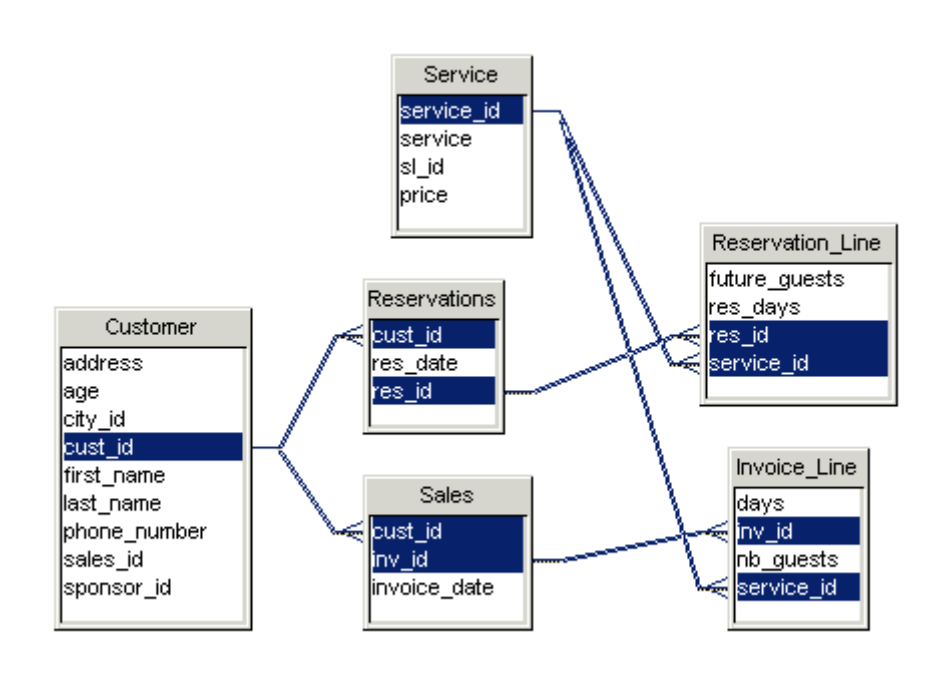
컨텍스트는 루프에 포함된 테이블 사이에 사용할 특정 경로를 지정하는 조인 집합을 정의하는 방법으로 루프를 해결합니다. 컨텍스트는 같은 SQL 쿼리 안에 경로가 다른 조인이 포함되지 않도록 합니다.

일반적으로 컨텍스트는 조회 테이블을 공유하는 팩트 테이블("여러 개의 별모양")이 여러 개 포함된 스키마에서 사용합니다.

예

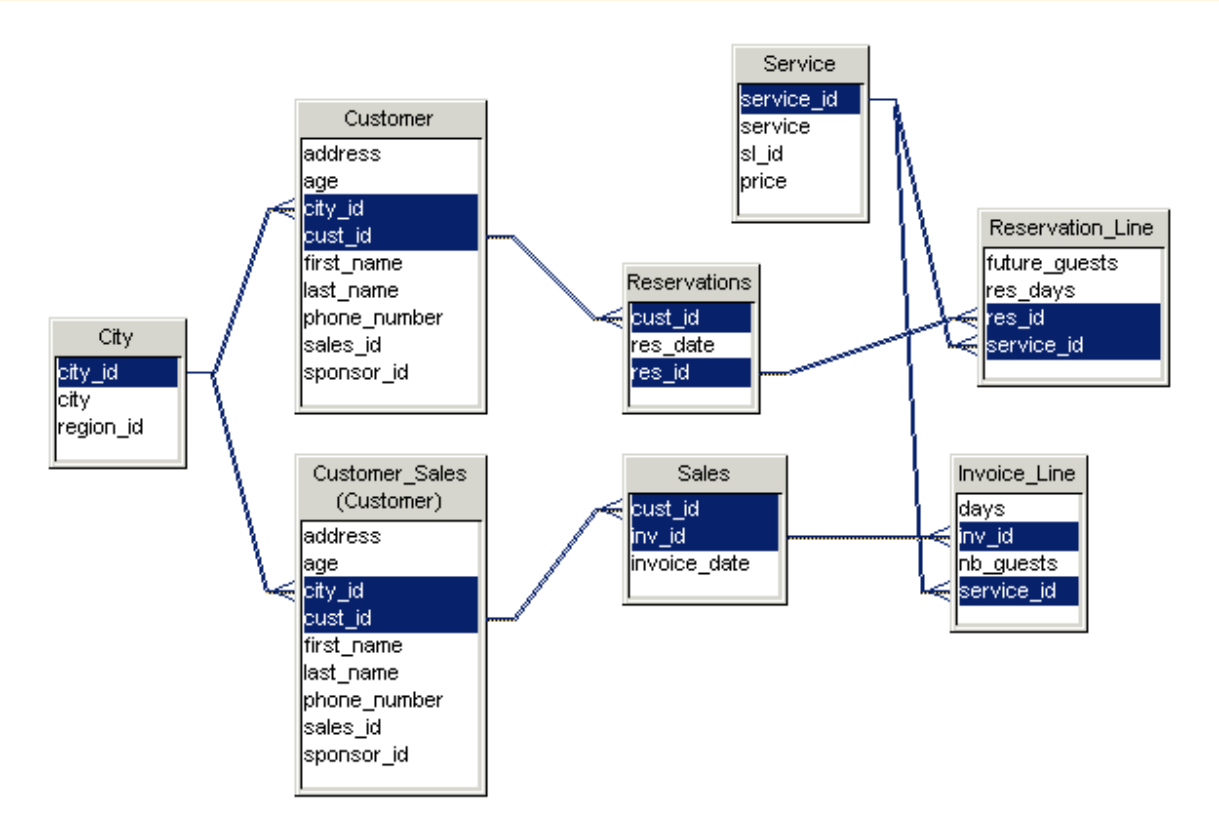
컨텍스트를 사용하여 루프 해결

아래 스키마에는 판매와 예약에 대한 통계 정보가 포함되어 있습니다. 각 트랜잭션 유형과 관련된 통계는 Sales 와 Reservations 라는 팩트 테이블에 저장되어 있습니다. 이 스키마에서는 서비스 정보를 찾을 때 판매 경로나 예약 경로 중 하나를 조인 경로로 사용할 수 있으므로 루프가 형성됩니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

Customer 의 별칭을 만드는 방법으로 Customer 와 Reservation 사이의 조인을 만들고 Customer_Sales 와 Sales 사이의 조인을 만들어 루프를 중단하더라도 스키마에 City 테이블을 추가하면 다음과 같이 루프 문제가 다시 발생합니다.



이 경우, 스키마에 테이블을 새로 추가할 때마다 별칭을 만들어야 합니다. 이 방법을 사용하면 스키마를 유지하기 어렵고 유니버스의 각 테이블을 사용하는 유사한 개체 수를 계속해서 늘려야 합니다.

이 루프를 해결할 수 있는 유일한 방법은 루프를 그대로 둔 상태에서 스키마 내에서 사용할 특정 경로를 지정하는 컨텍스트를 만드는 것입니다. 이렇게 하면 쿼리는 트랜잭션 중 하나에 대한 질문(예: 판매 또는 예약의 관점에서 고객 정보가 필요합니까?)에 응답합니다.

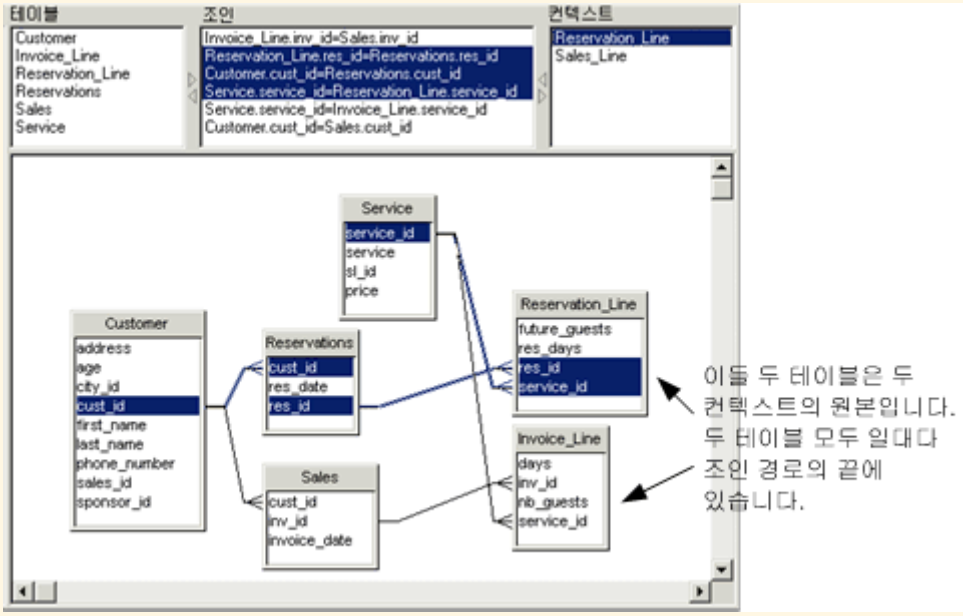
이 예제에서는 Customer 테이블에서 Service 테이블 사이에 서로 다른 두 개의 경로를 사용할 수 있습니다.

표 122:

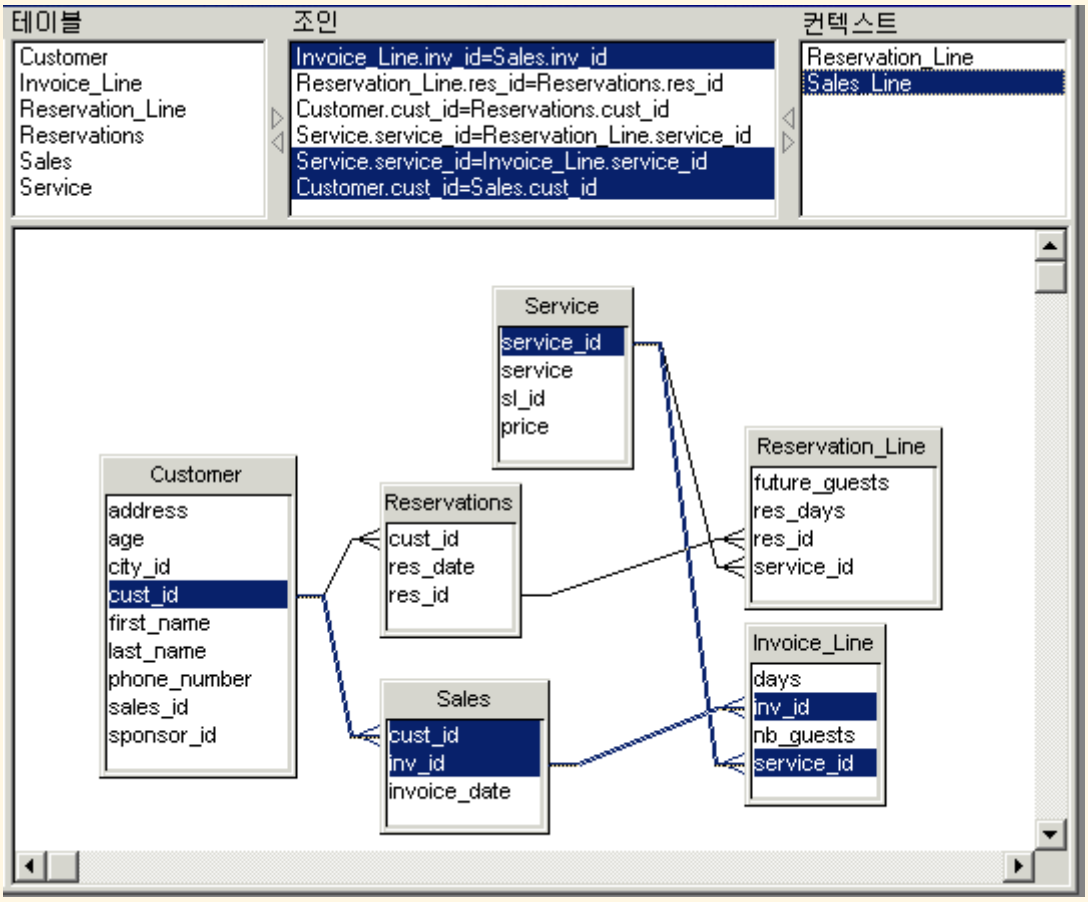
경로	검색되는 컨텍스트
Reservations 및 Reservation_Line	Reservation_Line
Sales 및 Invoice_Line	Sales_Line

Reservation_Line 컨텍스트는 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



Sales_Line 컨텍스트는 다음과 같습니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

그런 다음 서로 다른 컨텍스트에 포함된 테이블에 기반하여 서로 다른 개체 집합을 만듭니다. 그러면 사용자는 선택한 개체에 따라 Reservation 쿼리나 Sales 쿼리 중 하나를 실행할 수 있습니다.

5.5.3 시각적으로 루프 식별

다음과 같은 지침은 루프를 해결할 때 별칭 또는 컨텍스트 중 어느 것을 사용할지 결정하기 위해 스키마를 분석하는 데 도움을 줍니다. 이 지침은 스키마를 이해하는 데 도움을 주지만 실제로 루프를 식별하고 해결할 때는 별칭 검색 및 컨텍스트 검색 기능을 사용하는 것이 좋습니다. 자세한 내용은 별칭 검색 및 만들기 [페이지 206] 및 컨텍스트 검색 및 만들기 [페이지 207] 단원을 참조하십시오.

표 123:

루프 구성	루프 해결 방법
조회 테이블 하나	별칭
조인의 "일(one)"쪽에만 해당하는 조회 테이블	별칭
둘 이상의 팩트 테이블	컨텍스트

5.5.4 자동으로 루프 식별 및 해결

유니버스 디자인 도구를 사용하여 루프를 자동으로 검색하고 루프를 해결하기 위해 스키마에 삽입할 수 있는 후보 별칭과 컨텍스트를 제안할 수 있습니다.

5.5.4.1 루프를 검색하기 전에 카디널리티 설정

자동 루프 검색 및 해결 기능을 사용하려면 먼저 스키마의 모든 조인에 대해 카디널리티를 설정해야 합니다.

카디널리티를 수동으로 정의하거나, 자동 루틴을 사용할 때 유니버스 디자인 도구에서 제안하는 각 카디널리티의 유효성을 수동으로 검사하는 것이 좋습니다.

다음과 같은 두 가지 방법으로 카디널리티를 설정할 수 있습니다.

- 수동으로 설정합니다. 자세한 내용은 카디널리티 사용 [페이지 162] 단원을 참조하십시오.
- 카디널리티 검색 기능을 사용합니다. 자세한 내용은 카디널리티 사용 [페이지 162] 단원을 참조하십시오.

5.5.5 루프를 검색하고 해결하는 도구 기능

다음과 같은 유니버스 디자인 도구 기능을 사용하여 루프를 식별하고 해결할 수 있습니다.

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 124:

루프 식별 및 해결 방법	설명
별칭 검색	구조 내의 루프를 해결하기 위해 별칭을 만들 수 있는 테이블을 검색하여 각 테이블에 대해 후보 별칭을 제안합니다. 상자에서 별칭을 직접 삽입하고 이름을 바꿀 수 있습니다. 구현하는 모든 컨텍스트에 새로 만든 별칭이 포함되도록 하려면 컨텍스트 검색을 실행하기 전에 별칭 검색을 실행하는 것이 좋습니다. 팬 트랩을 해결하는 데 필요한 별칭은 검색되지 않습니다.
컨텍스트 검색	구조 내의 루프를 해결하는 데 사용할 수 있는 컨텍스트를 검색하여 후보 컨텍스트를 제안합니다. 상자에서 컨텍스트를 직접 구현하고 이름을 바꿀 수 있습니다. 구현하는 모든 컨텍스트에 새 별칭이 포함되도록 하려면 별칭 검색을 실행한 후에 컨텍스트 검색을 실행해야 합니다. 캐즘 트랩을 해결하는 데 필요한 컨텍스트는 검색되지 않을 때도 있습니다. 이런 경우에는 컨텍스트를 수동으로 식별해야 합니다.
루프 검색	구조 내의 루프를 검색하고 강조 표시한 후 각 루프를 해결하기 위해 삽입할 수 있는 별칭이나 컨텍스트를 제안합니다. 제안된 별칭이나 컨텍스트를 루프 검색 상자에서 직접 구현할 수 있습니다. 루프 검색 기능을 사용하면 스키마를 신속하게 검사하거나 루프를 시각화할 수 있습니다. 후보 별칭을 삽입하기 전에는 해당 별칭을 편집하거나 볼 수 없으므로 이 기능은 루프를 식별한 후 해결하는 데 사용하면 안 됩니다.

5.5.5.1 루프를 식별하고 해결하는 일반적인 방법

루프를 검색하고 해결하는 일반적인 절차는 아래에 나와 있습니다. 각 단계를 자세하게 설명하는 단원도 확인할 수 있습니다.

1. 모든 카디널리티가 설정되었는지 확인합니다.
   카디널리티 사용 [페이지 162] 단원을 참조하십시오.
2. 스키마의 루프를 해결하는 데 별칭이 필요한지 식별하기 위해 별칭 검색을 실행합니다.
   자세한 내용은 별칭 검색 및 만들기 [페이지 206] 단원을 참조하십시오.
3. 별칭 검색 기능을 통해 제안된 후보 별칭을 삽입합니다.
4. 스키마에서 별칭만으로는 해결할 수 없는 루프를 해결하는 데 컨텍스트가 필요한지 식별하기 위해 컨텍스트 검색을 실행합니다.
   자세한 내용은 컨텍스트 검색 및 만들기 [페이지 207] 단원을 참조하십시오.
5. 컨텍스트 검색 기능을 통해 제안된 후보 컨텍스트를 구현합니다.
6. 개체를 만들고 쿼리를 실행하여 루프가 해결되었는지 테스트합니다.
   개체 만들기 및 유니버스 구조 테스트에 대한 내용은 유니버스 만들기 [페이지 242]?장을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

> **노트**
>
> 루프를 해결할 대상 스키마의 테이블에 개체가 이미 정의되어 있는 경우에는 기본 테이블 대신 새 별칭을 사용하는 개체를 다시 정의해야 합니다.

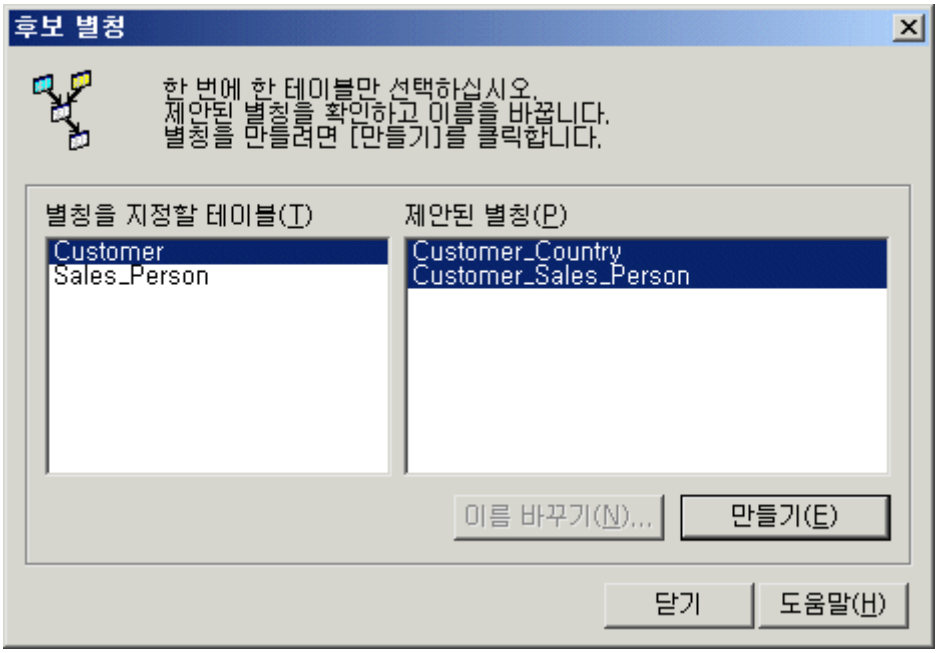
5.5.5.2 별칭 검색 및 만들기

별칭 검색을 사용하면 활성 유니버스 내에서 루프의 원인이 되는 테이블을 자동으로 검색하고 식별할 수 있습니다. 별칭 검색 기능은 사용자가 편집하고 스키마에 삽입할 수 있는 후보 테이블을 제안합니다.

> **노트**
>
> 스키마의 모든 테이블이 조인으로 연결되었는지, 그리고 모든 카디널리티가 설정되었는지 확인한 후에 별칭 검색을 사용해야 합니다.

별칭을 검색하고 만들려면

1. 도구 > 자동화된 검색 > 별칭 검색을 선택합니다.
   또는
   별칭 검색 단추를 클릭합니다.
   후보 별칭 대화 상자가 나타납니다. 왼쪽 창에는 별칭이 필요한 테이블이 나열됩니다. 오른쪽 창에는 루프를 중단하기 위해 삽입할 수 있는 제안 별칭이 나열됩니다.



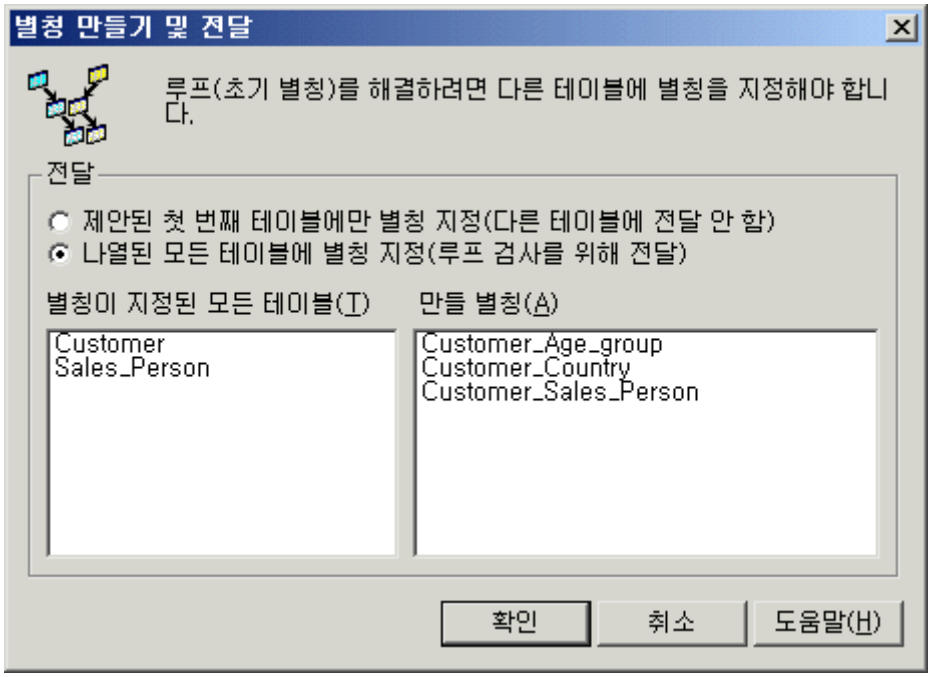
2. 왼쪽 창에서 테이블을 선택합니다.
   후보 별칭에 대해 제안된 이름이 오른쪽 창에 표시됩니다.
3. 제안된 별칭의 이름을 바꾸려면 이름 바꾸기를 클릭하고 이름 바꾸기 상자에 새 이름을 입력합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4. 만들기를 클릭합니다.
   별칭을 만들 것인지 확인하는 메시지 상자가 표시됩니다.
5. 확인을 클릭합니다.
   별칭이 구조 창에 나타납니다.
6. 나머지 테이블에 대해 2-5 단계를 반복합니다.
7. 닫기를 클릭합니다.

5.5.5.3 여러 개의 별칭 검색 및 만들기

경우에 따라서는 별칭을 만들 때 새 조인 경로에 사용할 별칭을 추가로 만들어야 할 수도 있습니다. 별칭 검색 기능을 사용할 때 유니버스 디자인 도구에서 별칭이 더 필요하다고 감지하면 사용자가 만들기 단추를 클릭할 때 다음 대화 상자가 나타납니다.



이 경우 두 가지 옵션 중에서 선택할 수 있습니다.

- 제안된 첫 번째 테이블의 별칭만 만듭니다.
- 나열된 모든 테이블의 별칭을 만듭니다.

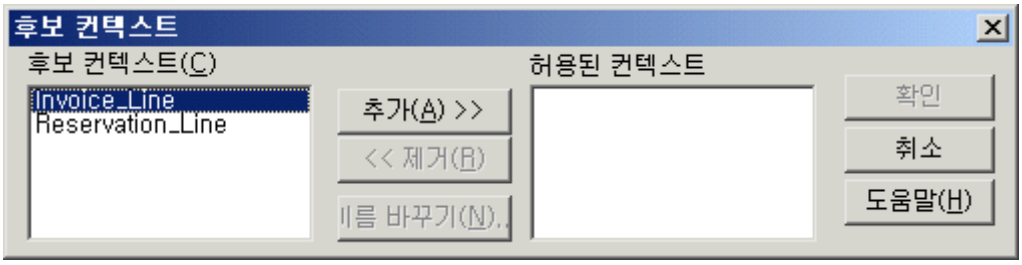
5.5.5.4 컨텍스트 검색 및 만들기

컨텍스트 검색을 사용하면 컨텍스트가 필요한지 자동으로 검색할 수 있습니다. 컨텍스트 검색 기능은 후보 컨텍스트도 제안합니다. 후보 컨텍스트를 구현하기 전에 편집할 수 있습니다.

컨텍스트를 검색하고 만들려면

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1. 도구 > 자동화된 검색 > 컨텍스트 검색을 선택합니다.
   또는
   컨텍스트 검색 단추를 클릭합니다.
   후보 컨텍스트 대화 상자가 나타납니다. 제안된 컨텍스트는 왼쪽 창에 표시됩니다.



2. 컨텍스트 이름을 클릭합니다.
   후보 컨텍스트에 포함된 테이블이 스키마에서 강조 표시됩니다.
3. 추가 단추를 클릭합니다.
   선택한 컨텍스트 창에 컨텍스트 이름이 표시됩니다. 오른쪽 창에서 컨텍스트를 선택한 다음 제거 단추를 클릭하여 컨텍스트를 제거할 수 있습니다.
4. 필요한 경우, 3-4 단계를 반복하여 다른 컨텍스트를 추가합니다.
5. 컨텍스트 이름을 바꾸려면 오른쪽 창에서 컨텍스트를 선택한 다음 이름 바꾸기 단추를 클릭합니다.
   컨텍스트 이름 바꾸기 대화 상자가 나타납니다. 새 이름을 입력합니다.
6. 확인 단추를 클릭합니다.
   유니버스 창의 컨텍스트 상자에 컨텍스트가 나열됩니다.

노트

사용자가 쉽게 알 수 없는 루프가 유니버스에 포함되어 있는 경우에는 해당 루프를 해결하는 컨텍스트 이름을 사용자가 이해하기 쉬운 이름으로 지정해야 합니다. 즉, 컨텍스트가 나타내는 정보 경로를 Web Intelligence 사용자가 쉽게 이해할 수 있어야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.5.5.5 자동으로 루프 검색

루프 검색을 사용하면 유니버스에서 루프를 검색할 수 있습니다. 이 기능은 스키마에서 루프를 자동으로 검사하고 해당 루프를 해결할 수 있는 별칭이나 컨텍스트를 제안합니다.

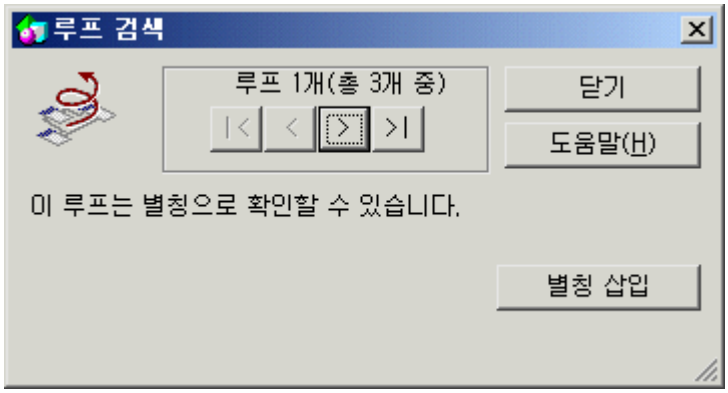
루프 검색 기능은 스키마에서 루프를 신속하게 검사하는 데 유용합니다. 또한 이 기능을 사용하면 검색된 루프를 해결할 수 있는 별칭과 컨텍스트가 제안됩니다. 그러나 별칭 검색 및 컨텍스트 검색을 통해 루프를 해결할 때보다 작성된 별칭과 컨텍스트의 순서를 제어하는 데 제한을 받습니다.

루프 해결에 권장되는 절차는 루프를 식별하고 해결하는 일반적인 방법 [페이지 205] 단원에서 설명합니다.

> **노트**
>
> 무결성 검사를 사용하여 조인, 카디널리티, 루프 등 유니버스 구조의 오류를 자동으로 검사할 수도 있습니다. 무결성 검사를 사용하면 발견된 오류에 대한 해결 방법이 제시됩니다. 자세한 내용은 수동으로 유니버스 무결성 검사 [페이지 237] 단원을 참조하십시오.

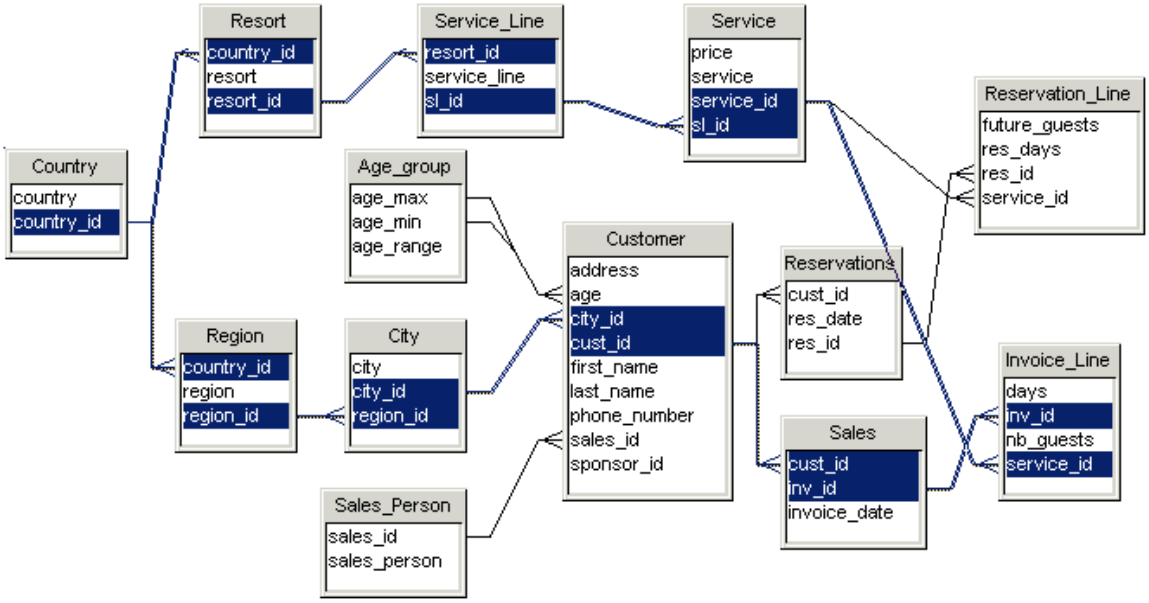
스키마에서 루프를 검색하려면

1. 스키마의 모든 조인에 대해 카디널리티를 설정했는지 확인합니다.
2. 도구 > 자동화된 검색 > 루프 검색을 선택합니다.
   또는
   루프 검색 단추를 클릭합니다.
   루프 검색 상자가 나타납니다. 여기에는 검색된 루프 수와 제안된 해결 방법이 표시됩니다.



   루프를 구성하는 조인 경로가 구조 창에서 다음과 같이 즉시 강조 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



3. 앞으로 단추를 클릭하여 다음 루프와 제안된 해결 방법을 표시합니다. 유니버스 디자인 도구에서 검색하는 각 루프에 대해 구조 창에서 해당 조인 경로가 강조 표시됩니다.
4. 닫기를 클릭합니다.

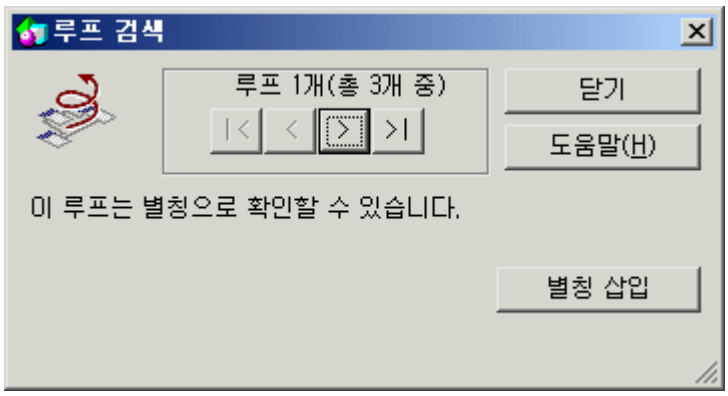
5.5.5.6 자동으로 별칭 및 컨텍스트 만들기

루프 검색 기능을 실행하면 유니버스 디자인 도구에서는 루프를 해결할 후보 별칭 또는 컨텍스트를 제안합니다. 루프 검색 상자에서 직접 후보 별칭을 삽입하거나 후보 컨텍스트를 구현할 수 있습니다.

루프 검색을 사용하여 별칭을 만들려면

1. 도구 > 자동화된 검색 > 루프 검색을 선택합니다.
   루프 검색 상자가 나타납니다. 여기에는 스키마에서 검색된 하나 이상의 루프가 표시되며 각 루프에 대해 후보 별칭이나 컨텍스트가 제안됩니다.
2. 검색된 루프에 대해 아래의 메시지가 표시될 때까지 앞으로 화살표 단추를 클릭합니다.
   이 루프를 별칭으로 해결할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

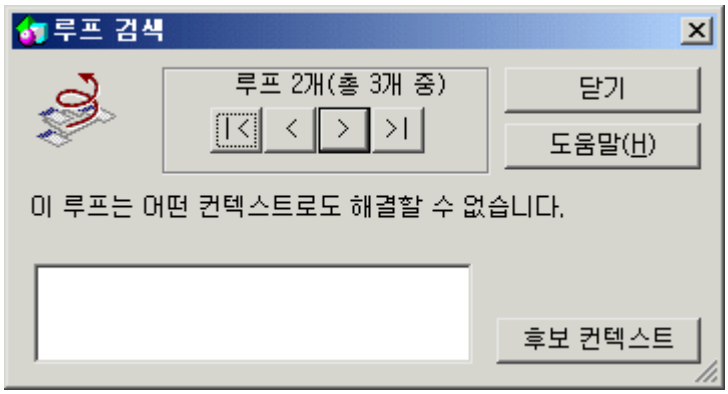


3. 별칭 삽입 단추를 클릭합니다.
   구조 창에 별칭이 자동으로 삽입됩니다. 이 별칭은 스키마에서 루프의 원인이 되는 테이블에 조인됩니다.

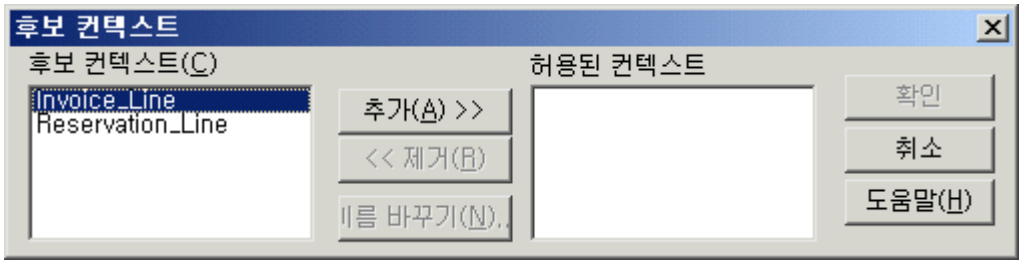
5.5.5.7 루프 검색을 사용하여 컨텍스트 만들기

루프 검색을 사용하여 컨텍스트를 만들려면

1. 도구 > 자동화된 검색 > 루프 검색을 선택합니다.
   루프 검색 상자가 나타납니다. 여기에는 스키마에서 검색된 하나 이상의 루프가 표시되며 각 루프에 대해 후보 별칭이나 컨텍스트가 제안됩니다.
2. 검색된 루프에 대해 아래의 메시지가 표시될 때까지 앞으로 화살표 단추를 클릭합니다.
   이 루프는 기존 컨텍스트로 해결할 수 없습니다.

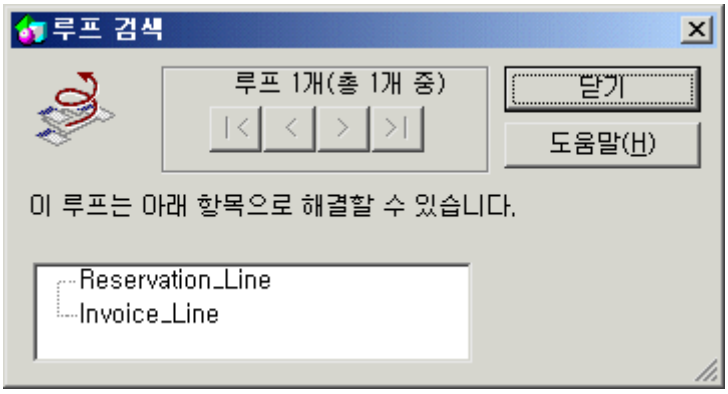


3. 후보 컨텍스트 단추를 클릭합니다.
   후보 컨텍스트 대화 상자가 나타납니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4. 컨텍스트 이름을 클릭합니다.
   후보 컨텍스트에 포함된 테이블이 스키마에서 강조 표시됩니다.
5. 추가 단추를 클릭합니다.
   선택한 컨텍스트 창에 컨텍스트 이름이 표시됩니다. 오른쪽 창에서 컨텍스트를 선택한 다음 제거 단추를 클릭하여 컨텍스트를 제거할 수 있습니다.
6. 필요한 경우, 3-4 단계를 반복하여 다른 컨텍스트를 추가합니다.
7. 확인을 클릭합니다.
   컨텍스트 확인 상자가 나타납니다.



8. 닫기를 클릭합니다.
   유니버스 창의 컨텍스트 상자에 컨텍스트가 나열됩니다.

5.5.6 루프 해결 예제

다음 예제는 아래의 작업을 수행하는 방법을 보여 줍니다.

- 공유된 조회 테이블로 인해 발생한 루프를 중단하는 별칭 만들기
- 공유된 조회 테이블로 인해 발생한 루프를 중단하는 별칭 만들기
- 별칭을 사용하여 루프를 중단할 수 없는 경우 식별
- 루프를 해결하는 컨텍스트 만들기
- 별칭과 컨텍스트를 함께 사용하여 루프 해결

이러한 예제에 사용된 스키마는 Beach 유니버스에 기반하지 않습니다. 아래의 예제에 사용된 스키마는 운송 업체를 기반으로 하며 이 장에서 Beach 유니버스를 사용하여 이미 제시된 특정 루프 해결 예제를 다른 관점에서 보여 줍니다.

5.5.6.1 공유된 조회 테이블로 인해 발생한 루프를 중단하는 별칭 만들기

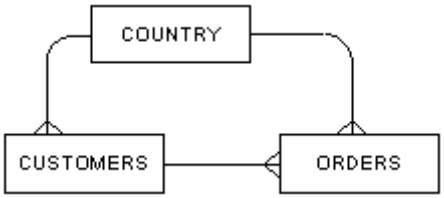
판매 데이터베이스에는 전세계적으로 고객에게 판매된 제품에 대한 정보가 저장되어 있습니다. 고객의 특징은 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 고객은 전세계 모든 곳에 거주할 수 있습니다.
- 고객은 회사의 제품을 주문할 수 있습니다.
- 고객은 주문한 제품을 원하는 모든 대상 국가에 배송하도록 요청할 수 있습니다.

예를 들어, 영국에 거주하는 고객이 자동차를 주문한 후 브라질로 배송하도록 요청할 수 있습니다.

이 데이터베이스 유형의 스키마는 다음과 같습니다.



이 스키마를 다음과 같이 해석할 수 있습니다.

- 각 고객은 한 국가에만 거주할 수 있습니다.
- 각 고객은 제품을 하나 이상 주문할 수 있습니다.
- 회사에서는 주문된 각 제품을 대상 국가로 배송하며 대상 국가는 고객이 거주하는 국가와 다를 수 있습니다.

다음은 테이블과 해당 열입니다.

country_id	country
1	USA
2	UK
3	France
4	Germany
5	Spain

cust_id	last_name	loc_country
100	COLTRANE	1
101	MULLIGAN	1
102	WALDRON	3
103	HANCOCK	4
104	DAVIS	2
105	BARBIERI	5
106	STREATS	5

order_id	cust_id	order_date	ship_country
12345	100	1/1/95	2
12346	101	1/6/95	1
12347	101	2/6/95	3
12348	102	8/4/95	5
12349	103	10/3/95	4
12350	104	15/8/95	2
12351	105	6/2/95	5
12352	106	7/3/95	4

쿼리를 실행하여 다음과 같은 정보를 검색할 수 있습니다.

- 고객의 이름
- 고객이 거주하는 국가

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 각 주문 날짜
- 배송 대상 국가

이러한 데이터를 추출하는 SQL 은 다음과 같습니다.

```
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)
```

이 SQL 을 실행하면 완전하지 않은 결과가 반환됩니다. 즉, 자신이 거주하는 국가로 배송을 요청한 고객만 반환됩니다. 다른 국가로 배송을 요청한 고객은 반환되지 않습니다.

반환된 행은 고객이 거주하는 국가와 배송 대상 국가의 교집합입니다. 이 경우 다음과 같은 전체 결과가 생성되지 않습니다.

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

대신 이 SQL 은 다음과 같은 결과만 반환합니다.

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

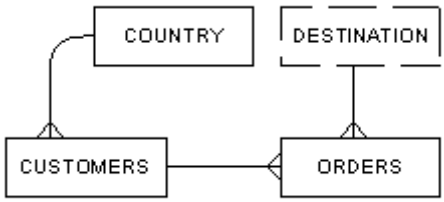
별칭을 삽입하면 루프를 중단할 수 있습니다. 별칭을 만들려면 먼저 데이터베이스 구조에서 둘 이상의 용도로 사용되는 조회 테이블을 식별해야 합니다. 이 작업은 다음 단원에서 설명합니다.

5.5.6.2 여러 가지 목적을 갖는 조회 테이블 식별

COUNTRY 테이블은 고객이 거주하는 국가 및 배송 대상 국가를 조회하는 데 사용됩니다. 이러한 테이블을 공유 조회 테이블이라고 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

스키마에 DESTINATION 이라는 별칭을 만듭니다.



원본 조인 세 개는 그대로 존재하지만 DESTINATION 별칭으로 루프가 중단되었으므로 닫힌 조인 경로가 더 이상 없습니다.

5.5.6.3 FROM 절에서 공유 조회 테이블 및 별칭 참조

이제 From 절에서 테이블 이름을 참조할 때 처음에는 원본 이름을 사용하고 두 번째에는 별칭을 사용하여 테이블 이름을 두 번 참조해야 합니다. 이렇게 하면 원본 이름 뒤에 대체 이름이 추가됩니다.

결과 SQL 은 다음과 같습니다.

```
SELECT   CUSTOMER.NAME,   COUNTRY.NAME,   ORDERS.ORDER_DATE   DESTINATION.NAME
FROM   CUSTOMER,   ORDERS,   COUNTRY,   COUNTRY DESTINATION WHERE
(CUSTOMER.CUST_ID=ORDERS.CUST_ID) AND   (ORDERS.SHIP_DEST_ID=
DESTINATION.COUNTRY_ID) AND   (CUSTOMER.CUST_LOC_ID=COUNTRY.COUNTRY_ID)
```

5.5.6.4 공유된 조회 테이블로 인해 발생한 루프를 중단하는 별칭 만들기

판매 데이터베이스에는 서로 다른 국가에 거주하는 고객에 대한 정보가 저장되어 있습니다. 각 고객은 여러 수송 기관이나 운송 업체를 통해 배송할 수 있는 상품을 주문할 수 있습니다.

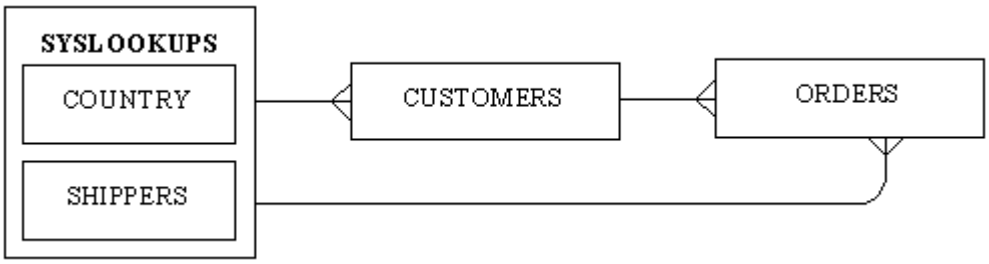
이 데이터베이스에는 수송 기관 및 운송 업체 이름이 조회 테이블에 정규화되어 있습니다. 정규화란 중복 내용을 제거하여 테이블 사이의 관계를 정리하는 과정입니다.

구조적인 이유로 인해 조회 테이블을 두 개 만드는 대신 코드, 설명 및 유형 필드를 가진 SYSLOOKUPS 라는 단일 조회 테이블을 만들었습니다. 유형 필드는 레코드에 저장된 정보 유형(예: 국가 또는 운송 업체)을 나타냅니다.

"융통성 있는 조회"라고도 하는 이러한 테이블 유형은 CASE 도구로 자동 생성된 스키마에 주로 사용됩니다.

스키마와 테이블 레이아웃은 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



cust_id	last_name	loc_country
100	COLTRANE	1
101	MULLIGAN	1
102	WALDRON	3
103	HANCOCK	4
104	DAVIS	2
105	BARBIERI	5
106	STREATS	5

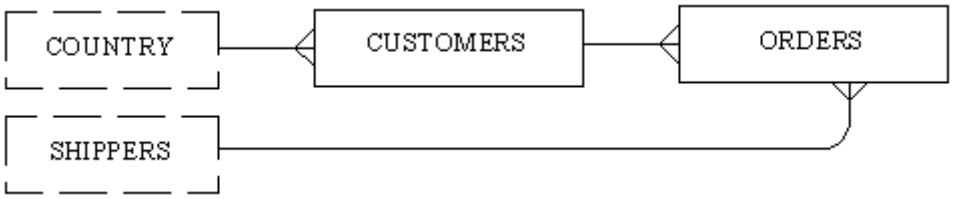
order_id	cust_id	order_date	ship_id
12345	100	1/1/95	2
12346	101	1/6/95	1
12347	101	2/6/95	3
12348	102	8/4/95	5
12349	103	10/3/95	4
12350	104	15/8/95	2
12351	105	6/2/95	5
12352	106	7/3/95	4

type	code	description
CTRY	1	USA
CTRY	2	UK
CTRY	3	France
CTRY	4	Germany
CTRY	5	Spain
SHIP	1	Man With A Van
SHIP	2	'Cut You Up' Couriers
SHIP	3	Parcel Fun
SHIP	4	Boggit & Leggit Couriers
SHIP	5	Deliveries 'R Us
SHIP	6	Sky Nut

SYSLOOKUPS 테이블은 한 가지 이상의 목적으로 사용되므로 테이블의 도메인 수(유형 필드의 고유한 값)만큼 별칭을 만들어야 합니다. SYSLOOKUPS 테이블이 나타내는 두 가지 목적에 기반하여 COUNTRY 와 SHIPPERS 라는 두 개의 별칭을 만들 수 있습니다.

결과 스키마는 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



유니버스 디자인 도구에서 `COUNTRY.DESCRIPTION` 으로 정의된 `Customer's Country` 개체를 만들고 `SHIPPERS.DESCRIPTION` 으로 정의된 `Shipper` 개체를 만듭니다.

해당 조인은 다음과 같습니다.

CUSTOMERS.LOC_COUNTRY=COUNTRY.CODE

ORDERS.SHIP_ID=SHIPPERS.CODE

자체 제한 조인을 사용하여 결과 제한

개체를 정의한 후에는 자신의 도메인 정보만 반환하도록 각 별칭을 제한해야 합니다. 자체 제한 조인을 만드는 방법에 대한 자세한 내용은 자체 제한 조인 [페이지 160] 단원을 참조하십시오.

예를 들어, 고객 101 에게 두 개의 주문을 배송한 운송 업체 이름을 확인하는 경우에는 두 개의 행이 반환될 것이라고 예상합니다.

그러나 다음 SQL 을 실행하는 경우

```
SELECT   ORDERS.ORDER_ID,    ORDERS.CUST_ID,   ORDERS.ORDER_DATE,
SHIPPERS.DESCRIPTION SHIPPER FROM   ORDERS,   SYSLOOKUPS SHIPPERS WHERE
(ORDERS.SHIP_ID=SHIPPERS.CODE)
```

다음과 같은 결과가 반환됩니다.

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

이 쿼리에서는 국가 이름과 운송 업체 이름을 반환했습니다. 결과에서 "Man With a Van"과 "USA"는 코드 1 을 공유하고 "France"와 "Parcel Fun"은 코드 3 을 공유합니다.

다음과 같이 이 오류를 수정할 수 있습니다.

- SHIPPERS 별칭에 자체 제한 조인을 새로 적용합니다. 조인 편집 대화 상자에서 테이블 1 과 테이블 2 모두를 SHIPPERS 로 설정하고 `SHIPPERS.TYPE='SHIP'`이라는 SQL 식을 입력합니다.
- COUNTRY 별칭에 자체 제한 조인을 새로 적용합니다. 조인 편집 대화 상자에서 테이블 1 과 테이블 2 모두를 COUNTRY 로 설정하고 `COUNTRY.TYPE='CTRY'`라는 SQL 식을 입력합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

제한을 사용할 때의 문제점

제한을 개체의 Where 절 또는 별칭과 CUSTOMERS/ORDERS 테이블 사이의 기존 조인에 추가하면 다음과 같은 문제가 발생할 수 있습니다.

- 개체의 Where 절에 제한을 추가하면 별칭에서 만든 모든 개체 각각에 동일한 제한을 추가해야 합니다. 여러 개의 열로 구성된 별칭에서 개체를 여러 개 만드는 경우에는 유니버스를 유지하는 데 어려움이 따를 수 있습니다.
- 별칭과 다른 테이블 사이의 조인을 제한하면 조인이 호출될 때만 제한이 적용됩니다. **운송 업체** 개체만 포함된 간단한 쿼리를 실행하면 ORDERS 테이블을 포함시킬 이유가 없으므로 SHIPPERS 별칭의 모든 행(원하지 않는 Country 행 포함)이 반환됩니다. 이 경우 조인이 필수 요소로 간주되지 않으므로 제한이 적용되지 않습니다.

요약

이 예제에서는 공유 조회 테이블이 있는 스키마를 사용했습니다. 이 예제에서 수행한 작업을 다음과 같이 요약할 수 있습니다.

1. 공유 조회 테이블에 대해 COUNTRY 및 SHIPPERS 별칭을 만듭니다.
2. 자체 제한 조인을 만들어 별칭을 제한합니다.

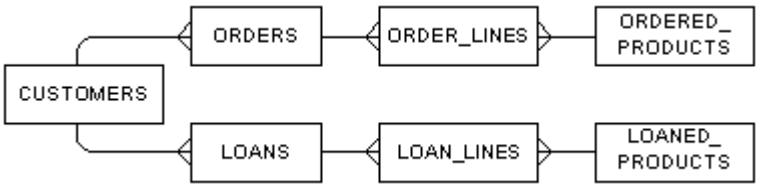
이 예제에서 사용한 별칭은 조합된 조회 테이블을 두 개의 서로 다른 조회 테이블로 사용하여 루프를 해결합니다. 일부 구조에서 별칭으로 인해 추가 조정 및 제한이 필요한 경우에 사용할 수 있도록 이들 별칭에 제한(자체 조인)을 적용해야 합니다.

5.5.6.5 별칭을 사용하여 루프를 중단할 수 없는 경우 식별

앞에서 설명한 방법, 즉 별칭을 만들어 루프를 해결하는 것이 최선의 방법은 아닙니다. 컨텍스트를 사용하는 것이 더 적절한 경우도 있습니다. 다음 예제에서는 별칭이 적절하지 않은 이유와 이 경우 컨텍스트가 더 적절한 이유를 설명합니다.

두 가지 이상의 목적으로 사용되는 조회 테이블을 식별할 때 해당 테이블이 PRODUCTS 테이블인지 CUSTOMERS 테이블인지 명확하지 않습니다.

다음과 같이 PRODUCTS 테이블의 별칭 두 개를 만든다고 가정합니다.

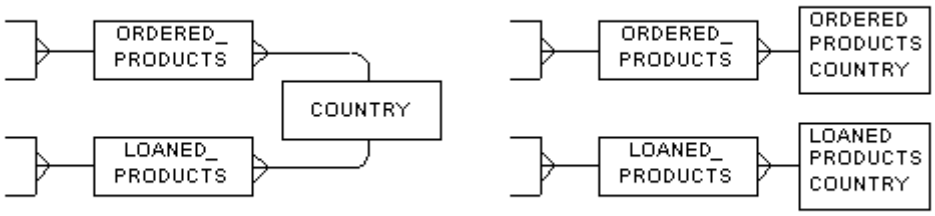


두 별칭은 ORDERED_PRODUCTS 와 LOANED_PRODUCTS 입니다. 사용자는 주문한 제품 또는 대여한 제품보다는 제품이라는 단어를 더 쉽게 이해할 수 있으므로 이 두 가지 별칭은 사용자에게 혼동을 줄 수 있습니다.

제품이 여러 국가에서 제조되었음을 나타내기 위해 COUNTRY 테이블도 추가하려면 이 테이블을 PRODUCTS 테이블에 직접 조인해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

결과 스키마는 다음과 같습니다.



이 스키마에서 ORDERED_PRODUCTS_COUNTRY 와 LOANED_PRODUCTS_COUNTRY 라는 두 개의 별칭을 새로 만들었습니다. 이 스키마에서는 별칭을 사용한 해결 방법이 적절하지 않으며 복잡합니다.

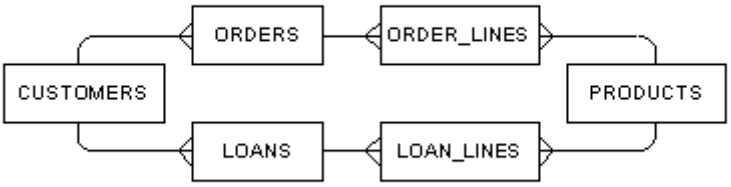
이 경우에는 컨텍스트를 만들어야 합니다.

5.5.6.6 루프를 해결하는 컨텍스트 만들기

제품을 구입하거나 대여할 수 있는 고객에 대한 정보가 들어 있는 데이터베이스가 있습니다. 이 데이터베이스에서는 다음과 같은 두 가지 방법으로 고객과 제품 사이의 관계를 나타낼 수 있습니다.

- 고객이 주문(또는 구입)한 제품
- 고객이 대여한 제품

이 데이터베이스의 스키마 유형은 다음과 같습니다.



고객 이름 목록과 제품 목록만 반환하는 쿼리를 실행하려면 ORDER 테이블과 ORDER_LINES 테이블을 사용하면 됩니다. 그러면 각 고객이 주문한 제품 목록이 반환됩니다.

LOANS 와 LOAN_LINES 테이블을 사용하면 각 고객이 대여한 제품 목록을 가져올 수 있습니다.

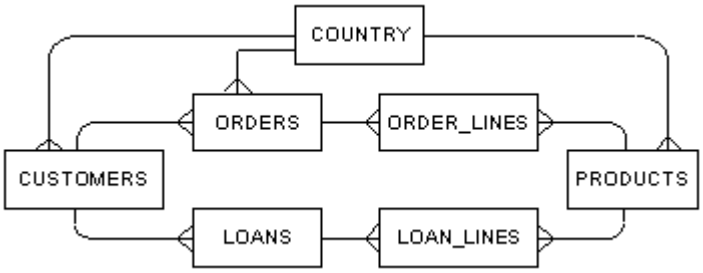
이 스키마에는 6 개 조인 모두를 동시에 사용하는 쿼리를 실행했을 때 고객에게 판매되고 동시에 대여된 제품의 전체 목록을 반환하는 루프가 포함되어 있습니다. 고객에게 판매되었지만 대여되지 않았거나 반대로, 대여되었지만 판매되지 않은 제품은 결과 목록에 표시되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

별칭과 컨텍스트를 함께 사용하여 루프 해결

컨텍스트와 별칭을 사용하여 유니버스 내의 루프를 해결할 수 있습니다. 다음 예제에서는 별칭과 컨텍스트를 함께 사용하여 루프를 해결하는 방법을 보여 줍니다.

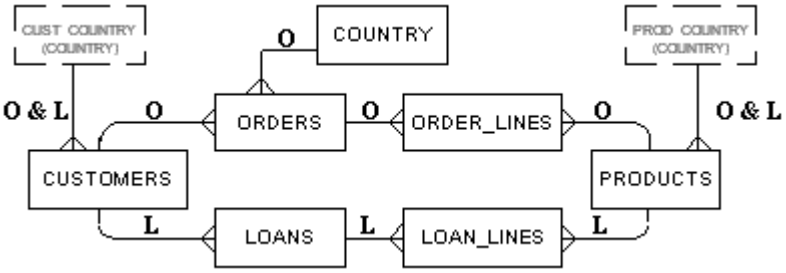
유니버스의 스키마는 다음과 같습니다.



다음과 같이 별칭과 컨텍스트를 사용하여 루프를 해결할 수 있습니다.

- COUNTRY 테이블의 별칭 두 개(CUST_COUNTRY 및 PROD_COUNTRY)를 만듭니다.
- CUSTOMERS 에서 PRODUCTS 까지의 루프(Orders 및 Loans)를 해결하는 컨텍스트 두 개를 정의합니다.
- CUSTOMERS 와 CUST_COUNTRY 사이의 조인 및 PRODUCTS 와 PROD_COUNTRY 사이의 조인이 두 컨텍스트 모두에 표시되는지 확인합니다.

결과 스키마는 다음과 같습니다.



5.6 캐즘 트랩 해결

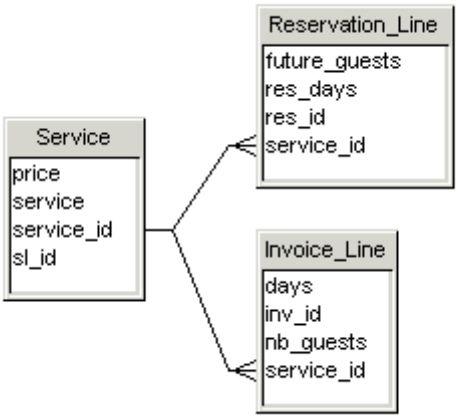
캐즘 트랩은 조인 경로에서 예상보다 많은 데이터를 반환하는 경우로 관계형 데이터베이스에서 일반적으로 발생하는 문제입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.6.1 캐즘 트랩이란?

캐즘 트랩은 "다대일" 조인 두 개가 단일 테이블에 수렴되고, 수렴된 조인 경로를 구분할 수 있는 컨텍스트가 없는 경우에 세 테이블 사이에 형성되는 조인 경로 유형입니다.

다음 예제에서는 Beach 유니버스 스키마의 일부를 보여 줍니다. 아래의 테이블 세 개는 캐즘 트랩을 설명하기 위해 스키마의 나머지 부분과 구분되었습니다. 여기에서는 동일한 클럽 연결을 통해 데이터를 가져옵니다. Service 테이블은 두 개의 일대다 조인에서 일(one)에 해당합니다.



다음 조건을 모두 충족하는 경우에만 잘못된 결과가 반환됩니다.

- 유니버스 구조 내의 테이블 세 개 사이에 "다대일대다(many to one to many) 관계"가 존재합니다.
- 테이블 두 개에 기반한 개체가 쿼리에 포함되어 있고, 두 테이블 모두 각 조인의 "다(many)"에 해당합니다.
- 단일 차원에 대해 여러 개의 행이 반환되었습니다.

다음 예제는 위의 조건을 모두 만족하는 경우에 쿼리를 실행했을 때 카디전 곱이 반환되는 방법을 보여 줍니다.

예

경고 없이 결과를 확장하는 캐즘 트랩

위의 스키마를 사용하여 Web Intelligence 사용자가 다음과 같은 별도의 쿼리를 실행합니다.

표 125:

쿼리	반환 결과
Service, Number of guests Service Equal to 'Sports'	Service: Sports Number of guests: 145.00

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

쿼리	반환 결과
Service Number of future guests Service Equal to 'Sports'	Service: Sports Number of future guests: 8.00

이제 현재 손님과 향후 손님을 모두 포함하는 쿼리를 실행합니다.

Service Number of guests Number of future guests
Service Equal to 'Sports'

다음과 같은 결과가 반환됩니다.

Service	Number of guests	Number of future guests
Sports	188.00	96.00

스포츠 서비스를 사용한 손님 수와 예약한 향후 손님 수가 크게 증가했음을 알 수 있습니다. 여기에서는 카디전 곱이 반환되어 결과가 정확하지 않습니다. 이 문제를 발견하지 못하면 심각한 결과가 초래될 수 있습니다. 위의 예제 결과를 확인한 Island Resorts 의 관리자는 휴양지에서 제공하는 스포츠 서비스가 실제 수치가 나타내는 것보다 손님에게 더 많은 인기를 얻고 있다고 생각할 수 있습니다.

5.6.2 캐즘 트랩이 결과를 확장하는 방법

캐즘 트랩은 특정 계수의 가능한 모든 행 조합과 다른 계수의 가능한 모든 행 조합을 쿼리에서 반환하도록 만듭니다. 위의 예제에서는 다음 작업이 발생했습니다.

- 손님 트랜잭션 수*향후 손님 트랜잭션 수
- 향후 손님 트랜잭션 수*손님 트랜잭션 수

다음 예제에서는 캐즘 트랩으로 인해 카디전 곱이 반환되는 방법을 자세하게 살펴봅니다.

예

캐즘 트랩의 카디전 곱 검사

쿼리에서 반환되어 집계 수치를 구성하는 행을 확인해야 합니다. 이 예제에서는 Days Billed 차원과 Days Reserved 차원을 쿼리에 추가하여 반환되는 개별 트랜잭션 세부 사항 확인할 수 있습니다.

손님 수 보고서는 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

Service	Days billed	Number of guests
Sports	3.00	4.00
Sports	4.00	133.00
Sports	6.00	8.00

향후 손님 수 보고서는 다음과 같습니다.

Service	Days reserved	Number of future guests
Sports	1.00	7.00
Sports	2.00	1.00

두 보고서의 트랜잭션 수는 다음과 같습니다.

- 손님 수 = 트랜잭션 3 개
- 향후 손님 수 = 트랜잭션 2 개

쿼리에 두 트랜잭션을 함께 추가하면 다음과 같은 결과가 반환됩니다.

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**

쿼리는 손님 수 행의 가능한 모든 조합과 향후 손님 수 행의 가능한 모든 조합을 반환합니다. 각 조합에 손님 수 트랜잭션이 두 번 나타나고 각 조합에 향후 손님 수 트랜잭션이 세 번 나타납니다.

반환된 데이터의 합계를 계산하면 해당 값도 올바르지 않습니다.

루프와 달리 캐즘 트랩은 유니버스 디자인 도구에서 자동으로 검색되지 않지만, 컨텍스트 검색(도구 > 컨텍스트 검색) 기능을 사용하면 스키마에서 후보 컨텍스트가 자동으로 검색되고 제안됩니다.

컨텍스트 검색 기능은 스키마에서 다대일 조인을 검사합니다. 컨텍스트 검색은 수렴되는 다대일 조인으로 연결된 테이블을 찾은 후 해당 테이블에서 실행되는 쿼리를 구분할 수 있는 컨텍스트를 제안합니다. 스키마에서 캐즘 트랩을 확인하는 데는 이 방법이 가장 효과적입니다.

스키마에서 일대다 조인 경로를 그래픽적으로 분석하여 캐즘 트랩을 찾을 수도 있습니다.

컨텍스트 검색을 실행하지 않거나 스키마에서 캐즘 트랩을 찾지 않으면 세부 행을 검토하는 방법으로 문제를 찾아야 합니다. 이러한 방법 외에는 캐즘 트랩 문제를 사전에 확인할 수 없습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.6.3 캐즘 트랩 검색

캐즘 트랩을 찾으려면 컨텍스트 검색 기능을 사용하여 후보 컨텍스트를 검색하고 제안한 후 컨텍스트 두 개가 분기되는 테이블을 확인합니다. 컨텍스트 두 개가 교차하는 지점이 캐즘 트랩의 원인입니다.

캐즘 트랩은 팩트 테이블 두 개에서 시작된 다대일 조인이 단일 조회 테이블에 수렴되는 경우에 발생할 수 있습니다.

> 팁
>
> 조인 문제를 검색할 수 있도록 테이블 스키마를 구성하는 데 대한 자세한 내용은 그래픽적으로 조인 문제 검색 [페이지 233]을 참조하십시오.

5.6.4 캐즘 트랩 해결

캐즘 트랩을 해결하려면 개별 쿼리 두 개를 만든 다음 결과를 조합해야 합니다. 팩트 테이블에 대해 정의된 개체 유형과 최종 사용자 환경 유형에 따라 다음과 같은 방법으로 캐즘 트랩을 해결할 수 있습니다.

- 팩트 테이블 각각에 대해 컨텍스트를 만듭니다. 이 해결 방법은 모든 경우에 사용할 수 있습니다.
- 각 계수에 대해 별도의 SQL 쿼리를 생성할 수 있도록 유니버스의 SQL 매개 변수를 수정합니다. 이 해결 방법은 계수 개체에 대해서만 사용할 수 있습니다. 차원이나 설명 개체에 대해서는 별도의 쿼리가 생성되지 않습니다.

각 해결 방법은 다음 단원에서 자세하게 설명합니다.

5.6.4.1 컨텍스트를 사용하여 캐즘 트랩 해결

조인의 "다(many)"에 해당하는 각 테이블에 대해 컨텍스트를 정의할 수 있습니다. 이 예제에서는 SERVICE 에서 RESERVATION_LINE 사이 및 SERVICE 에서 INVOICE_LINE 사이에 대해 컨텍스트를 정의할 수 있습니다.

두 컨텍스트의 개체가 포함된 쿼리를 실행하면 Select 문 두 개가 만들어진 후 동기화되어 Web Intelligence 에 개별 테이블 두 개가 생성되기 때문에 카디전 곱이 반환되지 않습니다.

5.6.4.2 컨텍스트를 사용하는 경우

컨텍스트를 만들면 유니버스에서 캐즘 트랩을 항상 해결할 수 있습니다. 팩트 테이블 둘 중 하나 또는 둘 모두에 차원 개체가 있는 경우에는 컨텍스트를 항상 사용해야 합니다.

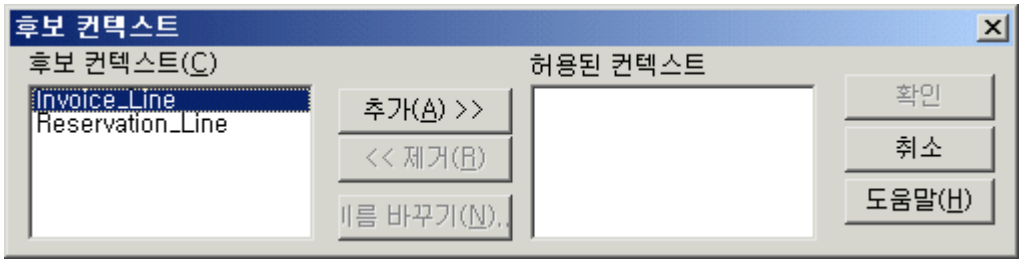
5.6.4.3 컨텍스트를 사용하여 캐즘 트랩 해결

컨텍스트를 사용하여 캐즘 트랩을 해결하려면

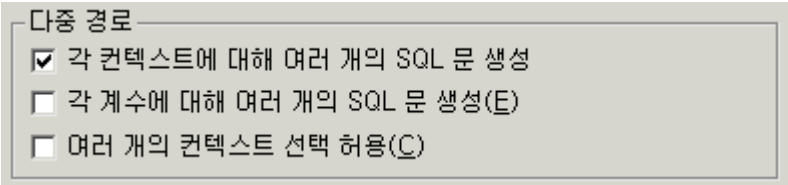
1. 스키마에서 "일대다대일(one-to-many-to-one)" 조인 경로 관계를 분석하여 잠재적인 캐즘 트랩을 식별합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 도구 > 컨텍스트 검색을 선택합니다.
   후보 컨텍스트 상자가 나타납니다.



3. 후보 컨텍스트 목록 상자에서 제안된 컨텍스트를 선택한 다음 추가 단추를 클릭하여 선택한 컨텍스트 목록 상자에 추가합니다.
4. 목록에 있는 다른 컨텍스트에 대해 단계를 반복합니다.
   목록 뷰 모음의 컨텍스트 창에 새 컨텍스트가 나열됩니다.
5. 파일 > 매개 변수를 선택합니다.
   유니버스 매개 변수 대화 상자가 나타납니다.
6. SQL 탭을 클릭합니다.
   SQL 페이지가 나타납니다.
7. 각 컨텍스트에 여러 SQL 문 사용 확인란을 선택합니다.



8. 확인을 클릭합니다.
   캐즘 트랩 내의 테이블에 대해 쿼리를 실행하면 영향을 받는 테이블에 정의된 계수와 차원에 대해 쿼리가 구분됩니다.

5.6.4.4 각 계수에 여러 SQL 문 사용 옵션 사용

두 팩트 테이블에 계수 개체만 정의되어 있는 경우에는 각 계수에 여러 SQL 문 사용이라는 유니버스 매개 변수 옵션을 사용할 수 있습니다. 이 옵션을 선택하면 쿼리 창에 표시되는 각 계수에 대해 별도의 SQL 쿼리가 강제로 생성됩니다.

차원 개체와 설명 개체에 대해서는 이 방법을 사용할 수 없습니다.

다음 표에서는 각 계수에 여러 SQL 문 사용 옵션을 사용해야 하는 경우와 사용하지 말아야 하는 경우를 설명합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 126:

옵션 사용 여부	조건
각 계수에 여러 SQL 문 사용 옵션 사용	두 팩트 테이블 모두에 대해 계수 개체만 포함된 유니버스가 정의됩니다. 여러 개의 SQL 문을 사용하면 나중에 유지 작업이 필요한 컨텍스트를 사용하지 않아도 됩니다.
각 계수에 여러 SQL 문 사용 옵션 사용 안 함	팩트 테이블 둘 중 하나 또는 둘 모두에 차원 개체나 설명 개체가 있습니다. 이 해결 방법을 사용하는 유니버스에 기반한 쿼리가 차원 개체나 설명 개체를 포함하고 있으면 카티전 곱이 반환됩니다. 이 방법을 사용하면 쿼리 응답 시간이 길어지고 올바르지 않은 결과가 반환될 수 있으므로 컨텍스트를 만들어 캐즘 트랩을 해결하는 것이 좋습니다.

각 계수에 여러 SQL 문 사용 옵션을 활성화하려면

1. 메뉴 모음에서 파일 > 매개 변수를 선택합니다.
   유니버스 매개 변수 대화 상자가 나타납니다.
2. SQL 탭을 클릭합니다.
3. 여러 경로 그룹 상자에서 각 계수에 여러 SQL 문 사용 확인란을 선택합니다.
4. 확인을 클릭합니다.

5.7 팬 트랩 해결

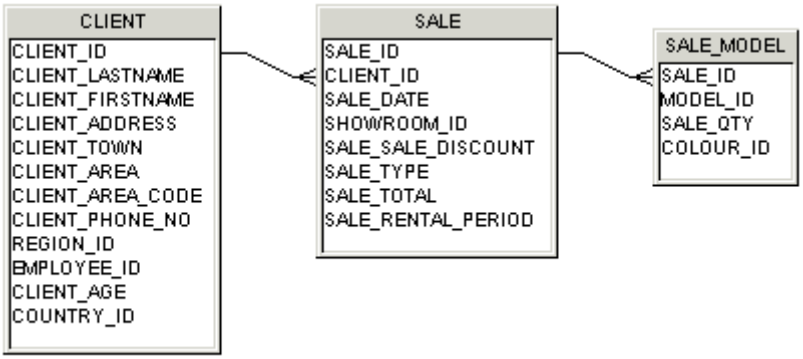
관계형 데이터베이스 스키마에서 팬 트랩은 캐즘 트랩보다 일반적이지 않습니다. 팬 트랩도 캐즘 트랩과 마찬가지로 예상보다 많은 데이터를 반환합니다.

5.7.1 팬 트랩이란?

팬 트랩은 "일대다" 조인으로 다른 테이블에 연결된 테이블에 "일대다" 조인으로 연결하는 경우 세 테이블 사이에 형성되는 조인 경로 유형입니다. 이렇게 "일대다" 조인으로 계속 연결할 경우 두 테이블 모두에 정의된 개체가 쿼리에 포함되어 있으면 잘못된 결과가 반환될 수 있습니다.

다음은 팬 트랩을 보여 주는 간단한 예제입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

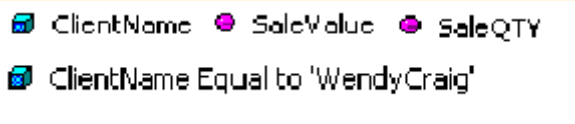


특정 고객에 대해 판매된 총 자동차 개수를 모델별로 검색하는 쿼리를 실행하면 "다(many)"쪽에 연결되어 있는 상태에서 조인의 "일(one)"쪽에서 테이블에 대해 집계 함수를 실행하게 되므로 올바르지 않은 결과가 반환됩니다.

예

경고 없이 결과를 확장하는 팬 트랩

위의 스키마를 사용하여 Web Intelligence 사용자가 다음과 같은 쿼리를 실행합니다.



다음과 같은 결과가 반환됩니다.

ClientName	SaleQTY	SaleValue
WendyCraig	2.00	57,092.00

이 결과에는 문제가 없습니다. 그러나 다음과 같이 최종 사용자가 모델 ID 차원을 쿼리에 추가한다고 가정합니다.

그러면 반환된 결과를 사용하여 다음과 같은 보고서가 생성됩니다.

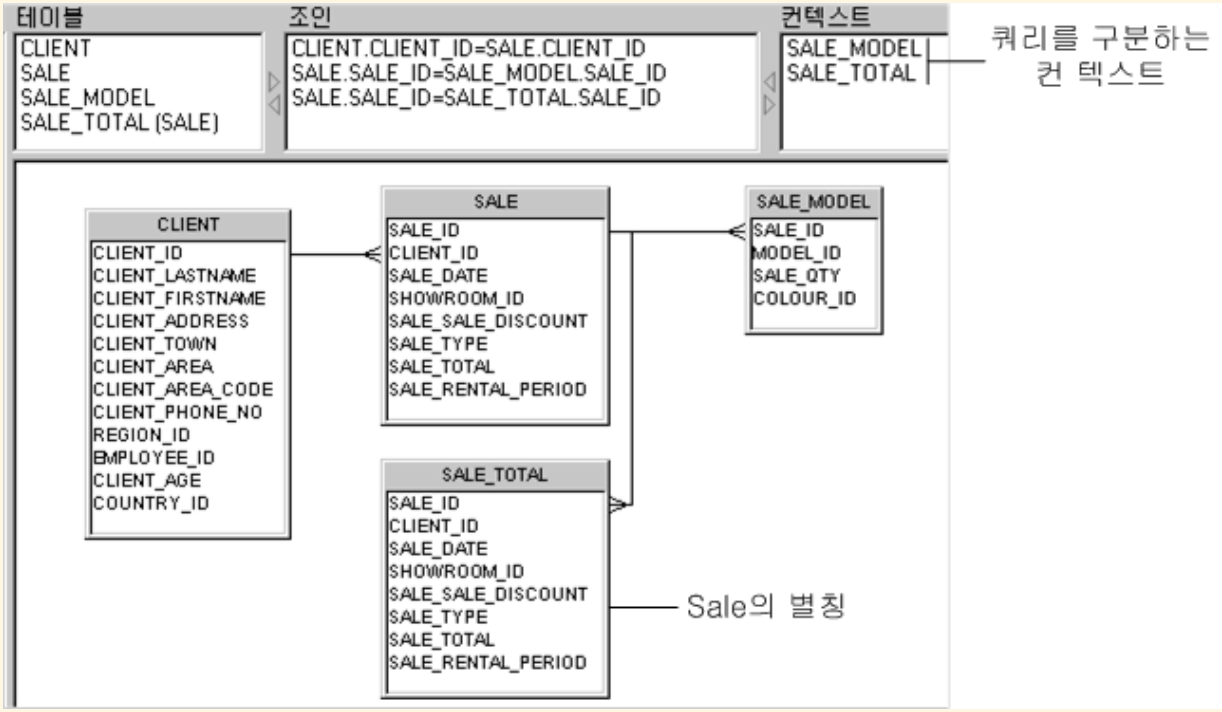
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

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**

판매액 집계가 두 번 표시됩니다. 즉, 각 모델_ID 인스턴스에 대해 한 번씩 표시됩니다. 이 결과를 보고서에서 집계하면 합계가 올바르지 않습니다. 이 보고서에는 팬 트랩으로 인해 카디전 곱이 반환되었습니다. 실제로 Wendy 는 보고서에 표시된 금액인 $114,184.00 가 아니라 $57,092.00 에 자동차 두 대를 구입했습니다. 이 예제에서는 쿼리에 모델_ID 를 포함시켰기 때문에 판매 가격을 모델_ID 의 행 수만큼 곱했습니다.

쿼리에 차원 개체를 사용하는 팬 트랩은 별칭과 컨텍스트를 사용하여 해결할 수 있습니다. 다음 스키마를 사용하여 팬 트랩 스키마를 해결할 수 있습니다.



Wendy Craig 에 대해 카디전 곱을 반환했던 원본 쿼리를 위의 해결 방법을 사용하여 실행하면 다음과 같은 테이블이 반환됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

WendyCraig

Sale Qty	Model Id
1.00	1,034.00
1.00	1,081.00

Sale Total
57,092.00

5.7.2 팬 트랩 해결 방법

팬 트랩은 자동으로 검색할 수 없습니다. 따라서 테이블 스키마에 표시된 카디널리티 방향을 시각적으로 확인해야 합니다.

팬 트랩은 두 개의 테이블이 일련의 다대일 조인으로 조인되어 있고 해당 테이블을 계수 개체에서 참조하는 경우에 발생할 수 있습니다.

조인 문제를 검색할 수 있도록 테이블 스키마를 구성하는 방법은 그래픽적으로 조인 문제 검색 [페이지 233] 단원을 참조하십시오.

5.7.3 팬 트랩 해결 방법

두 가지 방법으로 팬 트랩 문제를 해결할 수 있습니다.

- 초기 집계가 포함된 테이블의 별칭을 만든 다음 컨텍스트 검색(도구 > 컨텍스트 검색)을 사용하여 별칭 테이블과 원본 테이블 각각에 대해 컨텍스트를 검색하고 제안합니다. 팬 트랩 문제를 해결하는 데는 이 방법이 가장 효과적입니다.
- 유니버스의 SQL 매개 변수를 변경합니다. 이 해결 방법은 계수 개체에 대해서만 사용할 수 있습니다.

두 가지 방법은 아래에서 자세하게 설명합니다.

5.7.3.1 별칭과 컨텍스트를 사용하여 팬 트랩 해결

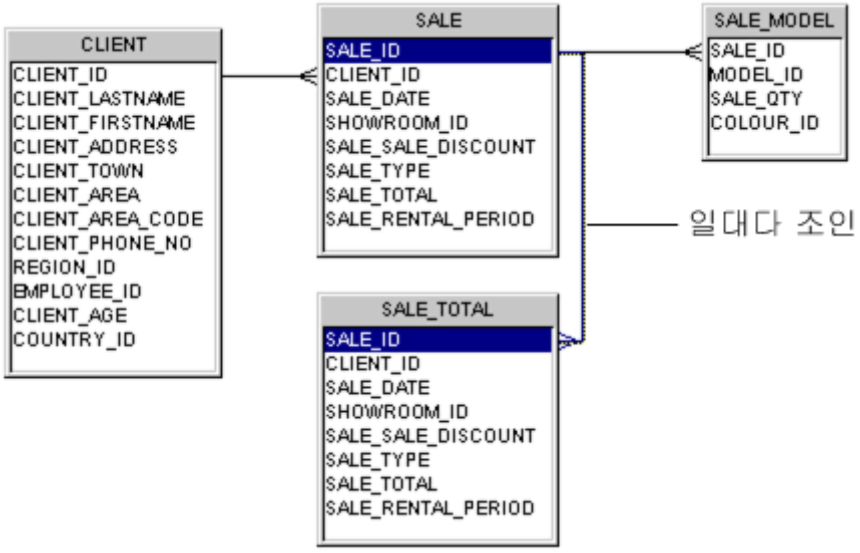
집계를 만드는 테이블의 별칭을 만든 다음 컨텍스트를 검색하고 구현하여 쿼리를 구분합니다. 다음과 같이 할 수 있습니다.

별칭과 컨텍스트를 사용하여 팬 트랩을 해결하려면

1. 스키마에서 "일대다대일대다(one-to-many-to-one-to-many)" 조인 경로 관계를 분석하여 잠재적인 팬 트랩을 식별합니다.
2. 곱셈 집계를 만드는 테이블의 별칭을 만듭니다.
   예를 들어, 앞에 설명한 예제에서 판매 가격은 Sale 테이블에 있는 Sale_Total 열의 집계입니다. 따라서 Sale 에 대해 Sale_Total 이라는 별칭을 만듭니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. 원본 테이블과 별칭 테이블 사이에 조인을 만듭니다.
   일대일 조인을 만들면 유니버스 디자인 도구에서 컨텍스트를 자동으로 검색하지 못하므로 컨텍스트를 수동으로 만들어야 합니다. 일반적으로 컨텍스트를 자동으로 검색하고 구현할 수 있는 일대다 조인을 사용하는 것이 좋습니다. 예를 들어, 다음과 같이 Sale 과 Sale_Total 사이에 일대다 조인을 만듭니다.



4. 집계의 원인이 되는 개체를 별칭 테이블에 정의합니다.
   예를 들어, 원본 판매 가격 개체는 다음과 같이 정의되었습니다.
   sum(SALE.SALE_TOTAL). 판매 가격을 다음과 같이 새로 정의합니다.
   sum(Sale_Total.SALE_TOTAL)
5. 도구 > 컨텍스트 검색을 선택합니다.
   후보 컨텍스트 상자가 나타납니다. 이 상자에는 기본 테이블의 조인 경로 및 별칭 테이블의 새 조인 경로에 대한 후보 컨텍스트가 제안됩니다.

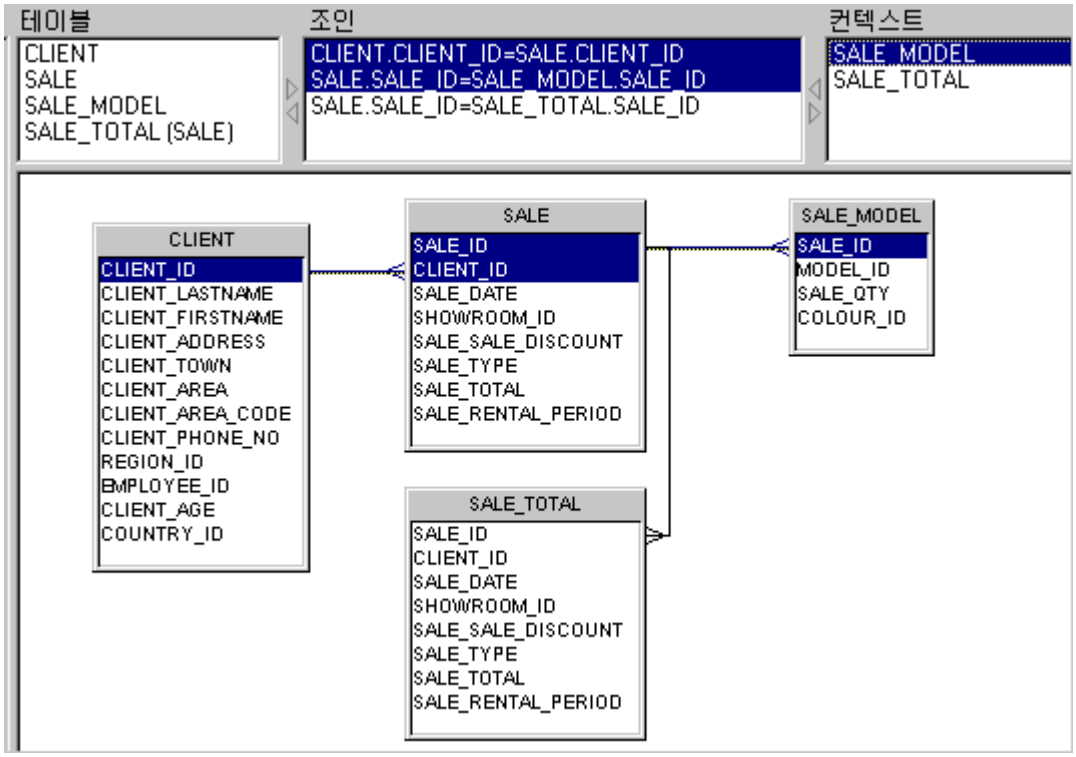
   노트

   별칭과 기본 테이블 사이에 일대일 조인을 사용한 경우에는 컨텍스트를 수동으로 만들어야 합니다.

6. 후보 컨텍스트를 클릭한 다음 추가를 클릭합니다.
7. 다른 후보 컨텍스트에 대해서도 이 작업을 반복합니다.

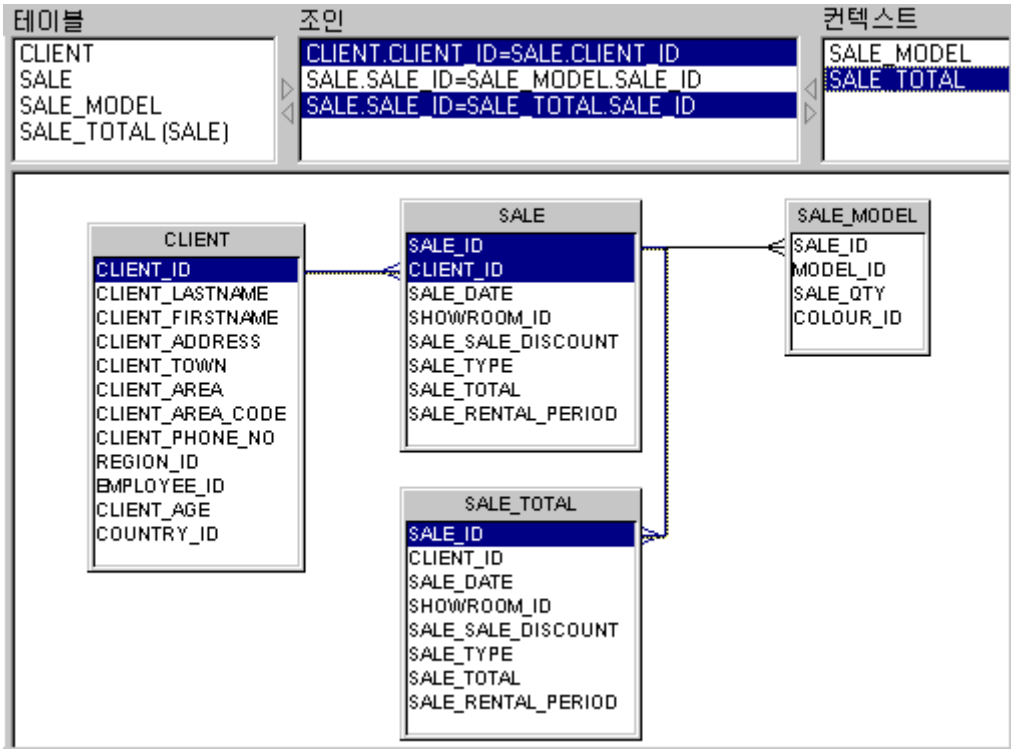
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8. 확인을 클릭합니다.
   스키마에 컨텍스트가 만들어졌습니다. 목록 모드(보기 > 목록 모드)를 활성화하면 컨텍스트 창에 이들 컨텍스트가 표시됩니다. CLIENT>SALE>SALE_MODEL 조인 경로의 컨텍스트는 다음과 같이 표시됩니다.

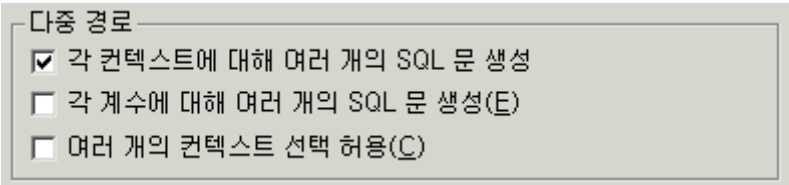


   CLIENT>SALE>SALE_TOTAL 조인 경로의 컨텍스트도 다음과 같이 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



9. 파일 > 매개 변수를 선택합니다.
   유니버스 매개 변수 대화 상자가 나타납니다.
10. SQL 탭을 클릭합니다.
    SQL 페이지가 나타납니다.
11. 각 컨텍스트에 여러 SQL 문 사용 확인란을 선택합니다.



12. 확인을 클릭합니다.
13. 쿼리를 실행하여 팬 트랩이 해결되었는지 확인합니다.

5.7.3.2 각 계수에 여러 SQL 문 사용 옵션 사용

연속 일대다 조인의 다(many)에 해당하는 두 테이블에 계수 개체만 정의되어 있는 경우에는 각 계수에 여러 SQL 문 사용이라는 유니버스 매개 변수 옵션을 사용할 수 있습니다. 이 옵션을 선택하면 쿼리 창에 표시되는 각 계수에 대해 별도의 SQL 쿼리가 강제로 생성됩니다.

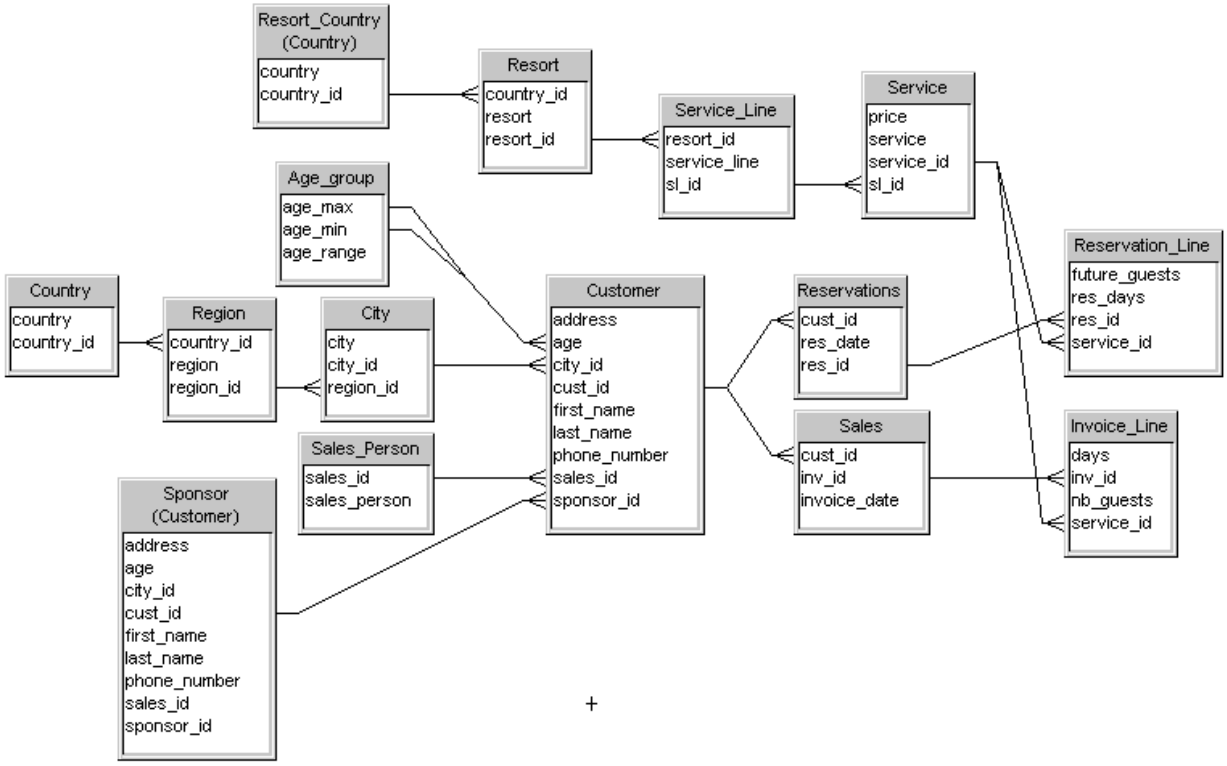
이 방법은 차원에 대해 쿼리를 여러 개 생성하는 데 사용할 수 없습니다. 최종 사용자가 계수 개체를 참조하는 테이블의 차원을 쿼리에 포함할 수 있는 경우에는 별칭과 컨텍스트를 사용하여 팬 트랩을 해결해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

이 옵션을 활성화하는 절차 및 기타 자세한 내용은 각 계수에 여러 SQL 문 사용 옵션 사용 [페이지 232] 단원을 참조하십시오.

5.8 그래픽적으로 조인 문제 검색

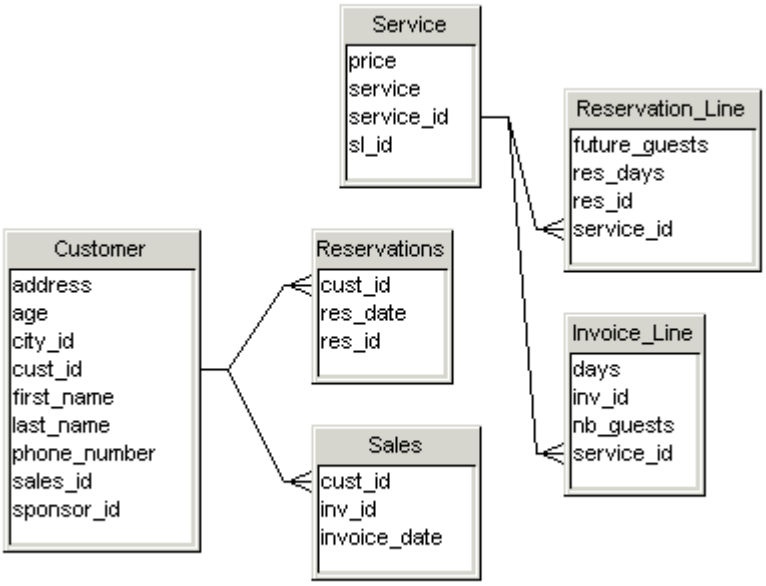
조인의 "다(many)"쪽이 한 쪽 방향을 향하도록 하고 "일(one)"쪽이 다른 쪽 방향을 향하도록 구조 창에서 테이블을 정렬하면 스키마의 테이블 중에서 잠재적인 캐즘 트랩과 팬 트랩을 시각적으로 검색할 수 있습니다. 다음 예제는 일대다 흐름이 왼쪽에서 오른쪽 방향으로 정렬된 Beach 유니버스 스키마를 보여 줍니다.



5.8.1 잠재적인 캐즘 트랩

다음은 잠재적인 캐즘 트랩을 보여 줍니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

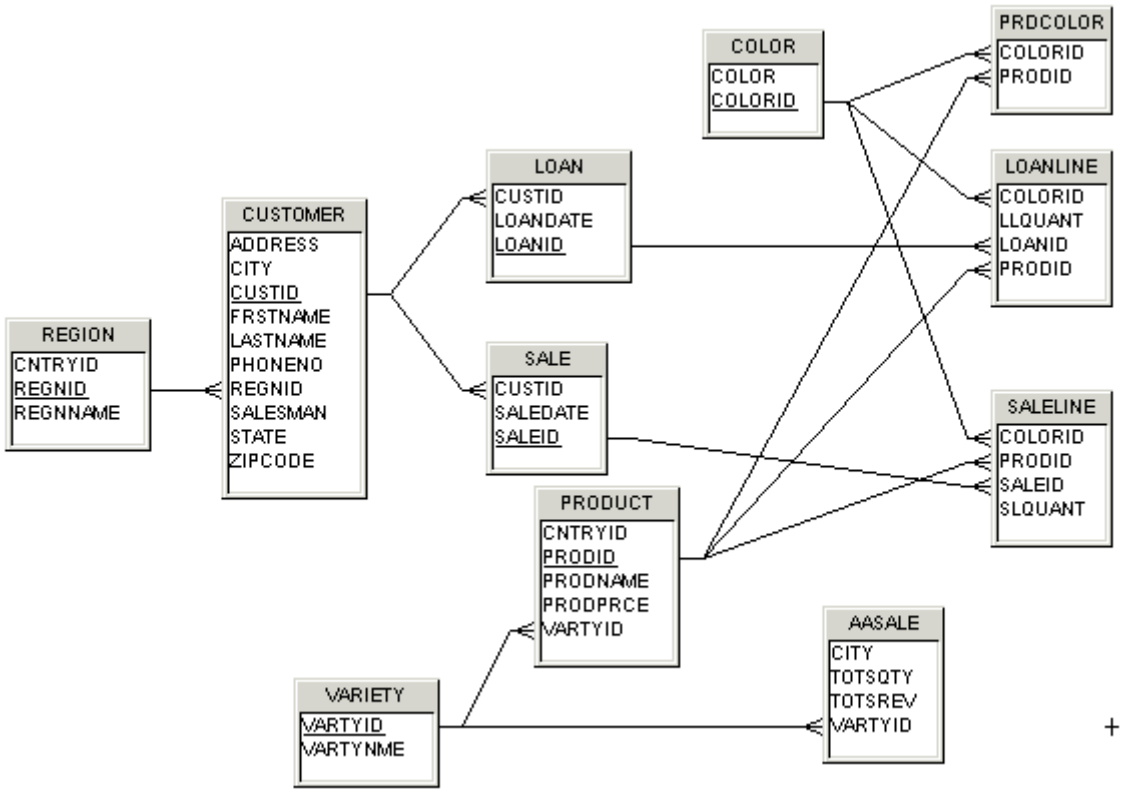


두 조인 경로 모두 Sales 와 Reservations 라는 컨텍스트를 사용하여 구분되었습니다.

5.8.2 잠재적인 팬 트랩

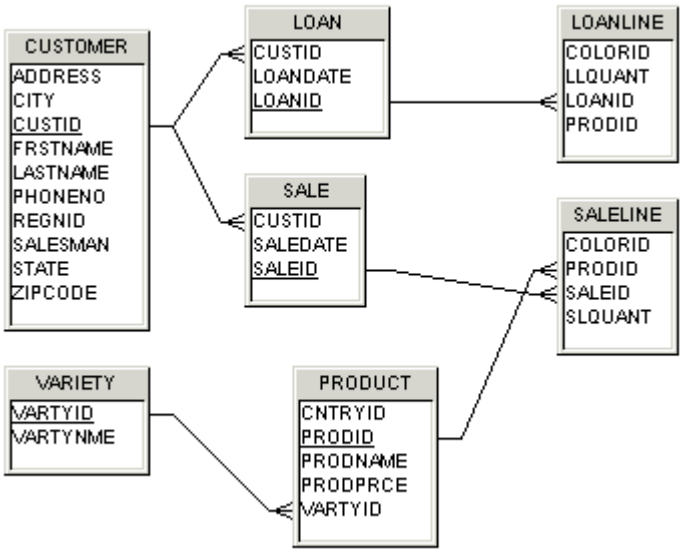
다음은 자동차 판매 데이터베이스의 유니버스 스키마를 보여 줍니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



잠재적인 팬 트랩과 관련된 테이블은 다음과 같습니다.

- CUSTOMER, LOAN 및 LOANLINE
- CUSTOMER, SALES 및 SALELINE
- VARIETY, PRODUCT 및 SALELINE



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

➔ 팁

필요한 테이블로 스키마를 채운 후 개체 정의를 곧바로 시작하지 않는 것이 좋습니다. 시간을 두고 테이블을 이동해야만 동일한 방향으로 일대다 조인을 모두 사용할 수 있습니다. 유니버스 디자인 도구는 그래픽 도구이므로 유니버스를 디자인할 때 제품의 시각적 기능을 사용하는 것이 유용합니다. 한 시간 정도 시간을 투자하여 테이블을 효과적으로 배치하면 이후 디자인 과정에서 상당한 시간을 절약할 수 있습니다.

5.9 유니버스 검사

유니버스를 디자인할 때는 무결성을 주기적으로 테스트해야 합니다. 다음과 같이 유니버스의 무결성을 확인할 수 있습니다.

표 127:

유니버스 검사	설명
자동	유니버스를 만들거나 유니버스를 내보내거나 유니버스를 열 때 유니버스 구조의 SQL 구문을 검사하도록 유니버스 디자인 도구 옵션을 설정할 수 있습니다.
수동	선택한 유니버스 구조에 대해 무결성 검사를 직접 실행할 수 있습니다.

5.9.1 자동으로 유니버스 무결성 검사

유니버스 디자인 도구에서는 유니버스를 만들거나 유니버스를 내보내거나 유니버스를 열 때 SQL 의 구문을 분석하도록 다음과 같이 무결성 검사 옵션을 설정할 수 있습니다.

표 128:

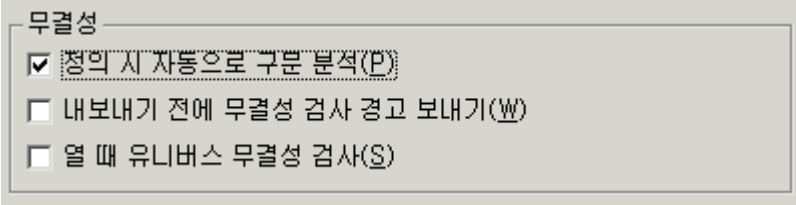
자동 검사 옵션	설명
정의할 때 자동으로 구문 분석	모든 개체, 조건 및 조인을 만들 때 해당 SQL 정의를 유니버스 디자인 도구에서 자동으로 검사합니다. 이 옵션은 구조 만들기 확인 메시지에서 확인을 클릭했을 때 적용됩니다.
무결성 검사 경고 보내기	검사되지 않은 유니버스를 내보내려고 할 때마다 유니버스 디자인 도구에서 경고를 표시합니다.
열 때 유니버스 무결성 검사	유니버스를 열 때마다 자동으로 검사합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5.9.1.1 유니버스 자동 검사 옵션 설정

유니버스 자동 검사 옵션을 설정하려면

1. 도구 > 옵션을 선택합니다.
   옵션 대화 상자의 일반 페이지가 열립니다.
2. 무결성 그룹 상자에서 적절한 유니버스 자동 검사 옵션의 확인란을 선택하거나 선택 취소합니다.



3. 확인을 클릭합니다.

5.9.2 수동으로 유니버스 무결성 검사

무결성 검사를 사용하면 활성 유니버스의 디자인이 정확하고 최신 상태인지 테스트할 수 있습니다.

무결성 검사는 다음 사항을 검색합니다.

- 유니버스 내의 개체, 조인, 조건, 카디널리티 등의 오류
- 조인 경로의 루프
- 필요한 컨텍스트
- 대상 데이터베이스에 대한 변경 사항

무결성 검사는 데이터베이스 요소를 기준으로 유니버스의 요소를 검사하기 전에 데이터베이스에 대한 연결이 유효한지 확인합니다. 연결이 유효하지 않으면 무결성 검사가 중지되고 오류 메시지가 반환됩니다.

5.9.2.1 무결성 검사로 검색할 수 있는 오류 유형

무결성 검사로는 다음 사항을 검색할 수 있습니다.

- 개체, 조건 또는 조인의 SQL 정의에 있는 잘못된 구문
- 루프
- 끊어진 테이블
- 끊어진 조인
- 컨텍스트 내의 루프
- 누락되거나 잘못된 카디널리티

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

연결된 데이터베이스의 변경 사항을 무결성 검사로 확인하는 방법

무결성 검사 기능은 데이터베이스에 테이블 목록 요청을 보냅니다. 그런 다음 이 목록을 유니버스 내의 테이블과 비교합니다. 열에 대해서도 동일한 작업이 수행됩니다.

데이터베이스의 테이블 목록과 일치하지 않는 모든 테이블이나 열은 존재하지 않는 항목으로 구조 창에 표시됩니다. 이들 테이블 또는 열은 데이터베이스에서 삭제되었거나 이름이 바뀌었을 수 있습니다. 무결성 검사를 사용하여 유니버스의 무결성 확인 [페이지 174] 단원을 참조하십시오.

> **노트**
>
> 데이터 양이 많으면 카디널리티 검사를 실행하는 데 시간이 많이 걸릴 수 있습니다. 또한 데이터가 정확하지 않거나 손실되었을 때는 결과가 정확하지 않을 수 있습니다. 따라서, 데이터베이스 크기가 크고 완전하지 않은 데이터 항목이 포함되어 있는 경우에는 카디널리티 검사 옵션을 선택하지 않는 것이 좋습니다. 그래도 이 옵션을 사용하려면 PRM 파일을 수정하여 카디널리티 검색을 최적화할 수 있습니다. 자세한 내용은 자동 카디널리티 검색 최적화 [페이지 170] 단원을 참조하십시오.

5.9.2.2 연결된 데이터베이스의 변경 사항을 무결성 검사로 확인하는 방법

무결성 검사 기능은 데이터베이스에 테이블 목록 요청을 보냅니다. 그런 다음 이 목록을 유니버스 내의 테이블과 비교합니다. 열에 대해서도 동일한 작업이 수행됩니다.

데이터베이스의 테이블 목록과 일치하지 않는 모든 테이블이나 열은 존재하지 않는 항목으로 구조 창에 표시됩니다. 이들 테이블 또는 열은 데이터베이스에서 삭제되었거나 이름이 바뀌었을 수 있습니다. 유니버스 구조 새로 고치기 [페이지 240] 단원을 참조하십시오.

> **노트**
>
> 데이터 양이 많으면 카디널리티 검사를 실행하는 데 시간이 많이 걸릴 수 있습니다. 또한 데이터가 정확하지 않거나 손실되었을 때는 결과가 정확하지 않을 수 있습니다. 따라서, 데이터베이스 크기가 크고 완전하지 않은 데이터 항목이 포함되어 있는 경우에는 카디널리티 검사 옵션을 선택하지 않는 것이 좋습니다. 그래도 이 옵션을 사용하려면 PRM 파일을 수정하여 카디널리티 검색을 최적화할 수 있습니다. 자세한 내용은 자동 카디널리티 검색 최적화 [페이지 170] 단원을 참조하십시오.

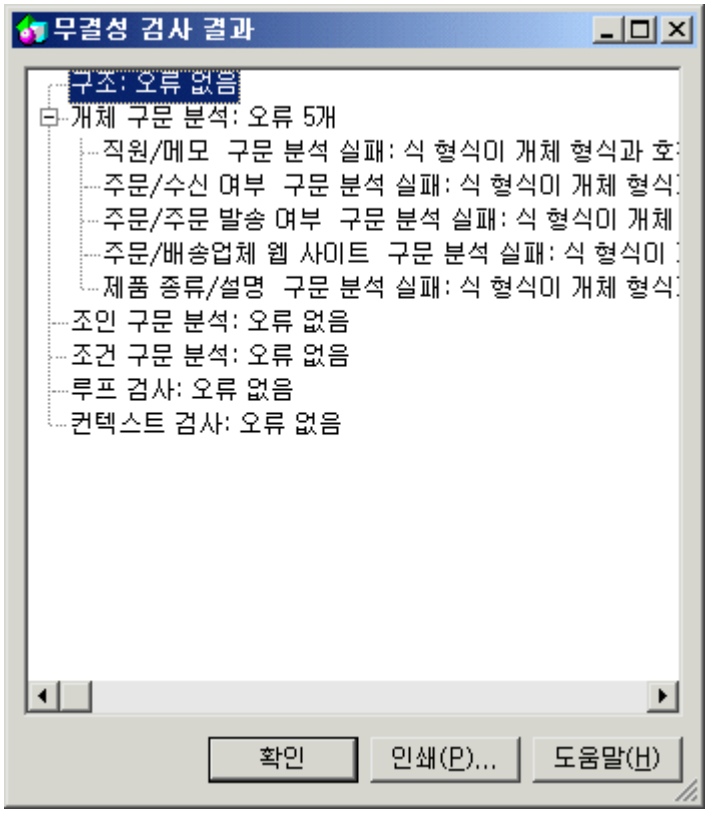
5.9.2.3 무결성 검사를 사용하여 유니버스의 무결성 확인

유니버스 무결성을 검사하려면

1. 도구 > 무결성 검사를 선택합니다.
   또는
   무결성 검사 단추를 클릭합니다.
   무결성 검사 대화 상자가 나타납니다.
2. 검사할 구성 요소에 해당하는 확인란을 선택합니다.

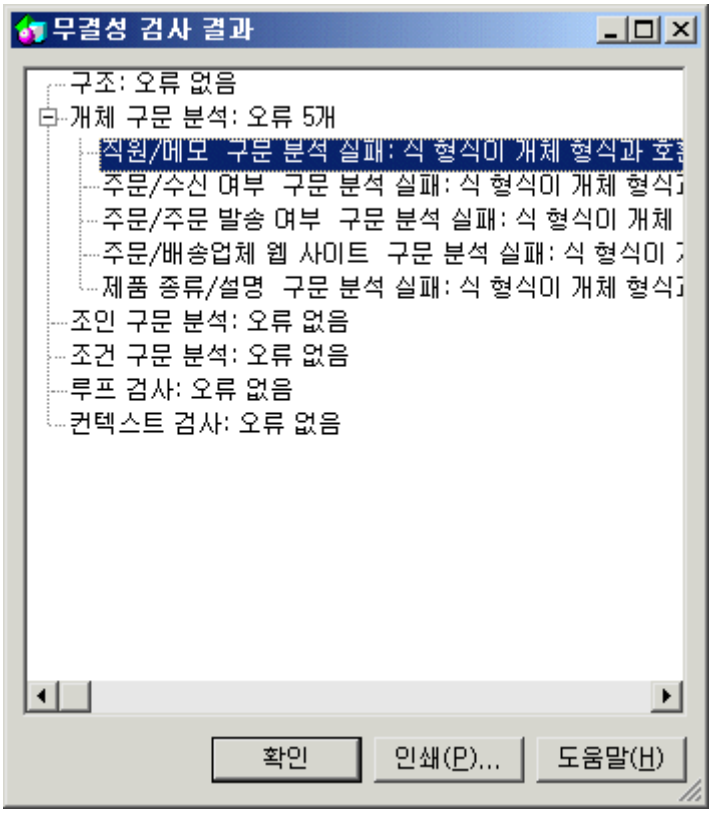
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. 검사하지 않을 구성 요소에 해당하는 확인란을 선택 취소합니다.
4. 빠른 구문 분석 확인란을 선택하여 구성 요소의 구문만 확인합니다.
   또는
   자세한 구문 분석 확인란을 선택하여 구성 요소의 구문과 의미를 모두 확인합니다.
5. 확인을 클릭합니다.
   유니버스 검사 진행률을 보여 주는 메시지 상자가 나타납니다.



   오류가 발견되지 않으면 각 오류 유형 옆에 "확인"이 표시됩니다.
6. 오류 유형 옆에 있는 더하기 기호(+)를 클릭하여 오류가 발생한 구성 요소 목록을 표시합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



목록에서 항목을 두 번 클릭하면 구조 창에 해당 구성 요소가 강조 표시됩니다.

7. 인쇄 단추를 클릭하여 창 내용을 인쇄합니다.
8. 확인을 클릭합니다.

> **노트**
>
> 루프 검사 확인란을 선택하려면 먼저 조인의 카디널리티를 검색해야 합니다. 그렇지 않으면 조인에서 루프를 식별할 때 오류가 발생합니다.

5.9.3 유니버스 구조 새로 고치기

무결성 검사를 실행한 결과 유니버스에 연결되어 있는 데이터베이스가 수정된 것으로 확인되면 구조 새로 고침을 사용하여 구조 창의 내용을 업데이트할 수 있습니다.

구조 새로 고침을 사용하면 다음과 같은 데이터베이스 변경 사항에 따라 유니버스 구조를 수정할 수 있습니다.

표 129:

조건	수행되는 작업
테이블에 열 추가	해당하는 유니버스 테이블에 열을 추가합니다.
테이블에서 열 제거	삭제해야 할 열과 관련 조인을 알려주는 경고 메시지를 표시합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조건	수행되는 작업
데이터베이스에서 테이블 제거	삭제해야 할 테이블과 관련 조인을 알려주는 경고 메시지를 표시합니다.
데이터베이스에서 테이블 이름 변경	해당하는 유니버스 테이블을 인식할 수 없다는 메시지를 표시합니다. 데이터베이스의 테이블과 일치하도록 이들 테이블의 이름을 바꿔야 합니다. 바꾼 이름도 일치하지 않으면 유니버스 디자인 도구에서는 이름이 바뀐 테이블이 데이터베이스에 없음을 알리는 메시지를 반환합니다.
데이터베이스 변경 사항 없음	유니버스를 업데이트하지 않아도 된다는 메시지를 표시합니다.

5.9.3.1 유니버스 새로 고치기

유니버스 구조를 새로 고치려면

- 보기 > 구조 새로 고침을 선택합니다.
  데이터베이스 변경 사항을 알려주거나 변경 사항이 없는 경우 업데이트하지 않아도 된다는 사실을 알려주는 메시지 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6 유니버스 만들기

스키마를 만들고 무결성 검사를 수행하여 루프 문제를 해결한 경우 보고 도구에 사용될 유니버스를 만들 수 있습니다.

6.1 개요

이 장에서는 Web Intelligence 사용자가 쿼리를 실행하고 보고서를 만들 때 사용할 수 있는 클래스와 개체를 만드는 방법에 대해 설명합니다. 여기에서는 개체 정의를 최적화하여 최종 사용자 보고서 작성 기능을 향상시키는 방법과 유니버스 최적화에 대한 내용도 다룹니다.

이전 장에서는 유니버스를 계획하고 테이블, 열 및 조인 같은 유니버스의 데이터베이스 구조가 포함된 테이블 스키마를 만드는 방법과 조인 경로에서 루프를 해결하는 방법에 대해 설명했습니다.

이렇게 만든 스키마는 Web Intelligence 사용자가 볼 수 없습니다. 데이터베이스 구조가 완성된 후에는 유니버스 창에서 표시되고 데이터베이스 구조에 대해 쿼리를 실행하여 문서와 보고서를 만드는 데 사용할 클래스와 개체를 만들 수 있습니다.

6.2 유니버스 작성 소개

유니버스 작성은 유니버스 개발 과정 중에서 개체를 만드는 단계입니다. 개체를 만들 때는 사용자 요구 사항에 대한 조사 결과에 기반해야 하며 조인 경로 문제가 테스트된 완전한 스키마 디자인을 사용해야 합니다.

다음 목록은 일반적인 유니버스 개발 주기에서 작성 및 테스트 단계의 위치(구현, 2 단계)를 나타냅니다.

- 준비

1. 사용자 요구 사항 분석
2. 계획

- 구현

1. 스키마 디자인 및 테스트
2. 유니버스 개체 작성 및 테스트
3. 리포지토리를 사용하여 유니버스 배포

- 유지 관리

1. 사용자 요구 사항 또는 데이터 소스의 변경 내용을 기반으로 유니버스 업데이트 및 유지 관리

6.2.1 개체란

Business Objects 제품에서 개체는 유니버스에 있는 명명된 구성 요소로 데이터베이스의 열이나 함수를 나타냅니다.

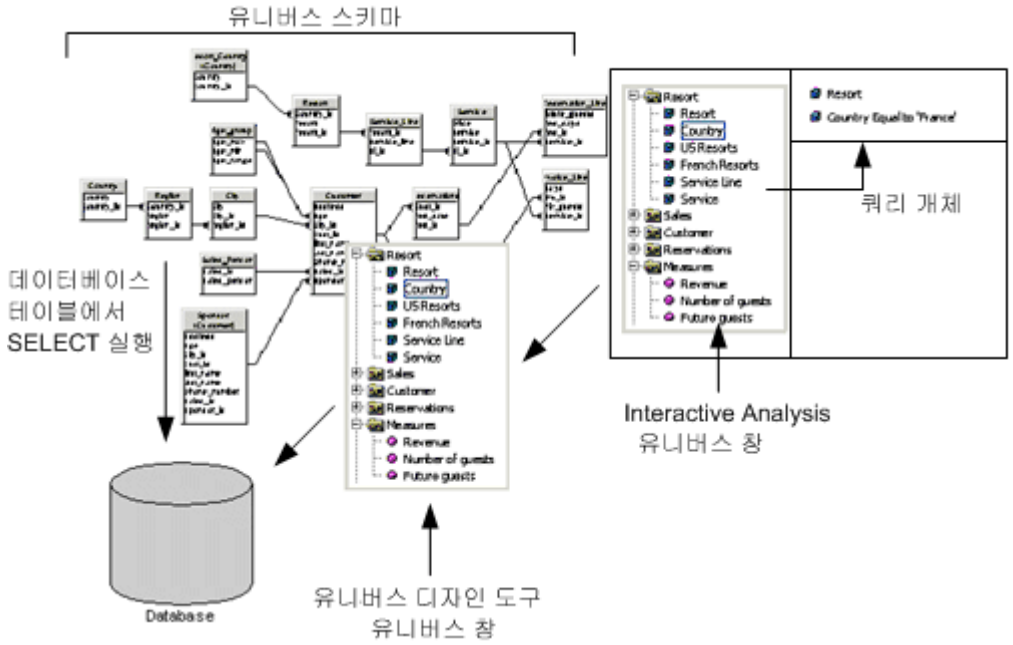
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

개체는 유니버스 창에 아이콘으로 표시됩니다. 각 개체는 최종 사용자의 비즈니스 환경에 사용되는 의미 있는 엔터티, 팩트 또는 계산을 나타냅니다. 유니버스 디자인 도구의 유니버스 창에 만드는 개체는 최종 사용자가 보고 도구에서 확인하고 사용하는 개체입니다. 유니버스 디자인 도구에서만 사용할 개체를 만들 수도 있는데, *Web Intelligence* 사용자에게 보여지는 유니버스 창에서 해당 개체를 숨기면 됩니다.

Web Intelligence 사용자는 유니버스 창에서 쿼리 창으로 개체를 끌어와서 쿼리를 실행한 후 반환된 데이터로 보고서를 만듭니다.

각 개체는 대상 데이터베이스에 있는 열이나 함수에 매핑되며 쿼리 창에서 사용될 경우에는 SELECT 문을 유추합니다. 개체를 여러 개 조합하면 데이터베이스에 대해 SELECT 문이 실행되며 SELECT 문에는 각 개체에서 유추된 SQL 이 포함되고 기본 WHERE 절이 적용됩니다.

아래 그림은 Web Intelligence 유니버스 창과 유니버스 디자인 도구 유니버스 창에 동일한 개체가 표시된 상태입니다. 유니버스 디자인 도구 유니버스 창에 있는 각 개체는 유니버스 스키마의 열에 매핑되고 쿼리에 사용될 경우 SELECT 문을 유추합니다.



유니버스 디자이너는 유니버스 디자인 도구를 사용하여 Web Intelligence 사용자가 쿼리 창에서 포함시킨 쿼리를 실행하는 데 사용할 개체를 만듭니다.

6.2.2 유니버스에 사용되는 개체 유형

다음과 같은 세 가지 유형 중 하나로 개체의 자격을 지정할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 130:

개체 자격	예제	설명
차원	Resort Country Service Line	쿼리에서 분석의 핵심입니다. 차원은 쿼리의 핵심인 데이터베이스 열이나 함수 하나 이상에 매핑됩니다.
설명	Customer Age Phone Number Address	차원에 대한 설명 데이터를 제공합니다. 설명 개체는 차원에 항상 연결되어 있습니다. 이 개체는 차원과 관련된 세부 정보를 제공하는 데이터베이스 열이나 함수 하나 이상에 매핑됩니다.
계수	Revenue Number of guests Future guests	데이터베이스의 통계에 매핑되는 집계 함수를 포함합니다.

개체를 만들 때는 쿼리 내에서의 역할에 기반하여 개체의 자격을 지정해야 합니다. 이 역할은 쿼리 창에서 개체를 사용할 때 유추되는 Select 문을 결정합니다.

6.2.3 클래스 및 개체 사용

개체가 나타내는 정보를 Web Intelligence 사용자에게 익숙한 방식으로 표시하기 위해 유니버스 창 안에 클래스와 개체를 함께 구성해야 합니다.

6.2.4 클래스란?

클래스는 개체 컨테이너입니다. 클래스는 Windows 환경의 폴더와 같습니다. 유니버스에서 공통적인 용도로 사용되는 개체를 보관하기 위해 클래스를 만듭니다.

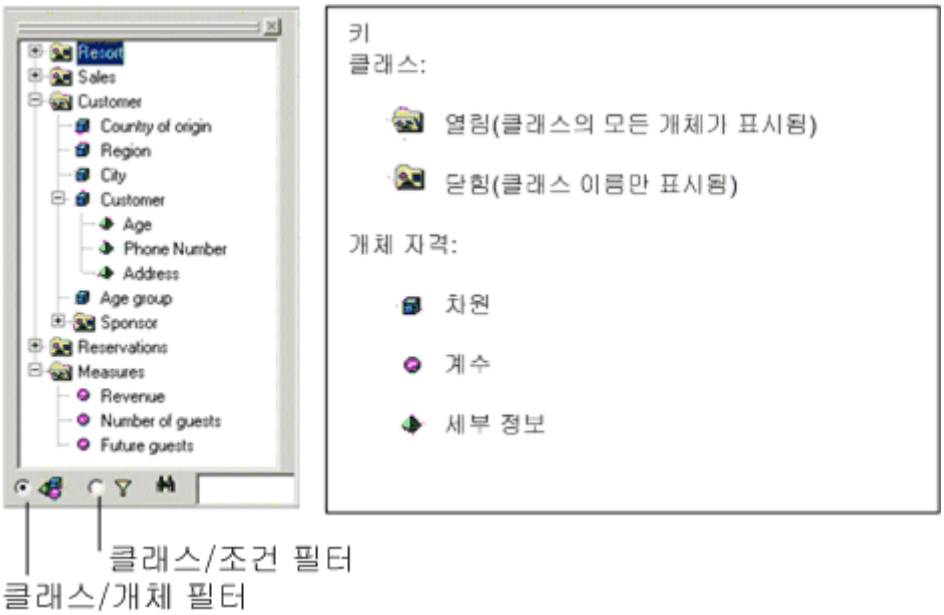
6.3 유니버스 창 사용

유니버스 창을 사용하여 유니버스에 클래스와 개체를 만듭니다.

유니버스 창에는 활성 유니버스의 클래스와 개체가 계층 뷰로 표시됩니다. 유니버스 창을 사용하여 클래스와 개체를 보고, 만들며, 편집 및 구성할 수 있습니다.

다음 그림은 유니버스 창을 보여 줍니다. 클래스 이름은 폴더 아이콘 옆에 표시되고 개체 이름은 해당 자격 기호 옆에 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



6.3.1 클래스와 개체 또는 조건 표시

창 아래쪽에 있는 두 개의 라디오 단추를 사용하면 유니버스 창에 클래스와 개체 또는 조건 개체를 표시할 수 있습니다. 조건 개체는 하나 이상의 Select 문에서 사용할 수 있는 미리 정의된 Where 절입니다.

유니버스 창을 두 가지 뷰로 표시할 수 있습니다.

표 131:

뷰	뷰 표시 방법	뷰 내용
클래스/개체	왼쪽 라디오 단추 선택	클래스 및 개체 전체
클래스/조건	오른쪽 라디오 단추 선택	클래스 및 각 클래스의 개체에 적용된 조건 전체

관련 정보

개체에 대한 제한 정의 [페이지 273]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.4 클래스, 개체 및 조건에 대한 기본 작업

유니버스 창에서는 클래스, 개체 및 조건에 대해 다음과 같은 공통적인 작업을 수행할 수 있습니다.

6.4.1 잘라내기, 복사, 붙여넣기

Windows 환경에서 일반적으로 사용하는 표준 명령을 사용하여 특정 구성 요소에 대해 잘라내기, 복사 및 붙여넣기를 할 수 있습니다.

6.4.2 클래스, 개체 또는 조건 이동

끌어서 놓는 방법으로 창 안에서 구성 요소를 원하는 위치로 이동할 수 있습니다.

6.4.3 클래스, 개체 및 조건 표시 또는 숨기기

유니버스 창에서 구성 요소를 하나 이상 숨길 수 있습니다. 이러한 구성 요소의 경우 Web Intelligence 사용자에게는 숨겨지지만 유니버스 디자인 도구에서는 계속 표시됩니다.

다음과 같은 경우 최종 사용자가 볼 수 없도록 개체를 숨기면 유용합니다.

- 연결된 유니버스에서 구성 요소를 가져오고 활성 유니버스에는 구성 요소가 필요하지 않은 경우
- SQL 구문을 최적화하는 데만 사용되는 개체를 최종 사용자가 볼 수 없도록 숨겨야 하는 경우
- 현재 개발 중인 구성 요소를 최종 사용자가 쿼리 창에서 보지 못하게 하려는 경우
- 구성 요소를 삭제하지 않고 일시적으로 비활성화하려는 경우

6.4.3.1 클래스, 개체 또는 조건 숨기기

클래스, 개체 또는 조건을 숨기려면

1. 유니버스 창에서 구성 요소를 클릭합니다.
2. 편집 > 항목 숨기기 를 선택합니다.
   또는
   편집 도구 모음에서 표시/숨기기 단추를 클릭합니다.
   유니버스 창에 구성 요소 이름이 기울임꼴로 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.4.3.2 숨겨진 클래스, 개체 또는 조건 표시

숨겨진 구성 요소 이름은 기울임꼴로 표시됩니다.

숨겨진 클래스, 개체 또는 조건을 표시하려면

1. 유니버스 창에 숨겨진 구성 요소를 클릭합니다.
2. 편집 > 항목 표시 를 선택합니다.
   구성 요소 이름이 더 이상 기울임꼴로 표시되지 않습니다. 이제 최종 사용자가 해당 구성 요소를 볼 수 있습니다.

6.5 클래스 정의

클래스는 하나 이상의 개체를 포함하는 컨테이너입니다. 유니버스의 각 개체는 클래스에 반드시 포함되어야 합니다. 클래스는 관련된 개체를 그룹화하는 데 사용됩니다. 클래스를 사용하면 최종 사용자가 특정 개체를 쉽게 찾을 수 있습니다. 클래스를 새로 만들거나 기존 클래스의 속성을 편집할 수 있습니다. 유니버스 창의 트리 계층구조에서 클래스는 폴더로 표시됩니다.

> **팁**
>
> 클래스를 사용할 때는 관련 차원 개체와 설명 개체를 클래스 하나에 그룹화하고 계수 개체를 별도의 클래스에 두는 것이 좋습니다. 그룹화된 관련 개체는 하위 클래스를 사용하여 더 작은 하위 집합으로 나누어 구성할 수 있습니다. 하위 클래스는 하위 클래스 사용 [페이지 249] 단원에서 설명합니다.

6.5.1 클래스 만들기

유니버스 창에서 다음과 같은 두 가지 방법으로 클래스를 만들 수 있습니다.

- 수동으로 클래스 정의
- 테이블 스키마에서 유니버스 창으로 테이블을 끌어와서 자동으로 만들기

두 방법은 아래에 설명되어 있습니다.

6.5.1.1 수동으로 클래스 만들기

유니버스 창 내에서 클래스를 수동으로 만들 수 있습니다. 사용자 요구 사항을 분석한 후 사용할 개체를 나열하고 클래스로 그룹화한 경우, 최종 사용자의 필요에 맞는 유니버스 구조를 만들려면 개체 목록에 기반하여 클래스를 수동으로 만드는 방법이 가장 좋습니다.

빈 유니버스 창에서 클래스를 만들려면

1. 삽입 > 클래스를 선택합니다.
   또는
   클래스 삽입 단추를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

클래스 속성 상자가 나타납니다.
2. 클래스 이름 텍스트 상자에 이름을 입력합니다.
3. 설명 텍스트 상자에 클래스에 대한 설명을 입력합니다.
4. 확인을 클릭합니다.
명명된 새 클래스 폴더가 유니버스 창에 표시됩니다.

➡ 팁

확인 단추 대신 적용 단추를 클릭하면 속성 상자가 열려 있는 상태에서 클래스 이름과 설명이 적용됩니다. 다른 클래스를 만들면 같은 상자 안에 새 클래스의 속성을 입력할 수 있습니다. 이렇게 하면 단일 속성 상자를 사용하여 클래스를 여러 개 만들 수 있습니다. 이 방법을 사용하면 클래스를 만들 때마다 새 속성 상자를 표시하지 않아도 되므로 클릭 횟수와 작업 시간을 줄일 수 있습니다.

6.5.1.2 기존 클래스가 있는 유니버스 창에서 클래스 만들기

기존 클래스를 사용하여 클래스를 만들려면

1. 트리 뷰에서 새 클래스 앞에 나올 클래스를 클릭한 다음 삽입 > 클래스를 선택합니다.
또는
트리 뷰에서 새 클래스 앞에 나올 클래스를 클릭한 다음 클래스 삽입 단추를 클릭합니다.
클래스 속성 상자가 나타납니다.
2. 새 클래스의 이름과 설명을 입력합니다.
3. 확인을 클릭합니다.
명명된 새 클래스 폴더가 유니버스 창에 표시됩니다.

6.5.1.3 테이블 스키마로부터 자동으로 클래스 만들기

테이블 스키마에서 테이블을 선택한 다음 유니버스 창으로 끌어와 클래스를 자동으로 만들 수 있습니다. 기본적으로 테이블 이름이 클래스 이름으로 사용됩니다. 이 경우 클래스 아래에 새 개체도 자동으로 만들어집니다. 새 개체 각각은 테이블의 열에 해당합니다.

새 클래스와 개체가 이름이 적절하고 최종 사용자의 필요에 적합하도록 그 속성을 편집해야 합니다. 개체 속성 편집은 개체 정의 [페이지 250] 단원에서 설명합니다.

클래스와 개체가 자동으로 만들어지는 방법은 유니버스 매개 변수 대화 상자의 전략 페이지에서 선택한 개체 전략에 따라 결정됩니다(파일>매개 변수>전략 탭). 이 전략은 수정 가능합니다. 전략을 만들어 클래스 및 개체 만들기 과정을 사용자 지정할 수도 있습니다. 전략에 대한 자세한 내용은 외부 전략을 사용하여 유니버스 만들기 사용자 지정 [페이지 365] 및 전략 선택 [페이지 81] 단원을 참조하십시오.

ℹ 노트

클래스와 개체를 자동으로 만드는 경우에는 데이터베이스 구조 그대로 유니버스 구성 요소가 만들어집니다. 데이터베이스의 테이블과 열을 그대로 사용하는 것보다는 사용자 요구 사항에 대한 분석 결과를 토대로 클래스와 개체를 만들어야 합니다. 사용자 요구 사항에 따라 유니버스를 디자인하는 방법은 유니버스 디자인 방법 [페이지 21] 단원에서 설명합니다.

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

테이블 스키마로부터 클래스를 자동으로 만들려면

1. 테이블 스키마에서 테이블을 선택합니다.
2. 테이블을 유니버스 창으로 끌어와 클래스 계층구조의 원하는 위치에 놓습니다.
   새 클래스가 계층구조에 표시됩니다. 이 클래스에는 유니버스 창으로 끌어온 테이블의 각 열에 해당하는 개체가 들어 있습니다. 기본적으로 클래스 이름은 테이블 이름과 동일하고 각 개체의 이름은 해당하는 열 이름과 동일합니다.

6.5.2 클래스 속성

정의할 수 있는 클래스 속성은 다음과 같습니다.

표 132:

속성	설명
이름	특수 문자를 포함할 수 있습니다. 이름은 유니버스 내에서 고유해야 합니다. 클래스 이름은 대소문자를 구분합니다. 이 이름은 언제든지 바꿀 수 있습니다.
설명	클래스를 설명하는 내용입니다. 이 설명은 쿼리 창에서 사용자가 볼 수 있습니다. 이 필드의 정보는 사용자의 비즈니스 언어로 표현해야 하며 쿼리 요구 사항과 관련되어야 합니다. Ctrl + Enter 를 누르면 줄을 바꿀 수 있습니다.

6.5.3 클래스 수정

클래스 이름과 설명은 클래스 속성 대화 상자에서 언제든지 수정할 수 있습니다. 다음 방법 중 하나를 사용하여 클래스 속성 대화 상자에 액세스할 수 있습니다.

- 클래스 폴더를 두 번 클릭합니다.
- 클래스 폴더를 마우스 오른쪽 단추로 클릭한 다음 편집 > 클래스 속성을 선택합니다.
- 클래스 폴더를 클릭한 다음 편집 > 클래스 속성을 선택합니다.

> 노트
>
> 클래스 폴더나 클래스 이름에서 위의 클릭 동작 중 하나를 수행하여 클래스 속성 대화 상자에 액세스할 수 있습니다.

6.5.4 하위 클래스 사용

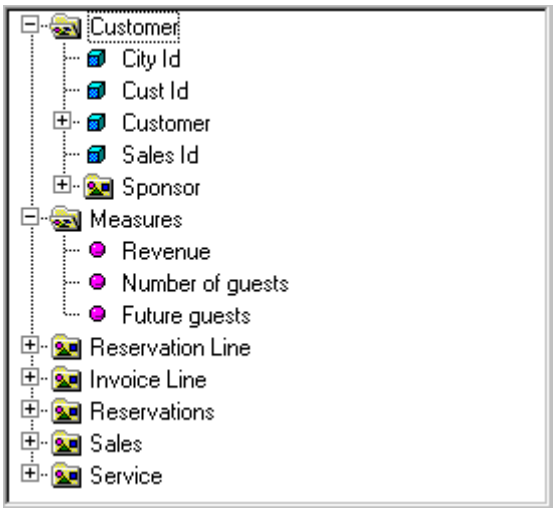
하위 클래스는 클래스 내의 클래스입니다. 하위 클래스는 서로 관련된 개체 그룹을 구성할 때 유용합니다. 하위 클래스 자체에도 다른 하위 클래스나 개체가 포함될 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.5.4.1 하위 클래스 만들기

하위 클래스를 만들려면

- 클래스 폴더나 클래스 이름을 클릭한 다음 삽입 > 하위 클래스를 선택합니다.
- 클래스 폴더나 이름을 마우스 오른쪽 단추로 클릭한 다음 상황에 맞는 메뉴에서 하위 클래스 삽입을 선택합니다.
  다음 그림은 유니버스 창에서 고객 클래스 아래의 후원자 하위 클래스를 보여 줍니다.



6.6 개체 정의

개체는 유니버스 데이터베이스 스키마에서 하나 이상의 테이블에 있는 하나 이상의 열에 매핑되는 유니버스 구성 요소입니다. 개체는 하나 이상의 열에 정의되어 있는 함수에 매핑될 수도 있습니다.

각 개체는 매핑되는 대상 열이나 함수에 대해 Select 문을 유추합니다. Web Intelligence 사용자가 쿼리 창에서 하나 이상의 개체를 사용하여 쿼리를 작성하면 각 개체가 나타내는 열이나 함수에 기반하여 Select 문에 사용할 Select 절의 내용이 유추됩니다.

6.6.1 개체 만들기

유니버스 창에서 개체를 만듭니다. Web Intelligence 사용자가 이름 및 자격으로 개체를 식별합니다. 개체를 만들 때는 유니버스 창에서 수동으로 만들거나 적절한 데이터베이스 구조를 구조 창에서 유니버스 창으로 끌어와 자동으로 만들 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.6.1.1 수동으로 개체 만들기

유니버스 창에 개체를 삽입한 다음 속성을 정의하는 방법으로 개체를 수동으로 만들 수 있습니다. 개체는 반드시 클래스에 속해 있어야 합니다.

개체를 수동으로 만들려면

1. 유니버스 창에서 개체를 마우스 오른쪽 단추로 클릭한 다음 상황에 맞는 메뉴에서 개체 삽입을 선택합니다.
   또는
   클래스를 클릭하고 개체 삽입 도구를 클릭합니다.
   선택한 클래스 아래에 개체가 삽입되고 해당 개체의 속성 편집 상자가 나타납니다.
2. 이름 상자에 이름을 입력합니다.
   개체 이름은 항상 최종 사용자의 비즈니스 용어로 지정해야 합니다. 이 이름은 개체와 관련된 데이터베이스 스키마 열의 실제 이름과 다를 수 있습니다.
3. 속성 탭을 클릭한 다음 개체 속성을 선택합니다.
4. Select 상자에 Select 문을 입력하거나 Select 단추를 클릭하여 SQL 편집기를 사용합니다.
5. 확인을 클릭합니다.

관련 정보

SQL 편집기 사용 [페이지 265]
개체 속성 [페이지 252]

6.6.1.2 자동으로 개체 만들기

구조 창에서 테이블의 열을 선택한 다음 유니버스 창으로 끌어와 개체를 자동으로 만들 수 있습니다. 끌어온 열을 놓은 지점과 가장 가까운 클래스 아래에 개체가 만들어집니다. 열 이름이 개체의 기본 이름으로 사용됩니다. 이때 밑줄은 공백으로 모두 바뀝니다. 개체의 기본 데이터 형식은 열의 기본 형식에서 파생됩니다. 개체의 속성 편집 시트에 있는 드롭다운 목록에서 새 데이터 형식을 선택하여 이 값을 변경할 수 있습니다.

새 개체가 이름이 적절하고 최종 사용자의 필요에 적합하도록 그 속성을 편집해야 합니다. 개체 속성 편집은 개체 정의 [페이지 250] 단원에서 설명합니다.

클래스와 개체가 자동으로 만들어지는 방법은 유니버스 매개 변수 대화 상자의 전략 페이지에서 선택한 개체 전략에 따라 결정됩니다(파일>매개 변수>전략 탭). 이 전략은 수정 가능합니다. 전략을 만들어 클래스 및 개체 만들기 과정을 사용자 지정할 수도 있습니다.

전략 사용에 대한 자세한 내용은 외부 전략을 사용하여 유니버스 만들기 사용자 지정 [페이지 365] 및 전략 선택 [페이지 81]을 참조하십시오.

> **노트**
>
> 클래스와 개체를 자동으로 만드는 경우에는 데이터베이스 구조 그대로 유니버스 구성 요소가 만들어집니다. 데이터베이스의 테이블과 열을 그대로 사용하는 것보다는 사용자 요구 사항에 대한 분석 결과를 토대로 클래스와 개체를 만들어야 합니다. 사용자 요구 사항에 따라 유니버스를 디자인하는 방법은 유니버스 디자인 방법 [페이지 21] 단원에서 설명합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

개체를 자동으로 만들려면

1. 구조 창에서 테이블 열을 클릭합니다.
2. 열을 유니버스 창으로 끌어와 클래스 계층구조의 원하는 위치에 놓습니다. 끌어온 열은 기존 클래스 아래에 놓아야 합니다.
   새 개체가 계층구조에 표시됩니다. 기본적으로 개체 이름은 열 이름과 동일합니다.

개체 이름은 항상 최종 사용자의 비즈니스 용어로 지정해야 합니다. 이 이름은 개체와 관련된 데이터베이스 스키마 열의 실제 이름과 다를 수 있습니다.

6.6.2 개체 속성

선택한 개체의 속성 편집 대화 상자에서 다음과 같은 개체 속성을 정의할 수 있습니다.

표 133:

속성 편집 페이지	속성
정의 사용 가능한 개체 정의 속성에 대한 자세한 내용은 개체 정의 [페이지 253]를 참조하십시오.	• 이름 • 데이터 형식 • 설명 • Select 문 • Where 절 이 페이지에서는 SQL 편집기에 액세스하여 SELECT 및 WHERE 구문을 정의할 수 있습니다.
속성 사용 가능한 개체 속성에 대한 자세한 내용은 속성 [페이지 256]을 참조하십시오.	• 개체 자격 • 관련 값 목록
고급 사용 가능한 고급 개체 속성에 대한 자세한 내용은 고급 [페이지 257]을 참조하십시오.	• 보안 • 개체에 대한 사용자 권한 • 날짜 형식
키 개체의 인덱스 인식 정의에 대한 내용은 인덱스 인식 정의 [페이지 258]를 참조하십시오.	• 키 형식 • 삽입을 • Where • 사용
소스 정보 이 탭 사용에 대한 내용은 소스 정보 [페이지 262]를 참조하십시오.	• 기술 정보 • 매핑 • 계보

개체 속성은 언제든지 수정할 수 있습니다. 위에 나열된 각 개체 속성은 각 속성 편집 페이지와 함께 개체 수정 [페이지 253] 단원에서 자세히 설명합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.6.3 개체 수정

개체 속성은 개체를 만들 때 정의하거나 언제든지 수정할 수 있습니다. 개체 속성을 정의할 때는 해당 개체의 속성 편집 대화 상자(개체를 마우스 오른쪽 단추로 클릭 > 개체 속성)를 사용해야 합니다. 속성 편집 대화 상자의 각 페이지에서는 다음과 같은 속성을 정의할 수 있습니다.

6.6.4 개체 정의

다음 그림은 정의 페이지를 보여 줍니다.

속성 편집 대화 상자의 정의 페이지에서는 다음과 같은 속성을 정의할 수 있습니다.

표 134:

속성	설명	필수/선택
이름	개체 이름입니다. 이름에는 특수 문자와 공백을 포함한 영숫자 문자를 사용할 수 있습니다. 이름은 대소문자를 구분합니다. 개체 이름은 클래스 내에서 고유해야 합니다. 서로 다른 클래스에 있는 개체는 같은 이름을 가질 수 있습니다.	필수

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

속성	설명	필수/선택
형식	개체 데이터 형식입니다. 다음 네 가지 형식 중 하나일 수 있습니다. • 문자 • 날짜 • 긴 텍스트 • 숫자 유니버스 디자인 도구의 현재 버전에서는 BLOB 가 지원되지 않습니다.	필수
설명	개체의 설명입니다. 이 필드는 쿼리 창에 표시되므로 최종 사용자에게 유용한 개체 정보를 입력하는 것이 좋습니다. Ctrl + Enter 를 누르면 포인터가 다음 줄로 이동합니다.	선택적
삽입을	개체에서 유추되는 Select 문입니다. SQL 편집기를 사용하여 Select 문을 만들 수 있습니다. 속성 [페이지 256] 단원을 참조하십시오.*	필수
Where	개체에서 유추되는 Select 문의 Where 절입니다. Where 절은 쿼리에서 반환되는 행 수를 제한합니다. SQL 편집기를 사용하여 Where 절을 만들 수 있습니다.*	선택적

* Select 문 또는 Where 절에서 @Prompt 를 삽입하거나 편집할 수 있습니다. Select 문 또는 Where 절을 마우스 오른쪽 단추로 클릭하면 바로 가기 메뉴가 표시됩니다. 문에 @Prompt 가 없는 경우 이 메뉴에 새 *@Prompt* 가 표시되고 @Prompt 가 있는 경우 이 @Prompt 를 클릭하면 *@Prompt* 편집이 표시됩니다. *@Prompt* 편집기가 열립니다.

테이블 단추

테이블 단추를 클릭하면 스키마에 사용된 테이블 목록이 나타납니다. 이 목록에서 다른 테이블의 다른 열을 선택하여 개체 정의에 포함시킬 수 있습니다. 이렇게 하면 개체에서 Select 문에 여러 테이블의 열을 유추할 수 있습니다. 자세한 내용은 여러 테이블을 유추하여 제한 적용 [페이지 283] 단원을 참조하십시오.

구문 분석 단추

구문 분석 단추를 클릭하면 개체의 Select 문이 구문 분석됩니다. 구문 오류가 발견되면 오류를 설명하는 메시지 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

SQL 편집기 사용 [페이지 265]
OLAP 유니버스의 계산된 계수 [페이지 411]
@Prompt 편집기 [페이지 344]

6.6.4.1 개체 정의 편집

개체 정의를 편집하려면

1. 개체를 두 번 클릭합니다.
   속성 편집 대화 상자의 정의 페이지가 열립니다.
2. 개체 정의 및 속성을 원하는 대로 입력하거나 선택합니다.
3. 확인을 클릭합니다.

6.6.4.2 동적 하이퍼링크로 개체 정의

셀의 텍스트를 하이퍼링크로 정의할 수 있습니다. 이 방법은 열의 셀 텍스트가 결과 개체에 따라 특정 리소스에 대해 하이퍼링크가 되는, 보고서의 동적 하이퍼링크에 유용합니다.

개체의 select 문을 편집하여 하이퍼링크 선언을 포함하고 개체의 Format 속성에 대해 하이퍼링크로 읽기를 선택합니다.

6.6.4.3 개체를 동적 하이퍼링크로 정의

이 방법을 사용하면 개체가 결과 보고서에 동적 하이퍼링크를 만듭니다.

1. 개체를 마우스 오른쪽 단추로 클릭하고 개체 속성을 선택합니다.
   개체 속성 편집 대화 상자가 표시됩니다.
2. select 문을 입력하고 적합한 하이퍼링크를 포함합니다.
3. 새 속성을 저장합니다.
4. 개체를 마우스 오른쪽 단추로 클릭하고 개체 서식을 선택합니다.
   개체 서식 창이 표시됩니다.
5. 하이퍼링크로 읽기 옵션을 선택합니다.
6. 확인을 클릭하여 서식 설정을 저장합니다.
7. 보고 도구를 사용하여 보고서를 만들고 링크를 테스트합니다.

보고서에 개체가 사용된 경우 결과 열에 하이퍼링크가 포함됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

예

달력 정보에 대한 하이퍼링크 사용

다음 Select 문은 열 셀의 연도 값에 따라 timeanddate.com 에서 연간 달력 정보를 검색합니다. 선언에서 오른쪽 끝에 있는 4 개의 문자(연도)를 가져오고 대상 URL 에서 인식하지 못하는 'FY'(회계 연도)를 문자열에서 제거합니다.

```
'<a href=http://www.timeanddate.com/calendar/?year=>' +right(@Select(Reservations
\Reservation Year),4) +'</a>'
```

6.6.5 속성

속성 편집 대화 상자의 속성 페이지에서는 값 목록에 대해 다음과 같은 개체 자격과 속성을 지정할 수 있습니다.

표 135:

속성	설명
자격	쿼리 창에서 사용할 경우 개체가 수행하도록 정의된 역할입니다. 다음과 같은 세 가지 유형 중 하나로 개체의 자격을 지정할 수 있습니다. • 차원 • 설명 • 계수 개체 자격에 대한 자세한 설명은 유니버스에 사용되는 개체 유형 [페이지 243] 단원을 참조하십시오.
값 목록 연결	이 확인란을 선택하면 데이터 값이 포함된 파일을 개체와 연결할 수 있습니다. 이 옵션은 기본적으로 활성화됩니다. 자세한 내용은 값 목록 사용 [페이지 295] 단원을 참조하십시오.

6.6.5.1 개체 자격 및 값 목록 속성 지정

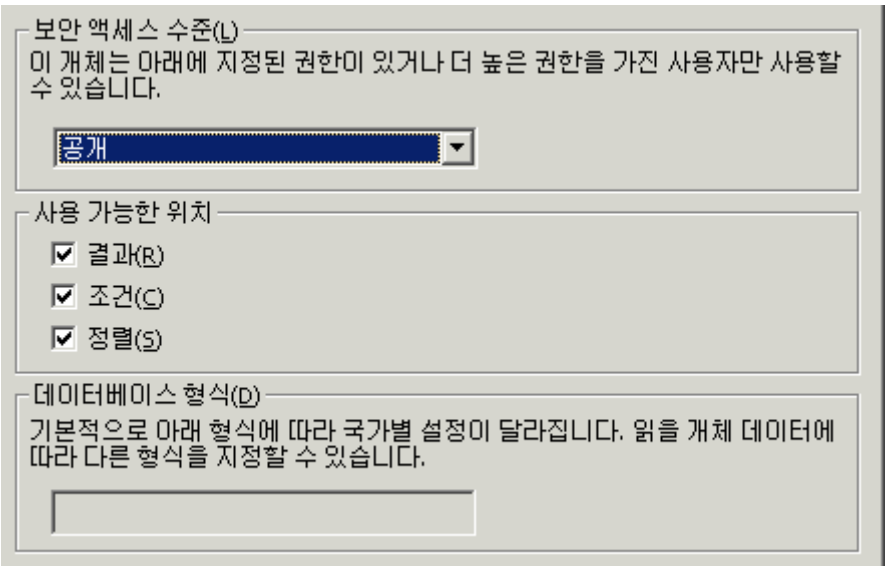
개체의 자격 및 값 목록 속성을 지정하려면

1. 개체를 두 번 클릭합니다.
   개체의 속성 편집 상자가 나타납니다.
2. 속성 탭을 클릭합니다.
   속성 페이지가 나타납니다.
3. 자격 라디오 단추 중 하나를 클릭하여 개체의 자격을 차원, 설명 또는 계수 중 하나로 지정합니다.
   반환 값 목록을 개체에 연결하려면 값 목록 연결 확인란을 선택합니다.
   값 목록 만들기 및 사용에 대한 내용은 값 목록 사용 [페이지 295]?단원을 참조하십시오.
4. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.6.6 고급

다음 그림은 고급 페이지를 보여 줍니다.



속성 편집 대화 상자의 고급 페이지에서는 다음과 같은 속성을 정의할 수 있습니다.

표 136:

속성	설명
보안 액세스 수준	개체의 보안 액세스 수준을 정의합니다. 적절한 보안 수준이 지정된 사용자만 개체를 사용할 수 있도록 제한하는 보안 수준을 선택할 수 있습니다. 다음과 같은 보안 액세스 수준을 지정할 수 있습니다. • 공개 • 제어 • 제한 • 기밀 • 개인 공개로 지정하면 모든 사용자가 개체를 보고 사용할 수 있습니다. 제한으로 지정하면 사용자 프로필이 제한 이상인 사용자만 개체를 보고 사용할 수 있습니다.
사용 가능한 위치 - 결과	이 옵션을 선택하면 개체를 쿼리에 사용할 수 있습니다.
다음에 사용할 수 있음 - 조건	이 옵션을 선택하면 조건을 설정하는 데 개체를 사용할 수 있습니다.
다음에 사용할 수 있음 - 정렬	이 옵션을 선택하면 반환 값을 정렬할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

속성	설명
데이터베이스 형식	이 옵션은 날짜 개체에 대해서만 사용할 수 있습니다. 기본적으로 개체의 날짜 형식은 MS Windows 제어판의 국가별 설정 속성 대화 상자에 정의됩니다. 이 날짜 형식을 수정하면 대상 데이터베이스 형식으로 날짜를 저장할 수 있습니다. 예를 들어, 미국 형식 또는 유럽 형식을 사용할 수 있습니다. 이 값 수정에 대한 내용은 개체 서식 정의 [페이지 265] 단원을 참조하십시오.

6.6.6.1 개체 보안 및 사용자 권한 정의

개체의 보안 및 사용자 권한을 정의하려면

1. 개체를 두 번 클릭합니다.
   개체의 속성 편집 상자가 나타납니다.
2. 고급 탭을 클릭합니다.
   고급 페이지가 나타납니다.
3. 보안 액세스 수준 드롭다운 목록 상자에서 보안 액세스 수준을 선택합니다.
4. 다음에 사용할 수 있음 그룹 상자 아래의 확인란 중 하나 이상을 선택합니다.
5. 기본 날짜 형식을 수정하려면 데이터베이스 형식 텍스트 상자에 날짜 형식을 입력합니다.
6. 확인을 클릭합니다.

6.6.7 인덱스 인식 정의

키 탭을 사용하면 개체에 대한 인덱스 인식을 정의할 수 있습니다. 인덱스 인식은 키 열의 인덱스를 활용하여 데이터 검색 속도를 향상시킬 수 있는 기능입니다.

유니버스 디자인 도구에서 만드는 개체는 최종 사용자에게 의미 있는 데이터베이스 열을 기반으로 합니다. 예를 들어, 고객 개체는 고객 이름이 들어 있는 필드를 검색합니다. 이 경우, Customer 테이블에는 최종 사용자에게는 의미가 없지만 데이터베이스 성능에는 매우 중요한 기본 키(예: 정수)가 일반적으로 있습니다. 유니버스 디자인 도구에서 인덱스 인식을 설정한 경우 기본 키인 데이터베이스 열과 외래 키인 데이터베이스 열이 식별됩니다. 이는 다음과 같은 방법으로 쿼리 성능에 큰 영향을 줄 수 있습니다.

- 유니버스 디자인 도구는 키 열의 인덱스를 활용하여 데이터 검색 속도를 향상시킬 수 있습니다.
- 유니버스 디자인 도구는 가장 효율적인 방식으로 필터링을 수행하는 SQL 을 생성할 수 있습니다. 이것은 별모양 스키마 데이터베이스에서 특히 중요합니다. 차원 테이블의 값을 필터링하는 관련 쿼리를 작성하는 경우 유니버스 디자인 도구는 차원 테이블 외래 키를 사용하여 팩트 테이블에 필터를 직접 적용할 수 있습니다. 이렇게 하면 차원 테이블에 대한 불필요한 조인을 줄일 수 있습니다.

유니버스 디자인 도구는 인덱스 인식에서 중복 항목을 무시하지 않습니다. 이름이 같은 두 명의 고객이 있으면 유니버스 디자인 도구에서는 각 고객의 기본 키가 다르다는 점을 인식하지 못할 경우 한 명의 고객만 검색합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

예

도시 목록에서 고객 찾기

이 예제에서는 Island Resorts Marketing Universe 에 대해 휴스턴, 댈러스, 샌프란시스코, 샌디에이고 또는 로스앤젤레스에 있는 고객별로 수익을 반환하는 보고서를 작성합니다. 먼저 고객 및 판매 수익 개체를 쿼리 창의 결과 개체 창으로 끌어온 다음 도시 개체를 조건 창으로 끌어와 앞에 언급한 도시 목록으로 제한합니다.

인덱스 인식을 사용하지 않을 경우 유니버스 디자인 도구에서는 다음과 같은 SQL 을 생성합니다.

```
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
```

이 경우 유니버스 디자인 도구에서는 검색 대상 도시를 제한하기 위해 City 테이블에 대한 조인을 만들었습니다.

인덱스 인식을 사용할 경우 유니버스 디자인 도구에서는 `city_id` 가 City 테이블의 기본 키이고 이 키가 Customer 테이블의 외래 키로도 사용된다고 인식합니다. 이 정보를 사용하여 유니버스 디자인 도구에서는 City 테이블을 조인하지 않고도 도시를 제한할 수 있습니다. SQL 은 다음과 같습니다.

```
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
```

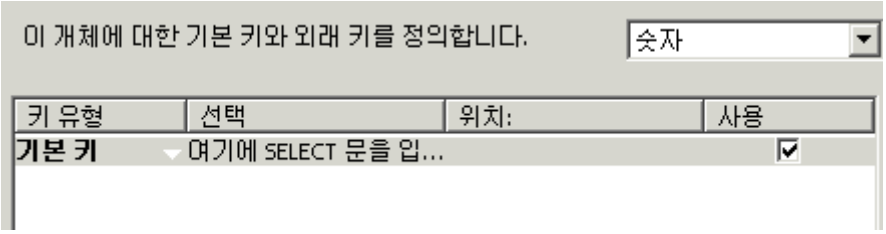
이 경우 유니버스 디자인 도구에서는 간단하게 `city_id` 외래 키를 필터링하여 도시를 제한하는 SQL 을 생성할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

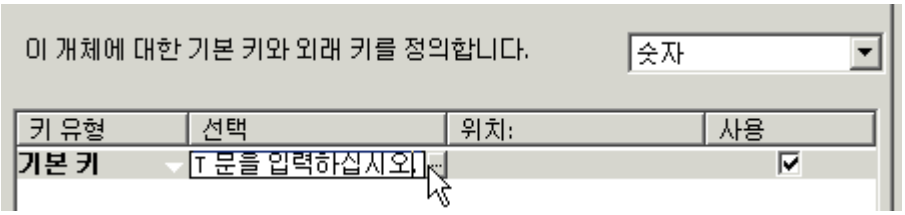
6.6.7.1 기본 키 인덱스 인식 설정

기본 키 인덱스 인식을 설정하려면

1. 인덱스 인식을 설정할 개체를 마우스 오른쪽 단추로 클릭한 다음 메뉴에서 개체 속성을 선택합니다.
   속성 편집 대화 상자가 나타납니다.
2. 키 탭을 클릭합니다.
3. 삽입을 클릭합니다.
   다음 그림과 같이 키 페이지에 기본 키 줄이 삽입됩니다.

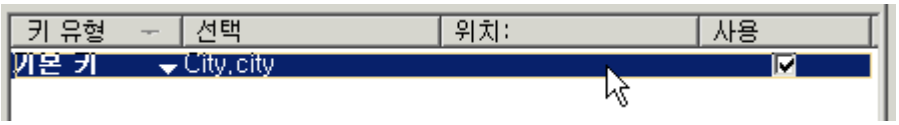


4. 기본 키에 대한 키 인식을 만들려면 다음 작업을 수행합니다.
   - 키 형식 목록에서 기본을 선택합니다.
     Select 필드에 있는 ... 단추를 클릭하여 SQL 편집 대화 상자를 엽니다.



   SQL 편집기가 나타납니다.
   - SQL 편집기를 사용하여 기본 키 SQL SELECT 절을 작성하거나 직접 입력합니다. 예를 들어, 위의 도시 개체에 대한 기본 키 SQL 은 `City.city_id` 입니다.

   SQL 편집기에 대한 자세한 내용은 SQL 편집기 사용 [페이지 265]을 참조하십시오.
   - 키 형식 드롭다운 목록에서 기본 키 데이터 형식을 선택합니다.
5. WHERE 절을 추가하려면 다음 작업을 수행합니다.
   - 아래 그림과 같이 Where 열 아래에서 줄 안쪽을 클릭합니다.



   - Where 필드에 있는 ... 단추를 클릭하여 SQL 편집 대화 상자를 엽니다.

   SQL 편집기가 나타납니다.
   - SQL 편집기를 사용하여 기본 키 SQL WHERE 절을 작성하거나 직접 입력합니다. 위 예제에는 Where 절이 없습니다.
   - 키 형식 드롭다운 목록에서 숫자를 선택합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6. 사용을 선택합니다.
7. 확인을 클릭합니다.

> **노트**
>
> 기본 키에 대해 둘 이상의 열을 추가하려면 여러 열을 연결하여 기본 키를 정의할 수 있습니다. 이러한 열은 같은 테이블에 속해야 하며 같은 데이터 형식이어야 합니다.
>
> 즉, 샘플 데이터베이스 "`club.mdb`"에서 Resort 테이블은 `Country_id` 및 `Resort_id` 를 기반으로 하는 여러 열 기본 키를 갖습니다.
>
> 따라서 <resort> 개체에 대해 인덱스 인식을 정의하기 위해 "`Country_id`" 및 "`Resort_id`"를 연결하여 기본 키를 정의할 수 있습니다.
>
> `Resort.country_id & Resort.resort_id`
>
> &는 MS Access 연결 연산자입니다.

6.6.7.2 외래 키 인식 설정

외래 키 인식을 설정하려면

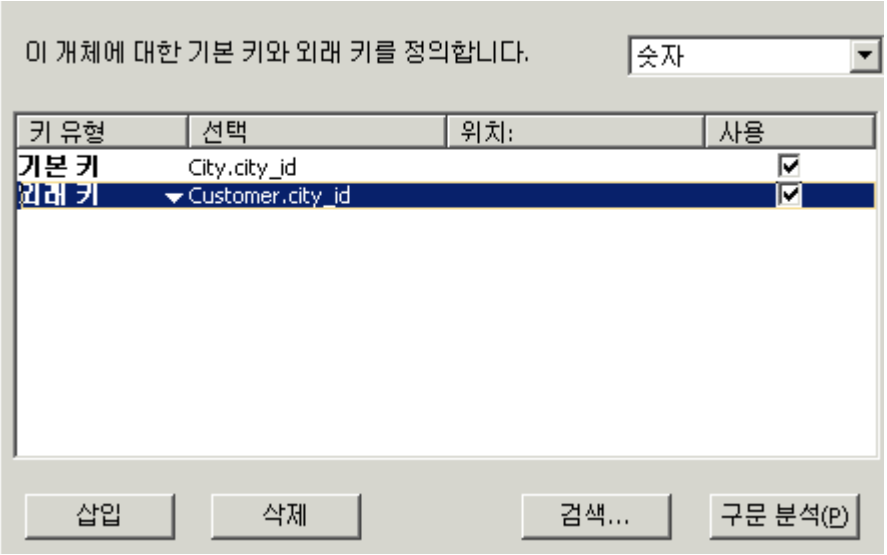
1. 인덱스 인식을 설정하려는 개체를 마우스 오른쪽 단추로 클릭합니다.
   메뉴에서 개체 속성을 선택합니다.
   속성 편집 대화 상자가 나타납니다.
2. 키 탭을 클릭합니다.
3. 삽입을 클릭합니다.
   키 페이지에 키 줄이 삽입됩니다.
4. 외래 키에 대한 키 인식을 만들려면 다음 작업을 수행합니다.
   - 키 형식 목록에서 외래 키를 선택합니다.
   - Select 필드에 있는 ... 단추를 클릭하여 SQL 편집 대화 상자를 엽니다.

   SQL 편집기가 나타납니다.
   - SQL 편집기를 사용하여 외래 키 SQL SELECT 절을 작성하거나 직접 입력합니다.
   - 키 형식 드롭다운 목록에서 외래 키 데이터 형식을 선택합니다.
5. 외래 키를 구성하는 모든 열에 대해 3-4 단계를 반복합니다.
6. WHERE 절을 추가하려면 다음 작업을 수행합니다.
   - Where 열 아래에서 강조 표시된 줄을 클릭합니다.
   - Where 필드에 있는 ... 단추를 클릭하여 SQL 편집 대화 상자를 엽니다.

   SQL 편집기가 나타납니다.
   - SQL 편집기를 사용하여 외래 키 SQL WHERE 절을 작성하거나 직접 입력합니다.
   - 키 형식 드롭다운 목록에서 숫자를 선택합니다.
7. 사용을 선택합니다.
8. 외래 키의 모든 열에 대해 위 단계를 반복합니다.

인덱스 인식 정의 단원에 나오는 예의 경우 키 탭은 다음과 같이 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



6.6.8 소스 정보

소스 정보 페이지는 Data Integrator 에서 생성된 유니버스에 사용됩니다. 소스 정보 탭은 다음과 같습니다.

Data Integrator 에서 생성된 유니버스의 경우 이 탭에는 기술적인 설명과 소스 테이블에서 대상 테이블을 계산하는 데 사용되는 수식이 표시됩니다. 이 정보는 Web Intelligence 사용자에게 표시됩니다.

소스 정보 탭에 다음과 같은 유형의 정보를 지정할 수 있습니다.

- 기술적인 설명: Data Integrator 에서 생성된 유니버스에서 사용할 수 있는 기술적인 설명입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 매핑 정보: Data Integrator 내에서 적용되는 소스 테이블과 대상 테이블 간의 매핑입니다. 목표는 매핑 식을 제공하는 것이 아니라 사용자에게 개체 정의에 사용되는 소스 열을 알리는 설명 주석으로 표시하는 것입니다.
- 데이터 계보 정보: 대상 열에 포함된 소스 열의 목록입니다. 이 정보를 사용하면 Data Integrator 및 Web Intelligence 보고서를 통해 영향 분석을 수행할 수 있습니다.

6.6.9 SQL 편집기를 사용하여 개체 정의

SQL 편집기를 사용하여 Select 문이나 개체의 Where 절을 정의하고 OLAP 유니버스 개체에 대한 MDX 연산자 및 함수를 삽입할 수 있습니다. SQL 편집기는 테이블, 열, 개체, 연산자 및 함수가 트리 뷰로 나열되는 그래픽 편집기입니다. 나열된 구조 중 원하는 항목을 두 번 클릭하면 Select 또는 Where 상자에 삽입됩니다.

SQL 편집기에서는 다음과 같은 편집 옵션을 사용할 수 있습니다.

표 137:

편집 옵션	설명
테이블 및 열	구조 창에 표시된 모든 테이블과 해당 열이 나열됩니다. **노트** 이 옵션은 관계형 유니버스에만 사용할 수 있고 OLAP 유니버스에는 사용할 수 없습니다.
클래스 및 개체	유니버스 창에 표시된 모든 클래스와 해당 개체가 나열됩니다.
연산자	Select 문에 SQL 구조를 조합하거나 Where 절에 조건을 설정하는 데 사용할 수 있는 연산자가 나열됩니다.
함수	• 숫자, 문자, 날짜 등의 데이터베이스 함수가 나열됩니다. • Business Objects 제품에서만 사용할 수 있는 @Functions 함수가 나열됩니다. 사용할 수 있는 함수는 대상 데이터베이스의 매개 변수 파일(.PRM)에서 Functions 항목 아래에 나열됩니다. 지원되는 데이터베이스마다 .PRM 파일이 있습니다. .PRM 파일은 BusinessObjects 경로의 Data Access 폴더에 저장되어 있습니다. .PRM 파일을 편집하면 사용 가능한 함수를 추가하거나 수정할 수 있습니다. .PRM 파일 편집에 대한 자세한 내용은 데이터 액세스 가이드를 참조하십시오.
개체 SQL 표시	이 확인란을 선택하면 Select 또는 Where 상자에 있는 개체의 SQL 구문이 표시됩니다.
구문 분석	이 단추를 클릭하면 구문 분석을 실행할 수 있습니다. 구문이 유효하지 않으면 문제를 설명하는 메시지 상자가 나타납니다.
설명	선택한 개체나 함수에 대한 설명이 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

큐브 쿼리의 MDX 함수 [페이지 264]
SQL 편집기 사용 [페이지 265]

6.6.9.1 큐브 쿼리의 MDX 함수

MDX 편집기를 사용하여 큐브 쿼리를 정의합니다.

OLAP 유니버스에 새 개체나 미리 정의된 필터를 추가할 때 특정 데이터 소스 연결에 대한 MDX 식 목록을 사용할 수 있습니다.

사용 가능한 식 라이브러리는 `.prm` 연결 파일에 저장됩니다. 개체의 속성 편집 창과 쿼리의 Select 문 편집 창을 열면 함수 창에 사용 가능한 식이 표시됩니다. SELECT 또는 WHERE 문에 식을 삽입하려면 문에서 식을 삽입할 위치를 클릭하고 삽입할 식을 더블 클릭합니다.

OLAP Universe MDX Dictionary - 함수 목록(PRM 파일)

OLAP 유니버스에 새 개체나 미리 정의된 필터를 추가할 때 식에서 사용할 수 있는 적절한 OLAP 연결(SAP 또는 MSAS)의 개체/필터 편집기에서 MDX 함수(주로 멤버 함수) 및 연산자의 명시적 목록을 사용할 수 있습니다. SAP 또는 mySQL(sap.prm, sqlsrv_as.prm) 연결을 설정하는 방법은 데이터 액세스 가이드를 참조하십시오. 유니버스의 연결 유형에 따라 사용 가능한 함수 및 연산자가 결정됩니다. 함수 목록은 각 연결에 대한 PRM 파일에서 제공합니다. PRM 파일에는 지원되는 함수 목록 중 가장 사용 빈도가 높은 함수만 표시됩니다.

다음 MDX 연산자를 쿼리에서 사용할 수 있습니다.

- Equal
- NotEqual
- InList
- NotInList
- Greater
- GreaterOrEqual
- Less
- LessOrEqual
- Between
- NotBetween
- Like
- NotLike

아래 목록에는 조건 편집 시 사용할 수 있는 몇 가지 MDX 폴더 함수가 나와 있습니다. 사용 가능한 함수는 기본 데이터베이스에 따라 결정됩니다.

- 집합 함수(ADDCALCULATEDMEMBERS, ALLMEMBERS ...)
- 통계/숫자 함수(AGGREGATE, AVG ...)
- 탐색/멤버 함수(ANCESTOR, ASCENDANTS...)
- 메타데이터 함수(AXIS, HIERARCHY...)

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.6.9.2 SQL 편집기 사용

SQL 편집기를 사용하여 개체 정의에 SQL 및 MDX 식을 삽입할 수 있습니다. Select 문을 마우스 오른쪽 단추로 클릭한 후 새 *@Prompt* 를 선택하여 SQL 에 @Prompt 식을 삽입하거나 *@Prompt* 편집을 선택하여 기존 @Prompt 식을 편집합니다. 그러면 @Prompt 편집기가 열립니다.

SQL 편집기를 사용하려면

1. 개체를 두 번 클릭합니다.
   개체에 대한 속성 편집 대화 상자가 열립니다.
2. Select 또는 Where 상자 옆에 있는 >> 단추를 클릭합니다.
   Select 문 편집 또는 Where 절 편집 대화 상자가 나타납니다.
3. Select 문이나 Where 절에서 구조의 구문을 추가할 위치를 클릭합니다. 상자가 비어 있으면 상자 안을 클릭합니다. 그러면 커서가 상자의 왼쪽 맨 위에 자동으로 표시됩니다.
4. 테이블 노드를 확장하여 열을 표시합니다.
5. 열을 두 번 클릭하여 Select 문이나 Where 절에 열 정의를 삽입합니다.

   팁

   선택한 열에 대해 값 목록에서 값을 하나 이상 선택하려면 열을 마우스 오른쪽 단추로 클릭한 다음 LOV(값 목록)를 선택합니다.

6. 클래스 노드를 확장하여 개체를 표시합니다.
7. 개체를 두 번 클릭하여 Select 문이나 Where 절에 @Select 또는 @Where 함수를 삽입합니다. 이들 함수는 선택한 개체의 Select 문이나 Where 절을 현재 개체에 사용하도록 지시합니다. @함수 사용에 대한 자세한 내용은 개체의 SQL 에서 @함수 사용 [페이지 339] 단원을 참조하십시오.
8. 연산자를 두 번 클릭하여 편집 상자에 삽입합니다.
9. 함수 노드를 확장하여 사용 가능한 함수를 표시합니다.
10. 함수를 두 번 클릭하여 편집 상자에 삽입합니다.
11. 구문 분석 단추를 클릭하여 구문의 유효성을 검사합니다.
12. 확인을 클릭합니다.

6.6.10 개체 서식 정의

선택한 개체의 데이터 값 서식을 정의할 수 있습니다. 서식은 Web Intelligence 보고서의 셀에 표시되는 관련 데이터 값에 적용됩니다.

개체 서식 대화 상자는 숫자, 정렬, 글꼴, 테두리 및 음영 설정 탭으로 구성됩니다.

예를 들어, 기본값인 1,000.00 대신 $1,000 등의 형식으로 정수를 표시할 수 있습니다. 또는 빨강 등의 색을 중요한 데이터 값에 적용할 수 있습니다.

숫자, 통화, 공학 및 백분율 범주는 숫자 형식의 개체와 변수에만 사용할 수 있으며, 날짜/시간 범주는 날짜 형식의 개체와 변수에만 사용할 수 있습니다.

유니버스를 내보내거나 가져올 때 서식 정보도 포함됩니다.

개체 서식 제거 명령을 사용하면 정의한 모든 서식을 제거할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

큐브 쿼리의 MDX 함수 [페이지 264]

6.6.10.1 개체 서식 수정

개체 서식을 수정하려면

1. 개체를 마우스 오른쪽 단추로 클릭합니다.
2. 상황에 맞는 메뉴에서 개체 서식을 선택합니다.
   개체 서식 시트가 나타납니다.
3. 원하는 서식 탭을 클릭한 다음 개체 서식을 선택하거나 입력합니다.
4. 확인을 클릭합니다.

6.6.10.2 개체 서식 제거

개체 서식은 언제든지 제거할 수 있습니다.

개체 서식을 제거하려면

- 개체를 선택한 다음 파일 > 서식 제거를 선택합니다.
  또는
- 개체를 마우스 오른쪽 단추로 클릭한 다음 상황에 맞는 메뉴에서 서식 제거를 선택합니다.

6.6.11 개체 정의에 사용된 테이블 보기

유니버스 창의 개체 정의에 사용된 테이블을 구조 창에서 볼 수 있습니다. 이 기능을 사용하면 개체 이름만으로 테이블을 식별할 수 없는 경우 해당 개체에서 사용하는 테이블을 신속하게 식별할 수 있습니다.

6.6.11.1 개체에서 사용하는 테이블 보기

개체에서 사용하는 테이블을 보려면

1. 유니버스 창에서 개체를 마우스 오른쪽 단추로 클릭합니다.
   상황에 맞는 메뉴가 나타납니다.
2. 상황에 맞는 메뉴에서 연결된 테이블 보기를 선택합니다.
   연결된 테이블이 구조 창에 강조 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.6.12 차원 정의

차원은 쿼리에서 분석의 핵심이 되는 개체입니다. 차원은 쿼리의 핵심인 데이터베이스 열이나 함수 하나 이상에 매핑됩니다. 예를 들어, 국가, 판매 직원, 제품 또는 판매 라인이 있습니다.

차원은 개체를 만들 때의 기본 자격으로 사용됩니다. 자격을 차원으로 언제든지 변경할 수 있습니다.

차원 개체를 정의하려면

1. 개체를 두 번 클릭합니다.
   개체에 대한 속성 편집 대화 상자가 열립니다.
2. 속성 탭을 클릭합니다.
   속성 페이지가 나타납니다.
3. 자격 그룹 상자에서 차원 라디오 단추를 선택합니다.
4. 확인을 클릭합니다.

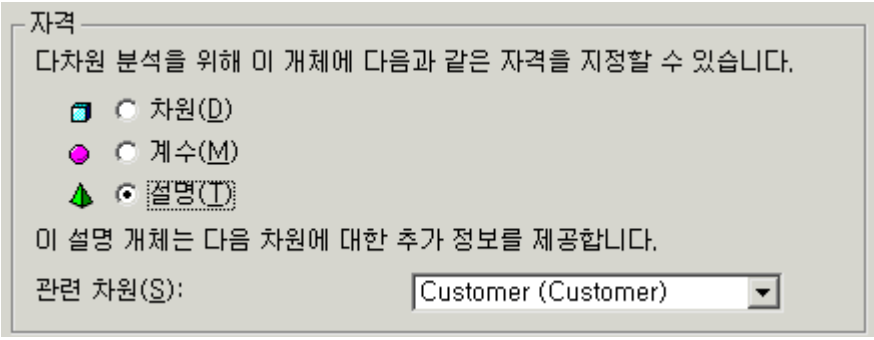
6.6.13 설명 정의

설명은 차원을 설명하는 데이터를 제공합니다. 설명 개체는 차원에 항상 연결되어 있습니다. 이 개체는 차원과 관련된 세부 정보를 제공하는 데이터베이스 열이나 함수 하나 이상에 매핑됩니다.

설명 개체를 정의하려면 개체 자격을 설명으로 설정하고 해당 설명과 연결된 차원을 지정해야 합니다.

설명 개체를 정의하려면

1. 개체를 두 번 클릭합니다.
   개체에 대한 속성 편집 대화 상자가 열립니다.
2. 속성 탭을 클릭합니다.
   속성 페이지가 나타납니다.
3. 자격 그룹 상자에서 설명 라디오 단추를 선택합니다.
   관련된 차원 드롭다운 목록 상자가 표시되고 그 안에 유니버스의 차원 개체가 모두 나열됩니다.
4. 드롭다운 목록 상자에서 차원을 선택합니다. 설명 개체는 선택한 차원의 자격이나 속성을 설명합니다.



5. 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.6.14 계수 정의

계수를 개체 자격으로 선택하여 계수 개체를 정의할 수 있습니다. 계수는 매우 유연하면서도 동적인 개체입니다. 계수 개체의 반환 값은 쿼리에 함께 사용된 차원 및 설명 개체에 따라 달라집니다. 예를 들어, 판매 수익 계수는 국가 개체와 함께 쿼리에 사용했을 때와 지역 및 국가 개체와 함께 다른 쿼리에 사용했을 때 서로 다른 값을 반환합니다.

계수 개체는 차원 개체나 설명 개체보다 복잡하고 강력하므로 다음 단원에서 더 자세하게 설명합니다.

6.6.14.1 계수 개체가 반환하는 정보 유형

계수 개체는 숫자 정보를 반환합니다. 집계 함수를 사용하여 계수를 만들 수 있습니다. 다음은 가장 일반적으로 사용하는 다섯 가지 집계 함수입니다.

- Sum
- Count
- Average
- Minimum
- 최대

6.6.14.2 계수 개체와 차원 및 설명 개체의 차이점

계수는 다음과 같은 점에서 차원 및 설명 개체와 다릅니다.

- 계수는 동적입니다.
- 계수는 집계를 프로젝션할 수 있습니다.

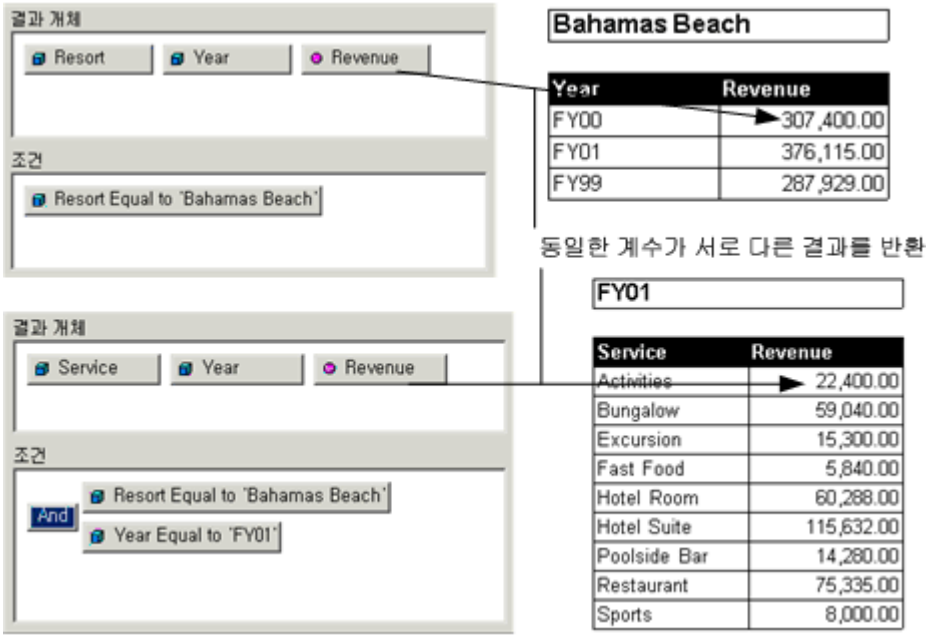
이 두 가지 속성은 아래에 설명되어 있습니다.

6.6.14.3 계수가 동적으로 동작하는 방법

계수 개체의 반환 값은 쿼리에 함께 사용된 차원 및 설명 개체에 따라 달라집니다.

다음 예제는 동일한 수익 계수 개체를 두 개의 쿼리에 각기 다른 차원과 함께 사용했을 때 계수에서 서로 다른 값을 반환하는 것을 보여 줍니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



6.6.14.4 Group By 절을 유추하는 계수

계수 개체와 다른 개체 유형이 포함되어 있는 쿼리를 실행하면 Select 문에 Group By 절이 자동으로 유추됩니다.

Group By 절은 다음과 같은 SQL 규칙에 따라 유추됩니다.

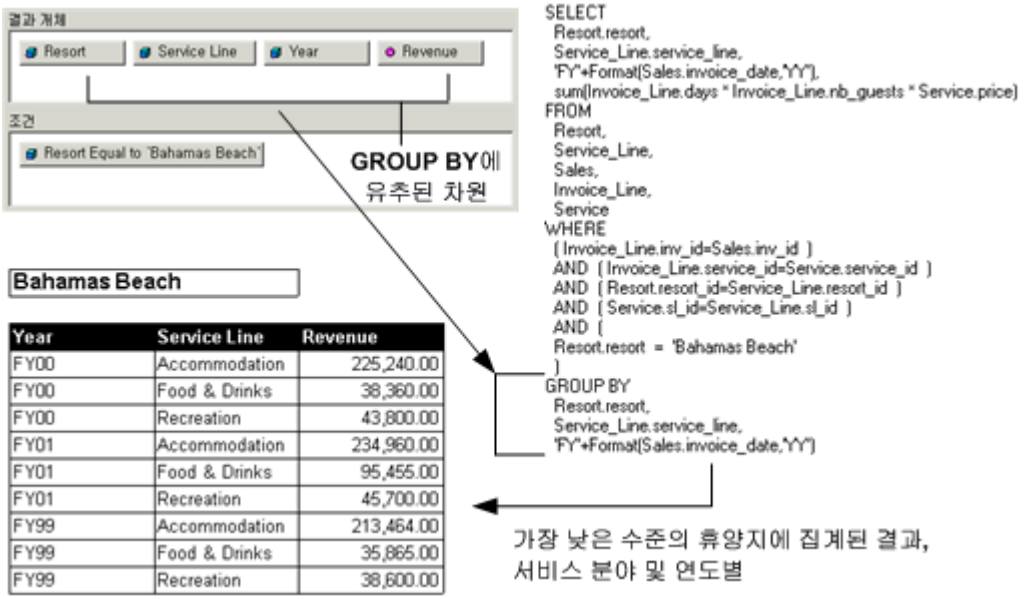
표 138:

Select 절 줄에 집계가 포함되어 있으면 그 절의 집계 밖에 있는 모든 항목이 Group By 절에도 포함되어야 합니다.

이 규칙에 따라, 쿼리에 계수 개체와 함께 사용된 모든 차원 또는 설명 개체는 자동으로 유추되는 Group By 절에 항상 포함됩니다. 쿼리에서 올바른 결과를 반환하려면 차원 및 설명 개체에는 집계가 포함되어 있으면 안 됩니다.

다음 예제는 휴양지, 서비스 분야 및 연도 차원 개체 모두 Select 절과 Group By 절에 유추되는 것을 보여 줍니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



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

노트

쿼리에 계수 개체만 있으면 Group By 절이 유추되지 않습니다.

6.6.14.5 계수의 집계 프로젝션 설정

계수를 만들 때는 집계 함수를 보고서에 프로젝션할 방법을 반드시 지정해야 합니다.

계수 개체의 반환 값은 쿼리 프로세스 중 두 단계에서 집계됩니다.

- 쿼리 단계. 유추된 SELECT 문을 사용하여 데이터가 집계됩니다.
- 마이크로큐브와 블록 사이의 단계. 마이크로큐브에서 보고서의 블록으로 데이터가 프로젝션될 때 집계됩니다. 이와 같은 계수 프로젝션 기능을 사용하면 마이크로큐브에 데이터를 로컬로 집계할 수 있습니다.

노트

마이크로큐브는 쿼리에서 반환한 데이터를 보고서에 프로젝션하기 전에 표시하는 개념적인 방식입니다. 마이크로큐브는 Business Objects 보고서 작성 제품의 메모리에 있는 반환 값을 나타냅니다. 블록 수준은 사용자가 반환된 데이터로 만드는 2 차원 보고서입니다. 사용자는 마이크로큐브에 있는 데이터 전체나 일부만을 사용하여 보고서를 만들 수 있습니다. 또는 마이크로큐브에 있는 반환 값(로컬 집계)에 대해 집계 함수를 적용하여 보고서에 새 값을 만들 수 있습니다.

다음 그림은 쿼리 프로세스의 두 가지 집계 단계를 보여 줍니다.

- 사용자가 Web Intelligence 에서 쿼리를 작성합니다.
- Web Intelligence 에서 쿼리에 기반하여 SQL 을 유추하고 대상 데이터베이스에 SELECT 문을 보냅니다.
- 마이크로큐브에 데이터가 반환됩니다. 이것이 첫 번째 집계 단계입니다.
- 마이크로큐브에서 집계 데이터를 보고서에 프로젝션합니다. 쿼리 창에 데이터가 분할되어 세부적인 집계가 필요합니다. 이것이 두 번째 집계 단계입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

쿼리를 처음 실행하면 Select 문의 결과 집합이 마이크로큐브에 저장된 다음 마이크로큐브에 보관된 모든 데이터가 블록에 프로젝션됩니다. 이 경우 데이터는 마이크로큐브의 최하위 수준에서 프로젝션되기 때문에 프로젝션 집계가 수행되지 않습니다.

그러나 쿼리 창을 사용하여 마이크로큐브의 데이터 중 일부만 프로젝션하는 경우에는 상위 수준의 계수 값을 표시하기 위해 집계를 수행해야 합니다.

예를 들어, 위의 예제에서 연도 데이터를 블록으로 프로젝션하지 않으면 휴양지의 총 판매 수익을 표시하기 위해 연도와 관련된 세 개의 행을 단일 행으로 줄어야 하므로 합계 집계를 사용해야 합니다.

프로젝션 집계는 계수의 속성 편집 시트에 있는 속성 페이지(개체를 마우스 오른쪽 단추로 클릭 > 개체 속성 > 속성)에서 설정합니다.

프로젝션 집계는 SELECT 집계와 다릅니다.

관련 정보

데이터베이스 위임 프로젝션 함수 [페이지 271]

6.6.14.5.1 데이터베이스 위임 프로젝션 함수

유니버스에서 모든 계수는 프로젝션 함수(*Sum*, *Min*, *Max*, *Count* 및 *Avg*)를 포함할 수 있습니다. 프로젝션 함수는 보고서에 표시된 차원 수가 쿼리 결과 집합의 차원 수보다 작을 때 Web Intelligence 에서 로컬로 계수를 집계하는 데 사용됩니다.

비율, 평균 및 가중치와 같은 비가산 계수는 쿼리 결과 집합과 같은 집계 수준에만 표시될 수 있습니다. 따라서 유니버스에서 비가산 계수의 프로젝션 함수는 일반적으로 없음으로 설정되어 있습니다.

프로젝션 함수 데이터베이스 위임을 사용하여 비가산 계수의 집계를 데이터베이스 서버로 위임할 수 있습니다. 이 기능을 Web Intelligence 에서는 스마트 계수라고 합니다. 스마트 계수의 프로젝션 함수는 개체 속성의 속성 페이지에서 데이터베이스 위임으로 설정되어 있습니다. 이 기능 및 Web Intelligence 에 사용된 다른 기능에 대한 자세한 내용은 *Using Functions, Formulas and Calculations in Web Intelligence* 문서의 *Calculating values with Smart Measures* 단원을 참조하십시오.

> **노트**
>
> MSAS 및 Essbase 데이터 소스를 기반으로 하는 OLAP 유니버스의 경우 모든 계수는 기본적으로 프로젝션 함수가 데이터베이스 위임으로 설정된 유니버스에 만들어집니다.

> **노트**
>
> 집계 인식 집합이 포함된 계수 기반의 스마트 계수를 사용할 경우 다음 제한에 대해 주의하십시오. 계수 정의에 사용된 집계 테이블의 데이터에 일관성이 있는지 확인하는 것이 좋습니다(세부 값 관련 집계 값이 정확). 그렇지 않으면 스마트 계수가 일관성이 없는 데이터를 만들 수 있습니다. 예를 들어, 연도 집계 테이블 및 일 집계 테이블이 특정 스마트 계수에 사용된 경우 연도 테이블이 현재 연도가 아닌 전체 연도에 대한 일 집계 테이블과 일치한다면 일 테이블의 데이터는 일 단위로 정확한 반면 연도 테이블이 비어있을 수 있습니다. 이 경우 현재 연도 및 일 단위 테이블 기반의 스마트 계수를 사용한 보고서의 경우 일관성이 없는 결과가 발생할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

예

스마트 계수

이 예의 쿼리에는 두 개의 차원인 국가 및 지역과 세 개의 계수인 주문 수량, 배송 수량 및 배송 %이 포함되어 있습니다.

L01 지역	배송 수량	주문 수량	배송 %
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
		합계:	299.992

배송 %의 합계는 배송 % 열의 합계이므로 정확하지 않습니다.

유니버스에서 이 계수의 프로젝션 함수가 데이터베이스 위임으로 설정된 경우 사용자가 보고서를 새로 고치면 Web Intelligence 는 해당 데이터베이스에 연결하여 올바른 값을 계산합니다.

L01 지역	배송 수량	주문 수량	배송 %
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
		합계:	299.992
		합계:	99.997

노트

비율 함수(Average) 등의 일부 함수를 사용할 때는 반드시 주의해야 합니다. 열 평균을 계산하는 경우 이 함수의 동작이 올바르게 구성되지 않으면 예기치 않은 결과가 발생할 수 있습니다.

예를 들어 SQL 함수 `sum(Shop_facts.Margin)/sum(Shop.facts.Quantity_sold)` 은 예기치 않은 결과가 발생할 수 있습니다. 올바르게 구성되지 않은 경우 각 셀에 대해 평균을 계산한 다음 해당 평균에 대한 합을 반환합니다. 이 동작을 수정하려면 다음과 같이 함수에 대한 매개변수화를 수행해야 합니다.

1. 함수에 대한 속성 편집 옵션으로 이동합니다.
2. 집계 시 이 계수를 처리할 방법을 선택하십시오 옵션에 대해 함수 드롭다운 목록에서 *Db delegated* 함수를 선택합니다.
3. 변경 사항을 저장합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

계수의 집계 프로젝션 설정 [페이지 270]

6.6.14.6 계수 만들기

계수를 만들려면

1. 개체를 두 번 클릭합니다.
   개체에 대한 속성 편집 대화 상자가 열립니다.
2. 속성 탭을 클릭합니다.
   속성 페이지가 나타납니다.
3. 자격 그룹 상자에서 계수 라디오 단추를 선택합니다.
   집계 함수가 나열된 함수 드롭다운 목록 상자가 나타납니다.
4. 함수를 선택합니다.
5. 확인을 클릭합니다.

관련 정보

계수의 집계 프로젝션 설정 [페이지 270]
데이터베이스 위임 프로젝션 함수 [페이지 271]

6.6.15 개체에 대한 제한 정의

제한은 쿼리의 반환 데이터를 제한하기 위해 SQL 안에 설정하는 조건입니다.

개체에 대해 제한은 사용자가 사용할 수 있는 데이터를 제한하기 위해 정의합니다. 사용자의 데이터 액세스를 제한할 때는 대상 사용자의 데이터 요구 사항에 기반해야 합니다. 예를 들면 개체에서 반환하는 값 중 일부만 사용자에게 필요할 수 있습니다. 또는 보안상의 이유로 인해 특정 값에 대한 사용자 액세스를 제한해야 할 수 있습니다.

유니버스에서는 두 가지 유형의 제한을 정의할 수 있습니다.

표 139:

제한 유형	설명
강제적	개체의 Where 절에 정의하는 제한입니다. 사용자가 액세스할 수 없으므로 Web Intelligence 에서 다시 정의할 수 없습니다.
선택적	특수 조건 개체에 정의되어 사용자가 쿼리에 선택적으로 사용할 수 있는 제한입니다. 조건 개체는 쿼리 창에서 작업할 때 개체에서 유추된 Select 문에 삽입할 수 있는 미리 정의된 Where 절입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

노트

사용자는 Web Intelligence 의 쿼리 창에서 조건을 적용할 수 있습니다. 유니버스 디자이너는 사용자 수준에서 쉽게 적용할 수 있는 선택적 제한은 되도록 만들지 않는 것이 좋습니다. 이러한 조건은 필요할 때 사용자가 직접 만들 수 있습니다.

6.6.15.1 개체의 Where 절 정의

개체에 대한 속성 편집 대화 상자의 정의 페이지에서 *Where* 상자에 조건을 추가하면 개체에 추가 제한을 적용할 수 있습니다.

조건은 개체를 만들 때 정의하거나 개체 정의에 언제든지 추가할 수 있습니다.

유니버스에서 다음과 같은 두 가지 방법으로 SQL 문의 Where 절을 사용하여 쿼리에서 반환하는 행 수를 제한할 수 있습니다.

- 스키마의 테이블을 연결하는 조인을 통해 개체의 SELECT 문에 WHERE 절이 자동으로 유추됩니다. 일반적으로 조인은 테이블 간의 조건이 동등하다는 것을 기반으로 합니다. 조인은 카디전 곱이 만들어지지 않도록 조인된 테이블에서 반환되는 데이터를 제한합니다.
- 개체의 WHERE 절에 조건을 추가합니다. 이는 조인에서 유추된 기존 WHERE 절의 추가 조건입니다. WHERE 절을 정의하면 데이터 하위 집합에 대한 쿼리 결과만 사용자가 볼 수 있도록 제한하려는 경우와 같이 쿼리에서 반환되는 데이터를 추가적으로 제한할 수 있습니다.

예

개체의 기본(조인에만 해당) Where 절 수정

아래 보고서는 모든 국적의 판매 직원 데이터가 포함된 제한 없는 블록입니다.

Sales Person	Country of origin
Barrot	France
Carlin	France
Edwood	UK
Fischer	Germany
Galagers	US
Ishimoto	Japan
Nagata	Japan

이 쿼리의 SQL 은 다음과 같습니다. 여기에서 Where 절에는 Customer, City, Region 및 Sales_Person 테이블 간의 조인에서 유추된 제한만 포함되어 있습니다.

```
SELECT
  Sales_Person.sales_person, Country.country
FROM
  Sales_Person,
  Country,
  Region,
  City,
  Customer
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

```
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  )
```

사용자가 프랑스에 대한 반환 값만 볼 수 있도록 제한하려면 국가 개체의 Where 절에 조건을 추가하면 됩니다. 아래의 보고서에는 프랑스 국적의 판매 직원만 표시됩니다.

Sales Person	Country of origin
Barrot	France
Carlin	France

쿼리 SQL 은 다음과 같습니다.

```
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 절에 줄을 추가합니다. 이 줄은 국가 개체의 WHERE 절에 추가한 제한입니다.

노트

자체 제한 조인 외에는 WHERE 절에 조인을 만들면 안 됩니다. WHERE 절 안에 있는 조인은 컨텍스트 검색(컨텍스트 자동 검색)이나 집계 인식 비호환성 검색 대상에서 제외됩니다. 구조 창에는 모든 조인이 표시되어야 합니다. 그래야만 유니버스 디자인 도구 자동 검색 도구에서 모든 조인을 대상으로 검색할 수 있습니다.

6.6.15.2 Where 절 정의

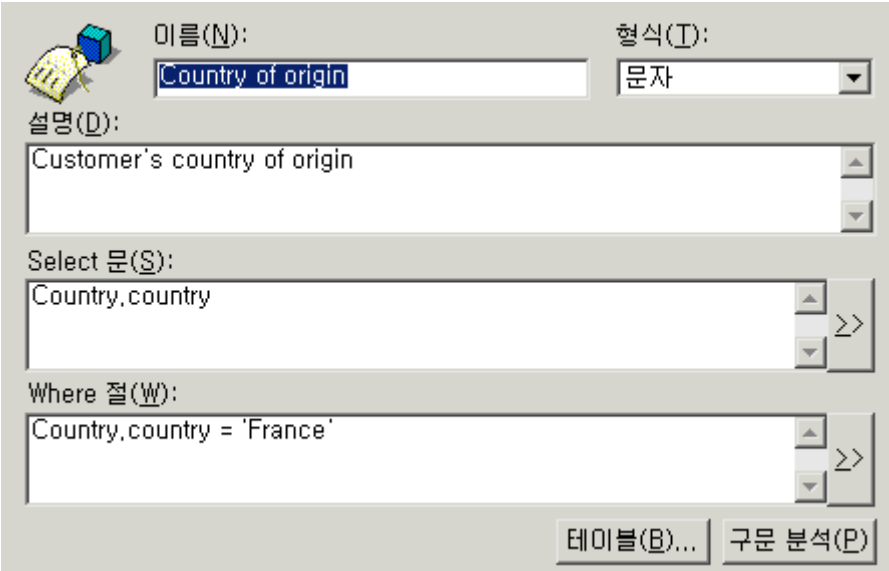
Where 절을 정의하려면

1. 개체를 두 번 클릭합니다.
   속성 편집 대화 상자의 정의 페이지가 열립니다.
2. Where 절 텍스트 상자에 구문을 직접 입력합니다.
   또는
   Where 상자 옆에 있는 >> 단추를 클릭하여 Where 절 편집기를 엽니다.
3. SQL 구조 및 기능 목록에 표시되는 열, 개체, 연산자 또는 함수를 두 번 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

> 팁
>
> Where 절의 값을 선택하려면 테이블 및 열 목록에서 열을 마우스 오른쪽 단추로 클릭한 다음 값 보기를 선택합니다. 그러면 선택한 열의 값 목록이 표시됩니다. 예를 들어, In 연산자를 사용하는 경우에는 Where 절에 삽입할 값을 하나 이상 선택합니다.

4. 확인을 클릭하여 편집기를 닫습니다.
   다음 그림은 국가 개체의 Where 절을 보여 줍니다. 여기에서는 프랑스 값만 표시되도록 국가를 제한합니다.



5. 확인을 클릭합니다.

6.6.15.3 Where 절을 사용할 때 발생할 수 있는 문제

Where 절은 데이터를 제한하는 데 유용하지만 유니버스에서 사용할 때 다음과 같은 문제가 발생하지 않도록 주의해야 합니다.

표 140:

문제	설명	해결 방법
유사한 개체의 확산	특정 개체의 데이터를 제한하기 위해 개체를 여러 개 만들었고, 이렇게 만든 각 개체가 데이터의 일부에 사용될 Where 절을 유추하는 경우에는 이름이 유사한 개체가 많아질 수 있습니다. 프랑스인 고객, 미국인 고객 및 일본인 고객을 예로 들 수 있습니다. 이렇게 유사해 보이는 개체가 여러 개 있으면 사용자에게 혼동을 줄 수 있습니다.	각 제한에 대해 조건 개체를 만듭니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

문제	설명	해결 방법
계층을 만들기 어려움	동일한 데이터에 대해 Where 절을 유추하는 개체가 여러 개 있으면 드릴다운하는 데 사용할 기본 논리 계층구조를 사용자가 쉽게 만들 수 없습니다.	각 제한에 대해 조건 개체를 만듭니다.
개체 이름과 적용된 제한 간의 혼동	개체 이름을 아주 명확하게 지정한 경우가 아니면 사용자가 제한과 개체 이름을 명확하게 구분하지 못할 수 있습니다. 쿼리의 SQL 을 보면 Where 절을 확인할 수 있지만 일부 사용자는 SQL 을 보지 않고 쿼리를 실행합니다.	• 각 제한에 대해 조건 개체를 만듭니다. • 각 개체의 이름을 적절하게 지정합니다.
Where 절 간의 충돌	유사하게 제한된 개체를 동일한 쿼리에 두 개 이상 사용하면 Where 절이 충돌하여 아무 데이터도 반환되지 않을 수 있습니다.	각 제한에 대해 조건 개체를 만들고 사용자가 보고서 수준에서 쿼리를 통합하거나 동기화하도록 합니다.

조건 개체를 만들면 여러 개체 문제, 계층구조 문제 및 개체 이름 혼동을 해결할 수 있습니다.

Where 절 간의 충돌은 조건 개체를 만들고, 사용자가 보고서 수준에서 UNION 또는 SYNCHRONIZE 연산자를 사용하여 쿼리를 조인하는 방법으로 해결할 수 있습니다.

개체 정의에 Where 절을 정의하면 앞에서 설명한 여러 가지 문제가 발생할 수 있으므로 가능하면 Where 절 대신 조건 개체를 사용하는 것이 좋습니다. 조건 개체를 제대로 사용하면 하드 코드된 Where 절과 관련된 문제를 방지할 수 있습니다.

> **노트**
>
> 자체 제한 조인 외에는 조인 개체에 조인을 만들면 안 됩니다. 조건 개체 안에 조인을 사용하면 재사용 가능한 Where 절에 조인을 만드는 것과 같으므로 해당 조인이 컨텍스트 검색(컨텍스트 자동 검색) 또는 집계 인식 호환성 검색 대상에서 제외됩니다. 구조 창에는 모든 조인이 표시되어야 합니다. 그래야만 자동 검색 도구에서 모든 조인을 대상으로 검색할 수 있습니다.

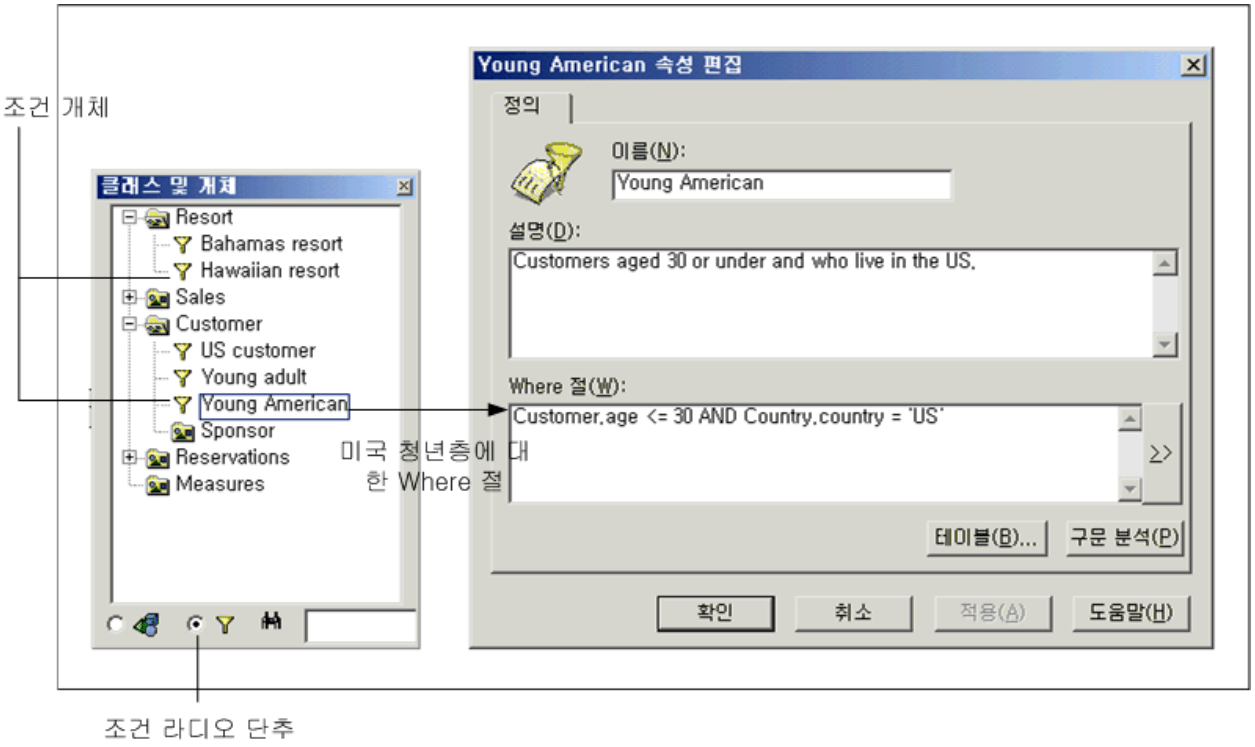
6.6.16 조건 개체 정의

조건 개체는 쿼리 창에서 작업할 때 개체에서 유추된 Select 문에 삽입할 수 있는 미리 정의된 Where 절입니다.

조건 개체는 유니버스 창의 조건 뷰에 저장됩니다. 유니버스 창에서 오른쪽 아래에 있는 조건 라디오 단추를 클릭하면 조건 뷰에 액세스할 수 있습니다.

다음 그림은 Beach 유니버스의 조건 개체 및 미국 청년층 조건이 유추하는 Where 절을 보여 줍니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



6.6.16.1 조건 개체를 사용할 때의 이점 및 제한

조건 개체를 사용하면 다음과 같은 이점이 있습니다.

- 복잡하거나 자주 사용되는 조건에 유용합니다.
- 사용자가 조건을 적용할지 여부를 선택할 수 있습니다.
- 개체를 여러 개 사용하지 않아도 됩니다.
- 조건 개체를 사용해도 유니버스 창의 클래스와 개체 뷰가 변경되지 않습니다.

> **노트**
>
> 유니버스 창의 조건 개체 뷰를 사용하도록 사용자에게 알리는 것이 좋습니다.

조건 개체의 유일한 단점은 데이터 집합의 일부에만 사용자가 액세스할 수 있도록 조건을 강제로 적용해야 하는 경우에는 사용할 수 없다는 것입니다. 이런 경우에는 개체 정의에 Where 절을 정의해야 합니다.

6.6.16.2 충돌하는 Where 절을 조건 개체로 해결할 수 없는 경우

Where 절 충돌로 인해 빈 데이터 집합이 반환되는 경우에는 조건 개체로 문제를 해결할 수 없습니다. 동일한 데이터에 액세스하는 조건 두 개가 포함된 쿼리를 사용자가 실행하면 두 조건이 AND 연산자로 조합되기 때문에 조건을 충족하는 데이터를 찾을 수 없으므로 결국 아무것도 반환되지 않습니다. 이 문제를 해결하려면 사용자가 보고서 수준에서 각 조건 개체에 대해 쿼리를 하나씩 만든 후 두 쿼리를 조합해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.6.16.3 필수 필터

필수 필터에는 다음 두 가지 유형이 있습니다.

- 유니버스: 유니버스 필수 필터는 해당 필터가 속한 클래스에 대해 종속성을 갖지 않습니다. 유니버스 필수 필터는 쿼리에 포함되어 있는 개체(차원, 계수 및 설명)와는 별도로 쿼리에 포함됩니다.
  SAP BW 에서 OLAP 유니버스를 생성할 때 대부분의 SAP Business Warehouse(BW) 변수는 유니버스 필수 필터로 만들어집니다.
- 클래스: 클래스 필수 필터는 개체 클래스 항목이 쿼리에 사용될 경우에만 나타납니다.
  클래스 필수 필터는 사용자가 다음 작업을 수행할 때 트리거됩니다.
  - Web Intelligence 의 쿼리 패널에 있는 결과 창에 개체(차원, 계수 또는 정보)를 추가하는 경우
  - 같은 클래스에 속하는 개체를 결과 창에서 선택하지 않았지만 쿼리 패널의 필터 창에 미리 정의된 유니버스 필터를 추가하는 경우
  - 필수 필터가 있는 클래스에 속하는 개체(차원, 계수 또는 정보)로 필터를 만드는 경우

필수 필터는 기본값을 갖거나 값 목록에 연결될 수 있습니다.

Web Intelligence 의 쿼리 패널에서는 필수 필터가 숨겨져 있으므로 선택할 수 없습니다. 유니버스 디자인 도구에서 쿼리의 필터를 필수 필터로 설정하면 해당 필터는 자동으로 숨겨지며 항목 표시 명령이 비활성화됩니다. 필수 옵션을 선택하지 않으면 필터가 더 이상 숨겨지지 않습니다. 항목 숨기기 명령이 활성화됩니다.

최종 사용자 쿼리에는 둘 이상의 필수 필터가 포함될 수 있습니다. 기본적으로 모든 필수 필터는 쿼리에서 AND 연산자로 연결됩니다.

모든 하위 클래스는 상위 클래스에서 필수 필터를 상속합니다. 하지만 다음 경우에는 주의하십시오.

- @Select 함수를 사용하여 다른 개체를 참조하는 개체(차원, 계수, 설명)는 참조된 개체의 클래스 필수 필터를 상속하지 않습니다.
- @Where 함수를 사용하여 다른 개체의 WHERE 절을 참조하는 개체의 WHERE 절은 참조된 개체의 클래스 필수 필터를 상속하지 않습니다.
- @WHERE 함수를 사용하여 다른 미리 정의된 필터 또는 개체의 WHERE 절을 참조하는 미리 정의된 필터는 참조된 개체의 클래스 필수 필터를 상속하지 않습니다.

예

OLAP 유니버스의 필수 필터

다음 필터(XML 코드에 표시됨)는 사용자가 프롬프트에 입력한 코드를 인증합니다.

```
<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>
```

관련 정보

필수 필터 예 [페이지 280]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

필수 필터 및 값 목록 [페이지 280]

6.6.16.3.1 필수 필터 예

다음 예제는 유니버스 필수 필터를 사용할 수 있는 방법을 보여줍니다.

사용자가 입력한 로그인을 테이블에 저장된 로그인과 비교하려면

```
1 = (Select 1 from Club.dbo.Login
where Login = @Variable('BOUSER')
AND Password = @Prompt('Password?','A',,mono,free) )
```

유니버스의 사용을 오전 9 시와 오후 6 시 사이로 제한하려면

```
1 = (select 1
where datepart(HH,getdate()) between 9 and 18)
```

다음은 클래스 필수 필터의 예입니다.

국가/지역/도시/고객을 포함하는 클래스에서 정의되며 쿼리를 특정 기간에 대한 판매 정보로 제한합니다. 기간을 지정하라는 메시지가 표시됩니다.

```
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))
```

관련 정보

필수 필터 [페이지 279]

6.6.16.4 필수 필터 및 값 목록

필수 필터는 값 목록에 연결될 수 있습니다. 값 목록을 연결하려면 필터가 적용된 개체의 개체 속성 페이지에서 값 목록 옵션을 명시적으로 선택해야 합니다.

유니버스 필수 필터는 계단식 값 목록에 연결될 수 있습니다.

해당 클래스의 하나 이상의 개체가 계단식 값 목록에 속할 경우 클래스 필수 필터를 계단식 값 목록에 연결할 수 있습니다. 계단식 값 목록이 다른 클래스의 개체를 그룹화할 경우에도 상관이 없습니다.

권장 사항

반드시 계단식 값 목록의 최상위 수준에 필수 필터를 생성하십시오.

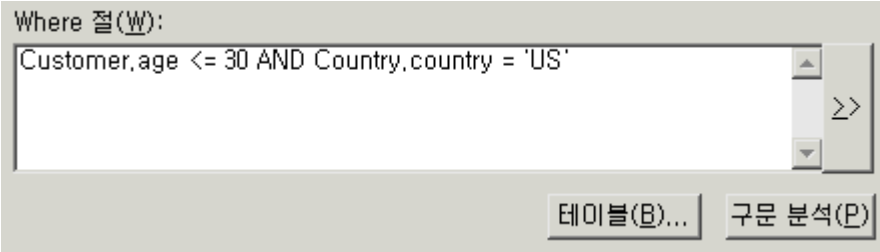
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

계단식 값 목록을 프롬프트가 들어 있는 필수 필터와 연결하지 마십시오. Web Intelligence 에서는 계단식 값 목록 프롬프트를 지원하지 않습니다.

6.6.16.5 조건 개체 만들기

조건 개체를 만들려면

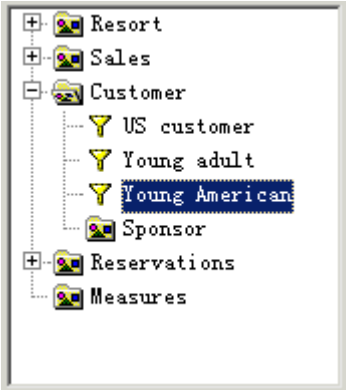
1. 유니버스 창의 오른쪽 아래에 있는 조건 라디오 단추를 클릭합니다.
   유니버스 창의 조건 뷰가 나타납니다. 이 뷰에는 유니버스의 모든 클래스가 트리 뷰로 표시됩니다.
2. 클래스를 마우스 오른쪽으로 클릭하고 조건 삽입을 선택합니다.
   또는
   클래스를 클릭하고 조건 삽입 단추를 클릭합니다.
   속성 편집 대화 상자가 나타납니다. 이름 상자에는 기본 이름이 표시됩니다. *Where* 상자는 비어 있습니다.
3. 조건 이름을 입력합니다.
4. *Where* 절 상자에 WHERE 절 구문을 직접 입력합니다.
   또는
   Where 절 상자 옆에 있는 >> 단추를 클릭하여 *Where* 절 편집기를 엽니다.
5. *SQL* 구조 및 기능 목록에 표시되는 열, 개체, 연산자 또는 함수를 두 번 클릭합니다.
6. 확인을 클릭하여 편집기를 닫습니다.
   다음 그림은 미국 청년층이라는 조건의 정의를 보여 줍니다. 이 조건은 반환 값을 30 세 이하인 미국인 고객으로 제한합니다.



7. 구문 분석을 클릭하여 구문을 검사합니다.
8. 필터를 강제 필터를 정의하려면 강제 필터 확인란을 선택합니다.
   기본적으로 강제 필터는 클래스에 적용되며 값 목록에는 적용되지 않습니다.
9. 라디오 단추를 선택하여 클래스 또는 유니버스에 적용되는 강제 필터를 정의할 수 있습니다.
10. 강제 필터를 값 목록에 적용하려면 값 목록에 적용 확인란을 선택합니다.
11. 확인을 클릭합니다.
    유니버스 창의 조건 뷰에 새 조건 개체가 표시됩니다.

> **노트**
>
> 미리 정의된 필터 편집기를 사용하여 조건 개체를 편집하려면 >>를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



관련 정보

필수 필터 [페이지 279]
필수 필터 및 값 목록 [페이지 280]
OLAP 유니버스의 미리 정의된 조건 [페이지 414]
OLAP 유니버스의 선택적 프롬프트 [페이지 418]

6.6.16.6 동일한 쿼리에 조건 개체 사용

동일한 개체에 대해 정의한 두 개의 조건 개체를 같은 쿼리에 사용하면 두 개의 WHERE 절 모두 거짓 조건을 만들기 때문에 반환되는 데이터가 없습니다. 가능하면 개체 정의에 WHERE 절을 하드 코드하지 않아야 하는 것과 마찬가지로 조건 개체에도 WHERE 절을 사용하지 않는 것이 좋습니다. 사용자는 이로 인해 발생할 수 있는 문제를 알고 있어야 합니다.

빈 데이터 집합이 반환되는 문제는 각 조건 개체에 대해 만든 쿼리 두 개를 조인하는 방법으로 해결할 수 있습니다.

> **노트**
>
> Web Intelligence 사용자가 동일한 쿼리에 두 개의 조건 개체를 사용하지 않도록 하려면 조건 개체 'X'를 'Y' 개체와 함께 사용하면 안 된다는 내용을 해당 개체에 대한 설명에 입력할 수 있습니다.

6.6.16.7 Where 절을 여러 개 사용했을 때 빈 데이터 집합이 반환되는 이유

개체 정의에 Where 절을 추가하면 이 제한은 조인을 기준으로 설정된 제한에 AND 연산자를 사용하여 추가됩니다. 따라서, 같은 데이터 집합에 제한을 적용하는 두 개체를 쿼리 안에 조합하면 Where 절 두 개가 연속적인 AND 절로 조합됩니다. 이러한 쿼리를 실행하면 두 조건 모두를 만족하는 데이터가 없으므로 아무 데이터도 반환되지 않습니다.

예를 들어, Bahamas 및 Hawaiian Club 호텔 휴양지에서 제공하는 서비스를 검색한다고 가정합니다. Bahamas 휴양지와 Hawaiian 휴양지에 대한 조건 개체를 사용하여 다음과 같은 쿼리를 실행합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

이 쿼리의 SQL 은 다음과 같습니다.

```
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'  ))
```

Where 절 마지막에 Where 절 제한 두 개가 AND 절로 조합됩니다.

쿼리가 실행될 때 같은 쿼리 안에서 국가에 대한 두 가지 제한을 모두 만족할 수 없으므로 아무 데이터도 반환되지 않습니다. 반입할 데이터가 없다는 메시지가 나타납니다.

쿼리 두 개를 만들어 제한 조합

각 Where 절에 대해 쿼리를 하나씩 만든 다음 UNION 연산자를 사용하여 결과를 조합하면 같은 쿼리 안에 조건 개체 두 개를 사용했을 때 발생하는 문제를 해결할 수 있습니다.

6.6.17 자체 제한 조인을 사용하여 제한 적용

자체 조인을 사용하면 두 개의 열 사이를 전환하는 데 사용되는 플래그에 기반하여 테이블의 특정 열 또는 다른 열로 데이터를 제한할 수 있습니다. 플래그는 이때 사용되는 세 번째 열로 플래그 값은 두 개의 대체 열 중 쿼리에 사용되는 열을 결정합니다.

자체 제한 조인 만들기 및 사용에 대한 자세한 내용은 자체 제한 조인 [페이지 160] 단원을 참조하십시오.

6.6.18 여러 테이블을 유추하여 제한 적용

개체가 유추하는 테이블과 다른 테이블에서 동시에 일치하는 값으로 개체의 반환 값을 제한할 수 있습니다.

예를 들어, 출신 국가라는 개체는 Country 테이블을 유추합니다. 출신 국가 개체는 다음과 같은 데이터를 반환합니다.

Country of origin
Australia
France
Germany
Holland
Japan
UK
US

판매 직원의 국적만 반환하도록 출신 국가 개체를 판매 직원 클래스에서 사용하려면 개체 이름을 판매 직원 국적으로 변경한 후 Sales_Person 테이블에 있는 판매 직원의 국적에 해당하는 값만 반환하도록 Country 테이블을 제한할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

판매 직원 국적 개체의 SQL 은 다음과 같습니다.

```
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  )
```

판매 직원 국적 개체는 다음과 같은 데이터를 반환합니다.

Sales people countries
France
Germany
Japan
UK
US

이 예제에서는 국가 개체를 쿼리에 사용할 때 Select 문의 From 절에 Sales_Person 테이블도 함께 유추하도록 지정하는 방법으로 제한을 적용합니다.

그러면 판매 직원 클래스에 사용된 국가 개체는 판매 직원의 국적에 해당하는 국가만 반환합니다. 개체 정의 시트에 있는 테이블 단추를 사용하여 이 제한을 적용할 수 있습니다.

Country 테이블은 동등 조인만 사용하는 중간 조인을 통해 Sales_Person 테이블과 조인되어야 합니다.

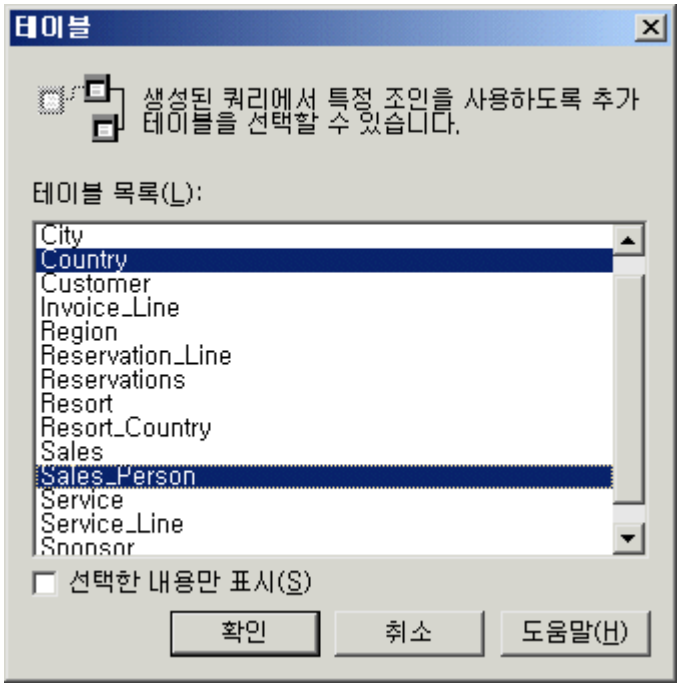
> **노트**
>
> Select 문에 테이블 제한이 정의되어 있는 개체의 SQL 을 변경하면 유니버스 디자인 도구에서는 해당 개체의 Select 문과 Where 절에 필요한 테이블을 자동으로 다시 결정합니다. 개체에서 유추하는 테이블에서 테이블 제한이 다시 정의되더라도 사용자에게 메시지가 표시되지 않습니다.

6.6.18.1 여러 테이블을 유추하여 조건 적용

여러 테이블을 유추하여 개체에 조건을 적용하려면

1. 개체를 두 번 클릭합니다.
   개체에 대한 속성 편집 대화 상자가 열립니다.
2. 테이블 단추를 클릭합니다.
   유니버스의 테이블 목록이 표시됩니다.
3. 현재 테이블 외에 개체에서 유추할 테이블을 하나 이상 선택합니다. Ctrl 키를 누른 채로 목록에서 테이블 이름을 클릭하면 테이블을 여러 개 선택할 수 있습니다. 다음 그림에는 Country 테이블과 Sales_Person 테이블이 선택되었습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



4. 각 대화 상자에서 확인을 클릭합니다.
5. Web Intelligence 에서 쿼리를 실행하여 테이블 제한을 테스트합니다.

6.6.18.2 각 제한 적용 방법을 사용하는 경우

다음과 같은 지침을 사용하여 유니버스에 제한을 설정할 수 있습니다.

- 개체 정의에는 Where 절을 사용하지 않아야 합니다. Where 절을 사용해야 할 경우에는 개체를 여러 개 사용하거나 충돌하는 Where 절을 사용했을 때 발생할 수 있는 문제에 대해 알고 있어야 합니다.
- 조건 개체는 개체를 여러 개 사용하거나 유니버스 창의 클래스 및 개체 뷰를 변경할 필요가 없도록 사용자에게 미리 정의된 선택적 조건을 제공할 때 사용합니다.
- 자체 제한 조인은 SQL 내에서의 테이블 위치와 관계없이 테이블에 대해 제한을 적용하려는 경우에 사용합니다. 이 방법은 테이블에서 플래그를 사용하여 둘 이상의 도메인 간에 전환하는 경우에 적절합니다.
- 유니버스에서 조회 테이블이 둘 이상의 용도로 사용되는 경우에는 추가 조인을 사용합니다.

6.6.19 연결 개체

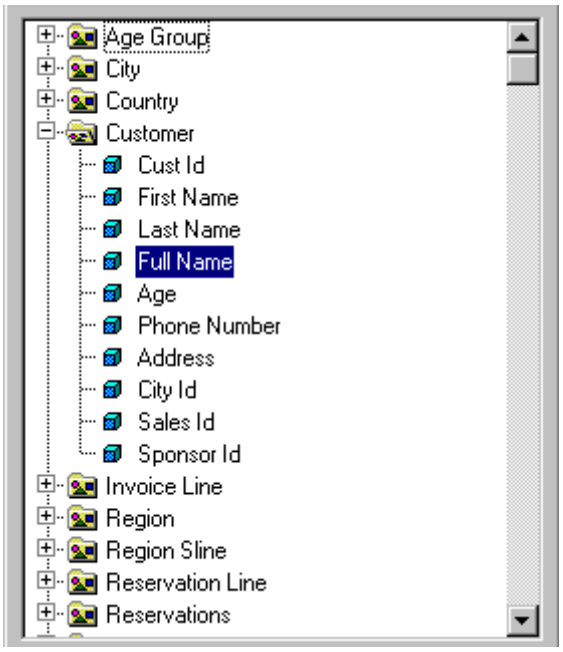
연결 개체는 기존 개체 두 개의 조합입니다. 예를 들어 **고객** 클래스에서 **성** 개체와 **이름** 개체를 조합하여 **설명**이라는 개체를 만듭니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.6.19.1 연결 개체 만들기

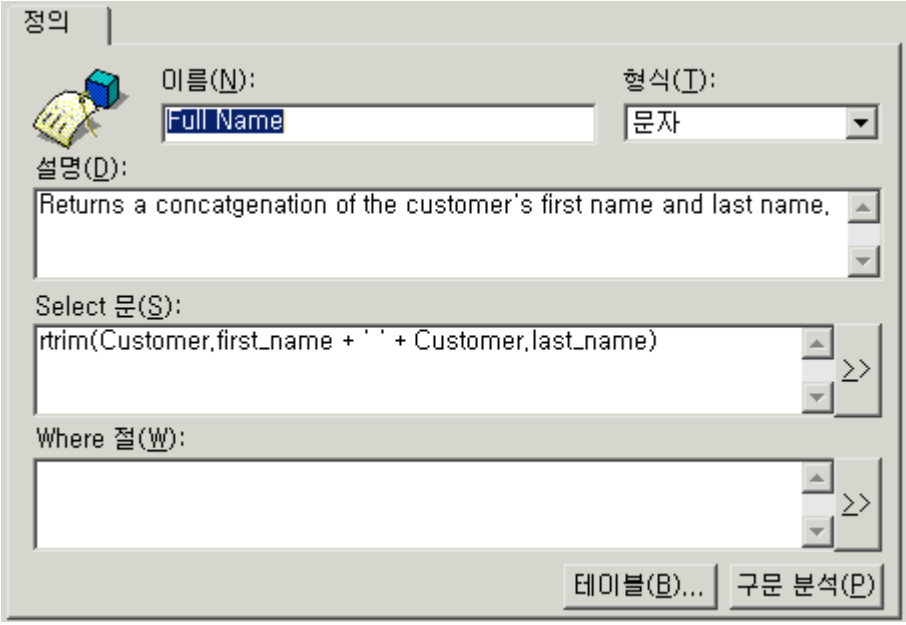
연결 개체를 만들려면

1. 개체를 만듭니다.
   예를 들어, Customer 클래스에 Full Name 개체를 새로 만듭니다. "이 개체는 고객의 이름과 성을 연결한 개체입니다"와 같이 개체에 대한 설명도 입력해야 합니다.



2. 개체를 두 번 클릭합니다.
   속성 편집 대화 상자가 나타납니다.
3. 연결 개체의 구문을 Select 상자에 입력합니다.
   예를 들어, 성명 개체에 대해 다음과 같이 구문을 입력합니다(MS Access 구문).
   rtrim (Customer.first_name + ' ' + Customer.last_name)
   여기서 rtrim 은 문자열 끝에 있는 공백을 제거하는 함수이고 따옴표 두 개는 이름과 성 사이에 공백을 삽입하는 데 사용됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



> **노트**
>
> 편집 단추를 클릭하여 SQL 편집기를 열 수도 있습니다. SQL 편집기의 그래픽 도구를 사용하여 개체의 SQL 구문을 지정할 수 있습니다. 이 편집기에 대한 자세한 내용은 스키마 디자인 장을 참조하십시오.

4. 각 대화 상자에서 확인을 클릭합니다.
   전체 이름 개체에 대해 쿼리를 실행하면 전체 이름이 이름의 알파벳 순서대로 나열되어 반환됩니다.

6.7 계층 정의

개체 계층을 만들면 사용자가 다차원 분석을 수행할 수 있습니다.

6.7.1 다차원 분석이란?

다차원 분석은 의미 있는 계층으로 구성된 차원 개체에 대한 분석입니다.

다차원 분석 기능을 사용하면 사용자는 다양한 관점에서 데이터를 관찰할 수 있습니다. 이를 통해 데이터에서 추세나 예외를 파악할 수 있습니다.

계층은 서로 관련된 차원의 배열입니다. 예를 들어, **지리**에 **국가**, **지역** 및 **도시**라는 차원이 그룹화되어 있는 경우를 계층이라고 합니다.

Web Intelligence 에서 드릴업이나 드릴다운을 사용하여 다차원 분석을 수행할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.7.1.1 Drill

드릴을 사용하면 계층의 여러 수준을 탐색할 수 있습니다. 사용자는 계층에서 "드릴업"하거나 "드릴다운"할 수 있습니다.

예를 들어, 관리자가 예약 데이터를 시간별로 추적하려 한다고 가정합니다. 이 경우 유니버스 디자이너는 **예약** 시간 계층을 만들어 그 아래에 **예약 연도**, **예약 분기**, **예약 달** 및 **예약 날짜** 차원을 포함시킬 수 있습니다.

이 예제에서 상위 수준의 집계인 **예약 분기**에서는 **예약 달**이나 **예약 날짜** 같이 더 세부적인 수준으로 드릴다운할 수 있습니다. 또는 **예약 분기**에서 **예약 연도**로 드릴업하여 요약 데이터를 볼 수 있습니다.

6.7.2 계층구조 식별 방법

계층은 다양한 형식으로 구성될 수 있습니다. 일반적인 계층은 다음과 같습니다.

- 지리: 대륙 국가 지역 구/군/시
- 제품: 범주 브랜드 제품
- 시간: 년 분기 월 주 일

다음과 같이 "혼합" 계층구조를 사용할 수도 있습니다.

지리/제품: 대륙 국가 범주 브랜드 제품

데이터 내의 암시적인 계층은 데이터 유형과 데이터가 데이터베이스에 저장된 방식에 따라 달라집니다. 시스템 내에서 사용자 그룹에 필요한 분석 요구 사항에 가장 적합한 계층을 만들려면 데이터를 신중하게 분석해야 합니다.

데이터 내에서 계층이 존재하는 위치를 찾는 데 정해진 규칙은 없지만 데이터베이스 구조에 일대다(1-N) 관계가 있는 경우에 계층을 만들 수 있습니다.

아래 스키마에서는 테이블 간의 일대다 관계를 기반으로 지리적 계층구조를 유추할 수 있습니다.

6.7.3 계층 설정

기본적으로 유니버스 디자인 도구에서는 다차원 분석에 대해 일련의 기본 계층 구조를 제공합니다. 이러한 기본 계층에는 유니버스 창에 표시되는 순서대로 클래스와 개체가 배열됩니다. 개체를 만들 때는 사용자가 기본 계층을 이해할 수 있도록 개체를 계층 구조로 구성해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

경우에 따라서는 사용자 지정 계층을 만들어 다른 클래스의 개체를 포함시켜야 할 수 있습니다. 이러한 경우에는 새 계층 구조를 만들어야 합니다.

계층 구조 편집기에서 기본 계층 구조를 확인하고 새 계층 구조를 만들 수 있습니다. 계층 편집기는 유니버스의 계층을 관리하는 데 사용할 수 있는 그래픽 편집기입니다.

> **노트**
>
> 사용자 지정 계층구조를 정의할 때 기본 계층구조는 비활성화되며 최종 사용자에게 제공되지 않습니다. 기본 계층구조를 활성화하려면 계층구조 편집기에서 기본 계층구조를 선택하여 사용자 지정 계층구조 목록에 추가해야 합니다.

6.7.3.1 계층 보기

다음과 같이 계층을 볼 수 있습니다.

6.7.3.1.1 유니버스에서 계층구조를 보려면

1. 도구 > 계층구조를 선택합니다.

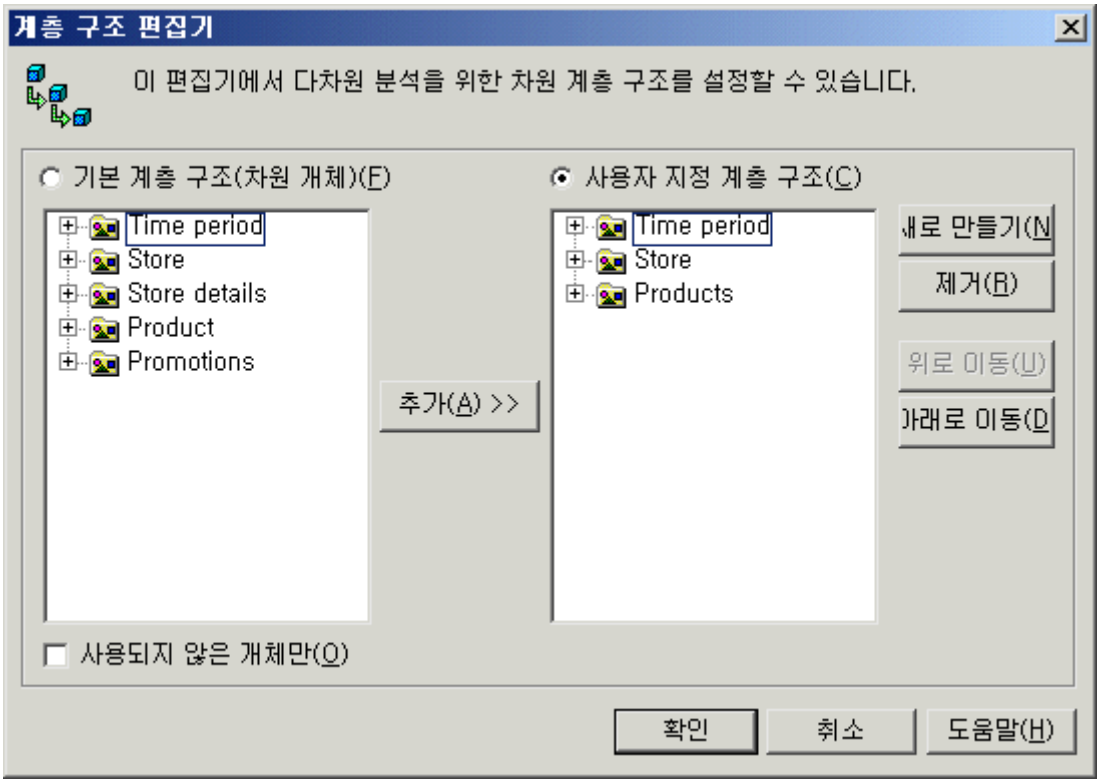
   또는

   계층구조 단추를 클릭합니다.

   계층구조 편집기가 나타납니다. 유니버스 디자인 도구에서 계층 구조는 폴더 모양으로, 차원은 큐브 모양으로 표시됩니다.

   왼쪽 창에는 활성 유니버스에서 차원 개체를 포함하는 클래스가 나열됩니다. 오른쪽 창에는 사용자가 만드는 모든 사용자 지정 계층이 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



2. 계층 노드(+ 기호)를 클릭하여 계층구조로 정리된 차원을 봅니다.
3. 취소를 클릭합니다.

6.7.3.2 계층 설정

새 계층구조를 만들려면 사용자 지정 계층구조 창에서 새 폴더를 만든 다음 적절한 차원을 순서대로 추가합니다.

계층구조나 차원을 선택한 다음 제거 단추를 클릭하면 계층구조나 계층구조 내의 차원을 삭제할 수 있습니다.

6.7.3.2.1 새 계층을 만들려면

1. 계층구조 편집기에서 새로 만들기 단추를 클릭합니다.

   또는

   계층 편집기의 왼쪽 창에서 클래스를 선택한 다음 오른쪽 창으로 끌어옵니다.

   그러면 계층을 나타내는 폴더 모양이 오른쪽 창에 표시됩니다.
2. 계층 이름을 입력합니다.
3. Enter 키를 눌러 이름을 적용합니다.
4. 새 계층구조를 선택합니다.

   계층구조가 강조 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5. 왼쪽 창에서 기본 계층구조 노드를 확장합니다.

   이 계층구조에는 새로 만든 사용자 지정 계층구조에 추가할 수 있는 차원이 들어 있습니다.
6. 차원을 클릭합니다. 여러 개의 차원을 선택하려면 Ctrl 키를 누른 채로 원하는 차원을 클릭합니다.

   하나 이상의 차원이 강조 표시됩니다.
7. 추가 단추를 클릭합니다.

   오른쪽 창의 선택한 계층구조 아래에 하나 이상의 차원이 표시됩니다.

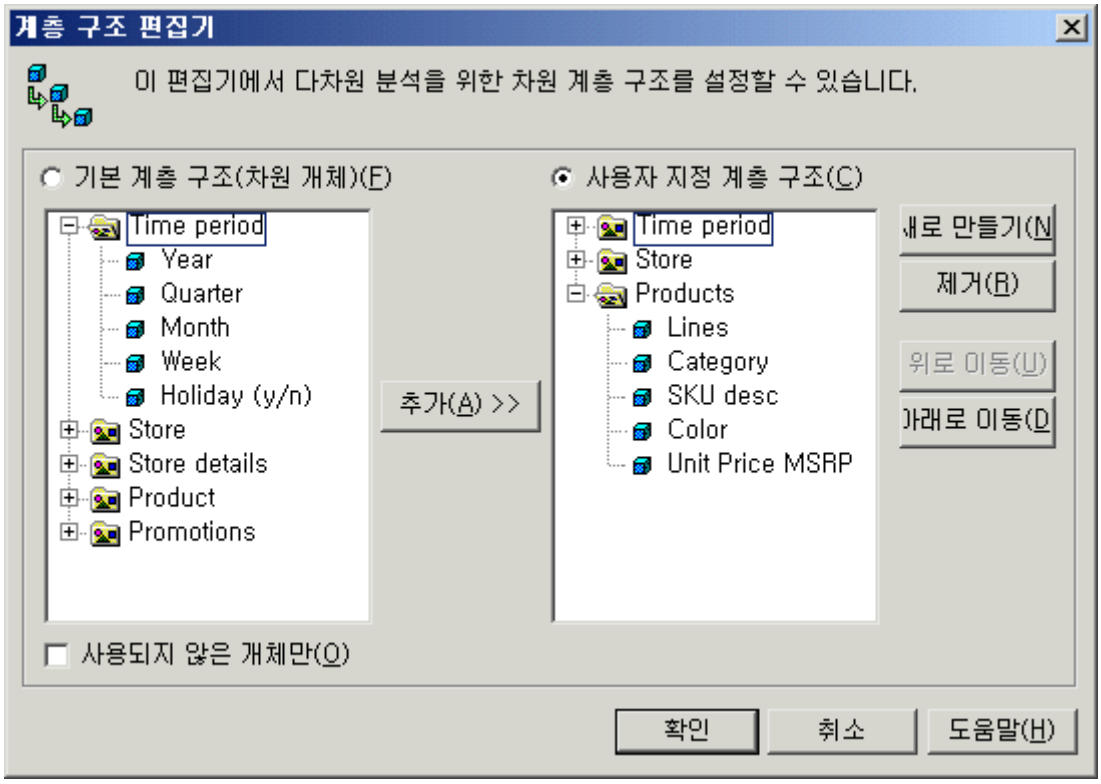
> **노트**
>
> 사용되지 않은 개체만 확인란을 선택하면 계층구조에 아직 포함하지 않은 차원 개체만 볼 수 있습니다.

6.7.3.3 차원 및 계층의 순서 변경

계층구조 내에 차원 개체가 표시되는 순서를 변경할 수 있습니다. 개체를 이동하려면 개체를 클릭한 다음 위로 이동 또는 아래로 이동 단추를 클릭합니다. 같은 방법으로 계층의 순서를 변경할 수도 있습니다.

끌어서 놓는 방법으로 차원 개체나 계층구조를 이동할 수도 있습니다.

다음 그림은 계층 및 차원 개체의 예제를 보여 줍니다.



위의 계층 편집기에는 기간, 매장 및 제품이라는 세 가지 사용자 지정 계층이 설정되어 있습니다. 제품 계층구조는 계열, 범주, SKU 설명, 색상 및 단위 가격 MSRP 라는 차원으로 구성되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.8 계층에 대한 계단식 값 목록 사용

계단식 값 목록이라는 값 목록에 기본 또는 사용자 지정 계층구조를 연결할 수 있습니다.

> **노트**
>
> LOV(값 목록)는 개체와 관련된 데이터 값이 들어 있는 목록입니다. 값 목록은 값 목록 사용 [페이지 295] 단원에서 자세히 설명합니다.

계단식 값 목록은 유니버스의 계층구조와 연결된 일련의 값 목록을 말합니다. 각 계층구조 수준에는 해당 수준에 대한 값 목록을 반환하는 프롬프트가 정의되어 있습니다.

계단식 값 목록에 연결된 계층구조를 포함하는 보고서를 새로 고치면 해당 계층구조가 표시되고, 쿼리가 실행되기 전에 수준을 선택하고 값 목록에서 하나 이상의 값을 선택하라는 프롬프트가 표시됩니다.

예를 들어 Reservation quarter 는 Year 계층구조와 연결되어 있습니다. 쿼리에 Reservation quarter 월이 사용되면 Year 계층구조가 표시되고, 쿼리를 실행하기 전에 해당 quarter 의 연도를 선택하라는 프롬프트가 표시됩니다.

6.8.1 계단식 값 목록 만들기

기본 계층구조 또는 사용자 지정 계층구조에 대한 계단식 값 목록을 만들 수 있습니다. 각 수준에 대해 .LOV 파일이 만들어집니다. 쿼리가 실행되면 프롬프트가 표시된 계층구조 수준에 대한 LOV 만 반환됩니다.

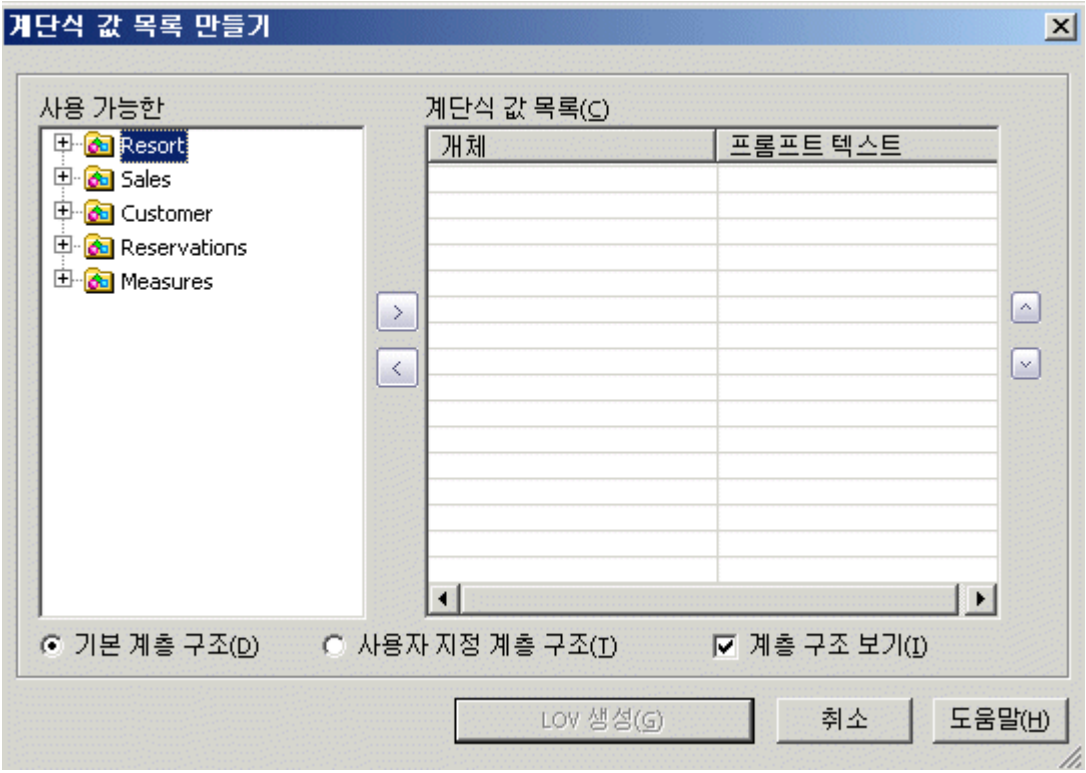
> **노트**
>
> 계층구조에 대한 값 목록의 이러한 반복적 사용은, 모든 계층구조 수준에 대한 모든 값 목록이 마이크로큐브에 반환되는, 값 목록에 대한 계층구조 만들기와는 다릅니다. 계단식 값 목록이 사용되면 해당 수준에 대한 프롬프트에 내용을 입력하기 전에는 마이크로 큐브에 어떠한 LOV 도 반환되지 않으며 해당 수준에 대한 LOV 만 반환됩니다.

6.8.1.1 계단식 값 목록을 만들려면

1. 도구 > 값 목록 > 계단식 값 목록 만들기 를 선택합니다.

   계단식 값 목록 만들기 대화 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



다음 옵션 중에서 선택할 수 있습니다.

표 141:

계단식 LOV 옵션	설명
기본 계층구조 사용자 지정 계층구조	하나를 선택하면 유니버스에 정의된 상응하는 기본 또는 사용자 지정 계층 구조가 사용 가능 창에 나타납니다. 해당 계층 유형에 대한 자세한 내용은 계층 설정 [페이지 288] 단원을 참조하십시오.
계층 구조 보기	이 옵션을 선택하면 쿼리 패널의 트리 뷰에 계층이 표시되어 계층을 쉽게 탐색할 수 있습니다. 수준을 클릭하면 쿼리 패널의 오른쪽 창에 값 목록이 나타납니다.
개체	특정 차원의 계층구조 수준
프롬프트 텍스트	해당 수준의 값 목록에 대한 프롬프트에 표시되는 텍스트

2. 기본 계층 구조 또는 사용자 지정 계층 구조 라디오 단추를 클릭합니다.

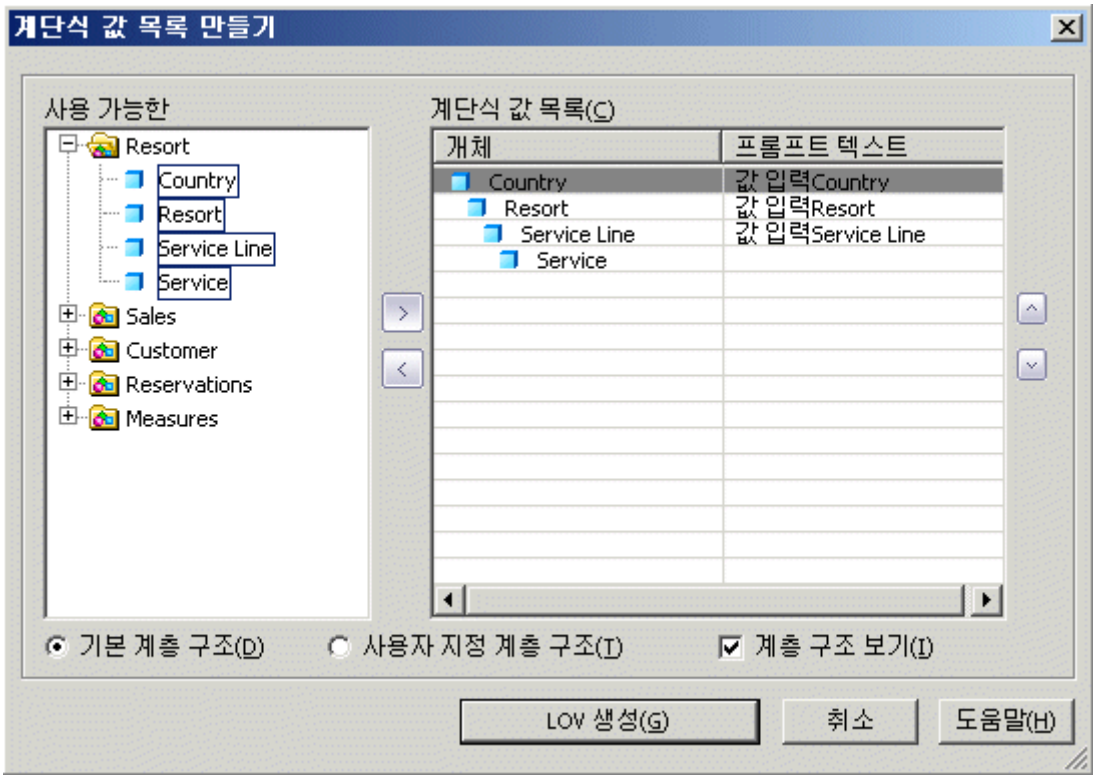
   유니버스에서 사용할 수 있는 해당 계층 목록이 나타납니다.
3. 클래스를 클릭하거나 클래스를 확장하고 하나 이상의 개체를 선택합니다.
4. 오른쪽 화살표를 클릭합니다.

   클래스의 모든 개체가 개체 목록에 나타납니다.

   또는

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

선택된 개체가 개체 목록에 나타납니다.



5. 각 개체에 대해 프롬프트 텍스트를 입력합니다.
6. 계단식 값 목록에서 특정 개체의 위치를 변경하려면 해당 개체를 클릭하고 위쪽/아래쪽 화살표를 이용하여 목록의 위 또는 아래 방향으로 옮깁니다.

   특정 개체를 제거하려면 해당 개체를 클릭하고 왼쪽 화살표를 클릭합니다.
7. 계층 구조 보기 확인란을 선택하거나 선택을 취소합니다.
8. *LOV 생성*을 클릭합니다.

   계단식 값 목록 만들기 대화 상자가 사라집니다. 계단식 값 목록의 각 수준마다 LOV 가 하나씩 만들어집니다. 각 .LOV 파일은 파일 시스템의 유니버스 하위 폴더에 저장됩니다. 예: `C:\Documents and Settings\<사용자>\Application Data\Business Objects\Business Objects 12.0\Universes\<CMS 이름>\beachXI3.0\`.

> **노트**
>
> 편집, CMS 로 내보내기, 개별 개체에 대한 값 목록 만들기와 관련된 내용은 값 목록 사용 [페이지 295] 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.9 값 목록 사용

값 목록은 개체와 관련된 데이터 값이 들어 있는 목록입니다. 값 목록에는 두 가지 유형의 데이터 소스에서 가져온 데이터가 포함될 수 있습니다.

표 142:

값 목록 데이터 소스	설명
데이터베이스 파일	개체를 만들면 유니버스 디자인 도구에서 값 목록이 개체에 자동으로 연결됩니다. 값 목록은 사용자나 디자이너가 쿼리 창에 개체의 값 목록을 표시하도록 선택해야만 만들어집니다. 그런 다음 개체에서 유추된 하나 이상의 열에 대해 SELECT DISTINCT 쿼리가 실행됩니다. 반환 데이터는 유니버스 파일이 저장되는 폴더 아래에 작성된 유니버스 하위 폴더에서 확장자가 .LOV 인 파일에 저장됩니다. 그런 다음 .LOV 파일은 목록의 값에 대한 소스로 사용됩니다.
외부 파일	개인 데이터(예: 텍스트 파일) 또는 Excel 파일을 값 목록에 연결할 수 있습니다. 외부 파일에 기반하는 값 목록은 고정되어 있습니다. 외부 파일과는 동적 연결을 할 수 없습니다. 따라서 외부 파일이 변경되면 .LOV 파일을 새로 고쳐야 합니다.

6.9.1 값 목록이 사용되는 방법

Web Intelligence 에서는 조건을 적용할 때 개체에 적용할 값 목록 표시 피연산자를 사용하여 쿼리 창에서 쿼리를 만들 수 있습니다.

> **노트**
>
> .LOV 파일은 개체가 유추하는 열 값을 제한하도록 요구하는 조건을 쿼리 창에서 개체에 적용할 때마다 만들어집니다.

개체의 값 목록에는 사용자가 조건의 항목으로 선택할 수 있는 값 목록이 나열됩니다. 값 목록을 처음 사용하면 파일 시스템에 있는 유니버스 하위 폴더에 .LOV 파일 형식으로 목록이 저장됩니다. 이렇게 하면 특정 개체에 대해 SELECT DISTINCT 쿼리가 한 번만 실행됩니다.

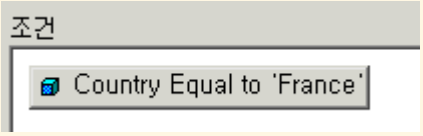
이 폴더에는 디자이너가 데이터 액세스를 제어할 개체에 대해 반환되는 값 목록을 제한하기 위해 유니버스 디자인 도구에서 만든 .LOV 파일도 저장됩니다.

> **예**
>
> **국가에 대한 값 목록 사용**
>
> 국가라는 개체에는 COUNTRY.COUNTRY_NAME 이라는 Select 절 정의가 있습니다. 개체에 연결된 기본 값 목록에는 COUNTRY_NAME 열의 고유한 국가 이름이 모두 포함되어 있습니다. 이 목록은 국가 개체를 쿼리 조건에 사용했을 때 반환됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

쿼리의 값을 프랑스로만 제한하려면 조건에 사용할 수 있는 Country 테이블의 모든 국가 값이 나열된 아래의 목록에서 프랑스를 선택하면 됩니다.

목록에서 프랑스를 선택하면 쿼리 창의 조건 창에 다음과 같이 조건이 표시됩니다.



쿼리는 프랑스에 해당하는 값만 반환합니다.

6.9.2 개체에 값 목록을 사용하는 방법 정의

유니버스 디자인 도구에서 차원 개체나 설명 개체를 만들면 연결된 값 목록이 자동으로 할당됩니다. 이 목록은 개체를 만들었을 때는 실제로 존재하지 않습니다. 그러나 기본적으로 개체가 쿼리 창에 사용되었을 때 데이터베이스를 쿼리하여 해당 개체의 값 목록을 반환하는 기능이 개체에 있습니다.

노트

계수 개체에 대해서는 기본 값 목록이 지정되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스 디자인 도구에 값 목록을 표시해야 하는 조건을 쿼리 창에서 개체에 처음으로 지정하면 개체가 유추하는 적절한 열에 대해 SELECT DISTINCT 문이 실행되어 값 목록이 반환됩니다.

값 목록을 보유할 .LOV 파일이 유니버스 하위 폴더에 자동으로 만들어집니다. 다음 번에 유니버스 디자인 도구에 개체의 값 목록이 필요하면 데이터베이스가 아닌 .LOV 파일에서 값이 반환됩니다.

6.9.2.1 값 목록을 제어할 때의 디자이너 역할

유니버스 디자이너는 목록에 데이터가 표시되는 방법 및 목록에 반환되는 데이터의 양과 형식에 대한 제한을 정의할 수 있습니다.

개체의 속성을 설정하면 값 목록에 대해 다음과 같은 작업을 수행할 수 있습니다.

- 개체에 연결된 값 목록이 있는지 여부를 지정합니다.
- 목록을 새로 고치는 시점을 지정합니다.
- 개체가 값 목록을 반환하는 데 사용하는 SELECT DISTINCT 쿼리에 조건을 설정하는 쿼리를 정의합니다. 이 쿼리는 개체 속성에 저장해야 합니다.
- 값을 간단한 목록으로 표시할지 또는 개체 계층구조로 표시할지 지정합니다.
- 목록이 열 값에 기반하는지 또는 외부 파일(예: Excel 스프레드시트)의 값에 기반하는지 지정합니다.

개체에 대해 영구적인 값 목록을 만든 다음 이 목록을 리포지토리로 내보낼 수도 있습니다. 그러면 이 .LOV 파일이 해당 개체의 값 목록으로 항상 사용됩니다. 이 목록은 업데이트되지 않습니다.

6.9.3 값 목록 속성 및 옵션

다음과 같은 개체 속성을 정의하면 Web Intelligence 에서 개체의 값 목록이 사용되는 방법을 제어할 수 있습니다.

표 143:

속성	설명
값 목록 연결	• 이 확인란을 선택하면 개체에 값 목록을 연결할 수 있습니다. 이 옵션은 기본적으로 선택되어 있습니다. • 이 확인란의 선택을 취소하면 개체에 값 목록이 연결되지 않습니다. • 차원 및 설명 개체에 대해서는 자동으로 선택됩니다. 그러나 계수 개체에 대해서는 선택되지 않습니다.
목록 이름	반환되는 목록 데이터를 저장하는 .LOV 파일 이름입니다. 8 자로 제한됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

속성	설명
값 목록 편집 허용	• 이 확인란을 선택하면 사용자가 Web Intelligence 에서 값 목록을 편집할 수 있습니다. • 선택을 취소할 경우 사용자가 목록을 편집할 수 없습니다. **노트** Excel 스프레드시트 같은 개인 데이터 파일에는 이 옵션을 사용할 수 없습니다. 개인 데이터 파일은 리포지토리로 내보낼 수 없습니다. 이러한 파일은 로컬 컴퓨터에 저장됩니다. 사용자는 로컬 파일을 편집하거나 다른 로컬 데이터 파일을 사용하도록 대상 값 목록을 변경할 수 있습니다. 일반적으로 값 목록은 사용자에게 허용되는 값 집합을 제한하는 데 사용됩니다. 사용자가 목록을 편집할 수 있다면 사용자가 선택한 값을 디자이너가 더 이상 제어할 수 없습니다. 일반적으로 개인 데이터 파일을 값 목록 원본으로 사용하는 경우가 아니면 사용자가 값 목록을 편집하지 못하도록 이 옵션을 선택하지 않는 것이 좋습니다.
사용하기 전에 자동으로 새로 고침 (BusinessObjects 에만 해당)	• 이 확인란을 선택하면 개체의 값 목록을 쿼리 창에 표시할 때마다 목록 데이터를 새로 고칠 수 있습니다. 이 옵션은 .LOV 파일을 새로 고칠 때마다 성능에 영향을 줄 수 있습니다. Web Intelligence 보고서에는 이 옵션을 사용할 수 없습니다. • 이 확인란의 선택을 취소하면 사용자 로그온 세션을 시작할 때 한 번만 목록을 새로 고칩니다. 정기적으로 변경되는 값이 목록에 포함되어 있으면 이 옵션을 선택해도 되지만 성능에 미치는 영향도 고려하는 것이 좋습니다. 목록의 내용에 변동이 없는 경우에는 이 옵션을 선택하지 않는 것이 좋습니다.
계층 표시	계층구조 표시 속성을 선택하면 Web Intelligence 의 계층구조로 계단식 값 목록을 표시할 수 있습니다.
유니버스와 함께 내보내기	• 이 확인란을 선택하면 개체의 .LOV 파일을 유니버스와 함께 리포지토리로 내보낼 수 있습니다. • 값 목록을 내보내려면 개체에 연결된 값 목록을 만들어야 합니다. 이 목록은 .LOV 파일로 저장됩니다. • 이 확인란의 선택을 취소하면 리포지토리에 내보낼 때 개체의 .LOV 파일이 포함되지 않습니다. 값 목록을 정기적으로 사용자 지정하는 경우에는 이 옵션을 선택하는 것이 좋습니다. 이렇게 하면 수정 사항을 유니버스와 함께 내보내고 가져올 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

속성	설명
위임 검색	Web Intelligence 사용자는 위임 검색 속성을 사용하여 값 목록에 반환되는 값을 제한할 수 있습니다. 위임 검색 속성을 선택하면 Web Intelligence 에서 쿼리 런타임에 사용자에게 빈 값 목록 상자를 표시합니다. 사용자는 검색 기준을 정의하는 값을 입력하여 값 목록을 필터링할 수 있습니다. 대부분의 데이터 소스의 경우 와일드카드 문자를 사용하여 데이터베이스에서 쉽게 검색을 수행할 수 있습니다. Web Intelligence 에서 지원하는 와일드카드 문자는 다음과 같습니다. * : 원하는 수의 문자(영(0) 문자도 포함)와 일치 ? : 정확히 하나의 문자와 일치 \ : 문자 이스케이핑을 통해 와일드카드 문자 검색 가능 위임 검색 옵션에는 다음과 같은 제한 사항이 있습니다. 계산식 값 목록에서는 위임 검색이 지원되지 않습니다. 개체의 값 목록이 형식 문자인 경우에만 위임 검색을 활성화할 수 있습니다. 값 목록에 대해 사용자 지정 SQL 을 입력할 때는 위임 검색을 활성화할 수 없습니다. 값 목록의 계층 표시를 사용할 때는 위임 검색을 활성화할 수 없습니다.

다음과 같은 단추를 클릭하면 값 목록을 편집 및 표시하거나 기본 이름을 지정할 수 있습니다.

표 144:

옵션	설명
기본값 복원	개체를 만들 때 지정한 .LOV 파일의 기본 이름을 복원합니다.
편집	목록에 표시된 값을 편집할 수 있습니다. 편집기를 사용하면 쿼리 창에서 값 목록을 사용할 때 해당 목록에 표시되는 값을 제한할 수 있습니다.
표시	개체의 값 목록을 표시합니다. 유니버스와 함께 리포지토리로 내보낼 영구적인 목록을 만들려면 표시를 클릭하여 .LOV 파일을 만들어야 합니다. 그런 다음 파일을 편집할 수 있습니다.

6.9.3.1 값 목록의 속성 및 옵션 정의

값 목록 파일(.LOV)의 속성과 옵션을 정의하려면

1. 개체를 두 번 클릭합니다.
   속성 편집 대화 상자의 정의 페이지가 열립니다.
2. 속성 탭을 클릭합니다.
   속성 페이지가 나타납니다.
3. 페이지 아래쪽에 있는 값 목록 그룹 상자의 확인란을 선택하거나 선택 취소합니다.
4. 연결된 .LOV 파일의 이름을 목록 이름 상자에 입력합니다.
5. 원하는 경우 편집 단추를 클릭하여 값 목록에 대해 제한을 정의합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6. 쿼리 창을 사용하여 목록 데이터에 대한 쿼리를 만듭니다.
7. 표시 단추를 클릭하여 값 목록을 봅니다.
   이 단추를 클릭하면 개체에서 유추되는 데이터베이스 열에 대해 SELECT DISTINCT 쿼리가 실행됩니다. 이 메서드는 보고서 작성 제품에서 개체의 .LOV 파일을 만드는 데 사용되는 동일한 메서드입니다.
8. 확인을 클릭합니다.

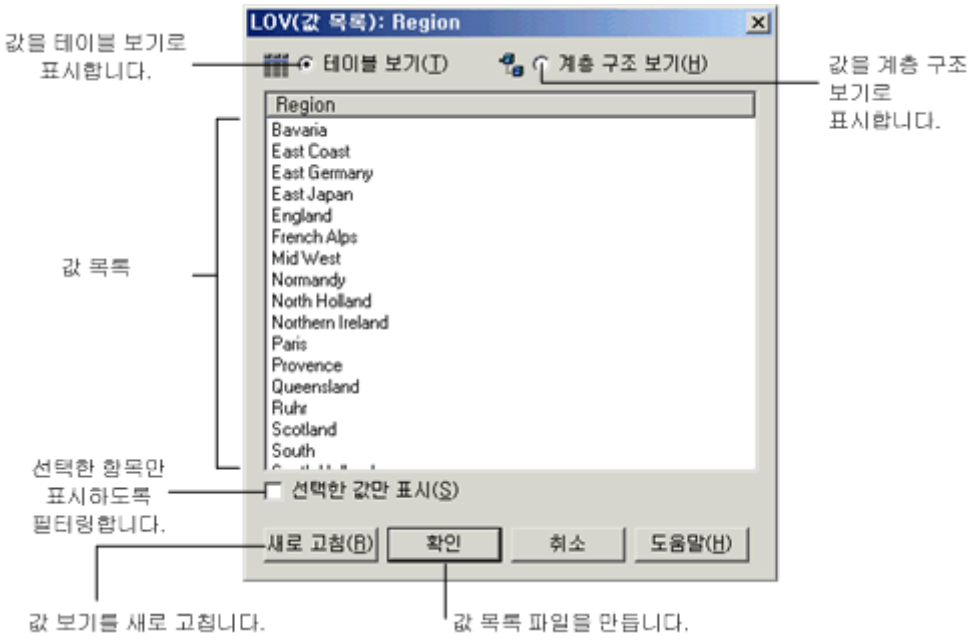
6.9.3.2 개체에 연결된 값 목록 보기

유니버스 디자인 도구에서는 개체에 연결된 값 목록을 볼 수 있습니다. 값 목록을 보면 반환되는 데이터가 저장될 기본 .LOV 파일이 User Docs 디렉터리에 자동으로 만들어집니다. 기본적으로 .LOV 파일은 값 목록을 볼 때 자동으로 만들어집니다.

값 목록은 목록 형식이나 개체 계층구조 형식으로 볼 수 있습니다.

값 목록을 보려면

1. 개체를 두 번 클릭합니다.
   속성 편집 대화 상자의 정의 페이지가 열립니다.
2. 속성 탭을 클릭합니다.
   속성 페이지가 나타납니다.
3. 표시 단추를 클릭합니다.
   값 목록 대화 상자에 개체에 연결된 모든 데이터 값이 표시됩니다.



4. 취소를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.9.3.3 값 목록 만들기

값 목록을 만들려면

1. 개체의 값 목록을 봅니다.
2. 확인을 클릭합니다.
   유니버스 디자인 도구에서는 유니버스 파일이 포함된 폴더의 유니버스 하위 폴더에 값 목록 파일(.LOV)을 저장합니다. 하위 폴더 이름은 .LOV 를 만드는 데 사용된 개체가 포함된 유니버스의 이름과 동일합니다.

.LOV 파일을 만든 후에는 .LOV 파일에 제한된 데이터만 반환되도록 목록을 편집하거나 데이터가 목록에 표시되는 방법을 수정할 수 있습니다.

6.9.4 값 목록 편집

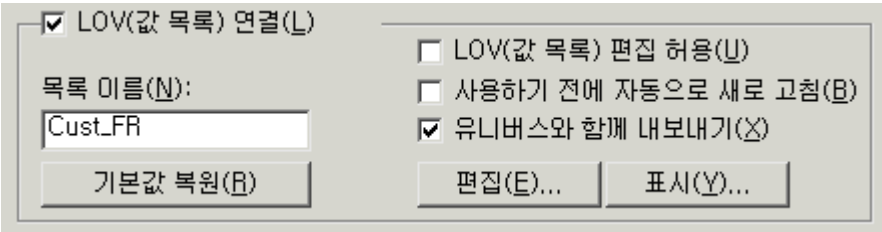
두 가지 방법으로 값 목록의 내용을 수정할 수 있습니다.

- 목록을 생성하는 SELECT DISTINCT 쿼리에 제한을 적용합니다. 예를 들면 예약 손님 수가 최소 수 이상인 휴양지만 반환하도록 휴양지 개체에 대한 휴양지 값 목록을 제한할 수 있습니다.
- 목록에서 값을 간단하게 선택할 수 있도록 계층구조를 만듭니다. 이 방법은 목록에 포함된 값 수가 많은 경우에 유용합니다.

6.9.4.1 값 목록에 조건 적용

값 목록에 조건을 적용하려면

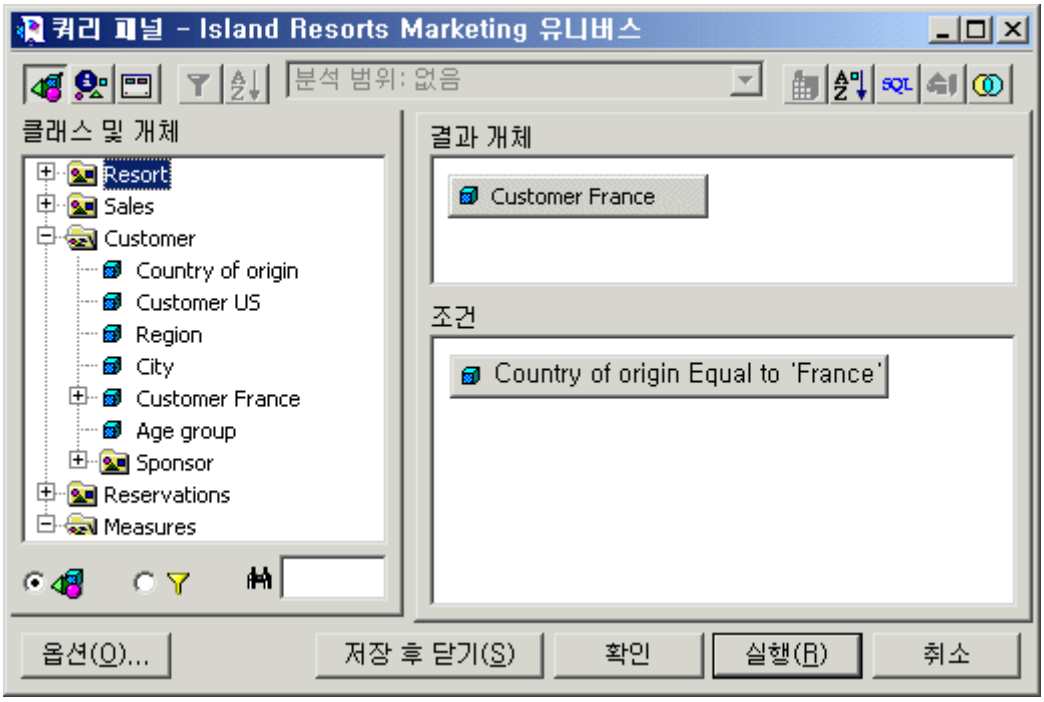
1. 개체를 두 번 클릭합니다.
   개체 속성 편집 시트가 나타납니다.
2. 속성 탭을 클릭합니다.
   속성 페이지가 나타납니다.
3. 값 목록 연결 확인란을 선택합니다.
4. 목록 이름을 바꾸려면 목록 이름 상자에 .LOV 파일 이름을 입력합니다.



5. 편집 단추를 클릭합니다.
   쿼리 창이 나타납니다. 결과 개체 창에 활성 개체가 나열됩니다.
6. 활성 개체의 값 목록에 대한 조건으로 사용할 개체를 조건 창으로 끌어옵니다.
7. 연산자 창에서 연산자를 두 번 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8. 피연산자 창에서 피연산자를 두 번 클릭합니다.
9. 필요한 값을 선택하거나 입력합니다.
   예를 들어, 다음 쿼리는 프랑스 고객만 반환합니다.



10. 확인을 클릭합니다.
11. 표시를 클릭하여 제한된 값 목록을 봅니다.
    빈 목록이 나타납니다.
12. 새로 고침을 클릭합니다.
13. 목록에 값이 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

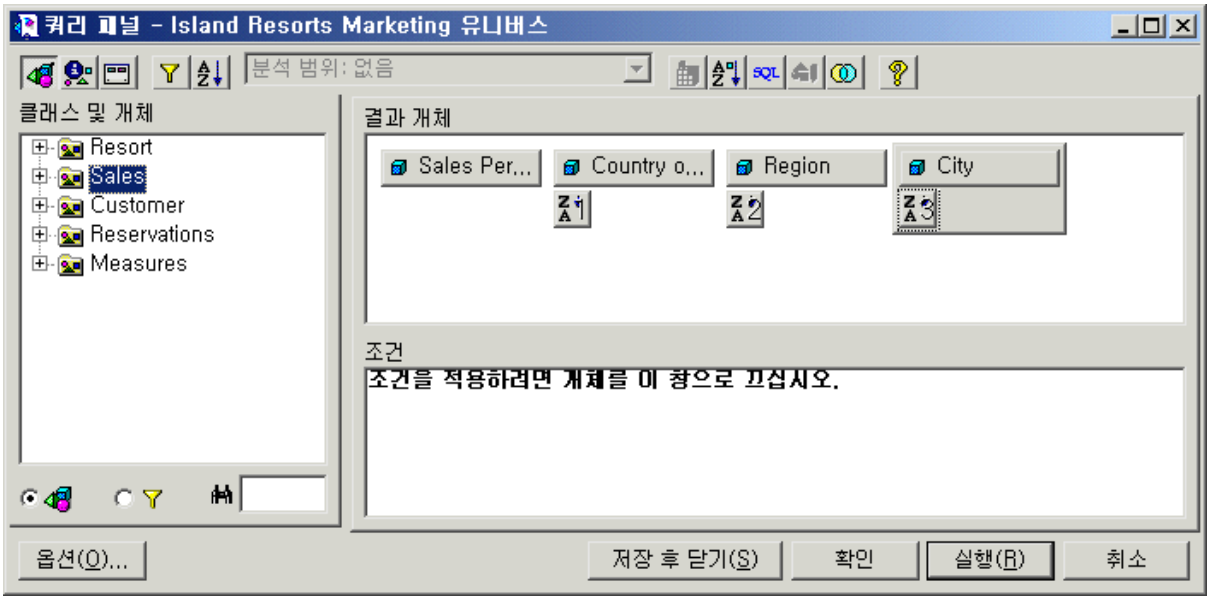
14. 각 대화 상자에서 확인을 클릭합니다.

6.9.4.2 값 목록 계층구조 만들기

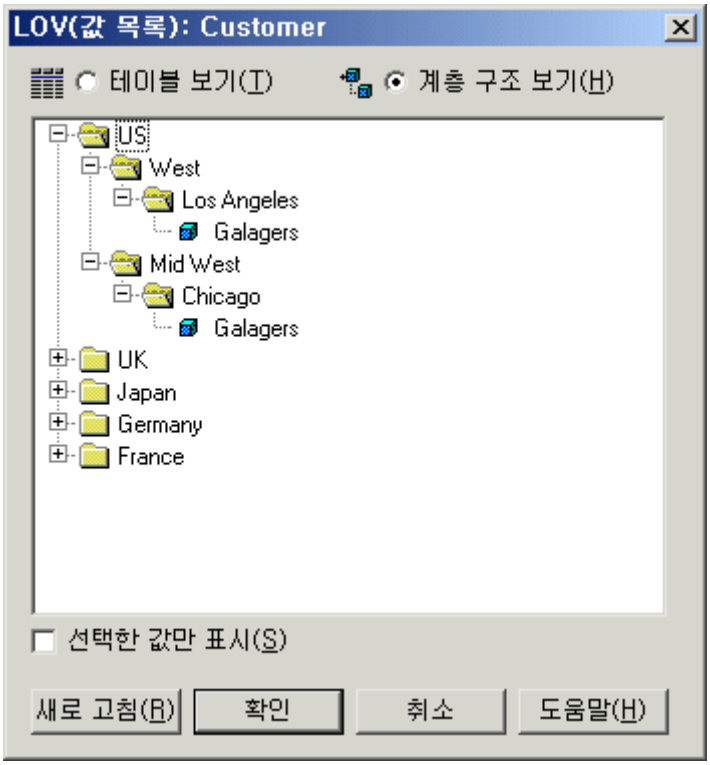
값 목록 계층구조를 만들려면

1. 개체를 두 번 클릭합니다.
   개체 속성 편집 시트가 나타납니다.
2. 속성 탭을 클릭합니다.
   속성 페이지가 나타납니다.
3. 값 목록 연결 확인란을 선택합니다.
4. 목록 이름을 바꾸려면 목록 이름 상자에 .LOV 파일 이름을 입력합니다.
5. 편집 단추를 클릭합니다.
   쿼리 창이 나타납니다. 결과 개체 창에 활성 개체가 나열됩니다.
6. 다음 그림과 같이 계층구조에 표시할 개체를 기존 개체 오른쪽에 있는 결과 개체 상자로 끌어옵니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



7. 확인을 클릭합니다.
8. 표시를 클릭하여 제한된 값 목록을 봅니다.
   빈 목록이 나타납니다.
9. 새로 고침을 클릭합니다.
   목록에 값이 표시됩니다.



10. 각 대화 상자에서 확인을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.9.5 값 목록 내보내기

값 목록을 유니버스와 함께 CMS 로 내보낼 수 있습니다. 파일 시스템에서 유니버스 파일이 저장되는 폴더의 유니버스 하위 디렉터리에 관련 .LOV 파일이 복사됩니다.

6.9.5.1 내보낸 .LOV 파일을 Web Intelligence 에서 사용하는 방법

유니버스 디자인 도구에서 내보낸 .LOV 파일과 연결된 개체를 사용하여 Web Intelligence 에서 쿼리를 실행하면 다음 중 하나를 기준으로 개체의 값 목록이 반환됩니다.

- .LOV 파일에 들어 있는 데이터
- .LOV 파일에 정의되어 있는 SELECT DISTINCT 쿼리의 SQL

유니버스 디자인 도구에서 개체에 대해 반환되는 데이터 값을 제한할 조건을 만든 경우 모든 데이터 값이 포함된 기본 목록 대신 제한된 목록이 표시됩니다. 이 목록에는 유니버스 디자인 도구에서 구현된 조건과 서식이 모두 유지됩니다.

.LOV 파일을 유니버스와 함께 내보내지 않은 경우에는 개체에서 조건이나 서식 없이 기본 목록만 반환됩니다. 이 경우에는 데이터가 저장될 기본 .LOV 파일이 만들어집니다.

6.9.5.2 데이터가 포함되거나 포함되지 않은 목록 내보내기

값 목록을 중앙 관리 서버(CMS) 리포지토리로 내보내는 데는 두 가지 방법을 사용할 수 있습니다.

표 145:

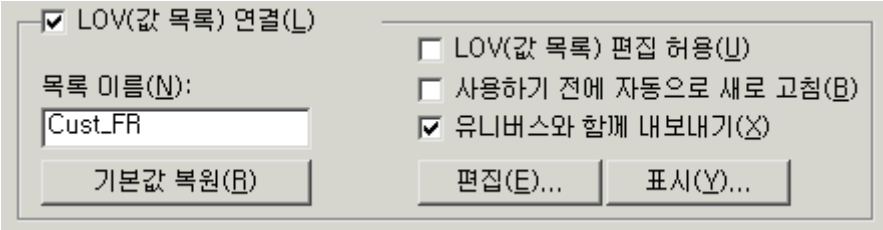
.LOV 내보내기	설명
쿼리 정의만 함께 내보내기(데이터 없음)	목록에 값을 반환하는 SELECT DISTINCT 쿼리의 정의와 함께 .LOV 파일을 내보냅니다. 유니버스 디자인 도구 쿼리 창에서 .LOV 파일에 대해 설정한 모든 조건이 유지됩니다. .LOV 파일에는 데이터가 들어 있지 않으며 값을 반환하기 위해 쿼리 창에서 개체를 처음 사용했을 때 데이터로 채워집니다. 이 방법은 데이터가 정기적으로 업데이트되거나 데이터 목록이 아주 큰 경우에 사용하는 것이 좋습니다.
데이터와 함께 내보내기	유니버스 디자인 도구에서 값 목록을 표시하거나 편집할 때 반환되는 모든 데이터와 함께 .LOV 파일을 내보내거나 가져옵니다. 이 방법은 .LOV 파일의 데이터가 변경되지 않는 경우에 유용합니다. 그러나 데이터가 정기적으로 업데이트되거나 목록에 많은 값이 포함된 경우에는 내보내기 속도가 느려질 수 있으므로 이 방법을 사용하지 않는 것이 좋습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

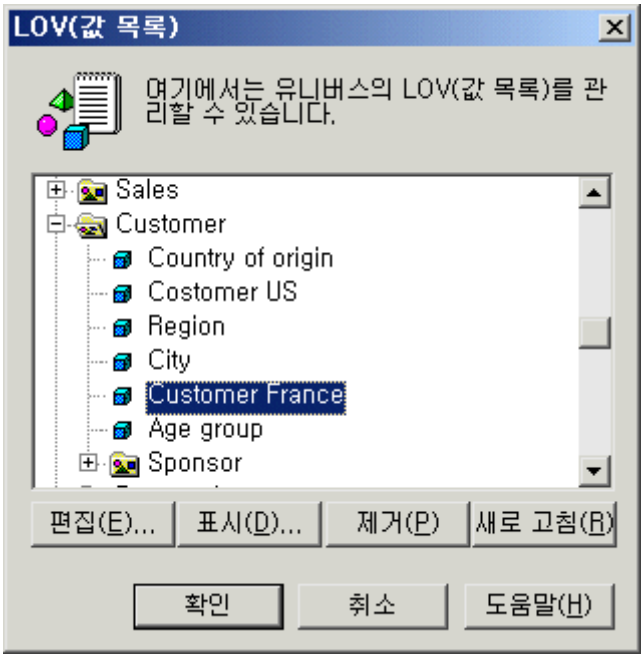
값 목록 정의 내보내기

값 목록 정의(데이터 없음)를 내보내려면

1. 개체의 값 목록을 만듭니다.
2. 개체의 속성 페이지에서 유니버스와 함께 내보내기 확인란을 선택합니다.
   아래에서 고객은 프랑스 고객의 값만 반환하는 Cust_FR 라는 값 목록과 연결되어 있습니다.



3. 도구 > 값 목록을 선택합니다.
   값 목록 대화 상자가 나타납니다. 이 대화 상자에는 현재 유니버스의 클래스와 개체가 나열되고 각 개체의 값 목록을 관리하는 데 필요한 옵션이 표시됩니다.
4. 클래스를 확장한 후 리포지토리로 내보낼 .LOV 파일을 가진 개체를 선택합니다.



5. 제거 단추를 클릭합니다.
   개체의 .LOV 파일에서 데이터가 삭제됩니다. 이렇게 하면 .LOV 파일에는 값 목록에 대한 쿼리 정의만 포함됩니다.
6. 확인을 클릭합니다.
7. 파일 > 내보내기를 선택합니다.
   유니버스 내보내기 상자가 나타납니다.
8. 유니버스 목록에서 유니버스 파일 이름을 선택합니다.
9. 확인을 클릭합니다.
   유니버스 내보내기가 완료되었음을 알려주는 메시지 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

데이터와 함께 값 목록 내보내기

데이터와 함께 값 목록을 내보내려면

1. 개체의 값 목록을 만듭니다.
2. 개체의 속성 페이지에서 유니버스와 함께 내보내기 확인란을 선택합니다.
3. 표시 단추를 클릭합니다.
   값 목록이 나타납니다.
4. 목록이 비어 있으면 새로 고침 단추를 클릭하여 목록을 채웁니다.
5. 각 대화 상자에서 확인을 클릭합니다.
6. 파일 > 내보내기를 선택합니다.
   유니버스 내보내기 상자가 나타납니다.
7. 유니버스 목록에서 유니버스 파일 이름을 선택합니다.
8. 확인을 클릭합니다.
   유니버스 내보내기가 완료되었음을 알려주는 메시지 상자가 나타납니다.

6.9.6 값 목록의 값 새로 고치기

다음과 같은 두 가지 방법으로 유니버스 디자인 도구의 값 목록 데이터를 새로 고칠 수 있습니다.

- 개체의 값 목록을 표시한 다음 새로 고침 단추를 클릭합니다.
- 도구 > 값 목록을 선택하여 값 목록 관리 상자를 표시하고 개체를 선택한 다음 새로 고침 단추를 클릭합니다.

6.9.7 개인 데이터 파일의 데이터 사용

데이터베이스 서버에서 검색한 회사 데이터가 아니라 개인 데이터가 포함된 개체에 대해 값 목록을 지정할 수 있습니다.

개인 데이터는 텍스트 파일 같은 기본 파일에 저장된 데이터이거나 Microsoft Excel, Lotus 1-2-3 또는 dBASE 같은 특정 응용 프로그램에서 가져온 데이터입니다.

개인 데이터 파일을 값 목록으로 사용하면 다음과 같은 이점이 있습니다.

- 개인 데이터 파일에서 데이터를 검색하면 회사 데이터베이스에 액세스할 때보다 속도가 빠릅니다.
- 데이터베이스에 없는 값을 사용할 수 있습니다.
- 사용자가 값 목록을 사용할 때 볼 수 있는 값을 Designer 가 제어할 수 있습니다.

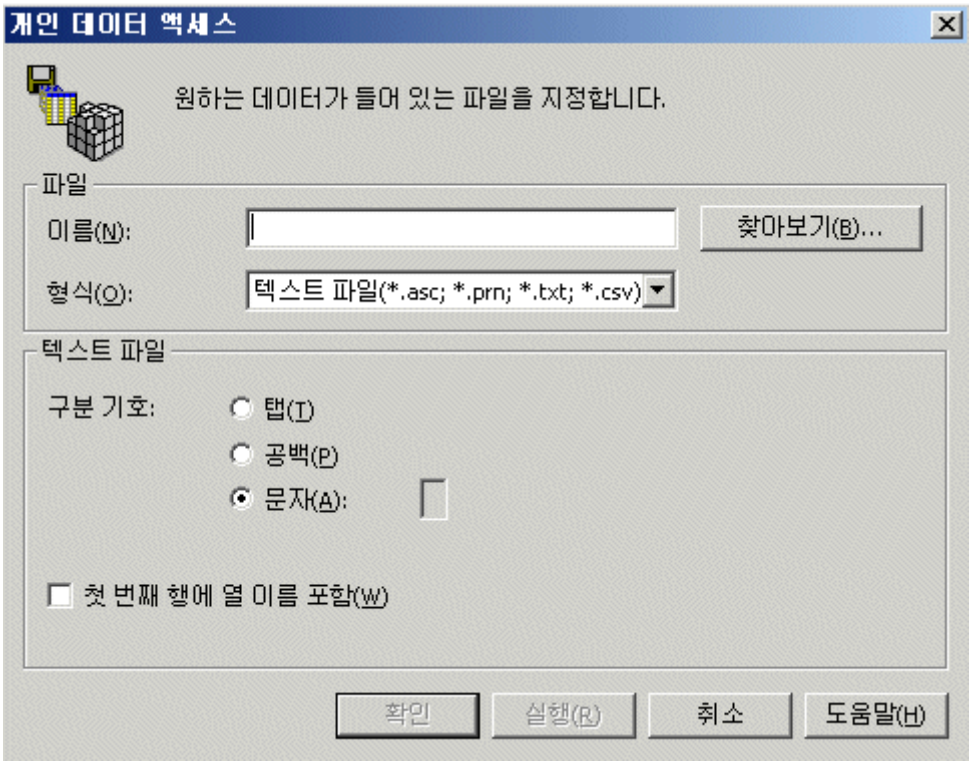
개인 데이터 파일을 사용할 때의 단점은 데이터가 고정되어 있다는 점입니다. 따라서 값을 변경해야 할 때 데이터를 수동으로 업데이트해야 합니다.

6.9.7.1 개인 데이터 파일에서 값 목록 만들기

개인 데이터 파일에서 값 목록을 만들려면

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

1. 도구 > 값 목록을 선택합니다.
   값 목록 대화 상자가 나타납니다.
2. 클래스를 확장한 후 개체를 클릭합니다.
3. 속성 그룹 상자에서 개인 데이터 라디오 단추를 클릭합니다.
   값 목록 유형을 회사에서 개인으로 변경한다고 알려주는 메시지 상자가 나타납니다.
4. 확인을 클릭합니다.
   개인 데이터 액세스 대화 상자가 나타납니다. 사용할 수 있는 옵션은 선택한 파일 형식에 따라 다릅니다.



5. 찾아보기 단추를 클릭한 다음 값 목록으로 사용할 파일을 선택합니다.
   또는
   이름 텍스트 상자에 파일 이름을 입력합니다.
6. 형식 목록 상자에서 파일 형식을 선택합니다.
7. 다음과 같은 형식 중 하나를 선택할 수 있습니다.
   - 텍스트 파일(*.asc; *.prn; *.txt; *.csv)
   - Microsoft Excel 파일
   - dBASE
   - Microsoft Excel 97

   노트

   Excel 97 에서 파일을 만든 경우에는 Microsoft Excel 파일 옵션이 아니라 Microsoft Excel 97 옵션을 사용해야 합니다.

8. 필요에 따라 나머지 옵션을 지정합니다.
   텍스트 파일의 줄 하나는 행 하나와 동등합니다. 텍스트 파일의 경우 탭 문자, 공백 또는 문자 같은 열 구분 기호 유형을 지정합니다. 문자 유형을 선택하면 텍스트 상자에 문자를 입력해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

9. 확인을 클릭합니다.

6.9.8 유니버스의 값 목록 관리

값 목록 대화 상자(도구 > 값 목록)에서는 활성 유니버스의 모든 값 목록을 관리할 수 있습니다. 여기에는 모든 클래스와 개체가 트리 뷰에 표시됩니다. 원하는 개체를 선택하면 해당 값 목록에 액세스할 수 있습니다. 값 목록 대화 상자에서는 다음과 같은 작업을 수행할 수 있습니다.

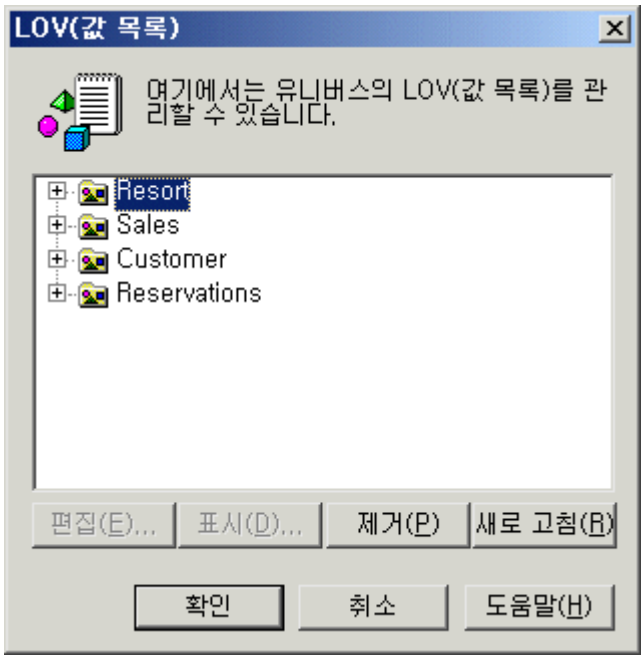
표 146:

옵션	설명
편집	선택한 개체에 대한 쿼리를 정의할 수 있는 쿼리 창을 표시합니다. 값 목록에 대해 기존 쿼리를 편집하거나 정의할 수 있습니다.
표시	선택한 개체의 현재 값 목록을 표시합니다.
제거	선택한 개체에 현재 지정되어 있는 값 목록의 내용을 지웁니다.
새로 고침	표시된 값 목록을 새로 고칩니다.

6.9.8.1 값 목록 관리 도구에 액세스

값 목록 관리 도구에 액세스하려면

1. 도구 > 값 목록 > 값 목록 편집을 선택합니다.
   값 목록 대화 상자가 나타납니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 클래스를 확장한 후 개체를 선택합니다.
3. 관리 작업을 수행하는 단추를 클릭하거나 옵션을 선택합니다.
4. 확인을 클릭합니다.

6.9.9 LOV 파일 최적화 및 사용자 지정

다음과 같은 일반적인 방법으로 LOV 를 최적화하고 사용자 지정할 수 있습니다.

표 147:

메서드	설명
LOV 가 더 작은 테이블을 가리키도록 지정	기본적으로 LOV 는 연결된 대상 개체를 가리킵니다. 그러나 이 개체가 행 수가 많은 큰 테이블을 가리키면 LOV 를 새로 고치는 데 시간이 오래 걸릴 수 있습니다. 같은 값을 반환하지만 크기가 더 작고 속도도 빠른 대체 테이블이 있는 경우에는 이 대체 테이블을 가리키도록 LOV 를 편집해야 합니다.
코드와 설명 조합	일반적으로 '코드'와 '설명'을 조합하는 방법으로 .LOV 를 사용자 지정합니다. 개체에서 '판매 유형 코드'를 반환하는 경우 일부 사용자는 이 값을 쉽게 이해하지 못할 수 있습니다. '판매 유형 설명'을 표시하도록 LOV 를 편집하면 사용자가 LOV 를 보는 데 도움이 될 수 있습니다. 반대로 '판매 유형 설명' 개체에 코드와 설명을 함께 표시할 수도 있습니다.

6.10 유니버스 연결

하나 이상의 유니버스를 동적으로 연결할 수 있습니다.

6.10.1 연결된 유니버스란?

연결된 유니버스란 매개 변수, 클래스, 개체 또는 조인과 같은 공통 구성 요소를 공유하는 유니버스를 말합니다.

두 유니버스를 연결하면 한 유니버스는 코어 유니버스가 되고 다른 유니버스는 파생 유니버스가 됩니다. 코어 유니버스에서 변경 작업이 수행될 경우 이러한 변경 내용은 파생 유니버스로 자동 전파됩니다.

> **노트**
>
> 연결된 유니버스 배포에 대한 내용은 파생 유니버스 및 값 목록 [페이지 319] 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.10.1.1 코어 유니버스란?

코어 유니버스란 다른 유니버스가 연결된 유니버스를 말합니다. 코어 유니버스에는 여기에 연결된 다른 유니버스와 공통되는 구성 요소가 포함됩니다. 이런 연결된 유니버스를 파생 유니버스라고 합니다. 코어 유니버스는 재사용 가능한 구성 요소 라이브러리를 나타냅니다.

코어 유니버스는 코어 유니버스 구성 요소가 파생 유니버스에서 사용되는 방식에 따라 커널 또는 마스터 유니버스가 될 수 있습니다. 커널 및 마스터 유니버스는 두 유니버스 사이에 링크 만들기 [페이지 314] 단원에서 설명합니다.

6.10.1.2 파생 유니버스란?

파생 유니버스란 코어 유니버스에 대한 링크를 포함하는 유니버스를 말합니다. 이 링크를 통해 파생 유니버스는 코어 유니버스의 공통 구성 요소를 공유할 수 있습니다.

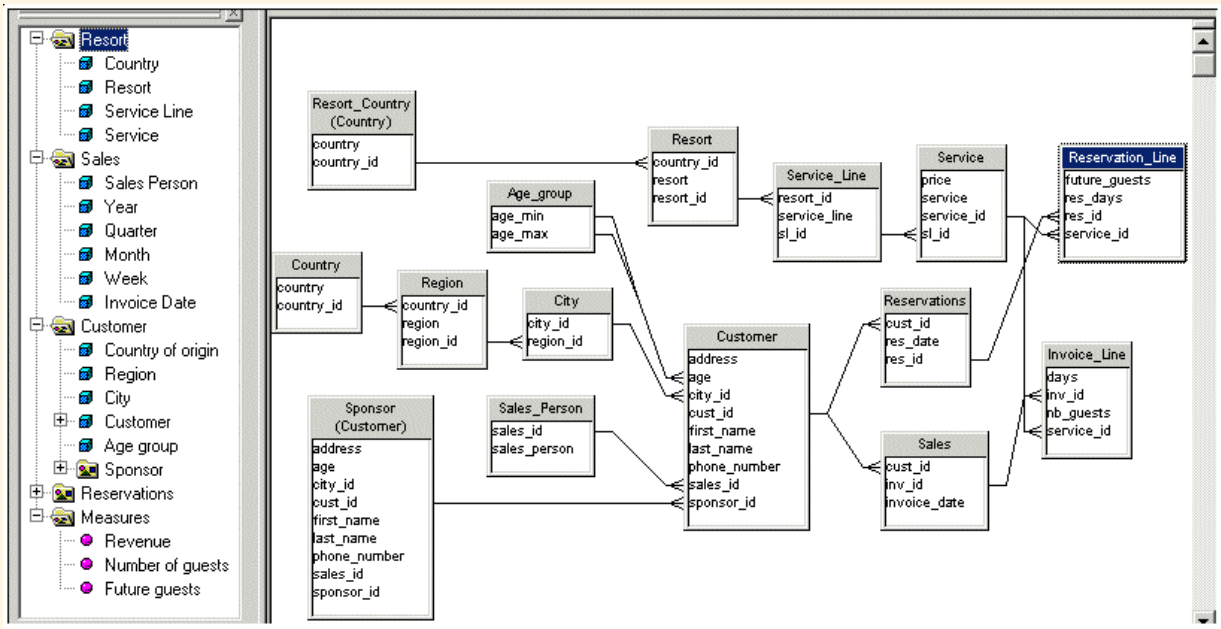
- 연결된 주요 유니버스가 커널 유니버스인 경우 구성 요소가 파생 유니버스에 추가될 수 있습니다.
- 연결된 코어 유니버스가 마스터 유니버스인 경우 연결된 유니버스가 모든 코어 유니버스 구성 요소를 포함할 수 있습니다. 클래스 및 개체는 파생 유니버스에 추가되지 않습니다. 이러한 구성 요소는 대상 사용자의 필요에 따라 파생 유니버스에서 숨길 수 있습니다.

예

연결된 코어 및 파생 유니버스

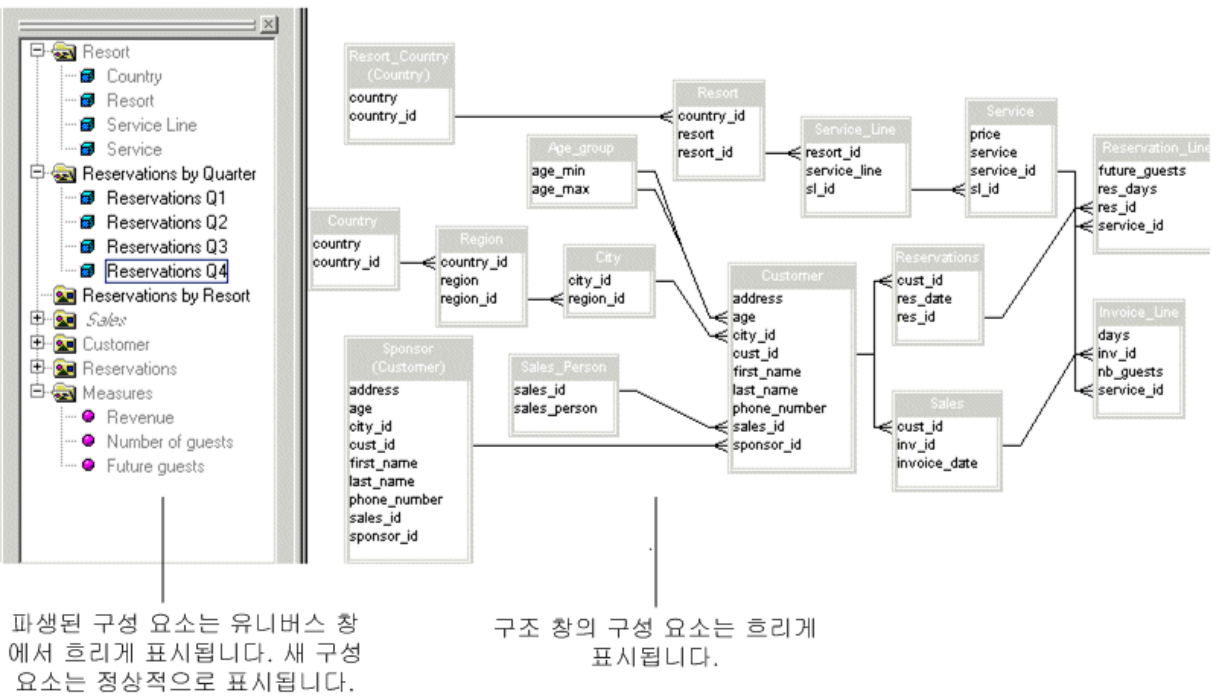
다음 예제에는 두 개의 연결된 유니버스가 있습니다. 하나는 공통 구성 요소를 포함하고 있는 코어 유니버스이고 다른 하나는 핵심 구조를 사용하지만 자체적으로 새로운 클래스와 개체도 가지고 있는 파생 유니버스입니다.

Beach.unv 는 코어 유니버스입니다. 이 유니버스는 Island Resorts 의 영업 관리자가 마케팅 분석을 수행하는 데 사용됩니다. 이 유니버스는 이 버전에 제공되는 데모 유니버스 중 하나입니다. 유니버스의 내용은 다음과 같습니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

이 코어 유니버스를 사용하여 영업 관리자는 예약에 초점을 맞춘 파생 유니버스를 만듭니다.



코어 유니버스에서 파생된 구성 요소는 흐리게 나타납니다. 관리자는 분기별 예약과 휴양지별 예약이라는 두 개의 새로운 클래스를 만들었습니다. 이러한 클래스와 해당 개체는 정상적으로 표시됩니다. 영업 관리자는 또한 예약 유니버스에서 필요하지 않은 판매 클래스를 숨기기로 결정했습니다. 코어 유니버스에 대한 모든 변경 내용은 파생 유니버스로 자동 전파됩니다.

6.10.2 유니버스를 연결하는 여러 가지 방법

유니버스를 연결할 때 다음과 같은 방법을 사용할 수 있습니다.

- 커널 방식
- 마스터 방식
- 구성 요소 방식
  이러한 세 가지 방식을 각각 따로 사용하거나 하나 이상을 결합하여 사용할 수 있습니다.

6.10.2.1 커널 방식

커널 방식에서는 하나의 유니버스가 핵심 구성 요소를 포함합니다. 핵심 구성 요소는 모든 유니버스에서 공통되는 구성 요소입니다. 커널 유니버스에서 만드는 파생 유니버스는 이러한 핵심 구성 요소와 함께 고유한 특정 구성 요소를 포함합니다.

커널 유니버스에 대한 모든 변경 내용은 모든 파생 유니버스의 핵심 구성 요소에 자동으로 반영됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.10.2.2 마스터 방식

마스터 방식은 연결된 유니버스의 공통 구성 요소를 구성하는 또 다른 방법입니다.

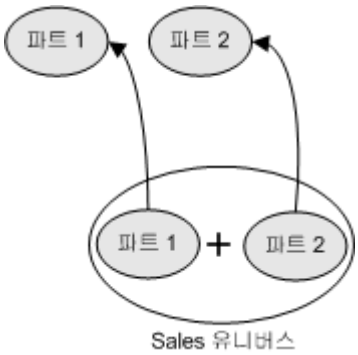
마스터 유니버스는 가능한 모든 구성 요소를 포함합니다. 마스터에서 파생된 유니버스의 경우 대상 사용자에 대한 관련성에 따라 일부 구성 요소를 숨길 수 있습니다.

파생 유니버스에서 보이는 구성 요소는 언제나 마스터 유니버스의 하위 집합입니다. 파생 유니버스에는 새 구성 요소가 별도로 추가되지 않습니다. 아래 예제는 마스터 유니버스에서 파생된 Human Resources 및 Sales 유니버스를 보여 줍니다. 이러한 유니버스에는 마스터 유니버스의 구성 요소가 포함되고 이 중 일부는 숨길 수 있습니다.

마스터 유니버스에 대한 모든 변경 내용은 모든 파생 유니버스의 핵심 구성 요소에 자동으로 반영됩니다.

6.10.2.3 구성 요소 방식

구성 요소 방식은 둘 이상의 유니버스를 하나의 유니버스로 병합하는 방식입니다. 아래의 Sales 유니버스는 부분 1 과 부분 2 의 두 유니버스를 병합하여 만들어졌습니다.



6.10.3 유니버스 링크 이점

유니버스를 연결할 경우 다음과 같은 이점이 있습니다.

- 개발 및 유지 관리 시간이 단축됩니다. 주요 유니버스의 구성 요소를 수정하면 유니버스 디자인 도구에서는 모든 파생 유니버스의 동일한 구성 요소로 변경 내용을 전파합니다.
- 자주 사용되는 구성 요소를 코어 유니버스에 모아 두면 새로운 모든 유니버스에 이러한 구성 요소를 포함시킬 수 있습니다. 새 유니버스를 만들 때마다 공통 구성 요소를 다시 만들 필요가 없습니다.
- 전문화가 용이해집니다. 기본적인 코어 유니버스를 설정하는 데이터베이스 관리자와 자신의 특정 분야를 기반으로 좀 더 기능적인 유니버스를 만드는 전문화된 디자이너로 개발 작업을 나눌 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.10.4 유니버스 링크 요구 사항

다음 요구 사항이 충족되는 경우에만 활성 유니버스를 코어 유니버스에 연결할 수 있습니다.

- 코어 유니버스와 파생 유니버스가 동일한 데이터 계정 또는 데이터베이스 및 동일한 RDBMS 를 사용합니다. 코어 유니버스와 파생 유니버스 모두에 대해 동일한 연결을 사용하면 유니버스 관리는 쉬워지지만 언제든지 변경될 수 있는 단점이 있습니다.
- 코어 유니버스와 파생 유니버스는 동일한 리포지토리에 있어야 합니다.
- 코어 유니버스에 대해 적어도 한 번 내보내기 및 다시 가져오기가 수행되었습니다. 파생 유니버스는 링크를 만들기 전에 내보낼 필요가 없습니다.
- 내보낸 파생 유니버스가 코어 유니버스와 동일한 유니버스 도메인에 있습니다.
- 해당 유니버스를 연결할 권한이 있습니다.

6.10.5 유니버스 링크 제한 사항

유니버스를 연결할 때 다음과 같은 제한 사항을 알고 있어야 합니다.

- 저장 프로시저를 사용하는 유니버스에는 연결할 수 없습니다.
- 한 수준의 링크만 사용할 수 있습니다. 파생된 유니버스에서 다시 파생 유니버스를 만들 수 없습니다.
- 모든 클래스와 개체는 코어 유니버스와 파생 유니버스 모두에서 고유합니다. 그렇지 않을 경우 충돌이 발생합니다.
- 두 유니버스의 구조는 한 유니버스의 테이블과 다른 유니버스의 테이블 사이에 조인을 만들 수 있는 구조여야 합니다. 그렇지 않을 경우 두 구조의 개체로 쿼리를 실행할 때 카티전 곱이 발생할 수 있습니다.
- 파생 유니버스에서는 코어 유니버스의 테이블 스키마, 클래스 및 개체만 사용할 수 있습니다. 파생 유니버스에서 컨텍스트가 다시 검색되어야 합니다.
- 코어 유니버스 구조로 파생 유니버스를 내보낼 때 코어 유니버스와 연관된 값 목록은 저장되지 않습니다.

6.10.6 두 유니버스 사이에 링크 만들기

활성 유니버스를 다른 유니버스에 연결할 수 있습니다. 이렇게 하면 활성 유니버스는 파생 유니버스가 되고 연결된 유니버스는 코어 유니버스가 됩니다. 코어 유니버스의 구성 요소는 파생 유니버스에서 상속됩니다.

유니버스를 코어 유니버스에 링크하려면 코어 유니버스를 리포지토리로 내보내야 합니다.

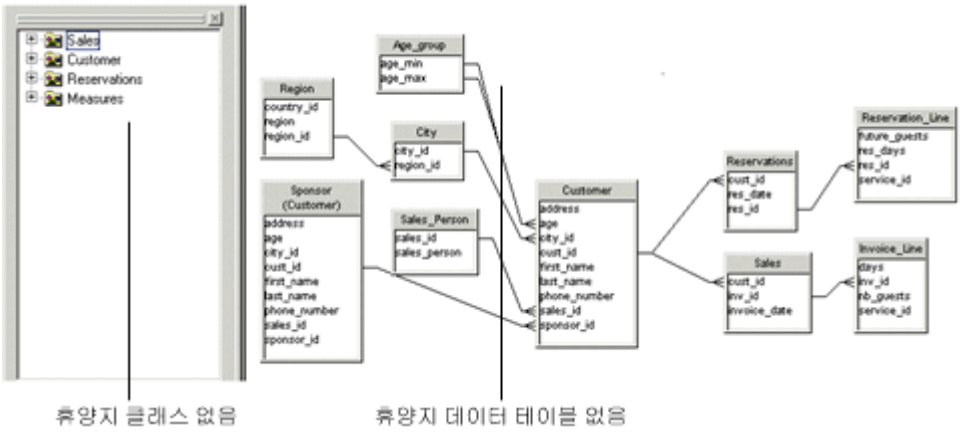
> **노트**
>
> 유니버스를 연결할 때 동일한 리포지토리 내에서 다른 연결을 사용하여 코어 유니버스를 위치 변경할 수 있습니다. 이렇게 하면 파생 유니버스와의 연결을 계속 유지한 상태에서 코어 유니버스를 다른 리포지토리 폴더에 내보낼 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.10.6.1 파생 유니버스 사이에 링크 만들기

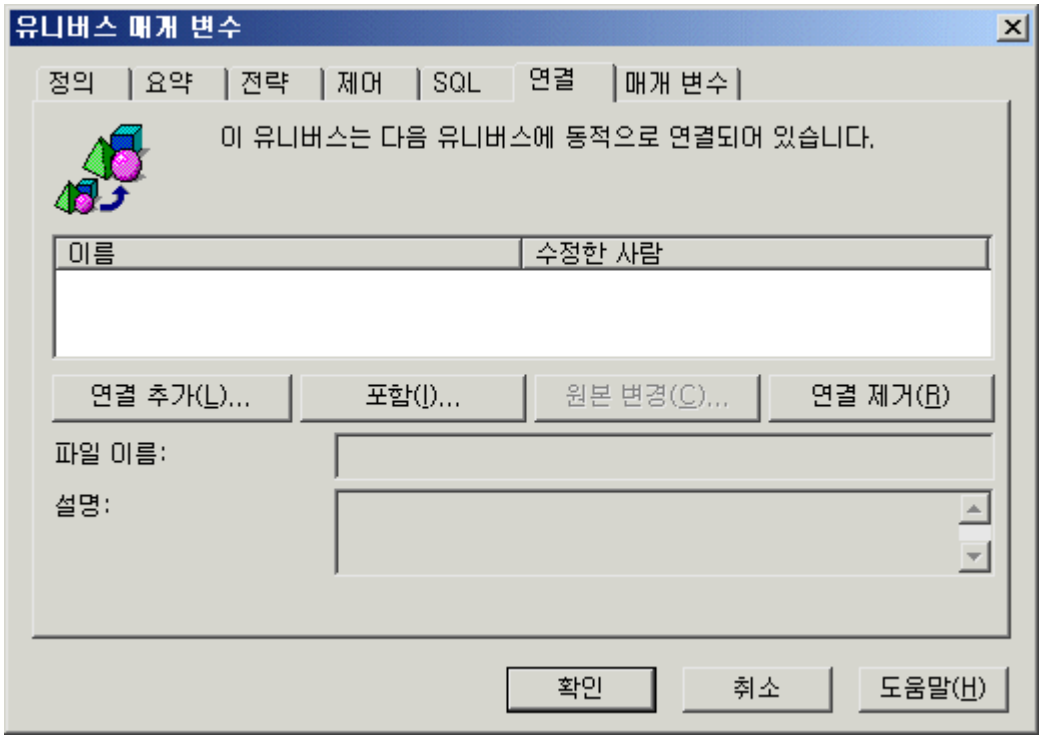
1. 활성 유니버스가 코어 유니버스에 링크하려는 유니버스인지 확인합니다.

   예를 들어, 다음 유니버스는 휴양지 데이터는 없고 국가별 판매 정보만 포함되어 있는 Beach 유니버스입니다. 이 Sales 유니버스를 휴양지 데이터가 포함된 Resort 유니버스와 링크하려고 합니다. 다음에서 Sales Beach 유니버스는 파생 유니버스이고 Resort 유니버스는 코어 유니버스입니다.



2. 편집 > 링크를 선택합니다.

   유니버스 매개 변수 대화 상자에 링크 페이지가 나타납니다.



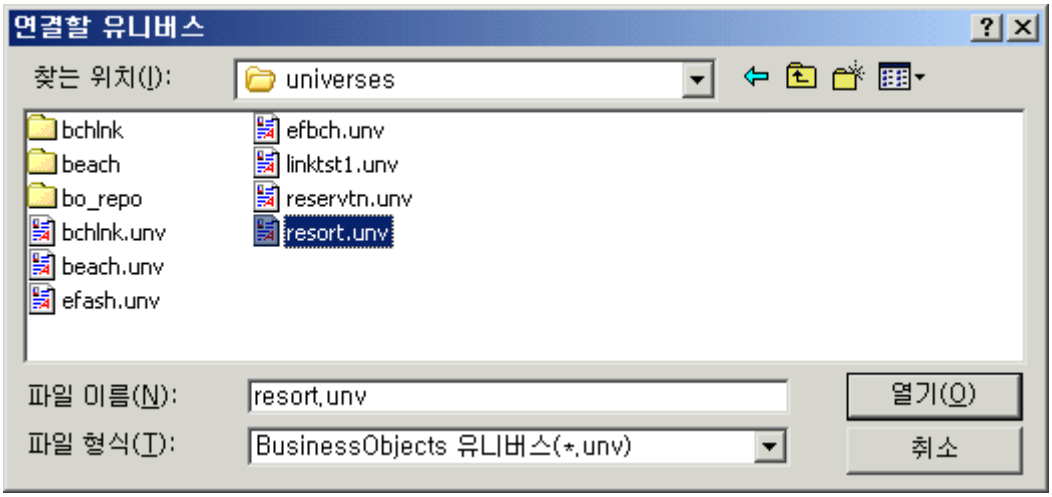
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. 링크 추가 단추를 클릭합니다.

   연결할 유니버스 대화 상자가 나타납니다. 여기에는 사용 가능한 도메인의 유니버스가 나열됩니다.

4. 연결할 유니버스를 찾습니다. 이 유니버스는 활성 유니버스에서 사용하려는 구성 요소를 포함하고 있는 코어 유니버스입니다.

   예제에서 Resort 유니버스를 선택합니다.

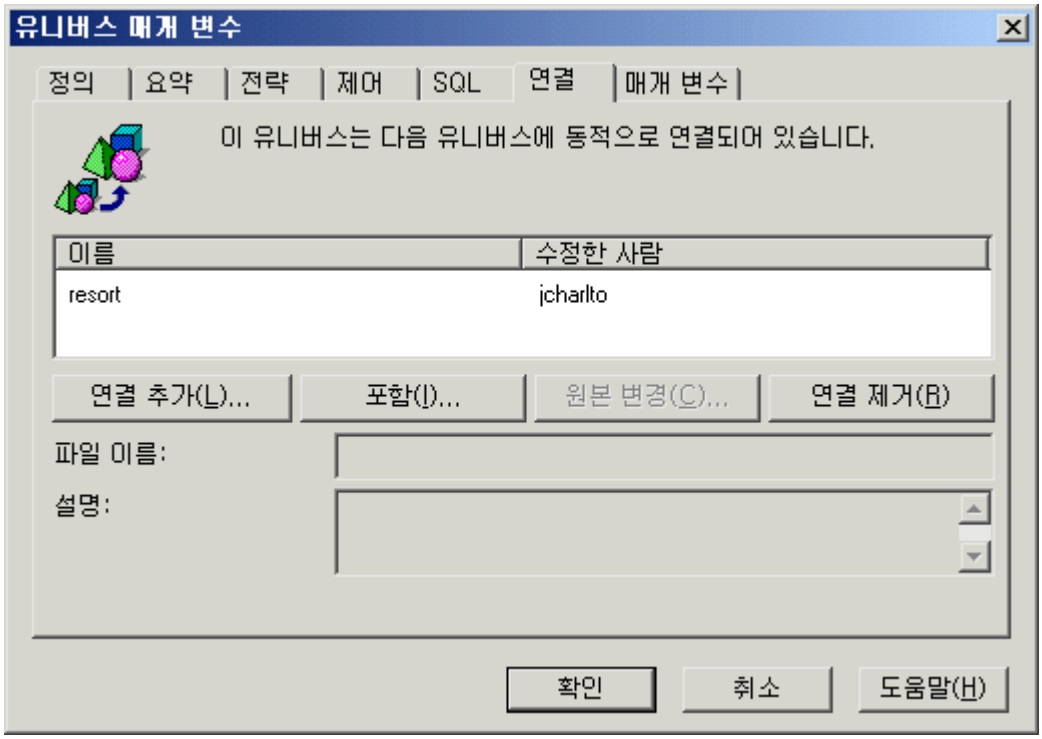


   선택한 유니버스를 내보낸 적이 없는 경우 오류 메시지가 나타납니다. 연결할 수 있으려면 먼저 유니버스를 내보내야 합니다.

5. 열기 단추를 클릭합니다.

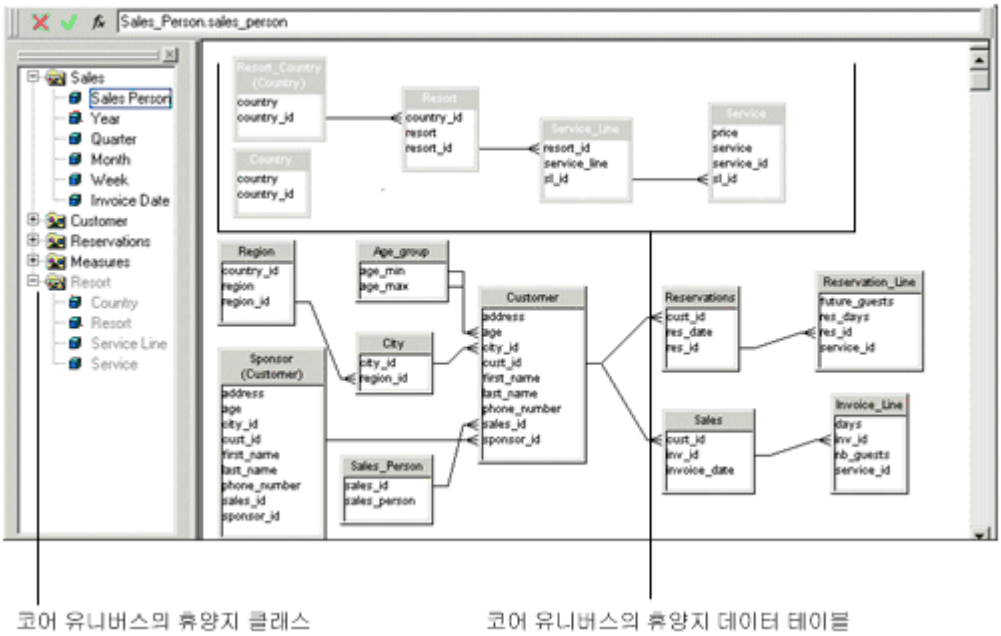
   선택한 유니버스가 목록에 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



6. 확인을 클릭합니다.

   링크가 만들어졌습니다. 코어 유니버스는 활성 유니버스 내에 흐리게 표시됩니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.10.7 파생 유니버스 편집

코어 테이블과 파생 유니버스 테이블 사이에 조인을 만들면 링크 프로세스가 완료됩니다. 현재의 모든 컨텍스트를 삭제하고 컨텍스트를 새로운 구조에 대해 다시 검색해야 합니다.

> **노트**
>
> 파생 유니버스 내에서는 연결된 유니버스(코어 유니버스)의 구조, 클래스 또는 개체를 편집할 수 없습니다.

6.10.7.1 파생 유니버스 편집

파생 유니버스를 편집하려면

1. 코어 유니버스와 파생 유니버스 구조 사이에 조인을 만듭니다.
   조인을 만들면 쿼리에 포함된 두 구조 모두의 개체에 대해 카티전 곱이 반환되지 않습니다.
2. 기존 컨텍스트를 제거합니다.
3. 별칭을 검색합니다.
4. 컨텍스트를 검색합니다.
5. 필요에 따라 개체를 숨기거나 새로 만듭니다.

> **노트**
>
> 구성 요소 숨기기에 대한 내용은 클래스, 개체 및 조건 표시 또는 숨기기 [페이지 246] 단원을 참조하십시오.

6.10.8 링크 제거

파생 유니버스에 핵심 구성 요소를 기반으로 하는 개체나 핵심 구성 요소에 대한 조인이 포함되어 있지 않은 경우에만 코어 유니버스 링크를 제거할 수 있습니다.

6.10.8.1 파생 유니버스에서 링크 제거

파생 유니버스에서 링크를 제거하려면

1. 파생 유니버스를 엽니다.
2. 편집 > 링크를 선택합니다.
   유니버스 매개 변수 대화 상자의 링크 페이지가 나타납니다.
3. 목록에서 코어 유니버스의 이름을 클릭합니다.
4. 링크 제거 단추를 클릭합니다.
5. 확인을 클릭합니다.
   코어 유니버스의 구성 요소가 활성 유니버스에서 제거됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.10.9 코어 유니버스 위치 변경

코어 유니버스의 위치가 변경된 경우 링크를 유지하기 위해서는 새 위치를 설정해야 합니다.

6.10.9.1 위치가 변경된 코어 유니버스로 링크 업데이트

위치가 변경된 코어 유니버스로 링크를 업데이트하려면

1. 파생 유니버스를 엽니다.
2. 편집 > 링크를 선택합니다.
3. 목록에서 연결된 코어 유니버스를 클릭합니다.
4. 소스 변경 단추를 클릭합니다.
   연결할 유니버스 대화 상자가 나타납니다.
5. 코어 유니버스의 새 위치를 찾습니다.
6. 열기 단추를 클릭합니다.
   새 코어 유니버스가 링크 목록에 나타냅니다.

6.10.10 파생 유니버스 및 값 목록

파생 유니버스를 리포지토리로 내보낼 때 핵심 개체와 연결된 값 목록은 파생 유니버스와 함께 저장되지 않습니다.

핵심 개체와 연결된 값 목록을 저장하는 데 사용할 수 있는 한 가지 방법은 다음과 같습니다.

1. 파생 유니버스와 함께 리포지토리로 내보낼 값 목록이 포함된 개체와 동일한 정의를 사용하여 새 개체를 만듭니다.
2. 새 개체에 핵심 개체와 동일한 값 목록을 지정합니다.
3. 이러한 새 개체를 숨깁니다.
   숨겨진 개체는 값 목록을 포함하는 기능을 수행하므로 파생 유니버스와 함께 내보내고 가져올 수 있습니다.

6.10.11 코어 유니버스의 순서로 개체 표시

기본적으로 파생 유니버스의 개체가 정렬된 순서는 나중에 코어 유니버스에서 순서가 변경되더라도 유니버스의 사용자들이 계속 보게 됩니다. 파생 유니버스에서 언제나 코어 유니버스에 표시된 순서대로 개체가 표시되도록 하려면 사용하는 각 데이터베이스의 *.PRM 파일에서 매개 변수를 설정해야 합니다.

매개 변수 설정은 CORE_ORDER_PRIORITY = Y 입니다.

관련 *.PRM 파일에서 매개 변수를 설정하는 방법에 대한 자세한 내용은 Data Access 가이드(도움말 > Data Access 가이드)를 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.11 다른 유니버스 안에 유니버스 포함

코어 유니버스의 구성 요소를 파생 유니버스에 복사할 수 있습니다. 파생 유니버스에서 이러한 구성 요소는 코어 유니버스의 구성 요소와 독립적입니다. 이러한 구성 요소는 코어 유니버스와 링크되지 않습니다. 코어 유니버스의 변경 내용이 파생 유니버스에서 상속되지 않습니다.

6.11.1 코어 유니버스를 파생 유니버스에 복사

코어 유니버스를 파생 유니버스에 복사할 때 파생 유니버스에서의 이러한 구성 요소는 코어 유니버스의 구성 요소와 독립적입니다. 이러한 구성 요소는 코어 유니버스와 링크되지 않습니다. 코어 유니버스의 변경 내용이 파생 유니버스에서 상속되지 않습니다.

다음과 같은 이유로 코어 유니버스를 파생 유니버스로 복사할 수 있습니다.

- 주어진 유니버스의 내용을 활성 유니버스에 복사
- 두 유니버스 사이에 더 이상 동적 링크를 유지하지 않음

> **노트**
>
> 이 작업 이전에 두 유니버스가 연결된 경우 이 작업을 수행하면 활성 유니버스의 동적 링크 구성 요소가 제거되고 더 이상 외부 유니버스와 동적으로 링크되지 않습니다.

6.11.1.1 코어 유니버스를 파생 유니버스에 복사

코어 유니버스를 파생 유니버스에 복사하려면

1. 유니버스를 엽니다.
2. 편집 > 링크를 선택합니다.
   유니버스 매개 변수 대화 상자의 링크 페이지가 나타납니다.
3. 링크 추가 단추를 클릭합니다.
   연결할 유니버스 대화 상자가 나타납니다. 여기에는 사용 가능한 도메인의 유니버스가 나열됩니다.
4. 복사할 유니버스를 찾아 선택합니다. 이 유니버스는 활성 유니버스에서 사용하려는 구성 요소를 포함하고 있는 코어 유니버스입니다.
5. 포함 단추를 클릭합니다.
6. 확인을 클릭합니다.
   코어 유니버스의 구성 요소가 활성 유니버스 안에 표시됩니다.

6.12 저장 프로시저 유니버스 만들기

저장 프로시저 유니버스는 Web Intelligence 사용자가 데이터베이스에 있는 저장 프로시저에 액세스할 수 있도록 하는 특수한 유니버스입니다. 이 유니버스는 Web Intelligence 사용자가 저장 프로시저에 액세스할 수 있는 유일한 방법입니

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

다. Web Intelligence 사용자는 저장 프로시저 유니버스를 사용하여 데이터베이스의 저장 프로시저를 기반으로 하는 보고서를 만들 수 있습니다.

저장 프로시저는 컴파일된 SQL 프로그램이며 대상 데이터베이스에 저장되어 실행되는 하나 이상의 SQL 문으로 구성됩니다.

Web Intelligence 는 저장 프로시저를 기반으로 하는 Desktop Intelligence 에서 만들어진 보고서를 열 수 없습니다. 즉, Web Intelligence 사용자는 저장 프로시저에 액세스하고 보고서를 만들려면 특정 저장 프로시저 유니버스를 사용해야 합니다.

저장 프로시저는 다음과 같은 이점을 제공합니다.

- 저장 프로시저는 코드를 캡슐화합니다. 데이터베이스 작업이 응용 프로그램 소스 전체에 걸쳐 여러 번 나타나지 않고 저장 프로시저에서 한 번만 나타납니다. 따라서 디버깅 및 유지 관리 효율이 향상됩니다.
- 데이터베이스 스키마를 변경하면 한 위치에 있는 소스 코드인 저장 프로시저에만 영향을 미칩니다. 즉, 스키마 변경 작업은 코드 수정이 아닌 데이터베이스 관리 작업이 됩니다.
- 저장 프로시저가 서버에 위치하므로 강력한 보안 제한을 설정할 수 있습니다. 따라서 잘 보호된 저장 프로시저에 대해 신뢰할 수 있는 사용 권한을 유지할 수 있습니다.
- 저장 프로시저가 클라이언트 응용 프로그램 외부에서 컴파일 및 저장되므로 암호나 개인 데이터와 같은 민감한 변수를 SQL 구문에 사용할 수 있습니다.
- 저장 프로시저를 사용하면 네트워크 트래픽이 감소합니다.

BusinessObjects XI Release 3.0 에서는 Desktop Intelligence 및 유니버스 디자인 도구의 유니버스에서 저장 프로시저를 사용할 수 있습니다. Crystal Reports 및 Web Intelligence 에 대한 저장 프로시저를 포함하는 유니버스를 사용할 수도 있습니다.

저장 프로시저 유니버스에는 다음과 같은 제한이 적용됩니다.

- 저장 프로시저 유니버스의 개체 간에는 조인이 허용되지 않습니다.
- 저장 프로시저 유니버스에는 필터를 사용할 수 없습니다.
- 저장 프로시저 유니버스를 표준 유니버스에 연결할 수 없습니다.
- Web Intelligence 감독자가 저장 프로시저가 위치한 데이터베이스 또는 계정에 대한 액세스 권한을 부여합니다.
- 모든 RDBMS 에서 저장 프로시저를 지원하는 것은 아닙니다. 지원 여부는 해당 데이터베이스 가이드를 참조하십시오.
- 저장 프로시저에 포함된 COMPUTE, PRINT, OUTPUT 또는 STATUS 문은 실행되지 않습니다.

보고서에서 저장 프로시저를 사용하는 데 대한 자세한 내용은 *Desktop Intelligence* 가이드를 참조하십시오.

6.12.1 Java bean 유니버스의 저장 프로시저

BusinessObjects XI Release 3.0 은 Java bean 을 기반으로 유니버스를 만들 수 있도록 지원합니다. Java bean 을 기반으로 하는 유니버스는 유니버스 엔터티 관계를 구축하는 데 사용되는 결과 집합을 반환합니다.

Java bean 을 기반으로 하는 유니버스는 저장 프로시저를 기반으로 하는 유니버스와 동일한 워크플로를 사용하며 동일한 이점을 제공합니다. 또한 다음과 같은 공통된 제한점이 적용됩니다.

- 조인이 허용되지 않습니다.
- 유니버스에 필터를 사용할 수 없습니다.

Java bean 에 액세스 방법에 대한 자세한 내용은 *Data Access 가이드*를 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.12.2 저장 프로시저를 기반으로 유니버스 만들기

Business Objects 에서는 다음 저장 프로시저를 지원합니다.

- 매개 변수 없는 저장 프로시저
- 매개 변수(IN)
- 다중 결과 집합이 있는 저장 프로시저
- 다중 문(SELECT 와 다른 SQL 문장)이 있는 저장 프로시저

매개 변수가 있는 저장 프로시저를 만들려면 빠른 디자인 마법사를 클릭하고 매개 변수가 있는 저장 프로시저로 유니버스 만들기에 설명된 단계를 따르십시오.

관련 정보

저장 프로시저를 사용하려면 [페이지 322]
입력 매개 변수를 사용하여 저장 프로시저를 기반으로 하는 유니버스 만들기 [페이지 323]
다중 결과 집합이 있는 저장 프로시저 [페이지 324]

6.12.2.1 저장 프로시저의 클래스 및 개체

- 유니버스 디자인 도구에서는 선택된 저장 프로시저마다 테이블을 하나씩 생성(결과 집합이 여러 개인 경우 테이블을 여러 개 생성)하고 저장 프로시저에서 반환한 열마다 개체를 하나씩 생성합니다.
- 결과 집합 구조는 사용자가 함수를 설명할 때 결정됩니다.

6.12.2.2 저장 프로시저를 사용하려면

매개 변수가 없는 하나 또는 여러 개의 저장 프로시저를 기반으로 유니버스를 만들 수 있습니다. 도구 모음에서 빠른 디자인 마법사를 사용합니다.

1. 빠른 디자인 마법사 도구 모음 단추를 클릭합니다.
   시작 창이 나타납니다.
2. 창 맨 아래의 저장 프로시저 유니버스를 선택하려면 여기를 클릭하십시오. 확인란을 클릭합니다.
3. 시작을 클릭합니다.
   유니버스 매개 변수 정의 패널이 나타납니다.
4. 유니버스 이름 입력 필드에 유니버스 이름을 입력합니다.
5. 데이터베이스 연결 선택 목록의 드롭다운 목록에서 데이터베이스 연결을 선택합니다.
6. 다음을 클릭합니다.
   초기 클래스 및 개체 만들기 패널이 나타납니다.
7. 저장 프로시저를 클릭합니다.
8. 추가를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

저장 프로시저가 유니버스 클래스 및 개체 창에서 만들어집니다.

9. 다음을 클릭합니다.
10. 마침을 클릭합니다. '완료했습니다.' 패널이 나타납니다.

6.12.2.2.1 유니버스의 저장 프로시저 매개 변수

성능을 향상시키려면 빠른 디자인 마법사 또는 삽입 > 저장 프로시저 > 업데이트를 통해 여러 개의 프로시저가 동일한 데이터 소스에 대해 동일한 유니버스를 기반으로 하도록 지정합니다.

유니버스 매개 변수에서 매개 변수 `STORED_PROC_UNIVERSE` 가 `YES` 로 설정되어 있는지 확인하십시오. 이것은 현재 유니버스가 저장 프로시저를 기반으로 함을 나타냅니다.

저장 프로시저 열의 구문 분석 오류를 피하려면 복합 SQL 을 기반으로(예: 집계 함수 sum, count 등 사용) 결과 열에 별칭을 지정할 것을 권장합니다. 별칭이 지정된 개체는 제한 없이 만들 수 있습니다.

> **제한**
>
> 저장 프로시저는 `OUT` 또는 동적 결과 집합 매개 변수를 지원하지 않습니다.

6.12.2.3 입력 매개 변수를 사용하여 저장 프로시저를 기반으로 하는 유니버스 만들기

프로시저 진행 중에 데이터베이스의 목록에서 값을 선택하라는 메시지가 표시될 경우 개체가 이미 선언되어 있는 것입니다.

하나 이상의 입력 매개 변수를 필요로 하는 저장 프로시저를 기반으로 유니버스를 만들 수 있습니다. 입력된 값에 따라, 프로시저는 해당 팩트 테이블 값의 팩트 데이터를 반환합니다.

1. 빠른 디자인 마법사 도구 모음 단추를 클릭합니다.
   시작 창이 나타납니다.
2. 창 맨 아래의 저장 프로시저 유니버스를 선택하려면 여기를 클릭하십시오. 확인란을 클릭합니다.
3. 시작을 클릭합니다.
   유니버스 매개 변수 정의 패널이 나타납니다.
4. 유니버스 이름 입력 필드에 유니버스 이름을 입력합니다.
5. 데이터베이스 연결 선택 목록의 드롭다운 목록에서 데이터베이스 연결을 선택합니다.
6. 다음을 클릭합니다.
   초기 클래스 및 개체 만들기 패널이 나타납니다.
7. 저장 프로시저를 클릭합니다.
8. 추가를 클릭합니다.
   저장 프로시저에 입력 매개 변수가 필요하면 저장 프로시저 편집기가 나타납니다.
9. 매개 변수 목록에서 매개 변수를 클릭합니다.
10. 값 필드에 매개 변수 값을 입력하거나 프롬프트를 입력합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

11. 이 값 사용 또는 값에 대한 프롬프트를 표시합니다.를 선택합니다.
    값을 입력하면 프로시저는 실행 시 열 및 결과 집합 구조를 가져오며 이 값은 프로시저로 전달됩니다.
12. 값에 대한 프롬프트를 표시합니다.를 선택한 경우 프롬프트를 입력합니다.
    프롬프트 메시지를 입력하거나 기존 개체를 찾아 선택할 수 있습니다(예: 테이블의 구독자 ID 목록).
13. 확인을 클릭합니다.
14. 다음을 클릭합니다.
15. 마침을 클릭합니다. '완료했습니다.' 패널이 나타납니다.

6.12.2.3.1 값 프롬프트

프롬프트를 사용하여 저장 프로시저 실행 시 매개 변수 값을 정의할 수 있습니다.

기본적으로 저장 프로시저 매개 변수 이름은 저장 프로시저 구조에서 가져오며 저장 프로시저 이름을 묻는 메시지가 표시됩니다.

의미를 조정하고 값 목록을 이 프롬프트에 연결하여 사용자가 목록에서 다른 값을 추가하도록 할 수 있습니다.

저장 프로시저 매개 변수 대화 상자의 각 매개 변수 앞에 고급 대화 상자를 여는 단추가 있습니다.

6.12.2.4 저장 프로시저에 대해 프롬프트에서 값 목록 사용

구문

동적 매개 변수로 저장 프로시저를 정의할 때 값 목록(LOV)을 프롬프트 정의에 연결하여 표준 테이블을 기반으로 유니버스 개체를 선택하도록 할 수 있습니다(값 목록을 표준 테이블에서 가져와야 함). 이 방법은 사용자에게 유니버스 값 목록을 제안하는 데 유용합니다.

값 목록에는 단일 값만 포함될 수 있습니다. 사용자 지정 값 목록을 편집하거나 만들 수는 없습니다.

노트

저장 프로시저 정의에 속하지 않는 클래스 또는 개체를 삽입할 때 이러한 클래스나 개체는 숨겨져 있습니다. 이러한 클래스나 개체를 표시하도록 상태를 변경할 수 없습니다.

6.12.2.5 다중 결과 집합이 있는 저장 프로시저

예: 둘 이상의 결과 집합을 반환하는 저장 프로시저. 디자인 타임에 동일한 저장 프로시저를 기반으로 유니버스 구조에 몇 개의 테이블이 만들어집니다.

```
CREATE PROCEDURE qaputel.sp_getcustomer_2results
@location varchar(10)
AS
SELECT customer_key as KEYID, CUST_LNAME as Lname
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

```
FROM CUSTOMER
WHERE ADDRESS_LINE1 like @location
SELECT PREFIX as PREFIX, GENDER as GENDER, BIRTH_DT as BirthDATE
FROM CUSTOMER
```

다중 RS(결과 집합)는 다음 개념에 따라 처리됩니다.

Stored procedure RS1: a, b + RS2: b, d, e

Table A1: A, B

Table A2: B, D, E

이 샘플 구문은 동일한 ID 를 기반으로 하는 두 개의 테이블을 생성합니다. 유니버스 디자인 도구 모듈에서 Table A1 을 편집할 때 Table A2 도 편집할 수 있습니다.

열은 저장 프로시저 결과 집합 구조에 따라 배포됩니다. 동일한 저장 프로시저를 기반으로 두 테이블이 생성됩니다. 이 예에서 두 흐름에 대해 결과 집합 구조는 동일합니다. 유니버스 디자인 도구에서는 저장 프로시저의 두 번째 결과 집합에서 가져온 테이블 이름을 바꿉니다. 유니버스 디자인 도구는 비즈니스 요소를 세부적으로 조정할 수 있습니다.

유니버스 디자인 도구에서는 유니버스에 각 결과 집합에 대해 하나의 테이블을 만들고, 각 테이블에 대해 상호 독립적인 몇 개의 해당 개체를 만듭니다. 일반 저장 프로시저인 것처럼 유니버스를 수정할 수 있습니다.

6.13 유니버스 테스트

무결성 검사(도구 > 무결성 검사) 기능을 사용하여 검사를 정기적으로 실행하거나 Web Intelligence 에서 개체를 테스트하는 방법으로 유니버스에 있는 개체와 클래스의 무결성을 테스트할 수 있습니다. 쿼리 패널에서 유니버스 개체를 사용하여 쿼리를 만들고 SQL 보기 단추를 클릭하면 쿼리에서 해당 개체가 생성하는 SQL 을 볼 수도 있습니다.

6.13.1 쿼리 패널에서 개체 테스트

다음과 같이 쿼리 패널을 사용하여 쿼리의 SQL 을 볼 수 있습니다.

1. 도구 > 쿼리 패널을 선택합니다.
   쿼리 패널이 나타납니다.
2. 개체를 오른쪽의 결과 창에 끌어 놓습니다.
3. SQL 단추를 클릭합니다.

4. 쿼리의 SQL 이 나타납니다.
5. 쿼리 패널을 닫으려면 확인을 클릭한 다음 취소를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6.13.2 유니버스의 무결성 테스트

클래스와 개체를 만들고 수정할 때는 무결성 검사를 정기적으로 사용하여 유니버스의 무결성을 테스트해야 합니다. 무결성 검사 사용에 대한 내용은 자동으로 유니버스 무결성 검사 [페이지 172]를 참조하십시오.

6.13.3 Web Intelligence 로 유니버스 테스트

Web Intelligence 에서 테스트 쿼리를 실행하여 개체를 테스트할 수 있습니다. 개체를 테스트할 때는 다음과 같은 질문을 고려하는 것이 좋습니다.

- 개체가 있습니까? 그렇지 않다면 마지막 개체를 만든 후 유니버스를 저장했습니까?
- SQL 이 올바릅니까?
- 쿼리 결과가 올바릅니까?

반환 결과가 올바른지 확인하고 무결성 검사를 통해 스키마 구성 요소를 검사하여 조인도 테스트해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7 유니버스 최적화

유니버스를 최적화하면 대체로 쿼리 시간이 단축됩니다. 유니버스를 최적화할 수 있는 방법에는 여러 가지가 있습니다.

- 유니버스 매개 변수에서 배열 반입 매개 변수 최적화
- 각 테이블에 가중치 할당
- 바로 가기 조인 사용
- 데이터베이스에서 집계 테이블 만들기 및 사용

이러한 각 방법은 다음 단원에서 설명합니다.

7.1 개요

다음 기술을 사용하여 유니버스를 최적화할 수 있습니다.

- 집계 테이블 사용 [페이지 327]
- 개체의 SQL 에서 @함수 사용 [페이지 339]
- 외부 전략을 사용하여 유니버스 만들기 사용자 지정 [페이지 365]
- 분석 함수 사용 [페이지 377]

7.2 집계 테이블 사용

유니버스 디자인 도구의 기능을 사용하여 기본 테이블 대신 데이터베이스의 집계 테이블을 기준으로 쿼리를 실행하도록 개체에 대한 Select 문을 정의할 수 있습니다. 쿼리가 최적화되는 경우 집계 테이블을 기준으로 쿼리가 실행되고 그렇지 않은 경우 기본 테이블을 기준으로 쿼리가 실행되도록 조건을 설정할 수 있습니다. 집계 테이블을 사용하여 쿼리를 최적화하는 개체의 이와 같은 기능을 집계 인식이라고 합니다.

이 장에서는 유니버스에서 집계 인식을 설정하는 방법에 대해 설명합니다.

7.2.1 집계 인식 개요

집계 인식은 데이터베이스의 집계 테이블을 사용하는 유니버스의 기능을 가리키는 용어입니다. 이러한 테이블에는 미리 계산된 데이터가 포함되어 있습니다. 개체에 대한 Select 문에서 @Aggregate_Aware 함수를 사용할 수 있습니다. 이 함수는 집계되지 않은 데이터가 포함된 테이블 대신 집계 테이블을 기준으로 쿼리를 실행하도록 지정합니다.

집계 테이블을 사용하면 쿼리 실행 속도가 빨라지고 SQL 트랜잭션의 성능이 향상됩니다.

유니버스에서 집계 인식의 신뢰성과 유용성은 집계 테이블의 정확도에 따라 달라집니다. 집계 테이블은 모든 팩트 테이블과 동시에 새로 고쳐야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

집계 테이블을 기반으로 한 대체 정의를 갖는 개체가 하나 이상 포함되어 있는 유니버스에서는 "집계가 인식"됩니다. 이러한 정의는 집계의 수준에 상응합니다. 예를 들어, 이윤이라는 개체는 월별, 분기별 또는 연도별로 집계할 수 있습니다. 이러한 개체를 집계 개체라고 합니다.

집계 개체를 사용하는 유니버스를 통해 작성된 쿼리는 적절한 수준으로 집계된 정보를 최적의 속도로 반환합니다.

7.2.2 데이터 웨어하우스에 집계 인식 적용

집계 인식은 데이터 웨어하우스를 작업할 때 특히 유용합니다. 예를 들어 시간, 지리 및 제품이라는 세 가지 차원으로 구성된 데이터 웨어하우스를 생각해 볼 수 있습니다.

이 데이터 웨어하우스의 최하위 수준에는 고객 및 제품에 대한 정보를 날짜별로 저장할 수 있습니다. 고객의 제품 구매에 대해 날짜별로 행을 하나씩 할당합니다. 그 결과는 다음과 같이 표현할 수 있습니다.

365 일 x 100 개 시 x 10 개 제품 = 365,000 개 행

연간 판매 정보를 요청하면 데이터베이스 엔진에서 많은 행을 더해야 합니다. 그러나 회사의 연간 판매는 실제로 다음과 같은 몇 개의 행으로 구성될 수 있습니다.

3 년 x 3 개 국가 x 3 개 회사 = 27 개 행

따라서, 이 예제의 경우 테이블의 행 27 개로도 충분히 사용자의 요청에 답할 수 있습니다. 이 정보를 기반으로 이러한 행을 집계 테이블에 미리 요약해 두면 훨씬 효율적일 수 있습니다.

7.2.3 집계 인식 설정

유니버스에서 집계 인식을 설정하는 과정은 크게 네 가지 단계로 이루어집니다. 주요 단계는 아래에 요약되어 있습니다.

- 개체 작성

1. 개체의 모든 가능한 정의(테이블/열 조합)를 확인합니다.
2. 집계 수준별로 개체를 정렬합니다.
3. @Aggregate_Awareness 함수를 사용하여 개체를 작성합니다.

- 호환되지 않는 개체 지정

1. 개체/집계 테이블 매트릭스를 작성합니다.
2. 첫 번째 집계 테이블의 경우 각 개체가 집계 수준 이상일 때와 이하일 때 호환성의 여부를 결정합니다.
3. 해당 테이블에 대해 호환되지 않는 개체의 확인란만 선택합니다.
4. 나머지 집계 테이블의 경우 1-3 단계를 반복합니다.

- 모든 필요한 컨텍스트 정의

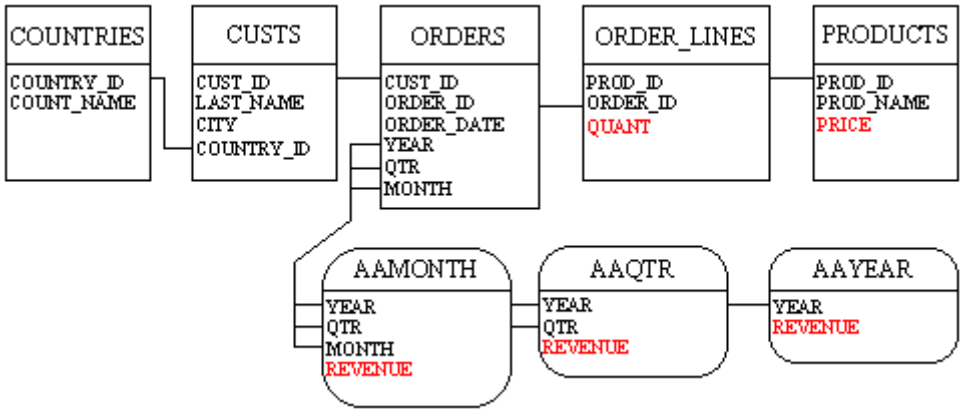
집계의 수준별로 컨텍스트 하나를 정의합니다.

- 결과 테스트

1. 여러 번 쿼리를 실행합니다.
2. 결과를 비교합니다.

위 과정의 각 단계에 대한 자세한 설명은 다음 단원에 나와 있습니다. 아래의 예제 스키마는 각 단계를 보여주는 데 사용됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



이 스키마에는 AAMONTH, AAQTR 및 AAYEAR 라는 세 가지 집계 테이블이 미리 정의되어 있습니다.

> **노트**
>
> 이 예제 스키마는 일반적인 스키마와 다소 차이가 있습니다. 이 스키마는 집계 인식을 설정하기 위한 단계를 설명하는 예제로만 사용 가능합니다. 실제 프로덕션 스키마의 경우 집계 테이블에는 일반적으로 시간을 기반으로 한 단일 차원 대신 여러 차원이 결합됩니다. 시간 차원(연도, 분기 및 월) 또한 일반적으로 집계 테이블이 아니라 마스터 테이블 내에서 정의됩니다.

7.2.4 개체 작성

유니버스에 집계 인식을 설정하는 첫 번째 단계는 집계 인식이 이루어질 개체를 결정하는 것입니다. 계수 개체 또는 차원 개체를 사용할 수 있습니다.

판매 수익이라는 개체에는 위 스키마를 기반으로 한 다음과 같은 정의가 있습니다.

PRODUCTS.PRICE*ORDER_LINES.QUANT

집계되지 않은 테이블을 사용하여 집계를 수행하는 대신 가능한 경우 집계 테이블을 사용하도록 Sales_Revenue 를 다시 정의할 수 있습니다.

집계를 인식하도록 판매 수익을 다시 정의하는 데 필요한 각 단계는 정의에서 집계 테이블을 사용하도록 하려는 다른 개체에 대해서도 완료해야 합니다.

7.2.5 집계 개체의 모든 조합 식별

다양한 테이블에서 개체의 가능한 조합을 모두 확인해야 합니다. 판매 수익 개체는 다음과 같은 방식으로 정의할 수 있습니다.

- AAMONTH.REVENUE
- AAYEAR.REVENUE

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- AAQTR.REVENUE
- PRODUCTS.PRICE*ORDER_LINES.QUANT

7.2.6 집계 순서에 따라 개체 정렬

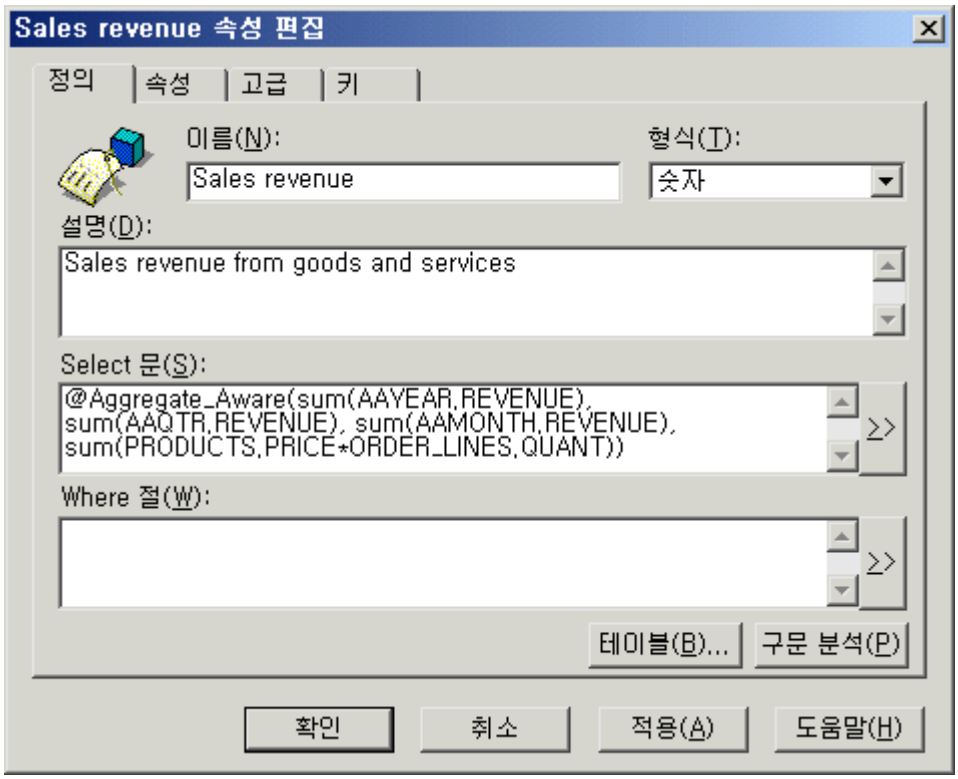
개체의 모든 조합을 확인한 후에는 다음과 같이 집계 수준에 따라 정렬합니다.

- AAYEAR.REVENUE 는 최상위 수준의 집계입니다.
- AAQTR.REVENUE 는 그 다음 수준입니다.
- AAMONTH.REVENUE 는 그 다음 수준입니다.
- PRODUCTS.PRICE*ODER_LINES.QUANT 는 최하위 수준의 집계입니다.

7.2.7 @Aggregate_Aware 함수를 사용하여 집계 개체 정의

모든 집계 인식 개체에 대해 @Aggregate_Aware 함수를 사용하여 Select 문을 다시 정의합니다. @Aggregate_Aware 함수를 사용하면 개체에서는 매개 변수로 나열된 집계 테이블을 가장 먼저 쿼리합니다. 집계 테이블이 적절하지 않은 경우 집계되지 않은 테이블을 기반으로 한 원래 집계를 대상으로 쿼리가 실행됩니다. @Functions 에 대한 자세한 내용은 개체의 SQL 에서 @함수 사용 [페이지 339] 단원을 참조하십시오.

아래에는 @Aggregate_Aware 함수를 사용하는 판매 수익에 대한 Select 문이 나와 있습니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

@Aggregate_Aware 함수의 구문은 다음과 같습니다.

표 148:

@Aggregate_Aware(sum(agg_table_1), ... sum(agg_table_n))

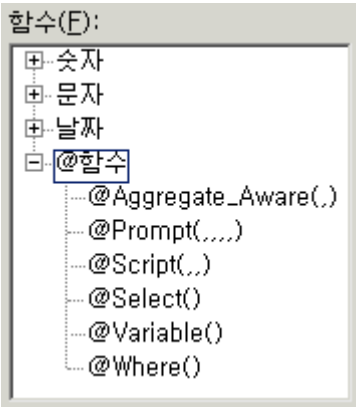
여기에서 agg_table_1 은 최상위 수준의 집계이고 agg_table_n 은 최하위 수준의 집계입니다.

모든 집계 테이블의 이름을 인수로 입력해야 합니다. 왼쪽에서 오른쪽으로 집계의 내림차순에 따라 테이블의 이름을 배치합니다.

7.2.7.1 @Aggregate_Aware 를 사용하여 개체 정의

@Aggregate_Aware 를 사용하여 개체를 다시 정의하려면

1. 개체를 두 번 클릭합니다.
   개체에 대한 속성 편집 대화 상자가 나타납니다.
2. 선택 상자 옆에 있는 >> 단추를 클릭합니다.
   Select 문 편집 대화 상자가 열립니다.
3. SELECT 문의 시작 부분을 클릭합니다.
   또는
   개체에 아직 SELECT 문이 없는 경우 선택 상자의 아무 곳이나 클릭합니다.
   상자의 왼쪽 위에 커서가 나타납니다.
4. 함수 창에서 @Functions 노드를 클릭합니다.
   사용 가능한 함수의 목록이 나타납니다.



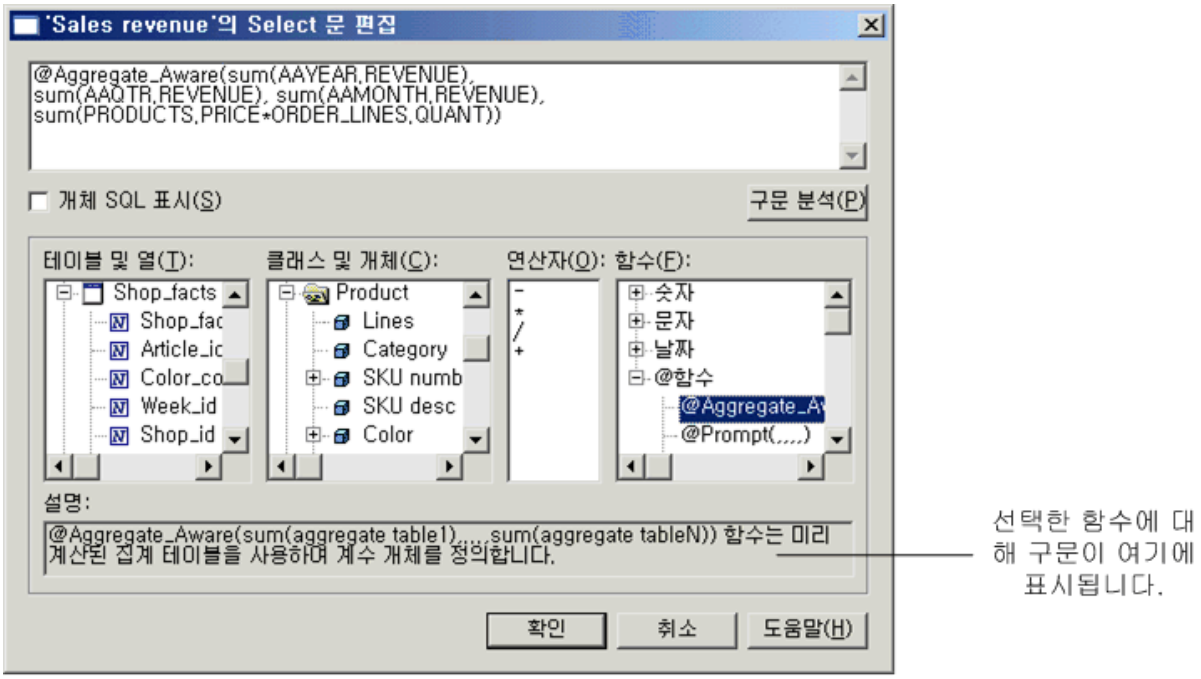
5. @Aggregate_Aware 를 두 번 클릭합니다.
   @Aggregate_Aware 의 구문이 Select 문에 삽입됩니다. 대화 상자의 아래쪽에 있는 설명 상자에 이 구문에 대한 설명이 표시됩니다. 이 설명은 @function 에 대한 매개 변수를 입력하는 데 도움이 됩니다.
6. @AggregateAware 함수의 괄호 안에 집계 데이터의 최상위 수준에서 최하위 수준으로 순서에 따라 집계를 삽입합니다.
7. 각 집계는 쉼표로 구분합니다. 예를 들어, 판매 수익에 대한 구문은 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 149:

@Aggregate_Aware(sum (AAYEAR.REVENUE), sum(AAQTR.REVENUE), sum (AAMONTH.REVENUE), sum(PRODUCTS.PRICE*ORDER_LINES.QUANT))

8. 구문 분석을 클릭하여 구문을 검사합니다.
   아래에는 판매 수익에 대한 *SQL 편집기*의 *Select* 문이 나와 있습니다.



9. 각 대화 상자에서 확인을 클릭합니다.
   이 예제의 경우 @Aggregate_Aware 함수를 사용하여 연도 및 분기 차원 개체도 다시 정의합니다.

7.2.8 호환되지 않는 개체 지정

이제 유니버스의 각 집계 테이블에 대해 호환되지 않는 개체를 지정해야 합니다. 사용자가 지정한 호환되지 않는 개체 집합에 따라 SQL 을 생성하는 동안 무시할 집계 테이블이 결정됩니다.

집계 테이블을 기준으로 볼 때 개체는 호환되거나 호환되지 않습니다. 호환성 규칙은 다음과 같습니다.

표 150:

• 개체가 테이블과 같거나 더 높은 수준의 집계이면 테이블과 호환됩니다. • 개체가 테이블보다 더 낮은 수준의 집계이거나 테이블과 전혀 관련이 없으면 테이블과 호환되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.2.8.1 매트릭스를 사용하여 개체 분석

개체와 집계 테이블의 호환성을 분석하기 위해 매트릭스를 작성하는 것이 유용할 수도 있습니다. 이 매트릭스의 처음 두 열에 클래스와 개체의 이름을 나열할 수 있습니다. 그런 다음 유니버스의 각 집계 테이블에 대한 열 머리글을 만들 수 있습니다. 예제의 스키마를 기반으로 한 빈 매트릭스는 다음과 같은 형식입니다.

표 151:

클래스	개체	AAYEAR	AAQTR	AAMONTH
고객	고객 코드 (CUSTOMER.CUST_ID)			
	고객 이름 (CUSTOMER.LAST_NAME)			
	고객 도시 (CUSTOMER.CITY)			
	고객 국적 (COUNTRIES.COUNT_NAME)			
제품	제품 코드 (PRODUCT.PROD_ID)			
	제품 이름 (PRODUCT.PROD_NAME)			
주문	주문 연도 (AAYEAR.PROD_NAME)			
	주문 분기 (AAQTR.QTR)			
	주문 월 (AAMONTH.MONTH)			
	주문 날짜 (ORDERS.ORDER_DATE)			
판매 계수	판매 수익 (@Aggregate_Aware(...))			

각 테이블에 대해 개체가 호환되지 않는 경우 X 를 입력합니다.

다음은 예제를 기반으로 한 완전한 매트릭스입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 152:

클래스	개체	AAYEAR	AAQTR	AAMONTH
고객	고객 코드 (CUSTOMER.CUST_ID)	X (n)	X (n)	X (n)
	고객 이름 (CUSTOMER.LAST_NAME)	X (n)	X (n)	X (n)
	고객 도시 (CUSTOMER.CITY)	X (n)	X (n)	X (n)
	고객 국적 (COUNTRIES.COUNT_NAME)	X (n)	X (n)	X (n)
제품	제품 코드 (PRODUCT.PROD_ID)	X (n)	X (n)	X (n)
	제품 이름 (PRODUCT.PROD_NAME)	X (n)	X (n)	X (n)
주문	주문 연도 (AAYEAR.PROD_NAME)	- (s)	- (h)	- (h)
	주문 분기 (AAQTR.QTR)	X (l)	- (s)	- (h)
	주문 월 (AAMONTH.MONTH)	X (l)	3 (l)	- (s)
	주문 날짜 (ORDERS.ORDER_DATE)	X (l)	X (l)	X (l)
판매 계수	판매 수익 (@Aggregate_Aware(...))	-	-	-

X (n): 이 개체는 집계 테이블과 아무런 관련이 없습니다. 따라서 이 개체는 호환됩니다.

X (l): 이 개체는 이 집계 테이블보다 낮은 수준의 집계이므로 정보를 얻는 데 사용할 수 없습니다. 따라서 이 개체는 호환되지 않습니다.

- (s): 이 개체는 이 집계 테이블과 같은 수준의 집계이므로 정보를 얻는 데 사용할 수 있습니다. 따라서 이 개체는 호환됩니다.

- (h): 이 개체는 이 집계 테이블보다 높은 수준의 집계이므로 정보를 얻는 데 사용할 수 있습니다. 따라서 이 개체는 호환됩니다.

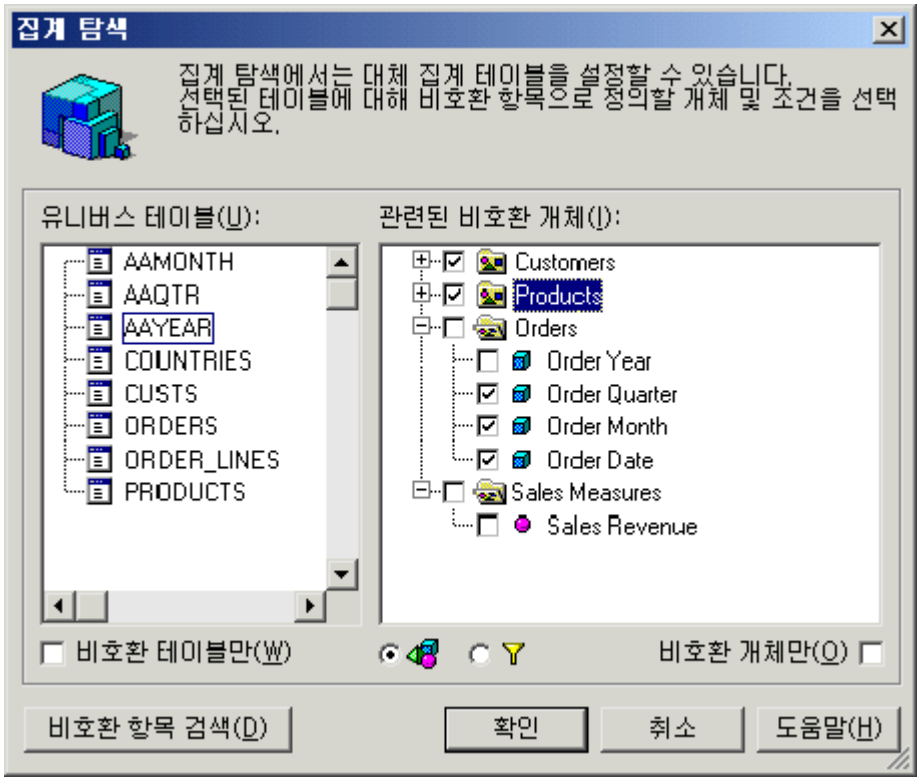
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.2.9 호환되지 않는 개체 지정

이제 호환되지 않는 개체를 지정합니다. 집계 탐색 대화 상자(▶ 도구 > 집계 탐색 ◀)를 사용하여 호환되지 않는 개체를 지정합니다.

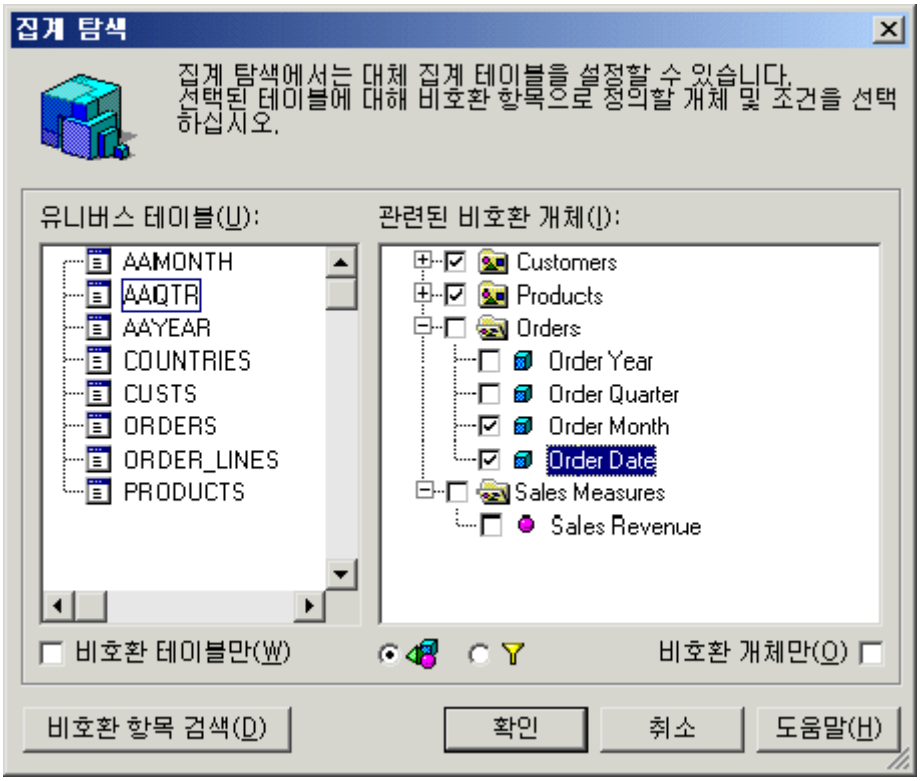
다음과 같이 집계 탐색 대화 상자를 사용하여 호환되지 않는 개체를 지정합니다.

1. ▶ 도구 > 집계 탐색 ◀을 선택합니다.
   집계 탐색 대화 상자가 나타납니다. 이 대화 상자에는 다음과 같은 두 개의 창이 있습니다.
   - 유니버스 테이블: 유니버스의 모든 테이블이 나열됩니다.
   - 호환되지 않는 연결된 개체: 유니버스의 모든 개체가 나열됩니다.
2. 왼쪽 창에서 집계 테이블을 클릭합니다.
3. 오른쪽 창에서 호환되지 않는 각 개체의 확인란을 선택합니다.
   예를 들어 매트릭스를 기준으로 할 때 고객 클래스의 모든 개체는 AAYEAR 테이블에 대해 호환되지 않습니다. 다음과 같이 클래스 이름 옆에 있는 확인란을 선택합니다.

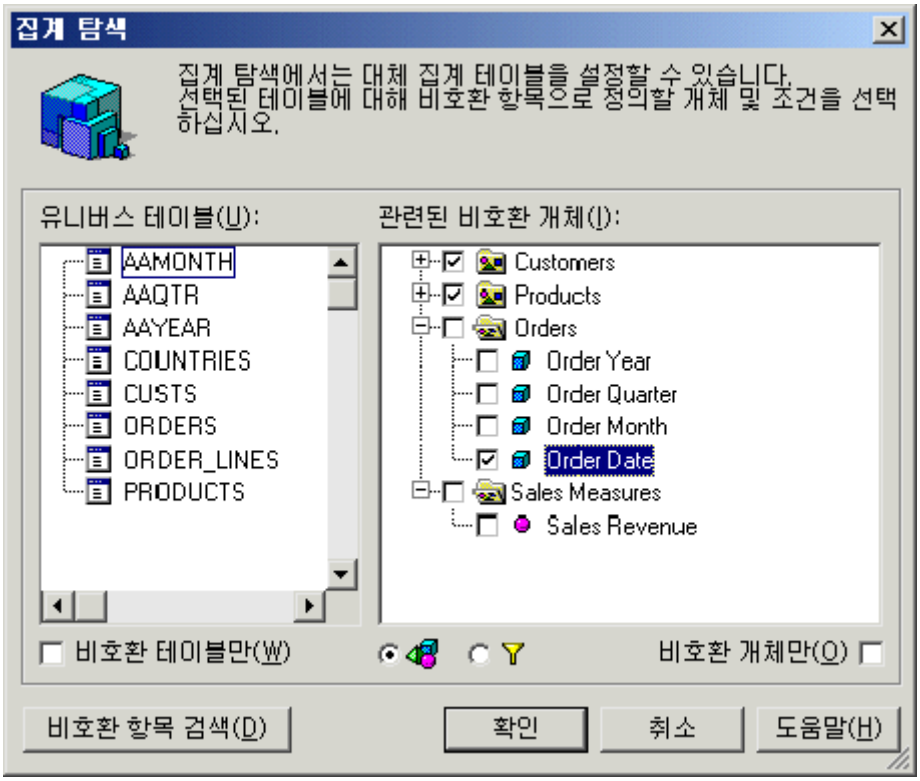


4. 유니버스의 각 집계 테이블에 대해 위 단계를 반복합니다.
   예를 들어 아래에는 AAQTR 테이블에 대해 호환되지 않는 개체가 나와 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



AAMONTH 테이블의 경우 한 개체만 호환되지 않습니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

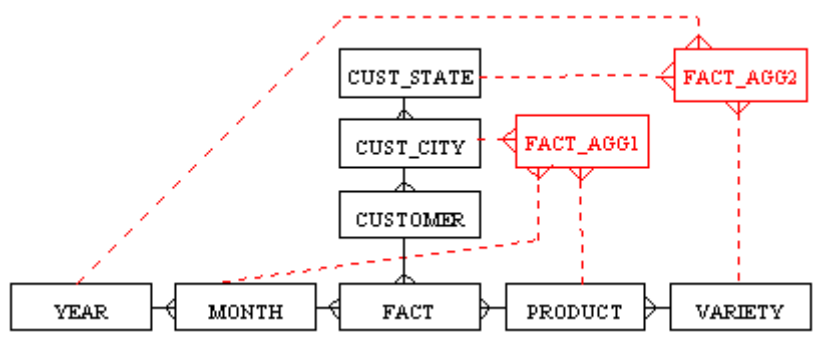
5. 각 테이블에 대해 호환되지 않는 개체를 모두 지정한 다음 확인을 클릭합니다.

> **노트**
>
> 이 대화 상자에 있는 비호환 항목 검색 단추를 사용하면 호환되지 않는 개체를 쉽게 지정할 수 있습니다. 테이블을 클릭한 다음 이 단추를 클릭하면 유니버스 디자인 도구에서는 호환되지 않는 것으로 간주되는 개체를 자동으로 선택합니다. 비호환 항목 검색 기능을 통해 선택된 호환되지 않는 개체는 최종 선택이 아니라 제안된 항목으로 간주해야 합니다.

7.2.10 집계 테이블에 관련된 루프 해결

데이터베이스에 하나 이상의 집계 테이블이 포함되어 있는 경우 컨텍스트를 사용하여 루프를 해결해야 합니다.

> **예**
>
> **집계 테이블에 관련된 루프 해결**
>
> 다음은 집계 테이블이 포함된 간단한 스키마입니다.



스키마에서 다음 사항에 유의하십시오.

- FACT_AGG1 은 FACT 테이블과 거의 동일한 집계 테이블입니다. 여기에는 고객의 도시, 제품 및 월에 대해 집계된 계수 이외에도 (고객) 도시 키, 제품 키 및 월 키가 포함되어 있습니다.
- FACT_AGG2 도 FACT 테이블과 유사한 집계 테이블입니다. 이 테이블의 계수는 고객의 주, 제품 및 연도에 대해 집계되어 있습니다.
- 계수(키 성능 표시기)는 모든 팩트 테이블에 저장됩니다. 판매 수익은 FACT_AGG1, FACT_AGG2 및 FACT 에 저장되지만 각 테이블의 각 수준에 따라 집계됩니다.

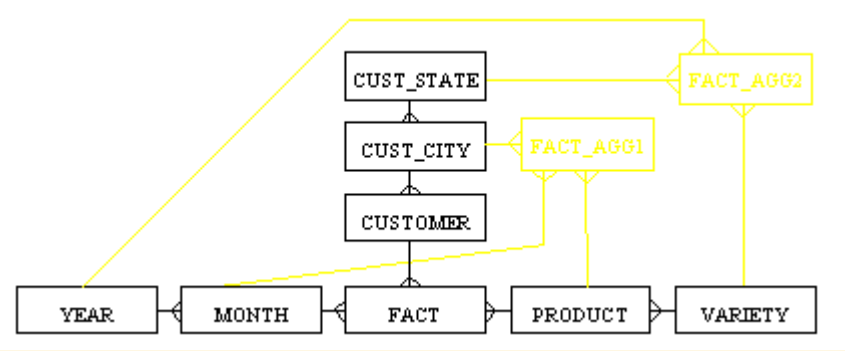
판매 수익 및 고객 도시를 사용한 쿼리의 경우 CUST_STATE 와 CUST_CITY 사이의 조인이 아니라 CUST_STATE 와 FACT_AGG2 사이의 조인을 사용해야 합니다.

그러나 이 쿼리를 실행하려면 먼저 FACT, FACT_AGG1 및 FACT_AGG2 같은 세 가지 컨텍스트를 정의해야 합니다. 컨텍스트는 사용자에게 표시되지 않으므로 의미가 더욱 분명한 레이블을 사용하여 컨텍스트의 이름을 다시 지정할 필요가 없습니다.

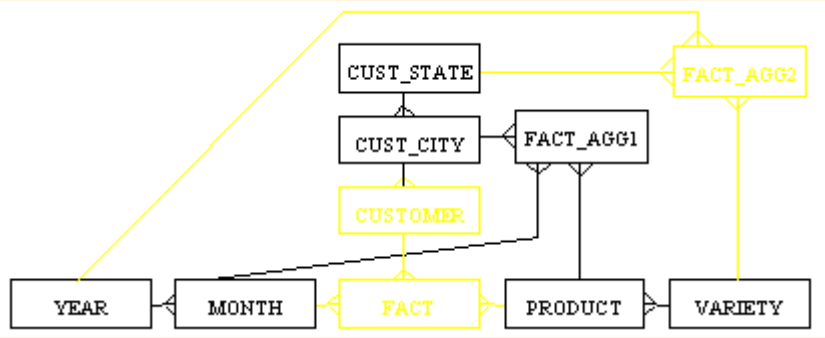
다음 페이지에는 이 세 가지 컨텍스트에 포함된 조인이 나와 있습니다. 각 스키마에서 지정된 컨텍스트는 더 짙은 색의 조인 집합으로 표시되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

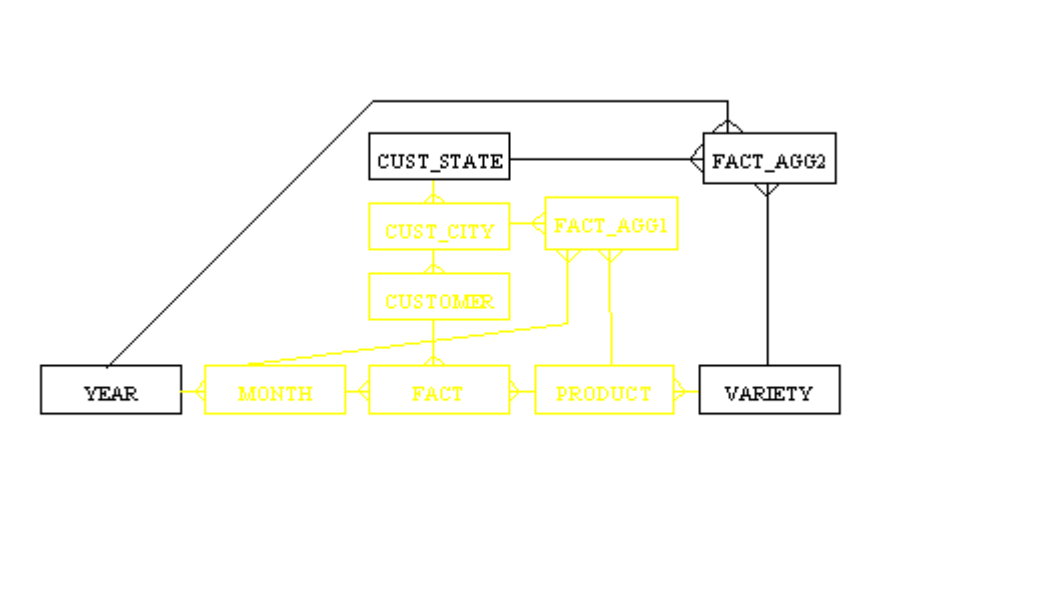
FACT 컨텍스트



FACT_AGG1 컨텍스트



FACT_AGG2 컨텍스트



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.2.11 집계 인식 테스트

집계 인식을 설정하는 마지막 단계는 Web Intelligence 에서 결과를 테스트하는 것입니다.

첫 번째 예제를 기준으로 다음과 같은 쿼리를 실행한 다음 서로 다른 결과를 비교할 수 있습니다.

- 판매 수익에 대한 주문 연도.
- 판매 수익에 대한 주문 분기.
- 판매 수익에 대한 주문 월.
- 판매 수익에 대한 고객.
- 판매 수익에 대한 제품.

7.3 개체의 SQL 에서 @함수 사용

`@`**함수**는 개체의 SQL 을 더 다양한 방식으로 지정하는 데 사용할 수 있는 특수 함수입니다. `@`**함수**는 개체의 *Select* 편집 상자에 있는 함수 창에서 사용할 수 있습니다.

개체의 Select 문이나 Where 절에 하나 이상의 `@`**함수**를 사용할 수 있습니다. 사용할 수 있는 `@`**함수**는 다음과 같습니다.

표 153:

@함수	설명	사용 위치
@Aggregate_Aware	집계 및 차원 데이터가 들어 있는 열을 개체에 포함시킵니다.	SELECT 문
@Prompt	SQL 에 프롬프트를 삽입합니다. 사용자가 쿼리를 실행하면 `@Prompt` 함수를 사용하는 개체가 쿼리에 포함되어 있을 때마다 사용자에게 제한 값을 입력하라는 메시지가 표시됩니다.	SELECT 문 WHERE 절
@Script	`@Script` 함수를 사용하는 개체를 쿼리에 포함시킬 때마다 스크립트를 실행합니다.	WHERE 절
@Select	다른 개체의 SELECT 문을 사용할 수 있게 합니다.	SELECT 문
@Variable	참조 텍스트 파일과 같이 메모리에 저장되어 있는 변수 값을 호출합니다.	WHERE 절

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

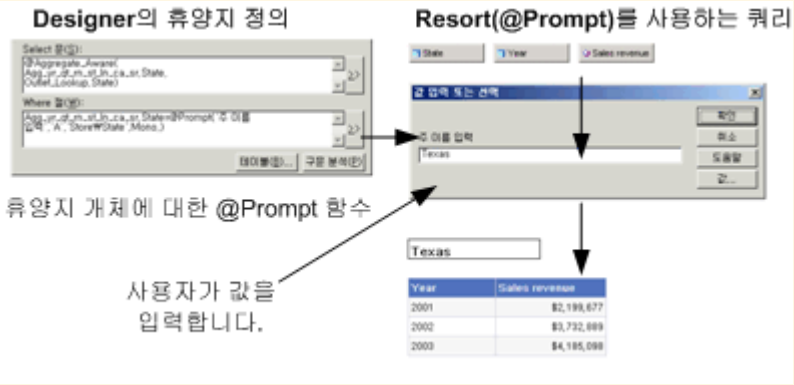
@함수	설명	사용 위치
@Where	다른 개체의 WHERE 절을 사용할 수 있게 합니다.	WHERE 절

예

`@Prompt` 함수를 사용하여 반환 값을 입력한 프롬프트 값으로 제한

`@Prompt` 함수는 유니버스 디자인 도구에서 사용 가능한 `@Functions` 중 하나입니다. `@Prompt` 함수를 사용하면 Web Intelligence 쿼리에서 개체를 사용할 때 메시지 상자가 표시됩니다.

이 메시지 상자에는 개체의 값을 입력하라는 내용이 표시됩니다. 그러면 쿼리는 사용자가 입력한 프롬프트 값에 대한 값을 다음과 같이 반환합니다.

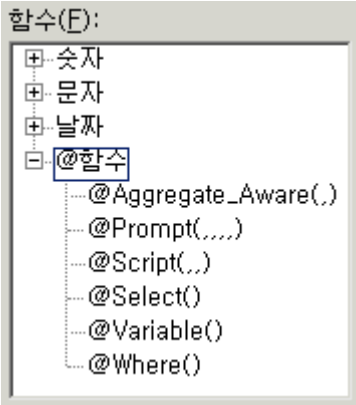


7.3.1 개체에 `@Function` 삽입

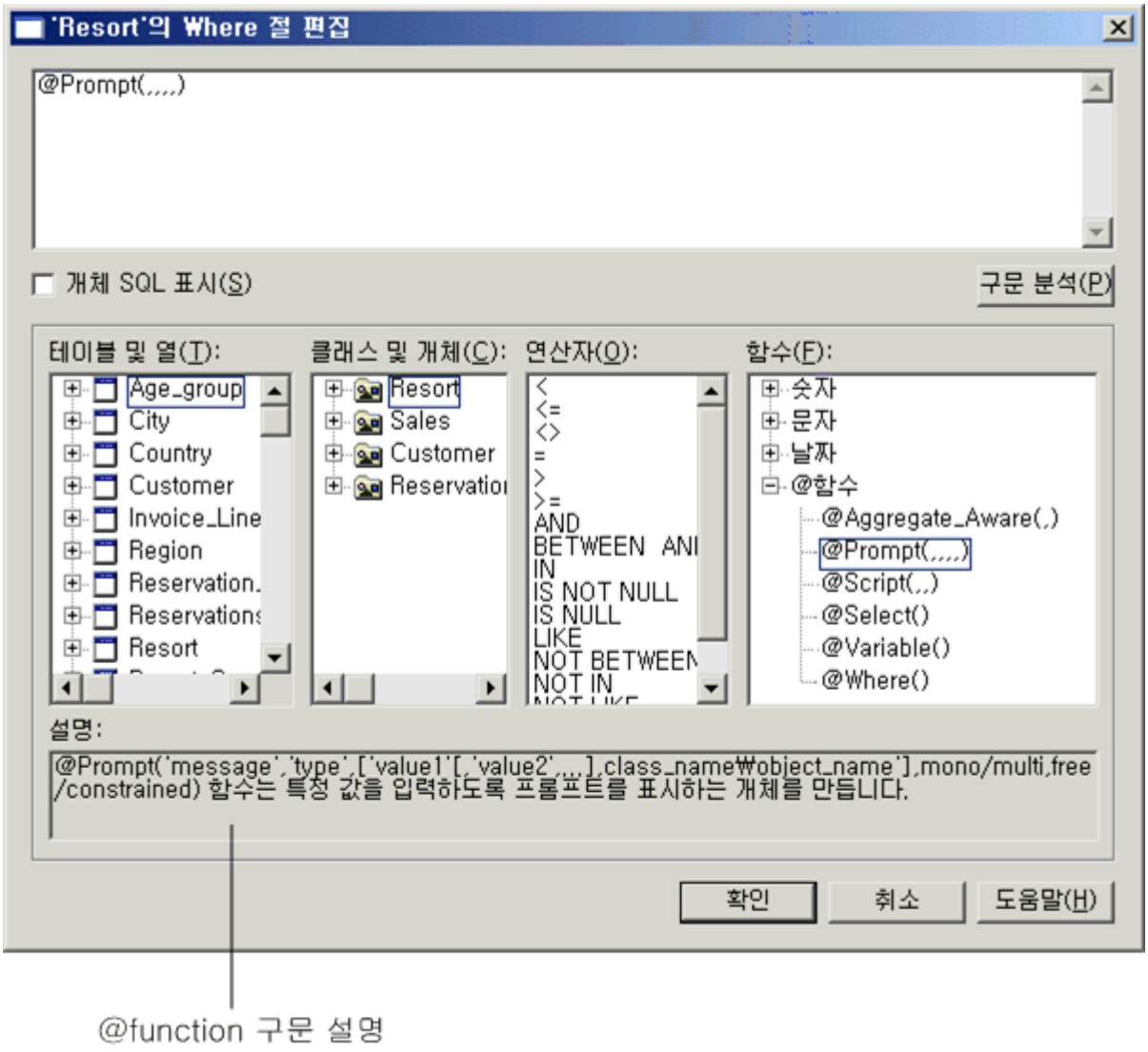
개체의 SQL 정의에 `@Function` 을 삽입하려면

1. 유니버스 창에서 개체를 두 번 클릭합니다.
   개체에 대한 속성 편집 대화 상자가 나타납니다.
2. 선택 상자 옆에 있는 >> 단추를 클릭합니다.
   또는
   Where 상자 옆에 있는 >> 단추를 클릭합니다.
   Select 문 편집 또는 *Where* 절 편집 대화 상자가 나타납니다.
3. `Select` 문이나 `Where` 절에서 `@Function` 을 추가할 위치를 클릭합니다. 위의 그림 같이 상자가 비어 있으면 상자 안을 클릭합니다. 그러면 커서가 상자의 왼쪽 맨 위에 자동으로 표시됩니다.
4. Click the *@Functions* node in the *Functions* pane.
   사용 가능한 `@Function` 의 목록이 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



5. `@Function` 을 두 번 클릭합니다.
   개체의 SELECT 문이나 WHERE 절에 `@Function` 의 구문이 추가됩니다. 대화 상자의 아래쪽에 있는 설명 상자에 이 구문에 대한 설명이 표시됩니다. 이 설명은 `@Function` 에 대한 매개 변수를 입력하는 데 도움이 됩니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

6. 대화 상자의 위쪽에 있는 창에 필요한 매개 변수를 입력합니다.
7. 구문 분석을 클릭하여 구문을 검사합니다.
8. 각 대화 상자에서 확인을 클릭합니다.

7.3.2 @Aggregate_Aware

`@Aggregate_Aware` 함수를 사용하면 데이터베이스 내에서 요약 데이터가 포함된 테이블을 개체에서 사용할 수 있습니다. 요약 테이블이 포함된 데이터베이스에서 집계 데이터를 반환하는 쿼리를 실행하는 경우 팩트나 이벤트 데이터가 들어 있는 열보다 요약 데이터가 들어 있는 열에 대해 SELECT 문을 실행하는 것이 더 빠릅니다. 집계 테이블과 호환되지 않는 것으로 선언된 개체는 집계 테이블을 사용할 수 없게 되지만, 쿼리 대신 기본 테이블을 사용할 수 있습니다.

`@Aggregate_Aware` 함수를 사용하면 유니버스에서 집계 인식을 설정할 수 있습니다. 이 설정 프로세스에는 `@Aggregate_Aware` 함수를 사용해야 하는 몇 가지 다른 단계도 포함됩니다.

- 각 집계 테이블에 대해 호환되지 않는 개체를 지정합니다.
- 집계 테이블에 대한 모든 루프를 확인합니다.
- 집계 테이블을 테스트하여 올바른 값을 반환하는지 확인합니다.

7.3.2.1 @Aggregate_Aware 함수 구문

`@Aggregate_Aware` 함수의 구문은 다음과 같습니다.

```
@Aggregate_Aware(sum(agg_table_1), ...
                         sum(agg_table_n))
```

모든 집계 테이블의 이름을 인수로 입력해야 합니다. 왼쪽에서 오른쪽으로 집계의 내림차순에 따라 테이블의 이름을 배치합니다.

표 154:

구문	설명
agg_table_1	최상위 수준의 집계입니다.
agg_table_n	최하위 수준의 집계입니다.

예제

```
@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)
          )
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

예제에서 이 함수를 사용한 개체가 테이블과 호환되지 않는 것으로 선언될 경우, 해당 테이블은 무시됩니다. 개체가 R_Country 테이블 및 R_Region 테이블과 호환되지 않을 경우, 다음의 SQL 이 사용됩니다.

```
@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

`@Prompt` 함수를 사용하여 쿼리에 프롬프트를 삽입합니다. 프롬프트를 사용하면 사용자가 보고서를 만들 때 데이터를 제한하거나 큰 값 개체를 손쉽게 사용할 수 있습니다. `@Prompt` 함수는 SELECT 문이나 개체의 WHERE 절에 사용합니다. 이렇게 하면 쿼리에 해당 개체가 사용된 경우 사용자가 하나 이상의 제한 값을 입력하거나 값 또는 값 목록을 선택해야 합니다. 사용자가 쿼리를 실행하면 값을 입력할 프롬프트 상자가 나타납니다.

`@Prompts` 는 유추된 SQL 에 강제로 제한을 적용해야 하지만 조건 값을 미리 설정하지 않으려는 경우에 유용합니다.

선택적으로 프롬프트에 대한 기본값을 정의할 수 있습니다. 기본값을 포함하는 프롬프트는 기본값이 있는 Web Intelligence 프롬프트와 같은 방식으로 동작합니다.

다음과 같은 방법으로 @Prompt 정의를 편집할 수 있습니다.

- @Prompt 편집기를 사용합니다.
- 조건에 대한 속성 편집 대화 상자의 정의 창에서 *SELECT* 또는 *WHERE* 필드에 정의를 입력합니다.
- 속성 편집 대화 상자의 고급 편집 창에 정의를 입력합니다.

> **노트**
>
> 예를 들어, 다른 기본 키를 사용하는 것만 제외하고는 매우 유사한 두 개의 프롬프트를 사용할 때는 같은 질문(프롬프트 텍스트)을 사용하지 마십시오. 시스템에서 두 개의 프롬프트를 구분할 수 없어 부적절한 응답을 제공할 수 있기 때문입니다.

> **노트**
>
> 특히 값 목록을 입력하는 경우 `@Prompt` 정의가 복잡해질 수 있으므로 `@Prompt` 편집기를 사용하는 것이 좋습니다.

> **노트**
>
> `@Prompt` 함수가 단일 값인 경우 같은 쿼리에서 `@Variable` 함수와 `@Prompt` 함수를 병합할 수 있습니다.

관련 정보

@Prompt 편집기 [페이지 344]
기존 @Prompt 식 편집 [페이지 349]
@Prompt 함수 구문 [페이지 355]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]

7.3.3.1 @Prompt 편집기

@Prompt 편집기를 사용하여 Web Intelligence 또는 Desktop Intelligence 사용자가 관계형 또는 OLAP 유니버스에 대해 쿼리를 실행할 때 나타나는 프롬프트를 정의하거나 편집할 수 있습니다. 편집기를 사용하면 프롬프트 정의 또는 편집 프로세스가 간편해집니다. 프롬프트 정의는 창 아래쪽에 표시되며 해당 프롬프트에 각기 다른 값을 정의하면 자동으로 업데이트됩니다. @Prompt 문자열의 구문이 올바른 경우에만 기존 @Prompt 식을 마우스 오른쪽 단추로 클릭했을 때 *@Prompt* 편집 메뉴 항목이 활성화됩니다.

속성 편집 대화 상자의 *SELECT* 또는 *WHERE* 창에 `@Prompt` 정의를 직접 입력할 수도 있습니다.

관련 정보

@Prompt 함수 구문 [페이지 355]
SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]
프롬프트의 정적 값 목록 정의 [페이지 347]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
프롬프트에 대한 사용자 지정 값 목록 선택 [페이지 348]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
기존 @Prompt 식 편집 [페이지 349]

7.3.3.2 @Prompt 편집기의 @Prompt 식 속성

프롬프트의 다음 속성을 편집할 수 있습니다.

표 155:

속성	설명
메시지	사용자에게 표시되는 프롬프트 메시지입니다. 예를 들어 '국가 선택'을 입력할 수 있습니다. 프롬프트 텍스트(질문). 기본값 = 값 입력
값 형식	사용자가 입력하거나 선택하는 데이터 형식입니다. 이를 통해 사용자가 올바른 형식의 데이터를 입력하거나 선택할 수 있습니다. 다음 중에서 선택합니다. • Alphanumeric(A) • Numeric(N) • Date(D) 기본값 = Alphanumeric

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

속성	설명
키 형식	Primary_key 가 선택 모드로 선택된 경우 사용자가 입력 또는 선택하는 키 유형을 설정합니다. 다음 중에서 선택합니다. • Alphanumeric(A) • Numeric(N) • Date(D) 기본값 = 없음
여러 선택 허용	이 옵션이 선택되면 사용자가 하나 이상의 값을 입력하거나 선택할 수 있습니다. 기본값 = 선택하지 않음 - 사용자가 하나의 값만 선택하거나 입력할 수 있음
선택 모드	사용자가 프롬프트에 필요한 값을 선택하는 방법을 정의합니다. 다음을 선택합니다. • Free: 사용자가 모든 값을 입력할 수 있습니다. • Constrained: 사용자가 제안된 값 중에서 선택해야 합니다. • Primary_key* 쿼리에서 기본 키 값을 사용하면 응답 시간이 크게 증가합니다. 유니버스의 개체에 대한 기본 키를 선택하거나 입력합니다. 프롬프트 시 사용자가 개체 이름을 선택해도 쿼리에서는 실제로 해당 기본 키 값을 사용합니다. *데이터베이스 테이블에 인덱스 인지가 설정된 경우에만 기본 키 설정을 사용할 수 있습니다. 기본값 = Free
마지막 값 선택 유지	이 옵션을 선택하면 사용자가 다음에 쿼리를 실행할 때 마지막 사용된 값이 제안됩니다. 이 옵션을 선택하지 않으면 항상 기본값이 사용됩니다. 기본값 = 선택하지 않음
표시 값	사용자는 값 목록에서 하나 이상의 항목을 선택해야 합니다. 다음과 같은 방법으로 값 목록을 정의할 수 있습니다. • 값 목록 상자에 값 목록을 입력합니다. 이는 정적 목록입니다. • 유니버스에서 개체를 선택합니다. • 파일 입력 마법사를 사용하여 파일을 가져옵니다. 기본값 = Static
기본값	프롬프트의 기본값을 선언할 수 있습니다.

관련 정보

@Prompt 함수 구문 [페이지 355]
SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]
프롬프트의 정적 값 목록 정의 [페이지 347]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
프롬프트에 대한 사용자 지정 값 목록 선택 [페이지 348]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
기존 @Prompt 식 편집 [페이지 349]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.3.3.3 프롬프트 편집기를 사용하여 @Prompt 식 만들기

유니버스가 유니버스 디자인 도구에서 열리고 개체를 만들거나 편집하고 있습니다.

1. 조건의 *SELECT* 또는 *WHERE* 필드에서 식에서 프롬프트를 추가할 위치를 마우스 오른쪽 단추로 클릭하고 바로 가기 메뉴에서 *@Prompt* 편집기를 선택합니다.
2. 메시지 상자에 사용자에게 표시할 메시지를 입력합니다.
3. 이전 보고서에서 사용한 값을 프롬프트에서 제안하려면 마지막으로 선택한 값 유지를 선택합니다.

   보고서를 처음 실행하는 경우에는 기본값(있는 경우)이 제안됩니다.
4. 사용자가 값을 둘 이상 입력하거나 선택할 수 있는 경우 여러 선택 허용을 선택합니다.
5. 선택 모드를 설정합니다. *Free* 를 선택하면 사용자가 허용되는 모든 값을 입력할 수 있습니다. *Constrained* 를 선택하면 사용자가 값 목록에서 값을 선택해야 합니다. *Primary key* 를 선택하면 사용자가 개체 이름을 선택해도 쿼리에는 개체의 기본 키가 사용됩니다. 인덱스 인지가 설정된 경우에만 *Primary key* 설정을 사용할 수 있습니다.

   > **노트**
   >
   > *Constrained* 옵션이 선택되었고 값 목록이 지정되지 않은 경우 해당 탭 색상이 빨간색으로 변하고 확인 단추가 비활성화됩니다. 또한 강조표시된 탭으로 커서를 이동하면 상황에 맞는 도구 설명에서 문제 해결 방법을 알려주는 메시지를 표시합니다.
6. 선택 모드의 값 형식을 *Alphanumeric*, *Number* 또는 *Date* 로 설정합니다.
7. 선택 모드로 기본 키를 선택한 경우 키 유형을 영숫자, 숫자 또는 날짜로 설정합니다.
8. 다중 선택을 사용하는 경우 값 목록을 정의합니다. 다음 중 하나를 수행합니다. 값 목록을 입력하거나 가져올 수도 있고, 유니버스 개체를 선택할 수도 있습니다.
9. 기본값 탭을 클릭하고 기본값을 정의합니다. 여러 선택을 허용하면 기본값을 하나 이상 설정할 수 있습니다.
10. 확인을 클릭합니다.
    @Prompt 편집기가 닫히고 속성 편집 창의 조건 문에 프롬프트 식이 표시됩니다.
11. 프롬프트의 유효성을 검사하고 구문 분석을 클릭합니다.
    구문이 잘못된 경우 오류가 들어 있는 SQL 부분을 알리는 구문 분석 실패 메시지가 표시됩니다.

관련 정보

@Prompt 함수 구문 [페이지 355]
SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]
프롬프트의 정적 값 목록 정의 [페이지 347]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
프롬프트에 대한 사용자 지정 값 목록 선택 [페이지 348]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
기존 @Prompt 식 편집 [페이지 349]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.3.3.4 프롬프트의 정적 값 목록 정의

@Prompt 편집기의 아래쪽 부분에는 정적 값 목록을 정의하는 데 사용되는 테이블 창이 있습니다. 위쪽 및 아래쪽 화살표를 사용하여 입력하는 값의 위치를 변경할 수 있습니다. 캡션 제목 텍스트가 빨간색일 경우 값을 입력하거나 값을 수정해야 합니다.

1. 캡션 필드에 첫 번째 값을 입력합니다. 기본 키를 선택 모드로 선택한 경우 두 번째 필드에 인덱스 값을 입력합니다.
2. +를 클릭하여 값을 정적 값 목록 표에 삽입합니다.
3. 표에 값을 추가로 입력하여 값 목록을 완성합니다.
4. 기본값 창에 기본값을 입력합니다.
   사용자가 보고서를 실행하면 기본값이 제안됩니다. 마지막으로 선택한 값 유지를 설정하면 마지막으로 보고서가 실행될 때 사용된 값이 제안됩니다. 그렇지 않으면 보고서를 실행할 때마다 기본값이 제안됩니다.

> **노트**
>
> 통계 목록의 값을 편집하려면 값을 선택 및 편집하고 업데이트 단추를 클릭합니다.

> **노트**
>
> 값을 삭제하려면 값을 선택하고 -를 클릭합니다.

관련 정보

@Prompt 함수 구문 [페이지 355]
SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
프롬프트에 대한 사용자 지정 값 목록 선택 [페이지 348]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
기존 @Prompt 식 편집 [페이지 349]

7.3.3.5 프롬프트의 값 목록으로 유니버스 개체 선택

프롬프트의 값 목록으로 유니버스 개체를 선택할 수 있습니다.

1. *@Prompt* 편집기의 값 목록 창에서 유니버스 개체를 선택합니다.
2. 프롬프트에서 사용할 유니버스 개체를 찾아 선택합니다.
3. 개체를 두 번 클릭합니다.
   @Prompt 편집기가 닫히고 속성 편집 창의 조건 문에 프롬프트 식이 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

@Prompt 함수 구문 [페이지 355]
SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]
프롬프트의 정적 값 목록 정의 [페이지 347]
프롬프트에 대한 사용자 지정 값 목록 선택 [페이지 348]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
기존 @Prompt 식 편집 [페이지 349]

7.3.3.6 프롬프트에 대한 사용자 지정 값 목록 선택

데이터 파일 입력 마법사를 사용하면 사용자 지정 값 목록을 *@Prompt* 편집기의 캡션 테이블로 가져와서 `@Prompt` 함수에 삽입할 수 있습니다. Excel 및 텍스트 파일만 지원됩니다.

1. *@Prompt* 편집기의 값 목록 창에서 파일 입력을 선택합니다.
   데이터 파일 입력 마법사가 나타납니다.
2. 찾아보기를 클릭하고 사용할 입력 파일을 찾습니다.
3. 열기를 클릭합니다.
4. 다음을 클릭합니다.
5. 값 파일 목록의 첫 번째 행이 목록 열의 머리글 또는 제목 값으로 사용되는 경우 첫 번째 행으로 열 머리글 지정을 선택합니다.
6. 파일 인코딩 형식을 선택합니다.
7. 입력 파일에서 데이터를 구분하는 데 사용되는 구분 기호 문자를 설정합니다.
8. 다음을 클릭합니다.
9. *Get unique column values(고유한 열 값 가져오기)*를 클릭하여 고유 값만 제안되도록 설정합니다.
10. *Get number of TOP records(최상위 레코드 수 가져오기)*를 클릭하여 사용자에게 제안할 값 수를 정의합니다.
11. *Column map(열 맵)* 창을 사용하여 캡션 및 기본 키 값에 사용되는 열을 선택합니다.
12. *Sort on column(열별 정렬)*을 클릭하여 선택한 열을 정렬합니다. 열을 오름차순으로 정렬하려면 오름차순을 선택합니다. 오름차순을 선택하지 않으면 열이 내림차순으로 정렬됩니다.
13. 마침을 클릭합니다.
    Column row append(열 행 추가) 확인 메시지가 표시됩니다. 예를 클릭하여 선택을 확인합니다.
14. 사용자 지정 값 목록이 *@Prompt* 편집기의 값 목록 테이블에 삽입되고 `@Prompt` 정의에 값이 삽입됩니다.
15. 확인을 클릭하여 *SELECT* 또는 *WHERE* 절에 `@Prompt` 를 삽입합니다.

관련 정보

@Prompt 함수 구문 [페이지 355]
SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]
프롬프트의 정적 값 목록 정의 [페이지 347]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
기존 @Prompt 식 편집 [페이지 349]

7.3.3.7 기존 @Prompt 식 편집

유니버스가 유니버스 디자인 도구에서 열립니다. 선택한 개체 식에는 `@Prompt` 함수가 들어 있습니다.

이미 정의 내에 있거나 개체 또는 조건의 where 절에 있는 `@Prompt` 함수를 편집하려고 합니다. *@Prompt* 편집기를 통해 편집할 수 있습니다.

1. `@Prompt` 함수를 마우스 오른쪽 단추로 클릭하고 바로 가기 메뉴에서 프롬프트 편집을 선택합니다.
2. *@Prompt* 편집기를 사용하여 프롬프트 식을 편집합니다.
3. 구문 분석을 클릭하여 `@Prompt` 의 구문을 확인합니다.

관련 정보

@Prompt 함수 구문 [페이지 355]
SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]
프롬프트의 정적 값 목록 정의 [페이지 347]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]
프롬프트에 대한 사용자 지정 값 목록 선택 [페이지 348]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]

7.3.3.8 SQL 문의 @Prompt 함수를 수동으로 정의

> **노트**
>
> 이전에 @Prompt 함수를 정의하지 않은 경우 @Prompt 편집기를 사용하는 것이 좋습니다. @Prompt 함수 구문은 복잡하므로 구문 정의 페이지를 참조하십시오.

개체에 대해 `@Prompt` 함수를 수동으로 정의하려면 다음 작업을 수행하십시오.

1. 유니버스 창에서 클래스, 개체 또는 조건을 두 번 클릭하거나 개체를 마우스 오른쪽 단추로 클릭하고 개체 속성을 선택하거나 편집 > 속성 을 선택하여 속성 편집 창을 엽니다.
2. 정의 창의 SELECT 또는 WHERE 대화 상자를 클릭하고 필요한 구문(아래 링크 참조)에 따라 @Prompt 값을 정의합니다.
3. 구문 분석을 클릭하여 SQL 의 구문을 확인합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

@Prompt 함수 구문 [페이지 355]
@Prompt 편집기 [페이지 344]
기존 @Prompt 식 편집 [페이지 349]
프롬프트의 정적 값 목록 정의 [페이지 347]
프롬프트의 값 목록으로 유니버스 개체 선택 [페이지 347]

7.3.3.9 프롬프트를 수동으로 정의하는 @Prompt 식 속성

@Prompt 구문은 다음과 같습니다.

```
@Prompt('message','type','lov',Mono|Multi,free|constrained|primary_key
,persistent|not_persistent,{'default value'})
```

예는 다음과 같습니다.

```
@Prompt('Displayed text ','A','Store\City',Mono,constrained,Persistent,{'Paris'})
```

다음 표에서는 @Prompt 식 값의 속성을 보여줍니다.

표 156:

속성	설명
'message'	필수 프롬프트 메시지 텍스트입니다. 텍스트는 작은 따옴표로 묶어야 합니다(예: '메시지 선택', '기간 선택' 또는 '전시실 선택'). 사용자가 쿼리를 실행하면 프롬프트 상자에 텍스트가 나타납니다.
'type'	필수 항목이지만 비워 둘 수 있습니다('A' 설정이 기본적으로 사용됨). 세 번째 매개 변수의 데이터 형식입니다. 형식은 다음 중 하나일 수 있습니다. • 'A': 영숫자 • 'N': 숫자 • 날짜의 경우 'D' 지정한 데이터 형식은 작은 따옴표로 묶어야 합니다. 하드 코딩된 값 쌍 목록을 사용하는 경우의 구문은 'value_type:key_type' 입니다(예: 'A:N'). 여기에서 첫 번째 값은 최종 사용자에게 표시되는 캡션이고, 두 번째 값은 쿼리 속도를 높이기 위해 쿼리에서 실제로 사용하는 기본 키 값입니다. 캡션 및 기본 키의 각 형식은 위의 설명과 같이 A, N 또는 D 일 수 있습니다. 예를 들면 'A:A' 또는 'A:N'과 같습니다. 이 경우 다음 매개 변수인 'lov'에는 매개 변수 쌍 목록이 들어 있습니다. 마찬가지로, 'default_value' 매개 변수는 값 쌍을 포함하게 됩니다. 기본 키를 사용하는 경우 인덱스 인지를 설정해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

속성	설명
lov	필수 항목이지만 비워 둘 수 있습니다. 빈 목록을 사용하는 경우에도 쉼표를 생략해서는 안 됩니다. 이 매개 변수가 유니버스 개체인 경우 5 번째 매개 변수(선택 모드 = free|constrained|primary key)는 primary_key 여야 하고 유니버스에 인덱스 인지가 설정되어야 합니다. 두 가지 유형의 값 목록을 지정할 수 있습니다. • 기존 유니버스 개체에서 값 목록으로의 포인터입니다. 클래스 및 개체 패널에서 사용하려는 값 목록이 들어 있는 개체를 두 번 클릭하여 대상 값 목록을 호출합니다. 그러면 클래스 이름과 개체 이름이 백슬래시로 구분되어 표시됩니다. 값은 작은 따옴표로 묶어야 합니다. 예를 들어, '클라이언트\국가'와 같은 형식입니다. 인덱스 인식을 사용하고 개체의 키 값을 반환하려는 경우 다섯 번째 값을 primary_key 로 설정합니다. • 하드 코딩된 단일 값 또는 값 쌍 목록입니다. 쌍을 이루는 값은 콜론으로 구분하고, 각 값은 작은 따옴표로 묶습니다. 값 쌍은 쉼표로 구분합니다. 목록 전체는 중괄호로 묶어야 합니다. primary_key 에 제약 사항을 설정합니다. 단일 기본값에 대한 구문: {'value'} 단일 기본값 여러 개에 대한 구문: {'value1','value2',... ,'valuen'} 기본값 쌍을 정의할 수 있습니다. 기본값 쌍에 대한 구문: {'value':'key'} 값과 키 간의 구분 기호는 콜론(:)입니다. 기본값 쌍에 대한 구문: {'value1':'key1','value2':'key2',...,'valuen':'keyn'} 예:{'Australia':'A', 'France':'F', 'Germany':'G', 'Japan':'J', 'Spain':'S', 'United Kingdom':'UK'}
Mono|Multi	필수 항목이지만 비워 둘 수 있습니다(Mono 설정이 기본적으로 사용됨). 쉼표를 생략해서는 안 됩니다. 사용자가 값 목록에서 값을 하나만 선택할 수 있는 경우에는 Mono 를 사용하십시오. 사용자가 값 목록에서 값을 여러 개 선택할 수 있는 경우에는 Multi 를 사용하십시오.
free|constrained|primary_key	필수 항목이지만 비워 둘 수 있습니다(free 설정이 기본적으로 사용됨). 쉼표를 생략해서는 안 됩니다. 사용자가 값을 직접 입력하거나 값 목록에서 선택할 수 있는 경우에는 free 를 사용하십시오. 사용자가 값 목록에서 값을 선택해야 하는 경우에는 constrained 를 사용하십시오. 유니버스에서 인덱스 인식을 설정할 때에는 primary_key 매개 변수를 사용하십시오. 입력된 값이나 표시된 값이 아닌 개체의 연결된 키 값이 사용됩니다. lov 매개 변수를 사용할 경우 primary_key 항목을 생략해서는 안 됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

속성	설명
`persistent|not_persistent`	(선택 사항) 이 항목이 설정된 경우 인수를 쉼표로 끝냅니다. 이 항목이 설정되어 있지 않지만 7 번째 매개 변수(기본값)가 설정된 경우 쉼표를 생략할 수 없습니다. **노트** Desktop Intelligence 에서 이 매개 변수는 어떠한 영향도 주지 않습니다. 문서를 새로 고칠 때 기본값이 정의되어 있어도 프롬프트에 사용된 마지막 값이 기본값으로 표시되는 경우에는 `persistent` 를 사용하십시오. 문서를 새로 고칠 때 프롬프트에 기본값으로 표시된 값이 사용되지 않는 경우에는 `not_persistent` 를 사용하십시오.
`'default value'`	(선택 사항) `'default value'` 매개 변수는 사용자에게 표시되는 기본값을 정의하는 데 사용됩니다. 하드 코딩된 목록을 사용하는 경우 여기에 입력하는 기본값이 `[lov]` 목록에 있어야 합니다. 단일 값의 예: {'France'} 값 쌍 한 개의 예: {'France':'F'} 값 쌍 두 개의 예: {'France':'F','Germany':'G'} 문서를 새로 고칠 때 이러한 값이 기본값으로 표시되지만 `persistent` 옵션이 설정되어 있으면 프롬프트에 사용된 마지막 값이 기본값 대신 사용됩니다. 단일 값이나 값 쌍을 사용할 수 있습니다. 프롬프트 정의에 `primary_key` 매개 변수를 지정한 경우 키 값을 제공해야 합니다.

7.3.3.10 예제: @Prompt 함수 사용

다음은 @Prompt 구문의 예입니다.

@Prompt 함수의 가장 단순한 사용:

```
@Prompt('Displayed text ','A',,,)
```

기본값이 없는 값 목록과 함께 @Prompt 사용:

```
@Prompt('Displayed text ','A',{'Paris','London','Madrid'},,)
```

값 목록 및 기본값 하나와 함께 @Prompt 사용:

```
@Prompt('Displayed text ','A',{'Paris','London','Madrid'},,,,{'Paris'})
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

개체 및 기본값과 함께 @Prompt 사용:

```
@Prompt('Displayed text ','A','Store\City',,,,{'Paris'})
```

가능한 모든 설정과 함께 @Prompt 사용:

```
@Prompt('Displayed text ','A','Store\City',Mono,Constrained,Persistent,{'Paris'})
```

기본값 없이 값 쌍을 포함하는 값 목록과 함께 @Prompt 사용:

```
@Prompt('Displayed text ','A:N',{'Paris':'12','London':'7','Madrid':'15'},,)
```

값 쌍을 포함하는 값 목록 및 값 쌍 기본값 하나와 함께 @Prompt 사용:

```
@Prompt('Displayed text ','A:N',{'Paris':'12','London':'7','Madrid':'15'},,,,
{'Paris':'12'})
```

예

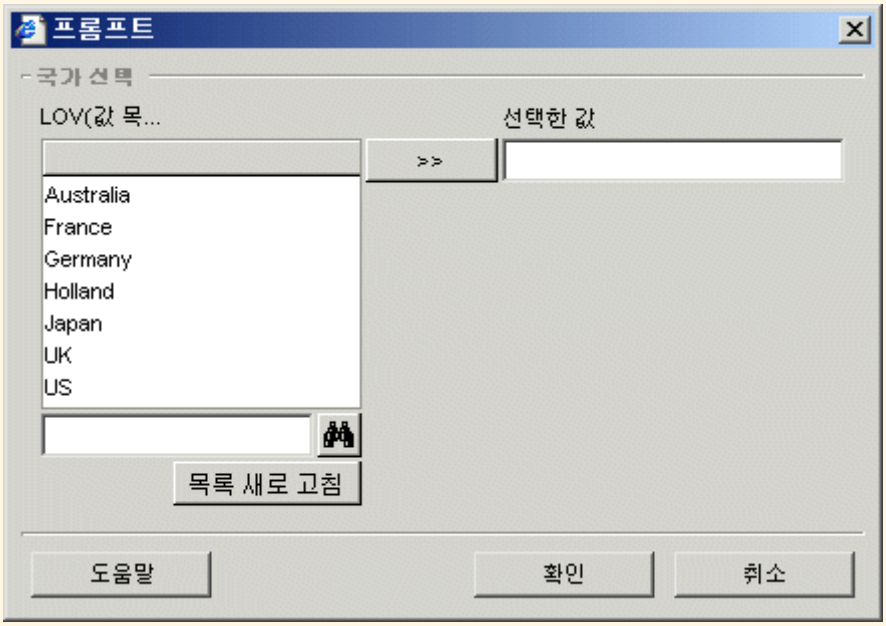
@Prompt 를 사용하여 기본값 없이 국가 제한

국가 개체는 휴양지 국가에 대한 값을 반환합니다. 특정 국가의 휴양지로 반환 값을 제한하려면 유니버스의 각 휴양지 국가에 대해 개체가 하나씩 있어야 합니다. 그러나 `@Prompt` 를 사용하면 다음과 같이 개체 하나만 있으면 됩니다.

```
Country.country=@prompt('Choose a country',
'A','Customer\Country of origin',Mono,primary_key,,,)
```

즉, 국가 이름을 입력하라는 메시지가 표시되고, 반환 값은 해당 특정 국가의 휴양지가 됩니다.

Web Intelligence 에서 쿼리를 실행하면 다음과 같은 프롬프트 상자가 나타납니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

예

기본값이 있는 @Prompt 구문

```
@prompt('Enter value(s) for Customer with IA:',
'A','Customer\Customer with I A',
Multi,primary_key,,{'Baker','Brendt'})
```

예

하드 코딩된 값 목록을 사용하는 @Prompt 구문

다음 예제에서는 기본값이 있는 국가 목록을 입력하는 방법을 보여 줍니다. 개체에 인덱스 인식 기능이 있고 제약 조건이 primary_key 로 설정되어 있으면 기본값이 쌍(값, 키) 집합이 될 수 있습니다(예: {'England':'21', 'Scotland':'39'}). 사용자가 지역을 하나만 선택해야 하므로 Mono 매개 변수를 설정합니다. 기본값은 값 목록에 표시되어야 합니다.

```
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'})
```

이 기능을 사용하면 OLAP 데이터베이스와 같이 CASE WHEN ELSE 절을 지원하지 않는 데이터베이스에서 해당 동작을 수행할 수 있습니다.

예

패턴 일치 프롬프트를 사용하여 미리 정의된 조건을 만드는 @Prompt 구문

다음 예제에서는 사용자가 이름의 첫 문자를 입력하여 고객 이름을 선택할 수 있습니다. Web Intelligence 사용자가 H%를 입력하면 성이 H 로 시작하는 모든 고객이 보고서에 반환됩니다.

```
(@Select(Client\Client Name)
    LIKE (@Prompt('enter','A',,,)+%)
```

고객이 대문자나 소문자를 사용할 수 있으므로 구문은 다음과 같습니다.

```
(@Select(Client\Client Name)
    LIKE lower(@Prompt('enter','A',,,)+%) OR
 (@Select(Client\Client Name)
    LIKE upper(@Prompt('enter','A',,,)+%)
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.3.3.11 @Prompt 함수 구문

`@Prompt` 함수는 용도가 다양하므로 구문이 복잡합니다. 프롬프트 메시지를 작성하고 데이터 형식, 그리고 데이터가 단일 값인지 다중 값인지 여부와 영구 데이터인지 여부를 지정할 수 있습니다. 기본값도 지정할 수 있습니다. 구문은 아래와 같습니다.

```
@Prompt('message','type',[lov],Mono|Multi,
free|constrained|primary_key,persistent|not_persistent,[default_values])
```

관련 정보

@Prompt [페이지 343]
SQL 문의 @Prompt 함수를 수동으로 정의 [페이지 349]
프롬프트를 수동으로 정의하는 @Prompt 식 속성 [페이지 350]

7.3.4 @Script

`@Script` 함수는 VBA(Visual Basic for Applications) 매크로의 결과를 반환합니다. VBA 매크로는 Windows 환경에서만 실행할 수 있습니다. `@Script` 함수를 사용하면 개체가 포함된 쿼리를 새로 고치거나 실행할 때마다 지정된 VBA 매크로가 실행됩니다.

일반적으로 `@Script` 함수는 WHERE 절에 사용하며 간단한 프롬프트 상자를 표시하는 `@Prompt` 함수보다 복잡한 작업을 수행합니다. VBA 매크로는 BusinessObjects 보고서 파일(.REP)에 저장됩니다. 이러한 보고서의 기본 디렉토리는 BusinessObjects 경로의 UserDocs 폴더이지만 .REP 파일을 저장할 다른 폴더를 정의할 수도 있습니다.

> **노트**
>
> `@Script` 는 유니버스 디자인 도구 및 클라이언트 버전의 Desktop Intelligence 또는 Desktop Intelligence 3-Tier 모드에서만 지원됩니다. 클라이언트 버전의 Desktop Intelligence 이외에서 사용하려는 경우에는 `@Script` 함수를 사용하지 않는 것이 좋습니다. Desktop Intelligence 보고서의 게시 또는 예약을 위해 InfoView 에서 사용할 수 있는 서버 버전의 Desktop Intelligence 및 Web Intelligence 에서는 지원되지 않습니다. Web Intelligence 의 경우 대화형 개체에 대해 `@Script` 함수를 사용하지 말고 `@Prompt` 함수를 사용하여 보다 간단한 디자인을 유지해야 합니다.

7.3.4.1 @Script 함수 구문

`@Script` 함수의 구문은 다음과 같습니다.

```
@Script('var_name', ['var_type'], 'script_name')
```

> **노트**
>
> 두 번째 인수는 선택적이며 생략하더라도 쉼표를 구분 기호로 입력해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

다음 표에서는 이 구문을 설명합니다.

표 157:

구문	설명
`'var_name'`	매크로에 선언된 변수 이름입니다. 이 이름은 실행된 매크로 결과를 개체의 SQL 정의에 복원하는 데 사용됩니다. 이 이름은 VBA 매크로와 개체의 SQL 정의에서 동일해야 합니다.
`'var_type'`	(옵션) 함수에서 반환되는 데이터 형식입니다. 형식은 다음 중 하나일 수 있습니다. • 'A': 영숫자 • 'N': 숫자 • 날짜의 경우 'D' 지정한 데이터 형식은 작은 따옴표로 묶어야 합니다.
`'script_name'`	실행할 VBA 매크로의 이름입니다.

7.3.5 @Select

`@Select` 함수를 사용하면 다른 개체의 SELECT 문을 다시 사용할 수 있습니다. 개체의 SELECT 문에 `@Select` 함수를 사용하면 유니버스 내에 있는 다른 개체의 경로가 클래스 이름\개체 이름 형식으로 `@Select` 함수의 매개 변수로 지정됩니다. 이 매개 변수는 참조된 개체의 SELECT 문에 대한 포인터 역할을 합니다.

`@Select` 함수의 사용은 다음과 같은 이점을 제공합니다.

- SQL 코드 인스턴스 하나만 유지하면 됩니다.
- 코드 일관성을 유지할 수 있습니다.

> **노트**
>
> `@Select` 및 `@Where` 함수를 사용하면 개체가 유니버스 내의 다른 개체에 종속됩니다. 즉, 새 개체 종속성이 만들어집니다. 개체 하나를 삭제하면 `@Select` 또는 `@Where` 함수를 사용하는 나머지 개체를 수동으로 업데이트해야 합니다.

7.3.5.1 @Select 함수 구문

`@Select` 함수의 구문은 다음과 같습니다.

```
@Select(Classname\Objectname)
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 158:

구문	설명
`Classname`	참조할 개체를 포함하는 클래스의 이름입니다.
`Objectname`	참조한 개체의 이름입니다.

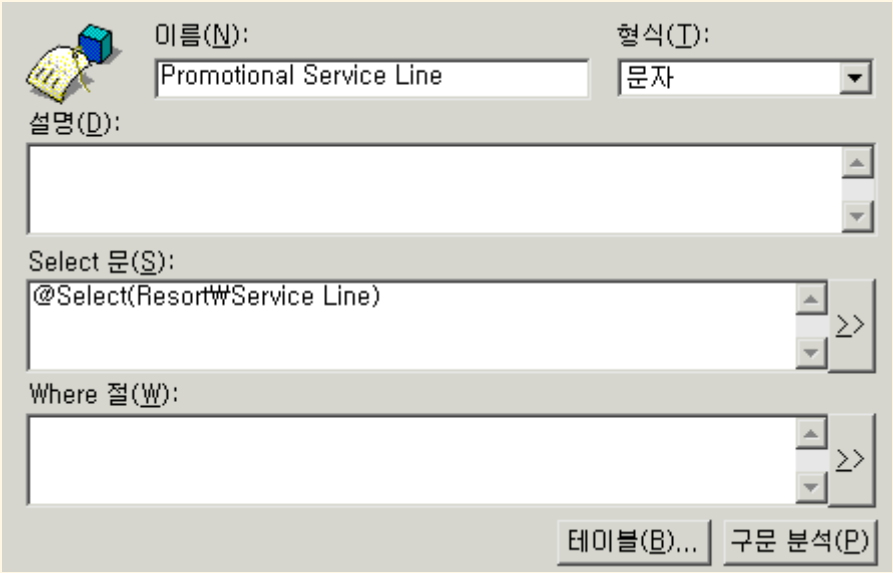
7.3.5.2 @Select 함수 사용의 예

예

`@Select` 를 사용하여 Service_line Select 문 다시 사용

Club 데이터베이스에 있는 여러 휴양지의 홍보 캠페인에 사용된 서비스를 반환하는 데 사용할 Promotional Service Line 이라는 개체를 만듭니다. 이 개체는 새 클래스인 홍보에 포함되어 있습니다. `@Select` 를 사용하여 Service_lines 개체의 기존 SELECT 문을 참조할 수 있습니다.

다음 그림은 Promotional Service Line 의 SELECT 문을 보여 줍니다.



7.3.6 @Variable

예를 들어 `@Variable` 함수는 WHERE 절에서 다음과 같은 변수 형식 중 하나에 지정된 값을 호출하는 데 사용됩니다.

- BusinessObjects 시스템 변수
- 보고서 변수

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 언어(로캘) 변수
- 운영 체제 변수
- Desktop Intelligence 의 사용자 지정 변수

대부분의 경우 개체에 대한 속성 편집 시트의 정의 페이지에서 개체의 WHERE 절에 있는 조건의 피연산자쪽에 `@Variable` 을 삽입합니다. 쿼리는 변수의 값을 검색합니다.

> **노트**
>
> `@Variable` 은 단일 값 함수이며 IN 또는 INLIST 연산자와 함께 사용할 수 없습니다.

> **노트**
>
> 쿼리에서 동일한 `@Variable` 함수를 여러 번 실행해도 프롬프트는 한 번만 나타납니다.

> **노트**
>
> `@Variable` 함수는 다음과 같이 설정한 단일 값 `@Prompt` 함수와 동일합니다.
>
> ```
> @Prompt('Question','A',,mono,free)
> ```
>
> `@Prompt` 함수가 단일 값인 경우 같은 쿼리에서 `@Variable` 함수와 `@Prompt` 함수를 병합할 수 있습니다.

관련 정보

@Variable 함수 구문 [페이지 358]
@Variable 속성 설명 [페이지 359]

7.3.6.1 @Variable 함수 구문

`@Variable` 함수의 구문은 다음과 같습니다.

```
@Variable('<Variablename>')
```

> **노트**
>
> 변수 이름은 작은 따옴표로 묶어야 합니다.

> **예**
>
> **BOUSER 값을 반환하는 @Variable 구문**
>
> ```
> @Variable('BOUSER')
> ```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

@Variable [페이지 357]
@Variable 속성 설명 [페이지 359]

7.3.6.2 @Variable 속성 설명

모든 경우에 변수 이름을 작은 따옴표로 묶어야 합니다.

표 159:

변수 이름	설명
BusinessObjects 시스템 변수 • BOUSER - 사용자 로그인 • DBUSER - 데이터베이스 사용자 이름 • DBPASS - 데이터베이스 사용자 암호	BusinessObjects 시스템 변수의 값입니다. 반환된 데이터는 해당 BusinessObjects 사용자의 로그인을 기준으로 제한됩니다. BusinessObjects 선언 데이터베이스 사용자에 대한 값입니다.
보고서 변수 • DOCNAME - 문서의 이름 • DPNAME - 데이터 공급자의 이름 • DPTYPE - 데이터 공급자의 유형 • UNVNAME - 유니버스의 이름 • UNVID - 사용된 유니버스의 ID	예를 들어, SELECT 문 이전에 실행되는 Begin_SQL 매개 변수에서 이러한 변수를 참조할 수 있습니다. 이 변수는 데이터베이스 사용과 관련된 감사 용도에 사용할 수 있습니다(예: 가장 많이 사용하는 보고서 쿼리나 유니버스의 종류 확인).
언어 변수 • PREFERRED_VIEWING_LOCALE • DOMINANT_PREFERRED_VIEWING_LOCALE	언어 변수 • PREFERRED_VIEWING_LOCALE - 사용자의 기본 설정 표시 로캘. 이는 사용자가 유니버스 메타데이터를 표시하도록 선택한 로캘입니다. • DOMINANT_PREFERRED_VIEWING_LOCALE - 사용자의 기본 설정 표시 로캘의 주 로캘. 이는 사용자가 데이터를 모든 로캘(fr_FR, fr_BE, fr_CA, ...)로 번역하지 않게 합니다. fr_FR 의 번역을 사용할 수 있는 경우 사용자 로캘이 fr_BE 또는 fr_CA 이면 같은 주 로캘을 공유하므로 fr_FR 의 번역을 다시 사용할 수 있습니다.
운영 체제 변수	Windows 환경 변수를 입력하여 설치 정보를 가져올 수 있습니다.
사용자 지정 변수	Desktop Intelligence 의 경우 미리 정의된 텍스트 파일을 사용하여 고정 변수 값 목록을 제공할 수 있습니다.

관련 정보

@Variable [페이지 357]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

@Variable 함수 구문 [페이지 358]

7.3.6.3 BusinessObjects 시스템 변수 사용

`@Variable` 함수를 BusinessObjects™ 시스템 변수와 사용하면 현재 로그인한 BusinessObjects™ 사용자의 ID 를 기준으로 데이터를 제한할 수 있습니다.

> **노트**
>
> BusinessObjects™ 로그인 매개 변수와 데이터베이스 로그인 매개 변수가 동일해야 합니다.

각 BusinessObjects™ 사용자에게 할당된 사용자 이름은 다음과 같은 BusinessObjects™ 시스템 매개 변수로 저장됩니다.

- `BOUSER` - 사용자 이름

이 매개 변수는 Business Objects 제품에 로그온할 때 사용자 ID 상자에 표시됩니다.

개체의 WHERE 절에 `@Variable` 함수를 사용하면 해당 개체를 쿼리에 사용했을 때 자신의 데이터베이스 프로필에만 사용자의 데이터 액세스를 제한할 수 있습니다.

개체에 대한 속성 편집 시트의 정의 페이지에서 개체의 WHERE 절에 있는 조건의 피연산자쪽에 `@Variable` 을 삽입합니다.

> **예**
>
> `@Variable` 을 사용하여 직원 데이터에 대한 직원의 액세스 제한
>
> 인사 관리 데이터베이스 유니버스에 직원 이름이라는 개체가 있습니다. 직원 이름의 반환 데이터를 데이터베이스에서 각 사용자에게 허용된 값으로 제한하려고 합니다. 이렇게 하면 각 사용자가 볼 수 있는 직원 정보를 제어할 수 있습니다. 이 정보는 각 직원의 데이터베이스 프로필을 통해 정의됩니다.
>
> 다음과 같이 `@Variable` 함수를 WHERE 절에 삽입합니다.
>
> ```
> Employees.Employee_Name = @Variable('BOUSER')
> ```
>
> 쿼리에 직원 이름 개체를 사용하면 `BOUSER` 값과 일치하는 값에 대한 데이터만 테이블에서 반환됩니다.

관련 정보

@Variable [페이지 357]
@Variable 함수 구문 [페이지 358]
@Variable 속성 설명 [페이지 359]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.3.6.4 로캘 변수 사용

`@Variable` 함수의 로캘 변수를 사용하여 로캘 설정을 정의하면 Web Intelligence 에서 적절한 로캘로 보고서를 검색하고 정보를 표시할 수 있습니다. 데이터베이스 테이블에는 데이터 번역이 들어 있는 행의 언어를 선언하는 열이 포함되어 있어야 합니다. 로캘은 언어 및 지리적 장소, 데이터의 정렬 방식, 날짜 서식 및 기타 특정 형식을 정의합니다. 개체의 WHERE 절에 `@Variable` 함수를 사용합니다. 이렇게 하면 쿼리에 개체가 사용된 경우 사용자가 로캘을 선택해야 합니다. 사용자가 쿼리를 실행하면 사용자의 로캘 입력을 요청하는 프롬프트 상자가 나타납니다. 번역 관리 도구 가이드에 로캘 코드 및 주요 로캘 코드 목록이 나와 있습니다.

정의할 수 있는 설정은 다음과 같습니다.

- `@Variable('PREFERRED_VIEWING_LOCALE')`
- `@Variable('DOMINANT_PREFERRED_VIEWING_LOCALE')`

예

아래의 PRODUCT 테이블은 여러 언어로 번역되었습니다. 사용자는 제품 이름을 특정 로캘로 나열하려고 합니다.

표 160: PRODUCT 테이블

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
...	...	...

```
SELECT Product_Name

FROM PRODUCT

WHERE PRODUCT.LOCALE = @Variable('PREFERRED_VIEWING_LOCALE')
```

쿼리 시 사용자가 변수를 올바른 로캘 값으로 바꾸면 Web Intelligence 에서 해당 로캘로 정보를 검색합니다.

관련 정보

@Variable [페이지 357]
@Variable 함수 구문 [페이지 358]
@Variable 속성 설명 [페이지 359]

7.3.6.5 보고서 변수 사용

보고서 변수를 요청에 포함시키려면 개체의 WHERE 절에 `@Variable` 함수를 사용합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

SELECT 문 이전에 실행되는 `Begin_SQL` 매개 변수에서 이러한 변수를 참조할 수 있으며, 이 변수는 데이터베이스 사용과 관련된 감사 용도에 사용할 수 있습니다(예: 가장 많이 사용하는 보고서 쿼리나 유니버스의 종류 확인).

다음 조건에서 변수를 참조할 수 있습니다.

- 개체의 정의: SELECT, WHERE 절 등
- 필터
- 조인 식
- Begin_SQL 매개 변수

관련 정보

@Variable [페이지 357]
@Variable 함수 구문 [페이지 358]
@Variable 속성 설명 [페이지 359]

7.3.6.6 운영 체제 변수 사용

Windows 환경 변수를 입력하여 설치 정보를 가져올 수 있습니다. NUMBER_OF_PROCESSORS, USERNAME 과 같은 변수를 예로 들 수 있습니다.

예

다음 `@Variable(NUMBER_OF_PROCESSORS)` 을 쿼리에 포함시킬 경우, 응답에 현재 사용 중인 컴퓨터 프로세서의 수가 포함됩니다.

관련 정보

@Variable [페이지 357]
@Variable 함수 구문 [페이지 358]
@Variable 속성 설명 [페이지 359]

7.3.6.7 Desktop Intelligence 에서 사용자 지정 변수 사용

Desktop Intelligence 에서 개체의 WHERE 절에 `@Variable` 함수를 사용하면 관련 텍스트 파일에 있는 변수를 참조할 수 있습니다. 이렇게 하면 개체에 사용자별 조건을 정의할 수 있습니다.

이 변수를 사용하려면 명령줄에서 `-vars` 매개 변수를 사용하여 BusinessObjects 를 시작해야 합니다. 이 기능을 사용하는 모든 컴퓨터에서 Windows 바로 가기의 명령줄을 변경해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

> **노트**
>
> 유니버스 배포의 사용자 수가 많은 경우에는 명령줄에서 BusinessObjects 를 시작하도록 설정하면 `@Variable` 함수를 유지하는 데 어려움이 있습니다. 사용자 수가 많거나 사용자가 여러 지역에 분포되어 있는 경우에는 제한을 구현하는 데 `@functions` 와 관련 텍스트 파일을 사용하지 않는 것이 좋습니다.

텍스트 파일 변수와 함께 `@Variable` 함수를 사용할 때의 장점:

`@Variable` 함수를 텍스트 파일 변수와 함께 사용하면 유니버스를 변경하지 않고도 텍스트 파일의 변수 값을 업데이트할 수 있다는 가장 큰 이점이 있습니다.

텍스트 파일 변수와 함께 `@Variable` 함수를 사용할 때의 단점:

- `-vars <textfile.txt>` 인수를 포함하도록 모든 클라이언트에서 명령 문자열을 변경해야 합니다.
- 컴퓨터에 있는 텍스트 파일을 로컬로 수정할 수 있으므로 보안상 문제가 될 수 있습니다.

`@Variable` 함수를 텍스트 변수와 사용할 때 이와 같이 몇 가지 문제가 발생할 수 있으므로 엔터프라이즈 환경에서 Business Objects 제품을 사용하는 경우에는 Supervisor 에 제공되는 보안 옵션을 사용하여 데이터에 대한 액세스를 제어하는 것이 좋습니다.

관련 정보

Desktop Intelligence 에서 사용자 지정 변수 사용 [페이지 363]
@Variable [페이지 357]
@Variable 함수 구문 [페이지 358]
@Variable 속성 설명 [페이지 359]

7.3.6.7.1 Desktop Intelligence 에서 사용자 지정 변수 사용

하나 이상의 미리 정의된 변수 값을 사용하려면 다음 단계를 수행하십시오.

1. 변수와 해당 값 목록이 들어 있는 텍스트 파일을 만듭니다. 변수 이름 = 값의 형식을 따릅니다.
2. BusinessObjects 를 시작하는 데 사용하는 명령줄에 다음을 추가합니다. `Busobj.exe -vars <textfile.txt>` 예를 들어, Bovars.txt 라는 텍스트 파일이 있는 경우 다음을 입력합니다. `C:\BusinessObjects\Busobj.exe -vars Bovars.txt` 여기에서 `-vars` 구문은 운영 체제에서 텍스트 파일을 메모리로 로드하여 BusinessObjects 가 사용할 수 있도록 명령하는 스위치입니다.
3. 텍스트 변수를 참조할 개체의 속성 편집 시트를 엽니다.
4. WHERE 절에서 조건의 피연산자 쪽에 `@Variable` 을 삽입합니다. 예: COUNTRY.COUNTRY_NAME = `@Variable`('국가'). Country 는 텍스트 파일에 있는 변수의 이름입니다. 이 이름은 작은 따옴표로 묶어야 합니다.
5. 확인을 클릭한 다음 유니버스를 저장합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.3.7 @Where

`@Where` 함수를 사용하면 다른 개체의 WHERE 절을 다시 사용할 수 있습니다. 개체의 WHERE 절에 `@Where` 함수를 사용하면 유니버스 내에 있는 다른 개체의 경로가 **클래스 이름\개체 이름** 형식으로 `@Where` 함수의 매개 변수로 지정됩니다. 이 매개 변수는 참조된 개체의 WHERE 절에 대한 포인터 역할을 합니다.

WHERE 절을 사용하면 두 개체 사이에 동적 연결이 만들어집니다. 원본 개체의 WHERE 절을 수정하면 참조되는 개체의 WHERE 절이 자동으로 업데이트됩니다.

`@Where` 함수를 사용하면 기존 코드를 사용할 수 있습니다. 이는 다음과 같은 이점을 제공합니다.

- SQL 코드 인스턴스 하나만 유지하면 됩니다.
- 코드 일관성을 유지할 수 있습니다.

`@Select` 및 `@Where` 함수를 사용하면 개체가 유니버스 내의 다른 개체에 종속됩니다. 즉, 새 개체 종속성이 만들어집니다. 개체 하나를 삭제하면 `@Select` 또는 `@Where` 함수를 사용하는 나머지 개체를 수동으로 업데이트해야 합니다.

7.3.7.1 @Where 함수 구문

이 함수의 구문은 다음과 같습니다.

```
@Where(Classname\Objectname)
```

구문	설명
`Classname`	클래스의 이름입니다.
`Objectname`	참조한 개체의 이름입니다.

7.3.7.2 예: @Where 함수를 사용하여 WHERE 절 다시 사용

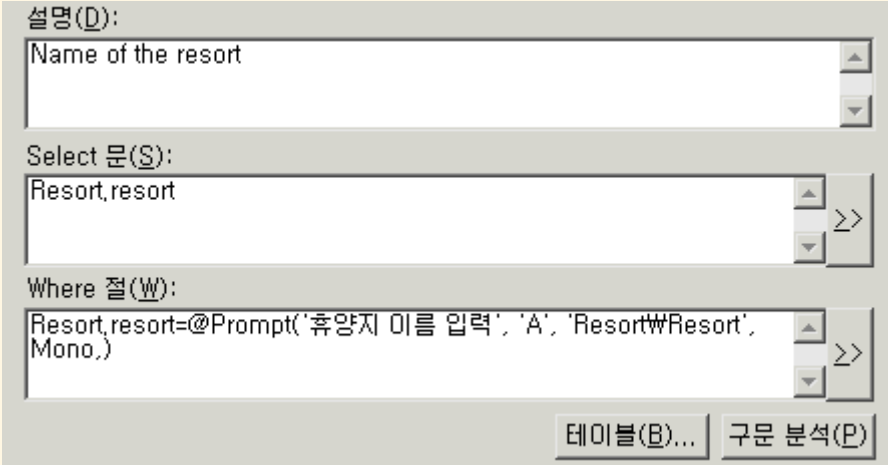
예

`@Where` 를 사용하여 휴양지 WHERE 절 다시 사용

각 휴양지에서 제공하는 서비스를 반환하는 데 사용할 휴양지 서비스 분야라는 개체를 만듭니다. 사용자가 특정 휴양지의 서비스를 쿼리할 때 해당 휴양지 이름을 입력하는 프롬프트가 표시되도록 휴양지 개체에 정의되어 있는 `@Prompt` 함수를 다시 사용하려고 합니다.

휴양지 개체(참조하려는 개체)의 SQL 은 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



새 휴양지 서비스 분야 개체에서는 다음과 같이 휴양지의 WHERE 절에 `@Prompt` 함수를 사용합니다.

휴양지 서비스 분야를 사용하여 쿼리를 실행하면 휴양지 이름을 입력하라는 프롬프트가 표시됩니다. 휴양지의 WHERE 절을 수정하면 변경 사항이 휴양지 서비스 분야 개체에 자동으로 적용됩니다.

7.4 외부 전략을 사용하여 유니버스 만들기 사용자 지정

유니버스 디자인 도구에서는 기본적으로 제공되는 자동 루틴을 사용하여 데이터베이스 구조를 기반으로 유니버스 구성 요소를 자동으로 만듭니다. 전략이라고 하는 이러한 루틴은 매개 변수 대화 상자의 전략 페이지(파일 > 매개 변수 > 전략)에서 사용할 수 있습니다. 이러한 전략은 유니버스 디자인 도구에서 기본적으로 제공됩니다. 이들 전략은 액세스하거나 수정할 수 없습니다. 전략 사용 및 활성화는 전략 선택 [페이지 81] 단원에서 설명합니다.

정의된 출력 구조를 따르는 SQL 스크립트를 직접 만들어 유니버스를 자동으로 만드는 작업을 사용자 지정할 수도 있습니다. 이러한 스크립트는 전략 페이지에서 다른 전략과 마찬가지로 선택할 수 있습니다. 사용자 정의 및 사용자 지정 스크립트를 외부 전략이라고 합니다.

이 단원에서는 외부 전략 및 사용 방법에 대해 설명합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.4.1 유니버스 디자인 도구로 외부 전략 마이그레이션

Universe Designer 6.5 이전 버전의 유니버스 디자인 도구에서 외부 전략은 st<xxxx>.txt 파일이라는 외부 텍스트 파일에 정의되었습니다. 이 파일은 더 이상 지원되지 않습니다.

> **노트**
>
> Universe Designer 6.5 에서 마이그레이션하는 경우 외부 전략은 유니버스 디자인 도구에서와 동일한 방식으로 처리됩니다.

이전 버전에서 사용했던 사용자 지정 및 사용자 정의 외부 전략을 유니버스 디자인 도구에서 사용하려면 다음 작업을 수행해야 합니다.

- 새 외부 전략 파일(<RDBMS>.STG)을 다음과 같이 편집합니다.
  - 대상 RDBMS 의 외부 전략 파일을 XML 편집기에서 엽니다.
  - 각 전략에 대한 새 항목을 만듭니다.
  - 각 전략에 대해 SQL 태그를 사용하여 SQL 스크립트를 직접 STG 파일에 복사합니다.

  또는
  - FILE 태그를 사용하여 외부 텍스트 파일의 데이터를 참조하는 파일 경로를 입력합니다.

  이 두 가지 방법은 외부 전략 만들기 [페이지 375] 단원에서 자세히 설명합니다.
- 도움말 텍스트를 두 번째 XML 파일(<RDBMS><language>.STG)에 복사합니다. 자세한 내용은 외부 전략에 대한 도움말 항목 만들기 [페이지 368] 단원에서 설명합니다.
- 외부 전략 파일이 이전 버전의 Universe Designer 에서와 마찬가지로 매개 변수 파일(PRM)이 아닌 일반 매개 변수 파일(SBO)에서 선언되었는지 확인합니다. 자세한 내용은 외부 전략 파일이 선언되었는지 확인 [페이지 369] 단원에서 설명합니다.

7.4.2 외부 전략 개요

다음 표는 외부 전략 생성 및 관리에 사용되는 파일과 해당 파일의 역할을 간략히 보여 줍니다.

표 161:

외부 전략 관리 프로세스에서의 역할 및 파일	설명
외부 전략이 외부 전략 파일(<RDBMS>.STG)에 저장되고 만들어집니다.	XML 파일에는 외부 전략 이름, 유형, SQL 스크립트 또는 데이터를 포함하는 외부 텍스트 파일에 대한 파일 참조가 포함됩니다. 파일은 다음 위치에 저장됩니다. $INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS>.stg. 각 RDBMS 에 대해 하나의 파일이 존재합니다. $INSTALLDIR/dataAccess/RDBMS/connectionServer/strategy.dtd 에 있는 strategy.dtd 파일을 사용합니다. 관련 단원: • 전략 파일(STG)의 구조 [페이지 370] • 외부 전략 만들기 [페이지 375]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

외부 전략 관리 프로세스에서의 역할 및 파일	설명
외부 전략에 대한 도움말 텍스트가 외부 전략 언어 파일 (<RDBMS><language>.STG)	XML 파일에는 외부 전략 파일의 각 외부 전략에 대한 도움말 텍스트가 포함됩니다. 전략 페이지에서 선택될 때 외부 전략 아래 나타나는 텍스트입니다. 파일은 다음 위치에 저장됩니다. $INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS><language>.stg. 다음 위치에 있는 strategy_localization.dtd 파일을 사용합니다.$INSTALLDIR/dataAccess/RDBMS/connectionServer/strategy_localization.dtd 관련 단원: 외부 전략에 대한 도움말 항목 만들기 [페이지 368]
외부 전략 파일이 대상 RDBMS 의 일반 데이터 액세스 파일(SBO)에서 선언됩니다.	XML 파일에는 대상 RDBMS 에 대한 일반 데이터 액세스 매개 변수가 포함됩니다. 기본적으로 외부 전략 파일 이름은 외부 전략 매개 변수의 값으로 설정됩니다. 관련 단원: 외부 전략 파일이 선언되었는지 확인 [페이지 369]

7.4.3 외부 전략이란?

외부 전략은 .UNV 파일에 외부에서 저장되는 SQL 스크립트로, 유니버스 디자인 도구가 유니버스에서 개체나 조인 생성 및 테이블 검색 작업을 자동화하는 데 사용할 수 있도록 구조화됩니다. 외부 전략은 STG 확장자를 가진 외부 전략 파일로 저장되며, 외부 전략 파일은 XML 형식입니다. 지원되는 각 RDBMS 에 대해 파일이 하나씩 존재합니다.

외부 전략 파일은 다음 디렉터리에 저장됩니다.

표 162:

$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<rdbms>.stg

노트

외부 전략을 편집할 때는 XML 편집기를 사용해야 합니다.

7.4.3.1 외부 전략에 액세스

외부 전략은 전략 페이지의 드롭다운 목록 상자에 기본 제공 전략과 함께 표시됩니다. 이 페이지에는 XML 파일의 전략 유형 범주에 해당하는 드롭다운 목록 상자가 하나씩 있습니다. 목록에서 외부 전략은 다음과 같이 전략 이름 앞에 외부 전략이 접두사로 사용되어 표시됩니다.

표 163:

외부 전략: <전략 이름>

예를 들어, 전략 파일에서 조인을 만드는 데 사용되는 외부 전략 이름이 제약 조건인 경우 이 외부 전략은 전략 페이지의 조인 드롭다운 목록에 '외부 전략: 제약 조건'으로 표시됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.4.4 외부 전략에 대한 도움말 텍스트 만들기

전략 페이지에서는 선택한 각 전략 아래에 설명 노트가 표시됩니다. 이것이 전략에 대한 도움말 텍스트입니다. 기본 제공 전략의 도움말 텍스트는 액세스하거나 편집할 수 없습니다. 그러나 외부 전략의 도움말 텍스트는 액세스하고 편집할 수 있습니다.

> **노트**
>
> 이전 버전의 유니버스 디자인 도구에서는 [도움말] 섹션에 있는 전략 텍스트 파일에 도움말 텍스트가 포함되었습니다. 이 섹션에 있던 텍스트는 현재 버전에서는 아래에서 설명할 외부 전략 언어 파일에 별도로 저장됩니다.

7.4.4.1 별도의 파일에 외부 전략 도움말 텍스트 저장

외부 전략에 대한 도움말 텍스트는 <RDBMS><language>.stg 라는 별도의 외부 전략 언어 파일에 저장됩니다. 예를 들어, oaracleen.stg 는 oracle.stg 파일의 전략에 대한 도움말 텍스트 파일입니다.

도움말 텍스트 항목을 편집하고 사용자 지정할 수 있습니다. 도움말 텍스트는 전략에 익숙하지 않은 디자이너를 위해 해당 전략의 작업을 간략하게 설명해야 합니다.

외부 전략 파일에 있는 외부 전략 각각에 해당되는 도움말 텍스트 항목이 외부 전략 언어 파일에 있어야 합니다.

설치된 유니버스 디자인 도구의 언어 버전마다 전략 언어 파일이 하나씩 있습니다. 외부 전략 언어 파일은 외부 전략 파일과 같은 디렉터리에 있습니다. 예를 들어, 유니버스 디자인 도구의 프랑스어 버전을 설치한 경우 Oracle 용 외부 전략 언어 파일은 oraclefr.stg 입니다. 영어 버전은 oracleen.stg 입니다.

외부 전략 파일에 외부 전략을 새로 만들면 해당하는 도움말 텍스트를 외부 전략 언어 파일에도 만들어야 합니다. 이러한 도움말 텍스트는 유니버스를 사용하는 디자이너에게 외부 전략에 대한 정보를 제공합니다.

> **예**
>
> **Oracle 데이터 액세스 드라이버와 함께 제공된 전략에 대한 도움말 텍스트 항목**
>
> 다음은 oracleen.stg 파일에 있는 클래스 및 개체 전략에 대한 도움말 텍스트를 보여 줍니다. 이 도움말 텍스트는 oracle.stg 파일에 정의되어 있는 클래스 및 개체 외부 전략에 대한 도움말 텍스트입니다.

```
<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 외부 전략에 대한 도움말 항목 만들기

외부 전략에 대한 도움말 항목을 만들려면

1. 대상 RDBMS 의 외부 전략 언어 파일을 XML 편집기에서 엽니다. 대상 RDBMS 의 외부 전략 언어 파일 위치는 다음과 같습니다.
   `$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS><language>.stg.`

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

예를 들면 다음과 같습니다.
`$INSTALLDIR/dataAccess/RDBMS/connectionServer/oracle/oracleen.stg`.

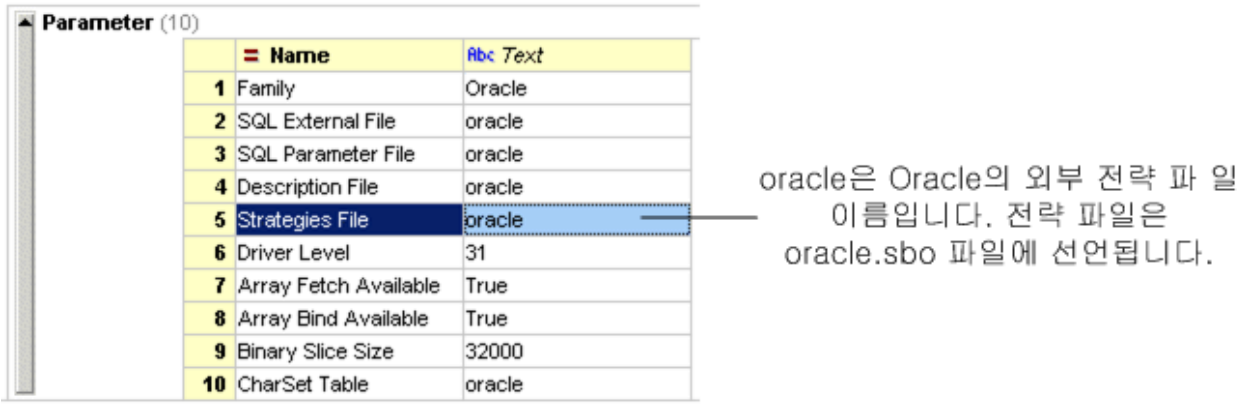
2. Name 요소를 새로 만듭니다.
3. 전략 이름을 입력합니다. 도움말 텍스트를 만들 대상 전략입니다.
4. Help 라는 Message ID 를 만듭니다. 이 태그에는 도움말 텍스트가 들어 갑니다.
5. 도움말 텍스트를 입력합니다.
6. Name 이라는 Message ID 를 만듭니다. 이 태그에는 외부 전략을 선택했을 때 전략 드롭다운 목록에 표시할 이름이 들어 갑니다.
7. 전략 이름을 입력합니다.
   파일의 유효성을 검사한 후 저장하고 닫습니다.
   다음 번에 유니버스 디자인 도구를 시작하면 선택한 외부 전략 아래에 도움말 텍스트가 나타납니다.

팁

기존 Name 요소를 복사한 후 새 전략에 대한 값을 새로 입력하면 새 Name 요소의 매개 변수를 손쉽게 만들고 설정할 수 있습니다.

7.4.5 외부 전략 파일이 선언되었는지 확인

외부 전략 파일은 대상 RDBMS 의 일반 매개 변수 파일(SBO) 내의 Parameter 섹션에 선언됩니다. 예를 들어, Oracle 의 외부 전략 파일은 oracle.stg 이며 다음과 같이 oracle.sbo 파일에 oracle 값이 있습니다.



	Name	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

7.4.5.1 전략 파일이 SBO 파일에 선언되었는지 확인

외부 전략 파일이 제대로 선언되었는지 확인하려면

1. 대상 RDBMS 의 SBO 파일을 엽니다.
2. Strategies Name 매개 변수에 외부 전략 파일의 이름이 설정되었는지 확인합니다. 이 값은 기본 설정입니다.
3. 이름이 제대로 설정되어 있지 않으면 외부 전략 파일 이름을 정확하게 입력합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4. 수정 사항이 있으면 파일을 저장한 후 닫습니다.
   또는
5. 수정 사항이 없으면 파일을 저장하지 않고 닫습니다.

> **노트**
>
> 이전 버전의 유니버스 디자인 도구에서는 외부 전략이 PRM 파일에서 선언되었습니다. Universe Designer 6.5 에서는 더 이상 사용되지 않습니다. SBO 파일의 전략 파일 매개 변수는 기본적으로 대상 RDBMS 에 대한 외부 전략 파일 이름으로 설정되어 있습니다. 외부 전략을 Universe Designer 6.5 로 마이그레이션하는 방법은 외부 전략에 대한 도움말 텍스트 만들기 [페이지 368] 단원을 참조하십시오.

7.4.6 외부 전략 예제 사용

모든 외부 전략 파일에는 Business Objects 제품과 함께 제공되는 기존 전략 여러 개가 들어 있습니다. 예를 들어, 파일에 개체 전략, 조인 전략 및 테이블 브라우저 전략이 하나씩 포함되거나 각 유형의 전략이 여러 개씩 포함될 수 있습니다.

예제 파일을 사용자 지정하거나 예제 파일에 기반하여 새 외부 전략을 만들 수 있습니다. 기존 전략을 사용자 지정하거나 전략을 새로 만들 수 있습니다.

파일을 수정할 때는 해당 파일의 복사본을 저장하는 것이 좋습니다.

7.4.7 전략 파일(STG)의 구조

지원되는 RDBMS 각각에 대해 XML 형식의 외부 전략 파일(STG)이 있습니다. 이 파일에 기존 외부 전략을 마이그레이션하거나 외부 전략을 새로 만듭니다. 모든 외부 전략 파일은 다음의 디렉터리에 있는 전략 dtd 파일(<RDBMS>.dtd)을 사용합니다.

표 164:

$INSTALLDIR/dataAccess/RDBMS/connectionServer

외부 전략 XML 파일의 요소는 외부 전략 DTD 파일에 정의됩니다. XML SPY 같은 XML 편집기를 사용하면 새 전략 요소를 만들 때 사용할 수 있는 매개 변수가 드롭다운 목록에 나열됩니다.

외부 전략 파일에는 Strategies 라는 주 섹션이 하나 있습니다. 모든 외부 전략은 이 섹션 안에 정의됩니다. Strategies 섹션 안에는 다음과 같은 요소와 매개 변수가 있습니다.

표 165:

파일 요소	설명
전략	주 요소입니다. 이 요소 안에 모든 외부 전략이 만들어집니다.
이름	외부 전략 이름입니다. 이 이름은 전략 페이지의 드롭다운 목록에 표시됩니다. 기본 요소입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

파일 요소	설명
형식	전략 페이지에서 외부 전략이 표시되는 목록입니다. 다음의 세 가지 값 중 하나를 사용할 수 있습니다. • JOIN: 조인 전략은 조인 목록에 표시됩니다. • OBJECT: 클래스 및 개체 전략은 클래스 및 개체 목록에 표시됩니다. • STRUCT: 테이블 검색 전략은 테이블 목록에 표시됩니다.
SQL	스크립트의 SQL 코드입니다. 전략이 선택될 때 유니버스 디자인 도구가 실행하는 SQL 스크립트입니다. SQL 스크립트가 특정 출력 형식을 따라야만 개체와 조인 만들기 루틴 및 테이블 검색 루틴이 제대로 실행될 수 있습니다. 외부 전략의 SQL 작성에 대한 내용은 개체 전략(OBJECT)의 출력 형식 [페이지 372] 단원을 참조하십시오.
연결	데이터베이스 연결을 지정합니다. 개인 연결 유형을 사용해야 합니다.
SkipMeasures	값을 Y 로 설정하면 빠른 디자인 마법사에서 계수를 만드는 화면을 건너뜁니다.
파일	유니버스를 자동으로 만들기 위해 특정 출력 형식으로 구성된 데이터가 포함된 외부 텍스트 파일의 경로입니다. 자세한 내용은 데이터 텍스트 파일 만들기 [페이지 376] 단원을 참조하십시오.

예

oracle.stg 의 클래스 및 개체 외부 전략

Oracle 용 외부 전략 파일은 oracle.stg 이며 $INSTALLDIR/dataAccess/RDBMS/connectionServer/oracle/oracle.stg 디렉터리에 저장됩니다. 이 파일에는 유니버스 디자인 도구와 함께 제공된 다양한 예제 외부 전략이 포함되어 있습니다. 이러한 예제 전략을 사용자 지정하거나 새 전략을 만들 때 템플릿으로 사용할 수 있습니다.

다음은 oracle.stg 에 있는 외부 전략으로 자동으로 테이블과 클래스를 연결하고 열과 개체를 연결합니다.

```
<Strategy Name="Classes_and_Objects">
        <Type>OBJECT</Type>
        <SQL>SELECT
       U1.table_name,'|',
       U1.column_name,'|',
       translate(initcap(U1.table_name),'_',' '),'|',
       translate(initcap(U1.column_name),'_',' '),'|',
       U1.table_name||'.'||U1.column_name,'|',
       '  ','|',
decode(SUBSTR(U1.DATA_TYPE,1,1),'N','N','F','N','D','D','C'),'|',
       SUBSTR(U2.comments,1,474),'|',
       'O','|'
FROM USER_TAB_COLUMNS U1,USER_COL_COMMENTS U2
WHERE
    U1.table_name=U2.table_name
and U1.column_name=U2.column_name
UNION
SELECT
       S.SYNONYM_NAME,'|',
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

```
        U1.column_name,'|',
        translate(initcap(S.SYNONYM_NAME),'_',' '),'|',
        translate(initcap(U1.column_name),'_',' '),'|',
        S.SYNONYM_NAME||'.'||U1.column_name,'|',
        '  ','|',
decode(SUBSTR(U1.DATA_TYPE,1,1),'N','N','F','N','D','D','C'),'|',
        SUBSTR(U2.comments,1,474),'|',
        'O','|'
FROM ALL_TAB_COLUMNS U1, ALL_COL_COMMENTS U2, ALL_OBJECTS O, USER_SYNONYMS S
WHERE
      S.table_owner=O.owner
AND   S.table_name=O.object_name
AND   (O.OBJECT_TYPE='TABLE' OR O.OBJECT_TYPE='VIEW')
AND   O.owner=U1.owner
AND   O.object_name=U1.table_name
AND   U1.owner=U2.owner
AND   U1.table_name=U2.table_name
AND   U1.column_name=U2.column_name</SQL>
    </Strategy>
```

7.4.8 전략 출력 형식

외부 전략 파일에서 <SQL> 태그 안에 SQL 스크립트를 작성하거나 복사합니다. SQL 스크립트가 반환하는 정보의 순서와 유형은 만들고 있는 전략이 개체인지, 조인인지 아니면 테이블 전략인지에 따라 다릅니다. 유니버스 디자인 도구에서는 전략 유형마다 다른 정보가 필요합니다.

전략에 대해 SQL 스크립트를 만들 때는 스크립트의 출력 내용이 아래에 설명된 출력 형식과 일치해야 합니다.

스크립트는 여러 개의 열 형식으로 출력됩니다. 각 열은 개체, 조인 또는 테이블 구성 요소를 만들기 위해 생성된 정보에 해당합니다.

이 단원에서는 다음과 같은 전략의 출력 형식을 설명합니다.

- 개체 전략
- 조인 전략
- 테이블 브라우저 전략

7.4.8.1 개체 전략(OBJECT)의 출력 형식

개체 전략의 출력 형식에는 9 개의 열이 포함됩니다. null 값이 포함된 열이 있더라도 전략 출력에는 모든 열이 포함되어야 합니다. 반환되는 모든 값은 파이프 '|' 문자로 분리해야 합니다. 반환되는 값 끝에 파이프가 추가되어야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 166:

열 번호	열 내용	설명
1	테이블	테이블 이름 형식은 [한정자.][소유자.]테이블이고, 각 이름은 35 자로 제한됩니다. 이 열을 비워 두면 Select(다섯째 열)와 Where(여섯째 열)에서 테이블을 가져옵니다.
2	열 이름	열 이름입니다.
3	클래스 이름	클래스 이름입니다. 하위 클래스는 클래스\하위 클래스 형식으로 작성됩니다.
4	개체 이름	개체 또는 조건 이름입니다. 개체 이름이 비어 있으면 클래스와 해당 설명이 만들어집니다.
5	Select	Select 문입니다.
6	Where:	Select 열을 비워 두고 Where 절만 사용하면 미리 정의된 조건과 해당 설명이 만들어집니다.
7	형식	C(문자), N(숫자), D(날짜), T(긴 텍스트) 형식 중 하나입니다. 열을 비워 두면 N 이 기본값으로 사용됩니다.
8	설명	개체에 대한 설명입니다.
9	자격	D(차원), M(계수) 또는 I(설명) 중 하나입니다. 열을 비워 두면 D 가 기본값으로 사용됩니다.

예

열 주석을 개체 설명에 복사하는 외부 개체 전략

아래 예제에서는 Where 절을 사용하지 않습니다. 따라서 Where 절의 출력 열이 비어 있습니다.

<Strategies>

<Strategy Name="Read Column descriptions">

<Type>OBJECT</Type>

<SQL>Select

표 167:

	열	설명
Table_name, '|',	1	테이블 이름

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

Column_name, '|',	2	열 이름
Replace (Table_name,'_',' '), '|',	3	테이블 이름에 사용된 밑줄을 클래스 이름에서 공백으로 바꿉니다.
Replace (Column_name, '_', ' '),'|',	4	열 이름에 사용된 밑줄을 개체 이름에서 공백으로 바꿉니다.
Table_name||'.'||Column_name, '|',	5	테이블 이름과 열 이름을 마침표로 구분하여 연결합니다. Select 문입니다.
,'|',	6	Where 절이 없습니다.
Column_type,'|',	7	시스템 테이블에서 열 형식을 가져옵니다.
Column_Desc,'|',	8	시스템 테이블에서 열 설명을 가져옵니다.
' ','|'	9	개체 유형이 null 이므로 차원이 기본값으로 사용됩니다.

</SQL>

7.4.8.2 조인 전략(JOIN)의 출력 형식

조인 전략의 출력 형식에는 다음과 같은 열이 포함됩니다.

표 168:

열 번호	열 내용	설명
1	테이블 1	조인의 첫 번째 테이블 이름입니다.
2	테이블 2	조인의 두 번째 테이블 이름입니다.
3	조인 정의	table1.column1=table2.column2 형식의 실제 조인 정의입니다.
4	외부 형식	외부 조인 형식입니다. L= 왼쪽 외부, R= 오른쪽 외부입니다. 이 열이 비어 있으면 외부 조인이 없는 것입니다.
5	카디널리티(선택)	11, 1N 및 N1 값을 사용할 수 있습니다.

7.4.8.3 테이블 브라우저 전략(STRUCT)의 출력 형식

테이블 브라우저 전략의 출력 형식에는 다음과 같은 열이 포함됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 169:

열 번호	열 내용	설명
1	한정자	RDBMS 마다 다릅니다. 테이블 한정자는 데이터베이스 이름이나 기타 다른 ID 입니다.
2	소유자	RDBMS 마다 다릅니다.
3	테이블	테이블, 뷰 또는 동의어의 이름입니다.
4	열	열 이름입니다.
5	데이터 형식	C(문자), N(숫자), D(날짜), T(긴 텍스트) 형식 중 하나입니다. 열을 비워 두면 C 가 기본값으로 사용됩니다.
6	Null 허용 Y(예) 또는 N(아니요)	열에 null 값이 포함될 수 있는지 여부를 나타냅니다.

7.4.9 외부 전략 만들기

두 가지 방법으로 외부 전략을 만들 수 있습니다.

표 170:

외부 전략을 만드는 방법	XML 파일 태그	설명
SQL 스크립트를 직접 삽입	SQL	SQL 태그를 사용하여 전략 SQL 스크립트를 외부 전략 파일에 직접 삽입합니다.
외부 파일의 데이터 참조	FILE	전략에 사용할 데이터가 들어 있는 외부 텍스트 파일의 경로와 이름을 입력합니다.

두 방법은 아래에서 설명합니다.

7.4.9.1 외부 전략 만들기

외부 전략을 직접 만들려면

1. 대상 RDBMS 의 외부 전략 파일을 XML 편집기에서 엽니다. 대상 RDBMS 의 전략 파일 위치는 다음과 같습니다.
   $INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS>.stg
2. Strategy 요소를 새로 만듭니다.
   새 전략입니다. XML Spy 같은 XML 편집기를 사용하면 전략의 Name, Type 및 SQL 요소가 자동으로 만들어집니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

3. 전략 이름을 입력합니다.
   전략 이름은 유니버스 매개 변수 대화 상자의 전략 탭 및 빠른 디자인 마법사에 표시됩니다.
4. OBJECT, JOIN 또는 STRUCT 중에서 TYPE 매개 변수를 입력합니다.
   예를 들어, TYPE=OBJECT 같이 입력합니다.
5. 전략의 SQL 문을 입력합니다. SQL 형식은 전략 출력 형식 [페이지 372] 단원에서 설명합니다.
   또는
   데이터가 들어 있는 텍스트 파일을 참조하려면 SQL 요소 대신 File 요소를 사용합니다. 데이터 파일의 경로를 입력합니다(예: C:\Path\Filename.txt).
6. 선택적인 요소를 추가하고 필요한 경우 값을 입력합니다.
7. XML 파일의 유효성을 검사한 다음 파일을 저장하고 닫습니다.
8. 외부 전략 파일이 대상 RDBMS 의 일반 데이터 액세스 파일(<RDBMS>.SBO)에 선언되었는지 확인합니다. 다음과 같이 하십시오.
   - 일반 데이터 액세스 파일(SBO)을 엽니다.

   이 파일은 $INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/ 디렉터리에 있습니다.
   - Strategies File 요소에 외부 전략 파일의 이름이 설정되었는지 확인합니다. 이것이 기본값입니다.
   - SBO 파일을 수정한 경우 파일을 저장한 후 닫습니다.

   매개 변수 대화 상자의 전략 페이지에 있는 조인, 개체 또는 테이블 드롭다운 목록에 외부 전략이 표시됩니다. 새로 만든 외부 전략이 표시되도록 하려면 유니버스 디자인 도구를 닫은 후 다시 시작해야 합니다.

> **노트**
>
> 전략 페이지에서 외부 전략을 선택했을 때 표시되는 도움말 텍스트를 추가하려면 외부 <RDBMS><language>.STG 라는 별도의 파일에 텍스트를 추가해야 합니다. 이 파일은 외부 전략 파일과 같은 디렉터리에 있습니다. 외부 전략에 대한 도움말 텍스트 추가는 외부 전략에 대한 도움말 텍스트 만들기 [페이지 368] 단원에서 설명합니다.

7.4.10 데이터 텍스트 파일 만들기

외부 전략에 사용할 데이터가 포함된 텍스트 파일을 만들 수 있습니다. 외부 전략을 만들 때 SQL 을 직접 입력하는 대신 텍스트 파일의 경로와 이름을 입력할 수 있습니다. 이 경우 외부 전략 파일에 FILE 요소를 삽입하고 여기에 파일 경로와 이름을 설정해야 합니다.

SQL 스크립트의 출력은 개체, 조인 또는 테이블 중 사용하는 전략 유형에 맞는 형식을 따라야 합니다. 출력 형식은 전략 출력 형식 [페이지 372] 단원에서 설명합니다.

모든 형식은 정보가 포함된 열로 구성되며 표 형식으로 구분됩니다.

7.4.11 유니버스 디자인 도구에서 외부 전략 적용

다음과 같이 외부 전략을 적용할 수 있습니다.

1. 사용할 외부 전략을 매개 변수 대화 상자의 전략 페이지에서 선택합니다.
   예를 들면 다음과 같습니다.
   - 개체 전략을 사용하여 추출한 개체를 삽입하려면 삽입 메뉴에서 후보 개체 명령을 선택합니다.
   - 조인 전략에서 파생된 조인을 삽입하려면 도구 메뉴에서 조인 검색 명령을 선택합니다.

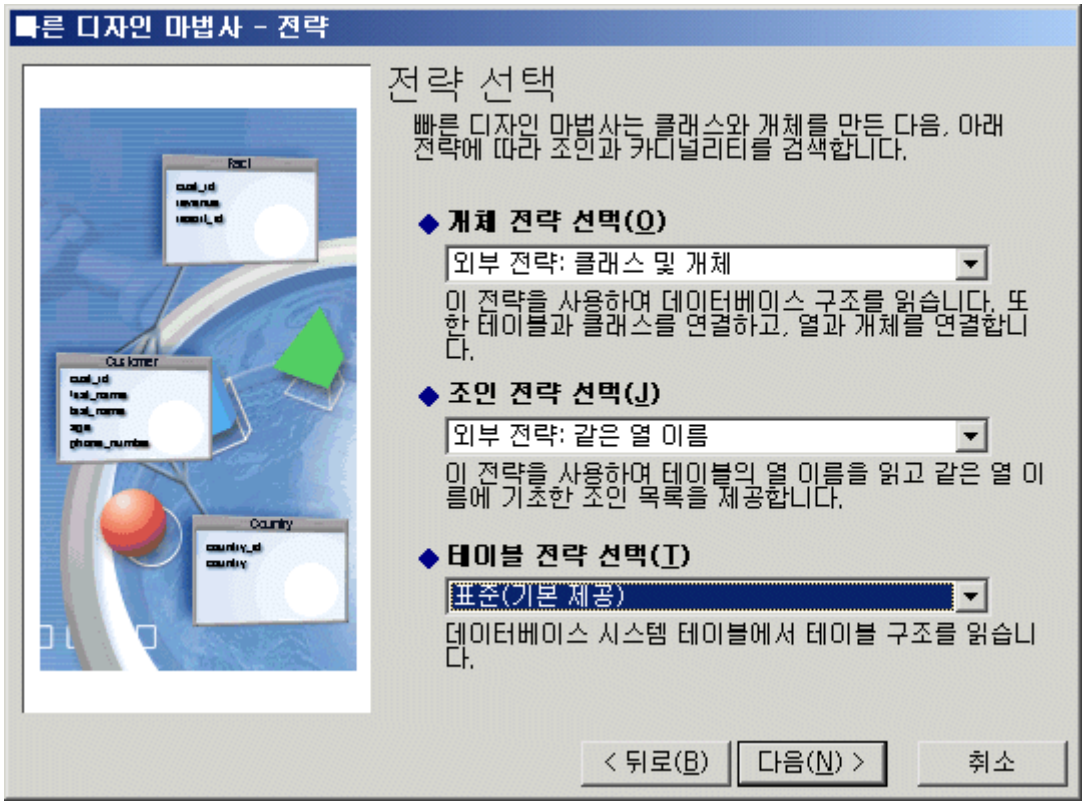
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 테이블 브라우저 전략을 사용하여 추출한 테이블을 삽입하려면 삽입 메뉴에서 테이블 명령을 선택합니다.

> **노트**
>
> 조인 전략을 선택하면 유니버스 디자인 도구에서는 해당 전략을 사용하여 후보 조인 및 카디널리티를 검색합니다. 제안된 조인이나 카디널리티를 선택적으로 적용할 수 있습니다. 선택한 전략에 기반하여 후보 조인과 카디널리티를 자동으로 적용하려면 옵션 대화 상자의 데이터베이스 페이지(도구 > 옵션 > 데이터베이스)에서 적절한 만들기 옵션을 선택해야 합니다. 자세한 내용은 전략의 자동 작성 기능 사용 [페이지 83] 단원을 참조하십시오.

7.4.11.1 빠른 디자인 마법사에서 전략 선택

설정한 외부 전략을 빠른 디자인 마법사에서 선택할 수도 있습니다. 그러려면 마법사 시작 창에서 "전략을 선택하려면 여기를 클릭하십시오." 옵션을 클릭해야 합니다.



7.5 분석 함수 사용

유니버스 디자인 도구는 특정 RDBMS 에 대해 분석 함수를 사용할 수 있도록 지원합니다. 분석 함수는 RedBrick 에서 RISQL 함수, Teradata 에서 OLAP 함수로 불립니다. 유니버스 디자인 도구를 사용하여 유니버스에서 개체에 대한 분석 함수를 정의할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

분석 함수를 사용하면 Web Intelligence 사용자가 InfoView 의 일반적인 보고서 작성 기능에 포함되지 않은 데이터 분석을 수행할 수도 있습니다. 이 기능 및 Web Intelligence 에 사용된 다른 기능에 대한 자세한 내용은 *Using Functions, Formulas and Calculations in Web Intelligence* 문서의 *Calculating values with Smart Measures* 단원을 참조하십시오.

이 단원에서는 다음 RDBMS 에 대해 유니버스의 개체에 대해 Analytic, RISQL 및 OLAP 함수를 정의할 수 있는 방법을 설명합니다.

- IBM DB2 UDB 및 Oracle [페이지 379]
- RedBrick(RISQL 함수) [페이지 383]
- Teradata(OLAP 함수) [페이지 385]

7.5.1 분석 함수 소개

분석 함수는 순서화된 행 그룹 또는 분할 영역으로 나눌 수 있는 결과 집합에 대해 분석 작업을 수행하는 함수입니다.

유니버스 디자인 도구에서는 분석 함수와 함께 개체를 정의하여 하나 이상의 파티션 내에서 순위, 누적 집계 및 비율을 계산할 수 있습니다. RDBMS 에 따라 분할 영역 내에서 분석을 적용할 행 범위를 정의할 수도 있습니다.

분석 함수에 대한 자세한 내용은 해당 RDBMS 설명서를 참조하십시오.

7.5.2 분석 함수 사용 이점

유니버스 디자인 도구의 분석 함수를 사용하여 개체를 정의하는 Web Intelligence 사용자는 다음과 같은 이점을 얻을 수 있습니다.

- 작업량이 줄어듭니다. 분석 함수로 정의된 개체는 보고서 수준에서 일반적으로 확장 구문을 사용해야 하는 데이터 분석을 수행할 수 있습니다.
- 추기 기능을 활용할 수 있습니다. InfoView 에서는 일반적으로 이동 평균 계산 및 고급 집계 처리 적용과 같은 몇 가지 데이터 분석 작업이 불가능합니다. 이제 분석 함수를 사용하는 개체를 통해 Web Intelligence 사용자는 이전에 불가능했던 고급 데이터 분석을 수행할 수 있습니다.
- 쿼리 성능 향상. 서버에서 계산이 수행됩니다.

7.5.3 지원되는 분석 함수 계열

분석 함수를 정의할 수 있는 함수 계열은 다음과 같습니다.

- 순위
- 누적 집계
- 비율, 보고서에 대한 비율 또는 보고 집계

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.5.4 유니버스 디자인 도구에서 분석 함수를 사용하는 방법

분석 함수는 개체에 대한 SELECT 문에서 분석 함수를 정의하여 사용합니다.

각 매개 변수(PRM) 파일의 RDBMS 섹션에는 SELECT 문에서 사용할 수 있는 분석 함수가 나열되어 있습니다. 이 목록에는 분석 함수에 대해 지원되는 각 RDBMS 에서 사용할 수 있는 모든 계열의 함수가 포함되어 있지 않을 수 있습니다.

7.5.4.1 PRM 파일 소개

PRM 파일은 Web Intelligence 제품에서 유니버스 작성 및 SQL 쿼리 생성을 구성하는 데 사용되는 매개 변수 파일입니다. 지원되는 각 RDBMS 에 대해 하나의 PRM 파일이 존재합니다. PRM 파일은 다음 폴더에 있습니다.

<INSTALLDIR>\dataAccess\RDBMS\connectionServer\<rdbms>\

매개 변수 파일 수정에 대한 자세한 내용은 데이터 액세스 가이드를 참조하십시오.

분석 함수를 사용하기 전에 PRM 파일에 나열되어 있는지 확인해야 합니다. 나열되지 않을 경우 목록에 함수 이름을 추가할 수 있습니다. 그러면 유니버스 디자인 도구에서는 개체에 대한 Select 문에 해당 함수를 사용할 수 있도록 지원합니다. 자세한 내용은 PRM 파일 확인 및 분석 함수 지원 기능 추가 [페이지 381] 단원을 참조하십시오.

7.5.4.2 각 RDBMS 에 대해 분석 함수 사용

각 RDBMS 에서 분석 함수 사용에 대해 다음 내용을 설명합니다.

- Select 문에서 분석, RISQL 및 OLAP 함수에 대해 사용 가능한 구문
- PRM 파일을 확인하고 나열되지 않은 분석 함수를 지원하도록 수정하는 방법
- 분석 함수 사용을 위한 RDBMS 특정 규칙 및 제한 사항
- Select 문을 편집할 때 자동으로 분석 함수 구문 삽입

7.5.5 IBM DB2 UDB 및 Oracle

이 두 RDBMS 에 대해 동일한 분석 함수 구문을 사용할 수 있습니다.

7.5.5.1 DB2, UDB 및 Oracle RDBMS 에 대한 SELECT 문 정의

개체에 대한 Select 문에서 분석 함수를 정의합니다. Select 문에 대한 편집 상자 중 하나에 구문을 입력해야 합니다.

> **노트**
>
> Select 문 편집 대화 상자의 함수 목록에 분석 함수를 추가하면 구문 입력을 자동화할 수 있습니다. 함수 목록에 함수가 제공되도록 하려면 .prm 파일의 [FUNCTIONS] 섹션에 분석 함수를 추가해야 합니다. 자세한 내용은 Select 문에 자동으로 구문 삽입 [페이지 387] 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

분석 함수는 키워드 OVER 로 식별됩니다. 예를 들면 다음과 같습니다.

표 171:

RANK() OVER (PARTITION BY calender.cal_year ORDER BY SUM(telco_facts.total_billed_rev)DESC)

OVER 키워드 뒤에 오는 절은 분할 영역을 정의하고, 결과 테이블에서 행이 정렬되는 방식을 정의합니다.

각 분석 함수 계열의 구문에 대한 설명은 다음과 같습니다.

표 172:

함수 계열	구문	설명
순위	`RANK() OVER(PARTITION BY arg1 ORDER BY arg2 ASC/DESC)`	• arg1 은 선택 사항입니다. 인수가 포함되지 않을 경우 분할 영역은 기본적으로 전체 결과 집합입니다. • arg2 는 필수입니다. 순위는 인수 값을 기준으로 합니다. • ASC/DESC 는 값이 오름차순 또는 내림차순으로 정렬되는지 여부를 결정합니다. ASC 가 기본값입니다.
창 집계	`SUM(arg1) OVER(PARTITION BY arg2 ORDER BY arg3)`	• arg1 은 누적 집계가 기준으로 하는 인수입니다. • arg2 는 재설정 절입니다. 선택 사항입니다. • arg3 은 그룹 절입니다. 선택 사항입니다.
보고 집계	`RATIO_TO_REPORT(arg1) OVER(PARTITION BY arg2)`	• arg1 은 비율의 기준 인수입니다. • arg2 는 재설정 절입니다. 선택 사항입니다.

Window 절 사용

창 집계의 경우 arg3 다음에 창 크기에 대한 범위를 정의하는 <window 절>도 정의할 수 있습니다. 예를 들면 다음과 같습니다.

표 173:

```
<window frame units> ::= ROW |RANGE <window frame start>::= UNBOUNDED PRECEDING |
<window frame preceding> |CURRENT ROW <window frame between>
```

BETWEEN 절 구문 및 기타 창 크기 정의에 대한 자세한 내용은 해당 RDBMS 설명서를 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.5.5.2 PRM 파일 확인 및 분석 함수 지원 기능 추가

IBM DB2 UDB 및 Oracle 용 PRM 파일은 분석 함수 사용을 지원하도록 업데이트되었습니다.

하지만 PRM 파일에 대상 RDBMS 에서 사용할 수 있는 모든 분석 함수가 포함되어 있지 않을 수 있습니다. 분석 함수를 사용하기 전에 분석 함수가 PRM 파일의 RDBMS 섹션에 나열되어 있는지 확인하고, 필요한 경우 목록에 추가하십시오.

다음과 같이 할 수 있습니다.

Oracle 또는 IBM DB2 PRM 파일에 분석 함수 지원 기능을 추가하려면 다음과 같이 하십시오.

1. Business Objects 경로의 Data Access 디렉터리로 이동합니다.
2. RDBMS 의 PRM 파일을 텍스트 편집기에서 엽니다.
3. PRM 파일에서 RDBMS 섹션으로 스크롤합니다.
4. 다음 매개 변수 및 값이 존재하는지 확인합니다.

   표 174:

PRM 의 매개 변수 및 값	설명
OVER_CLAUSE = Y	적절한 SQL(OVER_CLAUSE)을 생성합니다.
RISQL_FUNCTIONS = <사용된 함수 목록>	사용 가능한 분석 함수입니다.

5. 나열되지 않은 분석 함수를 사용하려는 경우 목록 끝에 함수의 이름을 입력합니다. 예를 들어 RATIO_TO_REPORT 를 사용하려면 다음과 같이 목록에 추가합니다.

```
[RDBMS]
(GENERAL)
...
OVER_CLAUSE=Y
RISQL_FUNCTIONS=RANK,SUM,AVG,COUNT,MIN,MAX,
VARIANCE,STDDEV,RATIO_TO_REPORT
```

6. 수정 내용을 저장하고 파일을 닫습니다.
   PRM 파일의 변경 내용을 적용하려면 유니버스 디자인 도구를 다시 시작해야 합니다.

7.5.5.3 DB2, UDB 및 Oracle RDBMS 분석 함수 사용 규칙

다음 규칙은 DB2 UDB 및 Oracle 에 대해 분석 함수를 사용할 때 적용됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 175:

규칙	설명
분석 함수는 GROUP BY 절에 나타날 수 없습니다.	분석 함수에서 정의된 SUM 과 같은 집계 함수는 GROUP BY 절에서 사용되지만, RANK 와 같은 분석 함수는 사용되지 않습니다. 분석 함수가 GROUP BY 절에서 사용되지 않도록 하기 위해 분석 함수는 PRM 파일에서 RISQL FUNCTIONS 매개 변수 다음에 나열됩니다. 이 매개 변수 앞에 오는 OVER_CLAUSE 는 Y 로 설정되어야 합니다. 이 값은 기본 설정입니다.
분석 함수는 GROUP BY 절을 생성하면 안 됩니다.	PRM 파일의 Functions 섹션에 분석 함수를 추가(SQL 편집 대화 상자에서 함수 목록 추가)한 경우 GROUP CLAUSE 가 N 으로 설정되어 있는지 확인해야 합니다. 그러면 분석 함수가 GROUP BY 절을 생성하지 못합니다. 자세한 내용은 Select 문에 자동으로 구문 삽입 [페이지 387] 단원을 참조하십시오.
분석 함수에서 집계 함수를 사용하는 경우 분석 함수에서 사용된 모든 차원이 GROUP BY 절에 나타납니다.	예를 들면 RANK() OVER (PARTITION BY year ORDER BY SUM(sales)입니다. 쿼리에서 rank 함수만 사용되었더라도 GROUP BY 절에는 year 차원이 포함됩니다.

7.5.5.4 Oracle 및 DB2 에서의 분석 함수 사용 제한 사항

IBM DB2 UDB v7.1 및 Oracle 8.1.6 에서 분석 함수를 사용할 때 다음과 같은 제한 사항이 있습니다.

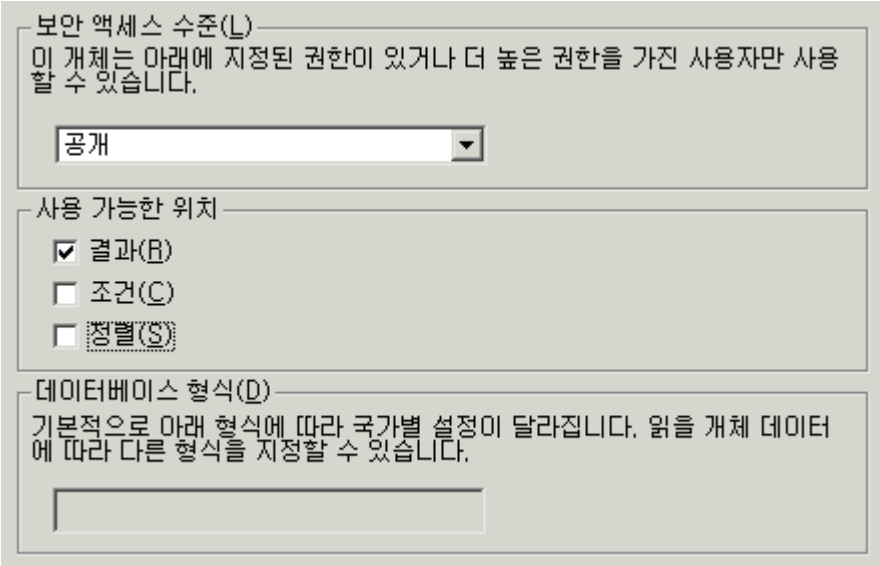
- 분석 함수를 사용하는 개체 정의에서 @Prompt 및 @Variable 함수를 사용할 수 없습니다.
- 분석 함수는 사용자 개체로 지원되지 않습니다. PRM 파일의 Functions 섹션에 분석 함수를 추가(SQL 편집 대화 상자에서 함수 목록 추가)한 경우 IN MACRO 가 N 로 설정되어 있는지 확인해야 합니다.
- 분석 함수를 사용하는 개체는 조건으로 사용되거나 정렬에서 사용될 수 없습니다. 일반 사용자가 이러한 개체를 사용하여 조건을 정의하려고 시도하면 SQL 오류 메시지가 나타납니다. 개체 속성을 다음과 같이 편집하면 일반 사용자가 조건이나 정렬에서 개체를 사용하지 못하도록 막을 수 있습니다.

조건 또는 정렬에서 분석 개체 사용 방지

조건 또는 정렬에서 분석 개체를 사용하지 못하게 하려면 다음과 같이 하십시오.

1. 유니버스 디자인 도구에서 개체를 마우스 오른쪽 단추로 클릭합니다.
2. 상황에 맞는 메뉴에서 개체 속성을 선택합니다.
   속성 편집 대화 상자가 나타납니다.
3. 사용 가능한 위치 그룹 상자에서 조건 및 정렬 확인란의 선택을 취소합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



4. 확인을 클릭합니다.

7.5.6 RedBrick(RISQL 함수)

다음 단원에서는 유니버스 디자인 도구에서 RISQL 함수를 사용할 수 있는 방법에 대해 설명합니다.

7.5.6.1 RedBrick RISQL 함수에 대한 SELECT 문 정의

개체에 대한 Select 문에서 분석 함수를 정의합니다. Select 문에 대한 편집 상자 중 하나에 구문을 입력해야 합니다.

> **i 노트**
>
> Select 문 편집 대화 상자의 함수 목록에 RISQL 함수를 추가하면 구문 입력을 자동화할 수 있습니다. 함수 목록에서 함수를 사용할 수 있게 하려면 PRM 파일의 [FUNCTIONS] 섹션에 RISQL 함수를 추가해야 합니다. 자세한 내용은 Select 문에 자동으로 구문 삽입 [페이지 387] 단원을 참조하십시오.

각 RISQL 함수 계열의 구문에 대한 설명은 다음과 같습니다.

표 176:

함수 계열	구문	설명
순위(RANK)	RANK(arg1) 예: `RANK(SUM(telco_facts.total_billed_rev))`	arg1 은 필수입니다. 순위는 이 인수를 기준으로 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

함수 계열	구문	설명
집계 계열(CUME, MOVINGAVG, MOVINGSUM)	MOVINGSUM(arg1,Number) 예: `MOVINGSUM (COUNT(complants.id),2)`	• arg1 은 필수입니다. 누적 집계는 이 인수를 기준으로 합니다. • 숫자는 선택 사항입니다. 이 숫자는 합계에 사용된 이전 줄의 수입니다.
비율(RATIOTOREPORT)	RATIOTOREPORT(arg1) 예: `RATIOTOREPORT (SUM(telco_facts.total_billed_rev))`	arg1 은 필수입니다. 비율은 이 인수를 기준으로 합니다.

7.5.6.2 PRM 파일 확인 및 RISQL 함수 지원 기능 추가

PRM 파일에는 사용 가능한 모든 RISQL 함수가 포함되어 있지 않을 수 있습니다. RISQL 함수를 사용하기 전에 RISQL 함수가 PRM 파일의 RDBMS 섹션에 나열되어 있는지 확인하고, 필요한 경우 목록에 추가하십시오. 다음과 같이 할 수 있습니다.

Redbrick PRM 파일에 분석 함수 지원 기능을 추가하려면 다음과 같이 하십시오.

1. Business Objects 경로의 Data Access 디렉터리로 이동합니다.
2. RDBMS 의 PRM 파일을 텍스트 편집기에서 엽니다.
3. PRM 파일에서 RDBMS 섹션으로 스크롤합니다.
4. 다음 매개 변수 및 값이 존재하는지 확인합니다.

   표 177:

PRM 의 매개 변수 및 값	설명
OLAP_CLAUSE = WHEN	조건을 적용합니다.
RISQL_FUNCTIONS = <사용된 함수 목록>	사용 가능한 분석 함수입니다.

   예를 들면 다음과 같습니다.

   ```
   [RDBMS]
   (GENERAL)
   ...
   OVER_CLAUSE=WHEN
   RISQL_FUNCTION= RANK, CUME, MOVINGSUM, MOVINGAVG, RATIOTOREPORT,TERTILE
   ```

5. 나열되지 않은 RISQL 함수를 사용하려는 경우 목록 끝에 함수의 이름을 입력합니다.
6. 수정 내용을 저장하고 파일을 닫습니다.
   PRM 파일의 변경 내용을 적용하려면 유니버스 디자인 도구를 다시 시작해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.5.6.3 RISQL 함수 사용 규칙

RISQL 함수를 사용할 경우 다음 규칙이 적용됩니다.

표 178:

규칙	설명
RISQL 함수는 GROUP BY 절에 나타날 수 없습니다.	RISQL 함수에서 정의된 SUM 과 같은 집계 함수는 GROUP BY 절에서 사용되지만, RANK 와 같은 분석 함수는 사용되지 않습니다. RISQL 함수가 GROUP BY 절에서 사용되지 않도록 하기 위해 RISQL 함수는 PRM 파일에서 RISQL FUNCTIONS 매개 변수 다음에 나타납니다. 이 매개 변수 앞에 오는 OVER_CLAUSE 는 WHEN 으로 설정되어야 합니다. 이 값은 기본 설정입니다.
RISQL 함수는 GROUP BY 절을 생성하면 안 됩니다.	PRM 파일의 Functions 섹션에 RISQL 함수를 추가(SQL 편집 대화 상자에서 함수 목록 추가)한 경우 GROUP CLAUSE 가 N 으로 설정되어 있는지 확인해야 합니다. 그러면 RISQL 함수가 GROUP BY 절을 생성하지 못합니다. 자세한 내용은 Select 문에 자동으로 구문 삽입 [페이지 387] 단원을 참조하십시오.
RISQL 함수는 조건에서 사용할 수 있습니다.	WHEN 절이 생성됩니다.

7.5.6.4 RedBrick 에서 분석 함수 사용 제한 사항

RISQL 함수를 사용할 때 다음과 같은 제한 사항이 있습니다.

- RESET BY 절이 지원되지 않습니다.
- SORT BY 절이 지원되지 않습니다. 최종 사용자가 개체 속성을 편집하는 과정에서 정렬된 개체를 사용하지 않도록 하는 방법은 *Oracle* 및 *DB2* 에서 분석 함수 사용 시 제한 단원을 참조하십시오.

7.5.7 Teradata(OLAP 함수)

다음 단원에서는 유니버스 디자인 도구에서 OLAP 함수를 사용할 수 있는 방법에 대해 설명합니다.

7.5.7.1 Teradata OLAP 함수에 대한 Select 문 정의

비율 함수는 Teradata V2R3 에서 사용할 수 없습니다. 개체에 대한 Select 문에서 OLAP 함수를 정의합니다. Select 문에 대한 편집 상자 중 하나에 구문을 입력해야 합니다.

구문 입력 자동화를 위해 함수를 함수 목록에서 사용할 수 있도록 하는 방법은 *Oracle* 및 *DB2* 에서 분석 함수 사용 시 제한 단원을 참조하십시오.

각 OLAP 함수 계열의 구문에 대한 설명은 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 179:

함수 계열	구문	설명
순위(RANK)	RANK(arg1 DESC/ASC) 예: `RANK(invoice_line.nb_guests)`	• arg1 은 필수입니다. 순위는 이 인수를 기준으로 합니다. 인수는 개체 또는 개체 목록이 될 수 있습니다. **노트** 집계 개체(sum, avg, min, count)를 arg1 로 사용하는 개체는 사용할 수 없습니다. • DESC/ASC 는 순위 정렬 방식을 지정합니다. ASC 가 기본값입니다.
집계 계열(CSUM, MAVG, MDIFF, MLINREG, MSUM)	CSUM(arg1 DESC/ASC) 예: `CSUM(invoice_line.nb_guests)`	• arg1 은 필수입니다. 누적 집계는 이 인수를 기준으로 합니다. 인수는 개체 또는 개체 목록이 될 수 있습니다. • DESC/ASC 는 결과 행의 정렬 방식을 지정합니다. ASC 가 기본값입니다.

7.5.7.2 PRM 파일에서 OLAP 함수 지원 확인 및 추가

Teradata 용 PRM 파일은 OLAP 함수 사용을 지원하도록 업데이트되었습니다. 하지만 PRM 파일에는 사용 가능한 모든 OLAP 함수가 포함되어 있지 않을 수 있습니다. OLAP 함수를 사용하기 전에 OLAP 함수가 PRM 파일의 RDBMS 섹션에 나열되어 있는지 확인하고, 필요한 경우 목록에 추가하십시오. 다음과 같이 할 수 있습니다.

Teradata PRM 파일에 분석 함수 지원 기능을 추가하려면 다음과 같이 하십시오.

1. Business Objects 경로의 Data Access 디렉터리로 이동합니다.
2. RDBMS 의 PRM 파일을 텍스트 편집기에서 엽니다.
3. PRM 파일에서 RDBMS 섹션으로 스크롤합니다.
4. 다음 매개 변수 및 값이 존재하는지 확인합니다.

   표 180:

PRM 의 매개 변수 및 값	설명
OLAP_CLAUSE = QUALIFY	조건을 적용합니다.
RISQL_FUNCTIONS = <사용된 함수 목록>	사용 가능한 분석 함수입니다.

   예를 들면 다음과 같습니다.

   ```
   [RDBMS]
   (GENERAL)
   ...
   OVER_CLAUSE=QUALIFY
   RISQL_FUNCTION= RANK, CSUM, MAVG, MDIFF,MLINREG,MSUM,QUANTILE
   ```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

5. 나열되지 않은 RISQL 함수를 사용하려는 경우 목록 끝에 함수의 이름을 입력합니다.
6. 수정 내용을 저장하고 파일을 닫습니다.
   PRM 파일의 변경 내용을 적용하려면 유니버스 디자인 도구를 다시 시작해야 합니다.

7.5.7.3 OLAP 함수 사용 규칙

OLAP 함수를 사용할 경우 다음 규칙이 적용됩니다.

- OLAP 함수는 GROUP BY 절에 나타날 수 없습니다. OLAP 함수가 GROUP BY 절에서 사용되지 않도록 하기 위해 OLAP 함수는 PRM 파일에서 RISQL FUNCTIONS 매개 변수 다음에 나열됩니다. 이 매개 변수 앞에 오는 OVER_CLAUSE 는 QUALIFY 로 설정되어야 합니다. 이 값은 기본 설정입니다.
- 동일한 쿼리에서 OLAP 함수를 사용하는 개체와 집계 함수를 사용하는 개체를 함께 사용할 수 없습니다.
- OLAP 함수는 조건에서 사용할 수 있습니다. QUALIFY 절이 생성됩니다.
- OLAP 함수는 SORT BY 절에서 사용할 수 있습니다.

7.5.7.4 Teradata 에서 분석 함수 사용 제한 사항

OLAP 함수를 사용할 때 다음과 같은 제한 사항이 있습니다.

- RESET BY 절이 지원되지 않습니다.
- OLAP 함수는 하위 쿼리에서 사용할 수 없습니다.
- OLAP 함수는 다른 함수와 동일한 Select 문에서 사용할 수 없습니다.
- OLAP 함수는 다른 함수를 기반으로 할 수 없습니다.
- OLAP 함수는 사용자 개체로 지원되지 않습니다.

7.5.8 Select 문에 자동으로 구문 삽입

Select 문 편집 대화 상자의 함수 목록에 분석 함수를 추가하면 분석 함수 구문을 자동으로 삽입할 수 있습니다.

대상 RDBMS 용 PRM 파일의 [FUNCTION] 섹션 아래 있는 함수 목록에 분석 함수를 추가하여 함수 목록 상자를 채웁니다.

PRM 파일에 함수를 추가하면 Select 문 편집 대화 상자의 함수 목록 상자에서 함수를 사용할 수 있게 됩니다. 함수 구문을 두 번 클릭하면 정의된 구문이 편집 상자에 삽입됩니다.

PRM 파일에 분석 함수를 추가할 때 다음 값을 설정해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 181:

매개 변수	설명
GROUP = N	분석, RISQL 및 OLAP 함수는 GROUP BY 절을 생성할 수 없습니다. 값을 N 으로 설정하면 분석 함수가 GROUP BY 절에서 사용되는 것을 막을 수 있습니다.
IBM DB2 UDB v.7.1 및 ORACLE 8.1.6 만 해당: IN_MACRO = N	DB2 UDB 및 Oracle 의 분석 함수가 사용자 개체로 사용되는 것을 막습니다. RedBrick 및 Teradata 의 경우 이 값을 Y 로 설정할 수 있습니다.

다음과 같이 PRM 파일의 [FUNCTION] 섹션에 분석 함수를 추가할 수 있습니다.

PRM 파일에 분석 함수를 추가하려면 다음과 같이 하십시오.

1. Business Objects 경로의 Data Access 디렉터리로 이동합니다.
2. RDBMS 의 PRM 파일을 텍스트 편집기에서 엽니다.
3. PRM 파일에서 [FUNCTION] 섹션으로 스크롤합니다.
4. 기존 함수를 복사하여 목록 끝에 붙여 넣습니다.
5. 새로 붙여 넣은 함수에 대한 고유 번호를 입력하고, 목록에 추가하려는 분석 함수에 알맞게 값을 수정합니다.
6. GROUP 값을 N 으로 설정합니다.
   IBM DB2 UDB 또는 ORACLE 을 사용하는 경우 IN_MACRO 값을 N 으로 설정합니다.
   예:

```
(n)
NAME: RANK
TRAD:
HELP: Return the rank of
TYPE=N
IN_MACRO=N
GROUP=N
SQL=
```

7. PRM 파일을 저장하고 닫습니다.
   변경 내용을 적용하려면 유니버스 디자인 도구를 다시 시작해야 합니다.

> **노트**
>
> 유니버스 디자인 도구를 다시 시작하면 추가한 분석 함수에 대한 구문이 해당하는 형식 노드(숫자, 문자 또는 날짜) 아래에 나타납니다.

7.6 SQL 접두사 함수 사용

SQL 접두사 함수는 BusinessObjects 가 생성한 SQL 문 이전에 유니버스 매개 변수 `Begin_SQL` 을 사용하여 SQL 명령을 삽입합니다. 이렇게 하면 해당 명령이 생성된 모든 SQL 문 이전에 실행됩니다. 이 함수는 SELECT 문 이전에 매개 변수 전달을 지원하는 모든 데이터베이스에서 작동합니다. 이에 대한 예는 다음과 같습니다.

- Teradata: 트랜잭션에 'QUERY_BAND' 사용(Teradata 문서 참조)
- Oracle: 읽기 전용 트랜잭션을 시작합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- Netezza: 최적화 옵션을 트리거합니다.

SQL 접두사 함수를 설정하려면 유니버스에 SQL_prefix 매개 변수를 설정합니다.

관련 정보

사용자 인터페이스에서 설정한 SQL 매개 변수 [페이지 91]

7.6.1 BEGIN_SQL 유니버스 매개 변수를 사용하여 SQL 문에 접두사를 사용하려면

데이터베이스가 SELECT 문 이전에 매개 변수 전달을 지원합니다.

BEGIN_SQL 매개 변수를 사용하면 SQL 문이 생성될 때마다 같은 매개 변수를 사용하여 SQL 문에 접두사를 사용할 수 있습니다.

1. 유니버스 매개 변수 대화 상자를 엽니다.
2. 매개 변수 탭을 클릭합니다.
3. 매개 변수 목록에서 BEGIN_SQL 매개 변수를 선택하고 적합한 접두사 명령을 입력합니다.
4. 설정을 저장합니다.
5. 유니버스를 저장합니다.

예

이 예에서는 BEGIN_SQL 매개 변수에 Teradata 를 사용합니다. 쿼리에는 보고에 사용할 수 있도록 쿼리에 바운딩된 사용자 ID 및 응용 프로그램 ID 가 포함됩니다. 유니버스 매개 변수 대화 상자의 매개 변수 창에서 BEGIN_SQL 매개 변수는 다음과 같이 설정됩니다.

```
BEGIN_SQL = SET QUERY_BAND = 'UserId=DG12234;AppId=TRM;' FOR TRANSACTION;
```

쿼리를 실행할 경우 두 SQL 문이 아래와 같이 실행됩니다.

1) BEGIN_SQL 문:

```
SET QUERY_BAND = 'UserId=DG12234;AppId=TRM;' FOR TRANSACTION;
```

2) 기본 쿼리 SQL 결과 문:

```
SELECT
    RESORT_COUNTRY.COUNTRY, sum(INVOICE_LINE.DAYS * INVOICE_LINE.NB_GUESTS
      * SERVICE.PRICE)
    FROM
    COUNTRY RESORT_COUNTRY, INVOICE_LINE, RESORT_COUNTRY.COUNTRY
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.7 배열 반입 매개 변수 최적화

CS.CFG 파일의 배열 반입 매개 변수를 사용하여 FETCH 프로시저에서 허용되는 열의 최대 수를 설정할 수 있습니다. CFG 파일은 데이터베이스에 대해 쿼리를 실행할 때 Business Objects 제품에 사용되는 특정 매개 변수에 대한 기본값을 지정하는 XML 파일입니다.

배열 반입 매개 변수는 네트워크에서 패킷 크기를 결정합니다. 예를 들어, 배열 반입 크기를 20 으로 설정하고 100 개 행을 검색하고자 하는 경우 데이터를 검색하기 위해 반입이 다섯 번 실행됩니다.

일부 데이터 소스에서는 FETCH 크기 수정이 허용되지 않습니다. 이 경우에는 모든 행이 한 번의 FETCH 로 검색됩니다. BLOB(binary long-objects)을 검색하고자 할 경우에는 배열 반입 크기를 1 로 설정해야 합니다.

큰 배열 반입을 보낼 수 있도록 허용하는 네트워크에서는 더 큰 값(가능한 범위는 1 부터 999 까지)을 새로 설정할 수 있습니다. 그러면 FETCH 프로시저 속도가 향상되어 쿼리 처리 시간이 단축됩니다.

7.7.1 배열 반입 매개 변수 수정

배열 반입 매개 변수를 수정하려면

1. XML 편집기에서 CS.CFG 파일을 엽니다.
   CFG 파일은 다음 디렉터리에 있습니다.
   <INSTALDIR>\dataAccess\RDBMS\connectionServer
2. Array Fetch 매개 변수를 찾습니다.
3. 매개 변수 값을 설정합니다. CFG 파일을 저장하고 닫습니다.
4. 유니버스 디자인 도구를 다시 시작합니다.

7.8 테이블 가중치 할당

테이블 가중치는 테이블에 존재하는 행에 대한 계수입니다. 가중치가 낮은 테이블은 가중치가 높은 테이블보다 행이 적습니다. 기본적으로 BusinessObjects 에서는 가중치가 낮은 테이블에서 가중치가 높은 테이블(행 수가 가장 적은 테이블에서 행 수가 가장 많은 테이블) 순으로 테이블을 정렬합니다. 이에 따라 SQL 문의 FROM 절에서 테이블 순서가 결정됩니다.

데이터베이스 수준에서 테이블이 정렬되는 순서는 데이터베이스에 따라 다릅니다. 예를 들어, Sybase 에서는 BusinessObjects 와 동일한 순서를 사용하지만 Oracle 에서는 반대의 순서를 사용합니다. 가장 작은 테이블이 정렬 순서에서 가장 앞에 오는 Oracle 의 경우를 제외하고 대부분의 데이터베이스에서는 SQL 이 최적화됩니다.

따라서 Oracle 데이터베이스를 사용하는 경우 BusinessObjects 에서 테이블이 정렬되는 순서를 바꾸면 SQL 을 최적화할 수 있습니다. 이렇게 하려면 데이터베이스의 관련 PRM 파일에서 매개 변수를 변경해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

7.8.1 PRM 파일을 수정하여 테이블 가중치 할당

PRM 파일을 수정하여 테이블 가중치를 할당하려면

1. 데이터베이스의 PRM 파일을 XML 편집기에서 엽니다.
   PRM 파일은 다음 디렉터리에 있습니다.
   <INSTALLDIR>\dataAccess\RDBMS\connectionServer\<rdbms>\
   예를 들어, Oracle 의 경우 이 파일은 oracle.prm 이며 그 위치는 다음과 같습니다.
   <INSTALLDIR>\dataAccess\RDBMS\connectionServer\oracle\oracle.prm
2. 파일의 Configuration 섹션에서 REVERSE_TABLE_WEIGHT 매개 변수를 찾습니다.
3. Y 를 N 로 변경합니다.
   예를 들어, 매개 변수가 REVERSE_TABLE_WEIGHT=N 로 나타납니다.
   파일에 이 줄이 없는 경우 기본값은 Y 입니다.
4. 이제 BusinessObjects 에서 행이 가장 많은 테이블에서 행이 가장 적은 테이블 순으로 테이블이 정렬됩니다.
5. .PRM 파일을 저장하고 닫습니다.
6. 유니버스 디자인 도구를 다시 시작하여 .PRM 파일에 변경 내용을 적용합니다.

7.9 테이블에 대해 반환되는 행 수 수정

유니버스 디자인 도구에서 테이블의 행 수를 수동으로 변경할 수도 있습니다. 테이블의 행 수를 보려면 보기 > 테이블의 행 수를 선택하십시오. 행 수가 각 테이블 기호의 왼쪽 아래쪽에 나타납니다. 이 수를 다음과 같이 수정할 수 있습니다.

1. 유니버스 디자인 도구에서 유니버스를 엽니다.
2. 관련 테이블을 마우스 오른쪽 단추로 클릭합니다.
3. 상황에 맞는 메뉴에서 테이블의 행 수를 선택합니다.
   테이블 행 수 대화 상자가 나타납니다.
4. 테이블 행 수 직접 수정 라디오 단추를 선택합니다.
   대화 상자의 왼쪽에 텍스트 상자가 나타납니다.
5. 텍스트 상자에 숫자를 입력합니다. 이 숫자는 테이블에 대해 사용하고자 하는 행 수입니다.
6. 확인을 클릭한 후 유니버스를 저장합니다.

7.10 바로 가기 조인 사용

바로 가기 조인은 공통 경로에서 이미 조인된 두 테이블을 링크합니다. 바로 가기 조인을 사용하면 쿼리에서 사용되는 테이블의 수를 줄일 수 있습니다. 자세한 내용은 외부 조인 사용에 대한 제한 사항 [페이지 158] 단원을 참조하십시오.

> **노트**
>
> 바로 가기 조인은 루프를 만들지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8 OLAP 유니버스 작업

8.1 OLAP 유니버스 정보

8.1.1 OLAP 유니버스란?

OLAP 유니버스는 OLAP 큐브나 쿼리에서 생성된 BusinessObjects 유니버스입니다. 이 유니버스는 OLAP 데이터 소스에 대한 선택된 연결에서 자동으로 만들어집니다.

유니버스를 만든 후에 중앙 관리 서버(CMS)에 다른 유니버스로 내보낼 수 있습니다. 그런 후 Web Intelligence 사용자는 이 유니버스를 사용하여 쿼리를 실행하고 보고서를 만들 수 있습니다.

다음 방법으로 OLAP 유니버스를 생성하고 유지 관리할 수 있습니다.

- OLAP 유니버스를 생성하려면 먼저 OLAP 데이터 소스를 선택합니다.

  노트

  OLAP 데이터 소스에 안전하게 연결하기 위해서는 유니버스를 생성하거나 구조를 새로 고쳐야 하는 사용자에게 연결에 대한 다운로드 권한이 있어야 합니다. 권한은 관리자가 CMC 에서 설정합니다.

- 새 연결 마법사를 사용하여 데이터 소스에 대한 연결을 정의하고 새 유니버스에 대한 연결을 선택합니다.
  유니버스 디자인 도구가 유니버스를 자동으로 생성합니다. OLAP 구조는 유니버스의 클래스, 계수, 차원, 설명 및 필터에 직접 매핑됩니다. 유니버스 구조가 유니버스 창에 나타납니다.
- OLAP 유니버스를 저장하고 CMS 로 내보낼 수 있습니다.
- 모든 OLAP 유니버스 구성 요소를 수정할 수 있습니다.
- OLAP 유니버스 업데이트 마법사를 사용하여 OLAP 유니버스의 수명 주기를 관리할 수 있습니다. 이 마법사는 OLAP 데이터 소스 변경 내용으로 유니버스 구조를 자동으로 새로 고칩니다. 마법사는 생성된 개체와 수동으로 추가 또는 수정된 개체를 구분할 수 있으므로 유니버스 디자인 도구에서 수동으로 변경한 내용을 보존할 수 있습니다.

관련 정보

유니버스를 만드는 데 사용할 수 있는 OLAP 데이터 소스 [페이지 392]
OLAP 데이터 소스 연결 정보 [페이지 398]
OLAP 유니버스에 대해 지원되는 유니버스 디자인 도구 기능 [페이지 407]
OLAP 유니버스 수명 주기 관리 정보 [페이지 420]

8.1.2 유니버스를 만드는 데 사용할 수 있는 OLAP 데이터 소스

다음과 같은 OLAP 데이터 소스에서 OLAP 유니버스를 자동으로 만들 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

SAP Business Warehouse(BW)
Microsoft Analysis Services(MSAS) 2000
Microsoft Analysis Services(MSAS) 2005
Hyperion Essbase

> **노트**
>
> 유니버스 디자인 도구, Web Intelligence Rich Client 및 Web Intelligence 를 사용하는 SAP BusinessObjects OLAP 제품에서 Essbase OLAP 데이터 소스로 연결하려면 Essbase 클라이언트 미들웨어가 해당 SAP BusinessObjects OLAP 제품을 호스트하는 컴퓨터에서 올바르게 설치되고 구성되어야 합니다. 특히 Essbase 클라이언트 환경 변수 ARBORPATH 및 ESSBASEPATH 가 만들어지고 Windows 사용자 환경 변수가 아닌 Windows 시스템 환경 변수로 설정되어야 합니다.

하나의 큐브 또는 쿼리에서 하나의 유니버스가 자동으로 생성됩니다. OLAP 유니버스는 단일 큐브만 포함할 수 있습니다.

관련 정보

SAP Business Warehouse(BW) 데이터 소스 [페이지 393]
유니버스에서 SAP BW 개체가 매핑 및 사용되는 방식 [페이지 449]
OLAP 유니버스에 대해 지원되는 MSAS 기능 [페이지 397]
MSAS 큐브가 유니버스 구성 요소에 매핑되는 방식 [페이지 458]
OLAP 유니버스에 대해 지원되는 Essbase 기능 [페이지 398]
Essbase 큐브가 유니버스 구성 요소에 매핑되는 방식 [페이지 457]

8.1.2.1 SAP Business Warehouse(BW) 데이터 소스

BW 데이터 소스를 기반으로 OLAP 유니버스를 만들 경우 인포큐브 또는 멀티큐브를 직접적인 기반으로 사용하거나 정보 제공자 위에서 사용하도록 설정된 BEx 쿼리를 기반으로 유니버스를 작성할 수 있습니다. 다음과 같은 정보 제공자가 있습니다.

인포큐브
멀티큐브 또는 멀티 정보 제공자
운영 데이터 저장소(ODS)
인포세트

관련 정보

SAP Business Warehouse(BW) 인포큐브를 데이터 소스로 사용 [페이지 394]
SAP BW 쿼리를 데이터 소스로 사용 [페이지 394]
권장 데이터 소스로의 쿼리 [페이지 395]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.1.2.1.1 SAP Business Warehouse(BW) 인포큐브를 데이터 소스로 사용

OLAP 유니버스를 작성할 때 데이터 소스로 다음 유형의 인포큐브가 지원됩니다.

- 표준 및 트랜잭션 인포큐브: 데이터 및 메타데이터가 동일한 SAP Business Warehouse(BW) 시스템에 실제로 저장됩니다.
- 원격 인포큐브: 데이터가 원격 시스템에 실제로 저장됩니다.

> **노트**
>
> 임시 쿼리, 보고 및 분석 사용 시나리오에서는 완전히 지원되는 경우에도 원격 인포큐브에서 유니버스를 작성 및 배포하는 것은 권장되지 않습니다. 이러한 아키텍처는 일반적으로 대화식 쿼리에 대한 쿼리 성능 요구를 충족하지 않습니다.

- 멀티큐브 및 멀티 정보 제공자

> **노트**
>
> 멀티큐브 또는 멀티 정보 제공자 위에 Business Objects 유니버스를 작성하고 배포하는 것은 인포큐브 위에 유니버스를 작성하고 배포하는 것과 같습니다.

시간 및 단위를 비롯하여 인포큐브의 모든 특성, 계층, 주요 수치는 유니버스에서 볼 수 있습니다.

8.1.2.1.2 SAP BW 쿼리를 데이터 소스로 사용

SAP BW 고객은 BEx 쿼리를 사용하여 SAP Business Explorer 프론트엔드에 액세스합니다.

> **노트**
>
> OLE DB for OLAP 용으로 출시된 BEx 쿼리만 데이터 소스로 작동하며 Business Objects 유니버스에 대한 OLAP 인터페이스를 통해 사용할 수 있습니다. SAP BW Query Designer 의 BEx 쿼리를 외부에서 액세스하려면 쿼리 속성 대화 상자의 확장 모드 탭을 선택합니다.

행, 열 및 자유 특성으로 선택된 BEx 쿼리의 모든 InfoObject 를 유니버스에서 볼 수 있습니다. 여기에는 특성, 계층, 주요 수치, 구조 및 변수가 포함됩니다.

인포세트와 운영 데이터 저장소(ODS) 모두 BEx 쿼리를 통해 유니버스에 노출될 수 있습니다.

ODS 를 기반으로 하는 쿼리

ODS 는 BEx 쿼리를 통해 유니버스에 노출될 수 있습니다.

ODS 개체는 인포큐브로 집계되기 전에 자세한 트랜잭션 수준 데이터를 관리하는 데 주로 사용됩니다. SAP NetWeaver 기술 플랫폼 데이터 저장소 디자인에 ODS 개체를 포함하면 인포큐브 크기를 최소화하고 로드 및 쿼리 성능을 향상시킬 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

> **노트**
>
> ODS 는 일반적으로 크고 자세한 관계형 구조입니다. OLAP BAPI 인터페이스를 통해 ODS 에 액세스하면 쿼리 성능이 저하될 수 있습니다. 빠른 보고서 전달에 대한 최종 사용자 요구를 만족시키려면 다음 방법을 고려하십시오.

- BAPI 호출을 통해 ODS 에 직접 액세스
- Web Intelligence 에서 직접 SQL 을 사용하여 ODS 테이블에 액세스

인포세트를 기반으로 하는 쿼리

인포세트는 BEx 쿼리를 통해 유니버스에 노출될 수 있습니다.

인포세트는 마스터 데이터를 보고하도록 SAP BW 에서 정의되기도 합니다.

> **노트**
>
> 인포큐브를 기반으로 유니버스를 작성하여 마스터 데이터를 보고할 수 있으므로 인포세트 및 BEx 쿼리를 사용하지 않아도 됩니다. 두 가지 방법의 주요 차이점은 인포큐브에서 보고된 마스터 데이터가 데이터를 유효한 트랜잭션으로 제한한다는 것입니다.

관련 정보

권장 데이터 소스로의 쿼리 [페이지 395]

8.1.2.1.3 권장 데이터 소스로의 쿼리

BEx 쿼리는 다음과 같은 이유로 Business Objects 유니버스 생성을 위한 데이터 소스로 권장됩니다.

- 다음 표에 요약된 것처럼 모든 SAP BW 메타데이터 기능을 인포큐브 수준에서 가져올 수 있는 것은 아닙니다.

표 182:

BW 메타데이터 기능	SAP OLAP Business Application Programming Interface(BAPI) 지원 수준
특성(시간 및 단위 포함)	인포큐브/BEx 쿼리
계층구조	인포큐브/BEx 쿼리
기본 주요 수치	인포큐브/BEx 쿼리
탐색 특성	BEx 쿼리만
표시 특성	인포큐브/BEx 쿼리
계산된 주요 수치/수식	BEx 쿼리만

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

BW 메타데이터 기능	SAP OLAP Business Application Programming Interface(BAPI) 지원 수준
제한된 주요 수치	BEx 쿼리만
사용자 지정 구조	BEx 쿼리만
변수	BEx 쿼리만

- BEx 쿼리를 통해 데이터 모델링 환경을 효과적으로 확장할 수 있습니다. 인포큐브를 변경할 때는 많은 주의를 기울여야 합니다.
- BEx 쿼리는 최종 사용자의 요구에 맞는 맞춤형 데이터 소스를 만들 수 있는 유용한 기능을 제공합니다.

BEx 쿼리는 데이터 소스로 많은 이점을 갖지만 모든 보고서에 BEx 쿼리가 필요한 것은 아니며, 기존의 모든 BEx 쿼리에 유니버스가 있어야 하는 것도 아닙니다. 유지 관리 비용을 최소화하려면 임시 쿼리 및 보고 요구를 모두 만족하는 데 필요한 BEx 쿼리 및 유니버스의 최종 개수를 제한하는 구현 전략을 구축하는 데 주력해야 합니다. 필요한 유니버스의 수를 줄이려면 다음 사항에 유의하십시오.

Web Intelligence 가 프런트 엔드 도구이면 BEx 쿼리의 출력 형식에는 제한이 없습니다.

큰 BEx 쿼리에서 만든 OLAP 유니버스로 작업할 경우 성능에 직접적인 영향은 없습니다. Web Intelligence 쿼리에 삽입되지 않은 OLAP 유니버스 개체는 쿼리 성능에 직접적으로 영향을 주지 않습니다.

노트

임시 쿼리 및 보고용으로 사용되는 모든 인포큐브 또는 멀티큐브에 대해 하나 또는 소수의 BEx 쿼리만 작성하는 것이 좋습니다. 그런 다음 해당하는 각 BEx 쿼리를 기반으로 유니버스를 구축합니다.

8.1.2.1.4 SAP BW 다국어 유니버스

Web Intelligence 에서는 SAP BW 의 다국어 기능을 활용할 수 있습니다. 다국어 환경을 구현하려면 BW 시스템에 다국어 메타데이터 및 다국어 데이터가 포함되어 있어야 합니다.

솔루션이 지원하는 각 언어에 대한 유니버스를 만들어야 합니다. 유니버스 연결이 만들어진 언어로 유니버스가 만들어집니다.

사용자의 SAP 인증이 쿼리에 반환되는 데이터의 언어를 결정합니다. 사용자는 SAP 인증을 사용하여 InfoView 에 로그인하고 SAP 서버에서 반환된 결과가 표시될 언어를 지정해야 합니다.

결과 집합 언어는 SAP 의 유니코드 지원에 따라 결정됩니다. SAP 시스템에 데이터가 원하는 언어로 포함되어 있지 않을 경우 데이터를 Web Intelligence 에서 해당 언어로 사용할 수 없습니다. BW 에서 설명이 번역되지 않은 경우 Web Intelligence 에서는 설명 대신 기술 이름이 표시됩니다.

8.1.2.1.5 유니버스 디자인 도구에서 SAP BW 를 사용하기 위한 필수 조건

SAP BW 데이터 소스에서 유니버스를 만들 경우 뷰 타임 동안 단일 로그온(SSO)을 활성화할 수 있습니다. SSO 를 사용하면 사용자들은 해당 SAP 자격 증명으로 SAP BusinessObjects Enterprise 에 로그온하고 SAP 인증을 사용할 수 있게 됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

SAP 상에서 OLAP 유니버스에 대한 SSO 를 활성화하려면 SAP Integration 을 설치하고 SAP 보안 플러그인을 구성해야 합니다.

SAP 보안 통합이 구성되면 SAP 자격 증명을 사용하여 유니버스 디자인 도구를 시작할 수 있습니다. SAP 사용자 ID 로 보안 통합을 구성할 때 정의된 대로 SAP 시스템 ID 와 SAP 클라이언트 ID 를 연결하여 BusinessObjects Enterprise 사용자 이름을 만듭니다.

자세한 내용은 *SAP 솔루션용 Business Objects XI Integration 설치 가이드* 및 *SAP 솔루션용 Business Objects XI Integration 사용자 가이드*를 참조하십시오.

8.1.2.2 OLAP 유니버스에 대해 지원되는 MSAS 기능

다음 표에서는 MSAS 데이터 소스에서 생성된 유니버스의 MSAS 기능에 대한 지원 수준에 대해 설명합니다.

MSAS 메타데이터 기능	OLAP 유니버스 지원 수준
큐브	지원
로컬 큐브	지원
가상 큐브(MSAS 2000)	지원
큐브 뷰(MSAS 2005)	지원
차원	지원
가상 차원(MSAS 2000)	지원
계층구조	지원
수준	지원
수준 속성	지원
특성(MSAS 2005)	지원
측정값	지원
계수 그룹(MSAS 2005)	지원
계산된 계수	지원
표시 폴더(MSAS 2005)	지원
KPI(MSAS 2005)	지원되지 않음
작업	지원되지 않음
데이터베이스 정렬 순서	Web Intelligence 에서 사용자 지정 정렬 순서를 정의해야 함
쓰기 되돌림	지원되지 않음

관련 정보

MSAS 큐브가 유니버스 구성 요소에 매핑되는 방식 [페이지 458]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.1.2.3 OLAP 유니버스에 대해 지원되는 Essbase 기능

다음 표에서는 Hyperion Essbase 데이터 소스에서 생성된 유니버스의 Essbase 기능에 대한 지원 수준에 대해 설명합니다.

Essbase 메타데이터 기능	OLAP 유니버스 지원 수준
블록 저장소 모드	지원
집계 저장소 모드	지원
하이브리드 모드	지원되지 않음
별칭 테이블	지원
차원	지원
특성 차원	지원
중복 멤버	지원
생성	지원
수준	지원되지 않음
UDA(사용자 정의 특성)	지원되지 않음
DTS(동적 시간 계열)	지원되지 않음
EIS(Essbase Integration Services) 드릴스루	지원되지 않음
대체 변수	지원되지 않음
연결된 파티션	지원되지 않음
LRO(Linked Reporting Objects)	지원되지 않음
데이터베이스 정렬 순서	Web Intelligence 에서 사용자 지정 정렬 순서를 정의해야 함
쓰기 되돌림	지원되지 않음

관련 정보

Essbase 큐브가 유니버스 구성 요소에 매핑되는 방식 [페이지 457]

8.2 OLAP 데이터 소스에 대한 연결 정의

8.2.1 OLAP 데이터 소스 연결 정보

OLAP 유니버스를 생성하려면 먼저 OLAP 데이터 소스에 대한 연결을 정의해야 합니다. 유니버스를 만드는 데 사용할 각 큐브 또는 쿼리에 대해 하나의 연결을 정의합니다.

새 연결 마법사를 사용하여 연결을 정의합니다. 마법사가 연결 생성을 위한 다음 단계를 안내합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스 디자인 도구에서 새 연결 마법사 시작
연결 이름 지정 및 데이터베이스 미들웨어 선택
연결에 대한 로그인 매개 변수 정의. 이러한 매개 변수는 선택한 데이터베이스 미들웨어에 따라 다릅니다.
유니버스를 만들 때 사용할 큐브 또는 쿼리 선택
연결 수명 정의
사용자 지정 매개 변수 정의 이러한 매개 변수는 선택한 데이터베이스 미들웨어에 따라 다릅니다.

연결 정의는 OLAP 유니버스 생성의 첫 단계입니다. 연결을 정의한 후에는 유니버스 디자인 도구가 자동으로 유니버스를 생성합니다.

> **노트**
>
> 도구 메뉴의 연결 목록에서 연결을 정의할 경우 별도의 단계로 유니버스를 만들어야 합니다.

관련 정보

새 연결 마법사 시작 [페이지 399]
OLAP 연결에 대한 데이터베이스 미들웨어를 선택하려면 [페이지 400]
SAP BW OLAP 연결에 대한 로그인 매개변수 [페이지 401]
MSAS OLAP 연결에 대한 로그인 매개 변수 [페이지 402]
Essbase 연결에 대한 로그인 매개 변수 정의 [페이지 402]
OLAP 연결에 대한 소스 큐브 또는 쿼리를 선택하려면 [페이지 403]
OLAP 연결에 대한 구성 매개 변수를 정의하려면 [페이지 404]
Essbase 연결에 대한 사용자 지정 매개 변수 정의 [페이지 404]

8.2.2 새 연결 마법사 시작

새 연결 마법사를 시작하려면 다음 작업 중 하나를 수행하십시오.

시작 위치...	수행 작업...
새 유니버스 아이콘	새 유니버스 아이콘을 클릭한 후 유니버스 매개 변수 상자의 정의 페이지에서 새로 만들기... 를 클릭합니다.
파일 메뉴	빈 세션에서 파일 > 매개 변수 를 선택한 다음 유니버스 매개 변수 상자의 정의 페이지에서 새로 만들기...를 클릭합니다.
빠른 디자인 마법사	빠른 디자인 마법사가 활성화된 경우 유니버스 디자인 도구를 시작할 때 자동으로 시작됩니다. 빠른 디자인 마법사 1 단계에서 새로 만들기...를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

시작 위치...	수행 작업...
	노트 마법사를 사용하지 않도록 설정한 경우 파일 > 새로 만들기 를 선택합니다. 파일 > 새로 만들기 를 선택해도 마법사가 시작되지 않으면 도구 > 옵션 을 선택합니다. 옵션 대화 상자의 일반 페이지에서 *[파일]에서 [새로 만들기]를 선택하면 빠른 디자인 마법사 시작* 확인란을 선택합니다. 확인을 클릭한 후 파일 > 새로 만들기 를 선택합니다.
도구 메뉴	도구 > 연결 을 선택합니다. 마법사 연결 대화 상자에서 추가... 를 클릭합니다.

8.2.3 OLAP 연결에 대한 데이터베이스 미들웨어를 선택하려면

새 연결 마법사의 데이터베이스 미들웨어 선택 페이지에서 연결 이름을 입력하고 연결 유형 및 연결에 대한 데이터베이스 미들웨어를 선택합니다.

데이터베이스 미들웨어 선택 매개 변수	설명
연결 유형	연결에 대한 액세스를 제어하려면 보안을 선택합니다(권장). 모든 사용자가 연결에 액세스할 수 있게 하려면 공유를 선택합니다. 유니버스 작성자로 액세스 권한을 제한하려면 개인을 선택합니다. 개인 연결을 사용하면 로컬 컴퓨터의 개인 데이터에만 액세스할 수 있습니다.
연결 이름	연결에 사용할 이름을 입력합니다.
필터 저장 프로시저 네트워크 계층	OLAP 연결에는 필터 저장 프로시저 네트워크 계층 매개 변수가 사용되지 않습니다.
사용 가능한 데이터 액세스 드라이버 목록	이 페이지에는 사용자의 데이터 액세스 드라이버 키에 해당하는 데이터베이스와 미들웨어가 나열됩니다. 대상 데이터베이스에 대한 노드를 확장하여 해당 데이터베이스에 대해 지원되는 미들웨어를 표시합니다. 미들웨어 노드를 확장하여 OLAP 미들웨어에 대한 Business Objects 데이터 액세스 드라이버를 표시합니다. 데이터 액세스 드라이버를 선택합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.2.4 SAP BW OLAP 연결에 대한 로그인 매개변수

새 연결 마법사의 로그인 매개 변수 대화 상자에는 다음 매개 변수가 포함될 수 있습니다.

표 183:

로그인 매개 변수	설명
인증 모드	• 지정된 사용자 이름 및 암호 사용: 로그인 세부 사항을 인증으로 사용합니다. • *Business Objects* 자격 증명 매핑 사용: 보고서를 새로 고칠 때 BusinessObjects 계정과 관련된 데이터베이스 사용자 암호를 입력하라는 메시지가 표시됩니다. 이 옵션은 *dbuser* 및 *dbpass* 매개 변수를 사용하여 설정합니다. 이러한 값은 관리자 수준에서 설정됩니다. 이 옵션 설정에 대한 내용은 *SAP Business Objects Enterprise* 관리자 가이드를 참조하십시오. • 보는 도중 보고서를 새로 고칠 때 단일 로그온 사용: 이 옵션을 선택하면 CMS 에 액세스하는 데 사용되는 사용자 이름 및 암호가 자동으로 데이터베이스 로그인 매개 변수로 사용됩니다. SSO(Single Sign-On) 설정에 대한 자세한 내용은 *Business Objects Enterprise* 관리자 가이드를 참조하십시오.
사용 가능한 경우 *SNC* 사용	SNC 를 사용하려면 이 확인란을 선택합니다.
클라이언트	SAP BW 시스템에서 클라이언트를 식별하는 데 사용되는 번호(필수)
사용자 이름	인증 모드가 지정된 사용자 이름 및 암호 사용인 경우 OLAP 서버에 액세스할 때 필요한 사용자 이름입니다.
암호	인증 모드가 지정된 사용자 이름 및 암호 사용인 경우 OLAP 서버에 액세스할 때 필요한 암호입니다.
언어	연결에 사용할 언어 **노트** 이 연결 언어로 유니버스가 생성됩니다.
언어 저장	연결에 사용할 언어: • 언어 저장을 선택할 경우 언어 필드의 값이 사용됩니다. • 언어 저장의 선택을 취소할 경우 사용자 세션의 값이 사용됩니다.
로그인 모드 또는 서버 유형	부하 분산을 사용하지 않고 SAP 서버에 직접 연결하려면 응용 프로그램 서버를 선택합니다. SAP 부하 분산 기능을 활용하려면 메시지 서버를 선택합니다.
응용 프로그램 서버	SAP 응용 프로그램 서버의 이름 또는 IP 주소를 선택하거나 입력합니다(응용 프로그램 서버 로그인 모드에 필요).
시스템 번호	시스템 번호(예: 00)를 입력합니다(응용 프로그램 서버 로그인 모드에 필요).
시스템 *ID*	메시지 서버 및 로그온 그룹을 입력하고, 메시지 서버 로그인 모드를 사용할 경우 필요에 따라 시스템 *ID* 를 입력합니다.
로그온 그룹	
메시지 서버	

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.2.5 MSAS OLAP 연결에 대한 로그인 매개 변수

새 연결 마법사의 로그인 매개 변수 대화 상자에는 다음 매개 변수가 포함될 수 있습니다.

표 184:

로그인 매개 변수	설명
인증 모드	• 지정된 사용자 이름 및 암호 사용: 로그인 세부 사항을 인증으로 사용합니다. • *Business Objects* 자격 증명 매핑 사용: 보고서를 새로 고칠 때 BusinessObjects 계정과 관련된 데이터베이스 사용자 암호를 입력하라는 메시지가 표시됩니다. 이 옵션은 *dbuser* 및 *dbpass* 매개 변수를 사용하여 설정합니다. 이러한 값은 관리자 수준에서 설정됩니다. 이 옵션 설정에 대한 내용은 *SAP Business Objects Business Intelligence* 플랫폼 관리자 가이드를 참조하십시오. • 보는 도중 보고서를 새로 고칠 때 단일 로그온 사용: 이 옵션을 선택하면 CMS 에 액세스하는 데 사용되는 사용자 이름 및 암호가 자동으로 데이터베이스 로그인 매개 변수로 사용됩니다. 단일 로그온(SSO) 설정에 대한 자세한 내용은 *SAP Business Objects Business Intelligence* 플랫폼 관리자 가이드를 참조하십시오.
서버	다음 중 하나를 입력합니다. • MSAS 서버에서 제공 및 구성하는 MSAS 라이브러리 관련 URL • MSAS 데이터 소스의 서버 이름 • MSAS 큐브 파일의 전체 경로 파일 이름 전체 경로 파일 이름을 다음과 같이 큰따옴표로 묶어 입력합니다. "Z:\All cubes\test.cub" **노트** 큐브 파일이 SAP BusinessObjects 호스트와 다른 호스트에 있는 경우 두 시스템에 공유 연결이 있어야 합니다. SAP BusinessObjects 호스트에서 직접 큐브 파일에 대한 연결을 만들어야 합니다.
사용자 이름	인증 모드가 지정된 사용자 이름 및 암호 사용인 경우 OLAP 서버에 액세스할 때 필요한 사용자 이름입니다.
암호	인증 모드가 지정된 사용자 이름 및 암호 사용인 경우 OLAP 서버에 액세스할 때 필요한 암호입니다.
언어	연결에 사용할 언어

8.2.6 Essbase 연결에 대한 로그인 매개 변수 정의

새 연결 마법사의 로그인 매개 변수 페이지에서 Essbase 데이터베이스에 연결하기 위한 로그인 세부 정보를 지정합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 185:

로그인 매개 변수	설명
인증 모드	사용자가 연결 사용 시 로그인 정보를 입력하도록 하려면 지정된 사용자 이름 및 암호 사용을 선택합니다. Essbase 보안을 BusinessObjects Enterprise 와 동기화하려면 사용자 이름 및 암호에 Essbase DBuser 및 DBpass 를 입력합니다. 연결 시 사용자의 BusinessObjects Enterprise 로그인 자격 증명을 사용하려면 *BusinessObjects* 자격 증명 매핑 사용을 선택합니다. **노트** Essbase 연결에는 보는 도중 보고서를 새로 고칠 때 단일 로그온 사용 옵션이 지원되지 않습니다.
사용자 이름	Essbase DBuser 를 입력합니다.
암호	Essbase DBpass 를 입력합니다.
서버	Essbase 서버 이름을 입력합니다.

8.2.7 OLAP 연결에 대한 소스 큐브 또는 쿼리를 선택하려면

큐브 브라우저에는 대상 서버에 사용할 수 있는 OLAP 큐브가 표시됩니다.

큐브 노드를 확장하여 사용 가능한 큐브 및 쿼리를 표시합니다. 브라우저에는 검색에 도움이 되는 다음과 같은 도구가 포함되어 있습니다.

표 186:

큐브 브라우저 도구	설명
즐겨찾기	빠른 액세스를 위해 선택한 큐브에 대한 링크가 포함된 폴더 즐겨찾기에 큐브를 추가하려면 OLAP 큐브 브라우저에서 큐브를 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 즐겨찾기에 추가를 선택합니다.
검색	사용 가능한 큐브 또는 쿼리 이름에서 텍스트 문자열을 검색합니다. 텍스트 상자에 문자열을 입력하고 검색을 클릭합니다. 찾은 각 인스턴스가 강조 표시됩니다. 검색을 계속하려면 검색을 클릭합니다.
$INFOCUBE 폴더	SAP BW 데이터 소스의 경우 인포큐브 및 멀티큐브가 $INFOCUBE 라는 폴더에 그룹화됩니다.

유니버스를 만들 때 사용할 큐브 또는 쿼리를 선택합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.2.8 OLAP 연결에 대한 구성 매개 변수를 정의하려면

새 연결 마법사의 구성 매개 변수 페이지에서 연결 수명을 관리하기 위한 연결 매개 변수를 정의합니다. 연결을 만들 때 기본 설정을 그대로 사용하고 나중에 수정할 수 있습니다.

표 187:

구성 매개 변수	설명
연결 풀 모드	연결 풀 모드 및 풀 제한 시간 매개 변수로 수명을 정의합니다. 기본적으로 연결 풀 모드는 다음 시간 동안 연결 유지로 설정되어 있으며 기본 풀 제한 시간은 10 분입니다. **노트** 기본 연결 수명을 그대로 사용할 것을 권장합니다. 각 트랜잭션 후에 연결이 끊어지도록 연결 풀 모드가 설정되면 유니버스 작성 속도가 크게 느려집니다. 또한 계층적 값 목록 작업 등의 주요 최종 사용자 워크플로에도 영향을 미칩니다. 연결 수명은 SAP BW 작업에 상당한 영향을 줄 수 있습니다. 하지만 연결 수명은 BEx 쿼리의 변경 내용으로 기존 유니버스를 업데이트하는 작업에도 영향을 미칠 수 있습니다. 이는 SAP BW 에 대한 연결이 설정될 때마다 OLAP BAPI 인터페이스가 클라이언트 쪽에 메타데이터 캐시를 작성하기 때문입니다. 이 캐시는 연결이 끊어질 때만 비워집니다. 메타데이터 캐시가 SAP BEx 쿼리 업데이트와 동기화되지 않는 위험을 최소화하기 위해 풀 제한 시간을 10 분에서 1 분으로 변경할 수 있습니다. BW 쿼리 편집 작업과 새 유니버스를 쿼리에 매핑하는 작업을 동시에 진행할 경우, BEx 쿼리 쪽에서 변경된 내용이 반영되도록 새 유니버스를 작성하기 전에 유니버스 디자인 도구를 닫는 것이 좋습니다. 이렇게 하면 유니버스 연결도 닫히고 메타데이터 캐시가 비워집니다.
풀 제한 시간	
배열 반입 크기	배열 반입 크기 매개 변수를 통해 FETCH 절차에 허용되는 최적 행 수를 설정할 수 있습니다.
배열 바인드 크기	배열 바인드 크기 및 로그인 시간 제한 매개 변수는 OLAP 연결에 사용되지 않습니다.
로그인 시간 제한	

8.2.9 Essbase 연결에 대한 사용자 지정 매개 변수 정의

새 연결 마법사의 사용자 지정 매개 변수 페이지에서 별칭 테이블을 지정하고 유니버스 생성 시 계수 차원으로 사용할 차원을 선택합니다.

표 188:

로그인 매개 변수	설명
별칭 테이블	기본 별칭 테이블 이외의 별칭 테이블에 유니버스를 생성하려면 목록에서 별칭 테이블을 선택합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

로그인 매개 변수	설명
계수 차원	계수 차원으로 사용할 차원을 선택합니다. 그러면 유니버스 디자인 도구가 계수로 선택된 차원의 멤버를 유니버스에 생성합니다.

8.3 OLAP 유니버스 사용자 지정

8.3.1 추가 매개 변수로 OLAP 유니버스 만들기

OLAP 유니버스에만 적용되는 이 기능을 사용하면 MSAS, SAP Business Warehouse(BW), Essbase 등으로 OLAP 유니버스를 만들 때 추가적인 메타데이터 매개 변수를 정의할 수 있습니다.

OLAP 유니버스를 만들 때 다음 매개 변수를 정의할 수 있습니다.

일반적인 OLAP 옵션	설명
기술 이름을 세부 정보로 생성	기술 이름을 유니버스에 있는 차원의 설명 개체로 생성하도록 응용 프로그램을 설정할 수 있습니다. 유니버스가 생성되면 기술 이름을 가리키는 설명 개체를 만듭니다.

SAP OLAP 옵션	설명
측정값 집계 위임 설정	위임된 데이터베이스에 계수의 집계 함수를 설정하도록 응용 프로그램을 설정할 수 있습니다.
접두사 L00, L01 바꾸기	유니버스 수준 접두사는 개체 계층구조 내의 수준을 나타냅니다. 수준 L00 은 최상위 수준 또는 루트 수준이고, L01 은 그 다음으로 낮은 수준입니다. 새 유니버스 마법사에서 OLAP 유니버스 수준 접두사를 다른 접두사로 바꿀 수 있습니다. 지정된 수준 번호는 유지되지만 접두사 'L'은 Level 등으로 바꿀 수 있습니다. 새 접두사 필드에 원하는 접두사를 입력합니다. 이 접두사는 OLAP 유니버스의 모든 수준 앞에 추가됩니다.
수준 00 을 '모두'로 이름 바꾸기	*수준 00 생성*이 아니요로 설정되어 있으면 이 옵션이 비활성화됩니다. 유니버스가 다음 번에 생성되면 최상위 수준(루트 수준) L00 의 이름을 '모두'로 바꿀 수 있습니다.
수준 00 생성	SAP 특성에만 적용되는 옵션입니다. 특성 및 계층구조에 대해 이 옵션을 비활성화할 수 있습니다. 수준 00 은 항상 계층구조 변수에 대해 생성됩니다. 유니버스를 생성하거나 업데이트할 때 수준 번호(L00, L01, L02...)를 다시 생성할 수 있습니다. 수준 번호는 수준 이름에 추가됩니다(예: "Monthly Sales_L01"). 이는 '모두' 수준을 사용하여 쿼리에 대한 결과를 집계하는 Web Intelligence 보고서에 유용합니다. Web Intelligence 보고서에 집계 필드를 만들 필요가 없기 때문입니다.

노트

"수준 00 생성" 옵션이 비활성화된 상태에서 유니버스가 만들어질 경우 계층구조에서 루트 수준이 생성되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.3.2 OLAP 유니버스에 대한 OLAP 옵션 정의

OLAP 옵션을 사용하여 OLAP 소스에서 특정 유니버스 메타데이터가 생성되는 방식을 정의할 수 있습니다. OLAP 옵션은 옵션 대화 상자의 *OLAP* 페이지(▶ 도구 > 옵션 > *OLAP* ◀)에서 선택할 수 있습니다. OLAP 소스의 모든 컨텐트가 추출되어 선택된 옵션에 따라 유니버스에 만들어집니다. 다음과 같은 OLAP 유니버스 생성 옵션을 선택할 수 있습니다.

일반적인 OLAP 옵션	설명
기술 이름을 세부 정보로 생성	유니버스를 생성할 때 유니버스의 기술 이름을 속성으로 생성하도록 응용 프로그램을 설정하여 기술 이름을 가리키는 개체를 만들 수 있습니다.

SAP OLAP 옵션	설명
측정값 집계 위임 설정	집계 함수를 사용하는 계수에 대해 위임된 계수를 생성하도록 응용 프로그램을 설정할 수 있습니다. 유니버스가 생성되면 집계 함수를 사용하는 모든 계수가 위임된 데이터베이스로 설정됩니다.
접두사 L00, L01 바꾸기	유니버스 수준 접두사는 개체 계층구조 내의 수준을 나타냅니다. 수준 L00 은 최상위 수준 또는 루트 수준이고, L01 은 그 다음으로 낮은 수준입니다. 새 유니버스 마법사에서 OLAP 유니버스 수준 접두사를 다른 접두사로 바꿀 수 있습니다. 지정된 수준 번호는 유지되지만 접두사 'L'은 Level 등으로 바꿀 수 있습니다. 새 접두사 필드에 원하는 접두사를 입력합니다. 이 접두사는 OLAP 유니버스의 모든 수준 앞에 추가됩니다.
수준 00 을 '모두'로 이름 바꾸기	'수준 00 생성'이 '아니요'로 설정되어 있으면 이 옵션은 비활성화됩니다. 유니버스가 다음 번에 생성되면 최상위 수준(루트 수준) L00 의 이름을 '모두'로 바꿀 수 있습니다. 이는 '모두' 수준을 사용하여 쿼리에 대한 결과를 집계하는 Web Intelligence 보고서에 유용합니다. Web Intelligence 보고서에 집계 필드를 만들 필요가 없기 때문입니다.
수준 00 생성	SAP 특성에만 적용되는 옵션입니다. 특성에 대해서만 이 옵션을 비활성화할 수 있습니다. 수준 00 은 항상 계층구조 및 계층구조 변수에 대해 생성됩니다. 유니버스를 생성하거나 업데이트할 때 수준 번호(L00, L01, L02...)를 다시 생성할 수 있습니다. 수준 번호는 수준 이름에 추가됩니다(예: "Monthly Sales_L01").

8.3.3 OLAP 유니버스의 개체 정의

SQL 편집기로 개체에 대한 Select 문이나 Where 절을 정의하고 OLAP 유니버스 개체에 대한 MDX 연산자 및 함수를 삽입할 수 있습니다. SQL 편집기에서 사용할 수 있는 옵션과 기능은 기본 데이터베이스에 따라 다릅니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.3.4 OLAP 유니버스에 대해 지원되는 유니버스 디자인 도구 기능

OLAP 유니버스는 자동으로 만들어집니다. OLAP 유니버스를 만든 후에는 유니버스 구성 요소를 수정할 수 있습니다.

생성된 OLAP 유니버스에 대해서는 다음과 같은 유니버스 디자인 도구 기능이 지원됩니다.

- 클래스 및 개체 숨기기, 복제 및 이름 바꾸기(차원, 설명 및 계수)
- 새 클래스 및 개체 삽입(차원, 설명 및 계수)
- 개체 형식 편집
- 개체 데이터 형식 편집
- 기본 및 외래 키 정의
- 차원, 설명 및 계수 개체 MDX 구문 분석
- 유니버스 무결성 검사
- 계층 편집
- 계단식 값 목록 만들기
- 값 목록에 대해 위임된 검색을 정의하여 사용자가 쿼리 실행 시에 값 목록 로드를 제한할 수 있도록 합니다.
- 변수의 기본값 사용
- 데이터베이스 위임 프로젝션 함수로 계수 정의(스마트 계수)
- 유니버스 구조 새로 고침

또한 다음 기능은 OLAP 유니버스에만 사용할 수 있습니다.

- 계산된 계수 만들기(SAP BW 및 MSAS 전용)
- 미리 정의된 조건 만들기
- 선택적 프롬프트 정의

OLAP 유니버스를 기반으로 하는 모든 개체는 인덱스 인식을 통해 생성됩니다. 개체의 계층구조에 중복된 값이 있으면 인덱스 인식은 값 목록에서 불일치를 제거합니다. 예를 들어, 한 계층구조에서 Paris 는 상위 France 항목 아래에 한 번, 그리고 상위 Texas 항목 아래에 한 번 등 총 두 번 나타날 수 있습니다. France 아래의 Paris 를 선택하면 Paris, France 에 대한 행만 반환됩니다.

OLAP 유니버스에 대해서는 다음과 같은 유니버스 디자인 도구 기능이 지원되지 않습니다.

- OLAP 유니버스에서는 행 수준 보안 권한 부여를 설정할 수 없습니다.
- OLAP 유니버스에서는 값 목록을 편집할 수 없습니다.
- OLAP 유니버스에 대해서는 스키마가 생성되지 않으므로 유니버스 엔터티-관계 스키마는 보거나 편집할 수 없습니다.

관련 정보

OLAP 유니버스의 계산된 계수 [페이지 411]
OLAP 유니버스의 미리 정의된 조건 [페이지 414]
OLAP 유니버스의 선택적 프롬프트 [페이지 418]
데이터베이스 위임 프로젝션 함수 [페이지 271]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.3.5 데이터베이스 위임 프로젝션 함수

유니버스에서 모든 계수는 프로젝션 함수(*Sum*, *Min*, *Max*, *Count* 및 *Avg*)를 포함할 수 있습니다. 프로젝션 함수는 보고서에 표시된 차원 수가 쿼리 결과 집합의 차원 수보다 작을 때 Web Intelligence 에서 로컬로 계수를 집계하는 데 사용됩니다.

비율, 평균 및 가중치와 같은 비가산 계수는 쿼리 결과 집합과 같은 집계 수준에만 표시될 수 있습니다. 따라서 유니버스에서 비가산 계수의 프로젝션 함수는 일반적으로 없음으로 설정되어 있습니다.

프로젝션 함수 데이터베이스 위임을 사용하여 비가산 계수의 집계를 데이터베이스 서버로 위임할 수 있습니다. 이 기능을 Web Intelligence 에서는 스마트 계수라고 합니다. 스마트 계수의 프로젝션 함수는 개체 속성의 속성 페이지에서 데이터베이스 위임으로 설정되어 있습니다. 이 기능 및 Web Intelligence 에 사용된 다른 기능에 대한 자세한 내용은 *Using Functions, Formulas and Calculations in Web Intelligence* 문서의 *Calculating values with Smart Measures* 단원을 참조하십시오.

노트

MSAS 및 Essbase 데이터 소스를 기반으로 하는 OLAP 유니버스의 경우 모든 계수는 기본적으로 프로젝션 함수가 데이터베이스 위임으로 설정된 유니버스에 만들어집니다.

노트

집계 인식 집합이 포함된 계수 기반의 스마트 계수를 사용할 경우 다음 제한에 대해 주의하십시오. 계수 정의에 사용된 집계 테이블의 데이터에 일관성이 있는지 확인하는 것이 좋습니다(세부 값 관련 집계 값이 정확). 그렇지 않으면 스마트 계수가 일관성이 없는 데이터를 만들 수 있습니다. 예를 들어, 연도 집계 테이블 및 일 집계 테이블이 특정 스마트 계수에 사용된 경우 연도 테이블이 현재 연도가 아닌 전체 연도에 대한 일 집계 테이블과 일치한다면 일 테이블의 데이터는 일 단위로 정확한 반면 연도 테이블이 비어있을 수 있습니다. 이 경우 현재 연도 및 일 단위 테이블 기반의 스마트 계수를 사용한 보고서의 경우 일관성이 없는 결과가 발생할 수 있습니다.

예

스마트 계수

이 예의 쿼리에는 두 개의 차원인 국가 및 지역과 세 개의 계수인 주문 수량, 배송 수량 및 배송 %이 포함되어 있습니다.

L01 지역	배송 수량	주문 수량	배송 %
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
		합계:	299.992

배송 %의 합계는 배송 % 열의 합계이므로 정확하지 않습니다.

유니버스에서 이 계수의 프로젝션 함수가 데이터베이스 위임으로 설정된 경우 사용자가 보고서를 새로 고치면 Web Intelligence 는 해당 데이터베이스에 연결하여 올바른 값을 계산합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

L01 지역	배송 수량	주문 수량	배송 %
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
		합계:	299.992
		합계:	99.997

ℹ 노트

비율 함수(Average) 등의 일부 함수를 사용할 때는 반드시 주의해야 합니다. 열 평균을 계산하는 경우 이 함수의 동작이 올바르게 구성되지 않으면 예기치 않은 결과가 발생할 수 있습니다.

예를 들어 SQL 함수 `sum(Shop_facts.Margin)/sum(Shop.facts.Quantity_sold)` 은 예기치 않은 결과가 발생할 수 있습니다. 올바르게 구성되지 않은 경우 각 셀에 대해 평균을 계산한 다음 해당 평균에 대한 합을 반환합니다. 이 동작을 수정하려면 다음과 같이 함수에 대한 매개변수화를 수행해야 합니다.

1. 함수에 대한 속성 편집 옵션으로 이동합니다.
2. 집계 시 이 계수를 처리할 방법을 선택하십시오 옵션에 대해 함수 드롭다운 목록에서 *Db delegated* 함수를 선택합니다.
3. 변경 사항을 저장합니다.

관련 정보

계수의 집계 프로젝션 설정 [페이지 270]

8.3.6 OLAP 유니버스에 대해 위임된 계수 설정

집계 함수를 사용하는 계수에 대해 위임된 계수를 생성하도록 응용 프로그램을 설정할 수 있습니다. 유니버스가 생성되면 집계 함수를 사용하는 모든 계수가 위임된 데이터베이스로 설정됩니다.

관련 정보

OLAP 유니버스 수준 접두사 바꾸기 [페이지 425]
OLAP 유니버스에 대해 수준 00 다시 생성 [페이지 425]
수준 00 을 '모두'로 이름 바꾸기 [페이지 425]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.3.7 계수의 집계 프로젝션 설정

계수를 만들 때는 집계 함수를 보고서에 프로젝션할 방법을 반드시 지정해야 합니다.

계수 개체의 반환 값은 쿼리 프로세스 중 두 단계에서 집계됩니다.

- 쿼리 단계. 유추된 SELECT 문을 사용하여 데이터가 집계됩니다.
- 마이크로큐브와 블록 사이의 단계. 마이크로큐브에서 보고서의 블록으로 데이터가 프로젝션될 때 집계됩니다. 이와 같은 계수 프로젝션 기능을 사용하면 마이크로큐브에 데이터를 로컬로 집계할 수 있습니다.

> **노트**
>
> 마이크로큐브는 쿼리에서 반환한 데이터를 보고서에 프로젝션하기 전에 표시하는 개념적인 방식입니다. 마이크로큐브는 Business Objects 보고서 작성 제품의 메모리에 있는 반환 값을 나타냅니다. 블록 수준은 사용자가 반환된 데이터로 만드는 2 차원 보고서입니다. 사용자는 마이크로큐브에 있는 데이터 전체나 일부만을 사용하여 보고서를 만들 수 있습니다. 또는 마이크로큐브에 있는 반환 값(로컬 집계)에 대해 집계 함수를 적용하여 보고서에 새 값을 만들 수 있습니다.

다음 그림은 쿼리 프로세스의 두 가지 집계 단계를 보여 줍니다.

- 사용자가 Web Intelligence 에서 쿼리를 작성합니다.
- Web Intelligence 에서 쿼리에 기반하여 SQL 을 유추하고 대상 데이터베이스에 SELECT 문을 보냅니다.
- 마이크로큐브에 데이터가 반환됩니다. 이것이 첫 번째 집계 단계입니다.
- 마이크로큐브에서 집계 데이터를 보고서에 프로젝션합니다. 쿼리 창에 데이터가 분할되어 세부적인 집계가 필요합니다. 이것이 두 번째 집계 단계입니다.

쿼리를 처음 실행하면 Select 문의 결과 집합이 마이크로큐브에 저장된 다음 마이크로큐브에 보관된 모든 데이터가 블록에 프로젝션됩니다. 이 경우 데이터는 마이크로큐브의 최하위 수준에서 프로젝션되기 때문에 프로젝션 집계가 수행되지 않습니다.

그러나 쿼리 창을 사용하여 마이크로큐브의 데이터 중 일부만 프로젝션하는 경우에는 상위 수준의 계수 값을 표시하기 위해 집계를 수행해야 합니다.

예를 들어, 위의 예제에서 연도 데이터를 블록으로 프로젝션하지 않으면 휴양지의 총 판매 수익을 표시하기 위해 연도와 관련된 세 개의 행을 단일 행으로 줄어야 하므로 합계 집계를 사용해야 합니다.

프로젝션 집계는 계수의 속성 편집 시트에 있는 속성 페이지(개체를 마우스 오른쪽 단추로 클릭 > 개체 속성 > 속성)에서 설정합니다.

프로젝션 집계는 SELECT 집계와 다릅니다.

관련 정보

데이터베이스 위임 프로젝션 함수 [페이지 271]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.3.8 OLAP 유니버스의 계산된 계수

유니버스에서 계산된 계수를 만들어 쿼리를 제한할 수 있습니다. OLAP 유니버스의 계산된 계수는 SQL 을 사용하지 않고 XML 태그에 포함된 MDX 함수를 사용하여 제한을 정의한다는 점을 제외하고 OLAP 이외의 유니버스에 있는 계수 개체에 대한 정의와 같습니다.

다음 OLAP 데이터 소스에 대해 계산된 계수를 사용할 수 있습니다.

- SAP Business Warehouse(BW)
- MSAS 2000 및 2005

계산된 계수는 필터나 WHERE 절에서 사용할 수 있습니다.

계산된 계수 식의 구문

계산된 계수 구문은 <EXPRESSION></EXPRESSION> 태그에 포함된 계산으로 구성됩니다.

계산된 계수 식에서는 다음과 같은 유니버스 디자인 도구 함수가 허용됩니다.

- @Select
- @Prompt
- @Variable
- @Where

> **노트**
>
> 계산된 계수의 식에는 @Aggregate_Aware 함수가 포함될 수 없습니다. 무결성 검사 함수는 MDX 문에 삽입된 구문을 비롯하여 XML 구문과 위에 나열된 @함수의 유효성을 검사합니다. 단, MDX 문은 구문 분석되지 않습니다.

이 식에는 "10" 또는 "ABC"와 같은 상수를 사용할 수 있습니다.

계산된 계수는 OLAP 메타데이터를 참조할 수 있습니다.

- 계수
- 차원
- 차원 수준
- MDX 식

계산된 계수 식에 대한 권장 사항

다음과 같은 이유로 계수 정의가 아닌 @Select(계수 이름)를 사용하십시오.

- @Select 는 쿼리 시에 확인됩니다.
- 계산된 계수는 @Select 함수 내에 있을 경우 다른 계산된 계수를 참조할 수 있습니다.
- @Select 함수 내의 개체에 대한 유효성을 검사합니다.

각 개체 정의의 인덱스 인식을 생성 및 설정하십시오.

해당 정의가 수준 또는 특성의 기술 이름 또는 고유 이름을 참조하는 개체나 설명에 대한 참조를 사용합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

예

계산된 계수 식

```
<EXPRESSION>@Select(Key Figures\Order Amount)*@Select(Key Figures\Order
Quantity)</EXPRESSION>
```

관련 정보

OLAP 유니버스에 계산된 계수 만들기 [페이지 412]

8.3.8.1 OLAP 유니버스에 계산된 계수 만들기

OLAP 유니버스에 계산된 계수를 만들려면

1. 유니버스 디자인 도구에서 OLAP 유니버스를 엽니다.
2. 유니버스에 새 계수 개체를 삽입합니다.
3. 위치: 상자에 개체 정의를 XML/MDX 식으로 입력하거나 붙여 넣습니다.
4. 구문 분석을 클릭하여 개체 정의를 검토하고 오류를 수정합니다.
5. 확인을 클릭하여 개체 정의를 저장합니다.
6. 도구 > 무결성 검사를 선택합니다.
   무결성 검사를 통해 XML 구문과 유니버스 디자인 도구 @FUNCTIONS 의 유효성이 검사됩니다.

관련 정보

OLAP 유니버스의 계산된 계수 [페이지 411]

8.3.9 큐브 쿼리의 MDX 함수

MDX 편집기를 사용하여 큐브 쿼리를 정의합니다.

OLAP 유니버스에 새 개체나 미리 정의된 필터를 추가할 때 특정 데이터 소스 연결에 대한 MDX 식 목록을 사용할 수 있습니다.

사용 가능한 식 라이브러리는 `.prm` 연결 파일에 저장됩니다. 개체의 속성 편집 창과 쿼리의 Select 문 편집 창을 열면 함수 창에 사용 가능한 식이 표시됩니다. SELECT 또는 WHERE 문에 식을 삽입하려면 문에서 식을 삽입할 위치를 클릭하고 삽입할 식을 더블 클릭합니다.

OLAP Universe MDX Dictionary - 함수 목록(PRM 파일)

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

OLAP 유니버스에 새 개체나 미리 정의된 필터를 추가할 때 식에서 사용할 수 있는 적절한 OLAP 연결(SAP 또는 MSAS)의 개체/필터 편집기에서 MDX 함수(주로 멤버 함수) 및 연산자의 명시적 목록을 사용할 수 있습니다. SAP 또는 mySQL(sap.prm, sqlsrv_as.prm) 연결을 설정하는 방법은 데이터 액세스 가이드를 참조하십시오. 유니버스의 연결 유형에 따라 사용 가능한 함수 및 연산자가 결정됩니다. 함수 목록은 각 연결에 대한 PRM 파일에서 제공합니다. PRM 파일에는 지원되는 함수 목록 중 가장 사용 빈도가 높은 함수만 표시됩니다.

다음 MDX 연산자를 쿼리에서 사용할 수 있습니다.

- Equal
- NotEqual
- InList
- NotInList
- Greater
- GreaterOrEqual
- Less
- LessOrEqual
- Between
- NotBetween
- Like
- NotLike

아래 목록에는 조건 편집 시 사용할 수 있는 몇 가지 MDX 폴더 함수가 나와 있습니다. 사용 가능한 함수는 기본 데이터베이스에 따라 결정됩니다.

- 집합 함수(ADDCALCULATEDMEMBERS, ALLMEMBERS ...)
- 통계/숫자 함수(AGGREGATE, AVG ...)
- 탐색/멤버 함수(ANCESTOR, ASCENDANTS...)
- 메타데이터 함수(AXIS, HIERARCHY...)

8.3.10 WHERE 절 및 필터에 대한 XML 구문

이 섹션에서는 OLAP 유니버스에서 필터 문이나 WHERE 절을 정의하는 XML 구문에 대해 설명합니다. FILTER 나 FILTER EXPRESSION 태그를 직접 추가한 후 태그 사이에 식을 수동으로 입력하거나 유니버스 디자인 도구 MDX 편집기를 사용하여 입력해야 합니다.

- 단일 개체 정의를 사용할 경우 `<FILTER= "your_object_definition">`을 사용합니다. 개체 정의는 큰따옴표 안에 입력합니다.
- 하나 이상의 개체가 포함된 복잡한 MDX 식을 사용할 경우 `<FILTER EXPRESSION= "yourcomplexMDX_expression">`을 사용합니다. 식은 큰따옴표 안에 입력합니다.

단일 필터 개체에 대한 구문은 다음과 같습니다.

```
<FILTER = “your_object_definition”><CONDITION
OPERATORCONDITION="yourOperator"><CONSTANT VALUE="your_Value"/></CONDITION></FILTER>
```

설명:

- yourMDX_expression 은 단일 개체 정의이며 큰따옴표로 묶입니다.
- `CONSTANT VALUE` 는 `CONSTANT CAPTION` 또는 `CONSTANT TECH_NAME` 입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- `yourOperator` 는 필터 식 연산자(`equals`, `inlist`...)입니다. `InIist` 연산자를 사용할 경우 목록의 각 항목에 대해 `CONSTANT CAPTION` 또는 `CONSTANT TECH_NAME` 요소를 삽입해야 합니다.
- `your_Value` 는 `CONSTANT CAPTION` 을 사용할 경우 정의된 필터 값이며 `CONSTANT TECH_NAME` 을 사용할 경우 개체 식별자입니다.

`InList` 연산자를 사용한 단일 필터 개체에 대한 구문입니다. 여기에는 세 개의 국가가 나열되며 다음과 같습니다.

```
<FILTER= "your_object_definition "><CONDITION OPERATORCONDITION="InList"><CONSTANT
CAPTION="England"/><CONSTANT CAPTION="France"/><CONSTANT CAPTION="Germany"/></
CONDITION></FILTER>
```

복잡한 필터 식에 대한 구문이며 필터링된 값에 대한 TECH_NAME 은 다음과 같습니다.

```
<FILTER EXPRESSION="yourComplex_MDX_Expression"><CONDITION
OPERATORCONDITION="Equal"><CONSTANT TECH_NAME="1"/></CONDITION></FILTER>
```

예

필터 식의 계산된 멤버로 필터링

```
<FILTER EXPRESSION="IIF([0CALYEAR].CurrentMember > “2000”, 1,0)"><CONDITION
OPERATORCONDITION="Equal"><CONSTANT CAPTION="1"/></CONDITION></FILTER>
```

8.3.11 OLAP 유니버스의 미리 정의된 조건

OLAP 유니버스의 미리 정의된 조건은 SQL 이 아닌 XML 을 사용하여 WHERE 절을 정의한다는 점을 제외하고 OLAP 가 아닌 유니버스의 조건과 같습니다. 수동으로 또는 미리 정의된 필터 편집기를 사용하여 필터를 선언할 수 있습니다.

8.3.11.1 미리 정의된 필터 옵션에 대한 XML 구문

미리 정의된 조건 구문

미리 정의된 단일 조건에는 AND 또는 OR 연산자로 결합된 여러 개의 필터가 포함될 수 있습니다. 기본적으로 모든 필터는 AND 연산자로 결합됩니다. OR 를 사용하여 필터를 포함시키려면 AND 및 OR 연산자 태그를 사용해야 합니다.

함수 @Select, @Prompt 및 @Variable 은 미리 정의된 필터 정의에서 허용됩니다.

미리 정의된 필터에는 하나 또는 여러 개의 프롬프트가 포함될 수 있습니다. 프롬프트는 필수 또는 선택적 프롬프트일 수 있습니다.

예

미리 정의된 조건에 대해 AND 및 OR 태그 사용

```
<OPERATOR VALUE="AND">
    <FILTER "[Level Object definition]">
        <CONDITION OPERATORCONDITION="Operator">
           <CONSTANT Level Attribute="Value"/>
        </CONDITION>
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

```
    </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 수동으로 OLAP 유니버스에서 미리 정의된 조건 만들기

미리 정의된 조건을 만들려면

1. 유니버스 디자인 도구에서 OLAP 유니버스를 열고 유니버스 창 아래쪽에 있는 조건 라디오 단추를 클릭합니다. 유니버스 창의 조건 뷰가 나타납니다. 이 뷰에는 유니버스의 클래스가 트리 뷰로 표시됩니다.
2. 클래스를 마우스 오른쪽 단추로 클릭하고 상황에 맞는 메뉴에서 조건...을 선택합니다.
3. 위치: 상자에서 XML 템플릿 필터를 편집합니다.

   템플릿 필터 형식은 다음과 같습니다.

```
<FILTER "[Level Object definition]">
    <CONDITION OPERATORCONDITION="Operator">
        <CONSTANT Level Attribute="Value"/
    </CONDITION>
</FILTER>
```

   템플릿의 요소를 다음과 같이 바꿉니다.

템플릿 요소:	가능한 값:
수준 개체 정의	필터에 적용된 차원 수준 또는 계수를 입력합니다. 개체 이름이 아닌 개체 정의를 입력합니다.
연산자	다음 중 하나를 입력합니다. ○ Equal ○ NotEqual ○ Greater ○ Less ○ GreaterOrEqual ○ LessOrEqual ○ Between ○ NotBetween ○ InList ○ NotInList ○ Like

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

템플릿 요소:	가능한 값:
	○ NotLike
수준 특성	다음 중 하나를 입력합니다. ○ NAME ○ CAPTION ○ TECH_NAME ○ DESCRIPTION
값	값 또는 프롬프트를 입력합니다. CONSTANT 태그당 하나의 값을 정의합니다.

미리 정의된 조건 편집의 예:

```
<FILTER KEY="[0D_DIV].[LEVEL01]">
    <CONDITION OPERATORCONDITION="InList">
        <CONSTANT CAPTION="Internal"/>
        <CONSTANT CAPTION="Service"/>
    </CONDITION>
</FILTER>
```

4. 구문 분석을 클릭하여 구문을 검토하고 오류를 수정합니다.
5. 확인을 클릭하여 조건을 저장합니다.

관련 정보

OLAP 유니버스의 미리 정의된 조건 [페이지 414]
OLAP 유니버스의 선택적 프롬프트 [페이지 418]

8.3.11.3 미리 정의된 필터 편집기 정보

미리 정의된 필터 편집기는 OLAP 유니버스에서 미리 정의된 필터를 편집하는 데 사용됩니다. 이 편집기를 사용하여 개체, 연산자, 값 목록, 프롬프트, 함수를 비롯하여 OLAP 유니버스에 대한 필터 정의에 사용되는 기타 옵션 요소를 선택할 수 있습니다.

필터의 조건 속성 패널에서 필터 식을 수동으로 입력하거나 >>을 클릭하여 미리 정의된 필터 편집기를 열 수 있습니다. 편집기가 열리면 필터 식에 `@Prompt` 를 삽입할 수 있습니다. 필터 식에서 적절한 지점을 마우스 오른쪽 단추로 클릭하고 바로 가기 메뉴에서 새 *@Prompt* 를 선택합니다. 미리 정의된 필터 편집기에서 쿼리/개체 정의에 필터 식을 삽입합니다.

예

국가 수준에서 고객 차원에 제한을 설정하여 국가를 캐나다로 제한

<FILTER KEY="[Customer].[Country].[Country]"> <CONDITION OPERATORCONDITION="Equal">
<CONSTANT CAPTION="Canada" /> </CONDITION> </FILTER>

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

미리 정의된 필터 편집기 옵션 정보 [페이지 417]
미리 정의된 필터 편집기를 사용하여 미리 정의된 필터 편집 [페이지 418]
큐브 쿼리의 MDX 함수 [페이지 264]

8.3.11.4 미리 정의된 필터 편집기 옵션 정보

미리 정의된 필터 편집기를 사용하여 OLAP 유니버스에 대한 유니버스 필터를 간편하게 정의할 수 있습니다. 다음과 같은 옵션을 정의할 수 있습니다.

옵션	설명
연산자 선택	사용 가능한 목록에서 연산자를 선택합니다. 기본값 = *Equal*
필터 기준	기존 유니버스 개체나 자유 정의(예: [계수].[인터넷 판매 금액])를 기준으로 필터를 설정합니다. 기본값 = *Universe 개체*
LoV 선택	기존 개체를 기준으로 필터가 설정되었을 경우 현재 유니버스에서 개체 목록을 선택합니다. 기본 선택 = 개체 목록의 루트 클래스
비교 값	개체/식을 비교할 값을 정의합니다. 선택된 연산자에 따라 하나 또는 두 개의 값 집합을 입력할 수 있습니다. 값은 정적이거나 프롬프트에 기반합니다. 기본값 = 정적 값
프롬프트 삽입	프롬프트를 수동으로 편집하거나 *@Prompt* 편집기를 사용합니다. *@Prompt* 편집기를 열려면 >>을 클릭합니다.
인덱스 인식 설정	인덱스 인식 함수를 사용합니다. 이 옵션이 제대로 작동하려면 기본 키가 선언되어야 합니다. 유니버스 디자인 도구에 인덱스 인식이 설정되면 기본 키 및 외래 키 열이 데이터 검색 속도를 높이고 유니버스 디자인 도구에서 보다 효과적인 SQL 필터를 생성하도록 하는 데 사용됩니다. 기본값 = 선택하지 않음
계산된 식 사용	이 옵션을 선택한 경우 <EXPRESSION> </EXPRESSION> 태그 사이의 필터 식이 포함됩니다. 기본값 = 선택하지 않음
선택적	현재 필터 식을 선택적으로 설정합니다. 필터 편집기의 현재 필터 식에만 적용되며 미리 정의된 조건 개체 전체에 대해 적용되지는 않습니다. 기본값 = 선택하지 않음

노트

Web Intelligence 의 미리 정의된 필터에는 "옵션" 태그를 사용할 수 없습니다. 이러한 태그는 사용되는 경우 쿼리의 필수 부분으로 처리되어 쿼리가 실행되지 않도록 합니다.

관련 정보

미리 정의된 필터 편집기를 사용하여 미리 정의된 필터 편집 [페이지 418]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.3.11.5 미리 정의된 필터 편집기를 사용하여 미리 정의된 필터 편집

OLAP 유니버스에서 필터를 편집합니다.

값을 선택하거나 입력하면 미리 정의된 필터 편집기가 업데이트됩니다. 필터 식을 마우스 오른쪽 단추로 클릭하여 필터 식에 `@Prompt` 식을 삽입할 수 있습니다. 마우스 오른쪽 단추를 클릭하고 새 *@Prompt* 를 선택합니다. 그러면 프롬프트 편집기가 열립니다.

1. 조건(필터) 창의 속성 창에서 >>을 클릭합니다.
   미리 정의된 필터 편집기가 표시됩니다.
2. 유니버스 개체를 기준으로 필터를 설정하려면 유니버스 개체를 선택하고 사용 가능한 개체 창에서 개체를 선택합니다. 고유한 식을 기준으로 미리 정의된 필터를 설정하려면 자유 정의를 선택하고 사용 가능한 개체 창에 식을 입력합니다.
3. 연산자 목록에서 연산자를 선택합니다. '목록에 있음' 및 '목록에 없음' 연산자에만 여러 값(오른쪽 피연산자)이 허용됩니다.
4. 고정 값을 하나 이상 정의하려면 정적 값을 선택하고 프롬프트 식을 삽입하려면 프롬프트를 선택합니다.
   프롬프트를 선택하면 편집 단추가 활성화됩니다. 편집을 클릭하여 *@Prompt* 편집기를 열고 필요에 따라 프롬프트 식을 정의합니다.
5. 확인을 클릭하여 필터 정의의 유효성을 검사합니다.
   파서가 무결성 검사를 포함하여 구문 오류를 검사합니다. 오류가 발견되면 오류 메시지가 포함된 경고 메시지가 표시됩니다. 오류가 발견되지 않으면 필터 정의가 있는 유니버스에 새로운 조건 개체가 추가됩니다.

관련 정보

미리 정의된 필터 편집기 옵션 정보 [페이지 417]
미리 정의된 필터 편집기 정보 [페이지 416]

8.3.12 OLAP 유니버스의 선택적 프롬프트

OLAP 데이터 소스에서 생성된 유니버스는 선택적 프롬프트를 지원합니다.

SAP BW 선택적 변수의 경우 선택적 조건이 있는 필터가 유니버스에서 자동으로 생성됩니다.

미리 정의된 조건이나 개체의 WHERE 절에서 프롬프트를 선택적 프롬프트로 만들려면 두 XML 태그 <OPTIONAL>과 </OPTIONAL> 사이에 XML 필터 식을 포함합니다.

예

미리 정의된 조건의 선택적 프롬프트

```
<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>
```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

```
</OPTIONAL>
```

관련 정보

수동으로 OLAP 유니버스에서 미리 정의된 조건 만들기 [페이지 415]

8.3.13 SAP BW 유니버스에 대한 특정 쿼리의 성능 향상

차원의 핵심 및 중간 이름 설명 개체만 포함하는 SAP BW 유니버스에 대한 쿼리의 경우 생성된 개체 구문을 수정하여 쿼리 성능을 높일 수 있습니다.

구문을 수정하려면

1. 유니버스 디자인 도구에서 유니버스를 엽니다.
2. 수정하려는 핵심 설명 개체를 두 번 클릭합니다.
3. 속성 편집 대화 상자의 정의 탭에 있는 선택 텍스트 상자에서 SAP 특징의 NAME 특성을 참조하도록 구문을 변경합니다.
   예를 들어 개체 *L01 Customer Key* 의 경우 생성된 다음 select 구문을

   ```
   [Z_CUSTOM].[LEVEL01].[[2Z_CUSTOM]].[Value]
   ```

   NAME 특성을 참조하도록 다음과 같이 변경하십시오.

   ```
   [Z_CUSTOM].[LEVEL01].[NAME]
   ```
4. 확인을 클릭하여 변경 내용을 저장합니다.
5. 이름 개체에 대해 같은 단계를 반복합니다. SAP 특징의 DESCRIPTION 특성을 참조하도록 구문을 변경하십시오.
   예를 들어 개체 *L01 Customer Medium Name* 의 경우 생성된 다음 select 구문을

   ```
   [Z_CUSTOM].[LEVEL01].[[5Z_CUSTOM]].[Value]
   ```

   DESCRIPTION 특성을 참조하도록 다음과 같이 변경하십시오.

   ```
   [Z_CUSTOM].[LEVEL01].[DESCRIPTION]
   ```


© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4 OLAP 유니버스 수명 주기 관리

8.4.1 OLAP 유니버스 수명 주기 관리 정보

> **노트**
>
> XIR3.1 SP2 이전 버전의 Universe Designer 로 만든 유니버스를 열 경우 유니버스나 OLAP 소스를 변경하기 전에 유니버스를 새로 고친 후 저장해야 합니다.

OLAP 유니버스는 OLAP 데이터 소스(예: SAP BEx 쿼리 또는 MSAS 2005 큐브)에서 자동으로 생성됩니다. 유니버스 디자인 도구에서 기존 OLAP 유니버스의 개체를 만들고 변경할 수 있습니다.

OLAP 유니버스 업데이트 마법사를 사용하면 OLAP 데이터 소스의 변경 내용으로 OLAP 유니버스의 구조를 자동으로 새로 고칠 수 있습니다. 마법사는 유니버스와 업데이트된 데이터 소스를 비교합니다. 마법사는 생성된 개체와 수동으로 추가 또는 수정된 개체를 구분할 수 있으므로 유니버스 디자인 도구에서 수동으로 변경한 내용을 보존할 수 있습니다. 유니버스 디자인 도구에서 수동으로 추가된 개체를 업데이트하지는 않습니다.

아래 표와 같이 검색 및 업데이트 가능한 대상은 항목 및 데이터 소스에 따라 다릅니다.

마법사에서 검색 가능한 대상	새 항목 검색 가능 위치	수정된 항목 검색 가능 위치	삭제된 항목 검색 가능 위치
차원	모든 데이터 소스	모든 데이터 소스	모든 데이터 소스
계층구조	SAP BW 및 MSAS 전용	모든 데이터 소스	모든 데이터 소스
수준	모든 데이터 소스	모든 데이터 소스	모든 데이터 소스
속성	MSAS 전용	MSAS 전용	MSAS 전용
계수	모든 데이터 소스	모든 데이터 소스	모든 데이터 소스
SAP BW 변수	SAP BW 전용	SAP BW 전용	SAP BW 전용
하위 클래스	모든 데이터 소스	모든 데이터 소스	모든 데이터 소스

> **노트**
>
> 버전 XIR3.1 SP2 이전의 Universe Designer 로 만들어진 유니버스를 업데이트하는 경우 차원 이름이 SAP 큐브에서 변경되었다면 차원을 새로 고칠 수 없으며 차원이 유니버스에서 중복됩니다. 또한 유니버스에서 클래스를 수동으로 업데이트해야 합니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
개요: 유니버스 개체 상태 및 OLAP 개체 상태 사이의 관계 [페이지 421]
OLAP 유니버스 업데이트 시 차원 관리 [페이지 426]
OLAP 유니버스 업데이트 시 계층구조 또는 특성 관리 [페이지 431]
OLAP 유니버스 업데이트 시 수준 관리 [페이지 437]
OLAP 유니버스 업데이트 시 SAP 변수 관리 [페이지 440]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

OLAP 유니버스 업데이트 시 주요 수치 또는 계수 관리 [페이지 443]
OLAP 유니버스 업데이트 시 SAP 주요 날짜 관리 방식 [페이지 447]

8.4.2 개요: 유니버스 개체 상태 및 OLAP 개체 상태 사이의 관계

아래 표에는 SAP OLAP 개체 상태와 유니버스 개체 상태 사이의 관계가 간략하게 나와 있습니다. 각 작업에 대한 자세한 설명은 이 장의 세부 단원을 참조하십시오.

표 189:

OLAP 메타데이터		유니버스 개체 상태				
		변경되지 않음	업데이트됨*	삭제됨	이동됨	숨김
차원		유니버스 등가 = 클래스				
상태	변경되지 않음	NoC	Upd	NoC	NoC	NoC
	업데이트됨*	Upd	Upd	NoC	Upd	Upd
	삭제됨	Del/Ob	Del/Ob	NoC	Del/Ob	NoC
	이동됨	Move	NoC	NoC	NoC	Move
	특성 생성됨	CreS	CreS	N/A	CreS	CreS
	생성됨	Cre	Cre	N/A	Cre	Cre
계층구조 또는 특성		유니버스 등가 = 하위 클래스				
상태	변경되지 않음	NoC	Upd	NoC	NoC	NoC
	업데이트됨*	Upd	Upd	NoC	Upd	Upd
	변경됨	UpdMDX	UpdMDX	NoC	UpdMDX	UpdMDX
	디스플레이 특성	Cre	Cre	Cre	Cre	Cre
	탐색 특성	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	삭제됨	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	이동됨	Move	Move	NoC	Move	Move
	신규	Cre	Cre	Cre	Cre	Cre
수준		유니버스 등가 = 수준				
상태	변경되지 않음	NoC	NoC	NoC	NoC	NoC
	업데이트됨*	Upd	Upd	NoC	Upd	Upd
	삭제됨	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	이동됨	Move	Move	NoC	Move	Move
	신규	Cre	Cre	Cre	Cre	Cre
변수		유니버스 등가 = 필터				
상태	변경되지 않음	NoC	NoC	NoC	NoC	NoC
	업데이트됨*	Upd	Upd	Cre	Upd	Upd

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

OLAP 메타데이터		유니버스 개체 상태				
		변경되지 않음	업데이트됨*	삭제됨	이동됨	숨김
	삭제됨	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	신규	Cre	Cre	Cre	Cre	Cre
주요 수치		유니버스 등가 = 계수				
상태	변경되지 않음	NoC	NoC	NoC	NoC	NoC
	업데이트됨*	Upd	Upd	NoC	Upd	Upd
	삭제됨	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	이동됨	Move	Move	NoC	Move	Move
	신규	Cre	Cre	Cre	Cre	Cre
주요 날짜		유니버스 등가 = 매개 변수				
상태	변경되지 않음	NoC	N/A	Cre	N/A	N/A
	삭제됨	Del	N/A	N/A	N/A	N/A
	신규	Cre	N/A	Cre	N/A	N/A

범례:

- *: 개체 속성(이름, 설명...) 중 하나가 변경됨
- Cre: 동등한 개체가 생성됨
- CreS: 동등한 하위 클래스 개체가 생성됨
- Del/Ob: 삭제되었거나 사용되지 않음(사용되지 않는 개체는 숨겨지며 이름 앞에 ## 표시)
- Move: 개체가 이동됨
- N/A: 해당 사항 없음
- NoC: 변경 사항 없음
- Upd: 업데이트됨
- UpdMDX: MDX 정의 업데이트

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
OLAP 유니버스 업데이트 시 차원 관리 [페이지 426]
OLAP 유니버스 업데이트 시 계층구조 또는 특성 관리 [페이지 431]
OLAP 유니버스 업데이트 시 수준 관리 [페이지 437]
OLAP 유니버스 업데이트 시 SAP 변수 관리 [페이지 440]
OLAP 유니버스 업데이트 시 주요 수치 또는 계수 관리 [페이지 443]
OLAP 유니버스 업데이트 시 SAP 주요 날짜 관리 방식 [페이지 447]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.3 OLAP 유니버스 새로 고침

OLAP 유니버스의 구조를 새로 고치려면 다음을 수행합니다.

- 유니버스 디자인 도구에서 업데이트할 유니버스를 엽니다.
- 보기 > 구조 새로 고침 을 선택합니다.
  OLAP 유니버스 업데이트 마법사가 나타납니다.
- 시작을 클릭합니다.

  업데이트된 메타데이터 정의 페이지가 나타납니다.

  수동으로 수행한 유니버스의 개체에 대한 변경 내용을 보존하려면 유지 옵션을 선택합니다. 모든 유지 옵션은 기본적으로 선택되어 있습니다. 다음 특성을 유지하도록 선택할 수 있습니다.

옵션	설명
비즈니스 이름 유지	클래스, 차원, 계수, 세부 정보 및 조건의 이름입니다.
개체 유형 유지	개체가 유니버스에서 변경된 경우(예: 세부 정보가 차원으로 변경됨), 업데이트를 하더라도 초기 개체 유형이 다시 생성되지 않습니다. 차원, 계수 및 세부 정보에 적용됩니다.
개체 설명 유지	이 옵션을 선택한 경우 OLAP 소스에서 설명이 업데이트되어도 이 정보가 유니버스에 업데이트되지 않습니다.
개체 데이터 유형 유지	문자, 숫자, 날짜 및 긴 텍스트 개체
개체의 값 옵션 목록 유지	초기에 설정된 옵션을 유지할 수 있습니다. ○ 값 목록 연결 ○ 사용하기 전에 자동으로 새로 고침 ○ 계층구조 표시 ○ 유니버스와 함께 내보내기 ○ 위임 검색
개체의 고급 옵션 유지	사용할 수 있는 옵션은 다음과 같습니다. 보안 액세스 수준 개체는 다음과 같이 사용할 수 있습니다. ○ 결과에서 사용 ○ 조건에서 사용 ○ 정렬에서 사용
사용되지 않는 개체 삭제	데이터 소스에 더 이상 없는 항목이 유니버스에서 삭제됩니다.
사용되지 않는 개체 숨기기	큐브에 더 이상 없는 항목이 유니버스에서 숨겨지며 접두사 /##/으로 표시됩니다.

- 원하는 옵션을 선택하고 다음을 클릭합니다.

  변경 관리 결과 페이지가 나타나며 추가/삭제/숨겨진 개체를 표시합니다. 숨겨진 개체는 유니버스의 별도 클래스로 이동되며 맨 앞에 /##/이 붙고 글꼴이 기울임꼴로 표시됩니다.
- 추가된 메타데이터 옵션 창에서 추가된 메타데이터에 적용할 옵션을 설정하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

일반적인 OLAP 옵션	설명
기술 이름을 세부 정보로 생성	유니버스를 생성할 때 유니버스의 기술 이름을 속성으로 생성하도록 응용 프로그램을 설정하여 기술 이름을 가리키는 개체를 만들 수 있습니다.
수동으로 삭제된 모든 개체 다시 생성	수동으로 삭제된 모든 유니버스 개체가 다시 생성됩니다.

SAP OLAP 옵션	설명
측정값 집계 위임 설정	위임된 데이터베이스에 계수의 집계 함수를 설정하도록 응용 프로그램을 설정할 수 있습니다.
접두사 L00, L01 바꾸기	유니버스 수준 접두사는 개체 계층구조 내의 수준을 나타냅니다. 수준 L00 은 최상위 수준 또는 루트 수준이고, L01 은 그 다음으로 낮은 수준입니다. 새 유니버스 마법사에서 OLAP 유니버스 수준 접두사를 다른 접두사로 바꿀 수 있습니다. 지정된 수준 번호는 유지되지만 접두사 'L'은 Level 등으로 바꿀 수 있습니다. 새 접두사 필드에 원하는 접두사를 입력합니다. 이 접두사는 OLAP 유니버스의 모든 수준 앞에 추가됩니다.
수준 00 을 '모두'로 이름 바꾸기	수준 *00* 생성이 아니요로 설정되어 있으면 이 옵션이 비활성화됩니다. 유니버스가 다음 번에 생성되면 최상위 수준(루트 수준) L00 의 이름을 '모두'로 바꿀 수 있습니다.
수준 00 생성	SAP 특성에만 적용되는 옵션입니다. 특성에 대해서만 이 옵션을 비활성화할 수 있습니다. 수준 00 은 항상 계층구조 및 계층구조 변수에 대해 생성됩니다. 유니버스를 생성하거나 업데이트할 때 수준 번호(L00, L01, L02...)를 다시 생성할 수 있습니다. 수준 번호는 수준 이름에 추가됩니다(예: "Monthly Sales_L01"). 이는 '모두' 수준을 사용하여 쿼리에 대한 결과를 집계하는 Web Intelligence 보고서에 유용합니다. Web Intelligence 보고서에 집계 필드를 만들 필요가 없기 때문입니다.

- 변경 관리 결과 페이지에서 다음 중 하나를 선택합니다.

옵션	설명
확인	결과가 만족스럽지 않으면 확인을 클릭하여 유니버스를 저장하거나 내보내지 않고 닫습니다.
내보내기	변경 내용이 만족스러울 경우 내보내기를 클릭하여 업데이트된 유니버스를 저장한 후 CMC 로 내보냅니다.
무결성 검사	무결성 검사를 수행하려면 무결성 검사를 클릭합니다. 이 기능은 구조를 검사하고, 개체를 구문 분석하고, 조인을 구문 분석하고, 조건을 구문 분석하고, 카디널리티를 검사합니다. 검사가 완료되면 무결성 검사 결과 페이지가 나타납니다. 이 페이지에서 검사 결과를 인쇄할 수 있습니다.

예상한 유니버스의 모든 변경 내용이 표시되지 않으면 유니버스 디자인 도구를 중지한 후 다시 시작한 다음 업데이트를 다시 시도하십시오. 그러면 데이터 소스에 대한 새 연결이 생성되고 캐시는 지워집니다.

관련 정보

유니버스와 OLAP 큐브 동기화 [페이지 426]
OLAP 유니버스 수명 주기 관리 정보 [페이지 420]

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.4 OLAP 유니버스에 대해 수준 00 다시 생성

유니버스를 생성하거나 업데이트할 때 수준 번호(L00, L01, L02...)를 다시 생성할 수 있습니다. 수준 번호는 수준 이름에 추가됩니다(예: "Monthly Sales_L01").

관련 정보

OLAP 유니버스에 대해 위임된 계수 설정 [페이지 409]
OLAP 유니버스 수준 접두사 바꾸기 [페이지 425]
수준 00 을 '모두'로 이름 바꾸기 [페이지 425]

8.4.5 수준 00 을 '모두'로 이름 바꾸기

유니버스가 다음 번에 생성되면 최상위 수준(루트 수준) L00 의 이름을 '모두'로 바꿀 수 있습니다. 이는 '모두' 수준을 사용하여 쿼리에 대한 결과를 집계하는 SAP BusinessObjects Web Intelligence 보고서에 유용합니다. Web Intelligence 보고서에 집계 필드를 만들 필요가 없기 때문입니다.

관련 정보

OLAP 유니버스에 대해 위임된 계수 설정 [페이지 409]
OLAP 유니버스 수준 접두사 바꾸기 [페이지 425]
OLAP 유니버스에 대해 수준 00 다시 생성 [페이지 425]

8.4.6 OLAP 유니버스 수준 접두사 바꾸기

유니버스 수준 접두사는 개체 계층구조 내의 수준을 나타냅니다. 수준 L00 은 최상위 수준 또는 루트 수준이고, L01 은 그 다음으로 낮은 수준입니다. 새 유니버스 마법사에서 OLAP 유니버스 수준 접두사를 다른 접두사로 바꿀 수 있습니다. 지정된 수준 번호는 유지되지만 접두사 'L'은 Level 등으로 바꿀 수 있습니다. 새 접두사 필드에 원하는 접두사를 입력합니다. 이 접두사는 OLAP 유니버스의 모든 수준 앞에 추가됩니다.

관련 정보

OLAP 유니버스에 대해 위임된 계수 설정 [페이지 409]
OLAP 유니버스에 대해 수준 00 다시 생성 [페이지 425]
수준 00 을 '모두'로 이름 바꾸기 [페이지 425]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.7 유니버스와 OLAP 큐브 동기화

유니버스를 업데이트하면 유니버스에 포함된 개체가 OLAP 큐브에 포함된 개체와 비교되어 큐브의 변경 내용이 유니버스에 부정적인 영향을 미치지 않도록 합니다. 다시 말해 유니버스에서 사용되는 모든 개체(삭제된 개체 포함)가 항상 사용 가능해야 한다는 의미입니다. OLAP 큐브에 새로 추가된 개체는 모두 유니버스에 사용할 수 있습니다. 다양한 개체들이 변경 내용에 어떤 영향을 받는지는 아래 링크를 참조하십시오.

개체 속성이 업데이트되면 유니버스에 특정 속성만 업데이트되고 다른 속성은 변경되지 않습니다. 다음 테이블과 같은 결과가 나타납니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
OLAP 유니버스 업데이트 시 차원 관리 [페이지 426]
OLAP 유니버스 업데이트 시 계층구조 또는 특성 관리 [페이지 431]
OLAP 유니버스 업데이트 시 수준 관리 [페이지 437]
OLAP 유니버스 업데이트 시 SAP 변수 관리 [페이지 440]
OLAP 유니버스 업데이트 시 주요 수치 또는 계수 관리 [페이지 443]
OLAP 유니버스 업데이트 시 SAP 주요 날짜 관리 방식 [페이지 447]

8.4.8 OLAP 유니버스 업데이트 시 차원 관리

SAP, MSAS 및 Essbase 데이터 소스에 적용되는 내용입니다. 유니버스 클래스는 OLAP 차원과 같습니다. OLAP 개체와 관련하여 유니버스 개체가 관리되는 방식은 변경 유형에 따라 결정됩니다. 특정 OLAP 개체 변경이 유니버스 개체에 미치는 영향에 대해 알아보려면 아래 나열된 항목을 참조하십시오.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
차원이 변경되지 않은 경우 [페이지 427]
차원이 업데이트된 경우(이름, 설명) [페이지 427]
차원이 삭제된 경우 [페이지 428]
차원이 이동된 경우 [페이지 429]
계층구조 또는 특성이 생성된 경우 [페이지 430]
새 차원이 만들어진 경우 [페이지 430]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.8.1 차원이 변경되지 않은 경우

아래 표에는 차원이 변경되지 않았을 때 동등한 유니버스 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 190:

유니버스 클래스의 상태	유니버스 클래스에 미치는 영향
변경 안 됨	유니버스 클래스가 변경되지 않습니다.
업데이트됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
삭제됨	유니버스 클래스가 변경되지 않습니다. 수동으로 삭제된 개체 다시 생성 옵션이 선택된 경우 개체가 생성됩니다. 삭제되지 않은 하위는 다시 생성되지 않습니다.
이동됨	유니버스 클래스가 변경되지 않습니다.
숨겨짐	유니버스 클래스가 변경되지 않습니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
차원이 업데이트된 경우(이름, 설명) [페이지 427]
차원이 삭제된 경우 [페이지 428]
차원이 이동된 경우 [페이지 429]
계층구조 또는 특성이 생성된 경우 [페이지 430]
새 차원이 만들어진 경우 [페이지 430]

8.4.8.2 차원이 업데이트된 경우(이름, 설명)

아래 표에는 차원의 이름 또는 설명이 업데이트되었을 때 동등한 유니버스 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 191:

유니버스 클래스의 상태	유니버스 클래스에 미치는 영향
변경 안 됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스 클래스의 상태	유니버스 클래스에 미치는 영향
업데이트됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
삭제됨	유니버스 클래스가 변경되지 않습니다. 수동으로 삭제된 개체 다시 생성 옵션이 선택된 경우 생성됩니다. 삭제되지 않은 하위는 다시 생성되지 않습니다.
이동됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
숨겨짐	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
차원이 변경되지 않은 경우 [페이지 427]
차원이 삭제된 경우 [페이지 428]
차원이 이동된 경우 [페이지 429]
계층구조 또는 특성이 생성된 경우 [페이지 430]
새 차원이 만들어진 경우 [페이지 430]

8.4.8.3 차원이 삭제된 경우

아래 표에는 차원이 삭제되었을 때 동등한 유니버스 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 192:

유니버스 클래스의 상태	유니버스 클래스에 미치는 영향
변경 안 됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다. 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
업데이트됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다. 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
삭제됨	유니버스 클래스가 변경되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스 클래스의 상태	유니버스 클래스에 미치는 영향
이동됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다. 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
숨겨짐	유니버스 클래스가 변경되지 않습니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
차원이 변경되지 않은 경우 [페이지 427]
차원이 업데이트된 경우(이름, 설명) [페이지 427]
차원이 이동된 경우 [페이지 429]
계층구조 또는 특성이 생성된 경우 [페이지 430]
새 차원이 만들어진 경우 [페이지 430]

8.4.8.4 차원이 이동된 경우

아래 표에는 차원이 이동되었을 때 동등한 유니버스 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 193:

유니버스 클래스의 상태	유니버스 클래스에 미치는 영향
변경 안 됨	클래스가 그에 따라 이동됩니다.
업데이트됨	변경 내용 없음
삭제됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다. 삭제되지 않은 하위는 다시 생성되지 않습니다.
이동됨	변경 내용 없음
숨겨짐	클래스가 그에 따라 이동됩니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
차원이 변경되지 않은 경우 [페이지 427]
차원이 업데이트된 경우(이름, 설명) [페이지 427]
차원이 삭제된 경우 [페이지 428]
계층구조 또는 특성이 생성된 경우 [페이지 430]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

새 차원이 만들어진 경우 [페이지 430]

8.4.8.5 계층구조 또는 특성이 생성된 경우

계층구조는 MSAS 또는 Essbase 데이터 소스에 적용되고 특성은 SAP 데이터 소스에 적용됩니다. 아래 표에는 SAP 특성이 생성되었을 때 동등한 유니버스 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 194:

유니버스 클래스의 상태	유니버스 클래스에 미치는 영향
변경 안 됨	하위 클래스가 생성됩니다.
업데이트됨	하위 클래스가 생성됩니다.
삭제됨	해당되지 않습니다.
이동됨	하위 클래스가 생성됩니다.
숨겨짐	하위 클래스가 생성됩니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
차원이 변경되지 않은 경우 [페이지 427]
차원이 업데이트된 경우(이름, 설명) [페이지 427]
차원이 삭제된 경우 [페이지 428]
차원이 이동된 경우 [페이지 429]
새 차원이 만들어진 경우 [페이지 430]

8.4.8.6 새 차원이 만들어진 경우

새 차원이 만들어지면 유니버스 클래스가 만들어집니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
차원이 변경되지 않은 경우 [페이지 427]
차원이 업데이트된 경우(이름, 설명) [페이지 427]
차원이 삭제된 경우 [페이지 428]
차원이 이동된 경우 [페이지 429]
계층구조 또는 특성이 생성된 경우 [페이지 430]

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.9 OLAP 유니버스 업데이트 시 계층구조 또는 특성 관리

이 단원의 내용은 MSAS 및 Essbase 데이터 소스의 계층구조와 SAP 데이터 소스의 특성에 적용됩니다. 유니버스 하위 클래스는 OLAP 특성과 같습니다. OLAP 개체와 관련하여 유니버스 개체가 관리되는 방식은 변경 유형에 따라 결정됩니다. 특정 OLAP 개체 변경이 유니버스 개체에 미치는 영향에 대해 알아보려면 아래 나열된 항목을 참조하십시오.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
계층구조 또는 특성이 변경되지 않은 경우 [페이지 431]
특성 비즈니스 이름 또는 설명이 업데이트된 경우 [페이지 432]
특성의 활성 계층구조가 변경된 경우 [페이지 433]
특성 표시 속성이 탐색 속성으로 변경된 경우 [페이지 433]
계층구조 또는 특성이 삭제된 경우 [페이지 435]
계층구조 또는 특성이 이동된 경우 [페이지 436]
새 계층구조 또는 특성이 만들어진 경우 [페이지 436]

8.4.9.1 계층구조 또는 특성이 변경되지 않은 경우

아래 표에는 계층구조 또는 특성이 변경되지 않았을 때 동등한 유니버스 하위 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 195:

유니버스 하위 클래스의 상태	유니버스 하위 클래스에 미치는 영향
변경 안 됨	변경 내용 없음
업데이트됨	변경 내용 없음
삭제됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다. 삭제되지 않은 하위 수준은 다시 생성되지 않습니다.
이동됨	변경 내용 없음
숨겨짐	변경 내용 없음

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
특성 비즈니스 이름 또는 설명이 업데이트된 경우 [페이지 432]
특성의 활성 계층구조가 변경된 경우 [페이지 433]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

특성 표시 속성이 탐색 속성으로 변경된 경우 [페이지 433]
계층구조 또는 특성이 삭제된 경우 [페이지 435]
계층구조 또는 특성이 이동된 경우 [페이지 436]
새 계층구조 또는 특성이 만들어진 경우 [페이지 436]

8.4.9.2 특성 비즈니스 이름 또는 설명이 업데이트된 경우

아래 표에는 특성이 업데이트되었을 때 동등한 유니버스 하위 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 196:

유니버스 하위 클래스의 상태	유니버스 하위 클래스에 미치는 영향
변경 안 됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
업데이트됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
삭제됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다. 삭제되지 않은 하위 수준은 다시 생성되지 않습니다.
이동됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되지 않은 경우 변경되지 않습니다.
숨겨짐	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
계층구조 또는 특성이 변경되지 않은 경우 [페이지 431]
특성의 활성 계층구조가 변경된 경우 [페이지 433]
특성 표시 속성이 탐색 속성으로 변경된 경우 [페이지 433]
계층구조 또는 특성이 삭제된 경우 [페이지 435]
계층구조 또는 특성이 이동된 경우 [페이지 436]
새 계층구조 또는 특성이 만들어진 경우 [페이지 436]

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.9.3 특성의 활성 계층구조가 변경된 경우

SAP 데이터 소스에만 적용됩니다. 아래 표에는 특성의 활성 계층구조가 변경되었을 때 동등한 유니버스 하위 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 197:

유니버스 하위 클래스의 상태	유니버스 하위 클래스에 미치는 영향
변경 안 됨	새 활성 계층구조를 참조하도록 하위 클래스에 있는 기존 개체에 대한 MDX 정의가 업데이트됩니다. 새로 고침이 계속되기 전에 보고서가 작성됩니다.
업데이트됨	새 활성 계층구조를 참조하도록 하위 클래스에 있는 기존 개체에 대한 MDX 정의가 업데이트됩니다. 새로 고침이 계속되기 전에 보고서가 작성됩니다.
삭제됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다. 삭제되지 않은 하위 수준은 다시 생성되지 않습니다.
이동됨	새 활성 계층구조를 참조하도록 하위 클래스에 있는 기존 개체에 대한 MDX 정의가 업데이트됩니다. 새로 고침이 계속되기 전에 보고서가 작성됩니다.
숨겨짐	새 활성 계층구조를 참조하도록 하위 클래스에 있는 기존 개체에 대한 MDX 정의가 업데이트됩니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
계층구조 또는 특성이 변경되지 않은 경우 [페이지 431]
특성 비즈니스 이름 또는 설명이 업데이트된 경우 [페이지 432]
특성 표시 속성이 탐색 속성으로 변경된 경우 [페이지 433]
계층구조 또는 특성이 삭제된 경우 [페이지 435]
계층구조 또는 특성이 이동된 경우 [페이지 436]
새 계층구조 또는 특성이 만들어진 경우 [페이지 436]

8.4.9.4 특성 표시 속성이 탐색 속성으로 변경된 경우

SAP 데이터 소스에만 적용됩니다. 아래 표에는 특성 표시 속성이 탐색 속성으로 변경되었을 때 동등한 유니버스 하위 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 198:

유니버스 하위 클래스의 상태	유니버스 하위 클래스에 미치는 영향
변경 안 됨	만들기
업데이트됨	만들기
삭제됨	만들기
이동됨	만들기
숨겨짐	만들기

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
계층구조 또는 특성이 변경되지 않은 경우 [페이지 431]
특성 비즈니스 이름 또는 설명이 업데이트된 경우 [페이지 432]
특성의 활성 계층구조가 변경된 경우 [페이지 433]
계층구조 또는 특성이 삭제된 경우 [페이지 435]
계층구조 또는 특성이 이동된 경우 [페이지 436]
새 계층구조 또는 특성이 만들어진 경우 [페이지 436]

8.4.9.5 특성 탐색 속성이 표시 속성으로 변경된 경우

SAP 데이터 소스에만 적용됩니다. 아래 표에는 계층구조 또는 특성 탐색 속성이 표시 속성으로 변경되었을 때 동등한 유니버스 하위 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 199:

유니버스 하위 클래스의 상태	유니버스 하위 클래스에 미치는 영향
변경 안 됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스를 숨기도록 설정됩니다. 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
업데이트됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스를 숨기도록 설정됩니다. 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
삭제됨	변경 내용 없음
이동됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스를 숨기도록 설정됩니다. 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
숨겨짐	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스를 숨기도록 설정됩니다. 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
계층구조 또는 특성이 변경되지 않은 경우 [페이지 431]
특성 비즈니스 이름 또는 설명이 업데이트된 경우 [페이지 432]
특성의 활성 계층구조가 변경된 경우 [페이지 433]
계층구조 또는 특성이 삭제된 경우 [페이지 435]
계층구조 또는 특성이 이동된 경우 [페이지 436]
새 계층구조 또는 특성이 만들어진 경우 [페이지 436]

8.4.9.6 계층구조 또는 특성이 삭제된 경우

아래 표에는 계층구조 또는 특성이 삭제되었을 때 동등한 유니버스 하위 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 200:

유니버스 하위 클래스의 상태	유니버스 하위 클래스에 미치는 영향
변경 안 됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다. 하위 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
업데이트됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제되고, 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다. 하위 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
삭제됨	변경 내용 없음
이동됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제되고, 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다. 하위 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.
숨겨짐	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제되고, 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다. 하위 클래스에 사용자 지정 개체가 포함된 경우 삭제되지 않습니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
계층구조 또는 특성이 변경되지 않은 경우 [페이지 431]
특성 비즈니스 이름 또는 설명이 업데이트된 경우 [페이지 432]
특성의 활성 계층구조가 변경된 경우 [페이지 433]
특성 표시 속성이 탐색 속성으로 변경된 경우 [페이지 433]
계층구조 또는 특성이 이동된 경우 [페이지 436]
새 계층구조 또는 특성이 만들어진 경우 [페이지 436]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.9.7 계층구조 또는 특성이 이동된 경우

특성이 동일한 차원 내에서 이동된 경우에는 아무것도 변경되지 않습니다. 아래 표를 무시하십시오. 아래 표에는 계층구조 또는 특성이 다른 차원으로 이동되었을 때 동등한 유니버스 하위 클래스에 미치는 영향이 상태별로 설명되어 있습니다.

표 201:

유니버스 하위 클래스의 상태	유니버스 하위 클래스에 미치는 영향
변경 안 됨	하위 클래스가 그에 따라 이동됩니다.
업데이트됨	하위 클래스가 그에 따라 이동됩니다.
삭제됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다. 삭제되지 않은 하위 수준은 다시 생성되지 않습니다.
이동됨	변경 내용 없음
숨겨짐	하위 클래스가 그에 따라 이동됩니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
계층구조 또는 특성이 변경되지 않은 경우 [페이지 431]
특성 비즈니스 이름 또는 설명이 업데이트된 경우 [페이지 432]
특성의 활성 계층구조가 변경된 경우 [페이지 433]
특성 표시 속성이 탐색 속성으로 변경된 경우 [페이지 433]
계층구조 또는 특성이 삭제된 경우 [페이지 435]
새 계층구조 또는 특성이 만들어진 경우 [페이지 436]

8.4.9.8 새 계층구조 또는 특성이 만들어진 경우

새 계층구조 또는 특성이 만들어지면 유니버스 하위 클래스가 만들어집니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
계층구조 또는 특성이 변경되지 않은 경우 [페이지 431]
특성 비즈니스 이름 또는 설명이 업데이트된 경우 [페이지 432]
특성의 활성 계층구조가 변경된 경우 [페이지 433]
특성 표시 속성이 탐색 속성으로 변경된 경우 [페이지 433]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

계층구조 또는 특성이 삭제된 경우 [페이지 435]
계층구조 또는 특성이 이동된 경우 [페이지 436]

8.4.10 OLAP 유니버스 업데이트 시 수준 관리

> **노트**
>
> 유니버스에서 수준을 다른 계층구조로 이동하지 마십시오. 수준을 이동하려면 새 계층구조로 수준을 복사하여 붙여 넣으십시오.

유니버스 수준 또는 차원 개체는 OLAP 수준과 같습니다. OLAP 개체와 관련하여 유니버스 개체가 관리되는 방식은 변경 유형에 따라 결정됩니다. 특정 OLAP 개체 변경이 유니버스 개체에 미치는 영향에 대해 알아보려면 아래 나열된 항목을 참조하십시오.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
수준이 변경되지 않은 경우 [페이지 437]
수준의 이름 또는 설명이 업데이트된 경우 [페이지 438]
수준이 삭제된 경우 [페이지 439]
수준이 이동된 경우 [페이지 439]
새 수준이 만들어진 경우 [페이지 440]

8.4.10.1 수준이 변경되지 않은 경우

아래 표에는 수준이 변경되지 않았을 때 유니버스 수준에 미치는 영향이 상태별로 설명되어 있습니다.

표 202:

유니버스 수준의 상태	유니버스 수준에 미치는 영향
변경 안 됨	변경 내용 없음
업데이트됨	변경 내용 없음
삭제됨	변경 내용 없음. 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다.
이동됨	변경 내용 없음
숨겨짐	변경 내용 없음

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
수준의 이름 또는 설명이 업데이트된 경우 [페이지 438]
수준이 삭제된 경우 [페이지 439]
수준이 이동된 경우 [페이지 439]
새 수준이 만들어진 경우 [페이지 440]

8.4.10.2 수준의 이름 또는 설명이 업데이트된 경우

아래 표에는 수준의 이름 또는 설명이 업데이트되었을 때 유니버스 수준에 미치는 영향이 상태별로 설명되어 있습니다.

표 203:

유니버스 수준의 상태	유니버스 수준에 미치는 영향
변경 안 됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
업데이트됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
삭제됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다.
이동됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
숨겨짐	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
수준이 변경되지 않은 경우 [페이지 437]
수준이 삭제된 경우 [페이지 439]
수준이 이동된 경우 [페이지 439]
새 수준이 만들어진 경우 [페이지 440]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.10.3 수준이 삭제된 경우

아래 표에는 수준이 삭제되었을 때 유니버스 수준에 미치는 영향이 상태별로 설명되어 있습니다.

표 204:

유니버스 수준의 상태	유니버스 수준에 미치는 영향
변경 안 됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다.
업데이트됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다.
삭제됨	변경 내용 없음
이동됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다.
숨겨짐	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
수준이 변경되지 않은 경우 [페이지 437]
수준의 이름 또는 설명이 업데이트된 경우 [페이지 438]
수준이 이동된 경우 [페이지 439]
새 수준이 만들어진 경우 [페이지 440]

8.4.10.4 수준이 이동된 경우

아래 표에는 수준이 이동되었을 때 유니버스 수준에 미치는 영향이 상태별로 설명되어 있습니다.

표 205:

유니버스 수준의 상태	유니버스 수준에 미치는 영향
변경 안 됨	수준이 그에 따라 이동됩니다(동일한 계층구조 내에서).
업데이트됨	수준이 그에 따라 이동됩니다(동일한 계층구조 내에서).
삭제됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다.
이동됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다.
숨겨짐	수준이 그에 따라 이동됩니다(동일한 계층구조 내에서).

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
수준이 변경되지 않은 경우 [페이지 437]
수준의 이름 또는 설명이 업데이트된 경우 [페이지 438]
수준이 삭제된 경우 [페이지 439]
새 수준이 만들어진 경우 [페이지 440]

8.4.10.5 새 수준이 만들어진 경우

OLAP 수준이 만들어지면 유니버스 수준이 만들어집니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
수준이 변경되지 않은 경우 [페이지 437]
수준의 이름 또는 설명이 업데이트된 경우 [페이지 438]
수준이 삭제된 경우 [페이지 439]
수준이 이동된 경우 [페이지 439]

8.4.11 OLAP 유니버스 업데이트 시 SAP 변수 관리

이 단원은 SAP 데이터 소스에만 적용됩니다. 유니버스 필터 및 관련 값 목록 개체는 OLAP 변수와 동등합니다. OLAP 개체와 관련하여 유니버스 개체가 관리되는 방식은 변경 유형에 따라 결정됩니다. 특정 OLAP 개체 변경이 유니버스 개체에 미치는 영향에 대해 알아보려면 아래 나열된 항목을 참조하십시오.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
SAP 변수가 변경된 경우 [페이지 441]
SAP 변수의 이름 또는 설명이 업데이트된 경우 [페이지 441]
SAP 변수가 삭제된 경우 [페이지 442]
새 SAP 변수가 만들어진 경우 [페이지 443]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.11.1 SAP 변수가 변경된 경우

아래 표에는 SAP 소스 변수가 변경되었을 때 유니버스 필터가 관리되는 방식이 상태별로 설명되어 있습니다.

표 206:

유니버스 필터의 상태	유니버스 필터에 미치는 영향
변경 안 됨	변경 내용 없음
업데이트됨	변경 내용 없음
삭제됨	생성됩니다. 변수에서 참조하는 특성이 유니버스에 없는 경우 해당 특성에 대한 하위 클래스도 생성됩니다.
이동됨	변경 내용 없음
숨겨짐	변경 내용 없음

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
SAP 변수의 이름 또는 설명이 업데이트된 경우 [페이지 441]
SAP 변수가 삭제된 경우 [페이지 442]
새 SAP 변수가 만들어진 경우 [페이지 443]

8.4.11.2 SAP 변수의 이름 또는 설명이 업데이트된 경우

다음 표에는 SAP 소스 변수의 이름이나 설명이 업데이트될 때 여러 상황에서 유니버스 필터가 관리되는 방법이 나와 있습니다.

표 207:

유니버스 필터의 상태	유니버스 필터에 미치는 영향
변경 안 됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
업데이트됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
삭제됨	생성됩니다. 변수에서 참조하는 특성이 유니버스에 없는 경우 해당 특성에 대한 하위 클래스도 생성됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스 필터의 상태	유니버스 필터에 미치는 영향
이동됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
숨겨짐	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
SAP 변수가 변경된 경우 [페이지 441]
SAP 변수가 삭제된 경우 [페이지 442]
새 SAP 변수가 만들어진 경우 [페이지 443]

8.4.11.3 SAP 변수가 삭제된 경우

아래 표에는 SAP 변수가 삭제되었을 때 유니버스 필터가 관리되는 방식이 상태별로 설명되어 있습니다.

표 208:

유니버스 필터의 상태	유니버스 필터에 미치는 영향
변경 안 됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스를 숨기도록 설정됩니다. 사용하지 않도록 설정되면 쿼리에 자동적으로 적용되는 것을 방지하기 위해 '필수'에서 '선택적'으로 변경됩니다.
업데이트됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스를 숨기도록 설정됩니다. 사용하지 않도록 설정되면 쿼리에 자동적으로 적용되는 것을 방지하기 위해 '필수'에서 '선택적'으로 변경됩니다.
삭제됨	변경 내용 없음
이동됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스를 숨기도록 설정됩니다. 사용하지 않도록 설정되면 쿼리에 자동적으로 적용되는 것을 방지하기 위해 '필수'에서 '선택적'으로 변경됩니다.
숨겨짐	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스를 숨기도록 설정됩니다. 사용하지 않도록 설정되면 쿼리에 자동적으로 적용되는 것을 방지하기 위해 '필수'에서 '선택적'으로 변경됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
SAP 변수가 변경된 경우 [페이지 441]
SAP 변수의 이름 또는 설명이 업데이트된 경우 [페이지 441]
새 SAP 변수가 만들어진 경우 [페이지 443]

8.4.11.4 새 SAP 변수가 만들어진 경우

아래 표에는 SAP 변수가 만들어졌을 때 유니버스 필터가 관리되는 방식이 상태별로 설명되어 있습니다.

표 209:

유니버스 필터의 상태	유니버스 필터에 미치는 영향
변경 안 됨	생성됩니다. 변수에서 참조하는 특성이 유니버스에 없는 경우 해당 특성에 대한 하위 클래스도 생성됩니다.
업데이트됨	생성됩니다. 변수에서 참조하는 특성이 유니버스에 없는 경우 해당 특성에 대한 하위 클래스도 생성됩니다.
삭제됨	생성됩니다. 변수에서 참조하는 특성이 유니버스에 없는 경우 해당 특성에 대한 하위 클래스도 생성됩니다.
이동됨	생성됩니다. 변수에서 참조하는 특성이 유니버스에 없는 경우 해당 특성에 대한 하위 클래스도 생성됩니다.
숨겨짐	생성됩니다. 변수에서 참조하는 특성이 유니버스에 없는 경우 해당 특성에 대한 하위 클래스도 생성됩니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
SAP 변수가 변경된 경우 [페이지 441]
SAP 변수의 이름 또는 설명이 업데이트된 경우 [페이지 441]
SAP 변수가 삭제된 경우 [페이지 442]

8.4.12 OLAP 유니버스 업데이트 시 주요 수치 또는 계수 관리

SAP 데이터 소스에서는 주요 수치를 사용하고 MSAS 및 Essbase 데이터 소스에서는 계수를 사용합니다. 유니버스 계수는 OLAP 주요 수치와 같습니다. OLAP 개체와 관련하여 유니버스 개체가 관리되는 방식은 변경 유형에 따라 결정됩니다. 특정 OLAP 개체 변경이 유니버스 개체에 미치는 영향에 대해 알아보려면 아래 나열된 항목을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
주요 수치 또는 계수가 변경되지 않은 경우 [페이지 444]
주요 수치 또는 계수의 이름, 설명 또는 데이터 형식이 업데이트된 경우 [페이지 444]
주요 수치 또는 계수가 삭제된 경우 [페이지 445]
주요 수치 또는 계수가 이동된 경우 [페이지 446]
새 주요 수치 또는 계수가 만들어진 경우 [페이지 447]

8.4.12.1 주요 수치 또는 계수가 변경되지 않은 경우

아래 표에는 SAP 주요 수치 또는 MSAS/Essbase 계수가 변경되지 않았을 때 유니버스 계수에 미치는 영향이 상태별로 설명되어 있습니다.

표 210:

유니버스 계수의 상태	유니버스 계수에 미치는 영향
변경 안 됨	변경 내용 없음
업데이트됨	변경 내용 없음
삭제됨	변경 내용 없음. 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다.
이동됨	변경 내용 없음
숨겨짐	변경 내용 없음

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
주요 수치 또는 계수의 이름, 설명 또는 데이터 형식이 업데이트된 경우 [페이지 444]
주요 수치 또는 계수가 삭제된 경우 [페이지 445]
주요 수치 또는 계수가 이동된 경우 [페이지 446]
새 주요 수치 또는 계수가 만들어진 경우 [페이지 447]

8.4.12.2 주요 수치 또는 계수의 이름, 설명 또는 데이터 형식이 업데이트된 경우

아래 표에는 SAP 주요 수치 또는 MSAS/Essbase 계수가 업데이트되었을 때 유니버스 계수에 미치는 영향이 상태별로 설명되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 211:

유니버스 계수의 상태	유니버스 계수에 미치는 영향
변경 안 됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 개체의 데이터 형식 유지 옵션이 선택되지 않은 경우 데이터 형식이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
업데이트됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 개체의 데이터 형식 유지 옵션이 선택되지 않은 경우 데이터 형식이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
삭제됨	변경 내용 없음. 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다.
이동됨	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 개체의 데이터 형식 유지 옵션이 선택되지 않은 경우 데이터 형식이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.
숨겨짐	비즈니스 이름 유지 옵션이 선택되지 않은 경우 비즈니스 이름이 업데이트됩니다. 개체 설명 유지 옵션이 선택되지 않은 경우 개체 설명이 업데이트됩니다. 개체의 데이터 형식 유지 옵션이 선택되지 않은 경우 데이터 형식이 업데이트됩니다. 이 두 옵션이 선택되어 있으면 변경이 이루어지지 않습니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
주요 수치 또는 계수가 변경되지 않은 경우 [페이지 444]
주요 수치 또는 계수가 삭제된 경우 [페이지 445]
주요 수치 또는 계수가 이동된 경우 [페이지 446]
새 주요 수치 또는 계수가 만들어진 경우 [페이지 447]

8.4.12.3 주요 수치 또는 계수가 삭제된 경우

아래 표에는 SAP 주요 수치 또는 MSAS/Essbase 계수가 삭제되었을 때 유니버스 계수에 미치는 영향이 상태별로 설명되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 212:

유니버스 계수의 상태	유니버스 계수에 미치는 영향
변경 안 됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다.
업데이트됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다.
삭제됨	변경 내용 없음
이동됨	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다.
숨겨짐	사용하지 않는 개체 삭제 옵션이 선택된 경우 삭제됩니다. 사용하지 않는 개체 숨기기 옵션이 선택된 경우 하위 클래스가 사용되지 않도록 설정됩니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
주요 수치 또는 계수가 변경되지 않은 경우 [페이지 444]
주요 수치 또는 계수의 이름, 설명 또는 데이터 형식이 업데이트된 경우 [페이지 444]
주요 수치 또는 계수가 이동된 경우 [페이지 446]
새 주요 수치 또는 계수가 만들어진 경우 [페이지 447]

8.4.12.4 주요 수치 또는 계수가 이동된 경우

아래 표에는 SAP 주요 수치 또는 MSAS/Essbase 계수가 이동되었을 때 유니버스 계수에 미치는 영향이 상태별로 설명되어 있습니다.

표 213:

유니버스 계수의 상태	유니버스 계수에 미치는 영향
변경 안 됨	개체가 그에 따라 이동됩니다.
업데이트됨	개체가 그에 따라 이동됩니다.
삭제됨	변경 내용 없음 수동으로 삭제된 개체 다시 생성 옵션이 예일 경우 생성됩니다.
이동됨	변경 내용 없음
숨겨짐	개체가 그에 따라 이동됩니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
주요 수치 또는 계수가 변경되지 않은 경우 [페이지 444]
주요 수치 또는 계수의 이름, 설명 또는 데이터 형식이 업데이트된 경우 [페이지 444]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

주요 수치 또는 계수가 삭제된 경우 [페이지 445]
새 주요 수치 또는 계수가 만들어진 경우 [페이지 447]

8.4.12.5 새 주요 수치 또는 계수가 만들어진 경우

OLAP 주요 수치 또는 계수가 만들어지면 유니버스 계수가 만들어집니다.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
주요 수치 또는 계수가 변경되지 않은 경우 [페이지 444]
주요 수치 또는 계수의 이름, 설명 또는 데이터 형식이 업데이트된 경우 [페이지 444]
주요 수치 또는 계수가 삭제된 경우 [페이지 445]
주요 수치 또는 계수가 이동된 경우 [페이지 446]

8.4.13 OLAP 유니버스 업데이트 시 SAP 주요 날짜 관리 방식

이 단원은 SAP 데이터 소스에만 적용됩니다. 유니버스 매개 변수는 OLAP 주요 날짜와 같습니다. OLAP 개체와 관련하여 유니버스 개체가 관리되는 방식은 변경 유형에 따라 결정됩니다. 특정 OLAP 개체 변경이 유니버스 개체에 미치는 영향에 대해 알아보려면 아래 나열된 항목을 참조하십시오.

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
새 SAP 주요 날짜가 변경되지 않은 경우 [페이지 447]
SAP 주요 날짜가 삭제된 경우 [페이지 448]
새 SAP 주요 날짜가 만들어진 경우 [페이지 449]

8.4.13.1 새 SAP 주요 날짜가 변경되지 않은 경우

유니버스 매개 변수는 OLAP 주요 날짜와 같습니다. 아래 표에는 SAP 주요 날짜가 변경되지 않았을 때 유니버스 매개 변수에 미치는 영향이 상태별로 설명되어 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 214:

유니버스 매개 변수의 상태	유니버스 매개 변수에 미치는 영향
변경 안 됨	변경 내용 없음
업데이트됨	해당 없음
삭제됨	해당 없음
이동됨	해당 없음
숨겨짐	해당 없음

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
SAP 주요 날짜가 삭제된 경우 [페이지 448]
새 SAP 주요 날짜가 만들어진 경우 [페이지 449]

8.4.13.2 SAP 주요 날짜가 삭제된 경우

유니버스 매개 변수는 OLAP 주요 날짜와 같습니다. 아래 표에는 SAP 주요 날짜가 삭제되었을 때 유니버스 매개 변수에 미치는 영향이 상태별로 설명되어 있습니다.

표 215:

유니버스 매개 변수의 상태	유니버스 매개 변수에 미치는 영향
변경 안 됨	삭제
업데이트됨	해당 없음
삭제됨	해당 없음
이동됨	해당 없음
숨겨짐	해당 없음

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
새 SAP 주요 날짜가 변경되지 않은 경우 [페이지 447]
새 SAP 주요 날짜가 만들어진 경우 [페이지 449]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.4.13.3 새 SAP 주요 날짜가 만들어진 경우

유니버스 매개 변수는 OLAP 주요 날짜와 같습니다. 아래 표에는 SAP 주요 날짜가 만들어졌을 때 유니버스 매개 변수에 미치는 영향이 상태별로 설명되어 있습니다.

표 216:

유니버스 매개 변수의 상태	유니버스 매개 변수에 미치는 영향
변경 안 됨	만들기
업데이트됨	해당 없음
삭제됨	만들기
이동됨	해당 없음
숨겨짐	해당 없음

관련 정보

OLAP 유니버스 새로 고침 [페이지 423]
새 SAP 주요 날짜가 변경되지 않은 경우 [페이지 447]
SAP 주요 날짜가 삭제된 경우 [페이지 448]

8.5 다양한 OLAP 큐브가 유니버스에 매핑되는 방식

8.5.1 유니버스에서 SAP BW 개체가 매핑 및 사용되는 방식

인포큐브 또는 BEx 쿼리에서 유니버스를 만들면 유니버스 디자인 도구에서는 SAP BW OLAP 구조를 유니버스의 동등한 클래스 및 개체에 매핑합니다.

BEx 쿼리에서 행, 열, 자유 특성 및 필터로 설정된 모든 InfoObject 는 유니버스에 노출됩니다. 여기에는 특성, 계층, 주요 수치, 구조 및 변수가 포함됩니다.

계층구조가 매핑되므로 Web Intelligence 사용자는 BW 계층구조를 따라 드릴다운할 수 있습니다.

인포큐브의 경우 모든 차원, 주요 수치 및 계층이 매핑됩니다.

다음 표에 각 BW 개체에 대해 만들어진 유니버스 개체가 나와 있습니다.

표 217:

SAP BW 개체:	만들어진 유니버스 개체
차원 그룹	클래스
특성	차원 및 설명 개체가 있는 하위 클래스

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

SAP BW 개체:	만들어진 유니버스 개체
계층구조의 특성	데이터 소스가 BEx 쿼리인 경우: 현재 정의된 계층구조의 각 계층구조 수준에 대해 차원 및 설명 개체를 포함하는 하위 클래스 데이터 소스가 인포큐브인 경우: 특성에 대해 정의된 모든 계층구조의 각 계층구조 수준에 대해 차원 및 설명 개체를 포함하는 하위 클래스
특성을 기반으로 하는 구조(BEx 쿼리만 해당)	구조에 대한 단일 차원 개체가 있는 클래스
탐색 특성	차원 및 설명 개체가 있는 하위 클래스(특성과 동일)
표시 특성	차원에 대한 설명 개체
주요 수치 구조	클래스
키 숫자	단위/통화에 대한 차원 개체가 있는 주요 수치 구조에 대한 클래스의 계수 개체
계산된 주요 수치(BEx 쿼리만 해당)	계수 및 차원 개체(주요 수치와 동일)
제한된 주요 수치(BEx 쿼리만 해당)	계수 및 차원 개체(주요 수치와 동일)
변수(BEx 쿼리만 해당)	쿼리의 임시 필터 변수가 적용되는 차원의 클래스에서 값 목록을 지원하는 두 개의 차원 개체(캡션 및 설명용)
주요 날짜 변수(BEx 쿼리만 해당)	유니버스의 주요 날짜 변수를 정의하는 유니버스 매개 변수

BEx 쿼리의 필터 섹션에 있는 특성은 매핑되지 않습니다. 그러나 필터링은 유니버스에 적용됩니다. 필터가 고정 값을 가질 경우 Web Intelligence 쿼리를 실행할 때 필터가 투명하게 적용됩니다. 특성의 변수가 정의된 경우 해당 변수는 다음과 같은 제한에 따라 매핑됩니다.

- 변수는 항상 필수 변수로 동작합니다.
- 계층구조 버전 변수를 제외하고 계층구조 및 계층구조 노드 변수가 지원됩니다.

이러한 제한을 피하기 위해 BEx 쿼리의 필터 섹션에서 자유 섹션으로 특성을 이동하십시오.

관련 정보

특성 매핑 및 사용 [페이지 451]
주요 수치 매핑 및 사용 [페이지 451]
계층구조 매핑 및 사용 [페이지 451]
유니버스에서 변수 지원 [페이지 452]
유니버스에 변수 매핑 [페이지 454]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

8.5.1.1 특성 매핑 및 사용

BEx 쿼리나 인포큐브의 특성에 계층구조가 정의되지 않은 경우 유니버스 디자인 도구에서는 특성이 들어 있는 클래스를 두 개의 차원 개체인 수준 00 및 수준 01 로 만듭니다. 모든 멤버가 선택되었을 때(SAP NetWeaver 기술 플랫폼에서 반환된 멤버가 모든 멤버임) 수준 00 차원은 특성의 집계를 나타냅니다. 수준 01 차원에는 특성에 대한 모든 멤버가 단순 값 목록으로 포함됩니다.

유니버스 디자인 도구에서는 차원 개체마다 키에 대해 하나의 설명 개체와 설명(간단한 설명, 설명 및 상세 설명)에 대해 최대 세 개의 설명 개체, 각 표시 특성에 대해 하나의 설명 개체를 만듭니다.

특성의 기술 이름을 사용하여 SELECT 절이 정의됩니다.

BW 쿼리에 정의된 탐색 특성은 특성이 매핑되는 것과 같은 방식으로 상위 개체 클래스에 매핑됩니다.

> **노트**
>
> 유니버스에 너무 많은 탐색 특성을 정의하면 Web Intelligence 의 쿼리 성능이 저하될 수 있습니다.

특성을 기반으로 하는 BEx 쿼리에 정의된 구조는 구조의 요소를 차원 멤버로 포함하는 단일 차원 개체로서 유니버스에 포함됩니다.

8.5.1.2 주요 수치 매핑 및 사용

인포큐브에 포함되어 있거나 BEx 쿼리에 정의된 모든 주요 수치는 유니버스의 주요 수치라는 단일 개체 클래스 아래에 포함됩니다.

대부분의 주요 수치는 통화 또는 단위 특성과 함께 BW 에 정의됩니다. 유니버스 디자인 도구에서는 각 주요 수치에 대해 다음을 만듭니다.

- 주요 수치에 해당하는 단위 없는 숫자 형식을 갖는 계수 개체
- 단위 또는 통화를 포함하는 문자 형식을 갖는 차원 개체 (예: 'USD', '€', 'km')
- SAP 서버에 구성된 사용자 기본 설정을 기반으로 하는 주요 수치 및 단위(형식이 지정된 값)가 포함된 문자 형식의 차원 개체 (예: '200 USD', '345 €', '25 km')

주요 수치 클래스에는 BEx 쿼리에 정의된 계산된 주요 수치 및 제한된 주요 수치가 포함됩니다. 원래의 계산 및 제한은 쿼리에 적용되지만 유니버스에 노출되지 않습니다.

8.5.1.3 계층구조 매핑 및 사용

계층구조가 매핑되므로 Web Intelligence 사용자는 사용자가 만든 유니버스 계층구조와 같은 방식으로 SAP BW 계층구조를 드릴다운할 수 있습니다.

> **노트**
>
> Web Intelligence 문서 속성 대화 상자의 쿼리 드릴 사용 옵션은 드릴다운 성능에 상당한 영향을 미칩니다.

계층구조가 BEx 쿼리의 특성에 정의된 경우 유니버스 디자인 도구에서는 유니버스에 계층구조의 각 수준에 대한 하위 클래스가 포함된 계층구조를 만듭니다. 구조는 현재 BEx 쿼리 정의에 따라 다음과 같이 달라집니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

- 계층구조가 BEx 쿼리에 정의된 경우 유니버스 디자인 도구에서는 유니버스에 이 계층구조를 만듭니다.
- 사용자가 런타임에 계층구조를 선택할 수 있도록 계층구조 변수가 BEx 쿼리에 정의된 경우 유니버스 디자인 도구에서는 유니버스에 일반 계층구조를 만듭니다. 이 구조에서는 특성에 사용할 수 있는 계층구조에 가장 많은 수의 수준이 정의되어 있습니다.

인포큐브 위에 유니버스를 작성할 경우 특성에 정의된 모든 계층이 결과 유니버스에 노출됩니다. 유니버스 디자인 도구에서는 각 계층구조에 대한 하위 클래스를 만들며 각 계층구조에는 해당 계층구조의 수준에 대한 하위 클래스가 포함됩니다.

유니버스에서 계층구조의 수준 00 은 구조의 최상위 노드를 나타냅니다. 계층구조에 최상위 노드가 여러 개 있는 경우 수준 00 차원에 모든 최상위 노드가 값 목록으로 포함됩니다. 계층구조 특성이 지정되지 않은 노드를 필터링하지 않도록 설정되면 지정되지 않은 멤버에 대한 최상위 노드를 수준 00 에 포함해야 합니다. 지정되지 않은 멤버는 계층구조의 최하위 수준에 그룹화됩니다.

> **노트**
>
> SAP BW 계층에는 최상위 노드가 하나뿐인 경우가 대부분입니다. 기본 유니버스에서 수준 00 개체를 삭제하면 유니버스를 보다 편리하게 사용할 수 있습니다. 일반적으로 지정되지 않은 멤버를 쿼리/보고해야 할 경우에만 수준 00 을 유지해야 합니다.

BEx 쿼리에서 계층구조 수준의 개수가 변경된 경우 유니버스를 업데이트해야 합니다.

관련 정보

OLAP 유니버스 수명 주기 관리 정보 [페이지 420]

8.5.1.4 유니버스에서 변수 지원

SAP 변수는 BW 쿼리에 정의된 사용자 프롬프트로 해석될 수 있습니다. 변수는 필수 변수나 선택적 변수가 될 수 있으며 기본값을 가질 수 있습니다.

특성 변수는 특성에 대한 값을 필터링하는 데 사용됩니다. 변수는 쿼리 실행 시 값으로 채워집니다. 특성 값, 계층구조, 계층구조 노드, 텍스트 및 수식 요소를 저장할 수 있습니다.

SAP BW 변수는 BEx 쿼리에만 적용됩니다.

> **노트**
>
> BEx 쿼리 디자이너에서 변수를 정의할 때 SAP BW 변수 마법사의 기타 설정 대화 상자에서 [입력 준비] 옵션을 선택해야 합니다.

유니버스에서는 다음 유형의 SAP BW 변수가 지원됩니다.

특성 변수
계층구조 버전 변수를 제외한 계층구조 변수
계층구조 노드 변수
통화 변수

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

수식 변수
텍스트 변수(대체 경로)
주요 날짜 변수

다음 표에서는 사용자 입력 BW 변수에 대한 유니버스 지원을 보여줍니다. 사용자 입력 변수는 필수 변수나 선택적 변수가 될 수 있으며 기본값을 가질 수 있습니다.

표 218:

변수 유형		지원 수준
특성(주요 날짜 및 통화 포함)	단일 값 프롬프트	지원
	여러 단일 값 프롬프트	지원
	간격 프롬프트	지원 단일 값 변수인 주요 날짜 변수에는 지원되지 않습니다.
	선택 옵션 프롬프트	간격 프롬프트로 지원 단일 값 변수인 주요 날짜 변수에 대한 간격 프롬프트로 지원되지 않습니다.
	미리 계산된 값 집합	지원되지 않음
텍스트		지원
수식		가격, 할당량 및 숫자 값 지원
계층구조		버전 변수를 제외하고 지원
계층구조 노드		지원

다음 표에 BW 변수의 다른 처리 유형에 대한 유니버스 지원 사항이 나와 있습니다.

표 219:

변수 유형	처리 유형			
	대체 경로	권한 부여	Customer Exit	SAP Exit
특성	지원	지원	지원, 유니버스에서 프롬프트가 만들어지지 않음	지원
텍스트	지원	해당 없음	지원	해당 없음
수식	지원	해당 없음	지원	사용자 입력 없이 지원
계층구조	해당 없음	해당 없음	지원	지원
계층구조 노드	해당 없음	해당 없음	지원	사용자 입력 없이 지원

제외 연산자는 지원되지만 Web Intelligence 에서는 선택한 값이 쿼리에서 제외되도록 지정하지 않습니다. 보다 작음 및 보다 큼과 같은 다른 연산자는 선택 옵션 항목 유형에만 사용할 수 있습니다. 선택 옵션 유형은 Web Intelligence 프롬프트 간격으로 변환됩니다.

노트

Web Intelligence 에서 BW 변수를 처리하려면 Web Intelligence 쿼리에 하나 이상의 계수를 포함해야 합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

유니버스에 변수 매핑 [페이지 454]
유니버스에서 주요 날짜 변수 지원 [페이지 455]
유니버스에서 계층구조 및 계층구조 노드 변수 지원 [페이지 456]

8.5.1.4.1 유니버스에 변수 매핑

결과 집합에서 차원이 사용되지 않는 경우에도 사용자가 결과 집합을 제한할 수 있도록 사용자에게 모든 선택적 및 필수 변수가 프롬프트되어야 합니다. 따라서 해당 특성이 쿼리에 없는 경우에도 BEx 쿼리에 정의된 변수는 매핑됩니다.

사용자는 변수가 필수 변수인지 선택적 변수인지 알아야 하며 선택적 변수는 무시해도 됩니다. 선택적 변수는 유니버스에서 선택적으로 정의되며 Web Intelligence 에서는 선택적 프롬프트가 됩니다. 필수 변수는 Web Intelligence 에서 필수 프롬프트가 됩니다.

특성 변수에 대해서는 유니버스 디자인 도구가 유니버스에 필수 필터를 만듭니다. 필수 필터는 Web Intelligence 사용자는 볼 수 없지만 유니버스에서 구축된 모든 Web Intelligence 쿼리에 시스템에서 자동으로 적용되는 미리 정의된 쿼리 필터 개체입니다.

표 220:

변수 유형	매핑 대상
특성 변수(통화 및 수식 변수 포함)	유니버스 필수 필터
계층구조 변수	유니버스 필수 필터
계층구조 노드 변수	클래스 필수 필터
주요 날짜 변수	유니버스 매개 변수

각 필수 필터에 대해 @Prompt 함수가 예상 값 목록을 표시할 수 있도록 두 개의 차원 개체가 참조 개체로 만들어집니다. 값 목록 차원은 유니버스에서 숨겨져 있습니다. 이러한 차원은 프롬프트의 올바른 작동을 위해 필요하므로 삭제해서는 안 되며 이동 또는 수정할 때도 주의해야 합니다.

변수의 기본값은 기본 키, 영구/비영구 및 기본값 매개 변수를 사용하여 필터의 @Prompt 함수에 정의됩니다. @Prompt 함수 구문은 유니버스의 필터에 대한 속성 페이지에서 확인할 수 있습니다.

BW 변수와 Web Intelligence 사용자가 정의한 필터 간의 충돌을 방지하기 위해 SAP 변수 정의에 포함된 개체는 개체 속성의 고급 페이지에서 사용 가능한 위치 - 조건 옵션이 선택되지 않은 상태로 생성됩니다. 이를 통해 Web Intelligence 사용자가 필터 창에 SAP 변수와 관련된 차원을 포함하는 경우를 방지할 수 있습니다.

예

SAP BW 변수에 대해 생성된 WHERE 절

이 예에서는 차원 개체 Customer2 의 BW 변수에 대해 생성된 WHERE 절을 보여 줍니다. 변수에 대해 생성된 WHERE 절의 구문은 필터에 대한 속성 페이지에서 확인할 수 있습니다.

```
<FILTER KEY="[Z_VAR002]">
    <CONDITION OPERATORCONDITION="Equal">
        <CONSTANT TECH_NAME="@Prompt(
            'Customer Variable Single Value Mandatory',
            'A',
```

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

```
            'Customer2\LovCustomer Variable Single Value MandatoryBase',
            mono,
            primary_key)"/>
    <CONDITION>
</FILTER>
```

프롬프트 텍스트는 BW 변수 이름에서 생성됩니다. 이 텍스트를 좀 더 이해하기 쉽게 편집할 수 있습니다.

`Customer2\LovCustomer Variable Single Value MandatoryBase` 는 값 목록 작성에 사용되는 숨겨진 유니버스 개체의 이름입니다.

> **노트**
>
> 클래스의 이름을 바꾸거나 값 목록 개체를 다른 폴더로 이동할 경우 필터 키에서 해당 구문을 업데이트해야 합니다.

8.5.1.4.2 변수 및 값 목록 지원

BEx 쿼리에는 10 개 이상의 변수가 포함될 수 있습니다. 이것은 10 개 이상의 값 목록이 로드될 수 있음을 의미합니다. 값 목록을 로드 및 새로 고치는 작업은 성능에 상당한 영향을 줄 수 있습니다. 변수가 있는 쿼리의 쿼리 성능을 향상시키는 데 다음 옵션을 사용할 수 있습니다.

- 유니버스 생성 시에 모든 SAP BW 변수(주요 날짜 제외)는 필수 필터에 매핑됩니다. 기본적으로 필터 개체는 값 목록(사용 가능한 계층구조 노드 제외)과 연결되어 있지 않습니다. 개체 속성 페이지에서 값 목록을 명시적으로 연결해야 합니다.
- 선택적 변수는 선택적 프롬프트로 생성됩니다. 선택적 프롬프트는 쿼리 실행 시에 값 목록을 자동으로 로드하지 않습니다.
- 값 목록 속성에 대한 위임 검색 옵션은 쿼리 실행 시에 빈 값 목록을 표시합니다. 사용자는 검색 조건을 입력하여 값 목록에 반환되는 값 수를 제한할 수 있습니다.
  값 목록에 대한 위임된 검색 옵션을 활성화하려면 값 목록이 적용되는 개체의 개체 속성 페이지에서 값 목록 속성을 편집합니다.

  > **노트**
  >
  > 계단식 값 목록에서는 위임된 검색이 지원되지 않습니다.

관련 정보

OLAP 유니버스의 선택적 프롬프트 [페이지 418]

8.5.1.4.3 유니버스에서 주요 날짜 변수 지원

BEx 쿼리의 주요 날짜 변수를 사용하여 시간 종속 데이터에 대한 날짜를 지정할 수 있습니다. 주요 날짜는 차원에서 검색되는 데이터에 영향을 미칠 수 있습니다. 예를 들어 제품 설명은 시간에 따라 변경될 수 있습니다. 주요 날짜는 계층구조

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

에 영향을 미칠 수 있습니다. 예를 들어 특정 가격 센터가 어떤 해에는 수준 01 에 있을 수 있지만 다른 해에는 수준 02 에 있을 수 있습니다.

사용자가 입력한 날짜 값은 BW 쿼리의 차원에 포함되지 않으므로 주요 날짜 변수는 특수한 SAP BW 변수입니다. 주요 날짜는 쿼리의 한 속성입니다.

BEx 쿼리에서 주요 날짜 변수는 다음 두 가지 용도로 정의될 수 있습니다.

- 특정 계층구조에 대해 유효 날짜를 지정하여 해당 계층구조에만 영향을 줍니다.
- 전체 쿼리에 대해 날짜를 지정합니다. 이 경우 쿼리에 설정된 주요 날짜는 다음에 영향을 줍니다.
  - 시간 종속 마스터 데이터
  - 통화 환율
  - 계층 목록
  - 시간 종속 계층구조

> **노트**
>
> 유니버스에서 주요 날짜의 사용은 전체 유니버스로 제한됩니다. 따라서 유니버스에서 생성된 주요 날짜는 다른 모든 SAP 변수 및 데이터에 영향을 미칩니다.

SAP BW 에서는 BW 쿼리당 하나의 주요 날짜 변수만 지원하므로 유니버스에는 주요 날짜 변수가 하나만 포함되어 있습니다.

주요 날짜 변수는 필수 또는 선택적 변수가 될 수 있으며 기본값을 가질 수 있습니다. 기본값이 정의되지 않았으며 사용자가 값을 입력하지 않으면 현재 시스템 날짜가 사용됩니다.

쿼리의 주요 날짜 변수 속성은 다음 표에 설명된 5 개의 유니버스 매개 변수에 매핑됩니다.

표 221:

매개 변수	설명
KEYDATE_ENABLED	유니버스에서 주요 날짜를 사용하도록 설정된 경우 *Yes* 로 설정합니다.
KEYDATE_NAME	주요 날짜 변수의 기술 이름입니다.
KEYDATE_CAPTION	사용자에게 값을 지정하라는 프롬프트가 표시되기 전에 주요 날짜 변수에 대한 캡션이 표시됩니다.
KEYDATE_DEFAULT_VALUE	주요 날짜의 기본값입니다(있는 경우).
KEYDATE_MANDATORY	사용자가 값을 입력하거나 기본값을 사용해야 할 경우 *Yes* 로 설정합니다.

쿼리 실행 시 Web Intelligence 는 모든 쿼리에 대해 동일한 주요 날짜를 제안합니다. 사용자는 주요 날짜를 수정할 수 있습니다. *Keydate 속성* 대화 상자에서 사용되는 주요 날짜를 관리할 수 있습니다. 사용자에게 다른 유형의 변수 프롬프트가 표시되기 전에 주요 날짜를 묻는 프롬프트가 표시됩니다.

8.5.1.4.4 유니버스에서 계층구조 및 계층구조 노드 변수 지원

계층구조 변수는 쿼리에 사용될 계층구조를 지정하라는 프롬프트를 표시하는 데 사용됩니다. Web Intelligence 사용자는 쿼리 및 보고서를 만들어서 계층구조에서 멤버를 검색하여 표시할 수 있습니다.

계층구조 변수가 선택적이고 사용자가 프롬프트를 비워 두면 보고서에 계층구조가 사용되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

보고서에는 선택된 계층구조와는 별도로 가장 많은 수의 계층구조 수준이 포함됩니다. 결과 집합에 반환되지 않은 계층구조 수준은 보고서에서 빈 상태가 됩니다.

계층구조 노드 변수는 쿼리에서 계층구조에 대한 노드 위에 정의할 노드에 대한 프롬프트를 표시하는 데 사용됩니다.

쿼리에 계층구조 및 계층구조 노드 변수가 모두 포함되어 있을 경우 Web Intelligence 사용자는 먼저 사용 가능한 계층구조 목록에서 계층구조를 선택해야 합니다. 그런 후 계층구조 노드를 선택합니다. 사용 가능한 계층구조 노드 목록에 모든 계층구조에 대한 계층구조 노드가 표시됩니다. 이 목록은 선택된 계층구조를 기반으로 필터링되지 않습니다. 사용자가 올바른 계층구조에서 노드를 선택해야 합니다. 다른 계층구조의 계층구조 노드를 선택하면 보고서가 빈 상태가 될 수 있습니다.

관련 정보

계층구조 매핑 및 사용 [페이지 451]

8.5.2 Essbase 큐브가 유니버스 구성 요소에 매핑되는 방식

유니버스 디자인 도구는 Essbase 개요를 동등한 클래스 및 개체에 매핑하여 Essbase 큐브에서 유니버스를 만듭니다. 연결을 만들 때 큐브 데이터 소스를 식별합니다.

Essbase 별칭 테이블은 개요의 차원, 수준 및 멤버에 대한 대체 이름 집합을 정의합니다. 유니버스 디자인 도구는 Essbase 데이터 소스에 대한 연결을 만들 때 선택한 별칭 테이블에 따라 이러한 이름을 사용하여 유니버스를 생성합니다.

Essbase 개요에서 계수는 차원으로 정의됩니다. Essbase 데이터 소스에 대한 연결을 만들 때 계수 차원으로 사용할 차원을 선택합니다. 그러면 유니버스 디자인 도구가 계수로 선택된 차원의 멤버를 유니버스에 생성합니다.

모든 차원은 여러 수준이 있는 계층을 지원합니다. 차원마다 최대 하나의 계층구조를 정의할 수 있습니다.

다음 표에서는 각 Essbase 개요 요소에 대해 유니버스에서 생성되는 개체를 보여 줍니다.

표 222:

Essbase 개체	생성되는 유니버스 개체
차원	차원에 대한 세대를 포함하는 클래스
세대	두 개의 세부 개체(캡션 및 이름용)가 있는 차원 클래스의 개체
계수 차원	유니버스 연결에서 계수 차원으로 선택한 차원에 따라 명명된 클래스(일반적으로 계수 클래스 또는 계정 클래스)
계수	계수 클래스 또는 하위 클래스의 계수 개체 계수는 Essbase 개요의 구조와 일치하는 클래스 및 하위 클래스의 구조로 만들어집니다.

계수는 기본적으로 집계 프로젝션 함수가 **데이터베이스 위임**으로 설정된 상태로 생성됩니다. Web Intelligence 보고서를 새로 고칠 때 계수의 집계는 데이터베이스 서버로 위임됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

관련 정보

OLAP 데이터 소스 연결 정보 [페이지 398]
데이터베이스 위임 프로젝션 함수 [페이지 271]

8.5.3 MSAS 큐브가 유니버스 구성 요소에 매핑되는 방식

유니버스 디자인 도구에서는 MSAS 구조를 동등한 클래스 및 개체에 매핑하여 MSAS 큐브에서 유니버스를 만듭니다. 연결을 만들 때 큐브 데이터 소스를 지정합니다.

다음 표에서는 각 MSAS 개체에 대해 유니버스 구조에서 생성되는 개체를 보여 줍니다. 이 매핑은 MSAS 표준 큐브뿐만 아니라 MSAS 가상 큐브와 로컬 큐브(.cub 파일)에도 적용됩니다.

표 223:

MSAS 개체	생성되는 유니버스 개체
차원	차원에 대한 개체를 포함하는 클래스
표시 폴더(MSAS 2005)	차원 클래스의 하위 클래스
계층구조	해당 차원 클래스의 하위 클래스 또는 해당 표시 폴더 클래스의 하위 클래스에 대한 하위 클래스
특성(MSAS 2005)	해당 차원 클래스의 하위 클래스 또는 해당 표시 폴더 클래스의 하위 클래스에 대한 하위 클래스
측정값	모든 계수 개체를 포함하는 계수 클래스 계수 개체는 계수 클래스 또는 계수 그룹에 대한 하위 클래스에 생성됩니다.
계수 그룹(MSAS 2005)	계수 클래스의 하위 클래스
수준	차원 클래스나 하위 클래스의 개체 및 모든 하위 수준의 집계를 나타내는 모든 수준 개체
수준 속성	해당 수준이 적용되는 수준 개체의 세부 사항

계수는 기본적으로 집계 프로젝션 함수가 **데이터베이스 위임**으로 설정된 상태로 생성됩니다. Web Intelligence 보고서를 새로 고칠 때 계수의 집계는 데이터베이스 서버로 위임됩니다.

관련 정보

OLAP 데이터 소스 연결 정보 [페이지 398]
데이터베이스 위임 프로젝션 함수 [페이지 271]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

9 메타데이터 소스의 유니버스 작업

9.1 메타데이터 소스에서 유니버스 생성 소개

유니버스 디자인 도구의 메타데이터 교환 기능을 사용하여 다른 데이터 웨어하우스 제품에서 생성된 XML 파일을 기반으로 유니버스를 만들 수 있습니다. 메타데이터 교환은 XML 파일의 내용을 분석하여 메타데이터 정보를 추출하고 이러한 정보를 클래스, 개체, 테이블, 열, 사용자 지정 계층구조 및 조인을 비롯한 BusinessObjects 메타데이터로 변환합니다. 그런 다음 이 응용 프로그램은 새 BusinessObjects 유니버스를 만듭니다. 다른 메타데이터 소스에서 유니버스를 만들 수도 있습니다.

유니버스 디자인 도구를 사용하여 다음과 같은 메타데이터 데이터 소스에서 유니버스를 만들 수 있습니다.

표 224:

메타데이터 소스	이름
다음 표준과 호환되는 XML 파일	• Common Warehouse Model(CWM 1.0) • Common Warehouse Model OLAP(CVM OLAP) • Oracle Warehouse 작성기 • BusinessObjects Data Integrator • IBM DB2 Data Warehouse Center • IBM DB2 Cube Views
데이터베이스 뷰	Oracle Analytic Workspaces

유니버스 디자인 도구를 사용하여 특정 XML 메타데이터 소스를 사용하는 유니버스를 업데이트하고 유니버스를 DB2CV(DB2 큐브 뷰) XML 형식으로 내보낼 수도 있습니다.

9.2 개요

메타데이터 교환 패널(파일 > 메타데이터 교환)에서 메타데이터 형식을 선택합니다. 이 형식은 대상 메타데이터 소스 파일에 사용됩니다. 자세한 내용은 메타데이터 소스 선택 단원을 참조하십시오.

형식을 선택한 다음에는 유니버스 작성기 마법사를 따라 대상 데이터베이스를 선택하고 유니버스를 작성하는 데 사용할 구조를 선택합니다. 그런 다음 대상 연결을 선택하고 유니버스를 생성합니다.

유니버스 만들기 프로세스는 모든 XML 메타데이터 소스에 대해 동일합니다. XML 메타데이터 소스에서 유니버스를 만드는 방법은 XML 소스에서 유니버스 만들기 [페이지 461] 단원에 설명되어 있습니다.

Oracle Analytic Workspaces 데이터 소스에 대한 유니버스 만들기 프로세스는 다릅니다. 연결을 선택하면 선택한 메타데이터 소스에 고유한 유니버스 만들기 패널이 나타납니다. 지원되는 각 메타데이터 소스는 해당 단원에 자세히 설명되어 있습니다.

메타데이터 소스에서 유니버스를 만든 다음에는 다른 유니버스에 맞게 유니버스 구성 요소를 수정할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스를 저장하고 중앙 관리 서버(CMS)로 내보냅니다. 유니버스를 CMS 로 내보내면 Web Intelligence 사용자가 이를 사용하여 쿼리와 보고서를 만들 수 있습니다.

9.3 유니버스 작성 개요

메타데이터 교환 패널(파일 > 메타데이터 교환)에서 메타데이터 형식을 선택합니다. 이 형식은 대상 메타데이터 소스 파일에 사용됩니다. 자세한 내용은 메타데이터 소스 선택 [페이지 460] 단원을 참조하십시오.

형식을 선택한 다음에는 Universe Builder 마법사를 따라 대상 데이터베이스를 선택하고 유니버스를 작성하는 데 사용할 구조를 선택합니다. 그런 다음 대상 연결을 선택하고 유니버스를 생성합니다.

유니버스 만들기 프로세스는 모든 XML 메타데이터 소스에 대해 동일합니다. XML 메타데이터 소스에서 유니버스를 만드는 방법은 XML 소스에서 유니버스 만들기 [페이지 461] 단원에 설명되어 있습니다.

Oracle Analytical Workspaces 데이터 소스에 대한 유니버스 작성 프로세스는 다릅니다. 연결을 선택하면 Oracle Analytic Workspaces 와 관련된 유니버스 만들기 패널이 나타납니다. 데이터베이스에서 뷰를 만든 다음 뷰에서 유니버스를 만듭니다.

메타데이터 소스에서 유니버스를 만든 다음에는 다른 유니버스에 맞게 유니버스 구성 요소를 수정할 수 있습니다.

유니버스를 저장하고 중앙 관리 서버(CMS)로 내보냅니다. 유니버스를 CMS 로 내보내면 Web Intelligence 사용자가 이를 사용하여 쿼리와 보고서를 만들 수 있습니다.

9.4 메타데이터 소스 선택

메타데이터 소스를 선택하여 메타데이터 교환 패널(파일 > 메타데이터 교환)에서 유니버스를 만들거나 업데이트합니다. 유니버스를 DB2CV XML 형식으로 내보내도록 선택할 수도 있습니다.

메타데이터 교환 패널에서 다음과 같은 옵션을 사용할 수 있습니다.

표 225:

메타데이터 교환 옵션	설명
뷰에서 유니버스 만들기	드롭다운 목록에서 메타데이터 소스 형식을 선택합니다. 이는 유니버스를 작성하는 데 사용할 소스 XML 파일 또는 데이터베이스 뷰입니다. 유니버스 만들기 마법사는 메타데이터 소스에 대한 연결을 선택하고 유니버스에 매핑할 메타데이터 구성 요소를 선택하여 유니버스 생성에 이르는 과정을 안내합니다.
다음에서 유니버스 업데이트	업데이트된 메타데이터 소스를 선택합니다. 이 메타데이터 소스는 유니버스를 만드는 데 사용된 것입니다. 소스가 업데이트되었으므로 이제 동일한 수정 작업으로 유니버스를 업데이트합니다. 유니버스 업데이트 마법사는 유니버스를 업데이트하는 데 필요한 과정을 안내합니다.

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

메타데이터 교환 옵션	설명
다음으로 유니버스 내보내기	유니버스를 내보낼 메타데이터 형식을 선택합니다. 예를 들어, DB2CV XML 표준을 선택한 다음 유니버스를 이 형식으로 저장할 수 있습니다.

9.5 메타데이터 소스 옵션을 선택하려면

1. 파일 > 메타데이터 교환을 선택합니다.

   메타데이터 교환 패널이 나타납니다.
2. 새 유니버스를 생성하려면 다음에서 유니버스 만들기 드롭다운 목록 상자에서 메타데이터 형식을 선택합니다.

   기존 유니버스를 업데이트하려면 다음에서 유니버스 업데이트 드롭다운 목록 상자에서 사용한 메타데이터 소스를 선택합니다.

   유니버스를 메타데이터 형식으로 내보내려면 다음으로 유니버스 내보내기 드롭다운 목록 상자에서 대상 메타데이터 형식을 선택합니다.
3. 확인을 클릭합니다.

   만들기, 업데이트 또는 내보내기 마법사가 시작됩니다.
4. 마법사 단계를 따릅니다. 각 마법사에서 사용할 수 있는 옵션에 대한 정보는 위의 표에서 메타데이터 교환 열을 참조하십시오.

   XML 메타데이터 소스를 선택한 경우 만들기, 업데이트, 내보내기 관련 마법사 사용에 대한 내용은 뷰 만들기 및 유니버스 생성 [페이지 481] 단원을 참조하십시오.

   Oracle OLAP(Oracle Analytic Workspaces)를 선택한 경우 자세한 내용은 Oracle Analytic Workspaces [페이지 473] 단원을 참조하십시오.

9.6 XML 소스에서 유니버스 만들기

메타데이터 교환(파일 > 메타데이터 교환)에서 사용할 수 있는 OLAP Universe Builder 마법사를 따라 XML 메타데이터 소스에서 유니버스를 만듭니다. 유니버스를 생성하기 전에 유니버스 연결 및 생성 옵션을 설정할 수 있습니다.

관련 정보

XML 메타데이터 소스 [페이지 462]

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

9.6.1 XML 메타데이터 소스

다음 데이터 소스 표준을 준수하여 XML 파일에서 유니버스를 만들 수 있습니다.

- Common Warehouse Model(CWM Relational 1.0)
- Common Warehouse Model OLAP(CWM OLAP)
- Oracle Warehouse Builder(Oracle WB)
- Data Integrator
- IBM DB2 Data Warehouse Center(IBM DB2 DWC)
- IBM DB2 Cube Views

메타데이터 교환(파일 > 메타데이터 교환)에서 사용할 수 있는 OLAP Universe Builder 마법사를 따라 XML 메타데이터 소스에서 유니버스를 만듭니다.

9.6.2 XML 메타데이터 소스에서 유니버스를 생성하려면

1. 파일 > 메타데이터 교환을 선택합니다.

   메타데이터 교환 패널이 나타납니다.
2. 다음에서 유니버스 만들기 드롭다운 목록 상자에서 메타데이터 형식을 선택합니다.

   확인을 클릭합니다.

   Universe Builder 마법사가 시작됩니다.

   다음을 클릭합니다.

   XML 파일 소스 페이지가 나타납니다.
3. 찾아보기 단추를 클릭하고 XML 소스 파일을 선택합니다. 이 파일은 유니버스를 생성할 때 사용할 파일입니다.

   다음을 클릭합니다.

   데이터베이스 선택 페이지가 나타납니다.
4. 소스 데이터베이스를 클릭합니다.

   다음을 클릭합니다.

   유니버스 요소 페이지가 나타납니다. 사용 가능한 데이터베이스 테이블 및 열이 왼쪽 창에 나열됩니다.
5. 하나 이상의 테이블 및 열을 선택하고 오른쪽 화살표를 클릭하여 오른쪽 창을 채웁니다. 오른쪽 창에 있는 테이블 및 열은 생성된 유니버스에 나타납니다. 필요에 따라 화살표 단추를 사용하여 유니버스 창에서 테이블을 추가하고 제거할 수 있습니다.

   다음을 클릭합니다.

   연결 및 유니버스 속성 페이지가 나타납니다. 여기에는 유니버스 디자인 도구에서 사용 가능한 연결이 나열됩니다.
6. 연결 목록에서 연결을 클릭합니다. 이 연결은 유니버스에서 데이터 검색에 사용하는 데이터 소스에 대한 연결입니다.

   유니버스 이름을 입력합니다.

   옵션 확인란을 선택하거나 선택을 취소합니다. 고급 단추를 클릭하여 추적 로그 파일 및 XML 소스 파일 옵션을 설정합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

다음을 클릭합니다.

유니버스 생성 요약 페이지가 나타납니다. 이 페이지에는 마법사에서 선택한 옵션에 대한 요약 정보가 표시됩니다.

마침을 클릭합니다.

생성된 유니버스가 유니버스 및 유니버스 디자인 도구의 구조 창에 나타납니다.

9.6.3 연결 및 유니버스 옵션 선택

메타데이터 Universe Builder 마법사의 연결 및 유니버스 작성 페이지에는 다음과 같은 옵션이 있습니다.

표 226:

마법사 페이지	유니버스 옵션	설명
유니버스 설정 작성	연결 선택	나열되는 연결은 유니버스 디자인 도구에서 사용 가능한 연결입니다. 이 연결은 대상 RDBMS 에 대한 연결입니다.
	유니버스 이름	생성될 유니버스의 이름입니다.
	유니버스를 자동으로 저장	이 옵션을 선택하면 유니버스가 생성 시 저장됩니다.
	기존 유니버스 바꾸기	이 옵션을 선택하면 이름이 동일한 유니버스가 있으며 유니버스를 자동으로 저장을 선택한 경우 새 유니버스가 기존 유니버스를 바꿉니다.
고급 설정	일반 탭 추적	추적 폴더의 경로입니다. 이 폴더에는 유니버스 생성 시 로그 파일이 저장됩니다. 폴더를 찾아서 선택할 수 있습니다.
	파일 위치 탭 기본 XML 소스 파일 폴더	유니버스를 만드는 데 사용하는 XML 파일을 저장하는 기본 폴더의 경로입니다. 폴더를 찾아서 선택할 수 있습니다.
	파일 위치 탭 매개 변수 파일	매개 변수 파일을 저장하는 기본 폴더의 경로입니다. 이러한 파일은 유니버스 생성 시 만들어집니다. 이러한 파일은 선택한 메타데이터를 다른 유니버스를 만들거나 업데이트하는 데 사용할 수 있도록 저장하고 참조합니다. 매개 변수 파일은 선택한 메타데이터를 저장하지 않습니다. 매개 변수 파일은 원본 XML 파일을 통해 선택한 메타데이터에 브리지를 리디렉션하는 필터입니다. 폴더를 찾아서 선택할 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

9.6.4 XML 메타데이터 소스에서 유니버스를 업데이트하려면

1. 파일 > 메타데이터 교환을 선택합니다. 메타데이터 교환 패널이 나타납니다.
2. 다음에서 유니버스 업데이트 드롭다운 목록 상자에서 메타데이터 형식을 선택합니다. 확인을 클릭합니다. Universe Builder 마법사가 시작됩니다. 다음을 클릭합니다. XML 파일 소스 페이지가 나타납니다.
3. 찾아보기 단추를 클릭하고 XML 소스 파일을 선택합니다. 이 파일은 유니버스를 업데이트할 때 사용할 파일입니다. 다음을 클릭합니다. 데이터베이스 선택 페이지가 나타납니다.
4. 소스 데이터베이스를 클릭합니다. 다음을 클릭합니다. 유니버스 파일 페이지가 나타납니다. 찾아보기 단추를 클릭하고 유니버스를 선택합니다. 이 유니버스는 선택한 XML 메타데이터 소스에서 업데이트할 유니버스입니다. 다음을 클릭합니다. 유니버스 요소 페이지가 나타납니다. 사용 가능한 데이터베이스 테이블 및 열이 왼쪽 창에 나열됩니다. 추가 또는 수정된 테이블은 빨강 확인 표시로 표시됩니다.
5. 하나 이상의 테이블 및 열을 선택하고 오른쪽 화살표를 클릭하여 오른쪽 창을 수정한 테이블로 채웁니다. 오른쪽 창에 있는 테이블 및 열은 생성된 유니버스에 나타납니다. 필요에 따라 화살표 단추를 사용하여 유니버스 창에서 테이블을 추가하고 제거할 수 있습니다. 다음을 클릭합니다. 연결 및 유니버스 속성 페이지가 나타납니다. 여기에는 유니버스 디자인 도구에서 사용 가능한 연결이 나열됩니다. 자세한 내용은 연결 및 유니버스 옵션 선택 [페이지 463] 단원에서 설명합니다.
6. 연결 목록에서 연결을 클릭합니다. 이 연결은 유니버스에서 데이터 검색에 사용하는 데이터 소스에 대한 연결입니다. 유니버스 이름을 입력합니다. 옵션 확인란을 선택하거나 선택을 취소합니다. 고급 단추를 클릭하여 추적 로그 파일 및 XML 소스 파일 옵션을 설정합니다. 다음을 클릭합니다. 유니버스 생성 요약 페이지가 나타납니다. 이 페이지에는 마법사에서 선택한 옵션에 대한 요약 정보가 표시됩니다. 마침을 클릭합니다. 업데이트된 유니버스가 유니버스 및 유니버스 디자인 도구의 구조 창에 나타납니다.

9.7 DB2CV 로 유니버스 내보내기

유니버스를 IBM DB2 Cube Views XML 형식 파일로 내보낼 수 있습니다.

유니버스 정의가 IBM DB2 Cube Views XML 형식 규격인 XML 파일로 내보내집니다. 그러면 IBM DB2 Cube Views 에서 API 또는 OLAP Center 도구를 사용하여 이 파일을 로드할 수 있습니다. IBM DB2 Cube Views 는 XML 파일에서 메타데이터를 읽은 다음 향후 쿼리 최적화를 위해 적절한 AST(Automatic Summary Table)를 권장합니다.

관련 정보

내보내는 데 필요한 유니버스 사전 조건 [페이지 464]
유니버스 메타데이터 확인 [페이지 465]

9.7.1 내보내는 데 필요한 유니버스 사전 조건

다음 목록은 유니버스를 XML 파일로 내보내는 데 필요한 유니버스 사전 조건에 대해 설명합니다.

 © 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스 수준 제한

- 각 유니버스를 큐브 모델로 내보냅니다.
- 유니버스는 단일 팩트 테이블을 사용하는 단일 눈송이 스키마와 일치해야 합니다.
- 유니버스에는 적어도 하나 이상의 계수가 있어야 합니다.
- 유니버스 간의 링크는 지원되지 않습니다.
- 컨텍스트를 적용하지 않으므로 내보내지 않습니다.
- 사용자 지정 계층구조: 사용자 지정 계층구조 수준은 동일한 클래스로 그룹화되어야 합니다.

클래스 및 개체

- @Select 함수는 유일하게 지원되는 @함수입니다. 다른 @함수는 내보낼 때 매핑되지 않습니다.
- 개체 정의의 Where 필드에 있는 조건을 내보내지 않습니다. 참고: 조건이 최적화에 사용되지 않으므로 DB2 Cube Views 개체에서 지원되지 않습니다.
- 다중 매개 변수 집계 함수를 내보내지 않습니다.
- 각 클래스에는 동일한 차원 테이블에 작성되는 개체가 있어야 합니다.
- IBMDB2CV 의 동일한 차원에서 참조하는 모든 개체(IBMDB2CV 의 특성)는 유니버스에 동일한 클래스로 그룹화되어야 합니다. 다른 클래스의 다른 계수는 IBMDB2CV 의 팩트 개체에 자동으로 추가됩니다.

조인

조인의 왼쪽 또는 오른쪽 열이 유니버스의 개체와 일치하지 않으면 이 열의 특성이 자동으로 만들어져 열 테이블이 포함된 차원(또는 팩트)에 추가됩니다.

9.7.2 유니버스 메타데이터 확인

이 단원에서는 유니버스 정의를 XML 파일에 내보내는 동안 IBM DB2 Cube Views 에 일치하는 다중 차원이 없는 유니버스의 개체를 확인하고 처리하는 방법에 대해 설명합니다.

관계형 메타데이터가 포함된 유니버스

유니버스는 다중 차원 디자인 제약 조건이 없는 관계형 메타데이터를 기반으로 합니다. 유니버스의 모든 개체는 해당 IBM DB2 Cube Views 개체와 일치할 필요가 없으므로 IBM DB2 Cube Views 다중 차원 규칙을 따르지 않습니다.

관계형 구조를 올바르게 일치시키려면 BusinessObjects UMB 에서 특정 자동 검사 프로세스를 실행하여 IBM DB2 Cube Views 에 적절한 필수 메타데이터를 확인하고 정의해야 합니다. 영향을 받는 다중 차원 개체는 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

팩트

IBM DB2 Cube Views 팩트 개체는 유니버스의 계수 집합에서 자동으로 작성됩니다.

차원

팩트로 확인되지 않는 테이블은 차원 테이블로 간주됩니다. IBM DB2 Cube Views 차원 개체는 BusinessObjects 클래스에서 직접 추론됩니다.

클래스 안의 모든 개체는 IBM DB2 Cube Views 차원의 특성을 확인합니다. 클래스 안의 BusinessObjects 개체에서 유추되는 테이블은 개체의 Select 필드를 구문 분석하여 검색됩니다.

특성

특성은 유니버스의 테이블 열에서 직접 추론되지 않습니다. 지원되는 특성은 다음 정보에서 검색 및 확인됩니다.

- 클래스 내의 BusinessObjects 개체
- @Select 문을 통해 다른 BusinessObjects 개체의 Select 필드에 참조되는 BusinessObjects 개체
- 조인에 포함된 열

특성 관계

유니버스의 설명-차원 관계는 IBM DB2 Cube Views 의 기능적 종속성 형식의 특성 관계로 변환됩니다.

조인

조인과 그 속성은 유니버스 구조에서 직접 읽습니다.

측정값

모든 클래스는 계수 개체를 검색합니다. 같은 팩트 테이블에 계수를 작성하지 않은 경우에는 검색하지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

계층구조

DB2 Cube Views 의 계층구조가 차원 개체에 연결되며 계층구조의 모든 수준은 동일한 이 차원의 멤버가 됩니다. 사용자 지정 계층구조에 다른 Business Objects 클래스를 기반으로 하는 수준이 포함될 수 있는 유니버스의 경우에는 해당되지 않습니다. 계층구조는 다음과 같이 처리됩니다.

- 유니버스에서 기본 계층만 사용하는 경우 IBM DB2 Cube Views 로 내보낼 계층은 각 클래스에 포함된 개체 순서를 사용하여 개체에서 추론됩니다.
- 유니버스에 사용자 지정 계층이 있는 경우 수정하지 않고 내보냅니다.

9.7.3 DB2CV XML 파일로 유니버스 내보내기

다음과 같이 BusinessObjects 유니버스를 IBM DB2 Cube Views XML 파일로 내보낼 수 있습니다.

1. 파일 > 메타데이터 교환을 선택합니다. 메타데이터 교환 패널이 나타납니다.
2. 다음으로 유니버스 내보내기 드롭다운 목록 상자에서 IBM DB2 Cube Views 를 선택합니다. 확인을 클릭합니다. 내보내기 마법사가 시작됩니다. 다음을 클릭합니다. 유니버스 소스 파일 페이지가 나타납니다.
3. 유니버스 파일을 찾아 선택합니다. 다음을 클릭합니다. OLAP 정보 페이지가 나타납니다.
4. 팩트 테이블의 이름을 입력하거나 기본 팩트 테이블 이름을 사용할 수 있습니다. 스키마의 이름을 입력합니다. 다음을 클릭합니다. 메타데이터가 로드됩니다. 내보낼 구조를 보여 주는 페이지가 나타납니다. 다음을 클릭합니다.
5. XML 파일의 이름을 입력하고 다음을 클릭합니다. 요약 페이지가 나타납니다. 내보내기 정보가 올바른지 확인합니다. 마침을 클릭합니다. 사용자 프로필의 유니버스 폴더에 XML 파일이 만들어집니다(예: C:\Documents and Settings\<사용자 이름>\Application Data\Business Objects\Business Objects 12.0\Universes).

9.7.4 유니버스와 DB2CV 메타데이터 간의 매핑

이 단원에서는 유니버스 구조와 IBM DB2 Cube Views 구조 사이의 매핑에 대해 자세히 설명합니다.

다음 단원에서는 유니버스를 XML 파일로 내보낼 때 유니버스에서 IBM DB2 Cube Views 로 매핑되는 구조에 대해 자세히 설명합니다.

유니버스와 큐브 모델

다음 표에서는 유니버스와 큐브 모델의 매핑에 대해 설명합니다.

표 227:

유니버스 속성	큐브 속성
약식 이름(파일 이름)	큐브 파일 이름

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스 속성	큐브 속성
유니버스 이름(긴 이름)	비즈니스 이름 기본적으로 이름은 유니버스 약식 이름(<유니버스 약식 이름>) 입니다.
설명	주석
팩트 테이블 이름	factsRef
클래스 목록	dimensionRef
팩트 테이블이 포함된 조인 목록	joinRef

클래스와 차원

다음 표에서는 클래스와 차원 매핑에 대해 설명합니다.

표 228:

클래스 속성	차원 속성
이름	이름 및 비즈니스 이름
설명	주석
차원 및 설명 개체 목록 계수 매핑에 대해서는 아래의 "계수와 계수" 표를 참조하십시오.	attributeRef
클래스에서 유추되는 차원 테이블 간의 조인	joinRef
계층구조	사용자 지정 계층구조인 경우 IBM DB2 Cube Views 에서 필요한 동일한 차원의 모든 계층구조 수준을 얻기 위해 차원이 수정됩니다. 계층은 herarchyRef 속성에 저장됩니다.

팩트 테이블과 팩트

다음 표에서는 팩트 테이블과 팩트 매핑에 대해 설명합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 229:

팩트 테이블 속성	팩트 속성
팩트 테이블 이름 유니버스 내보내기 패널의 팩트 상자에 이 이름을 직접 입력합니다. 기본 이름 Facts_<유니버스 이름>을 사용할 수도 있습니다.	이름 및 비즈니스 이름
테이블 설명	주석
유니버스의 모든 계수 목록	measureRef
계수에 참조되는 열 및 개체 목록	attributeRef

계수와 계수

다음 표에서는 계수와 계수 매핑에 대해 설명합니다.

표 230:

계수 속성	계수 속성
이름	이름 및 비즈니스 이름
설명	주석
Select 문에서 유추되는 열 및 개체	sqlExpression 열
Select 문 수식	sqlExpression 템플릿
집계 함수	집계 함수

차원 및 설명 개체와 특성

다음 표에서는 차원 및 설명과 특성 매핑에 대해 설명합니다.

표 231:

차원 및 설명 개체	특성
이름	이름 및 비즈니스 이름
설명	주석
Select 문에서 참조하는 열 및 개체	sqlExpression 열
Select 문 수식	sqlExpression 템플릿

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

차원 및 설명 관계와 특성 관계

다음 표에서는 차원/설명 관계와 특성 관계 매핑에 대해 설명합니다.

표 232:

차원/설명 관계	특성 관계
차원 이름 + 설명 이름 연결 문자: "_"	이름 및 비즈니스 이름
차원	왼쪽 특성
설명	오른쪽 특성

기본 계층구조와 계층구조

다음 표에서는 기본 계층과 계층 매핑에 대해 설명합니다.

표 233:

기본 계층구조	계층구조
이름	이름 및 비즈니스 이름
개체 목록. 설명 개체는 계층구조의 일부일 수 없습니다.	AttributeRef

노트

사용자 지정 계층구조가 없으면 클래스가 계층구조로 사용됩니다.

사용자 지정 계층구조와 계층구조

다음 표에서는 사용자 지정 계층구조와 계층구조 매핑에 대해 설명합니다.

표 234:

사용자 지정 계층구조	계층구조
이름	이름 및 비즈니스 이름
개체 목록	attributeRef

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조인과 조인

다음 표에서는 조인과 조인 매핑에 대해 설명합니다.

표 235:

조인	조인
왼쪽 테이블 이름 + 오른쪽 테이블 이름 연결 문자: "_"	이름 및 비즈니스 이름
왼쪽 열	왼쪽 특성
오른쪽 열	오른쪽 특성
복합 식: 각 단순 식의 경우 왼쪽 열과 오른쪽 열이 동일합니다.	각 단순 식은 특성 쌍에 매핑합니다.

9.7.5 특정 SQL 식 매핑

특정 SQL 식은 내보내기 과정에서 특별한 방식으로 매핑됩니다. 이 단원에서는 다음과 같은 경우의 SQL 식에 대해 설명합니다.

- 계수의 SELECT 식
- @AggregateAware 함수
- 복합 조인 식
- 테타 조인
- 바로 가기 조인

계수의 SELECT 식

BusinessObjects UMB 는 계수의 SELECT 에서 다음 정보를 가져옵니다.

- 계수에 포함된 테이블과 열을 검색하여 sqlExpression:column 에 매핑합니다.
- 집계 함수를 확인합니다.
- 수식을 확인하여 sqlExpression:template 에 매핑합니다.

@AggregateAware 함수

개체에 @AggregateAware 함수가 포함되어 있으면 @AggregateAware 함수의 마지막 매개 변수만 적용됩니다. 이 식은 함수에 사용되는 최하위 수준의 집계가 포함된 식입니다. 예를 들면 다음과 같습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

유니버스의 계수 식에 대한 @AggregateAware 식은 다음과 같습니다.

```
@Aggregate_Aware(
sum(AggregatedTable1.Sales_revenue),
sum(AggregatedTable2.Sales_revenue),
sum(Fact_Table.Amount_sold))
```

IBM DB2 Cube Views 에 매핑되는 식은 다음과 같습니다.

```
sum(Fact_Table.Amount_sold))
```

복합 조인 식

유니버스의 복합 조인 식은 다음 형식의 식으로 구성될 수 있습니다.

```
LeftTable.Column=RightTable.Column
```

복합 조인에서 이러한 형식의 식은 AND 연산자로 연결할 수 있습니다. BusinessObjects UMB 는 복합 조인의 각 식을 같은 조인 안의 IBM DB2 Cube Views 특성 쌍에 매핑합니다.

테타 조인

테타 조인은 IBM DB2 Cube Views 조인 두 개로 나뉩니다. 여기서 연산자 BETWEEN 은 <= 및 >= 연산자로 바뀝니다. 예를 들면 다음과 같습니다.

유니버스의 조인은 다음 식을 사용합니다.

```
Customer.age between Age_group.age_min and Age_group.age_max
```

이 조인은 다음 식을 사용하여 두 조인으로 나뉩니다.

```
Join1: Customer.age >= Age_group.age_min
Join2:  Customer.age <= Age_group.age_max
```

바로 가기 조인

IBM DB2 Cube Views 로 내보내지 않습니다. 유니버스의 바로 가기 조인은 중간 테이블을 고려하지 않으므로 쿼리의 성능을 향상시키는 다른 경로를 나타냅니다. 바로 가기 조인은 큐브 모델에 루프를 생성하므로 내보낼 수 없습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

9.8 Oracle Analytic Workspaces

Oracle OLAP Universe Builder 마법사는 유니버스 작성 단계를 안내합니다. 메타데이터 교환 패널(파일 > 메타데이터 교환)에서 Oracle OLAP Universe Builder 마법사에 연결합니다.

Oracle Universe Builder 마법사로 유니버스를 만드는 방법을 요약하면 다음과 같습니다.

메타데이터 교환을 시작하고 다음에서 유니버스 만들기 드롭다운 목록에서 Oracle OLAP 을 선택합니다.

Oracle OLAP Universe Builder 마법사가 시작됩니다. 다음 단계를 수행합니다.

- 유니버스를 작성하는 데 사용할 정보 제공자에 연결합니다.
- 데이터베이스를 선택합니다.
- 대상 메타데이터 소스에 해당하는 큐브를 선택합니다.
- 큐브 메타데이터를 기반으로 뷰를 만듭니다.
- 뷰를 기반으로 유니버스를 생성합니다.

기존 뷰에서 유니버스를 만들 수도 있습니다.

9.8.1 OLAP 큐브에서 유니버스를 생성하는 방법

Oracle OLAP Universe Builder 로 유니버스를 만들 때 Oracle Analytic Workspaces 에 대한 SQL 액세스가 자동으로 설정됩니다. BusinessObjects Oracle OLAP Universe Builder 는 다음과 같은 주요 작업을 수행합니다.

- 유니버스의 관계형 팩트 뷰를 실제 뷰나 파생 테이블로 삽입합니다.
- 차원 수준 및 계층을 나타내기 위한 별칭을 추가합니다.
- 관계형 뷰를 일반 조인 및 바로 가기 조인을 사용하여 차원 테이블에 조인합니다. 조인의 식은 이 솔루션에만 사용됩니다.
- 각 큐브 차원에 대해 개체 클래스를 만들고 차원의 각 수준에 대해 개체를 만듭니다.
- 차원에 계층구조가 둘 이상 있을 경우 각 계층구조에 대해 하위 클래스를 만듭니다. 다중 계층구조 차원은 뷰 정의와 유니버스에서 사용할 수 있습니다.
- 다중 계층구조 차원에서 발생하는 개체 비호환성을 해결할 집계 탐색을 정의합니다.
- 집계 탐색을 처리하기 위해 AggregateAware 함수를 사용하여 개체 식을 정의합니다.
- 실제 차원 멤버(식별자)를 매핑하는 개체를 멤버 설명을 나타내는 개체의 설명으로 변환합니다.
- 계수 개체를 만듭니다.

9.8.2 Oracle OLAP 구조를 유니버스 구성 요소에 매핑

이 단원에서는 Oracle OLAP 큐브 구조에서 유니버스를 만드는 방법에 대해 설명합니다. 생성되는 유니버스 구조를 설명하고 매핑 프로세스에 대한 몇 가지 일반적인 질문에 대한 대답을 제공합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

9.8.3 관계형 뷰 분석

BusinessObjects Oracle OLAP Universe Builder 는 OLAP_TABLE 함수를 호출하여 뷰 열을 차원의 계층과 큐브의 계수에 매핑하는 뷰를 생성합니다. 생성되는 스크립트의 형식은 다음과 같습니다.

```
CREATE VIEW BOBJ_FK_UNITS_CUBE_VIEW AS SELECT * FROM
TABLE(OLAP_TABLE('GLOBAL_AW2.TEST DURATION session','','','&LIMIT_MAP'
```

LIMIT_MAP 은 OLAP_TABLE 의 limit_map 매개 변수 텍스트를 저장하는 변수입니다. 이 텍스트는 Oracle OLAP Universe Builder 에서 생성됩니다. 다음은 limit_map 매개 변수의 예입니다.

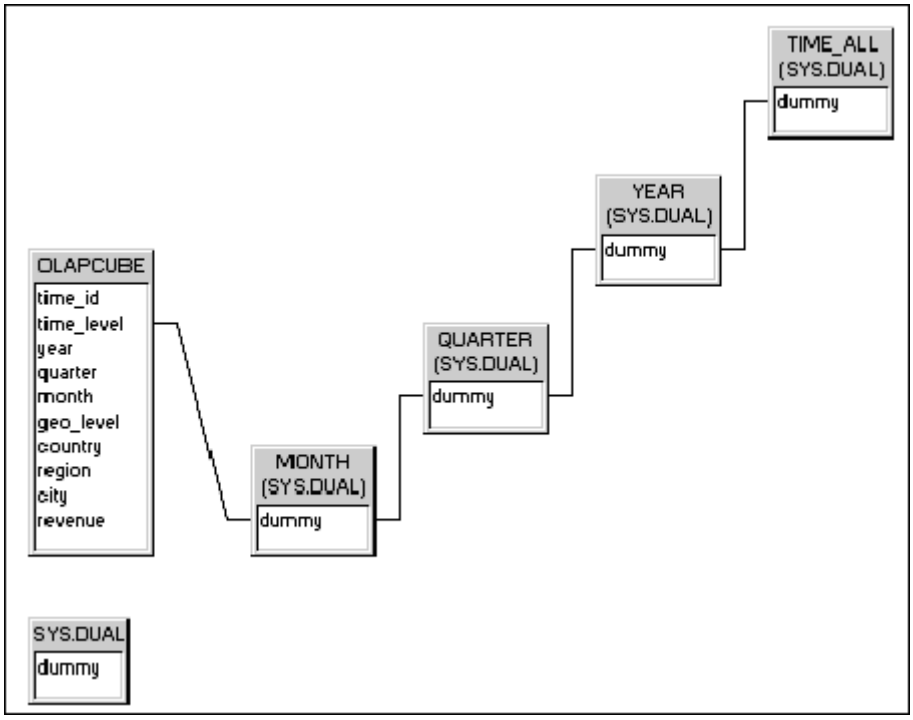
```
DIMENSION GLOBAL_AW2.TEST!FK_TIME WITH
    HIERARCHY GLOBAL_AW2.TEST!FK_TIME_PARENTREL (FK_TIME_HIERLIST \''CALENDAR\'')
        LEVELREL FK_TIME_YEAR,FK_TIME_QUARTER,FK_TIME_MONTH
        FROM GLOBAL_AW2.TEST!FK_TIME_FAMILYREL USING GLOBAL_AW2.TEST!
FK_TIME_LEVELLIST
        LEVELREL FK_TIME_YEAR_DESC,FK_TIME_QUARTER_DESC,FK_TIME_MONTH_DESC
        FROM GLOBAL_AW2.TEST!FK_TIME_FAMILYREL USING GLOBAL_AW2.TEST!
FK_TIME_LEVELLIST
        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_LEVELLIST
        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_LEVELLIST
        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_LEVELLIST
        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_LEVELLIST
        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
```

9.8.4 유니버스의 바로 가기 조인의 용도

바로 가기 조인은 BusinessObjects 에서 각 개체보다는 각 개체 조합에 대해 SQL 을 생성하도록 합니다.

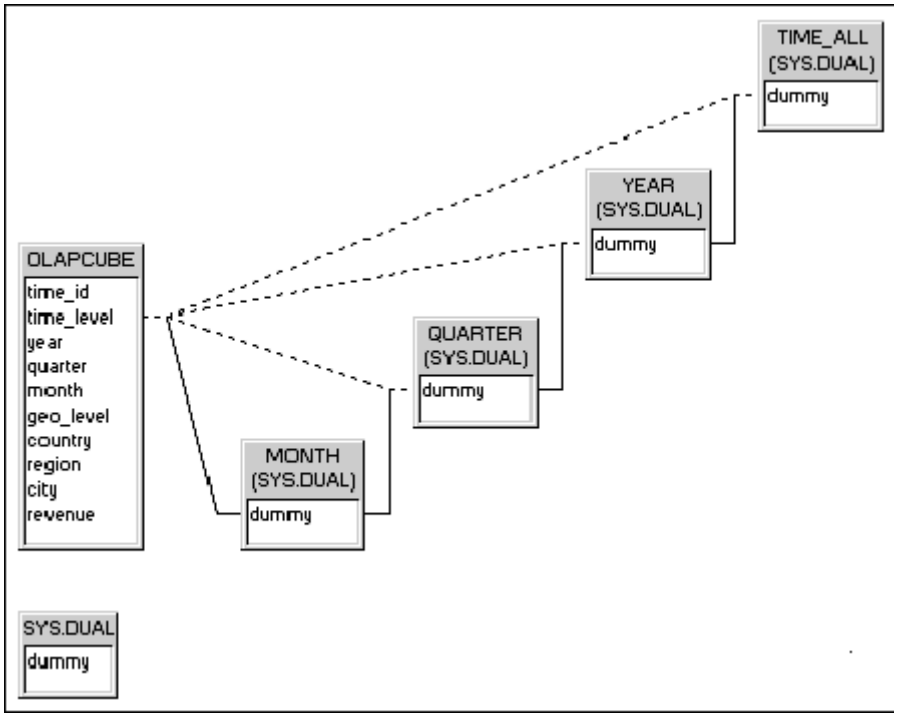
BusinessObjects 는 쿼리에서 테이블을 생략하고도 계층구조에서 직접 연결되지 않는 두 테이블 간에 '바로 가기'를 만들 수 있는 경우에 바로 가기 조인을 사용합니다. 예를 들어, 다음과 같은 스키마를 기반으로 하는 경우

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



QUARTER 테이블과 OLAPCUBE 테이블 간에 바로 가기 조인이 정의되어 있으면 BusinessObjects 는 분기마다 수익을 검색하기 위해 MONTH 테이블을 통해 조인할 필요가 없습니다.

최저 수준의 테이블을 제외하고 시간 계층구조의 각 테이블은 아래 그림과 같이 바로 가기 조인을 통해 OLAPCUBE.time_level 에 조인되어야 합니다.



© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조인 식에는 OLAPCUBE 에서 반환되는 행을 제한하는 식이 포함되어 있어야 하며, QUARTER 의 경우 OLAPCUBE.time_level = 'QTR'이 됩니다. 유니버스 디자인 도구가 조인을 허용하도록 하려면 식이 설명 안에 나타나는 MONTH 테이블을 참조해야 합니다. 생성하려는 실제 조인 식에서 어떠한 역할도 하지 않기 때문입니다. 따라서 전체 조인 식은 다음과 같습니다.

```
/* QUARTER.DUMMY */ OLAPCUBE.time_level = 'QTR'
```

예제 시간 계층구조에 대한 바로 가기 조인 식의 전체 목록은 다음과 같습니다.

표 236:

조인된 테이블	식
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'

9.8.5 유니버스 구성 요소에 **Oracle OLAP** 구조를 매핑하는 방법

원하는 유니버스를 얻고 설정하기 위해 Oracle OLAP Universe Builder 는 다음과 같이 유니버스 개체를 추가하고 구성합니다.

뷰

Oracle OLAP Universe Builder 는 관계형 뷰를 oracle 테이블 sys.dual 및 유니버스 테이블로 삽입합니다. 파생 테이블을 사용하도록 선택하면 파생 테이블이 뷰의 정의와 함께 삽입됩니다(OLAP_TABLE 함수로 부분 선택).

계층구조 테이블

관계형 뷰에 표시되는 각 계층구조의 경우 계층구조의 각 수준마다 sys.dual 의 별칭이 만들어집니다. 이 별칭의 이름은 수준 이름에 해당합니다. 예를 들어, TIME 차원에 4 개의 수준(ALL, YEAR, MONTH, QUARTER)이 있는 경우 ALL, YEAR, MONTH, QUARTER 라는 4 개의 별칭을 만듭니다.

다중 계층구조 테이블

> **노트**
>
> 다중 계층구조는 특수한 경우입니다. 자세한 내용은 이 장의 마지막 단원을 참조하십시오.

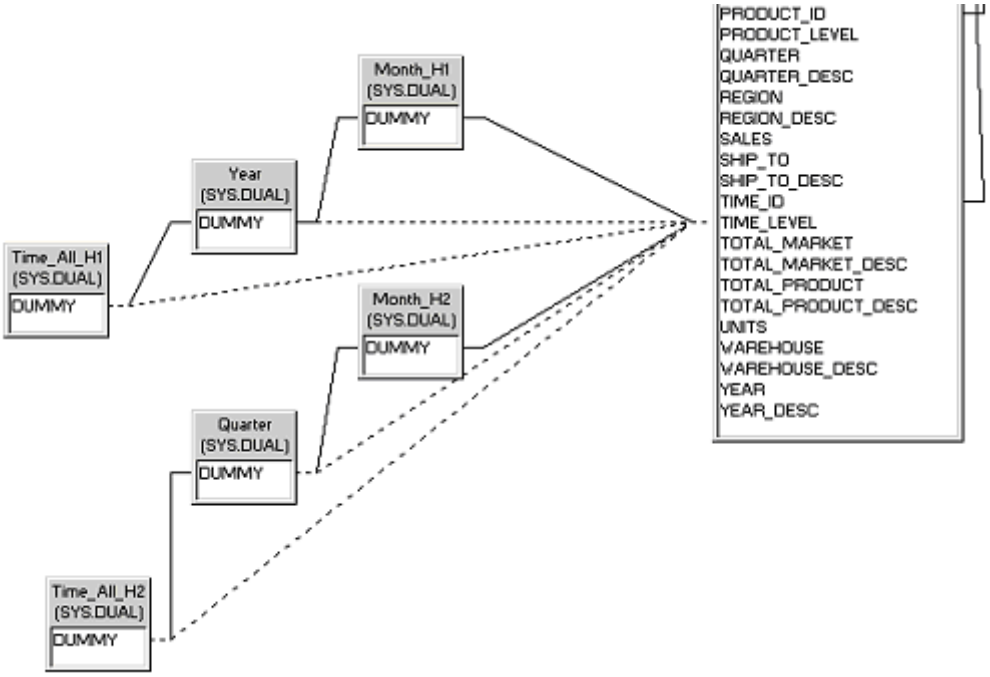
© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

차원에 계층구조가 둘 이상 있는 경우 일부 계층구조가 같은 수준을 공유하더라도 각 계층구조마다 다른 테이블 집합이 만들어집니다. 이 사실은 공유 수준에 대해 계층 수만큼 별칭이 만들어짐을 의미합니다. 그러한 별칭의 이름은 수준 이름과 계층구조 이름을 연결하여 지정됩니다. 예를 들면 다음과 같습니다.

차원 시간에 H1 (All_Time, Year, Month) 및 H2 (All_Time, Quarter, Month) 계층이 있습니다.

All_Time 과 Month 는 두 계층에서 공유되므로 All_Time 에 대해 All_Time _H1 과 All_Time_H2 라는 두 개의 별칭을 갖습니다.

그리고 Month 의 별칭은 Month _H1 과 Month _H2 입니다.



차원 조인

- 수준을 나타내는 각 테이블은 같은 계층구조에서 바로 아래 수준에 조인됩니다. 조인 식은 다음과 같습니다.
  /* Alias1.DUMMY=Alias2.DUMMY */ 1=1
  여기서 Alias1 은 수준을 나타내고 Alias2 는 계층구조에서 바로 위 수준을 나타냅니다. 예를 들면 다음과 같습니다.
  /* Quarter.DUMMY=Year.DUMMY */ 1=1
- 각 테이블은 조인이 일반 조인인 가장 낮은 수준을 제외하고 바로 가기 조인 형식을 사용하여 뷰에 조인됩니다. 조인 식은 값을 정의하여 뷰에서 반환되는 행을 필터링하며 형식은 다음과 같습니다.
  /* Alias.DUMMY */
  VIEW.levelColumn = 'level_value'
  여기서 Alias 는 별칭 이름이고, levelColumn 은 뷰 내에 있는 수준을 나타내는 열이며, level_value 는 해당 열의 값인데 수준 이름과 일치합니다.

예: MYVIEW 는 OLAP 큐브를 나타내는 뷰이고, 수준이 포함된 열은 time_level 이며, 수준 값은 ALL, YEAR, QTR, MONTH 입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

조인된 테이블 식

```
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'
```

클래스 및 개체 매핑

브리지는 OLAP 차원마다 클래스를 만들고 수준마다 개체를 만듭니다. 클래스 및 개체 속성은 다음과 같이 매핑됩니다.

표 237:

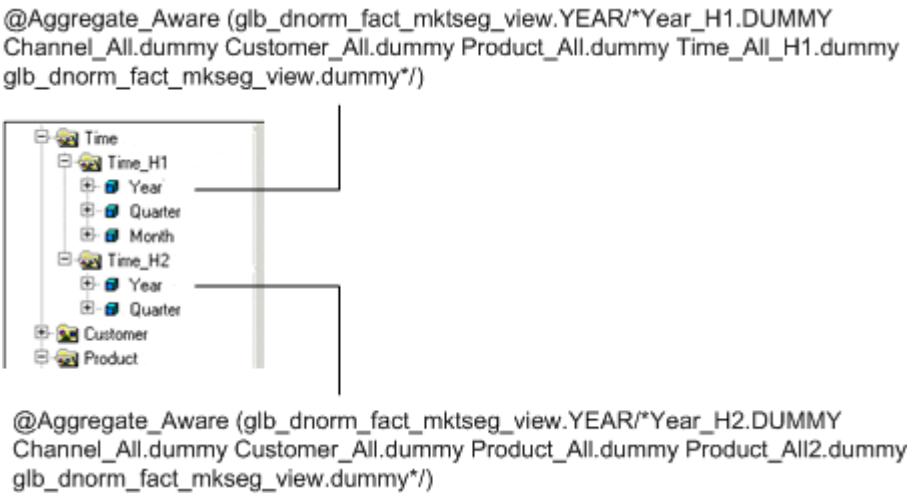
유니버스 항목	속성	OLAP 항목에서 매핑됨...
클래스 이 장의 뒷부분에 있는 특수한 경우를 참조하십시오.	이름	차원 이름
	개체	수준 및 특성
차원	이름	뷰의 필드 이름으로 식별되는 현재 수준 이름입니다.
	선택	뷰 필드(예: MYVIEW.YEAR)
	테이블	적합한 조인을 사용하도록 하는 추가 테이블 • 현재 뷰 테이블 • 뷰 • 모든 차원 중 최고 수준의 테이블
설명(옵션)	상위 차원 참고: 모든 다른 속성은 위의 차원과 동일합니다.	ID 필드와 관련된 설명 필드에서 만든 차원 개체입니다. 예를 들어, ID 필드는 YEAR 이고 설명 필드는 YEAR_DESC 입니다. YEAR 개체는 YEAR_DESC 개체에 대한 설명입니다.
계수	이름	큐브의 계수 이름
	선택(드릴스루 없음)	뷰 필드(예: MYVIEW.SALES)
	테이블	적합한 조인을 사용하도록 하는 추가 테이블 • 뷰 • 모든 차원 중 최고 수준의 테이블
	집계 함수	없음

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

다중 계층구조의 특수한 경우(다중 계층구조 차원 지원)

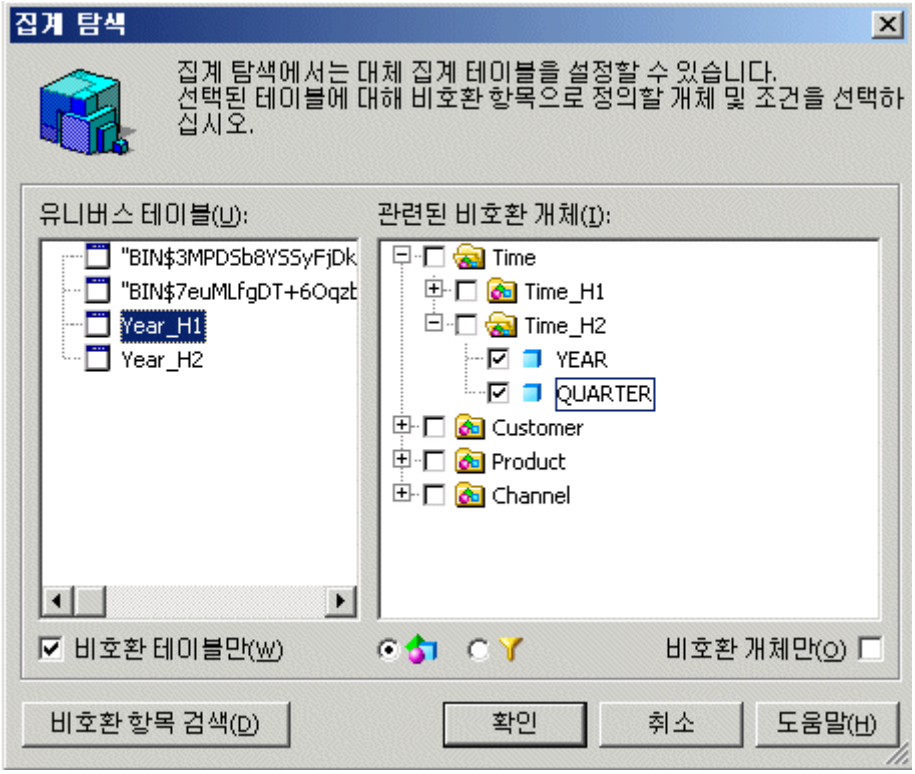
다중 계층구조 차원을 지원하려면 유니버스에서 다음 작업을 수행합니다.

- 이 단원의 앞부분에 있는 설명에 따라 각 계층구조에 대해 테이블 집합을 만듭니다.
- 계층구조마다 차원과 하위 클래스에 대해 클래스가 만들어집니다. 하위 클래스의 이름은 차원 이름과 계층구조 이름을 연결하여 지정됩니다.
- 계층구조 내의 수준에 해당하는 각 개체마다 Select 식에 집계 함수가 추가됩니다. 현재 계층구조의 별칭을 제외하고 최고 수준의 모든 별칭을 주석으로 참조합니다. 예:



- 하위 클래스(계층구조)의 개체를 다른 계층구조에 해당하는 테이블과 호환되지 않도록 집계 탐색을 설정합니다. 이렇게 하면 최종 사용자가 다른 계층에 속해 있는 수준을 나타내는 보고서 개체에 사용할 수 없게 됩니다.
  예를 들어, 계층구조 H1 의 Year_H1 테이블은 계층구조 H2 의 개체와 호환되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



H2 계층구조의 Year_H2 테이블은 H1 계층구조의 개체와 호환되지 않습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

9.8.6 뷰 만들기 및 유니버스 생성

먼저 Analytic Workspace 큐브 메타데이터를 사용하여 뷰를 정의하여 Oracle OLAP 유니버스를 생성한 다음 유니버스 만들기 옵션을 설정하고 새로운 유니버스를 생성합니다.

9.8.7 Oracle Analytic Workspace 에서 유니버스와 뷰를 만드는 옵션

유니버스와 뷰를 다음과 같이 만들 수 있습니다.

표 238:

유니버스 작성 옵션	설명
뷰를 만들고 유니버스 생성	뷰를 만든 다음 유니버스를 만들기 위해 매핑할 구조를 선택합니다.
Oracle Analytical Workspace 에서 뷰만 만들기	유니버스를 만들지 않으려는 경우 뷰를 만들고 저장할 수 있습니다. 뷰는 뷰 목록에 있으며 언제라도 유니버스를 만드는 데 사용할 수 있습니다.
기존 뷰에서 유니버스 생성	기존 뷰를 선택하고 이 뷰에서 직접 유니버스를 생성합니다.

관련 정보

뷰 만들기 및 유니버스 생성 [페이지 481]
Oracle Analytical Workspace 에 뷰만 만들기 [페이지 483]
기존 뷰에서 유니버스 생성 [페이지 483]

9.8.8 뷰 만들기 및 유니버스 생성

먼저 Analytic Workspace 큐브 메타데이터를 사용하여 뷰를 정의하여 Oracle OLAP 유니버스를 생성한 다음 유니버스 만들기 옵션을 설정하고 새로운 유니버스를 생성합니다.

뷰를 만들고 유니버스를 생성하려면

1. 파일 > 메타데이터 교환을 선택합니다.

   메타데이터 브리지 패널이 나타납니다.

   다음에서 유니버스 만들기 드롭다운 목록에서 Oracle OLAP 을 선택합니다.

   Oracle OLAP Universe Builder 마법사가 시작됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

2. 뷰 만들기 및 유니버스 생성을 선택하고 다음을 클릭합니다.
3. 연결을 선택하고 사용자 이름 및 암호를 입력한 다음 마침을 클릭합니다.

   연결에 사용할 수 있는 Analytic Workspace 큐브가 표시된 Analytic Workspaces(AW) 큐브 패널이 나타납니다.
4. 큐브 노드를 클릭합니다.

   연결에 사용할 수 있는 Analytic Workspace(AW)가 표시됩니다.
5. AW 노드를 확장하여 AW 에 사용할 수 있는 큐브를 표시합니다.
6. 큐브를 선택하고 다음을 클릭합니다.
7. 선택한 큐브에서 메타데이터를 로드하는 작업의 진행률이 상태 상자에 표시됩니다.

   뷰 만들기 페이지가 나타납니다. 이 페이지에 큐브에 사용할 수 있는 차원과 계수가 나열됩니다.
8. 필요한 경우 데이터 형식 및 길이 값을 수정합니다. 다음과 같이 하십시오.

   * 데이터 형식 또는 길이 값을 두 번 클릭합니다.

   * 드롭다운 목록 상자에서 데이터 형식을 선택합니다.
9. 다음을 클릭합니다.

   계층구조 수준 페이지가 나타납니다. 이 페이지에는 데이터 형식 및 값과 함께 계층구조 수준이 나열됩니다.
10. 필요한 경우 계층구조 값을 편집한 후 다음을 클릭합니다.

    뷰 및 유니버스 속성 페이지가 나타납니다.
11. 뷰의 이름을 입력하고 뷰 및 유니버스 옵션을 선택합니다. 뷰 속성과 유니버스 옵션은 다음과 같습니다.

    뷰 이름 보기: 이 필드를 편집할 수 있습니다.

    OLAP_EXPRESSION 에 대한 열 만들기: 선택하면 유니버스에서 OLAP_EXPRESSION 함수를 사용할 수 있도록 뷰에서 Raw(32) 유형의 열이 추가로 만들어집니다.

    식별자에 대한 열 만들기: 선택하면 차원 멤버(식별자)를 나타내는 열이 만들어집니다.

    기존 데이터베이스 개체 바꾸기: 선택하면 데이터베이스에서 기존의 Type 및 View 구조가 대체됩니다.

    파생 테이블 사용: 선택하면 데이터베이스에 실제로 만든 뷰에서 유니버스가 생성되지 않고 큐브 구조를 참조하기 위한 파생 테이블을 사용하여 생성됩니다. 파생 테이블은 데이터베이스 구조를 참조하는 유니버스에만 있는 가상 테이블입니다. 데이터베이스에서는 만들어지지 않습니다. 사용자에게 뷰 만들기 권한이 없거나 데이터베이스에 뷰를 누적시키지 않으려는 경우에 유용합니다. 파생 테이블 사용에 대한 자세한 내용은 유니버스 디자인 도구 사용자 가이드를 참조하십시오.

    개체 *ID* 를 설명으로 변환: 뷰에 대해 식별자 열 만들기 옵션을 선택한 경우에만 활성화됩니다. 선택하면 생성된 유니버스에서 개체 ID 가 설명 개체로 변환됩니다.
12. 다음을 클릭합니다.

    SQL 확인 페이지가 나타납니다.
13. 뷰에 대한 SQL 을 확인하고 다음을 클릭합니다.

    유니버스 정보 요약 페이지가 나타납니다.
14. 유니버스 정보를 확인하고 마침을 클릭합니다.

    유니버스 디자인 도구가 시작되고 새로 생성된 유니버스가 열립니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

9.8.9 Oracle Analytical Workspace 에 뷰만 만들기

Analytic Workspace 큐브 메타데이터를 사용하여 뷰를 만들 수 있습니다. 저장한 뷰가 뷰 목록에 나타납니다. 뷰를 만들면 나중에 뷰를 선택하고 유니버스를 생성할 수 있습니다. 뷰만 만들려면 뷰 만들기 및 유니버스 생성 [페이지 481] 단원에 나오는 절차를 그대로 수행하되 Oracle OLAP Universe Builder 마법사의 시작 부분에서 뷰만 만들기 라디오 단추를 선택하십시오.

대상 데이터베이스에 뷰가 만들어집니다. 언제든지 이 뷰에 연결하여 유니버스를 만들 수 있습니다. 뷰를 사용하여 유니버스를 만드는 절차에 대해서는 기존 뷰에서 유니버스 생성 [페이지 483] 단원을 참조하십시오.

9.8.10 기존 뷰에서 유니버스 생성

기존 뷰에서도 유니버스를 생성할 수 있습니다. 기존 뷰는 목록에 나타나 있습니다. 이 목록에서 뷰를 선택하고 아래의 절차를 따라 유니버스를 생성합니다.

1. 파일 > 메타데이터 교환을 선택합니다. 메타데이터 교환 패널이 나타납니다. '다음에서 유니버스 만들기' 드롭다운 목록에서 Oracle OLAP 을 선택하고 '확인'을 클릭합니다. Oracle OLAP Universe Builder 가 시작됩니다.
2. Oracle OLAP Universe Builder 마법사 시작 페이지에서 뷰에서 유니버스 생성 라디오 단추를 선택합니다. 다음을 클릭합니다. 연결 매개 변수 상자가 나타납니다.
3. 연결을 선택하고 사용자 이름 및 암호를 입력한 후 다음을 클릭합니다. 연결에 사용할 수 있는 Analytic Workspace 큐브를 보여 주는 큐브 패널이 나타납니다.
4. 큐브 노드를 클릭합니다. 연결에 사용할 수 있는 Analytic Workspace(AW)가 표시됩니다.
5. AW 노드를 확장하여 AW 에 사용할 수 있는 큐브를 표시합니다. 큐브를 선택하고 다음을 클릭합니다. 큐브에 정의된 사용 가능한 뷰 목록이 나타납니다.
6. 목록에서 뷰 이름을 클릭하고 다음을 클릭합니다. 선택한 큐브에서 메타데이터를 로드하는 작업의 진행률이 상태 상자에 표시됩니다. 유니버스 만들기 페이지가 나타납니다. 이 페이지에는 유니버스를 만드는 데 사용할 수 있는 뷰에 정의된 차원, 계수 및 계층구조 수준이 나열됩니다.
7. 필요한 경우 열 이름이나 계층구조 수준을 수정합니다. 다음과 같이 하십시오. 열 이름 또는 수준 값을 두 번 클릭합니다. 적합한 이름을 선택하거나 입력합니다.
8. 마침을 클릭합니다. 유니버스 디자인 도구가 시작되고 새로 생성된 유니버스가 열립니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

10 유니버스 배포

10.1 개요

이 장에서는 유니버스 배포 및 관리에 대해 다루며, 다음과 같은 내용에 대해 설명합니다.

- 유니버스 배포 방법 [페이지 484]
- 유니버스에 액세스 제한 설정 [페이지 485]
- 사용자 및 로그인 관리 [페이지 494]

10.2 유니버스 배포 방법

유니버스 배포는 Web Intelligence 사용자나 다른 디자이너가 유니버스를 사용할 수 있도록 만드는 작업입니다. 유니버스를 배포하려면 유니버스를 중앙 관리 서버(CMS) 리포지토리로 내보내야 합니다.

유니버스를 실제 운용 리포지토리에 내보내기 전에 테스트 리포지토리에 내보낸 다음 Web Intelligence 에서 테스트를 실행하여 유니버스를 테스트합니다.

유니버스는 디자인, 작성 및 테스트 단계를 모두 마친 경우에만 Web Intelligence 사용자에게 배포해야 합니다.

리포지토리에서 유니버스를 가져오고 리포지토리에 유니버스를 내보내어 유니버스를 배포하는 방법에 대한 자세한 내용은 다음 단원을 참조하십시오.

- 유니버스 가져오기 [페이지 40]
- 유니버스 내보내기 [페이지 41]

10.2.1 리포지토리에서 유니버스 식별

유니버스는 다음 매개 변수로 확인합니다.

표 239:

식별자	설명
파일 이름	최대 100 자이며 확장자는 .unv 입니다.
긴 이름	최대 35 자로 구성됩니다. 일반 사용자가 Web Intelligence 에서 유니버스를 확인하는 이름이므로 유니버스의 용도를 설명하는 이름이어야 합니다.
고유한 시스템 식별자	CMS 에서 할당되는 식별자입니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

10.2.1.1 유니버스 식별자 규칙

리포지토리의 유니버스 폴더에 저장되는 유니버스에 대한 유니버스 식별자에는 다음과 같은 규칙이 적용됩니다.

- 유니버스 식별자는 CMS 에서 중복되지 않아야 합니다.
- 파일 이름과 폴더 위치(경로)의 조합입니다. 유니버스는 상위 폴더에서 중복되지 않아야 합니다.

10.3 모든 사용자에게 유니버스에 대한 액세스 권한 부여

작업 그룹 모드에서 유니버스를 저장하여 작업 그룹 모드와 엔터프라이즈 모드 모두에서 모든 유니버스 디자인 도구 사용자가 유니버스에 액세스하도록 할 수 있습니다. 유니버스에 대한 연결은 보안이 적용된 연결이 될 수 없습니다. 모든 사용자가 유니버스를 사용할 수 있도록 하려면 보안이 적용되지 않은 연결을 사용하여 유니버스를 저장해야 합니다.

모든 유니버스 디자인 도구 사용자가 유니버스에 액세스할 수 있도록 하려면

1. 모든 사용자에게 액세스를 허용할 유니버스가 보안이 설정된 연결을 사용하지 않도록 합니다.
2. 유니버스를 CMS 에 내보내려는 경우에는 보안 연결이 필요합니다. 유니버스의 연결에 보안이 설정되어 있으면 새 공유 연결을 선택하거나 만듭니다. 자세한 내용은 유니버스 식별 매개 변수 수정 [페이지 72] 단원을 참조하십시오.
3. 파일 > 다른 이름으로 저장 을 선택합니다.
   다른 이름으로 저장 대화 상자가 나타납니다.
4. 모든 사용자에 대해 저장 확인란을 선택합니다.
5. 확인을 클릭합니다.

10.4 유니버스에 액세스 제한 설정

유니버스를 사용하는 정의된 사용자 및 그룹에 제한을 적용할 수 있습니다.

유니버스 보안은 두 가지 수준에서 관리됩니다.

표 240:

보안 수준	설명
CMS	CMS 에 저장된 유니버스에 적용할 제한 사항을 중앙 관리 콘솔에서 설정할 수 있습니다. 사용자 그룹에 정의된 권한에 따라 사용자가 액세스할 수 있는 유니버스를 설정할 수 있고 유니버스의 보기, 편집, 삭제 및 기타 작업을 제한할 수 있습니다. 이 가이드는 **CMS** 수준의 설정 제한에 관해 다루지 않습니다. 중앙 관리 시스템의 사용 방법에 대한 자세한 내용은 **BusinessObjects Enterprise** 관리자 가이드를 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

보안 수준	설명
유니버스	유니버스를 사용하도록 허용된 사용자에 대한 제한 사항을 정의할 수 있습니다. 제한에는 개체 액세스, 행 액세스, 쿼리 및 SQL 생성 제어, 연결 제어가 포함될 수 있습니다. 이 가이드에서는 유니버스에 정의할 수 있는 제한 유형에 대해 설명합니다.

10.4.1 제한이란?

제한은 유니버스에 적용되는 명명된 제한 그룹입니다. 유니버스에 대해 선택된 그룹이나 사용자 계정에 제한을 적용할 수 있습니다. 사용자가 유니버스에 연결하면 사용자에게 적용된 제한에 따라 유니버스에서 사용할 수 있는 개체, 행, 쿼리 유형 및 연결이 결정됩니다.

제한을 BusinessObjects 사용자나 그룹에 할당합니다. 이렇게 하면 사용자 그룹의 프로필을 기반으로 유니버스 개체나 리소스에 대한 액세스가 제한됩니다.

10.4.2 유니버스에서 적용할 수 있는 제한

사용자 그룹에 적용되는 액세스 제한은 제한에 정의됩니다. 유니버스에 대해 여러 제한을 정의할 수 있습니다. 제한은 언제든지 편집하거나 삭제할 수 있습니다.

제한은 다음과 같은 유형의 제한을 정의하여 유니버스에 대해 선택한 사용자나 그룹에 적용할 수 있습니다.

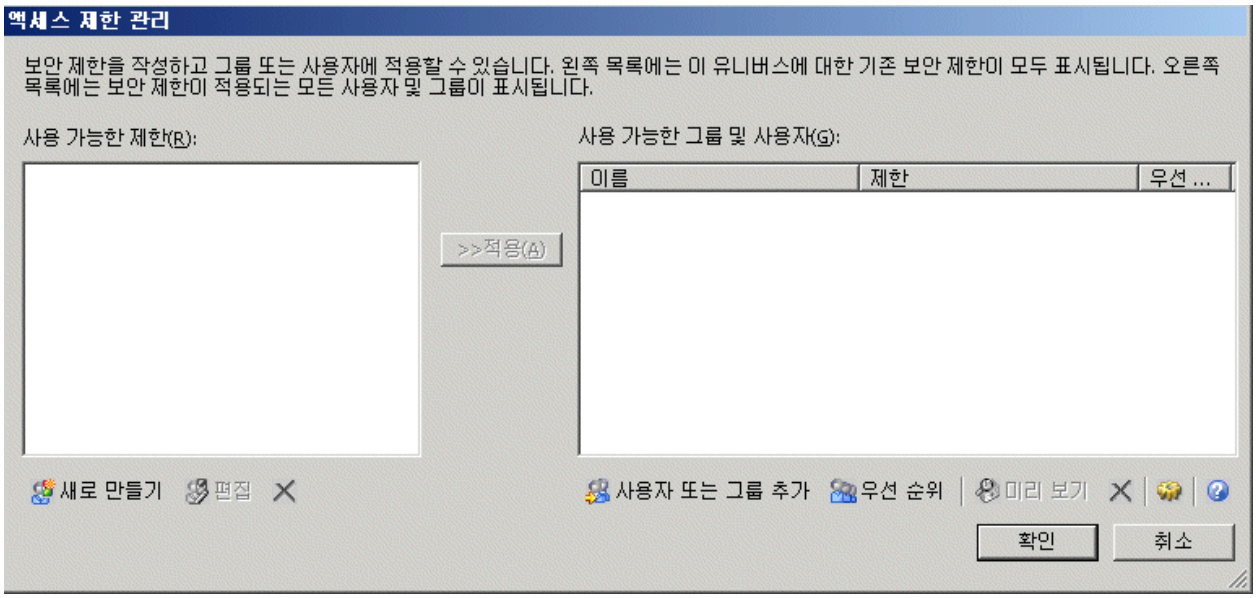
표 241:

제한 유형	설명
연결	데이터 소스에 대한 유니버스 연결입니다. 유니버스에 대한 대체 연결을 선택할 수 있습니다. 연결을 만들고 편집하는 방법에 대한 자세한 내용은 유니버스 식별 매개 변수 수정 [페이지 72] 단원을 참조하십시오.
쿼리 제어	결과 집합의 크기와 쿼리 실행 시간을 제한하는 옵션입니다. 자세한 내용은 SQL 제한 지정 [페이지 86] 단원을 참조하십시오.
SQL 생성 옵션	쿼리에 대해 생성되는 SQL 을 제어하는 옵션입니다. 자세한 내용은 SQL 생성 매개 변수 설정 [페이지 88] 단원을 참조하십시오.
개체 액세스	유니버스에 사용하지 못하도록 할 개체를 선택할 수 있습니다.
행 액세스	행에 대한 액세스를 제한하고 쿼리에서 반환되는 결과 집합을 제한하는 WHERE 절을 정의할 수 있습니다.
대체 테이블 액세스	유니버스에 참조되는 테이블을 데이터베이스의 다른 테이블로 바꿀 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

10.4.3 액세스 제한을 관리하는 방법

액세스 제한은 액세스 제한 관리 대화 상자에서 관리합니다. 도구 > 보안 관리 > 액세스 제한 관리를 선택하면 이 대화 상자에 액세스할 수 있습니다. 다음과 같이 대화 상자가 나타납니다.



유니버스에 현재 사용할 수 있는 제한은 사용 가능한 제한 창에 나열됩니다.

각 제한에 대해 정의된 사용자 및 사용자 그룹은 사용 가능한 그룹 및 사용자 창에 나타납니다.

다음은 액세스 제한을 관리하는 데 사용할 수 있는 옵션에 대한 설명입니다.

표 242:

제한 옵션	설명
새로 만들기	새 제한을 정의합니다.
편집	기존 제한을 수정합니다.
선택한 제한 삭제	목록에서 제한을 제거합니다.
사용자 또는 그룹 추가	BusinessObjects 시스템에 정의된 BusinessObjects 사용자 및 그룹 목록의 사용자나 그룹을 추가합니다.
우선 순위	하나 이상의 사용자 그룹의 우선 순위를 설정할 수 있습니다.
미리 보기	BusinessObjects 시스템에 대해 정의된 모든 사용자 및 그룹을 볼 수 있습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

제한 옵션	설명
선택한 사용자 또는 그룹에서 보안 옵션 제거	선택한 사용자 또는 그룹에 설정된 모든 제한을 제거합니다.
제한 옵션	행 제한이 AND 연산자로 구현되는지 아니면 OR 연산자로 구현되는지 선택할 수 있습니다.

10.4.4 제한 만들기

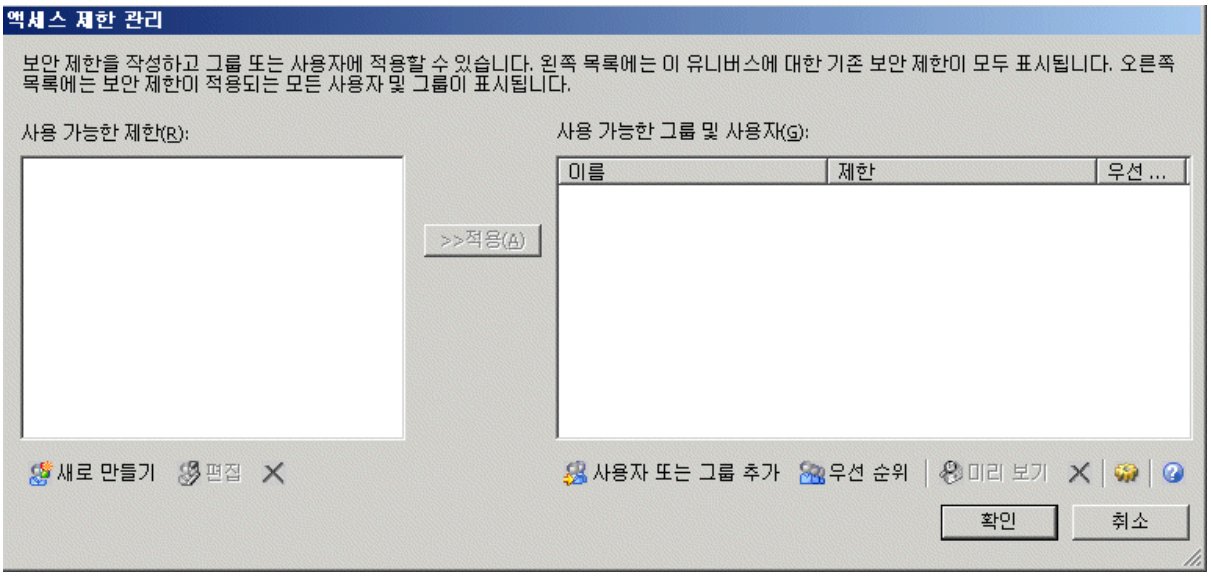
유니버스를 CMS 에 내보낸 다음 언제든지 제한을 작성, 편집 및 삭제할 수 있습니다.

대상 사용자 그룹의 쿼리 요구 사항에 따라 여러 제한을 만들 수 있습니다.

10.4.4.1 제한을 만들려면

1. 도구 > 보안 관리 > 액세스 제한 관리를 선택합니다.

   액세스 제한 관리 상자가 나타납니다.



2. 새로 만들기를 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

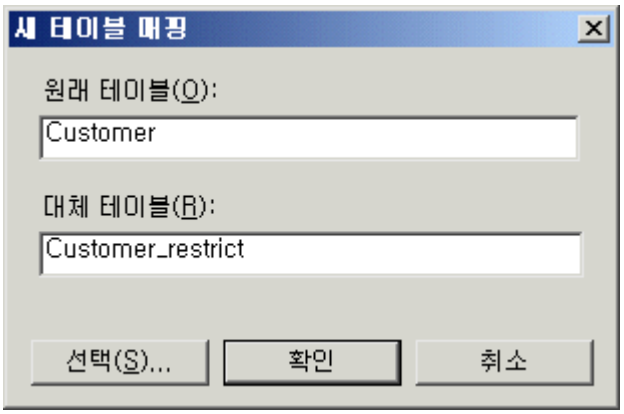
제한 편집 대화 상자가 나타납니다. 유니버스의 연결에 대해 대체 연결을 선택하거나 연결 대화 상자 옆에 있는 단추를 사용하여 연결 속성을 편집할 수 있습니다.

3. 다음 중 필요한 작업을 수행합니다.

표 243:

설정	작업
새 연결	연결 목록 상자에서 연결 이름을 선택합니다.
쿼리 제어	○ 제어 탭을 클릭합니다. ○ 쿼리 옵션을 선택하고 각 옵션의 값을 입력합니다.
SQL 생성 옵션	○ SQL 탭을 클릭합니다. ○ 쿼리, 다중 경로 또는 카티전 곱 옵션에 대한 적절한 확인란을 선택합니다.
개체 액세스 제한	○ 개체 탭을 클릭합니다. ○ 추가를 클릭합니다. 제한된 개체 상자가 나타납니다. ○ 선택을 클릭합니다. 개체 브라우저가 나타납니다. ○ 제한할 개체를 선택합니다. ○ 확인을 클릭하여 개체 브라우저와 제한된 개체 상자를 닫습니다.
행 액세스 제한	○ 행 탭을 클릭합니다. ○ 추가를 클릭합니다. ○ 테이블 상자 옆에 있는 찾아보기 단추를 클릭합니다. ○ 테이블 이름을 클릭하고 확인을 클릭합니다. ○ Where 절 상자 옆에 있는 찾아보기 단추를 클릭합니다. ○ 정의 상자에 WHERE 절을 입력합니다. 또는 SQL 편집기에서 열, 연산자 및 함수를 선택하여 WHERE 절을 작성합니다. 이 편집기 사용에 대한 내용은 조인 SQL 편집기 사용 [페이지 142] 단원을 참조하십시오.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

설정	작업
대체 테이블에 대한 참조	○ 테이블 매핑 탭을 클릭합니다. ○ 추가를 클릭합니다. 새 테이블 매핑 상자가 나타납니다. ○ 원래 테이블 상자에 커서를 놓고 선택을 클릭합니다. 테이블 브라우저가 나타납니다. ○ 테이블을 선택하고 확인을 클릭합니다. ○ 대체 테이블 상자에 커서를 놓고 선택을 클릭합니다. ○ 테이블 탐색기에서 테이블을 선택하고 확인을 클릭합니다. 

4. 확인을 클릭합니다.

   새로운 제한이 목록에 나타납니다.
5. 확인을 클릭합니다.

10.4.5 유니버스 액세스 제한 적용

하나 이상의 사용자 또는 사용자 그룹에 제한을 적용하여 유니버스에 액세스 제한을 설정합니다.

10.4.5.1 유니버스 사용자에게 제한 적용

유니버스의 연결에 대해 대체 연결을 선택할 수 있습니다.

1. 도구 > 보안 관리 > 액세스 제한 관리를 선택합니다.

   액세스 제한 관리 대화 상자가 나타납니다.
2. 사용 가능한 제한 창에서 제한을 클릭합니다.
3. 사용 가능한 사용자 및 그룹 창에서 사용자나 그룹을 클릭합니다.

   또는

   또는 여러 사용자나 그룹을 선택하려는 경우에는 Ctrl 키를 누른 채로 여러 사용자나 그룹을 클릭합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

4. 적용을 클릭합니다.
5. 확인을 클릭합니다.

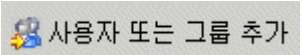
10.4.6 유니버스에 대해 사용 가능한 사용자 목록에 사용자 그룹 추가

BusinessObjects 시스템에 정의된 사용자 그룹에 제한을 적용합니다. 이러한 사용자는 BusinessObjects 관리 콘솔의 그룹 및 사용자 계정 관리 섹션에 정의됩니다. BusinessObjects 시스템에 사용자 및 그룹을 설정하는 방법에 대한 내용은 BusinessObjects Enterprise XI 3.0 관리자 가이드를 참조하십시오.

사용 가능한 그룹 및 사용자 칭에 없는 사용자 그룹에 제한을 적용해야 하는 경우 다음과 같이 목록에 사용자 그룹을 추가할 수 있습니다.

10.4.6.1 "사용 가능한 그룹 및 사용자" 창에 사용자 그룹 추가

1. 액세스 제한 관리 상자(도구 > 보안 관리 > 액세스 제한 관리)에서 사용자 추가 또는 그룹 아이콘을 클릭합니다.

   사용자 및 그룹 선택 대화 상자가 나타납니다. 이 대화 상자에는 BusinessObjects 시스템에 액세스할 수 있도록 BusinessObjects 관리 콘솔에 정의되어 있는 모든 사용자 그룹이 나열됩니다. 사용자 목록이 너무 커서 대상 사용자나 그룹을 쉽게 찾을 수 없는 경우 다음과 같이 목록을 검색할 수 있습니다.
   - 드롭다운 목록에서 이름 또는 설명을 선택합니다.
   - 앞에서 선택한 이름 필드나 설명 필드의 목록에서 텍스트 문자열을 검색하려면 텍스트 상자에 텍스트 문자열을 입력합니다.
   - 검색 아이콘을 클릭하여 검색을 시작합니다.

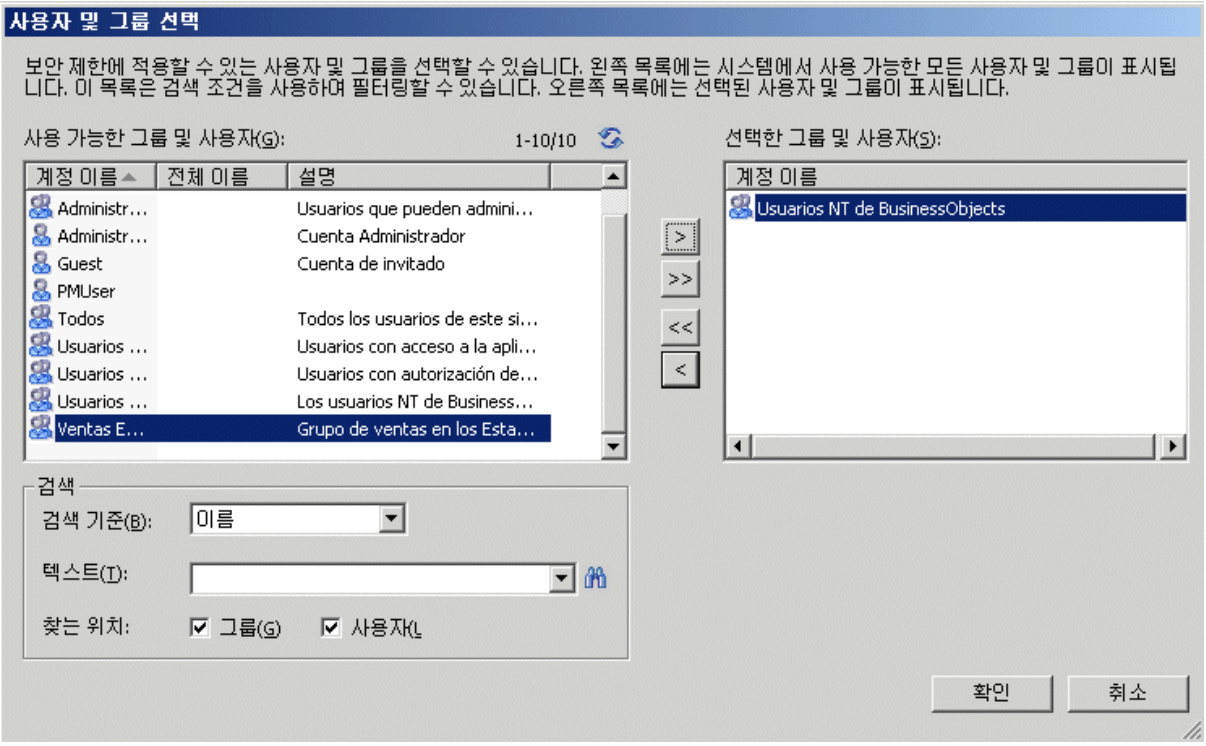
   목록을 필터링하기 위해 그룹 또는 사용자 확인란을 선택하여 목록에 그룹 또는 사용자만 표시할 수도 있습니다.
2. 사용자나 그룹을 클릭합니다.

   또는

   또는 Ctrl 키를 누른 채로 여러 사용자나 그룹을 클릭합니다.
3. 오른쪽 화살표를 클릭합니다.

   대화 상자 오른쪽의 선택한 그룹 및 사용자 목록 창에 해당 사용자나 그룹이 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



4. 확인을 클릭합니다.

   이제 액세스 제한 관리 대화 상자의 사용 가능한 그룹 목록 및 사용 가능한 사용자 목록에 해당 사용자나 그룹이 나타납니다.

10.4.7 제한 그룹 우선 순위 설정

유니버스를 사용하여 여러 그룹에 속한 사용자에게 적용할 제한을 지정할 수 있습니다. 예를 들어, 데이터를 5000 행까지 볼 수 있도록 제한된 판매 그룹과 10000 행까지 볼 수 있도록 제한된 마케팅 그룹에 모두 속한 사용자를 생각해 볼 수 있습니다. 사용자가 보고서를 새로 고치면 최하위 수준 그룹에 관련된 제한이 적용됩니다. 위 예제에서 판매 그룹의 순서가 1 이고 마케팅 그룹의 순서가 2 이면 마케팅 그룹의 제한(10000)이 사용됩니다.

사용자 그룹을 순서에 따라 정렬할 수 있습니다. 나열된 순서에서 가장 낮은 그룹에 대한 제한이 사용됩니다.

> **노트**
>
> 이 규칙은 연결, 테이블 매핑 또는 SQL 제어 같은 단독 제한에만 적용됩니다. 두 그룹에 모두 개체 제한이 설정된 경우에는 모든 제한이 적용됩니다.

10.4.7.1 여러 제한에 대한 사용자 그룹 우선 순위를 설정하려면

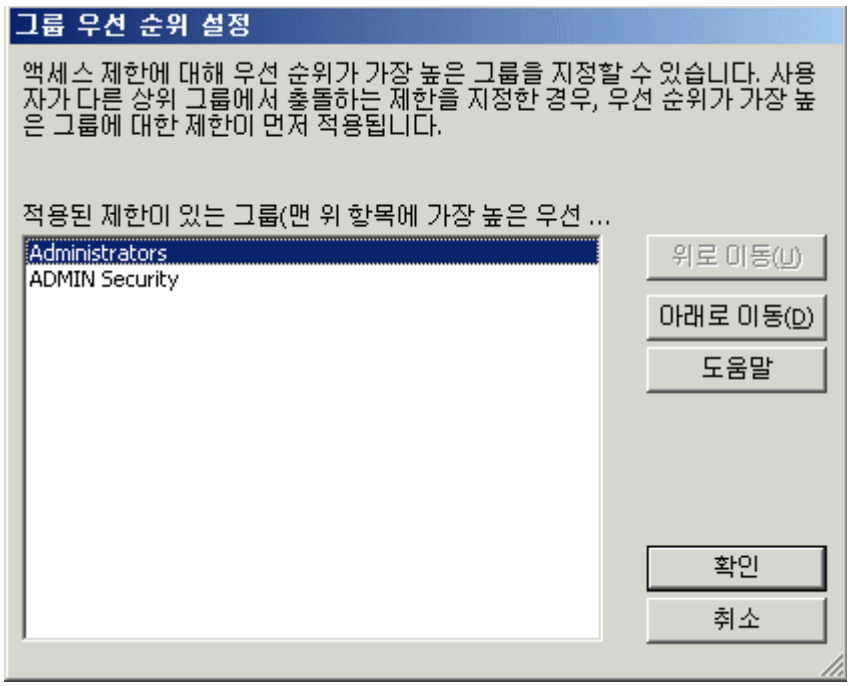
1. 도구 > 보안 관리 > 액세스 제한 관리를 선택합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

액세스 제한 관리 대화 상자가 나타납니다.

2. 사용 가능한 사용자 창 및 사용 가능한 그룹 창에서 사용자나 그룹을 클릭합니다.
3. 우선 순위 아이콘을 클릭합니다.

   그룹 우선 순위 설정 상자가 나타납니다.



4. 사용자 또는 그룹을 선택하고 위로 이동 또는 아래로 이동 단추를 클릭하여 우선 순위를 변경합니다.
5. 확인을 클릭합니다.

10.4.8 사용자 및 그룹 보안 제한 보기

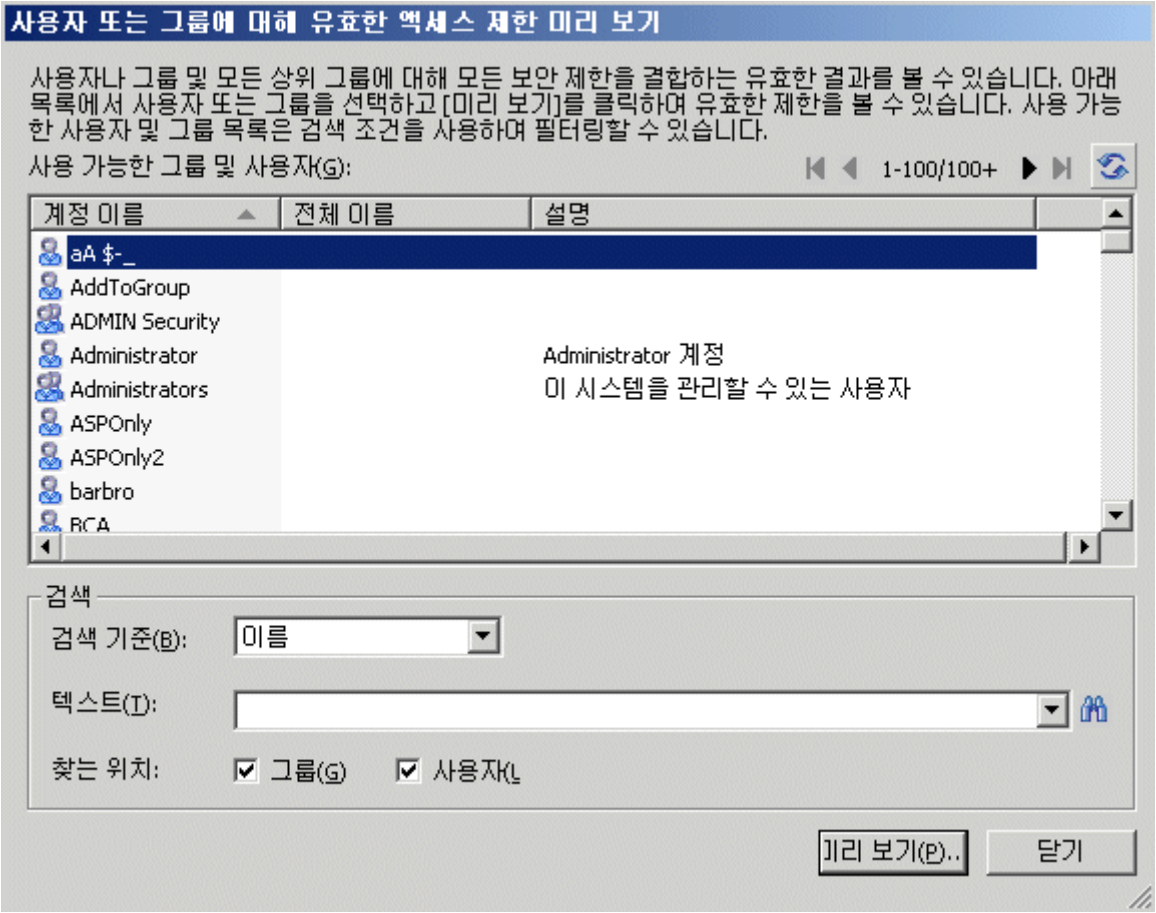
모든 사용자 및 그룹에 적용된 제한을 볼 수 있습니다.

10.4.8.1 모든 유니버스 사용자 및 그룹의 제한을 보려면

1. 도구 > 보안 제한 미리 보기를 선택합니다.

   사용자 및 그룹 미리 보기 대화 상자가 나타납니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.



2. 목록에서 사용자 계정 이름을 클릭합니다.
3. 미리 보기를 클릭합니다.

   해당 사용자 계정에 적용되는 보안 제한이 표시됩니다. 빨간색으로 표시되는 매개 변수와 옵션은 수정된 매개 변수와 옵션이며, 특별히 해당 제한에 적용됩니다.
4. 상자를 닫으려면 확인을 클릭합니다.

10.5 사용자 및 로그인 관리

다른 사용자로 유니버스 디자인 도구에 로그인하고 로그인을 변경할 수도 있습니다. 사용자 계정은 대상 리포지토리에서 유효한 계정이어야 합니다.

또한 독립 실행형 모드로 유니버스 디자인 도구에 로그인할 수 있습니다. 유니버스 디자인 도구를 사용하여 유니버스, 개인 및 공유 연결을 만들 수 있지만 CMS 의 연결 및 유니버스에 액세스할 수는 없습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

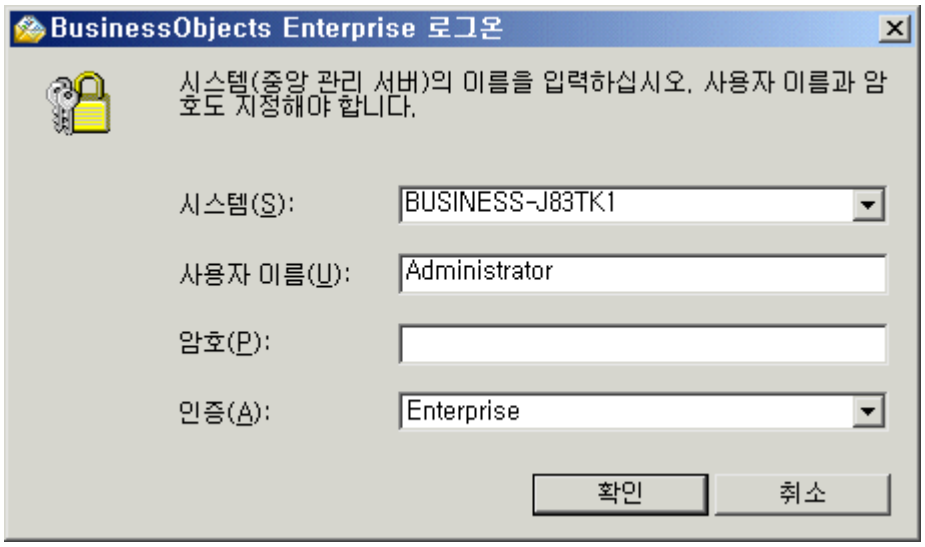
10.5.1 로그인 관리

작업 세션을 끝내지 않고도 다른 사용자로 유니버스 디자인 도구에 로그인할 수 있습니다. 사용자 계정은 대상 리포지토리에서 유효한 계정이어야 합니다. 올바른 사용자 이름과 암호를 알고 있는 경우에만 다른 사용자로 로그인할 수 있습니다.

10.5.1.1 다른 사용자로 로그인하려면

1. 도구 > 다른 이름으로 로그인을 선택합니다.

   열린 유니버스가 있을 경우 유니버스 디자인 도구가 자동으로 닫습니다. 사용자 확인 대화 상자가 나타납니다.



2. 사용자 이름 상자에 유효한 사용자 이름을 입력합니다.
3. 새 사용자 이름과 암호를 입력합니다.
4. 확인을 클릭합니다.

   다른 사용자로 로그인하면 해당 사용자의 모든 권한이 자동으로 부여됩니다. 하지만 사용자 프로필에 설정된 제한 사항에 따라 일부 작업을 수행하지 못할 수도 있습니다.

10.5.2 암호 관리

사용자 계정에 적절한 권한이 부여된 경우 세션 도중 로그인한 계정의 암호를 변경할 수 있습니다. 하지만 사용자 이름을 변경할 수는 없습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

10.5.2.1 암호를 변경하려면

1. 도구 > 암호 변경을 선택합니다.

   암호 변경 대화 상자가 나타납니다.
2. 이전 암호 입력 상자에 기존 암호를 입력합니다.
3. 새 암호 입력 상자에 새 암호를 입력합니다.
4. 새 암호 확인 상자에 새 암호를 다시 입력하여 확인합니다.
5. 확인을 클릭합니다.

   암호가 변경되었습니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

11 샘플 자료 사용

11.1 개요

이 부록에서는 Microsoft Access 로 작성된 Club 데이터베이스의 구조에 대한 자세한 정보를 제공합니다. 이 가이드에 포함된 대부분의 예제와 설명은 이 데이터베이스를 사용한 것입니다.

데이터베이스 파일인 Club.mdb 는 Business Objects 경로의 \Samples\<language>\Databases 하위 폴더에 있습니다. 이 폴더에는 efashion 데모 데이터베이스도 포함되어 있습니다.

11.2 Club 데이터베이스

Club 데이터베이스는 이 가이드에서 설명하는 대부분의 예제에서 사용됩니다.

11.2.1 테이블 구조

Club 데이터베이스는 패키지형 휴양지 상품을 전문으로 하는 가상 기업인 Island Resorts 의 영업 관리자에 의해 사용됩니다. 이 데이터베이스의 정보를 기반으로 영업 관리자는 영업과 마케팅 분석을 수행할 수 있습니다. 데이터베이스는 다음 테이블로 구성됩니다.

- Age_group
- City
- Country
- Customer
- Invoice_Line
- Region
- Region_Sline
- Reservation_Line
- Reservations
- Resort
- Sales
- Sales_Person
- Service
- Service_Line

다음 단원에서는 위의 각 테이블과 해당 열에 대해 설명합니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

11.2.1.1 Age_group 테이블

Age_group 테이블에는 고객의 연령대에 대한 정보가 저장됩니다.

표 244:

열 이름	설명
age_min	연령대의 하한선
age_max	연령대의 상한선
age_range	고객의 연령대

11.2.1.2 City 테이블

City 테이블에는 고객이 거주하는 도시에 대한 정보가 저장됩니다.

표 245:

열 이름	설명
city_id	시스템에서 생성된 도시 번호
city	고객이 거주하는 도시(Albertville, Amsterdam, Augsburg...Versailles, Washington D.C., Yokohama)
region_id	시스템에서 생성된 지역 번호

11.2.1.3 Country 테이블

Country 테이블은 고객이 거주하는 국가와 관련됩니다.

표 246:

열 이름	설명
country_id	시스템에서 생성된 국가 번호
country	고객이 거주하는 국가명(Australia, France, Germany, Holland, Japan, UK, US)

11.2.1.4 Customer 테이블

Customer 테이블에는 성명 및 주소와 같은 고객 식별과 관련된 정보가 포함됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 247:

열 이름	설명
cust_id	시스템에서 생성된 고객 번호
first_name	고객의 이름
last_name	고객의 성
age	고객의 나이
phone_number	고객의 전화 번호
address	고객의 세부 주소
city_id	시스템에서 생성된 도시 번호
sales_id	시스템에서 생성된 영업 담당자 번호(패키지형 휴양지 상품을 판매한 담당자).
sponsor_id	시스템에서 생성된 후원업체 번호(선택 사항)

11.2.1.5 Invoice_Line 테이블

이 테이블에는 송장 정보가 포함되며, 고객에게 청구하는 데 사용됩니다.

표 248:

열 이름	설명
inv_id	시스템에서 생성된 송장 번호
service_id	시스템에서 생성된 서비스 번호
days	휴양지에서의 숙박일 수(3-15). 비용 청구 문제 때문에 숙박일은 최대 15 일까지로 제한됩니다. 15 일을 초과할 경우 시스템은 나머지 일 수를 새로운 숙박일로 간주합니다.
nb_guests	송장이 작성된 대상 고객 인원

11.2.1.6 Region 테이블

Region 테이블에는 고객이 거주하는 지역에 대한 정보가 저장됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 249:

열 이름	설명
region_id	시스템에서 생성된 지역 번호
region	고객이 거주하는 지역(Bavaria, East Coast, East Germany...Wales, West, West Japan)
country_id	시스템에서 생성된 국가 번호

11.2.1.7 Region_Sline 테이블

이 테이블은 유니버스에서 판매 수익 집계를 계산하는 데 사용됩니다. 집계 인식에 대한 내용은 이 가이드의 5 장을 참조하십시오.

표 250:

열 이름	설명
sl_id	시스템에서 생성된 서비스 분야 번호(서비스 분야 정보는 Service_Line 테이블에서 제공)
region_id	시스템에서 생성된 지역 번호
sales_revenue	지역별 총 판매 수익

11.2.1.8 Reservation_Line 테이블

Reservation_Line 테이블에는 고객 예약과 관련된 정보가 저장됩니다.

표 251:

열 이름	설명
res_id	시스템에서 생성된 예약 번호
service_id	시스템에서 생성된 서비스 번호
res_days	주간 예약일 수(1 - 7)
future_guests	향후 고객 인원 수(1 - 5)

11.2.1.9 Reservation 테이블

Reservation 테이블에는 고객 예약 날짜에 대한 정보가 포함됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 252:

열 이름	설명
res_id	시스템에서 생성된 예약 번호
cust_id	시스템에서 생성된 고객 번호
res_date	고객이 예약한 날짜

11.2.1.10 Resort 테이블

Resort 테이블에는 각 휴양지에 대한 정보가 포함됩니다.

표 253:

열 이름	설명
resort_id	시스템에서 생성된 휴양지 번호
resort	휴양지 이름: Australian Reef, Bahamas Beach, French Riviera, Hawaiian Club, Royal Caribbean
country_id	시스템에서 생성된 국가 번호

11.2.1.11 Sales 테이블

Sales 테이블에는 영업 정보가 포함됩니다.

표 254:

열 이름	설명
inv_id	시스템에서 생성된 송장 번호
cust_id	시스템에서 생성된 고객 번호
invoice_date	송장 날짜

11.2.1.12 Sales_Person 테이블

Sales_Person 테이블에는 Island Resorts 영업 담당자에 대한 정보가 저장됩니다.

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

표 255:

열 이름	설명
sales_id	시스템에서 생성된 영업 담당자 번호
sales_person	영업 담당자의 이름(Andersen, Barrot, Bauman... Moore, Nagata, Schmidt)

11.2.1.13 Service 테이블

Service 테이블에는 해당 휴양지에서 사용 가능한 서비스의 가격과 종류에 대한 정보가 포함됩니다.

표 256:

열 이름	설명
service_id	시스템에서 생성된 서비스 번호
service	휴양지에서 사용 가능한 서비스(아래 쿼리 결과 참조)
sl_id	시스템에서 생성된 서비스 분야 번호(서비스 분야 정보는 다음 테이블에서 제공)
price	서비스의 가격

11.2.1.14 Service_Line 테이블

Service_Line 테이블에는 휴양지의 서비스 분야에 대한 정보가 저장됩니다. 서비스 분야는 단순히 서비스의 범주를 의미합니다.

표 257:

열 이름	설명
sl_id	시스템에서 생성된 서비스 분야 번호
service_line	서비스 분야 포함 내용: 숙박, 음식과 음료, 레크리에이션
resort_id	시스템에서 생성된 휴양지 번호(1 ~ 5)

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

중요 법적 면책 사항 및 법률 정보

코딩 샘플

이 문서에 포함된 어떠한 소프트웨어 코딩 및/또는 코드 라인/문자열 ("코드")도 예시 목적으로만 사용되며 운영 시스템 환경에의 사용을 의도하지 않습니다. 코드는 특정 코딩의 구문 또는 구문 지정 규칙을 좀 더 잘 설명하고 표시하기 위해서만 사용됩니다. SAP 는 이 문서에 제공된 코드의 정확성과 완전성을 보증하지 않으며, SAP 의 의도나 중과실로 인해 발생한 손해가 아닌 한, 코드의 사용으로 인해 발생한 오류나 손해 부분에 대한 책임을 지지 않습니다.

접근성

SAP 문서에 포함된 정보는 해당 게시일 현재 SAP 의 접근성 기준에 대한 관점을 나타내는 것입니다. 소프트웨어 제품의 접근성을 보장하기 위한 법적 가이드라인이 될 의도는 전혀 없습니다. 특히 SAP 는 이 문서에 대해 어떠한 책임도 없습니다. 단, SAP 측에서 의도적인 부적합 행위나 중과실을 저지른 경우에는 이 면책 주장이 적용되지 않습니다. 또한 이 문서는 SAP 에 어떠한 직/간접적인 구속적 의무도 발생시키지 않습니다.

성 중립적 언어 사용

SAP 문서는 가능한 범위에서 성 중립성을 유지합니다. 문맥에 따라 독자의 경우 직접 "사용자"로 언급되고, 성 중립적 명사(예: "영업 사원" 또는 "근무일")가 사용됩니다. 양쪽 성별을 모두 나타낼 때 3 인칭 단수를 배제할 수 없거나 성 중립적 명사가 없는 경우, SAP 는 명사 및 대명사의 남성형을 사용할 권리가 있습니다. 이는 문서의 이해를 돕기 위한 것입니다.

인터넷 하이퍼링크

SAP 문서에는 인터넷으로 연결된 하이퍼링크가 포함될 수 있습니다. 이러한 하이퍼링크는 관련 정보를 찾기 위한 힌트로 사용됩니다. SAP 는 이와 관련된 정보의 가용성 및 정확성, 또는 이 정보가 특정 목적으로 사용될 가능성에 대해 보증하지 않습니다. SAP 는 SAP 의 중과실 또는 고의적 불법 행위에 의해 손해가 발생한 경우 외에, 관련 정보의 사용으로 발생한 어떠한 손해에 대해서도 책임을 지지 않습니다. 투명성을 위해 모든 링크가 범주별로 분류되어 있으니 참고해 주시기 바랍니다(참조: http://help.sap.com/disclaimer).

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.

www.sap.com/contactsap

© 2015 SAP SE 및 SAP 계열사. 모든 권한 보유.
본 발행물의 어떠한 부분도 SAP SE 또는 SAP 계열사의 명시적 허가 없이는 어떠한 형태나 목적으로도 복제 또는 배포할 수 없습니다. 본 문서의 정보는 사전 예고 없이 변경될 수 있습니다.
SAP SE 및 그 유통업자가 판매하는 일부 소프트웨어 제품에는 다른 소프트웨어 공급업체가 소유한 소프트웨어 구성 요소가 포함되어 있습니다. 국가별 제품 명세는 다를 수 있습니다.
이 문서는 SAP SE 또는 SAP 계열사에 의해 정보 전달 목적으로만 제공되며 어떠한 종류의 진술이나 보증도 포함되지 않습니다. SAP 또는 SAP 계열사는 이 문서의 오류나 누락 부분에 대해 책임을 지지 않습니다. SAP 또는 SAP 계열사 제품 및 서비스에 대한 유일한 보증은 해당 제품 및 서비스와 함께 제공되는 보증서에 명시된 내용으로 제한됩니다. 본 문서의 어떤 내용도 추가 보증의 근거로 해석할 수 없습니다.
SAP 및 본 문서에서 언급된 기타 SAP 제품, 서비스와 해당 로고는 독일 및 기타 국가에서 사용되는 SAP SE(또는 SAP 계열사)의 상표 또는 등록 상표입니다. 기타 언급된 모든 제품 및 서비스 이름은 각각의 해당 기업 상표입니다.
추가 상표 정보 및 공지는 http://www.sap.com/corporate-en/legal/copyright/index.epx 에서 확인하십시오.