Data Modeling and Relational Database Design(한글판)

학생용 – 볼륨 2

mternal & OAI Use Only

20000KR13 Edition 1.3 2002년 8월 D38040



만든 이

Patrice Daux Jeff Gallus Jan Speelpenning

기술 제공자 및 검토자

Kate Heap Simmie Kastner Joni Lounsberry Satyajit Ranganathan Sunshine Salmon Stijn Vanbrabant Gabriella Varga

발행인

Christine Markusic

Copyright © Oracle Corporation, 2002. All rights reserved.

이 문서는 Oracle International Corp.의 독점적 정보를 포함하고 있습니다. 이 정보는 사용 제한 및 기밀 유지 규정을 포함하는 사용권 계약에 따라 제공되며 저작권법에 의 해 보호됩니다. 이 소프트웨어를 리버스 엔지니어링하는 것은 금지되어 있습니다. 이 문서를 미국 국방성 내의 정부 기관에 제공할 때는 제한된 권리 규정이 적용되며 다음 범례를 적용할 수 있습니다.

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c) (1) (ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of Oracle International Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with 'Restricted Rights,' as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Education Products, Oracle Corporation, 500 Oracle Parkway, Box, Redwood Shores, CA 94065. Oracle International Corporation does not warrant that this document is error-free.

Oracle is a registered trademark and Oracle, SQL*Plus, SQL*Net, Oracle Developer, Oracle8*i*, Oracle9*i*, Oracle9*i* Designer and PL/SQL are trademarks or registered trademarks of Oracle Corporation.

인용된 모든 다른 회사명 또는 제품명은 명시의 목적으로만 사용되었으며 각 소유 회사들의 상표일 수 있습니다.



목차

1 엔티티, 속성 및 관계 소개

개요 1-2

왜 개념적 모델을 생성합니까? 1-3

이상과 실제... 1-5

엔티티 관계 모델링 1-7

엔티티 관계 모델링의 목표 1-8

클러스터 유형 1-9

엔티티 1-10

엔티티 및 인스턴스 1-11

엔티티 및 세트 1-12

속성 1-13

속성 예제 1-14

관계 1-15

관계 예제 1-16

Employee는 Job을 가짐 1-17

도표 상의 엔티티 표시 1-18

도표의 속성 1-20

도표 상의 관계 1-21

도표를 통한 용이한 의사 전달 1-22

관계 라인의 특성 1-23

두 가지 관점 1-24

한쪽 방향 1-25

, 1-40 연습 1-41 연습: 인스턴스인가 아니면 엔티티인가? 1-42 연습: 투숙객 1-43 연습: 읽기 1-44 연습: 읽기 및 주석 달기 1 45 연습: 호텔 1-46

연습: 호텔 1-46

2 엔티티 및 속성 세부 사항

개요 2-2

정보에 데이터 비교 2-3

데이터 2-4

엔티티 2-6

1: 필수 m 3-9

일부 배경 정보 2-9

일부 요구되는 기능 2-13

엔티티 정의의 변천 과정 2-15

업무 요구 사항 2-17

속성... 2-19

명사, 엔티티, 속성 2-20

EM 엔티티 및 속성 2-22

속성 및 엔티티 2-23

중복성 2-24

하위 유형... 2-25

하위 유형 예제 2-26

하위 유형: 규칙 2-27

하위 유형: 세 레벨 2-28

하위 유형에 대한 자세한 내용 2-29

요약 2-30

연습 2-31

요약 2-34

판매점 목록 2-36

하위 유형 2-38

Oracle Internal & OAI Use Only 연습: 주소(1/2) 2-40

연습: 주소(2/2) 2-41

3 관계 세부 설명

개요 3-2

관계 설정 3-3

관계 이름 3-5

관계 명명 3-6

선택 가능성 3-7

선택 가능성 3-8

필수

정도 3-10

비전이성 3-12

관계 유형 1:m 3-13

관계 유형 m:1 3-15

관계 유형 m:m 3-16

관계 유형 1:1 3-18

1:1 관계 롤 3-19

1:1 관계 프로세스 3-20

중복 관계 3-21

관계 및 속성 3-22

관계에 속성 비교 3-23

속성 또는 엔티티 3-24

관계에 속성 비교 3-25

관계에 속성 비교 3-26

m:m 관계로 특정 사항을 숨길 수 있음 3-27

Quantity는 ...의 속성 3-28

관계의 속성 3-29

새로운 ORDER 엔티티 3-30

한 주문에 대한 다수의 제품 3-31

또 다른 새 엔티티: ORDER ITEM 3-32

테이블 3-33

m:m 관계 분석 3-34

m:1 관계 분석 3-37

정규화 규칙 3-39

. J . 기(2) 3-47 1스트 찾기(3) 3-48 컨텍스트 찾기(4) 3-49 연습: 교차 엔티티 명명 3-50 연습: 영수증 3-51 연습: Moonlight P&O 3-52 연습: 가격 목록 3-54 변습: 전자 때

연습: 휴일 3-56

연습: ER 모델 정규화 3-57

4 제약 조건

개요 4-2

렘브란트 4-3

신원 증명과 표시 4-5

고유 식별자 예제 4-6

고유 식별자 4-7

고유 식별자 4-8

복수 관계 UID 4-10

올바르게 정의된 고유 식별자 4-12

잘못된 고유 식별자 4-13

정보를 포함하는 코드 4-14

호 4-15

배타적 호 4-16

가능한 호 구성 4-17

몇 가지 잘못된 호 구성 4-18

호 또는 하위 유형 4-20

호 및 하위 유형 4-21

하위 유형은 호 내의 관계를 숨김 4-22

값 집합 4-23

기타 제약 조건: 범위 검사 4-24 기타 제약 조건: 상태 값 변환 4-25

조건부 관계 4-26

경계 4-28

요약 4-29

- +-37 Julight UID 4-38 - 급: 테이블 14-39 연습: 테이블 2 4-41 연습: 제약 조건 1 4-42 연습: 제약 조건 2 4-43 연습: 제약 조건 3 4-44

5 모델링 변화

개요 5-2

변경 및 시간 5-3

DAY 엔티티 5-4

모델링 변경 5-6

국가에도 수명 주기가 있습니다. 5-8

제품 및 가격 5-10

지불할 가격은 얼마입니까? 5-11

가격 목록 검색 5-12

Priced Product에 대한 주문 5-13

협상 가격 5-14

현재 가격 5-16

저널링 5-17

요약 5-19

연습 5-20

연습: 근무 교대 5-21 연습: 딸기 와퍼 5-22 연습: 번들(1) 5-24 연습: 번들(2) 5-25

연습: 제품 구조 5-26

6 고급 모델링 항목

개요 6-2

패턴 6-3

패턴: 마스터-디테일 6-5

패턴: 바구니 6-6

패턴: 분류 6-7

패턴: 계층 구조 6-8

패턴: 체인 6-10

패턴: 네트워크 6-11

BOM 6-13

BOM-예제 6-14

대칭 관계 6-15

패턴: 롤 6-16

롤 6-17

Fan Trap 6-18

분석된 Fan Trap 6-19

ternal & OAI Use Only 패턴: 데이터 웨어하우스 6-20

그리기 규칙 6-21

적절한 규칙 사용 6-22

모델 가독성 6-23

일반 모델링 6-24

일반적인 모델 6-26

일반적인 모델 6-27

보다 일반적인 모델 6-28

보다 일반적인 것 이상의 모델 6-29

가장 일반적인 모델 6-30

두 분야의 최고 6-31

요약 6-32

연습 6-33

연습 6-34

연습: 데이터 웨어하우스 6-35

연습: Argo 및 Erat 6-38

연습: 동의어 6-39

7 엔티티 모델 매핑

개요 7-2

왜 데이터 디자인 모델을 생성합니까? 7-3

테이블 표시 7-4

변환 프로세스 7-6

용어 매핑 7-7

일반 이름 지정 항목 7-8

Oracle의 이름 지정 제한 7-11

엔티티에 대한 기본 매핑 7-12

속성에 대한 기본 매핑 7-13

기본 매핑 7-14

관계에 대한 규칙 7-16

요약 7-37

연습 7-38

연습: 기본 엔티티, 속성 및 관계 매핑 7-39

연습: 상위 유형 매핑 7-40 Moonlight ER 모델 일부 7-41

연습: 품질 검사 하위 유형 구현 7-42

연습: 품질 검사 호 구현 7-43 연습: 기본 키와 열 매핑 7-44

8 비정규화된 데이터

개요 8-2

비정규화 개요 8-3

비정규화 기법 8-4

파생 가능 값 저장 8-5

파생 가능 값을 저장하는 전자 메일 예제 8-6

테이블 사전 조인 8-7

테이블 사전 조인의 전자 메일 예제 8-8

하드 코딩된 값 8-9

하드 코딩된 값의 전자 메일 예제 8-10

마스터로 디테일 유지 8-11

마스터로 디테일을 보관하는 전자 메일 예제 8-12

마스터에 최근 디테일 반복 8-13

마스터에 단일 디테일을 반복하는 전자 메일 예제 8-14

단락 키 8-15

단락 키의 전자 메일 예제 8-16

종료일 열 8-17

종료 날짜 열 예제 8-18

현재 지시자 열 8-19

현재 지시자 열 예제 8-20

계층 레벨 지시자 예제 8-21

계층 레벨 지시자 예제 8-22

비정규화 요약 8-23

연습 8-24

zoi 27 Remai & OAI Use Only 연습: 비정규화 명명(1/3) 8-25 연습: 비정규화 명명(2/3) 8-26 연습: 비정규화 명명(3/3) 8-27

연습: 트리거(1/6) 8-28

연습: 트리거(2/6) 8-29

연습: 트리거(3/6) 8-30 연습: 트리거(4/6) 8-31

연습: 트리거(5/6) 8-32

연습: 트리거(6/6) 8-33

연습: 가격 목록 비정규화 8-34 연습: 전역 이름 지정 8-35

9 데이터베이스 설계 고려 사항

개요 9-2

왜 데이터 설계를 수정해야 합니까? 9-3

Oracle 데이터 유형 9-4

제안된 열 순서 9-6

기본 키 9-7

인공 키 9-11

시퀀스 9-13

외래 키 기능 9-14

인덱스 9-16

인덱스 선택 9-17

어떤 열을 인덱스화해야 합니까? 9-19

인덱스는 언제 사용할 수 있습니까? 9-21

테이블과 인덱스 분할 9-22

뷰 9-23

뷰를 사용하는 이유 9-24

기존 방식의 설계 9-25

일반 호 구현 9-26

분산 데이터베이스 9-27

분산 데이터베이스의 이점 9-28

데이터베이스 구조 9-29

요약 9-31

연습 9-32

데이터 유형(1) 9-34

Oracle Internal & OAI Use Only 데이터 유형(2) 9-35

부록 A

부록 B



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



개요

- 패턴
- 그리기 규칙
- 일반 모델링

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

소개

이 단원에서는 데이터 모델에서 발견할 수 있는 패턴에 대해 대략적으로 설명합니다. 이 단원에서는 몇 개의 일반적인 모델도 소개합니다. 사용자는 이러한 모델을 사용하여 예측 가능하지만 아직은 알 수 없는 미래의 변화를 자신의 모델이 견뎌낼 수 있도록 만들 수 있습니다.

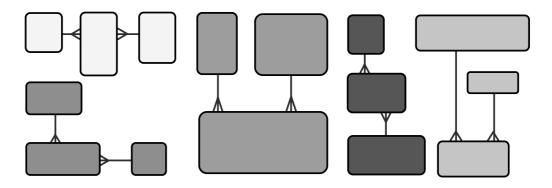
목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- 개념적 데이터 모델에서의 공통 패턴 인식
- 이러한 패턴의 일반적인 동작(예: 공통 제약 조건) 파악
- 특정 그리기 규칙 사용
- 개념적 데이터 모델의 선택 섹션에 대해 보다 일반적인 모델 생성

패턴

- 유사한 구조
- 유사한 규칙 및 제약 조건



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

패턴

유사한 구조

많은 모델들은 컨텍스트가 완전히 다를 수 있어도 유사한 구조를 가지는 부분을 포함합니다. 예를 들어, 표제어, 항목, 의미, 동의어와 같은 개념을 다루는 사전의 컨텍스트에서 개념적 데이터 모델의 구조는 놀랍게도 철로, 역, 연결로 구성된 철도의 구조와 거의 동일하고, 야구 또는 축구 경기의 구조와도 거의 동일합니다.

더 쉽게 눈에 띄는 것은 예를 들면, ORDER ITEM이 있는 ORDER HEADER와 QUOTATION ITEM이 있는 QUOTATION HEADER 간의 유사성 또는 MARRIAGE와 JOB ASSIGNMENT 간의 유사성입니다.

패턴

왜 유사성을 찾는가?

유사성을 찾는 것이 중요한 주된 이유는 시간이 절약되기 때문입니다. 특정 컨텍스트에서 문제를 해결했고 그 해결책을 다른 문제에 적용할 수 있다면 분명히 시간이 절약됩니다. 더욱이상황을 잘 파악하고 있다는 자신감을 느끼게 될 것입니다. 이것은 올바른 질문을 하는 데 도움을 줄 것이며, 대단히 복잡하고 불쾌한 일들을 식별하는 데 도움을 주고 동일한 실수를 두번 저지르지 않도록 방지해 줄 것입니다.

결혼과 작업 할당 간에는 유사성이 있습니까? 물론 결혼이라는 컨텍스트에서의 업무 규칙은 작업 할당의 업무 규칙과 비교하여 다르게 결정되기 때문에 다릅니다. 그러나 유사성을 인식하면, 첫번째 컨텍스트의 업무 규칙이 두번째에 적용되는지 쉽게 확인할 수 있습니다. 예를 들어, 다음과 같은 질문을 던져보십시오.

- 둘이상의 작업에 대한 할당이 가능합니까?
- 두 개의 할당을 동시에 가질 수 있습니까? 비공식적으로는 어떻습니까?
- 할당은 어떻게 시작되며 어떻게 끝납니까?

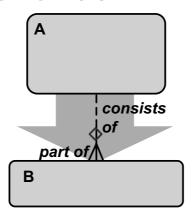
다음 단락에서는 모델 생성 시 직면하게 되는 일련의 패턴에 대해 설명하며, 이러한 모든 패턴에 대해 일반적으로 적용되는 특징과 규칙이 제시됩니다.



패턴: 마스터-디테일

특성: consists of
 B의 인스턴스는 A의 컨텍스트에만 존재합니다.

• 은유법: 마스터 – 디테일



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

마스터 디테일

마스터-디테일 구조는 1:m 관계가 매우 일반적인 것처럼 매우 일반적입니다. 일반적으로 1에서 many로 방향이 진행되는 1:m 관계와 그와 반대로 방향이 진행되는 관계를 구분하십시오. (아래 참조) 마스터-디테일은 마스터 A가 B로 분할된다는 점이 특징입니다. B는 단독으로 존재하지 않으며, 항상 A의 컨텍스트에 존재합니다.

이러한 관계가 전이될 가능성은 매우 희박합니다. B의 인스턴스가 A의 잘못된 인스턴스와 연결되어 있을 경우, B의 인스턴스는 삭제된 다음 올바른 A의 컨텍스트에 재생성될 가능성이 훨씬 더 많습니다.

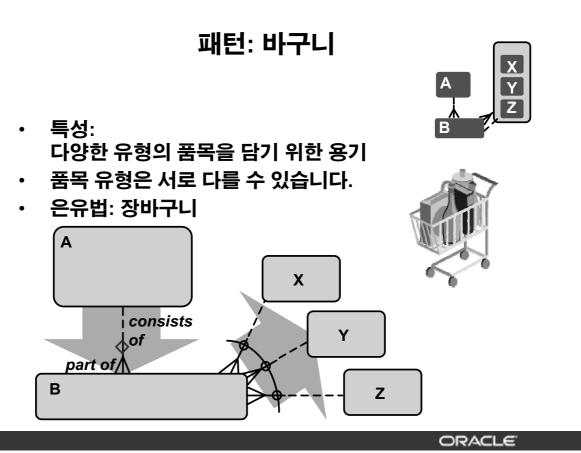
일반적인 마스터-디테일 관계 이름은 다음과 같습니다.

Consists of - Divided into - Made of - (Exists) With

마스터 A에 B가 없을 때, 예를 들어, 관계가 한 쪽에서만 필수일 때, 대개 마스터 A는 중요하지 않습니다. 앞에서 보았듯이, 이 필수 관계 끝은 일반적으로 우회될 수 있습니다.

구현

이 마스터-디테일 패턴에서 비롯된 테이블은 클러스터된 것으로 간주되어야 합니다.



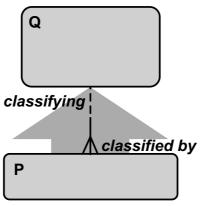
Copyright © Oracle Corporation, 2002. All rights reserved.

바구니

바구니 구조는 마스터-디테일 패턴의 특수한 경우입니다. 바구니는 한 개 이상의 물건을 담을수 있지만, 이러한 물건들(품목이라고도 함)의 유형은 서로 다를 수 있습니다. 단일 품목은 항상 한 개의 유형만으로 되어 있습니다. 이것은 호에 대한 이유입니다. 호는 품목이 한 개 유형으로만 되어 있어야 한다는 것을 나타냅니다.

패턴: 분류

- 특성: classified by, grouped by
 Q는 독립적으로 존재하지만 관련되어 있을 수도 있습니다.
- · 은유법: EMPLOYEE-DEPARTMENT



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

분류

이것도 1:m 관계이지만 이제는 기본 방향이 P에서 Q로 향합니다.

이것은 일반적으로 Q가 P와는 독립적으로 존재할 수 있는 경우입니다. Q는 P 그룹화에 관련된 P에 대한 클래스 역할을 합니다.

일반적으로 개념적 데이터 모델의 엔티티에는 이러한 클래스가 여러 개 있습니다.

일반적인 분류 유형 관계 이름은 다음과 같습니다.

- Classified by
- Grouped by
- Assigned to

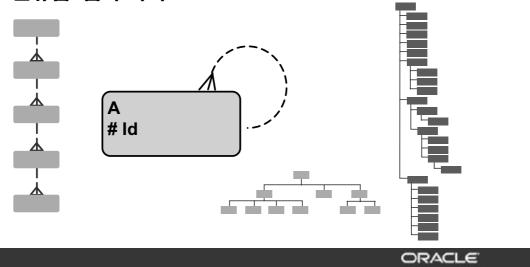
Oracle

• (Exists) In

분류는 시간이 경과하면서 변경될 수 있기 때문에 관계는 일반적으로 전이가 가능합니다.

패턴: 계층 구조

- · 특성: manager of / subordinate of
- 계층 구조 특성을 보호하기 위한 추가적인 제약 조건
- 은유법: 엄마-아이



Copyright © Oracle Corporation, 2002. All rights reserved.

계층 구조

대부분의 계층 구조에는 최대 레벨 수에 대해 알려진 제한이 있습니다. 그럴 경우 그리고 제한이 5보다 낮은 숫자일 경우, 일반적으로 최상의 모델은 그림의 왼쪽에 표시된 레벨당 하나의 엔티티가 있는 모델입니다.

다음과 같은 경우에 순환 관계를 가진 구조를 모델링합니다.

- 구조에 알려진 레벨 제한이 없을 경우
- 구조에 레벨 한계가 있지만 그 한계가 높을 경우(예: 6 이상)
- 구조의 인스턴스가 쉽게 위치를 변경할 수 있고 그로 인해 레벨이 변경되는 경우
- 사용자가 제약 조건 유지를 선호할 경우

논쟁의 여지가 있거나 잘못된 계층 구조

대개 구조는 계층적이어야 하지만 사용자는 이를 확신할 수 없습니다. 때때로 구조는 계층적인 것처럼 보이지만 실제로는 그렇지 않습니다. 예를 들어, 회사들 간에 is owner of 관계를 가질 수 있습니다. C1 회사가 C2 회사를 소유하고, C2 회사는 C3 회사를 소유한다고 가정하면, C3 회사는 C1 회사의 주식을 소유할 수 있습니까? 법률이 그러한 이상한 구조를 금지한다 해도, 이를 확신합니까? 많은 사람들은 계층적 관계의 은유로 부모/자식 관계를 생각합니다. 보통 아이에게는 부모가 모두 있고 양부모도 있을 수 있기 때문에 이것은 명확히 잘못된 것입니다.

계층 구조

또한 파일과 특정한 유형의 파일인 폴더를 가진 FILE SYSTEM의 계층 구조도 Windows의 단축키(또는 UNIX의 링크)의 개념을 생각할 때 논쟁의 여지가 있는 계층입니다. 단축키와 링크는 특수한 롤을 가진 파일이지만, 이러한 단축키는 개념적으로 계층을 네트워크로 변환합니다.

순환 관계 및 선택 가능성

계층은 어디에서든 시작하거나 끝나야 하므로 실제 계층을 기술하는 순환 관계는 일반적으로 양쪽 끝에서 선택적입니다.

계층에 적용되는 제약 조건

위의 가운데 그림에서 알 수 있듯이, 순환 모델은 A의 유효 인스턴스를 참조하는 데 A의 인스턴스만 필요로 합니다. 이 모델에 따르면, A1을 참조하는 A1은 허용됩니다. A3를 참조하는 A2와 A2를 참조하는 A3도 허용됩니다. 이들은 하나의 실계층에서 나올 수 있는 명백한 변형일 뿐입니다.

계층 구조에 적용되는 제약 조건은 계층 보호를 다루며 테이블이 위와 같은 종류의 데이터를 포함하지 못하도록 해야 합니다.

구혀

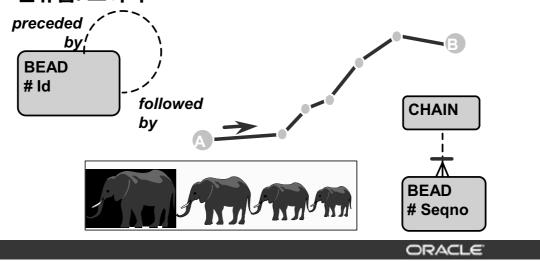
첫번째 제약 조건 A1은 A1을 참조하지 않을 수 있고, 사용자는 Oracle 검사 제약 조건을 사용하여 이를 쉽게 확인할 수 있습니다. 다른 제약 조건은 약간의 프로그래밍이 필요하고 데이터베이스 트리거로 이어집니다.

예를 들어, 레벨 수가 제공된 값을 초과하지 않을 경우, 추가 업무 규칙을 검사해야 할 수도 있습니다.



패턴: 체인

- 특성: preceded by / followed by
- 순서가 중요합니다.
- 은유법: 코끼리



Copyright © Oracle Corporation, 2002. All rights reserved.

체인

체인(구슬로 엮은)은 특수한 종류의 계층으로 간주할 수 있습니다. 체인은 엔티티의 순환 관계입니다. 체인의 특징은 체인에 있는 객체의 앞뒤에 최대 한 개의 객체가 온다는 점이므로 체인의 관계는 1:1 관계입니다.

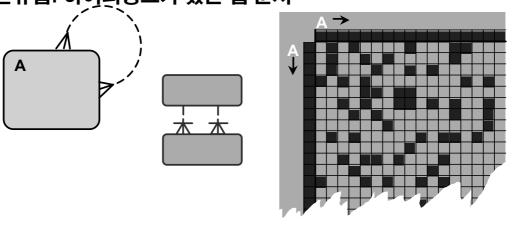
체인은 순서가 중요한 구조입니다. 한 장에서의 페이지 순서, 한 문서에서의 장 순서, 한 프로 시저에서 중요한 경로의 순서, A에서 B까지의 선호하는 도로의 순서 등을 예로 들 수 있습니다.

체인은 또한 마스터-디테일로 모델링될 수도 있습니다. 순환 모델은 체인에 쉽게 삽입할 수 있습니다. CHAIN 및 BEAD 엔티티를 가진 오른쪽 모델은 삽입된 구슬 뒤의 모든 구슬의 순서 번호를 변경해야 할 수 있습니다.

패턴: 네트워크

특성: pairs
 모든 A는 모든 A에 연결될 수 있습니다.
 (때때로 그 외 다른 A에 연결될 수 있음)

• 은유법: 하이퍼링크가 있는 웹 문서



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

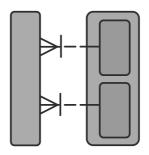
네트워크

네트워크 구조는 일반적으로 동일한 유형으로 된 사물의 쌍을 기술합니다. 예를 들면, 결혼, 철로(발차역과 종착역), 동의어(의미가 동일한 두 개의 낱말), 다른 웹 문서에 연결되는 하이 퍼링크를 가진 웹 문서 등입니다.

특성

일반적 특징:

- m:m 관계는 결혼 날짜 또는 철로 길이와 같이 쌍에 대한 특정 정보를 보유하도록 분석 되어야 합니다.
- 교차 엔티티의 두 관계는 고유 식별자를 구성합니다.
- 예를 들면, 시간 관련 제약 조건은 "순서에 의한 일부일처주의"를 다루는 일종의 규칙을 보호해야 하는 네트워크에 적용됩니다.
- 두 개의 관계는 상이한 하위 유형의 엔티티를 참조합니다.



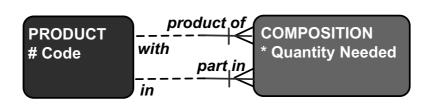
Copyright © Oracle Corporation, 2002. All rights reserved.

네트워크

Oracle Internal & OAI Use Only 계층은 특정한 일련의 업무 규칙이 적용되는 네트워크입니다.

Data Modeling and Relational Database Design 6-12

BOM



COMPOSITIONS

Code	Name	Prod_code	Part_code	Quantity
914.53	AAAAAAA	854.01	604.18	1
914.54	AAA	854.01	604.19	1
914.55	BBBBBBBB	854.01	914.54	2
914.56	В	914.54	914.55	1
	2 2 2	914.54	914.56	1
	DDDDD	03 40	-1100	

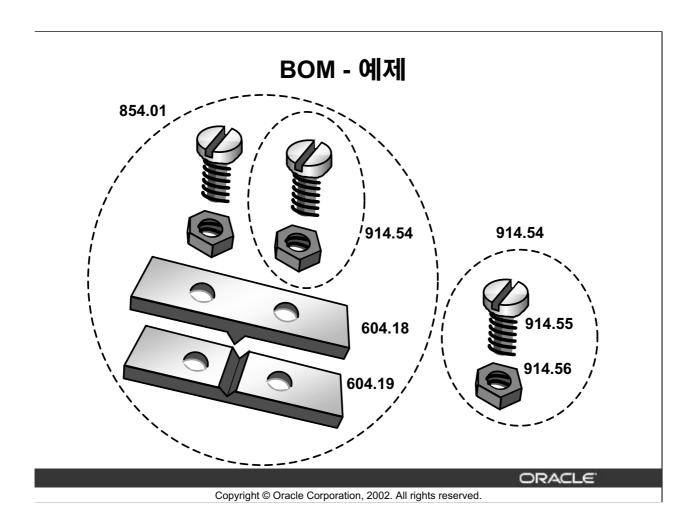
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

네트워크

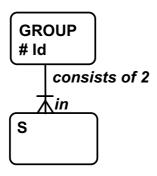
BOM

네트워크 구조에 대한 특별한 예는 BOM(Bill of Material)입니다. BOM은 사물이 다른 사물로 구성되는 방식과 이러한 다른 사물이 얼마나 필요한지(여기서 이것은 PRODUCT의 인스턴스 임)를 기술합니다. COMPOSITION 엔티티는 Quantity Needed 속성을 가진 교차 엔티티입니다.





대칭 관계



Group_id	S
1	S1
1	S2
2	S3
2	S4
3	S5
3	S6

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

대칭 관계

대칭 순환 관계는 생각보다 훨씬 복잡한 매우 특별한 종류의 문제를 야기합니다. 대부분의 컨텍스트에서 한 쌍 (A1, A2)의 레코드는 (A2, A1)으로 언급될 때와 다른 의미를 가집니다. 예를들어, 모델이 PERSON 엔티티에 대한 것이고 관계가 mother of /daughter of일 경우, 사람 쌍 (P1, P2)의 존재는 쌍 (P2, P1)의 가능성을 배제한 것임을 의미합니다. PERSON의 순환 관계는 family of / family of 입니다. 여기에서 (P1, P2)가 참이면 (P2, P1)도 참입니다. 이를 대칭 관계라고 합니다. 다음과 같은 다른 대칭 순환 관계도 있습니다. 역은 레일에 의해 직접 역에 연결됩니다.

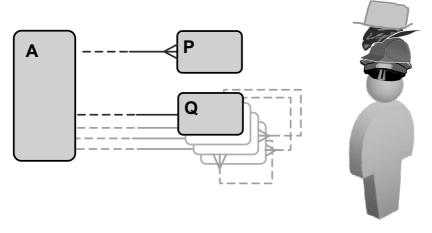
대칭 관계: 문제 대칭 관계에서 쌍 (S1, S2)가 유효할 경우 쌍(S2, S1)도 유효해야 합니다. 그래도 본질적으로 동일한 정보를 두 번 저장하는 것(데이터베이스 설계의 한 가지 기본 원칙에 배치됨)과 같이 두 쌍을 모두 기록하는 것은 그다지 의미가 없을 것입니다. 하지만 한 쌍만 기록할 경우, 어느 쌍을 기록해야 합니까? 그리고 다른 사람이 기록했다면 두 쌍 중 어느 것이 사용되었는지 어떻게 알 수 있습니까?

대칭 관계: 해결 방법 이러한 대칭 상황을 모델링하는 데 자주 사용되는 방법은 (S1, S2)를 Group1로, (S3, S4)를 Group2로 ...등등으로 생각하는 아이디어에 준한 것입니다. 관계를 이러한 방식으로 바라본다면, GROUP은 항상 정확히 두 개의 S의 인스턴스로 구성된다고 말할 수 있습니다. 모델과 테이블 구현은 다음과 같습니다.

패턴: 롤

• 특성: is / is 1:m (또는 1:1) 관계

• 은유법: 사람-많은 모자 (반드시 동시에 일어나는 것은 아님...)

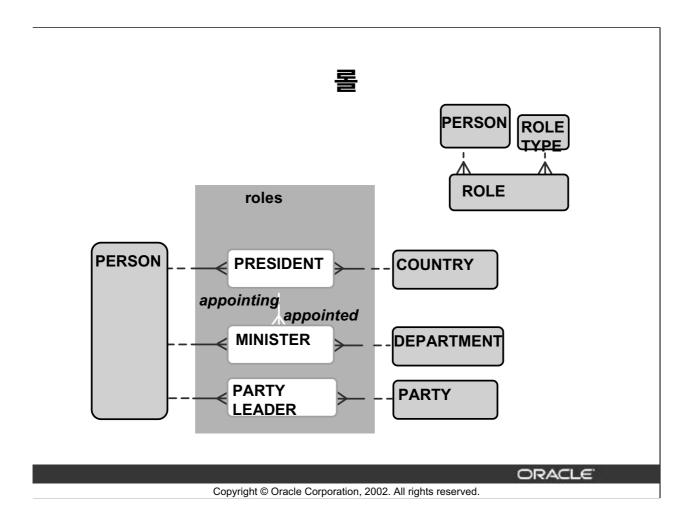


ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

롴

롤은 시스템이 기본적인 이름/주소/도시 정보 이외에 사람들에 대해 더 많은 정보를 알아야할 때 종종 발생합니다. 롤을 별도의 엔티티로 모델링하면 특정 롤에 대해 어느 속성이 필수인지를 보여주고, 필요한 경우 다양한 롤 간의 관계를 보여줄 수 있습니다. 아래 예제는 한 국가의 대통령으로서의 롤을 가진 사람이 장관 롤에 사람을 임명할 수 있다는 것을 보여줍니다. 아마도 "대통령직"과 "장관직"이라는 말이 도표에 표시된 것보다 이 개념에 더 가까울 것입니다.

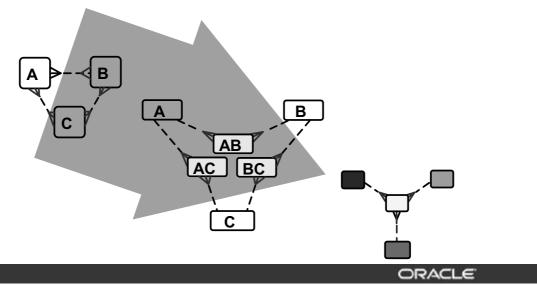


Oracle Internal & OAI Use Only

Fan Trap

• 특성: m:m 관련 엔티티의 고리

• 은유법: ABC 조합



Copyright © Oracle Corporation, 2002. All rights reserved.

Fan Trap

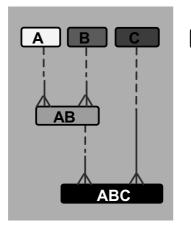
Fan Trap(솔루션의 특징적인 모양을 본 딴 이름)은 셋 이상의 엔티티가 m:m 관계를 통해 관련되어 있으면서 고리를 형성할 때 발생합니다. 일반적으로 이러한 관계는 여러 개의 m:1 관계를 가지는 중앙의 하나의 엔티티로 교체해야 합니다. fan trap 방지는 두 개의 엔티티 사이에서 m:m 관계를 분석하는 것과 유사합니다.

Trap이 발생하는 이유

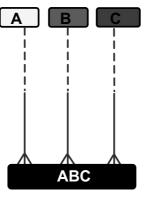
세 개의 m:m 관계를 분석하면 AB, BC 및 AC라는 세 개의 교차 엔티티로 나뉘며, 이들은 관련 쌍을 포함하게 됩니다. 하지만 AB와 BC를 조인하면 연습 3-8에서 보았던 것과 같이 AC가 포함하는 내용과 다른 정보를 초래할 수 있습니다.

그림에서와 같이 Trap을 피하는 방법은 여러 가지입니다. 컨텍스트에 따라 모두 옳을 수 있습니다.

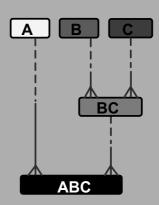
분석된 Fan Trap



Ab는 값 목록의 기능을 함



D



BC는 값 목록의 기능을 함

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

Oracle Internal & OAI Use Only

패턴: 데이터 웨어하우스

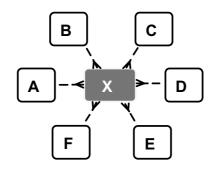
• 특성: 다차원, 수많은 디테일 인스턴스

• 은유법: 별 모델

별은 모양이 이상할 수 있습니다:

• 눈송이 모델







ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

데이터 웨어하우스

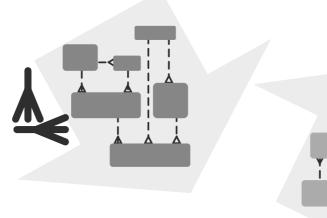
데이터 웨어하우스 시스템은 여타 시스템처럼 모델링될 수 있습니다. 데이터 웨어하우스는 여타 간단한 트랜잭션 처리 정보 시스템과 동일한 종류의 정보를 포함합니다. 데이터 웨어하우스는 주로 개요와 통계 분석을 위해 구축되기 때문에 일반적으로 덜 자세하고 요약된 정보를 포함합니다. 하지만 일반적으로 데이터 웨어하우스는 세부 정보를 포함하는 온라인 트랜잭션 시스템으로부터 입력을 받아들입니다. 데이터 웨어하우스는 종종 별 모양 모델을 가집니다. 이 모델은 축약되고 요약된 정보를 포함하는 하나의 중앙 엔티티(사실)와 세부 정보를 분류하고 그룹화하는 여러 개의 차원으로 구성됩니다.

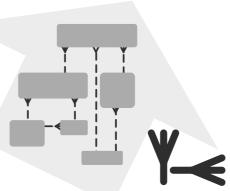
일반적인 차원은 다음과 같은 엔티티를 나타냅니다.

- 시간
- 장소
- 행위자(예: 판매 사원, 환자, 고객, 강사)
- 제품(예: 기사, 진료, 교과 과정)

종종 차원도 분류됩니다. 시간은 일, 주, 월, 분기, 연도로 구조화될 수 있습니다. 사용자는 앞의 예제에서 보았던 것과 같이 다양한 방식으로 제품을 분류할 수 있습니다. 이 경우, 모델은 일반적으로 눈송이의 수정 모양처럼 보이기 때문에 눈송이 모델이라고 기술됩니다.

그리기 규칙





이 규칙 중 하나를 따르는 한 어느 규칙을 선택하는가는 중요하지 않습니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

그리기 규칙

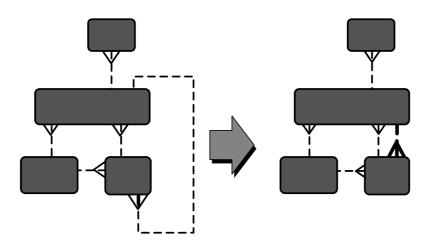
두 가지 그리기 규칙이 널리 사용됩니다. 하나는 대용량 엔티티를 종이의 상단에 놓는 규칙이고 또 하나는 그와 반대로 놓는 규칙입니다. 두 규칙 모두 교차하는 관계 선과 부분적으로 겹치는 엔티티, 엔티티와 교차하는 관계 선을 피하기 위한 것입니다. 어느 규칙을 선택하든 하나를 선택하여 일관되게 사용하십시오. 그래야 오류가 방지되며 대형 도표의 판독이 훨씬 더쉬워집니다.

구조를 변경하면 많은 사람들이 혼란을 겪게 되므로 모델링 프로젝트 동안 전체 레이아웃 구조를 변경하지 말고 유지하십시오.

각 업무 영역에 대해 별도의 도표를 만드십시오. 이러한 도표는 각기 다른 레이아웃을 가질 수 있으며, 주로 주제 전문가와 의사 소통을 하는 데 사용됩니다.

이 과정을 마치면 임의의 그리기 규칙으로 생성된 모델을 읽을 수 있고, 사용된 규칙에 따라모델을 완성할 수 있어야 합니다.

적절한 규칙 사용



그러나 가독성이 우선입니다.

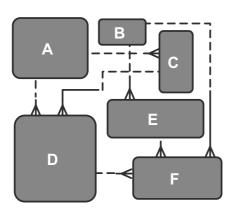
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

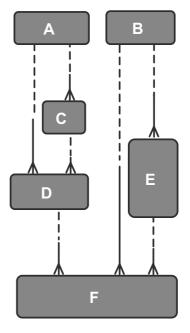
적절한 규칙 사용

도표(모델이 아님) 생성의 주요 목적은 의사 소통을 위해 사용될 수 있는 모델을 표현하는 것입니다. 이는 규칙이 가독성과 명료성을 방해하게 해서는 안된다는 것을 의미합니다. 가독성에는 공간이 필요하다는 우려는 하지 마십시오. 일반적으로 엔티티 모델은 미래 시스템의 특정한 기능 부분을 다루는 엔티티와 관계만 표시하는 몇 개의 도표로 나타냅니다. 여러 도표에 걸쳐 모델을 분할하면 가독성이 향상됩니다.

모델 가독성



- 공간이 필요함
- 취향에 따라 다를 수 있음

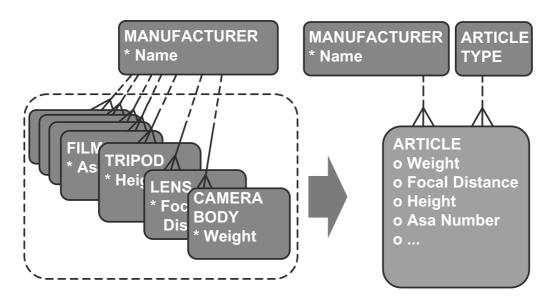


ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

Oracle Internal & OAI Use Only

일반 모델링



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

일반 모델링이란?

일반 모델링이란 좀 더 멀리 떨어진 다른 관점에서 동일한 컨텍스트를 보는 것입니다. 멀리에서 보면 많은 사물들이 똑같아 보입니다.

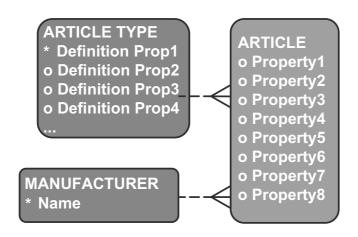
사진관용 모델을 만든다고 가정합시다. 이곳의 업무는 일반적으로 각종 품목(예: 카메라 본체, 소형 카메라, 렌즈, 필름)을 판매하는 것이고, 각 품목 유형마다 10개에서 500개 사이의 상이한 유형이 있다고 합시다. 이 경우, 모든 유형을 CAMERA BODY, LENS, FILM과 같은 하나의엔티티로 모델링할 수 있습니다.

또한 이 모두를 ARTICLE 엔티티의 하위 유형으로 모델링하거나, 하위 유형 없이 모두를 단순히 ARTICLE로 모델링할 수도 있습니다.

하지만 이러한 모델링은 작동하지 않을 것입니다. 예를 들면, 자주 새로운 종류의 품목이 사진관에 비축된다는 사실이 있습니다. 이럴 때마다 모델에 자체 속성을 가진 새로운 엔티티가 발생합니다.

시스템 주기 동안 ARTICLE TYPE의 새 인스턴스가 없었을 경우(또는 아주 적었을 경우) ARTICLE 엔티티를 가진 모델만 제대로 작동하는 모델이 됩니다.

일반 모델링



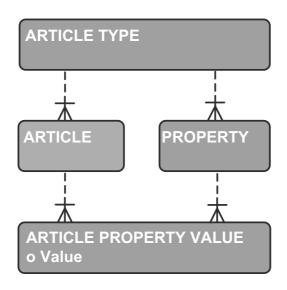
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

일반 모델

oracle internal 8. OAI Use 추가적인 일반 모델이 아래에 표시되어 있습니다. 이러한 모델은 특정 상황에서 유용할 수 있 습니다.

일반 모델



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

일반 모델

속성 재활용

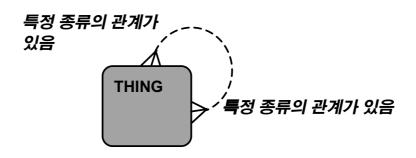
품목의 속성 수가 한정되어 있다고 간주하는 것이 안전할 경우 이 모델을 사용할 수 있습니다. 이 한계는 높은 수일 수 있지만 사전에 설정되어야 합니다. Property1은 ARTICLE of TYPE Film의 인스턴스에 대한 Asa Number를 포함할 수 있고 ARTICLE of TYPE Camera Body의 인스턴스에 대한 Weight를 포함할 수 있으며, 이런 식으로 계속 이어집니다. 이 모델의 가장 큰이점은 모델을 변경할 필요 없이 ARTICLE TYPE의 새 인스턴스를 추가할 수 있는 점입니다.

Property1, Property2, 등에 대해 입력해야 하는 정보의 유형은 예를 들면, Definition Prop1, ARTICLE TYPE의 속성을 사용하여 기술될 수 있습니다. 여기서 또한 이러한 속성의 데이터 유형에 대한 정보도 저장할 수 있습니다.

PROPERTY 인스턴스로 모델링된 속성

이 모델은 또 다른 접근 방식을 취합니다. ARTICLE의 PROPERTY에 대한 모든 값은 별도로 저장됩니다. 이 모델은 시스템 주기 동안 새로운 품목과 속성을 자유롭게 정의할 수 있도록 합니다.

일반적인 모델



ORACLE

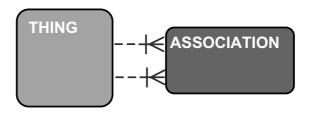
Copyright © Oracle Corporation, 2002. All rights reserved.

보다 일반적인 모델

Everything = "Thing"

Oracle Internal & OAI Use Only 세상은 사물(thing)과 관련될 수 있는 사물로 가득 차 있습니다.

보다 일반적인 모델



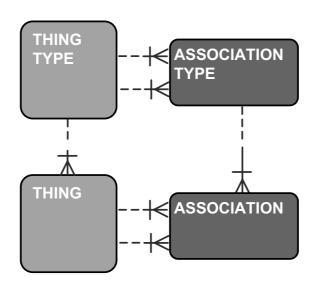
ORACLE

Oracle Internal & OAI Use Only Copyright © Oracle Corporation, 2002. All rights reserved.

보다 일반적인 모델

m:m 관계 분석

보다 일반적인 것 이상의 모델



ORACLE

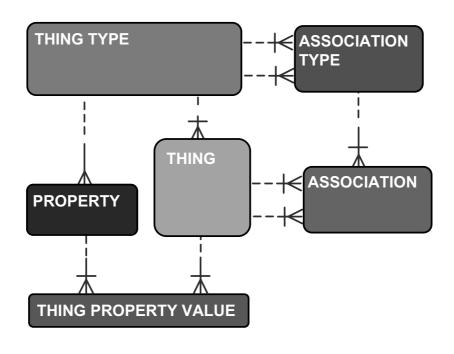
Copyright © Oracle Corporation, 2002. All rights reserved.

보다 일반적인 모델

이제 몇 가지 정의 정보를 추가하십시오.

oracle internal & 이것은 다소 일반적인 모델입니다. 사실 이것은 우주와 그 이상에 대한 모델입니다. THING 엔티티에 대한 속성 수는 상당히 많을 수 있습니다.

가장 일반적인 모델



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

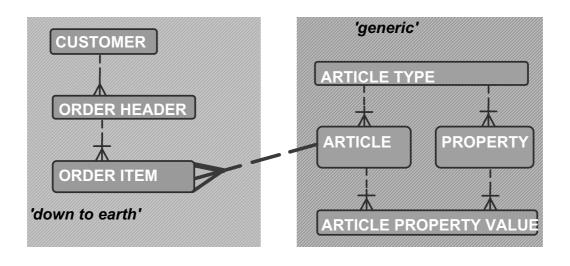
가장 일반적인 모델

이 모델은 "사물"과 속성/속성값의 개념을 결합함으로써 모든 사물이 원하는 수의 유형당 속성으로 표현될 수 있도록 합니다.

일반 모델링의 가치

일반 모델링을 사용하는 목적은 주로 개념적 데이터 모델의 카능한 향후 변경 수를 최소로 줄이는 것입니다. 이것은 시스템 주기 동안 유지 관리 비용을 절감시키기 때문에 커다란 이점이될 수 있습니다. 반면에 엔티티가 "실제" 사물이 아니기 때문에 프로그램의 초기 코딩 작업이좀 더 복잡합니다.

두 분야의 최고



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

가장 일반적인 모델

두 분야의 최고

그 일반적인사 많은 모델에서 사용자는 이해하기 쉽고 간단한 엔티티와 좀 더 일반적인 사물 같은 엔티티를 혼합하여 사용할 것입니다.

요약

패턴

- 유사성을 표시합니다.
- 바퀴는 한 번만 발명합니다.

일반 모델

- 엔티티 수를 극적으로 감소시킵니다.
- 구현이 좀 더 복잡합니다.
- 매우 유연합니다.
- 일반적으로 불안정한 상황에서 최상의 선택입니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

요약

패턴의 측면에서 생각하는 것이 개념적 데이터 모델에 대한 품질 검사를 수행하는 귀중한 방법입니다. 종종 한 컨텍스트에서의 제약 조건과 고려 사항은 간단한 변환을 통해 다른 컨텍스트로 전이될 수 있습니다.

모델에서 그리기 규칙을 사용하면 가독성과 명료성이 향상됩니다. 따라서 실수와 부정확성을 방지할 수 있습니다.

일반 모델링은 나중에 데이터 구조를 변경해야 하는 필요성을 방지하고 테이블과 프로그램 의 수를 극적으로 줄일 수 있습니다. 단점은 데이터 모델과 프로그램 모두에서 복잡성이 증가 한다는 것입니다.

연습

- 패턴
- 데이터 웨어하우스
- Argo 및 Erat
- 동의어

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



연습: 패턴

- 체스 게임에서의 게임 동작 모델
- 입찰 모델(견적)
- 조리법 모델
- 대학에 속한 모든 사람 모델: 학생, 교수, 학부모, ...
- 비디오 가계에서의 대여
- 프로세스의 단계 모델

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-1: 패턴

목표

이 연습의 목적은 제공된 컨텍스트에서 주된 패턴을 예측하는 것입니다...

과제

제공된 컨텍스트에서 어떤 패턴을 기대할 수 있습니까? 기대한 패턴을 찾을 수 없으면 모델을 개략적으로 그려보십시오. 상상력과 상식을 동원하십시오.

연습: 데이터 웨어하우스

- 지난 달 커피 판매량은 작년 동월 커피 판매량과 비교하여 달러로 얼마입니까?
- 일본의 1인당 커피 판매량은 전세계 Moonlight 커피점 이 있는 국가의 평균 커피 판매량과 비교하여 달러로 얼 마입니까?
- 스웨덴의 커피 판매량 성장은 동일 지역의 모든 제품의 판매량 성장과 비교하여 달러로 얼마입니까? 현지화로 는 얼마입니까?

ORACI E

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-2: 데이터 웨어하우스

목표

이 연습에서는 Moonlight Coffees Inc.의 데이터 웨어하우스에 대한 개념적 데이터 모델을 생성합니다.

시나리오

Moonlight에서는 판매점에서 매일 보고하는 세부 판매량을 기준으로 데이터 웨어하우스를 구축하려고 합니다. Moonlight가 데이터 웨어하우스를 통해 답변하려는 질문의 예제는 위와 같습니다.

과제

- 지금까지 생성한 Moonlight 모델을 확인하십시오. 이러한 모델을 통해 나열된 질문에 충분히 대답할 수 있습니까? 그렇지 않다면 모델을 적절히 변경하십시오.
- 데이터 웨어하우스 데이터 모델에 대해 중앙의 "facts" 엔티티를 제시하십시오.

연습: 데이터 웨어하우스

- 최소한 18개월 동안 영업한 판매점의 경우, 지난 달 총 커피 판매량은 작년 동월 총 커피 판매량과 비교하여 달 러로 얼마입니까?
- 비음식 판매량 성장은 음식 판매량 성장과 비교하여 달 러로 얼마입니까?
- 각국에서 어느 요일이 총 판매량이 가장 높습니까? 그 것은 평균 판매량과 어떤 관련이 있습니까? 가장 판매 량이 높은 요일은 위치 유형에 따라 다릅니까?

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-2: 데이터 웨어하우스

목표

이 연습에서는 Moonlight Coffees Inc.의 데이터 웨어하우스에 대한 개념적 데이터 모델을 생성합니다.

시나리오

Moonlight에서는 판매점에서 매일 보고하는 세부 판매량을 기준으로 데이터 웨어하우스를 구축하려고 합니다. Moonlight가 데이터 웨어하우스를 통해 답변하려는 질문의 예제는 위와 같습니다.

과제

- 지금까지 생성한 Moonlight 모델을 확인하십시오. 이러한 모델을 통해 나열된 질문에 충분히 대답할 수 있습니까? 그렇지 않다면 모델을 적절히 변경하십시오.
- 데이터 웨어하우스 데이터 모델에 대해 중앙의 "facts" 엔티티를 제시하십시오.

연습: 데이터 웨어하우스

- 국가마다 가장 수익성이 높은 제품은 어느 제품입니까?
 전세계적으로는 어느 제품입니까?
- 서비스 수준(판매된 1000개의 품목당 직원 수)은 판매 에 영향을 끼칩니까?

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-2: 데이터 웨어하우스

목표

이 연습에서는 Moonlight Coffees Inc.의 데이터 웨어하우스에 대한 개념적 데이터 모델을 생성합니다.

시나리오

Moonlight에서는 판매점에서 매일 보고하는 세부 판매량을 기준으로 데이터 웨어하우스를 구축하려고 합니다. Moonlight가 데이터 웨어하우스를 통해 답변하려는 질문의 예제는 위와 같습니다.

과제

- 지금까지 생성한 Moonlight 모델을 확인하십시오. 이러한 모델을 통해 나열된 질문에 충분히 대답할 수 있습니까? 그렇지 않다면 모델을 적절히 변경하십시오.
- 데이터 웨어하우스 데이터 모델에 대해 중앙의 "facts" 엔티티를 제시하십시오.

연습: Argo 및 Erat

"Erat는 고유한 이름을 가집니다. Erat는 Argo를 가질 수 있습니다. Argo도 이름을 가집니다. Argo의 이름은 Argo가 속해 있는 Erat 내에서 고유해야 합니다. Erat는 서로 rondel을 가집니다. rondel의 유형은 극히 소수입니다. Erat는 하나 이상의 ubin을 가집니다. ubin은 항상 erat의 하나 이상의 argo, erat의 하나 이상의 rondel 또는 두 가지의 조합으로 구성됩니다."

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-3: Argo 및 Erat

Olacle

목표

정보를 모델링할 때 의식하지 않아도 많은 가정을 하게 됩니다. 대부분의 이러한 가정은 일반 적으로 비슷한 상황에서의 경험이나 상식에 기준하기 때문에 정확할 가능성이 높습니다.

이 연습은 이에 대한 인식을 높이는 데 도움을 줍니다.

시나리오

이 연습에 대한 시나리오는 Stranger in a Strange Land입니다. Lost in Darkness. Wanderer in the Mist. 어느 것이든 상관 없습니다!

과제

텍스트의 정보를 기준으로 개념적 데이터 모델을 만드십시오. 도표에서 텍스트를 통해 확인할 수 있는 모든 부분에 표시하십시오.

연습: 동의어

practice - exercise

order - command

entity - being

order - sequence

order - arrangement

command - demand

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-4: 동의어

시나리오

사전에 의하면 동의어란 "다른 단어와 의미가 동일한(일반적으로 거의 동일한) 단어"입니다.

과제

. 필을 만드십 동의어 사전의 기초가 될 수 있는 개념적 데이터 모델을 만드십시오



엔티티 모델 매핑

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



개요

- 왜 설계 모델링을 사용합니까?
- 구성 요소 소개:
 - 테이블
 - _ 열
 - 제약 조건
- 기본 매핑
- 복합 매핑

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

소개

이 단원에서는 관계형 데이터베이스의 원리와 엔티티 관계 모델을 물리 데이터베이스 설계 로 변환할 때 사용할 수 있는 여러 가지 기술을 설명합니다.

목표

- 이 단원을 마치면 다음을 수행할 수 있습니다.

왜 데이터 디자인 모델을 생성합니까?

- 구현 솔루션에 가깝습니다.
- 논의를 원할하게 합니다.
- 적합한 모델을 RDBMS 모델에 맞게 조정할 수 있습니다.
- 물리 데이터베이스 설계에 적절한 기준입니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

왜 데이터베이스 설계를 생성합니까?

엔티티 관계 모델은 업무에 필요한 데이터를 설명합니다. 이 모델은 구현 고려 사항과 전적으로 무관합니다. 이 동일한 ER 모델을 DBMS 유형의 구현 기준 또는 파일 시스템으로도 사용할 수 있습니다.

새 시작점

엔티티 관계 모델은 있는 그대로 구현하기 어려운 상위 레벨의 표현 방식입니다.

이러한 모델을 생성하는 사람은 물리적 제약 조건과 데이터베이스 제약 조건을 정확히 알지 못해도 개념적으로 "작동 가능한" 솔루션을 제공해야 합니다. 왜냐하면 물리 데이터베이스 설계를 시작하기 전에 검증 및 합의된 ER 모델이 있어야 하기 때문입니다.

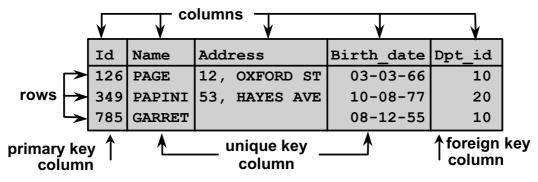
ER 모델을 변환할 때는 "기본(first-cut)" 데이터베이스 설계를 생성합니다. 이 기본 설계는 물리 데이터베이스 구현을 정의하는 데 새로운 기반으로 삼기 위한 것입니다.

이 새로운 모델은 설계자, 개발자 및 데이터베이스 관리자 간의 보다 원활한 논의를 위해 사용될 수 있습니다.

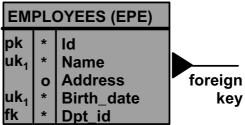
데이터 설계 모델은 ER 모델(이 과정에서 권장하고 따르는 하향식 접근 방법)에서 파생될 수 있지만 기존 데이터베이스에서 직접 생성할 수도 있습니다.

테이블 표시

테이블: EMPLOYEES



테이블 도표: EMPLOYEES



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

테이블 표시

테이블은 데이터베이스의 데이터와 구조를 보호하는 무결성 규칙에 의해 지원됩니다. 무결성 규칙을 사용하려면 각 테이블에 하나의 기본 키와 해당 기본 키와 연관되는 각 외래 키가 있어야 합니다.

테이블:

테이블은 데이터가 구성되고 저장되는 간단한 구조입니다. 테이블에는 열과 행이 있습니다. 각 열은 특정 유형의 값을 저장하는 데 사용됩니다. 위의 예제에서 EMPLOYEES 테이블은 사 원 정보를 저장하는 데 사용된 구조입니다.

행:

각 행은 사원을 설명합니다. 예제에서 각 행은 시스템에 필요한 속성 전체를 설명합니다.

열:

각 열에는 사원 번호, 이름, 주소, 생일 및 사원이 할당된 부서의 ID와 같은 특정 유형의 정보가 있습니다.

테이블 표시

기본 키:

Id 열은 기본 키입니다. 즉, 모든 사원은 이 테이블에서 각 개별 행을 구분하는 고유한 식별 번호가 있습니다.

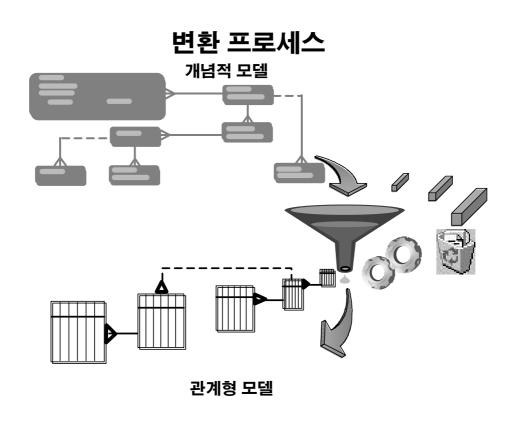
고유키:

Name 열과 Birth_date 열은 모두 고유 키 제약 조건과 연관됩니다. 즉, 두 행에서 동일한 이름과 Birth_date를 가질 수 없습니다. 이 제약 조건은 시스템에 대한 제한을 정의합니다.

외래 키:

외래 키 열에서는 Dpt_id 값을 사용하여 특정 사원이 근무하고 있는 부서의 속성을 검색할 수 있습니다.





ORACI E

Copyright © Oracle Corporation, 2002. All rights reserved.

변환 프로세스

변환 규칙을 사용하여 개념적 모델을 기반으로 새 모델을 생성합니다.

개념적 모델

데이터 업무에 대한 요구 사항을 설명하려면 그래픽 표현을 통해 의미상 풍부한 구문을 사용해야 합니다. 이전 장에서 살펴본 바와 같이 하위 유형, 호 및 관계(세로줄 표시가 있는 관계와비전이적 관계)와 같은 그래픽 요소로 많은 업무 규칙을 설명할 수 있습니다. 지금까지 업무복잡성을 나타내는 데 유일한 제약 조건은 그래픽적 요소입니다. 이 모델은 물리적 고려 사항과 관련이 없으므로 일반 모델로 사용됩니다. 따라서 모든 유형의 데이터베이스에 이 모델을 사용할 수 있습니다. 하지만 사용하고자 하는 DBMS 유형(관계형 또는 기타)이 ER 모델에 그래픽으로 표시된 모든 의미 규칙을 지원하지 않을 수도 있습니다.

관계형 모델

관계형 모델은 수학 규칙에 기반을 두고 있습니다. 즉, ER 모델의 모든 구문을 물리적 데이터 베이스 모델에 맞출 때 모델의 일부가 관계형 모델에 대응하지 않을 수 있습니다. 지정된 이러한 규칙을 유지하려면 이러한 규칙을 추적하고 정확한 구현 방법을 찾아야 합니다.

용어 매핑



ORACLE

Al Use Only

Copyright © Oracle Corporation, 2002. All rights reserved.

용어 매핑

모델에 따라 용어가 바뀝니다.

간단한 기준을 사용하면 다음과 같습니다.

- 엔티티는 테이블이 됩니다.
- 속성은 열이 됩니다.
- 기본 고유 식별자는 기본 키를 생성합니다.
- 보조 고유 식별자는 고유 키를 생성합니다.
- 관계는 외래 키와 외래 키 열로 변환됩니다.
- 제약 조건은 일관성을 위해 데이터베이스가 처리해야 하는 규칙입니다. 일부 데이터베이스 규칙은 검사 제약 조건으로 변환되고 기타 복잡한 규칙은 추가 프로그래밍이 필요하며 클라이언트측이나 서버측 또는 클라이언트와 서버측 모두에서 이러한 규칙을 구현할 수 있습니다.

ER 모델의 이러한 초기 매핑은 테이블 설계, 열 및 선언될 수 있는 제약 조건으로 한정됩니다. 선언 제약 조건은 데이터베이스 언어 명령문만 사용하여 서버 레벨에서 확인할 수 있고 코딩 이 필요하지 않은 업무 제약 조건입니다.

일반 이름 지정 항목

다음에 대한 규칙을 결정합니다.

- 테이블 이름
- 특수 문자(%, *, #, -, 공백, ...)
- 테이블 단축 이름
- 열이름
- 기본 및 고유 키 제약 조건 이름
- 외래 키 제약 조건 이름
- 외래 키 열 이름

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

이름 지정 규칙

프로젝트를 사용하는 사용자가 동일한 표준을 사용하여 동일한 소스에서 동일한 모델을 생성할 수 있도록 ER 도표로 변환하기 전에 이름 지정 규칙을 정의해야 합니다. 여기에 설명된 규칙은 Oracle 내에서 사용되는 규칙입니다. 이 규칙이 효율적이지만 이 규칙만 사용할 수 있는 것은 아닙니다. 회사의 표준을 방법의 일부로 제공할 수 있습니다.

테이블 이름 지정

엔티티 이름의 복수형은 해당 테이블 이름으로 사용됩니다. 엔티티는 추상적인 개념입니다. EMPLOYEE, CUSTOMER 등에 대해 언급할 수 있으므로 이름 지정 규칙에는 단수가 적합하지만 테이블은 여러 개의 행(EMPLOYEES 테이블 또는 CUSTOMERS 테이블)으로 구성되므로 복수가 더 적합합니다.

열이름지정

열 이름은 속성 이름과 동일하며 몇 가지 예외가 있습니다. 특수 문자를 밑줄 문자로 바꿉니다. 특히, SQL은 관계 요소 이름에 공백을 사용할 수 없으므로 속성 이름에서 공백을 제거합니다. Start Date 속성은 Start_date 열로 변환되고, Delivered Y/N 속성은 Delivered_y_n(또는, 가능하면 Delivered_Ind)으로 변환됩니다. 종종 속성 이름보다 열 이름에 약어를 많이 사용합니다.

이름 지정 규칙

단축 이름

모든 테이블에 대한 고유한 단축 이름은 외래 키 열이나 외래 키 제약 조건의 이름을 지정할 때 매우 유용하게 사용할 수 있습니다. 이러한 단축 이름은 다음 규칙을 기준으로 지정하는 것이 좋습니다.

- 둘 이상의 단어로 된 엔티티 이름의 경우 다음을 사용합니다.
 - 첫번째 단어의 첫 문자
 - 두번째 단어의 첫 문자
 - 마지막 단어의 마지막 문자
- 예를 들면, PRICED PRODUCT 엔티티의 단축 테이블 이름은 PPT가 됩니다.
- 한 단어이지만 둘 이상의 음절로 된 엔티티 이름의 경우 다음을 사용합니다.
 - 첫번째 음절의 첫 문자
 - 두번째 음절의 첫 문자
 - 마지막 음절의 마지막 문자
- 예를 들면, EMPLOYEE의 단축 이름은 EPE가 됩니다.
- 한 음절이지만 둘 이상의 문자로 구성된 엔티티 이름의 경우 다음을 사용합니다.
 - 첫번째 문자
 - 두번째 문자
 - 마지막 문자
 - 예를 들면, FLIGHT는 FLT가 됩니다.

이 단축 이름 구성 규칙에 따라 단축 이름의 고유성을 보장할 수는 없지만 지금까지의 경험상 이름이 중복되는 경우는 상대적으로 거의 없습니다.

두 개의 단축 이름이 같을 경우 자주 사용되지 않는 단축 이름에 숫자를 추가합니다. 예를 들면, 가장 많이 사용되는 단축 이름을 CTR로 지정하고 다른 것에는 CTR1로 지정합니다.

외래 키 제약 조건 이름 지정

외래 키 제약 조건 이름 지정 규칙은 다음과 같습니다. <원본 테이블의 단축 이름> <대상 테이블의 단축 이름> <fk>

예를 들어, EMPLOYEES 테이블과 DEPARTMENT 테이블 사이에 외래 키를 사용하면 제약 조건 이름은 epe dpt fk가 됩니다.

외래 키열이름지정

외래 키 열은 참조하는 테이블의 단축 이름이 앞에 붙여집니다. 따라서 외래 키 열 이름은 dpt_no와 같이 됩니다. 속성 이름을 22 문자로 제한하면 열 이름에 두 개의 접두어와 두 개의 밑줄 문자를 추가할 수 있습니다. 이러한 상황은 연쇄 세로줄 표시 관계에서 발생할 수 있습니다. 이에 대해서는 단원의 뒷부분에서 설명합니다.

이름 지정 규칙

두 테이블 사이에 두 개(이상)의 외래 키가 있으면 그 외래 키와 외래 키 열에 같은 이름이 지정됩니다. 이런 경우에는 관계 이름을 두 외래 키 이름에 추가합니다. 외래 키 열에도 똑같이 수행합니다. 이 방법을 사용하면 두 외래 키가 혼동되지 않습니다.

예를 들면, Electronic Mail 모델에서 LIST ITEM 엔티티에는 ALIAS와 두 가지 관계가 있으며, 이 중 한 관계는 하위 유형 레벨입니다. 이름을 지정하면 lim_als_in과 lim_als_referring_to의 두 외래 키 이름이 만들어집니다. 외래 키 열 이름은 Als id in과 Als id referring to가 됩니다.

검색 제약 조건 이름 지정

검색 제약 조건 이름은 <테이블 단축 이름>_ck_<일련 번호>가 됩니다. 예를 들면, EMPLOYEES 테이블의 첫번째와 두번째 검색 제약 조건의 경우 epe_ck_1, epe_ck_2가 됩니다.



Oracle의 이름 지정 제한

- 테이블 및 열 이름:
 - 문자로 시작해야 합니다.
 - 30자까지 문숫자를 사용할 수 있습니다.
 - 공백이나 "!"와 같은 일부 특수 문자를 사용할 수 없습니다.
- 테이블 이름은 스키마 내에서 고유해야 합니다.
- 열 이름은 테이블 내에서 고유해야 합니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

Oracle의 이름 지정 제한

RDBMS마다 고유한 이름 지정 제한 규칙을 가질 수 있습니다. 사용하려는 규칙이 RDBMS와 호환되는지 알고 있어야 합니다.

- 다음과 같은 이름의 경우 이름을 지정할 때 문숫자를 사용할 수 있습니다.
 - 문자로 시작합니다.
 - 최대 30자입니다.
 - "!"와 같은 특수 문자는 포함되지 않지만 "\$", "#", " "는 사용할 수 있습니다.
- 테이블 이름은 뷰 및 동의어를 사용하여 공유되는 스키마 내에서 고유해야 합니다.
- 같은 테이블 내의 두 열은 동일한 이름을 사용할 수 없습니다.
- 객체 이름 지정에 사용할 수 없는 프로그래밍 언어의 예약어를 알아두십시오. 다음과 같은 이름은 사용하지 않습니다.
 - Number
 - Sequence
 - Values
 - Level
 - Type
- 테이블이나 열 이름 지정에 대해서는 RDBMS 참고서를 참조하십시오.

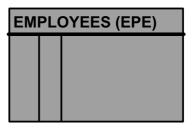
엔티티에 대한 기본 매핑

1 – 엔티티

테이블 이름: EMPLOYEES

단축 이름: EPE

EMPLOYEE



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

Oracle Internal & OAI Use Only

속성에 대한 기본 매핑

- 1 엔티티
- 2 속성

테이블 이름: EMPLOYEES

단축 이름: EPE

EMPLOYEE # Id

- * Name
- o Address
- * Birth Date

* Id
* Name
o Address
Birth_date

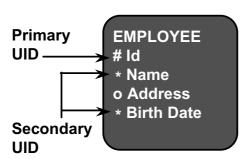
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

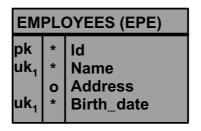
Oracle Internal & OAI Use Only

고유 식별자에 대한 기본 매핑

- 1 엔티티
- 2 속성
- 3 고유 식별자



테이블 이름: EMPLOYEES 단축 이름: EPE



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

기본 매핑

엔티티 매핑

복잡한 변환에 들어가기 전에 간단한 엔티티를 변환하는 방법을 살펴보겠습니다.

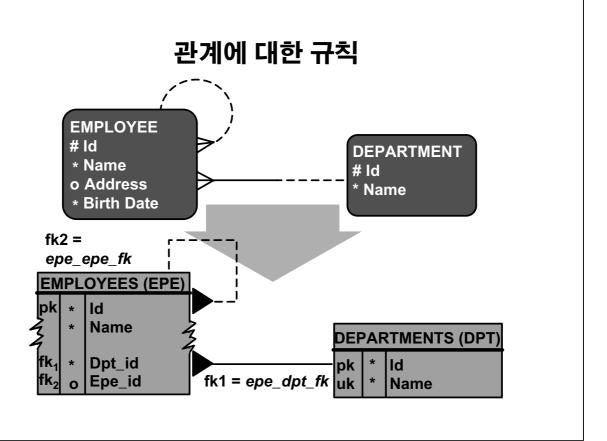
- 사용자가 만든 이름 지정 규칙이나 이전에 설명한 이름 지정 규칙을 사용하여 엔티티를 테이블로 변환합니다.
- 이 예제에서 엔티티 EMPLOYEE는 테이블 이름 EMPLOYEES와 단축 이름 EPE를 생성합니다.
- 상자를 사용하여 도표에 테이블을 나타냅니다.
- 각 속성은 테이블에 열을 생성하고 필수 또는 선택 사항과 같은 특성은 각 열에 유지되어야 합니다. "*"나 "o" 등 동일한 표기법을 사용하면 도표에서 이러한 특성을 쉽게 인식할 수 있습니다.
- 고유한 모든 식별자가 변환됩니다. 기본 고유 식별자는 기본 키로 변환됩니다. 열 이름 옆에 있는 "pk" 표시는 기본 키 속성을 나타냅니다. 둘 이상의 열이 기본 키의 일부이면 각 열에 "pk" 표시를 사용합니다.

기본 매핑

보조 고유 식별자

보조 고유 식별자가 ER 도표에 표시되지 않아도 보조 고유 식별자를 구현해야 합니다. 이러한 속성을 보존하려면 보조 UID를 고유 키로 변환합니다. 위 예제에서 두 열의 조합 값은 고유해야 합니다. 이 값은 동일한 고유 키에 속하므로 각 열에 이를 나타내는 uk1 표시가 있습니다. 나중에 해당 테이블에 또 다른 고유 키가 생기면 이 고유 키는 uk2로 표시됩니다.





Copyright © Oracle Corporation, 2002. All rights reserved.

ORACLE

관계에 대한 규칙

외래키열:

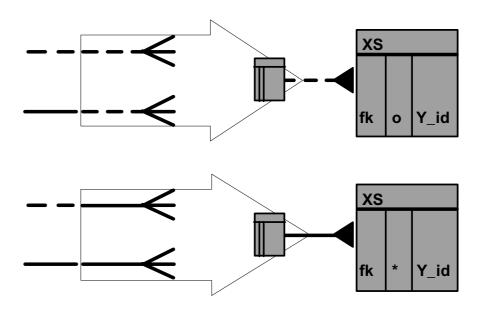
관계는 여러 측면에서 테이블에 하나 이상의 외래 키 열을 생성합니다. 기존 이름 지정 규칙을 사용하면 이 외래 키 열의 이름은 Department와의 관계의 경우 Dpt_id가 되고 순환 관계의 경우 Epe_id가 됩니다. 따라서 Id처럼 다른 테이블에서 가져온 열 이름은 테이블에 고유한 열이름을 제공합니다.

관계가 필수인지 아닌지 여부에 따라 외래 키 열은 필수이거나 선택 사항입니다.

외래 키 제약 조건:

EMPLOYEES와 DEPARTMENTS 간의 외래 키 제약 조건은 epe_dpt_fk입니다. EMPLOYEES와 EMPLOYEES 간의 순환 제약 조건은 epe_epe_fk입니다.

1:m 매핑 관계



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

관계 매핑

1:m 매핑 관계

앞에서 언급한 바와 같이 ERD에 표현된 일부 의미는 물리 데이터베이스 설계에서 재생성할 수 없습니다.

ER 도표에서 관계는 두 가지 관점에서 두 엔티티 사이에 적용되는 규칙을 나타냅니다. ERD에 사용된 표기법은 예를 들어, 관계가 양쪽에서 필수임을 알 수 있을 정도로 풍부합니다. 그림은 한쪽에서 필수인 1:m 관계가 한쪽에서 선택 사항인 관계와 같은 방식으로 구현된 것임을 보여줍니다. 즉, 관계 모델 제한 사항으로 인해 ER 모델의 내용 일부가 변환 도중 손실됩니다. 이러한 불완전한 변환을 추적해야 합니다. 이러한 변환은 선언 제약 조건 이외의 처리 방식을 사용하여 구현해야 합니다.

관계 매핑

한쪽에서 필수 관계 매핑

한쪽에서 필수인 관계를 구현하는 경우 다음 두 가지 사항을 확인해야 합니다.

- 적어도 한 개의 디테일 레코드가 있어야 마스터 레코드를 생성할 수 있습니다.
- 디테일을 삭제할 때 마스터 레코드에 대한 마지막 디테일을 삭제하지 않도록 해야 합니다. 또는 마스터 레코드를 삭제할 때 해당 레코드의 마지막 디테일도 같이 삭제해야 합니다.

서버측이나 클라이언트측에서 이를 확인하기 위해 코드를 구현할 수 있습니다. Oracle 환경에서는 일반적으로 클라이언트측에서 수행되었습니다. 이전 리리스와 달리 Oracle 8부터 서버측 구현이 가능해졌습니다.

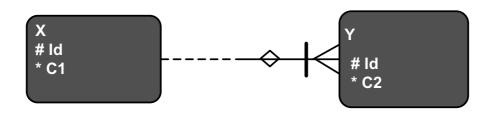
선택적으로 구성된 외래 키

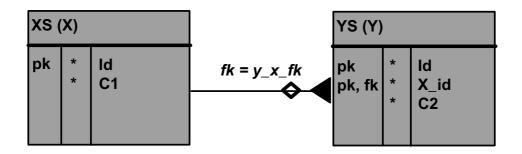
외래 키가 둘 이상의 열로 구성되고 외래 키가 선택 사항이면 모든 외래 키 열은 선택 사항으로 정의해야 합니다. 외래 키 열 중 한 열에만 값을 입력하고 다른 열에는 입력하지 않으면 Oracle은 외래 키 제약 조건 검사를 실행하지 않습니다.

모든 외래 키 열에 값이 있는지 또는 없는지 여부를 검사하려면 추가 코드가 있어야 하지만 키를 부분적으로 입력할 가능성은 제외합니다.



세로줄 표시가 있는 관계와 비전이적 관계의 매핑





ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

관계 매핑

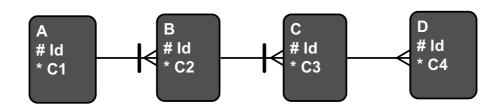
비전이적인 관계 매핑

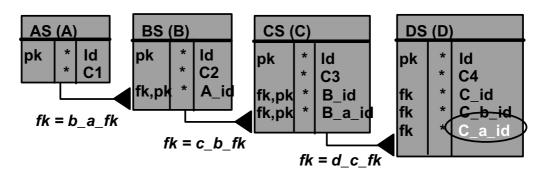
서버측에서 솔루션을 코드화할 수 있지만 이 관계 속성은 RDBMS에 자연적인 대응 속성이 없기 때문에 물리 데이터베이스 설계로 이전되지 않습니다. 예제에서는 YS 테이블에 X_i d 외래 키 열이 갱신되면 실패하는 갱신 트리거를 생성합니다.

세로줄 표시가 있는 관계 매핑

세로줄 표시가 있는 관계는 다른 모든 관계와 마찬가지로 외래 키에 매핑됩니다. 외래 키 열은 기본 키의 일부이므로 두 가지 역할을 합니다.

연쇄적인 세로줄 표시가 있는 관계 매핑





ORACL

Copyright © Oracle Corporation, 2002. All rights reserved.

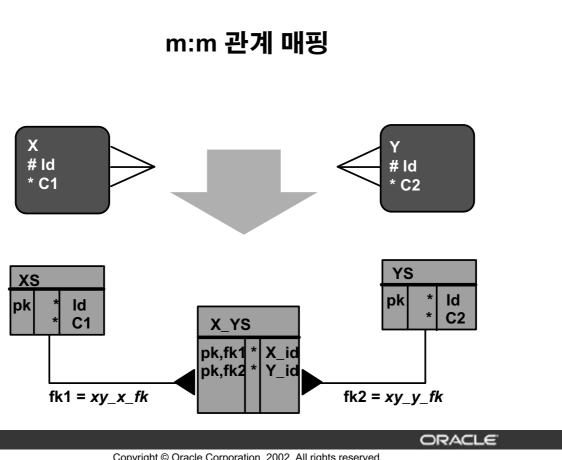
관계 매핑

연쇄적인 세로줄 표시가 있는 관계 매핑

연쇄적인 세로줄 표시가 있는 관계로 인해 그림에서처럼 긴 열 이름이 생성될 수 있습니다. 열 이름이 30자를 초과하지 않게 하려면 셋 이상의 테이블 접두어를 사용하지 마십시오. 일반적으로 다음과 같은 외래 키 열 이름을 사용합니다.

<가장 가까이에 있는 테이블 단축 이름> _ <가장 멀리 있는 테이블 단축 이름> _ <열 이름> 위의 예제에서 AS부터 BS 및 CS의 순서로 가져온 DS의 외래 키 열 이름은 $C_b_a_i$ d가 아닌 C_a_i d입니다.

단축 이름은 일반적으로 3자이므로 이 규칙에 따라 속성 이름이 22자를 넘지 않는 이유를 알수 있습니다.



Copyright © Oracle Corporation, 2002. All rights reserved.

관계 매핑

m:m 관계 매핑

m:m 관계를 변환할 때 교차 테이블을 생성합니다.

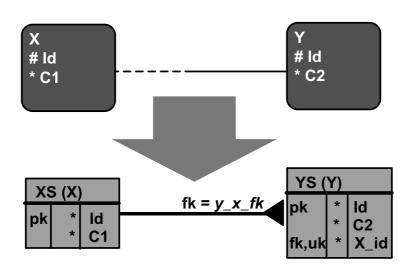
교차 테이블은 XS와 YS 사이에 존재하는 모든 조합을 포함합니다.

이 테이블에는 외래 키 열 이외의 열은 없습니다. 이러한 열은 모두 기본 키를 구성합니 다.

only

- 이 테이블 이름 지정 규칙은 첫번째 테이블의 단축 이름(문자순)과 두번째 테이블의 전 체 이름입니다. 이 규칙은 EMPLOYEES 테이블과 PROJECTS 테이블 간의 m:m 관계에 EPE PROJECTS라는 교차 테이블을 부여합니다.
- 관계가 필수이든 아니든, 외래 키 열은 항상 필수입니다.
- 이 테이블은 m:m 관계를 대체할 수 있는 교차 엔티티의 결과로 생성된 테이블과 동일(이름 제 외)합니다. Oracle

1:1 관계 매핑



다른 기수의 FK에 대한 쪽을 선택합니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

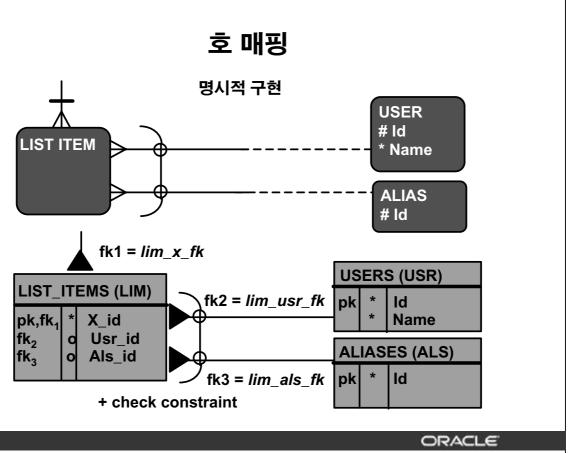
관계 매핑

1:1 관계 매핑

1:1 관계를 변환할 때 외래 키와 고유 키를 생성합니다. 이 외래 키의 모든 열은 고유 키의 일부이기도 합니다. 관계가 한쪽에서 필수이면 해당 테이블에 외래 키가 생성됩니다. 관계가 양쪽에서 필수이거나 선택 사항이면 외래 키를 생성할 테이블을 선택할 수 있습니다. 이를 구현할 쪽을 결정하는 절대적인 규칙은 없습니다. 관계가 양쪽에서 선택 사항이면 행수가 적은 테이블에 외래 키를 구현하도록 결정할 수 있습니다. 이렇게 하면 공간이 절약됩니다. 관계가 양쪽 끝에서 필수이면 앞에서 본 동일한 RDBMS 제한이 생깁니다. 따라서 한 끝에서 필수인 m:1 관계를 구현하기 위해 수행했던 것처럼 한 끝에서 필수 관계를 검사하는 코드를 작성해야 합니다.

다른 구현 방식

두 엔티티 간의 1:1 관계는 단일 테이블에 의해 구현될 수 있습니다. 이 방식은 제일 먼저 고려해야 할 구현 방식입니다. 이 방식은 외래 키 제약 조건이 필요하지 않습니다. 세번째로 가능한 구현은 관계가 마치 m:m 유형인 것처럼 교차 테이블을 생성하는 것입니다. 교차 테이블의각 외래 키 열은 고유 키의 일부가 됩니다.



Copyright © Oracle Corporation, 2002. All rights reserved.

관계 매핑

호 매핑

위에 설명된 첫번째 솔루션은 관계 수만큼 외래 키가 생성됨을 보여줍니다. 따라서 외래 키중 하나가 채워지면 나머지 외래 키는 채워지지 않아야 하는지(한 호의 관계에 대한 배타 원칙), 그리고 필수 조건을 구현하기 위해 하나의 외래 키 값이 항상 있어야 하는지를 확인하도록 규칙을 설정해야 합니다.

도표에서 보면 모든 외래 키는 선택 사항이지만 추가 코드는 논리적 제어를 수행합니다. 서버 측의 한 가지 솔루션은 LIST ITEMS에 다음과 같이 검색 제약 조건을 생성하는 것입니다.

• CHECK (usr_id IS NOT NULL AND als_id IS NULL) OR(usr_id IS NULL AND als_id IS NOT NULL).

이것은 필수 관계에 대한 배타성을 제어합니다.

관계가 선택 사항이면 다음을 추가해야 합니다.

• OR (usr_id IS NULL AND als_id IS NULL)

관계 매핑

호 구현

자주 사용되는 기타 구문은 다음과 같습니다.

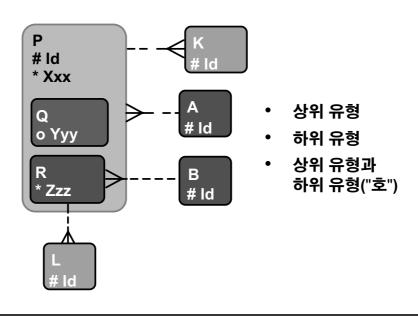
- DECODE (usr id,NULL,0,1)
- + DECODE (als id,NULL,0,1) =1;
- (또는 선택적 관계인 경우 =<1)

일반 호 구현을 사용하여 다른 방법으로 호를 매핑할 수도 있습니다. 이 방법은 기존 시스템에서 사용되던 솔루션입니다. 새 시스템에서는 이 방법을 사용하지 마십시오. 이에 대해서는 설계 고려 사항 단원에서 자세히 설명합니다.



하위 유형 매핑

다양한 구현 옵션



Copyright © Oracle Corporation, 2002. All rights reserved.

ORACLE

하위 유형 매핑

하위 유형 매핑에서 세 가지 다른 유형의 구현 중 선택해야 합니다. 세 가지 유형 모두 자세히 설명되어 있습니다.

상위 유형 구현

이 옵션은 P, Q, R 엔티티 구현을 위해 하나의 단일 테이블을 생성합니다. 상위 유형 구현은 단일(또는 하나의) 테이블 구현이라고도 합니다.

하위 유형 매핑

규칙

- 1. 테이블:
 - 하위 유형 수에 관계 없이 하나의 단일 테이블만 생성됩니다.

2. 열:

- 테이블은 원래 선택 가능성이 있는 상위 유형의 모든 속성에 대한 하나의 열을 가 져옵니다.
- 테이블은 또한 하위 유형에 속한 각 속성에 대해 하나의 열을 얻지만 열은 모두 선 택 사항으로 전환됩니다.
- 또한 엔티티의 다른 하위 유형을 구분하는 식별자 열 역할을 담당할 하나의 필수 열이 생성됩니다. 이 열은 모든 하위 유형의 단축 이름 집합(예제에서 DBE, DBU) 에서 값을 가져올 수 있습니다. 일반적으로 이러한 식별자 열을 <테이블 단축 이름> type이라고 하며, 예제에서 Dba type이 식별자 열이 됩니다.

3. 식별자:

- 고유 식별자는 기본 키와 고유 키로 변환됩니다.
- 하위 유형 레벨에서 고유 식별자는 일반적으로 고유 키나 검사 제약 조건으로만 변환됩니다.

4. 관계:

- 상위 유형 레벨에서 관계는 보통 때와 마찬가지로 변환됩니다. 하위 유형 레벨에 서 관계는 외래 키로 구현되지만 외래 키 열은 모두 선택 사항이 됩니다.
- 5. 무결성 제약 조건:
 - 각각의 특정 하위 유형에 대해 필수 속성에 속하는 모든 열이 NULL 값이 아닌지 확인해야 합니다.
 - 각각의 특정 하위 유형에 대해 다른 하위 유형의 속성 또는 관계에 속하는 모든 열 이 NULL 값인지 확인해야 합니다.



상위 유형 구현 Ρ # Id * Xxx **PS (P)** Q # Id о Үуу ld pk Xxx 0 Yyy # Id Zzz 0 fk_1 A id fk_2 B id 0 필수 P_type 식별자 열 추가 제약 조건 ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

하위 유형 매핑

주: 각 하위 유형에 하나의 필수 속성이 있을 경우 식별자 열을 사용하지 않을 수 있습니다. 이 러한 열에서는 직접 검사를 수행하여 특정 행이 속한 유형을 찾을 수 있습니다.

상위 유형 구현을 고려할 시기

단일 테이블 구현이 일반적이며 융통성 있는 구현 방식입니다. 이 방식은 제일 먼저 고려하는 방식이며, 다음과 같은 경우 특히 유용합니다.

- 대부부의 속성이 상위 유형 레벨에 있습니다.
- 대부분의 관계가 상위 유형 레벨에 있습니다.
- 여러 가지 하위 유형에 필요한 기능이 중복됩니다.

하위 유형 매핑

- 여러 가지 유형의 데이터에 대한 액세스 경로는 동일합니다.
- 업무 규칙은 하위 유형에 대해 전역적으로 동일합니다.
- 하위 유형당 인스턴스 수는 많이 다르지 않습니다. 예를 들면, 한 유형에 다른 유형의 인스턴스 수의 1000배가 넘는 인스턴스가 있을 수 없습니다.
- 한 하위 유형의 인스터스는 다른 하위 유형의 인스턴스가 될 수 있습니다. 예를 들면, OPEN ORDER 및 PROCESSED ORDER 하위 유형이 있고 각각 해당 속성을 갖고 있는 ORDER 엔티티의 경우를 고려해 보십시오. OPEN ORDER는 결국 PROCESSED ORDER가 됩니다.

추가 객체

일반적으로 모든 하위 유형에 대해 하나의 뷰를 생성하며, 이 뷰는 특정 하위 유형에 속하는 열만 표시합니다. 행은 식별자 열을 기준으로 정확히 선택됩니다. 이러한 뷰는 삽입 및 갱신 을 비롯한 모든 데이터 작업에 사용됩니다. 모든 응용 프로그램은 성능 저하 없이 이 뷰를 기 반으로 할 수 있습니다.

상위 유형 테이블과 하위 유형 뷰는 단순하고 적절한 구현이므로 가장 먼저 고려해야 합니다. K 및 L에 기반한 테이블의 결과

K를 기반으로 한 테이블의 외래 키가 간단합니다.

L을 기반으로 한 테이블의 외래 키는 훨씬 복잡합니다. 상위 유형 구현은 외래 키가 제한된 R 집합이 아닌 사용 가능한 P를 참조함을 의미합니다. 추가 제약 조건으로 이를 확인해야 합니다



하위 유형 구현 Ρ **QS (Q)** # Id pk ld q_a_fk * Xxx Xxx Yyy 0 Q A_id fk # Id о Үуу В RS(R) # Id fk₁=*r_a_fk* pk ld Xxx Zzz fk₂=r_b_fk fk₁ A id fk B_id ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

하위 유형 구현

이 하위 유형 테이블 구현(두 테이블 구현이라고도 함)은 Q 및 R과 같은 두 개의 하위 유형만 있다고 가정하여 각 하위 유형에 대해 하나의 테이블을 생성합니다.

규칙

- 1. 테이블:
 - 첫번째 레벨의 하위 유형마다 하나의 테이블이 있습니다.
- 2. 열:
 - 각 테이블은 원래 선택 가능성이 있는 상위 유형의 모든 속성에 대한 하나의 열을 가져옵니다.
 - 각 테이블은 또한 원래 선택 가능성이 있는 하위 유형에 속하는 각 속성에 대한 하나의 열을 가져옵니다.
- 3. 식별자:
 - 상위 유형 레벨의 기본 고유 식별자는 각 테이블에 대해 하나의 기본 키를 생성합니다. 또는 하위 유형에 이미 자체 UID가 있을 경우 이 식별자는 기본 키에 대한 기준으로 사용됩니다.
 - 상위 유형의 보조 식별자는 각 테이블 내에서 고유 키가 됩니다.

하위 유형 구현

- 4. 관계:
 - 모든 테이블은 원래 선택 가능성이 있는 상위 유형 레벨의 관계에 대한 외래 키를 가져옵니다.
 - 하위 유형 레벨의 관계의 경우 외래 키는 매핑된 테이블에 구현됩니다. 원래 선택 가능성은 유지됩니다.

5. 무결성 제약 조건:

- 특정 추가 검사는 필요하지 않습니다. Id 값이 모든 하위 유형에 고유해야 할 경우 에만 다른 주의가 필요합니다.

하위 유형 구현을 고려할 시기

이 구현은 상위 유형을 수평으로 분할하는 것으로도 볼 수 있습니다. 이 구현은 다음과 같은 경우 적합합니다.

- 결과 테이블이 다른 데이터베이스에 상주합니다(분산). 이러한 상황은 정보의 특정 부분에서 여러 다른 업무 위치가 유일한 관심 대상인 경우에 발생합니다.
- 하위 유형에 대한 일반 액세스 경로가 다릅니다.
- 하위 유형에 공통점이 전혀 없습니다. 이러한 상황은 속성이 상위 유형 레벨에 거의 없고 하위 유형 레벨에 많은 경우 발생할 수 있습니다. Electronic Mail 모델에서 예를 찾을수 있습니다. ADDRESS 엔티티의 MAIL LIST와 ALIAS 하위 유형이 바로 그러한 예입니다. 이러한 하위 유형은 메시지의 수신자로 사용될 수 있다는 사실을 공유할 뿐, 다른속성들은 완전히 다릅니다.
- 대부분의 관계가 하위 유형 레벨에 있습니다. 이것은 두 테이블이 다른 데이터베이스에 서 구현될 경우 해당되며, 상위 유형에 대한 외래 키 무결성 제약 조건은 모든 경우에 대해 확인하지 않았을 수 있습니다.
- 업무 기능과 업무 규칙이 하위 유형마다 완전히 다릅니다.
- 테이블 사용 방법이 다릅니다. 예를 들면, 한 테이블이 질의되는 동안 다른 테이블이 갱신될 수 있습니다. 테이블이 하나인 솔루션은 성능 문제를 일으킬 수 있습니다.
- 한 하위 유형의 인스턴스 수가 다른 유형의 인스턴스 수에 비해 매우 적습니다.

추가 객체

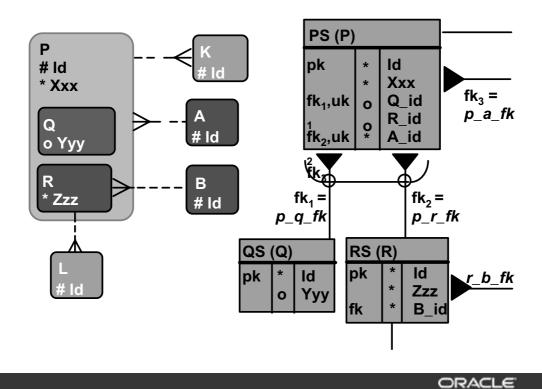
일반적으로 상위 유형의 모든 열을 보여주는 상위 유형과 여러 하위 유형을 나타내는 추가 뷰를 생성합니다. View Select 문은 union 연산자를 사용해야 합니다. 뷰는 질의에만 사용할 수있고 데이터 조작에는 사용할 수 없습니다.

K 및 L에 기반한 테이블의 결과

L을 기반으로 한 테이블의 외래 키는 간단하며 R을 기반으로 한 테이블을 참조해야 합니다. K를 기반으로 한 테이블의 외래 키는 현재 훨씬 복잡합니다. 이러한 키는 Q와 R을 기준으로 한 각 테이블에 하나씩, 두 개의 선택적 외래 키로 구현되어야 합니다.

두 외래 키가 동시에 같은 값을 갖지 않도록 하려면 추가 검사가 필요합니다. 추가 검사는 일반 호 검사와 동일합니다.

상위 유형 및 하위 유형(호) 구현



Copyright © Oracle Corporation, 2002. All rights reserved.

상위 유형과 하위 유형 "호" 구현

이 옵션은 PS측에서 배타적인 호의 외래 키에 연결된 모든 엔티티에 대해 하나의 테이블을 생성합니다. 이 옵션은 마치 하위 유형이 각각 상위 유형에 대해 is subtype of / is supertype of 관계를 갖고 있는 독립형 엔티티로 모델링된 것처럼 모델을 구현한 것입니다. 이러한 관계는 호구현에서 볼 수 있습니다. 따라서 이 구현을 호구현이라고도 합니다. 호와 하위 유형 비교에 대한 자세한 내용은 제약 조건에 대한 장을 참조하십시오.

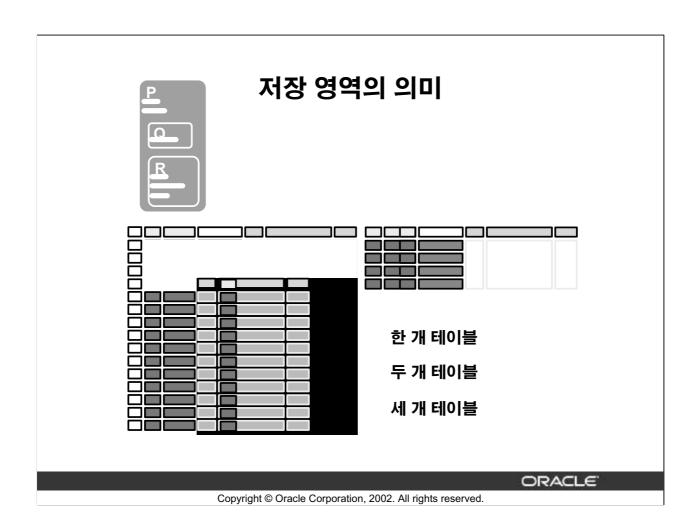
상위 유형과 하위 유형 "호" 구현

규칙

- 1. 테이블:
 - 상위 유형뿐만 아니라 하위 유형 수만큼 테이블이 생성됩니다.
- 2. 열:
 - 각 테이블은 원래 선택 가능성이 있고 기준으로 하는 엔티티의 모든 속성에 대한 하나의 열을 가져옵니다.
- 3. 식별자:
 - 상위 유형 레벨의 기본 UID는 각 테이블에 대한 하나의 기본 키를 생성합니다.
 - 다른 모든 고유 식별자는 해당 테이블의 고유 키로 변환됩니다.
- 4. 관계:
 - 모든 테이블은 원래 선택 가능성이 있는 엔티티 레벨의 관련 관계에 대한 외래 키를 가져옵니다.
- 5. 무결성 제약 조건:
 - 두 개의 추가 열이 상위 유형에 따라 테이블에 생성됩니다. 이들 열은 하위 유형을 구현하는 테이블을 참조하는 외래 키 열입니다. 외래 키가 호에 있으므로 이 열은 선택 사항입니다. 또한 외래 키 열은 실제로 필수 1:1 관계를 구현하므로 고유 키의 일부입니다.
 - 호를 구현하려면 추가 검색 제약 조건이 필요합니다.

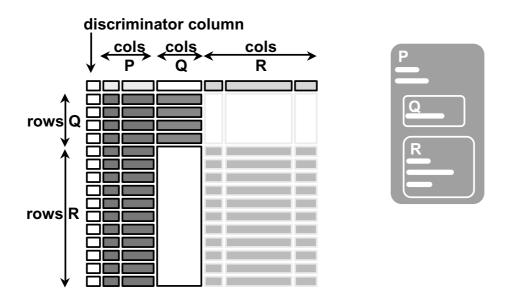
상위 유형과 하위 유형 구현을 고려할 시기

- 이 솔루션은 이중 분할 방식을 수행합니다. 이 솔루션은 자주 사용되지 않지만 다음과 같은 경우에 적합합니다.
- 결과 테이블이 다른 데이터베이스에 상주합니다(분산). 이러한 상황은 정보의 특정 부분에서 여러 다른 업무 위치가 유일한 관심 대상인 경우에 발생합니다.
- 하위 유형에 공통점이 전혀 없고 각 테이블은 독립적으로 사용할 수 있는 정보를 나타냅니다. 예를 들면, PS 테이블은 모든 전역 정보를 제공하고 QS와 RS는 모두 특정 정보를 제공하며 전역 및 특정 정보 조합이 거의 필요하지 않는 경우입니다.
- 업무 규칙이 유형마다 전혀 다릅니다.
- 테이블 사용 및 액세스 방법이 다릅니다.
- 다른 업무 영역의 사용자들이 동시에 같은 행에서 작업해야 하지만, 행의 다른 부분에서 작업을 해야 하는 경우입니다. 이것은 잠금 문제와 성능 문제를 유발시킬 수 있습니다.
- **추가 객체**: 추가 객체를 거의 사용하지 않아도 상위 유형과 여러 하위 유형 전체를 나타내는 추가 보기의 생성을 고려할 수 있습니다.
- K 및 L에 기반한 테이블의 결과: 두 외래 키는 추가 검사 없이도 직접 구현할 수 있습니다.



Oracle Internal & OAI Use Only

저장 영역의 의미 상위 유형 구현



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

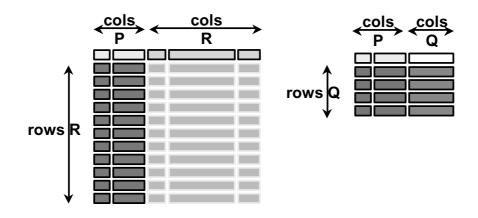
저장 영역의 의미

그림은 한 개, 두 개, 세 개 테이블 구현의 차이를 보여줍니다. 모순처럼 들릴 수 있지만 대부분의 데이터베이스 시스템에서 빈 열 값은 데이터베이스 공간의 일부 바이트를 사용합니다. Oracle에서 빈 열이 테이블 끝에 있고 데이터 유형이 가변 크기이면 빈 열 값은 아주 작습니다.

상위 유형 구현

두 유형의 모든 행은 하나의 테이블에 있습니다. Q 행의 빈 공간은 R 열에 있고 Q 열의 빈 공간은 R 행에 있습니다.

저장 영역의 의미 하위 유형 구현

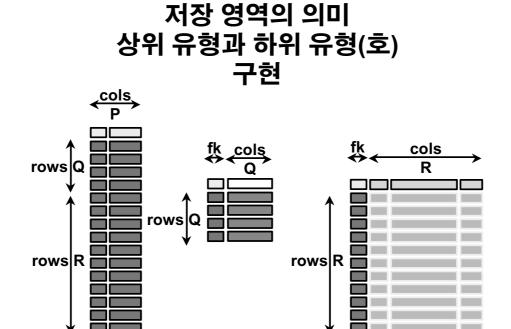


Copyright © Oracle Corporation, 2002. All rights reserved.

저장 영역의 의미

하위 유형 구현

하위 유형 구현
두 개 테이블 구현에서 한 개 테이블 구현의 "빈 공간"은 없어졌습니다. 이것이 테이블의 가로분할입니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

저장 영역의 의미

호 구현

이 세 개의 테이블 구현에서 한 테이블은 P 열 부분으로만 세로로 분할됩니다. 나머지 부분은 Q 및 R 열과 행으로 가로로 분할됩니다. 모든 조각을 서로 연결하려면 P의 추가 외래 키 열이나 Q와 R의 외래 키 열이 있어야 합니다.

요약

- 관계 개념
- 이름 지정 규칙 규약
- 기본 매핑
- 복합 매핑

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

요약

관계형 데이터베이스는 데이터베이스가 기준으로 하는 관계 이론을 구현합니다. 연관 이름 지정 규칙은 많은 오류와 문제를 예방할 수 있고 데이터베이스 스키마 구조에 대한 이해를 도와 줍니다.

ER 모델에서 엔티티 및 관계와 같은 기본 요소를 매핑하는 방법을 살펴보았습니다. 이런 작업은 매우 간단하게 처리할 수 있습니다. 변환 방법을 결정해야 하는 복잡한 구조도 있습니다. 일부 ER 모델 요소는 검색 제약 조건이나 데이터베이스 트리거를 코딩해야만 구현할 수 있습니다. 이러한 방법은 Oracle에만 해당되며 관계형 데이터베이스의 ISO 표준의 일부는 아닙니다.

Practice

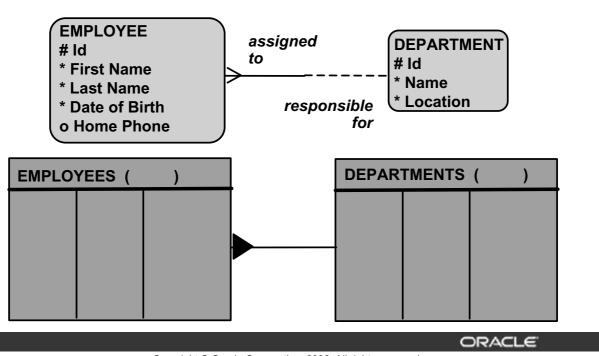
- 기본 엔티티, 속성 및 관계 매핑
- 상위 유형 매핑
- 품질 검사
 하위 유형 구현
- 품질 검사 상위 유형과 하위 유형(호) 구현
- 기본 키와 열 매핑

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



연습: 기본 엔티티, 속성 및 관계 매핑



Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-1: 기본 엔티티, 속성 및 관계 매핑

목표

이 연습에서는 개념적 모델을 데이터베이스의 기본(first cut) 논리적 매핑으로 생성합니다.

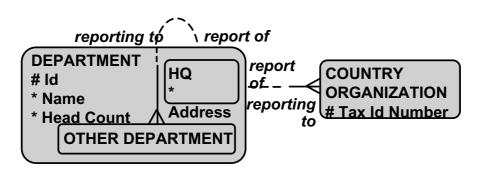
시나리오

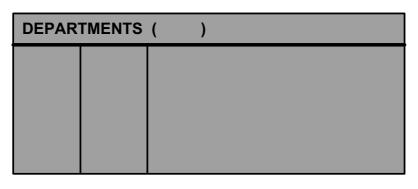
다음은 DEPARTMENT 및 EMPLOYEE 엔티티를 보여주는 간단한 Moonlight ER 모델의 일부입니다. 다음 도표의 엔티티, 속성, 관계, 선택 가능성 및 키를 매핑합니다.

과제

- 엔티티를 테이블에 매핑하고 모든 속성을 열에 매핑합니다.
- 관계를 외래 키 열에 매핑하고 (fk)로 표시합니다.
- 모든 선택 가능성 태그를 Null 값이 아니도록 매핑합니다(*).
- UID 태그를 기본 키(pk)에 매핑합니다.
- 테이블 도표에서 이 구현 다음에 생성해야 하는 모든 요소의 이름을 지정하십시오. 이 단원에서 설명하는 대로 이름 지정 규칙을 사용하거나 사용자 자체 규칙을 사용하십시오. 열 및 외래 키 제약 조건에 적절한 이름을 지정하십시오.

연습: 상위 유형 매핑





ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-2: 상위 유형 매핑

목표

이 연습에서는 복잡한 매핑을 생성하고 변환 프로세스에 대한 이해를 점검합니다.

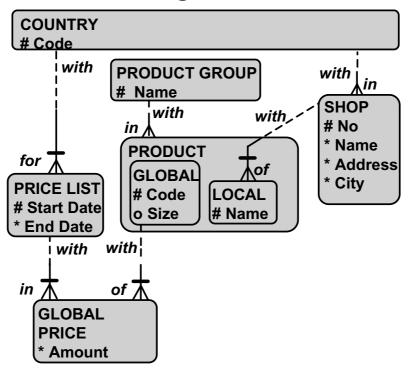
시나리오

다음은 DEPARTMENT 엔티티를 보여주는 Moonlight ER 모델의 일부입니다. 분석가 중 한 명이 DEPARTMENT 엔티티와 해당 하위 유형을 단일 테이블로 구현하기로 하였습니다.

과제

- 이렇게 선택한 이유는 무엇입니까?
- 테이블 도표에서 이 상위 유형을 구현한 후에 생성해야 하는 모든 요소의 이름을 지정하십시오. 이 단원에서 설명하는 대로 이름 지정 규칙을 사용하거나 사용자 자체 규칙을 사용하십시오. 열 및 외래 키 제약 조건에 적절한 이름을 지정하고 검사 제약 조건이 있는 경우 식별하십시오.

Moonlight ER 모델 일부



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-3: 품질 검사 하위 유형 구현

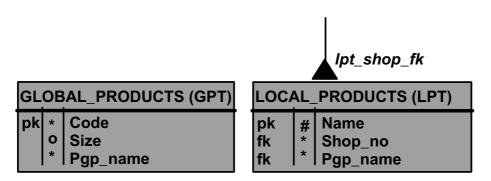
목표

- FZ777 W OAI WS 이 연습에서는 이 단원에서 설명한 이름 지정 규칙을 사용하는 누군가가 생성한 테이블 매핑 에 대해 품질 검사를 수행합니다.

시나리오

다음은 Moonlight ER 모델의 일부입니다.

연습: 품질 검사 하위 유형 구현



ORACLE

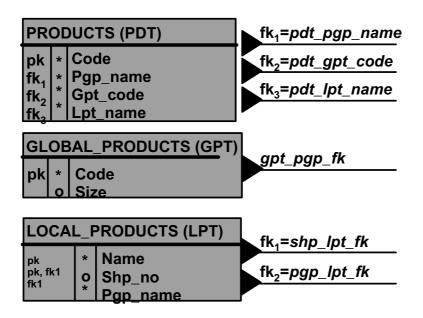
Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-3: 품질 검사 하위 유형 구현

고 기 기 규정 구현 과체 PRODUCT 엔티티의 제안된 하위 유형 구현에 대해 품질 검사을 수행하십시오.

Data Modeling and Relational Database Design 7-42

연습: 품질 검사 호 구현



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-4: 품질 검사 호 구현

목표

이 연습의 목적은 이 단원에서 설명한 이름 지정 규칙을 사용하는 누군가가 생성한 테이블 매핑에 대해 품질 검사를 수행하는 것입니다.

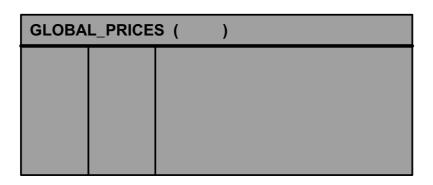
시나리오

이 연습은 이전 연습과 동일한 ER 도표를 기반으로 합니다.

과제

PRODUCT 엔티티 및 해당 하위 유형의 제안된 상위 유형 및 하위 유형 구현에 대해 품질 검사를 수행하십시오. 또한 선택된 이름을 확인하십시오.

연습: 기본 키와 열 매핑



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-5: 기본 키와 열 매핑

, ㄴ /] 와 열의 복잡한 매핑을 수행하는 것입니다. **' | 나리오**이 연습은 이전 연습에서 사용한 동일한 모델을 기반으로 합니다. **과제**GLOBAL PRICE 엔티티를 변환하여 메드이름을 지정합니다. oracle internal GLOBAL PRICE 엔티티를 변환하여 만들어지는 기본 키 열 및 이름을 식별하십시오. 간단한



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



개요

- 비정규화
- 이점
- 비정규화 유형

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

소개

○ 단 국식
이 단원에서는 예제와 함께 가장 일반적인 비정규화 유형을 보여줍니다.
목표
이 단원을 마치면 다음을 수행할 수 있습니다.
● 비정규화를 정의하고 이점을 설명합니다.
● 비정규화가 적합한 여러 환경은 구브러 - **

- - 는 하고 설팅

Data Modeling and Relational Database Design 8-2

비정규화 개요

비정규화

- "정규화된" 모델로 시작합니다.
- 설계에 "중복"을 추가합니다.
- 설계의 "무결성"을 줄입니다.
- 보정을 위해 응용 프로그램 코드가 추가됩니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

비정규화를 사용하는 이유 및 시기

비정규화 정의

비정규화는 데이터베이스에 중복을 체계적으로 추가하는 프로세스에 도움을 주어 인덱스화와 같은 다른 가능성이 실패해도 성능을 향상시킵니다. 인덱스화에 대한 자세한 내용은 설계고려 사항 단원을 참조하십시오. 비정규화는 특정 유형의 데이터 액세스를 비약적으로 향상시킬 수 있지만 성공이 보장되지 않으며 항상 비용이 듭니다. 데이터 모델은 강력하지 않고항상 DML 속도를 저하시킵니다. 또한 처리가 복잡하고 데이터 무결성 문제의 가능성이 있으며, 항상 비정규화된 데이터를 관리할 추가 프로그래밍이 필요합니다.

비정규화에 대한 힌트

- 항상 완전히 정규화된 개념적인 데이터 모델을 생성합니다.
- 비정규화는 성능 향상을 위한 마지막 옵션으로 고려하십시오. 비정규화가 필요하다고 가정하지 마십시오.
- 비정규화는 데이터베이스 설계 중에 수행해야 합니다.
- 성능 목표가 만족되면 더 이상 비정규화를 구현하지 마십시오.
- 모든 비정규화를 완전히 문서화하여 테이블에 수행한 작업과 비정규화를 보정하기 위해 추가된 응용 프로그램 코드를 설명합니다.

비정규화 기법

- 파생 가능 값 저장
- 테이블 사전 조인
- 하드 코딩된 값
- 마스터로 디테일 유지
- 마스터에 단일 디테일 반복
- 단락 키

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

비정규화 기법과 문제

다음 페이지에서 일반적으로 사용되는 여러 가지 비정규화 기법을 설명합니다. 모든 유형의 al & OAI USE 비정규화 유형에 대해 적합한 사용 시기와 장단점이 설명되어 있습니다. 여기서 다루는 항목은 다음과 같습니다.

- 파생 가능 값 저장
- 테이블 사전 조인
- 하드 코딩된 값
- 마스터로 디테일 유지
- 마스터에 단일 디테일 반복
- 단락 키

가장 일반적인 예제는 다음과 같습니다

- 파생 가능한 종료 날짜 열
- 파생 가능한 현재 지시자 열
- 계층 레벨 지시자

파생 가능 값 저장

이전

A

pk * Id

* X

B

pk,fk * A_id

Sequence_No
Quantity

파생 가능한 데이터를 저장할 열을

파생 가능한 네이터들 서상할 열을 외래 키의 "참조" 끝에 추가합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

파생 가능 값 저장

질의 중에 계산이 자주 실행되면 계산 결과를 저장하는 것이 좋습니다. 계산에 디테일 레코드가 포함되면 파생된 계산을 마스터 테이블에 저장합니다. 해당 DML이 디테일 레코드에 대해 실행될 때마다 값을 재계산하려면 응용 프로그램 코드를 작성하십시오. 파생 가능 값을 저장하는 모든 상황에서 비정규화된 값을 직접 갱신할 수 없는지 확인합니다. 이러한 값은 항상시스템에 의해 재계산됩니다.

사용하면 적절한 경우:

- 소스 값이 여러 레코드나 테이블에 있을 경우
- 파생 가능 값이 자주 필요하고 소스 값이 필요하지 않은 경우
- 소스 값이 자주 변경되지 않을 경우

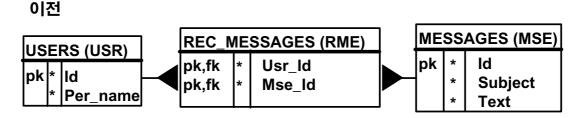
장점:

- 파생 가능 값이 필요할 때마다 소스 값을 조회하지 않아도 됩니다.
- 질의 또는 보고 중에 계산을 수행하지 않아도 됩니다.

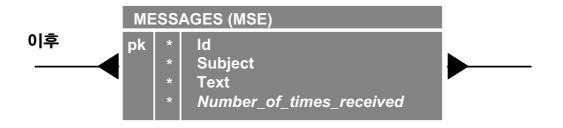
단점:

- 소스 데이터에 대한 DML은 파생 가능 데이터의 재계산이나 조정이 필요합니다.
- 데이터를 복제하면 데이터가 일치하지 않을 가능성이 있습니다.

파생 가능 값을 저장하는 전자 메일 예제



파생 가능한 열을 외래 키의 '참조' 끝에 저장합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

파생 가능한 값을 저장하는 전자 메일 예제

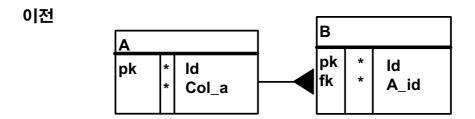
Oracle

메시지가 수신자에게 전달되면 사용자는 RECEIVED_MESSAGES에 기록되는 해당 메시지에 대한 포인터만 받습니다. 그 이유는 한 메시지가 수백 명의 수신자에게 전송될 때 메일 시스템이 동일한 수백 개의 메시지 복사본을 저장하지 않도록 하기 위한 것입니다.

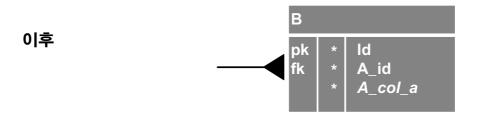
그러면, 누군가가 자신의 계정에서 메시지를 삭제했을 때 RECEIVED_MESSAGES 테이블의 항목만 제거됩니다. 특정 메시지에 대한 모든 RECEIVED_MESSAGE 항목이 삭제되어야 실 제 메시지도 삭제됩니다.

특정 메시지를 보관하는 RECEIVED_MESSAGES의 총 수를 추적하기 위해 비정규화된 열을 MESSAGES 테이블에 추가하는 것을 고려해볼 수 있습니다. 그러면 사용자가 RECEIVED_MESSAGES에서 행을 삭제할 때마다 즉, 사용자가 메시지에 대한 포인터를 삭제할 때마다 Number_of_times_received 열이 감소될 수 있습니다. 비정규화된 열의 값이 0이면 또한 MESSAGES 테이블에서 메시지가 삭제될 수 있습니다.

테이블 사전 조인



외래 키를 사용하여 테이블에 non key 열을 추가합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

테이블 사전 조인

기본 키의 실제 값과 외래 키에 업무 의미가 없으면 키가 아닌 열을 테이블에 포함하여 테이블을 사전 조인할 수 있습니다. 업무 의미가 있는 키가 아닌 열을 포함하여 테이블 조인을 방지하면 특정 질의 속도를 빠르게 할 수 있습니다.

"마스터" 열 값이 참조된 레코드에서 변경될 때마다 비정규화된 열을 갱신하는 응용 프로그램 코드를 포함해야 합니다.

사용하면 적절한 경우:

- 여러 테이블에 대한 질의가 자주 필요한 경우
- 오래된 데이터를 허용하는 경우

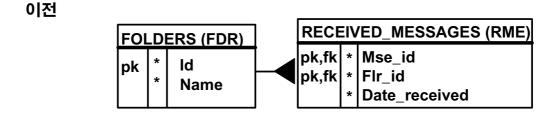
장점

- 시간이 많이 걸리는 조인을 방지할 수 있습니다.
- 오래된 데이터를 허용할 수 있으면 갱신이 연기될 수 있습니다.

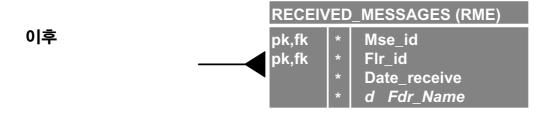
단점

- 비정규화되지 않은 원래 열을 갱신하려면 추가 DML이 있어야 합니다.
- 추가 열과 큰 인덱스를 사용하려면 더 많은 작업 공간과 디스크 공간이 필요합니다.

테이블 사전 조인의 전자 메일 예제



자주 질의되는 모든 열로 테이블을 생성합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

테이블 사전 조인 예제

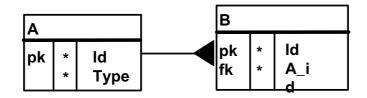
받은 메시지가 파일로 저장된 폴더의 이름을 사용하여 사용자가 RECEIVED_MESSAGES를 자주 질의해야 한다고 가정합시다. 이 경우 폴더 이름을 RECEIVED_MESSAGES 테이블에서 사용할 수 있으면 시간을 절약할 수 있습니다.

이제 사용자가 특정 폴더에서 모든 메시지를 찾아야 할 경우 RECEIVED_MESSAGES만 질의 하면 됩니다.

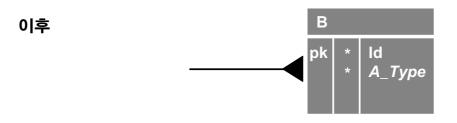
단점은 아주 큰 테이블에 열을 추가하려면 추가 저장 공간이 필요하다는 점입니다.

하드 코딩된 값

이전



외래 키를 제거하고 응용 프로그램에 허용 가능한 값과 검증을 하드 코딩합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

하드 코딩된 값

참조 테이블에 상수로 남아 있는 레코드가 있으면 그러한 값을 응용 프로그램 코드로 하드 코딩하는 것을 고려할 수 있습니다. 즉, 참조 값 목록을 검색하기 위해 테이블을 조인하지 않아도 됩니다. 이것은 값이 데이터베이스의 테이블 외부에 보관되는 특수한 비정규화 유형입니다. 예제에서 허용 가능한 참조 값에 대해 값을 검증할 데이터베이스의 B 테이블에 검색 제약조건을 생성하는 것을 고려해야 합니다. 검색 제약조건이 데이터베이스에 상주해 있어도 하드 코딩된 형식이 유지됩니다. A의 새 값이 필요할 때마다 제약조건을 다시 작성해야 합니다.

사용하면 적절한 경우

- 시스템 주기 중에 허용 가능한 값 집합이 정적인 것으로 간주되는 경우
- 가능한 값 집합이 작을 때 즉, 30 이하인 경우

장점

- 조회 테이블 구현을 방지합니다.
- 조회 테이블에 대한 조인을 방지합니다.

단점

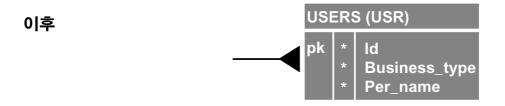
• 조회 값을 변경하면 재코딩하고 재시작해야 합니다.

하드 코딩된 값의 전자 메일 예제

BUSINESS_TYPES

(BTE) pk * Id pk * Bte_id * Per_name

응용 프로그램에 허용 가능한 값과 검증을 하드 코딩합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

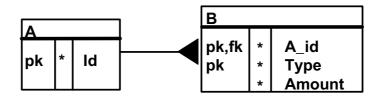
하드 코딩된 값 예제

전자 메일은 사용자의 업무 유형과 같은 사용자에 대한 일부 배경 정보를 알고 싶어 합니다. 따라서 EM은 구분하려는 적합한 모든 BUSINESS_TYPES를 저장할 테이블을 생성했습니다. 이 테이블의 값은 맨 앞에 설정되며 변경될 가능성이 없습니다.

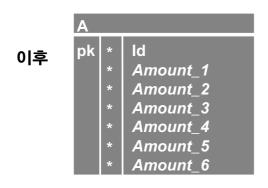
이것은 허용 가능한 값을 하드 코딩할 후보입니다. 데이터베이스의 열에 검색 제약 조건을 두는 것을 고려할 수 있습니다. 이 외에 또는 이 대신 사용자가 EM 서비스에 로그인할 수 있는 화면 응용 프로그램의 필드 검증 검색을 구축할 수 있습니다.

마스터로 디테일 유지

이전



마스터 테이블에 반복되는 디테일 열을 추가합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

마스터로 디테일 유지

마스터당 디테일 레코드 수가 고정된 값(또는 고정된 최대값)이고 일반적으로 모든 디테일 레코드가 마스터로 질의되는 상황일 때 디테일 열을 마스터 테이블에 추가하는 것을 고려할 수 있습니다. 이러한 비정규화 작업은 디테일 테이블의 레코드 수가 적을 때 가장 적합합니다. 이러한 방법으로 질의 중에 조인 수를 줄입니다. 예를 들면, 매일 개인당 하나의 레코드가 있는 계획 시스템을 들 수 있습니다. 이것을 매달 개인당 하나의 레코드 즉, 매일에 대한 열을 포함하는 테이블로 바꿀 수 있습니다.

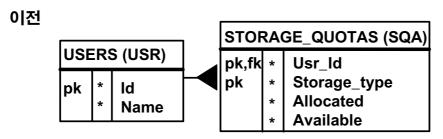
사용하면 적절한 경우

- 모든 마스터의 디테일 레코드 수가 고정되어 있고 정적인 경우
- 디테일 레코드 수와 디테일 열 수를 곱한 값이 작을 때, 즉 30 이하일 때

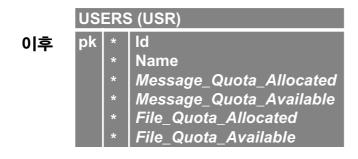
장점

- 조인이 필요하지 않습니다.
- 키가 전달되지 않으므로 공간이 절약됩니다.

마스터로 디테일을 보관하는 전자 메일 예제



마스터 테이블에 반복되는 디테일 열을 추가합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

마스터로 디테일 유지(예제)

단점

- 디테일 값에 대한 SELECT 및 DML(data manipulation languag)이 더 복잡해집니다.
- Amount 열에 대한 검사를 Amount1, Amount2 등에 대해 반복해야 합니다.
- 테이블 이름 A가 더 이상 테이블의 실제 내용과 일치하지 않습니다.

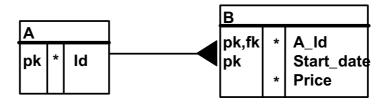
각 전자 메일 사용자에게 메시지와 파일에 대해 각각 하나씩 할당량이 지정된다고 가정합시다. 각 할당량이 다르므로 두 할당량을 개별적으로 추적해야 합니다. 할당량은 자주 바뀌지 않습니다. 단순히 관계를 지정하기 위해 각 사용자에 대한 레코드와 각 할당량 유형에 대한 레코드를 가진 STORAGE_TYPES 테이블과 STORAGE_QUOTAS 테이블을 생성했습니다. 대신 비정규화된 아래 열을 USER 테이블에 생성할 수 있습니다.

- Message Quota Allocated
- Message Quota Available
- File Quota Allocated
- File Quota Available

USERS 테이블의 이름은 비정규화된 테이블의 데이터와 실제로 일치하지 않습니다.

마스터에 최근 디테일 반복

이전



마스터에 열을 추가하여 가장 최근의 디테일을 저장합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

마스터에 단일 디테일 반복

종종 기록 데이터 저장이 필요할 때 대부분의 질의에는 가장 최근 레코드만 필요합니다. 새외래 키 열을 추가하여 해당 마스터에 단일 디테일을 저장할 수 있습니다. 언제든지 새 레코드가 기록 테이블에 추가되면 비정규화된 열을 변경하기 위해 코드를 추가해야 합니다. 마스터 레코드에서 중복된 단일 디테일 값을 관리하려면 추가 코드를 작성해야 합니다.

사용하면 적절한 경우

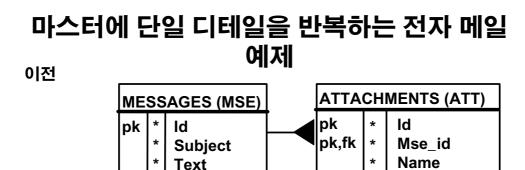
- 마스터당 디테일 레코드에 한 레코드는 "현재"로 간주되고 나머지는 "기록"으로 간주될 수 있는 속성이 있는 경우
- 질의가 이러한 특정 단일 디테일을 자주 필요로 하고 다른 디테일은 가끔씩만 필요로 하는 경우
- 마스터에 주로 하나의 단일 디테일 레코드만 있는 경우

장점

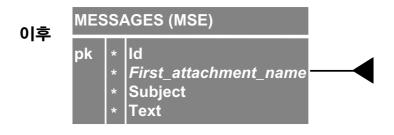
• 특정 단일 디테일만 필요로 하는 질의에 대해 조인이 필요하지 않습니다.

단점

• 디테일 값이 반복되어야 하므로 데이터 불일치 가능성이 있습니다.



마스터에 열을 추가하여 가장 최근의 디테일을 저장합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

마스터에 단일 디테일 반복 예제

언제든지 메시지를 보낼 때 첨부 파일을 포함하여 보낼 수 있습니다. 메시지에 두 개 이상의 첨부 파일이 있을 수 있습니다. 대부분의 메시지에 첨부 파일이 없거나 하나만 있다고 가정합 시다. 테이블 조인을 방지하기 위해 MESSAGES 테이블에 첨부 파일 이름을 저장할 수 있습 니다. 두 개 이상의 첨부 파일을 포함하는 메시지의 경우 첫번째 첨부 파일만 가져올 수 있습 니다. 나머지 첨부 파일은 ATTACHMENTS 테이블에 있습니다.

단락 키 이전 C В pk ld pk ld ld pk fk B_id fk A_id 최하위 디테일에서 최상위 마스터로 연결되는 새 외래 키를 생성합니다. В C 이후 pk ld pk ld ld fk A id fk B id fk A_id ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

단락 키

마스터 디테일의 세 개(또는 그 이상) 레벨을 포함하는 데이터베이스 설계의 경우 그리고 최하위 및 최상위 레벨 레코드만 질의할 필요성이 있는 경우 단락 키 생성을 고려하십시오. 이러한 새로운 외래 키 정의는 최하위 레벨 디테일 레코드를 상위 레벨의 최상위 레코드에 직접 연결합니다. 그 결과 질의가 실행되면 소수의 테이블 조인이 생성됩니다.

사용하면 적절한 경우

• 질의에 상위가 아닌 최상위 및 최하위의 값을 필요로 하는 경우

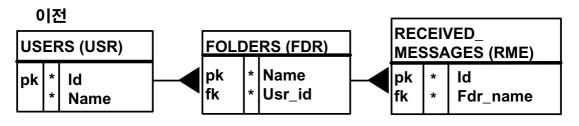
장점

• 질의가 소수의 테이블을 조인합니다.

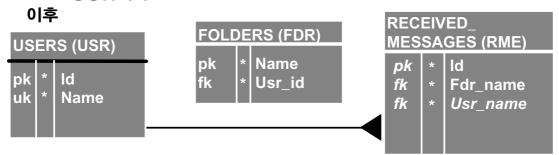
단점

- 추가 외래 키가 필요합니다.
- 비정규화된 열 A_id의 값이 테이블 B와 조인한 후에 찾은 값과 일치하도록 하려면 추가 코드가 있어야 합니다.

단락 키의 전자 메일 예제



최하위 디테일에서 최상위 마스터로 연결되는 새 외래 키를 생성합니다.



ORACLE

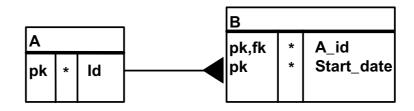
Copyright © Oracle Corporation, 2002. All rights reserved.

단락 키 예제

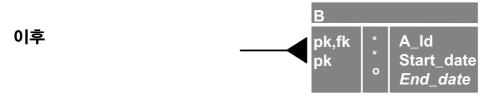
FOLDERS 테이블이 아닌 RECEIVED_MESSAGES 테이블과 USERS 테이블에서 데이터를 요청하는 질의가 제출된다고 가정합시다. USERS 및 FOLDERS 테이블을 조인하지 않도록 하려면 USERS 테이블의 기본 키나 고유 키를 RECEIVED_MESSAGES 테이블에 이전하여 하나적은 테이블 조인으로 또는 테이블 조인 없이 USERS 및 RECEIVED_MESSAGES에 대한 정보를 제공할 수 있습니다.

종료 날짜 열

이전



질의 속도를 빠르게 하려면 between 연산자를 사용할 수 있도록 종료 날짜 열을 추가합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

종료 날짜 열

가장 일반적인 비정규화 결정은 연속하는 기간의 종료 날짜를 저장하는 것입니다. 그러면 한 기간에 대한 종료 날짜를 이전 기간의 시작 날짜에서 구할 수 있습니다.

이렇게 하면 특정 날짜에 대한 디테일 레코드를 찾기 위해 복잡한 하위 질의를 사용할 필요가 없습니다.

사용하면 적절한 경우

• 긴 기록 목록이나 레코드가 있는 테이블에서 질의해야 하고 가장 최신 레코드에 관심이 있는 경우

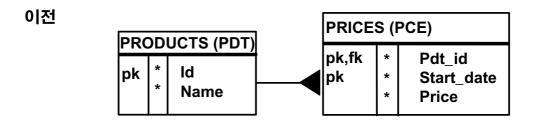
장점

• 날짜 선택 질의에 시간이 많이 걸리는 동기화된 하위 질의 대신 between 연산자를 사용할 수 있습니다.

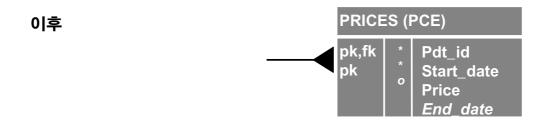
단점

• 종료 날짜 열을 이전 시작 날짜 레코드에서 찾은 값으로 채우려면 추가 코드가 필요합니다.

종료 날짜 열 예제



파생 가능한 추가 열 End date 열을 생성합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

종료 날짜 열 예제

회사에서 제품의 가격 기록을 추적하고자 할 때 PRODUCTS 테이블에 대한 외래 키와 가격 및 해당 시작 날짜에 대한 열을 포함하는 PRICES 테이블을 사용할 수 있습니다. 특정 날짜의 가격을 찾을 때 하위 질의를 사용하지 않으려면 종료 날짜 열을 추가하는 것을 고려할 수 있습니다. 그러면 새 가격이 삽입될 때마다 종료 날짜를 갱신하기 위해 일부 응용 프로그램 코드를 작성해야 합니다.

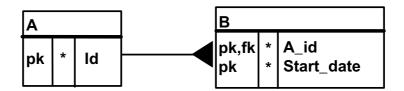
비교:

...WHERE pdt_id = ... AND start_date = (SELECT max(start_date) FROM prices WHERE start_date<= sysdate AND pdt_id = ...)

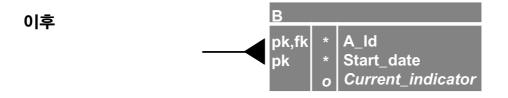
...WHERE pdt_id = ... AND sysdate between start_date and nvl(end_date, sysdate) 첫번째 테이블 구조는 해당 제품의 처음 가격 시작 날짜 이후 제품에 항상 가격이 있다는 것을 전제로 합니다. 이것이 바람직할 수 있지만 많은 업무 상황에서 항상 그런 것은 아닙니다. 또한 기간이 겹치지 않도록 하려면 코드가 필요합니다.

현재 지시자 열

이전



긴 레코드 목록에 가장 최근 레코드를 나타내는 열을 추가합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

현재 지시자 열

이러한 비정규화 유형을 종료 날짜 열 기술과 비슷한 상황에서 사용할 수 있으며, 종료 날짜 에 추가하여 사용할 수도 있습니다. 이것은 아주 일반적인 비정규화 유형입니다.

대부분의 질의가 가장 최근 디테일 레코드를 찾는 것이라고 가정합시다. 이러한 유형의 요구 사항이 있는 경우 현재 활성 레코드를 나타내기 위해 디테일 테이블에 새 열을 추가하는 것을 고려할 수 있습니다.

새 레코드를 삽입할 때마다 해당 열을 갱신하려면 코드를 추가해야 합니다.

사용하면 적절한 경우

• 긴 목록에서 가장 최근 레코드를 검색해야 하는 상황인 경우

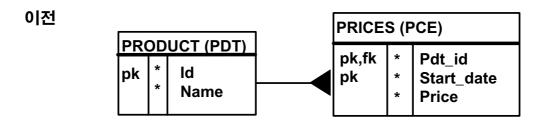
장점

• 질의 또는 하위 질의가 그다지 복잡하지 않습니다.

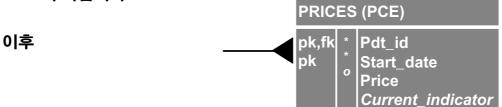
단점

- 관리하기 위한 추가 열과 응용 프로그램 코드가 필요합니다.
- "현재"의 개념은 데이터를 미리 조정할 수 없게 합니다.

현재 지시자 열 예제



긴 레코드 목록에 가장 최근 레코드를 나타내는 열을 추가합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

현재 지시자 열 예제

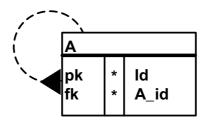
첫번째 테이블 구조에서 제품의 현재 가격이 필요하면 다음을 사용하여 PRICES 테이블을 질 의해야 합니다.

oracle internal & OAII ...WHERE pdt id = ... AND start date = (SELECT max(start_date) FROM prices WHERE start date <= sysdate AND pdt id = ...)

두번째 상황에서 질의는 다음과 같습니다.

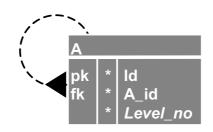
계층 레벨 지시자

이전



레코드의 계층 레벨을 나타내는 열을 생성합니다.

이후



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

계층 레벨 지시자

특정 계층에 포함될 수 있는 레벨 수에 대한 업무 제한이 있다고 가정합시다. 또는 여러 상황에서 한 계층의 같은 레벨에 있는 레코드를 알고 있어야 한다고 가정합시다. 이러한 두 가지 상황에서는 계층을 순회하는 데 connect-by 절을 사용해야 합니다. 이러한 유형의 절은 성능을 저하시킬 수 있습니다. SQL의 connect-by 절 대신 계층에 레코드 레벨을 나타내는 열을 추가한 다음 해당 값을 사용할 수 있습니다.

사용하면 적절한 경우

- 한 계층 내의 레벨 수에 제한이 있고 그 제한에 도달했는지 여부를 확인할 때 connect-by 검색을 사용하지 않으려는 경우
- 계층의 같은 레벨에 있는 레코드를 찾으려는 경우
- 특정 업무상 이유로 레벨 값이 자주 사용되는 경우

장점

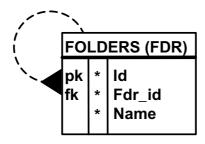
• 질의 코드에 connect-by 절을 사용할 필요가 없습니다.

단점

• 외래 키가 갱신될 때마다 레벨 지시자를 다시 계산해야 하고 변경 작업을 연쇄적으로 수행해야 합니다.

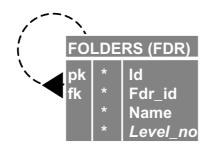
계층 레벨 지시자 예제

이전



레코드의 계층 레벨을 나타내는 열을 생성합니다.





ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

계층 레벨 지시자 예제

저장 영역의 제한으로 인해 중첩된 폴더 수에도 제한이 있다고 가정합시다. 사용자가 기존 폴더 인스턴스 내에 한 폴더의 새 인스턴스를 생성할 때마다 코드는 그 한계에 도달했는지 여부를 판단해야 합니다. 그러면 처리 속도가 느려질 수 있습니다.

폴더의 중첩된 레벨을 나타내는 열을 추가하면, 해당 폴더에 새 폴더를 생성할 때 새 폴더 생성이 가능한지 여부를 즉시 판단할 수 있습니다. 새 폴더 생성이 가능하면 새 폴더의 레벨은 상주하는 폴더보다 한 레벨 증가합니다.

비정규화 요약

비정규화 기법

- 파생 가능한 정보 저장
 - 종료 날짜 열
 - 현재 지시자
 - 계층 레벨 지시자
- 테이블 사전 조인
- 하드 코딩된 값
- 마스터로 디테일 유지
- 마스터에 단일 디테일 반복
- 단락 키

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

비정규화 요약

비정규화는 구조화된 프로세스이므로 간단히 수행해서는 안 됩니다. 모든 비정규화 단계에 추가 응용 프로그램 코드가 필요합니다. 이러한 중복 데이터를 사용할 것인지 반드시 확인하 십시오.

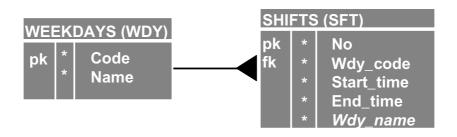
연습

- 비정규화 이름 지정
- 트리거
- 가격 목록 비정규화
- 전역 이름 지정

ORACLE



연습: 비정규화 이름 지정(1/3)



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-1: 비정규화 이름 지정

앞에서 설명한 비정규화 유형을 구분하는 방법을 학습합니다.

과제

se only 다음 테이블 도표의 경우 사용된 비정규화 유형을 판단하고 도표가 사용자가 나열한 비정규 - 시상
- 시선조인

• 하드코딩된 값

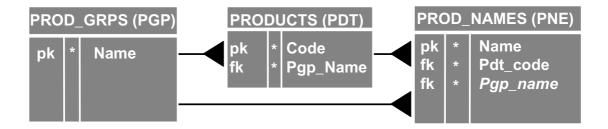
• 마스터로 디테일 유지

• 마스터에 단일 디테일 바 니

• 단락 키 화를 나타내는 이유를 설명합니다.

다음 중 하나를 사용하십시오.

연습: 비정규화 이름 지정(2/3)



ORACLE



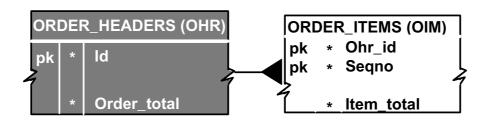
연습: 비정규화 이름 지정(3/3)



ORACLE



연습: 트리거(1/6)



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-2: 트리거

목표

이 연습의 목적은 제안된 비정규화를 처리하는 데 필요한 데이터베이스 트리거를 조사하는 것입니다.

과제

1. ORDER_HEADERS의 비정규화된 Order_total 열을 처리하는 데 필요한 트리거와 이 트리거가 어떤 작업을 수행하는지 확인하십시오.

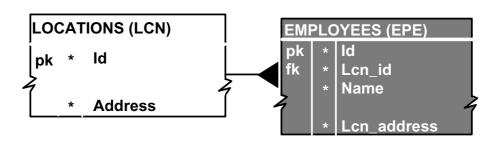
연습: 트리거(2/6)

Table	Trg Type	Column	Needed?	What should it do?
OHR	Insert			
	Delete			
	Update	ld		
		Order_total		
OIM	Insert			
	Delete			
	Update	Ohr_id		
		Item_total		

ORACLE



연습: 트리거(3/6)



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-2: 트리거

oracle internal 8. OAI USE 2. EMPLOYEES의 비정규화된 Lcn address 열을 처리하는 데 필요한 트리거와 이 트리거



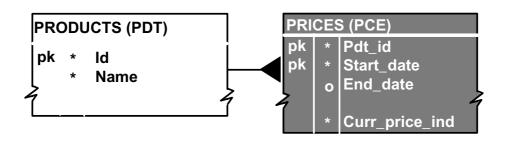
연습: 트리거(4/6)

Table	Trg Type	Column	Needed?	What should it do?
LCN	Insert			
	Delete			
	Update	Address		
		other cols		
EPE	Insert			
	Delete			
	Update	Lcn_id		
		Lcn_address		

ORACLE



연습: 트리거(5/6)

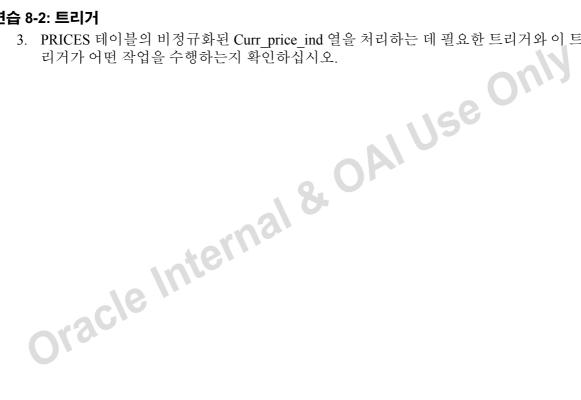


ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-2: 트리거

3. PRICES 테이블의 비정규화된 Curr_price_ind 열을 처리하는 데 필요한 트리거와 이 트



연습: 트리거(6/6)

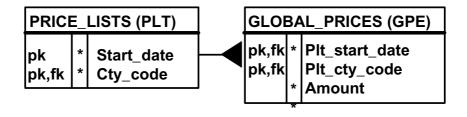
Table	Trg Type	Column	Needed?	What should it do?
PDT	Insert			
	Delete			
PCE	Insert			
	Delete			
	Update	Pdt_id		
		Start_date		
		End_date		
		Curr_price_Ind		

ORACLE



연습: 가격 목록 비정규화

- 합계에 대한 질의 성능을 향상시킵니다.
- 유효 일자 전에 새 가격 목록을 삽입합니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-3: 가격 목록 비정규화

목표

이 연습의 목적은 사용할 수 있는 비정규화 유형과 데이터베이스 무결성을 확인하는 데 필요한 코드를 판단하는 것입니다.

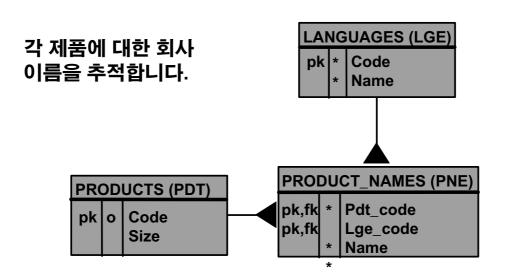
시나리오

일반 사용자가 질의 성능에 대한 불평을 시작했다고 가정합시다. 전역 제품의 가격을 질의 중일 때 이러한 불평이 특히 두드러집니다. GLOBAL_PRICES 테이블에 큰 레코드 목록이 있고 PRICE_LISTS 테이블과 조인해야 하기 때문에 질의하는 데 시간이 오래 걸리는 것입니다. 다른 기법을 사용하여 질의를 최적화한 결과 수용할 만한 응답 시간을 얻지 못했습니다. 따라서 비정규화를 사용하여 이 문제를 해결하기로 결정했습니다. 본사에서는 또 다른 관심사를 갖고 있습니다. 유효 일자 이전에 전역 제품의 새 가격 목록 변경 사항을 지점에 알리려고 합니다. 시작 날짜에 이르러서가 아닌 새 가격 목록 정보가 결정된 때에 그 정보를 입력하려고 합니다. 이 제한을 완화하는 규정을 추가해야 합니다.

과제

구현해야 할 비정규화 유형과 데이터베이스의 무결성 유지를 위해 추가해야 하는 코드를 설명하십시오. 다음 도표는 현재의 테이블 스키마를 보여줍니다. 구현할 비정규화 유형을 결정할 때 위에서 설명한 두 가지 문제점을 고려하십시오.

연습: 전역 이름 지정



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-4: 전역 이름 지정

목표

사용자 요구 사항을 비정규화된 테이블 설계로 변환합니다.

시나리오

se only 본사에서 영어를 본사 언어로 공식화하기로 결정했습니다. 본사에서 모든 전역 제품을 준비 하는 IS 부서에 해당 제품 이름을 영어로 저장하도록 요청했습니다. 반면, 각 국가는 동일한 의미를 가진 자국어로 저장할 수 있어야 합니다.

과제

아래의 설계를 사용하여 테이블 설계를 비정규화하고 이러한 요구 사항을 구현할 수 있도록 Oracle Inte 하는 추가 코드를 설명하십시오.



데이터베이스 설계 고려 사항

ORACLE



개요

- Oracle 고유의 설계 고려 사항
- 데이터 무결성 문제
- 성능 고려 사항
- 저장 영역 문제

ORACLE

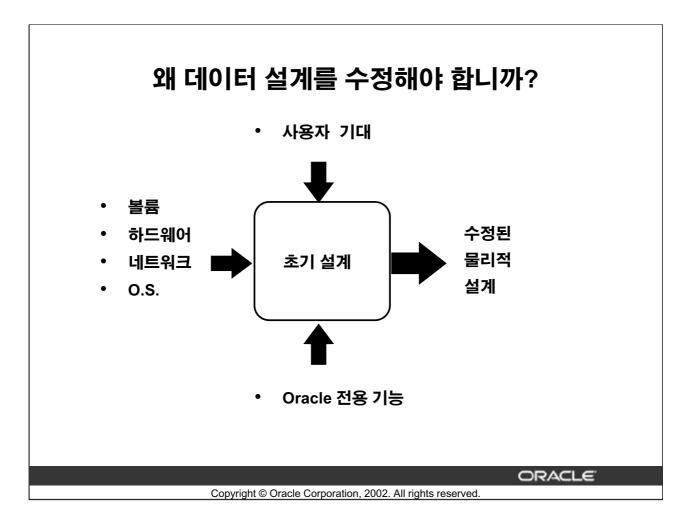
Copyright © Oracle Corporation, 2002. All rights reserved.

소개

이 단원에서는 Oracle RDBMS의 몇 가지 원칙을 설명하고 물리적 설계를 세분화할 때 사용할 OAIUse 수 있는 여러 가지 기법을 보여줍니다.

목표

- 이 단원을 마치면 다음을 수행할 수 있습니다.
 - 열에 사용할 데이터 유형을 설명합니다.
 - 기본 키의 품질을 평가합니다.
 - 필요한 경우 인공 키와 시퀀스를 사용합니다.
 - 참조 무결성에 대한 규칙을 정의합니다.
 - 인덱스 사용에 대해 설명합니다.
 - 분할 및 뷰를 설명합니다.
 - 구식 데이터베이스 기법을 구분합니다.
 - 분산 데이터베이스 원리를 설명합니다.
 - 오라클 데이터베이스 모델을 설명합니다.



데이터베이스 설계 재고

RDBMS마다 고유의 내부 처리 방법이 있습니다. 이 단원에서는 최고 RDBMS 성능을 얻기 위해 Oracle이 제공하는 주요 기능에 대해 설명합니다.

초기 설계에서 올바르게 수정된 물리적 설계를 얻으려면 많은 매개변수를 분석해야 합니다. "똑같이" 수정된 물리적 설계가 아니라 "올바르게" 수정된 물리적 설계임을 유념하십시오. 많은 설계 문제와 마찬가지로 여기에는 절대적인 진실이 없습니다.

여기서 언급된 사항은 가장 중요한 사항입니다.

- 예상 테이블 볼륨, CPU 속도, 메모리 크기, 디스크 수 및 해당 공간과 같은 하드웨어 특징, 구조(클라이언트/서버 또는 3계층), 네트워크 대역폭, 속도 및 운영 체제가 결정적인 요소입니다.
- 사용자 요구 사항은 또 다른 커다란 문제입니다. 응답 시간, GUI 및 모듈 사용 빈도에 따라 사용자 요구 사항은 사용자의 기대에 부응하기 위해 Oracle에서 사용할 수 있는 객체에 영향을 줍니다.
- 현재 사용하고 있는 Oracle 버전에 따라 일부 요소가 있을 수도 있고 없을 수도 있습니다.

Oracle 데이터 유형

영향을 주는 요소:

- 도메인
- 저장 영역 문제
- 성능
- 사용

열에 사용할 데이터 유형 선택:

- 문자
- 숫자
- 날짜
- 대형 객체

ORACLE!

Copyright © Oracle Corporation, 2002. All rights reserved.

Oracle 데이터 유형

테이블이나 클러스터를 생성할 때 각각의 해당 열에 대한 내부 데이터 유형을 지정해야 합니다. 이러한 데이터 유형은 각 열이 포함할 수 있는 값의 일반 도메인을 정의합니다.

- 일부 데이터 유형은 숫자 및 날짜와 같이 포커스가 좁습니다. 일부 데이터 유형은 여러 가지 문자 데이터 유형처럼 일반적인 용도의 데이터 유형입니다.
- 일부 데이터 유형은 가변 길이를 허용하고 일부는 허용하지 않습니다.
- 한 열에 대부분의 행에 대해 아주 작은 바이트를 저장하도록 큰 고정 길이를 선택하면 테이블 크기가 아주 커질 수 있습니다. 이렇게 하면 실제로 한 행에 소수의 바이트만 포함되고 여러 블록에 저장되어 I/O 수가 많아지므로 성능이 저하됩니다.
- 대형 객체 데이터 유형을 검색할 수 없습니다. 이러한 유형은 where 절에 사용할 수 없습니다. 이러한 유형은 다른 열을 검색하여 가져올 수 있습니다.

Oracle 데이터 유형

가장 일반적으로 사용되는 Oracle 데이터 유형

- 1. CHAR(size). 바이트 길이의 고정 길이 문자 데이터입니다. 최대 크기는 2000바이트입니다.
 - 일반 용도: USD, FFR처럼 길이가 세 문자로 고정된 공식적인 국가별 통화 코드에 사용합니다.
- 2. VARCHAR2(size). 최대 바이트 길이의 가변 길이 문자열입니다. 최대 크기는 4000이고 최소 크기는 1입니다. 이 유형은 가장 많이 사용되는 데이터 유형이며 어느 유형을 사용 해야 할지 모를 경우 이 유형을 사용합니다. 이 유형은 기존 Oracle 버전 6 CHAR 데이터 유형을 대체합니다.
 - **일반적인 용도:** 대체 문자를 사용하여 검색할 수 있어야 하는 무제한 길이의 ASCII 텍스트의 개별 ASCII 텍스트 *행*을 저장하는 데 사용합니다.
- 3. NUMBER. 이 데이터 유형은 소수가 있거나 없는 무제한 크기의 숫자 값에 사용합니다. 이 데이터 유형은 계산이나 정렬이 가능한 데이터에 사용합니다. 값에 아무 의미가 없는 전화번호같은 숫자에 사용하지 마십시오.
 - **일반적인 용도:** 금액, 수량, 생성된 고유 키 값에 사용됩니다.
- 4. DATE. 유효한 날짜 범위는 기원전 4712년 1월 1일부터 서기 4712년 12월 31일까지입니다. 날짜 데이터 유형에 시간 구성 요소를 포함할 수도 있습니다. 이 데이터 유형은 날짜, 월 및 연도를 포함하여 전체 날짜를 알고 있을 때만 사용해야 합니다. 시간 구성 요소는일반 날짜 사용에서 종종 00:00(자정)으로 설정됩니다.
 - 일반적인 용도: 전체 날짜가 알려진 임의의 날짜에 사용합니다.
- 5. LONG. 최대 가변 길이가 2GB인 문자 데이터입니다. Oracle8부터 사용되지 않습니다. 이 유형은 대체 문자나 부분 문자열 기능을 사용하여 검색하지 않아도 되는 ASCII 텍스트 파일에 사용되었습니다. 대신 CLOB 데이터 유형을 사용합니다.
 - 일반적인 용도: HTML 페이지의 소스 코드를 저장하는 데 사용합니다.
- 6. LONG RAW. 최대 가변 길이 2GB인 원시 이진 데이터입니다. Oracle8부터 사용되지 않습니다. 데이터베이스가 데이터 해석을 시도하지 않도록 하는 대형 객체 유형에 사용되었습니다. 대신 BLOB 데이터 유형을 사용합니다.
 - 일반적인 용도: 이미지 또는 비디오 클립에 사용합니다.
- 7. CLOB. CLOB(Character large object) 유형입니다. LONG을 대체합니다. 주요 차이점은 LONG은 한 테이블에 하나만 허용된 반면 CLOB 열은 두 개 이상 가질 수 있다는 점입니다. 최대 크기는 4GB입니다.
 - 일반적인 용도: LONG을 참조하십시오.
- 8. BLOB. CLOB(Character large object) 유형입니다. LONG RAW를 대체합니다. 주요 차이 점은 LONGRAW는 한 테이블에 하나만 허용된 반면 BLOB 열은 두 개 이상 가질 수 있다는 점입니다. 최대 크기는 4GB입니다.
 - **일반적인 용도:** LONG RAW를 참조하십시오.
- 9. BFILE. 데이터베이스 외부에 저장된 대형 이진 파일에 대한 위치 지정자를 포함하여, 데이터베이스 서버에 상주하는 외부 LOB에 바이트 스트림 I/O 액세스를 가능하게 합니다.
 - 일반적인 용도: 영화에 사용합니다.

제안된 열 순서

- 기본 키 열
- 고유 키 열
- 외래 키 열
- 필수 열
- 선택 열

대형 객체 열은 항상 끝에 옵니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

열 순서

테이블의 열 순서에 관계없이 모든 테이블 작업이 가능하지만 열 순서에는 의미가 있습니다. .반테 (S) (A) US 열 순서는 특히 데이터 조작 작업 성능에 영향을 줄 수 있습니다. 또한 테이블 크기에 영향을 줄 수도 있습니다.

제안된 최적의 열 순서는 다음과 같습니다.

- 기본키열
- 고유 키 열
- 외래키열
- 나머지 필수 열
- 나머지 선택 열

테이블에 LONG 또는 LONG RAW 열이 있는 경우 그 열이 필수 열이어도 테이블의 마지막 열 로 지정합니다.

이유는 널 열이 테이블 끝에 있어야 하고, 검색 조건에 자주 사용되는 열은 맨 앞에 있어야 하 기 때문입니다. 이것은 저장 영역과 성능상의 이유 때문입니다.

기본 키

CREATE TABLE countries

(code NUMBER(6) NOT NULL

,name VARCHAR2(25) NOT NULL

, currency NUMBER (10,2) NOT NULL

);

ALTER TABLE countries

ADD CONSTRAINT cty_pk PRIMARY KEY

(code);

제약 조건 및 인덱스 이름

ORACLE

only

Copyright © Oracle Corporation, 2002. All rights reserved.

기본 키 및 고유 키

기본 키

일반적으로 모든 테이블에 실행되는 강력한 개념입니다.

- 기본 키는 하나 이상의 열로 구성될 수 있고 각 열은 필수이어야 합니다.
- 기본 키는 선언 제약 조건이므로 이름을 지정할 수 있습니다. 기본 키 제약 조건을 생성할 때 Oracle은 이와 연관된 고유 인덱스를 자동으로 생성합니다.
- 외래 키는 일반적으로 테이블의 기본 키를 참조하지만 고유 키를 참조할 수도 있습니다. 기본 키가 없는 테이블에는 고유 키가 있어야 합니다.

주: Oracle은 기본 키를 갱신할 수 있도록 허용하지만 관계 이론은 갱신하지 않을 것을 권장합니다.

기본 키 및 고유 키

고유 키

고유 키는 몇 가지 이유로 인해 기본 키로 선택되지 않았던 키입니다. 다음과 같은 이유 때문일 수 있습니다.

- 널을 허용합니다. 널은 고유 키 열에서 허용될 수 있습니다.
- 갱신할 수 있습니다. 고유 키 값은 변경될 수 있지만 고유한 상태로 남아 있어야 합니다. 예를 들면, 사원의 집 전화번호나 자동차 번호판을 들 수 있습니다.

테이블마다 두 개 이상의 고유 키가 있을 수 있습니다.

주: 고유 인덱스는 Oracle이 기본 키와 고유 키 값의 고유성을 검사하기 위해 사용하는 추가 구조입니다. 고유 키를 생성하면 고유 인덱스가 자동으로 생성됩니다.



기본 키

적합한 키 선택

- 단순성
- 사용의 편리성
- 성능
- ヨカ
- 무의미
- 안정성



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

기본 키를 선택하는 방법

다음 분석에는 기본 키에 대해 사용할 수 있는 선택 사항이 있습니다. 사용자에게 이 사항을 표시하거나 알릴 필요는 없습니다. 이 사항은 배경에서 수행됩니다.

바람직한 기본 키 속성

단순성:

Oracle8에서는 최대 32개의 열을 구성할 수 있지만 기본 키는 가능한 한 단순해야 합니다. 기본 키열의 데이터 유형은 다양할 수 있습니다. 데이터 분석 시 UID가 종종 구성되며 이는 단순하지 않습니다. 그러한 기본 키를 단순 키로 교체하는 것을 고려해야 합니다.

사용의 편리성:

기본 키는 일반적으로 조인 문에 사용되므로 기본 키는 사용하기가 쉬워야 합니다. 조인 술어에 열이 두 개만 포함된 경우 두 테이블 간에 조인을 생성하는 SQL 문을 작성하기는 훨씬 쉽습니다.

기본 키를 선택하는 방법

성능을 저하시키지 않음:

단일 키를 사용한 조인 작업은 네 개의 키 열을 사용하는 조인보다 성능이 훨씬 향상됩니다.

작은 크기:

기본 키의 크기가 크면 기본 키를 참조하는 외래 키의 크기도 커집니다. 일반적으로 참조하는 테이블에는 참조된 테이블보다 훨씬 많은 행이 들어 있습니다. 기본 키가 너무 크면 불필요한 바이트의 배수가 될 수 있습니다.

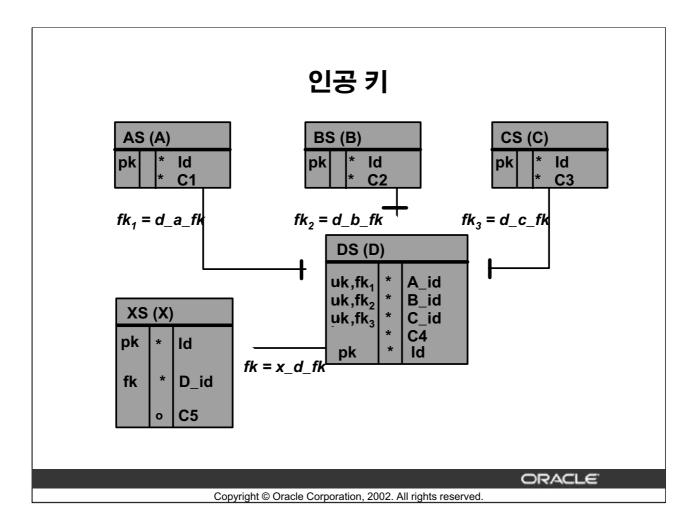
무의미:

예를 들면, 국가명을 기본 키로 사용하도록 선택했을 수 있지만 최근 기록은 국가명이 변경되었음을 보여줍니다. 문자 값이 아닌 숫자 값을 선택하고 숫자를 사용하는 경우 특정 의미가 있는 숫자는 피하십시오.

안정성:

갱신될 가능성이 있는 기본 키를 선택하지 않도록 해야 합니다. 실제로 안정된 상태로 영구적 으로 유지되는 경우는 거의 없습니다.





인공 키

인공 키는 테이블의 기본 키 기능을 하는 레코드에 지정된 무의미한 숫자 값입니다. 인공 키는 복잡한 기본 키 대신 적합한 기능을 제공합니다. 인공 키는 대리 키라고도 합니다.

장점

인공 키에는 구성 키에 비해 다음과 같은 장점이 있습니다.

- 인공 키 열과 인덱스에 필요한 추가 공간은 참조하는 테이블의 외래 키 열에 대해 저장 할 수 있는 공간보다 훨씬 작습니다.
- 조인 조건은 단일 방정식으로 구성됩니다.
- 조인이 더 잘 수행됩니다.
- 사용자가 전혀 볼 수 없는 내부 참조는 관리가 가능합니다. 그러면 모델 UID를 고유 키로 구현할 수 있고 계단식 갱신을 하지 않고도 갱신이 가능합니다.
- 의미가 없으면 기억하기 어렵습니다. 심지어 사용자는 기억해 내려고 하지 않습니다.
- 어떤 사용자는 실제로 인공 키를 선호합니다.



인공 키

다점

인공 키의 단점은 다음과 같습니다.

- 인공 키는 의미가 없기 때문에 조인해야 외래 키 열의 의미를 수집할 수 있습니다.
- 원래 기본 키 열로 구성된 추가 고유 키를 생성할 경우 인덱스에 대한 추가 공간이 필요합니다.
- 의미가 없으면 기억하기 어렵습니다. 사용자가 외래 키 값을 입력하려면 항상 값 목록이나 기타 도움이 필요합니다.
- 일부 사용자는 실제로 인공 키를 선호하지 않습니다.

인공 키에 대한 결정 사항

설계 전

- 부정적: 업무 의미가 없는 요소를 추가하므로 데이터 모델을 손상시킬 수 있습니다.
- 궁정적: 개념적 모델과 기술적 모델을 근접하게 매핑시킴으로써 오해의 가능성을 줄입니다.

설계 후

• 긍정적: 설계 시 실제로 현재 성능 고려 사항을 기준으로 했습니다.

Oracle Designer와 같은 툴을 통해 ER 모델의 초기 매핑 중에 인공 키에 대해 결정할 수 있습니다. 인공 키는 적절한 절충점입니다.





CREATE SEQUENCE sequence_name INCREMENT BY number

START WITH number
MINVALUE number
MAXVALUE number
CACHE number | NOCACHE

CYCLE | NOCYCLE;

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

시퀀스

일부 시퀀스 특징

Olacle

- 시퀀스는 데이터베이스 테이블의 열에 대해 고유 번호 목록을 생성하는 데이터베이스 객체입니다.
- 시퀀스는 고유 번호를 생성하는 가장 빠른 방법을 제공합니다.
- 시퀀스는 인공 키 값으로 사용할 수 있는 고유한 숫자 값을 자동으로 생성하여 응용 프로그램 프로그래밍을 단순하게 합니다.
- 임의의 수의 테이블에 대한 시퀀스 번호를 생성하는 데 시퀀스를 사용할 수 있습니다. 인공 키가 있는 각 테이블에 대해 별도 시퀀스를 생성할 필요는 없지만 일반적으로 그러 한 시퀀스가 생성됩니다.
- 시퀀스를 사용하면 고유한 오름차순 번호나 내림차순 번호가 생성되지만 연속되는 번호가 모두 실제로 사용되지 않을 수도 있습니다.

외래 키 기능

	Delete	Update
Restrict		
Cascade		
Default Nullify		



선언을 통해 Oracle에서 지원됨

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

외래 키

정의상으로 외래 키는 기본 키나 고유 키 값을 참조해야 합니다. 기본 키(또는 고유 키) 값이 11 Use 변경될 경우 발생하는 결과를 고려해야 합니다.

참조 무결성

다음 두 가지 측면을 고려해야 합니다.

- 업무 제약 조건을 지원하기 위해 구현할 규칙
- Oracle이 이러한 규칙에 제공하는 기능

관계 이론은 외래 키에 대한 네 가지 종류의 기능을 설명합니다. 모든 외래 키에 대해 외래 키 의 동작을 결정합니다.

기능은 외래 키가 참조하는 키 값이 변경되었을 때 외래 키가 수행해야 하는 작업을 설명합니 다.

삭제 제한

삭제 제한은 참조 값이 존재하면 기본(또는 고유) 키 값을 삭제할 수 없음을 의미합니다. 이것 은 Oracle에서 지원됩니다. 이것이 가장 일반적으로 사용되는 외래 키 기능입니다.

외래 키

갱신 제한

갱신 제한은 참조 값이 존재하면 기본(또는 고유) 키 값을 갱신할 수 없음을 의미합니다. 이것 은 Oracle에서 지원됩니다. 인공 키는 전혀 갱신되지 않으므로 인공 키의 경우 이 기능이 필요 하지 않습니다.

갱신 제한은 이전 불가능과 다른 개념입니다. 갱신 제한은 참조된 기본 키 값이 갱신되지 않 도록 합니다. 이전 불가능이란 외래 키 열을 갱신할 수 없음을 의미합니다.

계단식 삭제

계단식 삭제는 행을 삭제하면 "계단식"으로 표시된 외래 키를 통해 해당 행을 참조하는 모든 행이 자동으로 삭제됩니다. 계단식 삭제는 Oracle이 지원하는 옵션입니다.

계단식 삭제 중에 삭제할 수 없는 레코드가 있으면 전체 삭제 작업이 실패합니다. 삭제할 레 코드가 삭제 제한 외래 키를 통해 참조된 경우 이러한 상황이 발생합니다.

계단식 삭제는 사용할 때 주의해야 하는 아주 강력한 처리 방식입니다.

계단식 갱신

계단식 갱신은 기본 키 값이 갱신되면 해당 변경 사항이 기본 키 값을 참조하는 모든 외래 키 열에 전달됩니다.

계단식 갱신과 이전 불가능은 종종 함께 수행됩니다.

기본값과 무효화

기본값 및 무효화 옵션은 기본 키 값 삭제 또는 갱신 시 관련 외래 키 값이 기본값을 필요로 하 거나 NULL로 설정됨을 의미합니다.

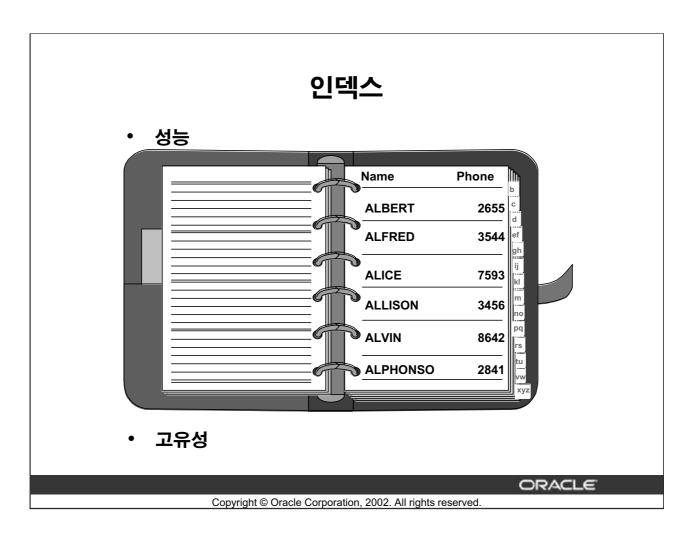
이러한 옵션은 외래 키에 의해 참조되는 테이블에 갱신 데이터베이스 트리거를 생성하여 구 현될 수 있습니다. 무효화 옵션은 외래 키가 선택 사항인 경우에만 적합합니다.

일반적인 용도

일반적으로 많은 외래 키가 삭제 제한으로 정의됩니다. 이렇게 하면 참조된 레코드가 삭제되 지 않으며 사용자가 모든 참조 행을 의식적으로 제거하거나 전송하도록 합니다.

물론 인공 키를 사용하면 인공 키 값을 갱신할 적절한 이유가 전혀 없기 때문에 모든 외래 키 갱신 속성을 "제한"으로 설정할 수 있습니다.





인덱스

인덱스는 종속되는 테이블과 별도로 저장되는 데이터베이스 구조입니다. 관계형 데이터베이 Al Use 스에서 해당 열의 인덱스 유무에 관계 없이 열을 질의할 수 있습니다.

인덱스는 다음과 같은 두 가지 목적을 위해 사용됩니다.

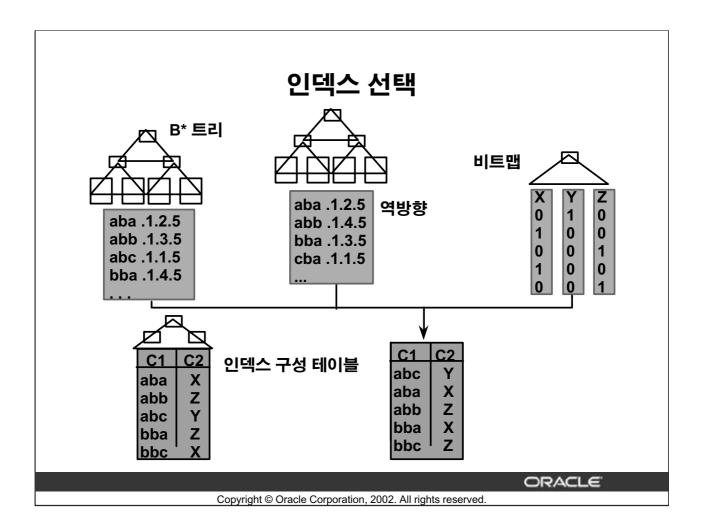
- 질의 속도를 향상시킵니다.
- 필요하면 고유성을 유지합니다.

성능

인덱스를 생성하여 값을 검색하는 빠른 방법을 제공합니다. 그러나 인덱스는 DML 문의 성능 을 저하시킬 수 있습니다. Oracle은 다양한 유형의 인덱스를 제공하므로 용도에 적합한 유형 을 선택해야 합니다.

고유성

고유 인덱스는 인덱스에 포함된 열 집합 내에서 값이 중복되지 않도록 하는 효율적인 구조입 니다. 고유 인덱스는 기본 키 또는 고유 키를 생성하면 자동으로 생성됩니다. 이 경우 인덱스 이름은 키 제약 조건 이름과 같습니다.



인덱스 유형

B* 트리

기존의 인덱스 구조는 명시적으로 달리 지정되지 않으면 B* 트리(또는 트리 균형이라고도 함) 인덱스입니다. 이 인덱스는 온라인 트랜잭션 처리 시스템용으로 설계되었습니다. 효율성이 입증된 이 인덱스를 Oracle은 오랫동안 제공해 왔습니다. 이 인덱스는 삽입, 갱신 및 삭제를 쉽게 지원합니다.

일반적인 용도: 범용

역방향키

Oracle은 기존 B* 트리 구조에 준하여 대부분의 B* 트리 속성을 갖고 있지만 인덱스화된 각열의 바이트를 거꾸로 표시하는 역방향 키 인덱스를 제공합니다.

일반적인 용도: Oracle Parallel Server 환경. 이러한 환경에서 배열을 사용하면 인덱스의 성능이 저하되는 것을 막을 수 있습니다.

인덱스 유형

비트맵

행이 인덱스화된 열의 각 개별 값을 포함하거나 포함하지 않거나 비트맵 인덱스는 이 값에 대 해 저장합니다.

일반적인 용도: 데이터 웨어하우스 환경, 비트맵 인덱스는 임시 질의가 집중적이고 인덱스화 된 열에 대한 고유 값의 수가 높지 않을 때 온라인 분석 처리 시스템에 효율성이 있음을 입증 했습니다.

비트맵 인덱스는 B* 트리 인덱스보다 공간을 적게 필요로 하지만 B* 트리뿐만 아니라 삽입, 갱신 및 삭제를 지원하지 않습니다.

인덱스 구성 테이블

인덱스 구성 테이블은 B* 트리 기법을 사용하여 정렬된 방식으로 저장된 행을 포함하는 테이 블입니다. 이 테이블은 인덱스가 제공하는 속도를 제공하며 별도 인덱스를 필요로 하지 않습 니다. 단, 이 인덱스 구성 테이블에 대한 추가 인덱스를 생성할 수 없다는 한 가지 제한이 있습 니다.

임바적인 용도: 특히 대형 객체를 저장할 때 항상 동일한 경로를 통해 액세스되는 테이블 연결된 인덱스

두 개 이상의 열을 포함하는 인덱스를 생성할 수 있습니다. 이러한 인덱스를 연결된 인덱스라 고 합니다. 열을 지정하는 순서는 Oracle에서 인덱스를 사용할 수 있는 방법에 상당한 영향을 줍니다. 항상 Where 절에 있는 열을 인덱스의 첫번째 열로 설정합니다. 이것을 인덱스의 선행 부분이라고 합니다.

함수 기반 인덱스

Oracle8i 이후 SOL 함수 기반 인덱스를 생성할 수 있습니다.

일반적인 용도: substr 함수를 사용하여 이름의 처음 세 글자에 대한 인덱스를 생성하거나 to char 함수를 사용하여 날짜에 대한 연도 구성 요소를 생성합니다.



어떤 열을 인덱스화해야 합니까?

- 기본 키 열과 고유 키 열(버전 6까지)
- 외래키열
- SELECT 문으로 성능이 현격하게 향상될 때



인덱스화 회피:



- 작은 테이블
- 자주 갱신되는 열

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

인덱스화할 열 선택

일반 B* 트리 인덱스 후보 열

- Al Use Only • 조인의 성능을 향상시키기 위해 조인 조건에 사용되는 열
- 광범위한 값을 포함하는 열
- Where 질의 절에 자주 사용되는 열
- Order By 질의 절에 자주 사용되는 열

비트맵 인덱스 후보 열

• 예를 들면, 지시자 값(Y/N)을 포함하는 열 또는 성별에 대한 열처럼 적은 수의 고유 값을 갖고 있는 열

인덱스화에 적합하지 않은 열

많은 NULL 값을 포함하고 있는 열. 일반적으로 이 열에서 NULL 값이 있는 행을 검색합 니다.

인덱스화할 수 없거나 인덱스화하지 않아야 하는 열

- LONG 및 LONG RAW 열은 인덱스화할 수 없습니다.
- Where /Order By 절에 거의 사용되지 않는 열
- 데이터 블록을 거의 차지하지 않는 작은 테이블

인덱스화할 열 선택

임시 인덱스

• 부수적인 사용을 위해 인덱스를 생성하고 삭제할 수 있습니다. 예를 들면, 보고서가 실행되기 직전에 인덱스를 생성한 다음 나중에 삭제하도록 결정할 수 있습니다.

일반 권장 사항

- 테이블당 인덱스 수를 제한합니다. 테이블에 포함될 수 있는 인덱스 수에는 제한이 없지 만 인덱스가 많다고 반드시 성능을 향상시키지는 않습니다. 인덱스가 많을수록 갱신이 나 삭제 시 더 많은 오버헤드가 발생됩니다.
- 경험적으로 볼 때 의심의 여지가 있으면 인덱스를 생성하지 않는 것이 좋습니다. 나중에 인덱스를 생성할 수 있습니다.
- 시스템의 특성은 변경되므로 초기 인덱스 집합은 때때로 변경해야 합니다. 일반적으로 한 열에 있는 다른 값의 개수는 처음에 아주 적지만 시스템 주기 중에 늘어날 수 있습니 다. 처음에 인덱스는 중요하지 않지만 나중에 중요합니다.



인덱스는 언제 사용할 수 있습니까?

- Where 절 또는 Order By 절에서 참조될 때
- Where 절에 일부 연산자가 포함되지 않을 때
- 최적기가 결정 시
- SQL 문에 힌트 사용 시

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

인덱스는 언제 사용합니까?

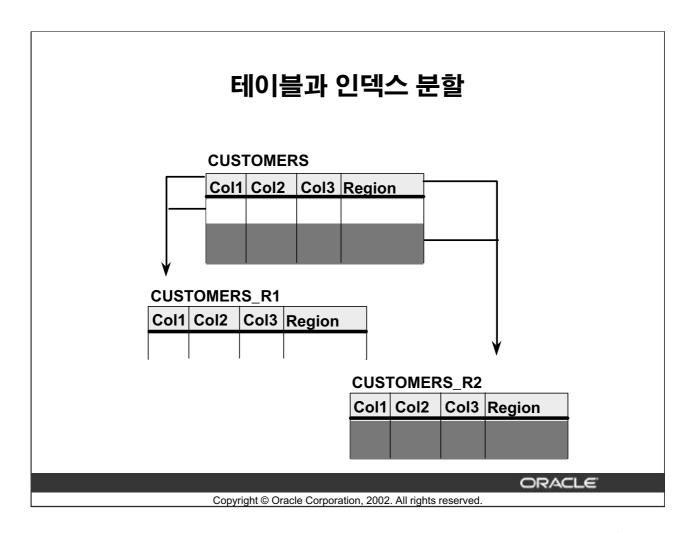
성능을 향상시키기 위해 인덱스를 생성했지만 이점이 없을 수도 있습니다.

Oracle에서 인덱스를 사용하려면 인덱스화된 열이 SQL 문의 Where 절이나 order by절에서 참 조되어야 하지만 Where 절에 다음이 포함되지 않아야 합니다.

- IS NULL
- IS NOT NULL
- !=
- LIKE
- 열이 연산이나 함수에 영향을 받을 경우(함수 기반 인덱스를 사용하지 않고 조건에 같은 함수를 사용하지 않을 경우)

예를 들면, X 열이 많은 널 값과 적은 수의 양수 값을 포함하고, 질의는 종종 NOT NULL 값을 갖는 행을 모두 선택한다고 가정합시다. 또한 인덱스가 X에 생성된다고 가정합시다. 이 경우 첫번째 상황에서 Oracle은 X에 인덱스를 사용하고 두번째 상황에서 Oracle은 X에 인덱스를 사용하지 않으므로 WHERE X > 0 조건이 WHERE X IS NOT NULL보다 선호됩니다.

그러나 이러한 방식으로 작성되어도 인덱스를 사용할 것인지 여부는 최적기가 결정합니다. 결정은 규칙이나 통계에 준합니다. SQL 문에 힌트를 사용하여 최적기가 인덱스를 사용하도 록 할 수 있습니다.



테이블과 인덱스 분할

분할된 테이블

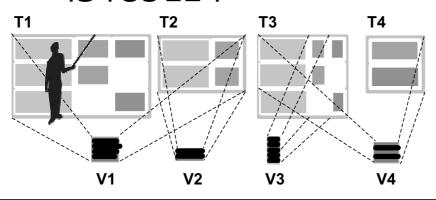
Oracle8 이후 테이블을 생성할 때 테이블을 나누고 수평 분할할 조건을 지정할 수 있습니다. 그러면 열의 고유 값만큼 분할된 테이블이 있습니다. 분할된 각 테이블에는 고유한 이름이 있지만 테이블의 전역 이름을 참조하여 액세스됩니다. 그러면 최적기는 Where 절의 값에 따라액세스할 분할 영역을 결정합니다. 이 기능의 핵심은 작은 데이터 조각을 처리한 다음 SQL 문의 속도를 향상시키는 것입니다. 특정 지역에 있는 고객에 대한 질의를 한다고 가정하면 Oracle은 CUSTOMERS 테이블의 모든 행에 액세스하지 않아도 되지만 이 지역의 모든 고객을 보유하는 조각으로만 검색을 제한할 수 있습니다. 논리적으로 테이블은 하나의 객체로 동작하고, 물리적으로 데이터는 다른 위치에 저장됩니다.

분할된 인덱스

인덱스는 동일한 개념을 사용하여 분할됩니다. 인덱스는 테이블 분할과 일치하지 않아도 되며, 다른 분할 조건을 가질 수 있고 테이블과 분할 수가 다를 수 있습니다. 이것은 특정 질의에 대한 응답을 항상 분할된 인덱스에서 찾을 수 있는 상황에 유용합니다.

뷰

- 액세스 제한
- 데이터 표시
- 데이터 구조에서 응용 프로그램 분리
- 복잡한 질의 저장
- 사용자 명령 단순화



Copyright © Oracle Corporation, 2002. All rights reserved.

ORACLE

뷰

뷰는 데이터베이스의 창입니다. 뷰는 데이터베이스에 이름이 지정되고 저장된 SELECT 문으로 정의합니다. 그러므로 뷰에는 별도의 데이터가 없습니다. 뷰는 기존 테이블에서 정보를 릴레이합니다.

뷰 사용

- 액세스 제한: 뷰 방식은 기준으로 하는 테이블에서 열과 행을 숨기는 방법 중 하나입니다.
- 데이터 표시: 뷰를 사용하여 일반 사용자가 이해하기 쉬운 방식으로 데이터를 표시할 수 있습니다. 예를 들면, 테이블에 저장된 기본 정보에서 생성된 계산 데이터를 나타낼 수 있습니다.
- 데이터 구조에서 응용 프로그램 분리: 응용 프로그램은 테이블이 아닌 뷰를 기준으로 할 수 있습니다. 여기서는 구조가 변경될 위험이 높습니다. 뷰가 사용되는 경우 기본 테이 블이 수정되어도 뷰가 그대로 남아 있으면 응용 프로그램은 유지 관리할 필요성이 없습니다.
- 복잡한 질의 저장 및 명령 단순화: 데이터 구조의 복잡성을 숨기는 데 뷰를 사용할 수 있으므로 사용자는 테이블을 조인하는 방법을 몰라도 여러 테이블에 대한 질의를 생성할 수 있습니다.
- 사용자 명령 단순화.

뷰를 사용하는 이유

- 장점
 - 동적 뷰
 - 정규화된 테이블에서 비정규화된 데이터 표시
 - SQL 문 단순화
- 단점
 - 성능에 영향을 줌
 - 경우에 따라 제한된 DML

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

뷰 사용

장점

- 뷰를 사용하면 데이터베이스에 파생된 데이터를 저장하지 않고도 사용자에게 그러한 데이터를 표시할 수 있습니다. 일반적으로 SELECT * FROM ... WHERE ...와 같은 단순 SELECT 문을 일반 사용자가 작성할 수 있도록 하는 완전히 비정규화되고 사전 조인된 정보를 뷰에 나타냅니다.
- 뷰를 동적으로 만들 수 있습니다. 예를 들면, 사용자나 날짜에 종속되는 데이터를 표시할 수 있습니다.
- 예제 뷰 유형을 사용하여 사용자에게 평일에만 오전 8:00와 오후 6:00 사이에 데이터에 액세스를 허용할 수 있습니다.

단점

- 구문 분석 시간이 조금 오래 걸리므로 뷰는 항상 조금 느립니다. 테이블과 해당 열을 찾으면 질의가 즉시 실행될 수 있습니다. 질의 조건은 뷰의 조건과 "and"로 연결됩니다. 이렇게 하면 최적기에서 생성되는 실행 계획에 영향을 줄 수 있습니다.
- 뷰는 거의 테이블처럼 동작하지만 insert, update 및 delete 문에 뷰를 사용할 때 몇 가지 제한 사항이 있습니다.

기존 방식의 설계

- 고유 인덱스
- "Check option" 절을 사용한 뷰
- 일반 호 구현

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

기존 방식의 설계

기존 시스템을 사용하면 기존 방식의 설계 기법을 볼 수 있을 것입니다. 이러한 기법은 RDBMS 기능이 그다지 향상되지 않았을 때 사용되었습니다.

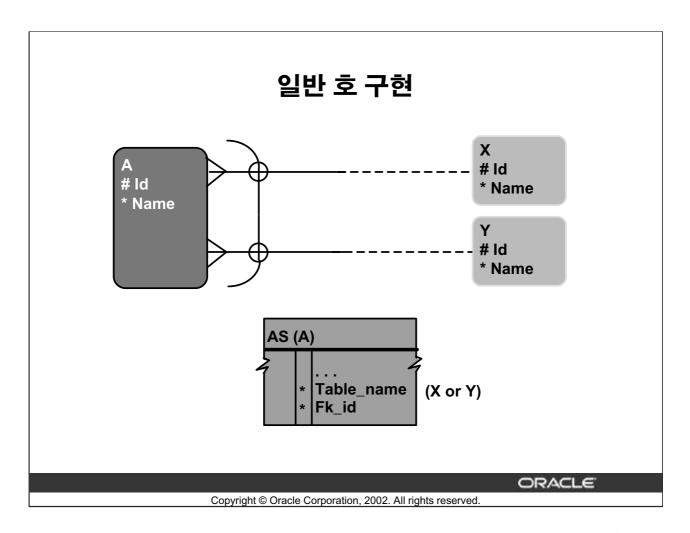
고유 인덱스

Oracle7까지는 기본 키 제약 조건을 선언할 수 없었기 때문에 고유 인덱스를 기본 키 열에 수 동으로 생성했습니다.

검사 옵션 뷰

이전 버전의 Oracle에서는 "검사 옵션이 있는" 뷰를 생성하는 것이 일반적이었습니다. 현재 폐기된 이러한 뷰는 Oracle7 이전 버전에서 데이터 무결성 및 참조 무결성을 강제로 적용하는 데 사용했습니다.

뷰의 검사 옵션 기능은 데이터베이스 트리거에 코딩할 수 있습니다. 무결성 제약 조건을 선언하고 데이터베이스 트리거를 코딩하는 것이 현재 이를 처리하는 데 주로 사용되는 방법입니다.



일반 호 구현

일반 호 구현은 기존 시스템에서 볼 수 있는 오래된 구성입니다.

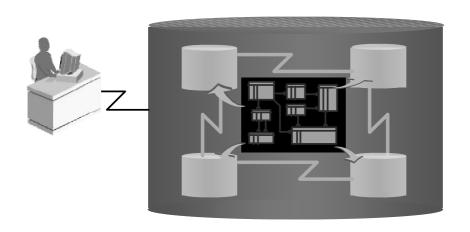
예제의 엔티티 A의 호 구현에서 호의 세 관계가 하나의 일반 외래 키 열 Fk_i id에 병합되었습니다. 외래 키 값이 참조하는 테이블에 대한 정보를 유지하는 NOT NULL 열이 AS 테이블에 추가되었습니다. 호가 필수일 때 Fk_i id에 NOT NULL 제약 조건을 사용할 수 있으므로 이 기법이 널리 사용되었습니다.

호를 구현하기 위한 이 솔루션은 다음과 같은 제한 사항으로 인해 현재 사용되지 않습니다.

- Oracle7 이후 현재 두 개의 외래 키를 선언하고 하나의 검색 제약 조건을 작성하여 호를 구현할 수 있습니다.
- 조인은 대부분의 경우 시간이 많이 걸리는 합집합 연산자를 필요로 하므로 아주 비효율적입니다.
 - select A.Name, X.Name, 'X' Type
 - from AS A, XS X where...union select A.Name, Y.Name, 'Y'
 - from AS A, YS Y where...
- 외래 키 열에 대한 외래 키 제약 조건은 두 개 이상의 기본 키를 참조할 수 없으므로 선언할 수 없습니다.

분산 데이터베이스

서로 다른 물리 데이터베이스가 하나의 논리 데이터베이스 로 나타납니다.



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

분산 설계

분산 설계는 여러 물리 데이터베이스가 서로 다른 노드에 있지만 단일 "논리 데이터베이스" OAIUse 로 나타나는 특성이 있습니다.

특성

- 여러 물리 데이터베이스
- 하나의 논리 데이터베이스 뷰
- 가능한 다른 프로세서
- 데이터베이스의 일부가 있으면 커널 실행

여러 물리 데이터베이스는 반드시 서로 상대 데이터베이스의 복사본이거나 일부일 필요가 없습니다.

다른 분할 원칙에 따라 개별 테이블 컨텐트를 다른 데이터베이스에 나타내는 방법을 결정할 수 있습니다. 수직 또는 수평 기법 또는 이 두 가지를 조합한 기법에 대해 결정할 수 있습니다. Oracle

분산 데이터베이스의 이점

- 유연성
- 회선 통신량 감소
- 위치 무관성
- 로컬 자율성
- 간단한 확장 경로

단점

• 분산된 복잡성 증가

ORACLE

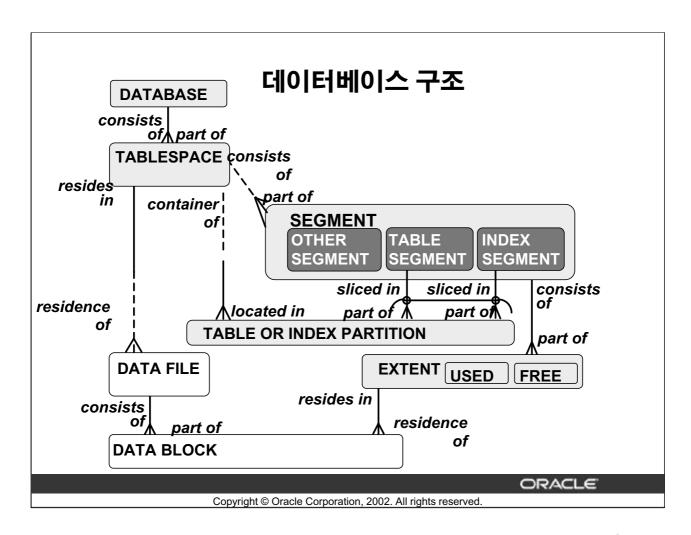
Copyright © Oracle Corporation, 2002. All rights reserved.

분산 설계의 이점

- 융통성 및 유연성 향상. 데이터 액세스는 하나의 시스템이나 링크에만 영향을 받지 않습니다. 오류가 있어도 일부 데이터는 여전히 로컬 노드에서 액세스할 수 있습니다. 실패한 링크는 대체 링크를 통해 자동으로 다시 경로가 지정될 수 있습니다.
- 데이터를 일반 데이터 사용자에게 근접하게 하여 응답 시간을 빠르게 합니다. 이렇게 하면 회선 통신량이 현격하게 줄어듭니다.
- 위치 무관성으로 인해 응용 프로그램을 변경하거나 사용자에게 통지하지 않고도 물리데이터를 이동할 수 있습니다.
- 로컬 자율성을 통해 각 물리 데이터베이스에 다음을 허용할 수 있습니다.
 - 독립적으로 관리합니다.
 - 정의 및 액세스 권한을 로컬로 생성하고 제어합니다.
- 확장 경로가 쉽게 확립됩니다.
 - 네트워크에 더 많은 프로세스가 추가될 수 있습니다.
 - 한 노드에 더 많은 데이터베이스를 포함할 수 있습니다.
 - 소프트웨어 갱신은 물리 구조와 무관합니다.

단점

분산 설계는 복잡하게 분산된 데이터로 인해 구성이 아주 복잡합니다. 시스템 관리가 복잡합 니다.



오라클 데이터베이스 구조

테이블스페이스

오라클 데이터베이스는 하나 이상의 테이블스페이스로 구성됩니다. 테이블스페이스마다 여러 개의 세그먼트를 보유할 수 있고 각 세그먼트는 해당 테이블스페이스에 완전히 포함되어 있어야 합니다. SYSTEM 테이블스페이스는 데이터베이스 생성의 일부로 생성되며 Oracle 데이터 딕셔너리 및 관련 테이블을 위해 예약되어 있어야 합니다. 이 테이블스페이스에 응용 프로그램 데이터 구조를 생성하지 않아야 합니다. 다른 세그먼트 유형에 대해 별도의 테이블스페이스를 생성하는 것이 좋습니다.

세그먼트

세그먼트는 데이터베이스 객체가 있는 공간입니다. 테이블 세그먼트, 인덱스 세그먼트, 클러스터에 사용되는 기타 세그먼트 등 세 가지 유형의 세그먼트가 있습니다. 기타 세그먼트만 한테이블스페이스의 일부여야 합니다.

분할 영역

일반적으로 세그먼트는 단일 테이블스페이스에 할당됩니다. 그러나 Oracle8을 사용하면 테이블이나 인덱스 세그먼트를 두 개 이상의 테이블스페이스에 나타낼 수 있습니다. 이러한 기법을 분할이라고 합니다. 분할 영역은 하나의 테이블스페이스에 상주하는 테이블 세그먼트(또는 인덱스 세그먼트)의 일부입니다.

오라클 데이터베이스 구조

확장 영역

세그먼트에 더 많은 공간이 필요할 때마다 다수의 연속 블록이 확장 영역으로 할당됩니다. 세 그먼트에 할당할 수 있는 확장 영역 수에 대한 최대 한도는 없습니다. 일반적으로 세그먼트에 충분히 큰 초기 확장 영역을 갖도록 하여 작은 확장 영역 수가 지나치게 많아지지 않게 하는 것이 바람직합니다.

데이터 파일

데이터 파일은 데이터베이스 데이터를 물리적으로 포함하는 운영 체제 파일입니다. 데이터 파일은 데이터 블록으로 구성됩니다.

데이터 블록

데이터 블록은 Oracle이 한 번의 읽기 작업에서 읽는 가장 적은 양의 데이터입니다. 데이터 블록은 항상 한 확장 영역의 정보만 포함합니다.

열과 행으로 구성된 논리 테이블과 데이터 파일에 위치하고 확장 영역에 구성되어 있는 데이 터베이스 블록으로 구성된 공간을 사용하는 물리 테이블은 차이가 있습니다.



요약

- 데이터 유형
- 기본 키, 외래 키, 인공 키
- 인덱스
- 분할
- 뷰
- 분산 설계

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

요약

- Oracle은 테이블 열에 대해 선택할 수 있는 다양한 데이터 유형을 제공합니다.
- 기본 키는 테이블에 필요합니다. 인공 키는 복잡한 기본 키 구현에 적합한 해결책이 될 수 있습니다.
- 인덱스는 질의 성능을 향상시키고 고유 값을 보장하는 방법을 제공합니다.
- 테이블 분할도 성능 문제에 대한 해결책이 될 수 있습니다.
- 뷰는 융통성 있고, 안전하며, 사용자에게 편리한 객체입니다.
- 분산 설계는 복잡한 기법입니다. 분산 설계를 사용하면 데이터는 사용자에게 보다 가까이 있을 수 있습니다.

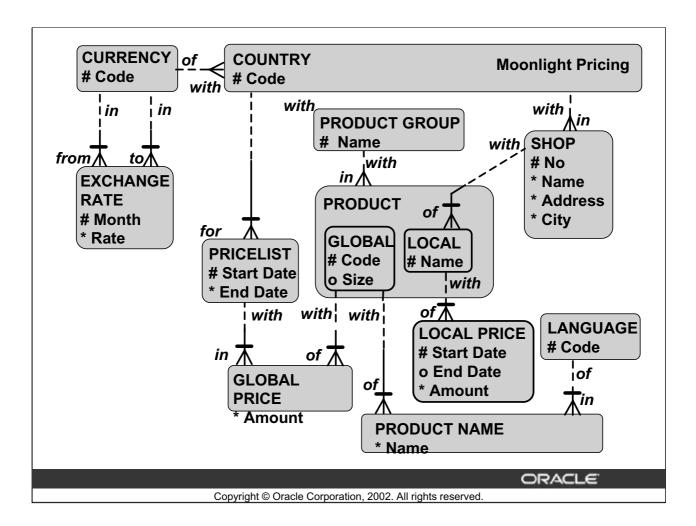
연습

- 데이터 유형
- 인공 키
- 제품 그림

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.





연습 9-1: 데이터 유형

이 연습의 목적은 제안된 데이터 유형에 대해 품질 검사를 수행하는 것입니다. **시나리오**Moorlight 기교 ^ oracle Internal & Oracle

Moonlight 가격을 보여주는 모델을 사용합니다.

데이터 유형(1)

테이블	열	권장 데이터 유형	사용자가 선택한 데이터 유형
COUNTRIES CURRENCIES EXCHANGE_RATES PRICE_LISTS PRODUCT_GROUPS PRODUCTS	Code Code Month Rate Start_date End_date Name Code Size Pdt_type	Varchar2(2) Varchar2(3) Date Number(8,4) Date Date Char(8) Char(10) Number(4,2) Number(1)	

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

과제

1. 다음은 테이블 이름 및 열 이름과 제안되는 데이터 유형입니다. 이러한 항목에 대해 품질 검사를 수행하십시오. 적합하다고 생각되면 대안을 제시하십시오.



데이터 유형(2)

테이블	열	사용자가 선택한 데이터 유형
GLOBAL_PRICES LOCAL_PRICES	Amount Start_date End_date Amount	
SHOPS	Name Address City	

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

과제

- 」기반으로 합 Only Oracle Internal & 다음 열에 대한 데이터 유형을 제안하십시오. 열들은 모두 이전 연습을 기반으로 합니

연습 9-2: 인공 키

목표

Moonlight Coffees 계약이 끝나 갑니다. 작업은 거의 완료되었습니다.

시나리오

일부 Moonlight 테이블에 사용할 수 있는 인공 키를 결정해야 합니다. 모델은 이전 연습에서 사용된 것과 동일합니다.

과제

- · 각 테이블에 대해 인공 키를 생성하면 유용하다고 생각되는지 지적하고 그 이유를 설명 하십시오.
 - COUNTRIES
 - GLOBAL PRICES
 - PRICE LISTS
- Moonlight 모델을 기반으로 하는 테이블(있는 경우)에 인공 키를 생성하는 것이 적합하지 않은 테이블은 무엇입니까?



연습 9-3: 제품 그림

목표

이 연습의 목적은 새 요구 사항을 충족하도록 설계를 수정하는 것입니다.

시나리오

이것이 Moonlight Coffees에 대한 마지막 작업입니다. 내일이면 Moonlight에 대한 모든 것을 잊고 여유롭게 커피를 마실 수 있을 것입니다.

전자 상거래 시장에 첫 발을 들여놓기로 결정했습니다. 한 가지 목표는 고객이 Moonlight의 웹 사이트를 참조할 수 있도록 하는 것입니다. 이 사이트는 제품 정보를 제공합니다. 각 제품에 대해 적어도 두 개의 추가 속성이 확인되었습니다.

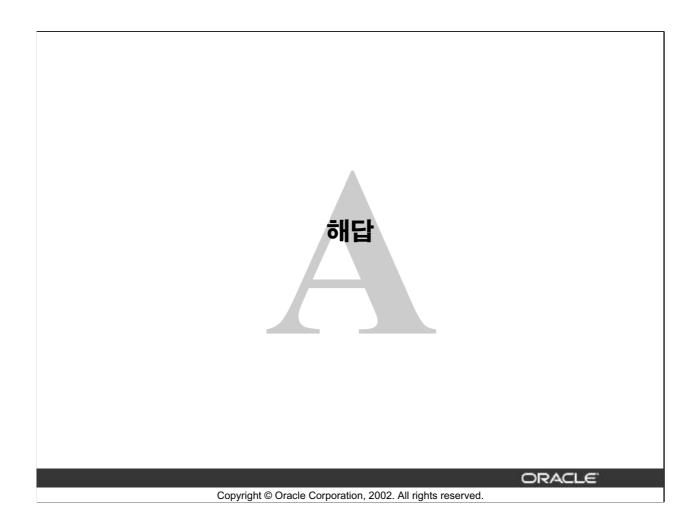
첫번째는 제품 이미지에 대한 Picture 속성입니다. 두번째는 브라우저에 표시할 수 있는 제품 설명을 보유하고 있는 HTML Document 속성입니다. 다른 속성이 올 수 있습니다.

과제

- 각 열에 사용하면 좋을 데이터 유형을 결정하십시오.
- 사용자는 이전 Oracle 버전에서는 테이블당 두 개 이상의 long 유형 열을 허용하지 않는 다는 사실을 알고 있습니다. 아직도 이러한 제한이 적용되는지는 정확히 모릅니다. 구현에 대한 의견을 제시하십시오.







Oracle Internal & OAI Use Only

해답: 인스턴스인가 엔티티인가?

개념	E/A/I?	인스턴스 또는 엔티티 예제
PRESIDENT	Е	Lincoln, Washington, Gorbachev
ELLA FITZGERALD	ı	STAR, SINGER, PERSON
DOG	E	Snoopy
ANIMAL	Е	Cat, Dog,
HEIGHT	Α	PERSON, BUILDING,
TYPE OF TRANSPORT	E	CAR
Number of Wheels	A	CAR
My current car	I	CAR

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 1-1 인스턴스 또는 엔티티: 해답

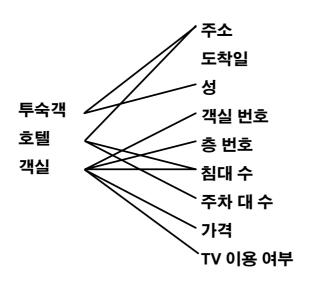
다음 개념 중 엔티티, 속성, 인스턴스에 해당하는 것을 각각 나열하십시오. 엔티티에 표시한 경우 예제 인스턴스를 제시하십시오. 속성 또는 인스턴스에 표시한 경우 엔티티를 제시하십시오. 마지막 세 행에 대해 적절한 개념을 찾으십시오.

가능한 해답:

사실, 모든 개념은 엔티티와 엔티티의 인스턴스 둘 다 될 수 있습니다. 이것은 업무 관점에 따라 좌우됩니다. 일부 업무의 경우에는 또 다른 관점이 적용될 수도 있습니다.

심지어 Ella Fitzgerald도 엔티티일 수 있습니다. Ella Fitzgerald의 여러 인스턴스가 있을 수 있는 닮은 사람 컨테스트나 사진 전시회 같은 상황을 고려해보십시오.

해답: 손님



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 1-2 손님: 해답

Olacle

속성과 해당 속성이 설명하는 엔티티 간에 선을 그립니다.

가능한 해답:

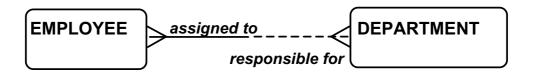
여러 엔티티가 동일한 속성을 가지는 것처럼 보일 수 있습니다. 실제로 이러한 상황은 엔티티가 우연히 이름이 같은 다른 속성을 가지는 경우로 보아야 합니다.

규칙은 하나의 속성이 항상 하나의 엔티티에 속해야 한다는 것입니다. 엔티티마다 속성의 데이터 유형이나 길이가 다를 수 있습니다.

제시된 해답 중 일부는 논쟁의 여지가 있습니다. 이전 도착일은 중요하지 않으며 변경될 수도 있다는 점에서 투숙객에 도착일이 속한다고 주장할 수 있습니다.

객실 마다 침대 개수를 일정하게 유지하려는 경우에는, 객실 당 침대 개수를 모두 더한 합계에 따라 값을 구할 수 있으므로 호텔 레벨에서 이 값을 유지할 필요가 없습니다.

해답: 읽기



- A 각 EMPLOYEE는 하나 이상의 DEPARTMENT에 할당될 수 있습니다. 각 DEPARTMENT는 하나 이상의 EMPLOYEE를 책임져야 합니다.
- B 각 EMPLOYEE는 하나 이상의 DEPARTMENT에 할당되어야 합니다. 각 DEPARTMENT는 하나 이상의 EMPLOYEE를 책임질 수 있습니다.
- C 각 EMPLOYEE는 정확히 하나의 DEPARTMENT에 할당되어야 합니다. 각 DEPARTMENT는 정확히 하나의 EMPLOYEE를 책임질 수 있습니다.

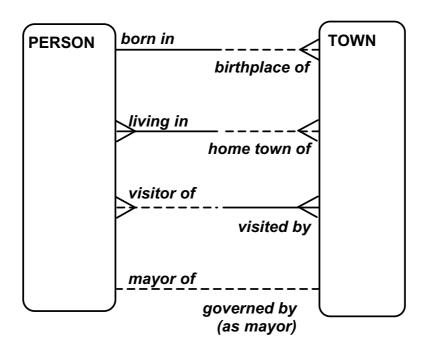
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 1-3 읽기: 해답

Oracle Internal & OAI Use Only 어느 내용이 도표에 해당합니까? 옵션 B가 올바르게 읽은 것입니다.

연습: 읽기 및 주석 달기



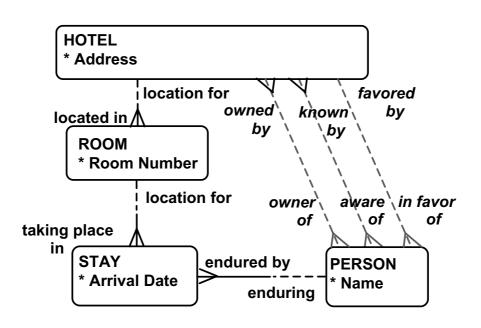
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 1-4 읽기 및 주석 달기: 해답

- 1. 여기 제시된 모델에서 각 관계를 읽어 보십시오.
- 2. 다음으로 방금 읽은 관계에 주석을 답니다. 사람 및 도시에 대한 지식을 사용하십시오. 하향식:
- 각 Person은 하나 이상의 Town에서 태어나야 합니다.
 - 각 Town은 정확히 한 Person의 출생지일 수 있습니다.
 - 의견: 양쪽에 문제가 있는 듯합니다.
- 각 Person은 하나 이상의 Town에 거주해야 합니다.
 - 각 Town은 하나 이상의 Person의 거주지일 수 있습니다.
 - 의견: 첫번째 문장에는 분명히 문제가 있습니다. 이 문장을 선택하는 것도 올바르지 않은 것처럼 보입니다.
- 각 Person은 하나 이상의 Town의 방문객일 수 있습니다.
 - 각 Town은 하나 이상의 Person에 의해 방문되어야 합니다.
 - 의견: 두번째 문장을 선택할 수도 있지만 Town의 정의에 따라 잘못된 것일 수 있습니다.
- 각 Person은 정확히 한 Town의 시장일 수 있습니다.
 - 각 Town은 정확히 한 Person(시장)에 의해 통치될 수 있습니다.
 - 의견: 두 가지가 모두 올바른 것 같습니다. 단, 과거 정보를 보관해야 하는 경우 양쪽 관계는 "m:m" 관계여야 합니다.

해답: 호텔



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 1-5 호텔: 해답

여기 제시된 모델의 관계에 대해 주석을 다십시오.

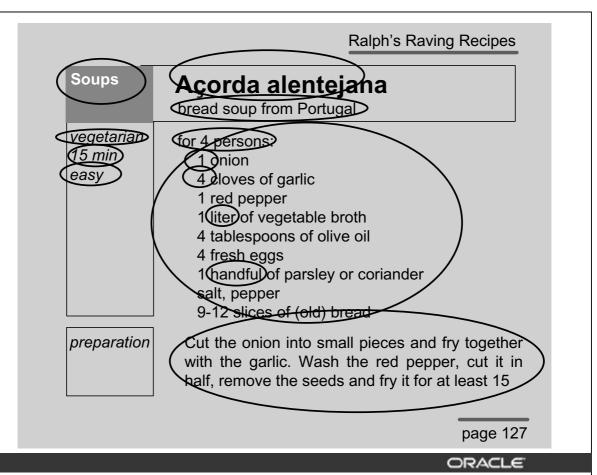
가능한 의견:

- STAY 엔티티와 HOTEL 엔티티 간의 lodging for/in 관계는 located in/location for와 같은 ROOM 및 HOTEL 간 관계로 대체되어야 합니다.
- 모든 PERSON에는 둘 이상의 STAY가 적용될 수 있으며 하나의 STAY는 한 사람에 대한 것이어야 합니다. 두 사람이 서명한 경우 STAY의 인스턴스는 두 개가 됩니다.
- PERSON과 HOTEL 간의 관계가 중복된 것 같습니다.

가능한 모델은 다음과 같습니다.

Olacle

- 호텔 업무에 다소 도움이 될 수 있는 PERSON과 HOTEL 간의 가능한 관계 두 가지를 추가로 구성합니다.
- PERSON과 HOTEL 간의 가능한 관계에 대해서는 도표를 참조하십시오.



Copyright © Oracle Corporation, 2002. All rights reserved.

연습 1-6 조리법: 해답

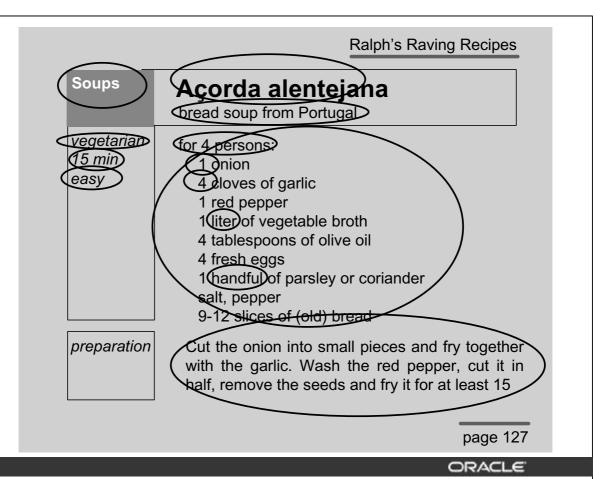
- 1 Ralph의 저명한 Raving Recipe 책자의 예제 페이지를 분석하고 중요하다고 여기는 여러 다른 정보 유형을 나열합니다.
- 2 다양한 정보 유형을 엔티티 및 속성으로 그룹화합니다.
- 3 찾아낸 관계에 이름을 지정하고 도표를 그립니다.

가능한 해답:

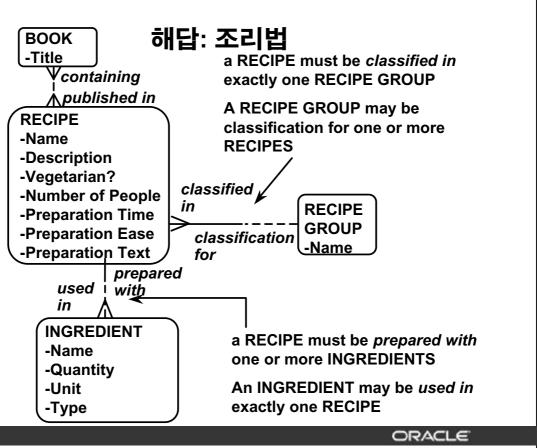
다음은 정보 유형의 목록입니다. 다른 정보 유형도 있을 수 있으며 이름을 다르게 지정할 수 있습니다. 해당 텍스트에서 가져온 예는 기울임꼴로 표시됩니다.

- Recipe (Açorda)
- Original Name (Açorda alentejana)
- Description (Bread Soup from Portugal)
- Vegetarian Indicator (Vegetarian)
- Number of Persons (4)
- Ease of Preparation (Easy)
- Preparation Time (15 min)
- Preparation Description ("Cut the onion ...")
- Ingredient (Garlic, bread)
- Quantity (4, for the cloves of garlic, 1 for the pepper)
- Unit (liter, piece, handful)
- Recipe Group (Soups)

Data Modeling and Relational Database Design A-7



Copyright © Oracle Corporation, 2002. All rights reserved.



Copyright © Oracle Corporation, 2002. All rights reserved.

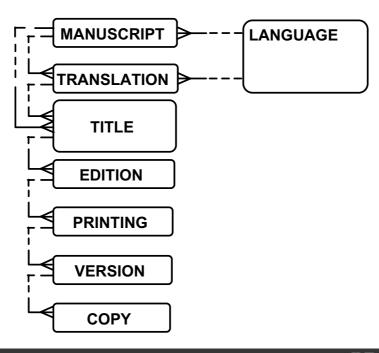
연습 1-6 조리법: 해답

재료는 적절히 분류되어 있습니다. 재료 분류에 사용된 인텔리전스는 숨겨져 있어야 합니다. 이에 따라 Ingredient Type(vegetable, spice)이 생성되었을 것입니다.

조리법에 대한 가능한 모델은 다음과 같습니다.

다른 해답에 AUTHOR, BOOK, BOOK TYPE, RECIPE, RECIPE ITEM, INGREDIENT 및 INGREDIENT GROUP 엔티티를 포함할 수도 있습니다.

해답: 서적



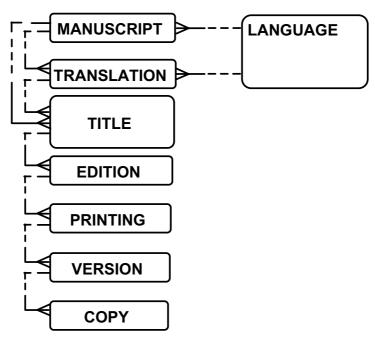
ORACLE!

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 2-1: 서적: 해답

- 1 이 텍스트에서 "서적"이라는 단어는 여러 의미로 사용됩니다. 이러한 의미는 출판사나 도서 판매업체의 관점에서 볼 때 여러 다른 엔티티에 해당합니다. 모두 서적을 나타내는 다양한 엔티티를 구별해 보십시오. 이러한 엔티티에 대해 보다 적합한 이름을 부여하고 이러한 이름을 구분하기 위한 한 두 개의 속성을 구성하십시오.
- 2 텍스트에 준하여 ER 모델을 생성합니다. 페이지 맨 위에 가장 일반적인 엔티티를 표시하고 맨 아래에 가장 구체적인 엔티티를 표시하십시오. 중간에 다른 엔티티를 적절히 표시하십시오. 관계 이름에 대해서는 신경쓸 필요가 없습니다.

해답: 서적



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 2-1 서적: 해답

예제에서 "서적"이라는 단어는 여러 의미를 가집니다. 속성은 대괄호로 묶입니다. 별표가 붙 은 속성은 다른 엔티티와의 관계로서 우선적으로 모델링됩니다.

- MANUSCRIPT (Title, Year of Completion, Author*)
- **EDITION (Sequence Number of Printing)**
- COPY (position on the shelf)
- TRANSLATION IN LANGUAGE (Language*, Translator*)
- TITLE (Authorized Text)
- **PRINTING (Font*, Year of Printing)**
- Oracle Intel **VERSION** (Hardcover Indicator)

및 기타 등등

해답: Moonlight

COUNTRY PRODUCT

COUNTRY COMMUNITY PRODUCT GROUP

CURRENCY PRICE

DEPARTMENT SALE

DRINK SHOP

EMPLOYEE STOCK OPTION

EXCHANGE RATE STOCK PRICE

FOOD TICKER SYMBOL

JOB

LOCATION TYPE

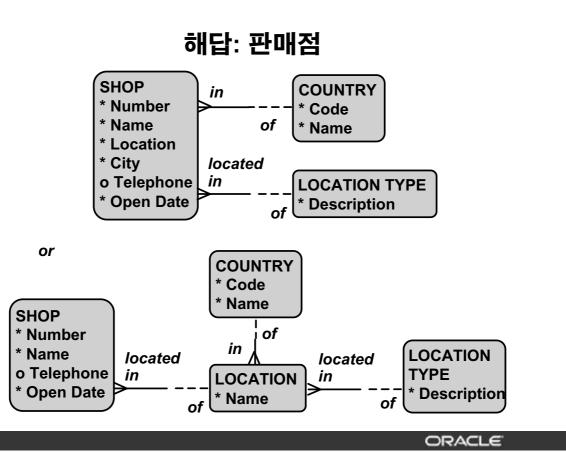
PASTRY

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 2-2: Moonlight: 해답

- 1. Moonlight Coffees에 중요하다고 생각되는 엔티티 목록을 약 15가지 만드십시오. 상상력 과 상식을 활용하여 아래 출력된 요약에서 얻은 정보를 사용하십시오.
 - 가능한 엔티티는 알파벳 순서로 정렬됩니다.
- 2. 다음을 나타내는 엔티티의 공식적인 정의를 작성하십시오. 가능한 정의는 다음과 같습니다.
 - 판매점은 Moonlight에서 고객에게 제품을 판매하거나, 판매했거나, 앞으로 판매할 소매점을 나타냅니다. 판매점의 예로 ...에서 개점한 ...의 판매점을 들 수 있습니 다..
 - 직원은 특정 직무를 맡아서 Moonlight에서 일하거나, 일했거나, 앞으로 일할 사람이며 Moonlight에서 급여를 받는 사람입니다. 직원의 예로 ...에서 일하는 ...을(를)들수 있습니다.



Copyright © Oracle Corporation, 2002. All rights reserved.

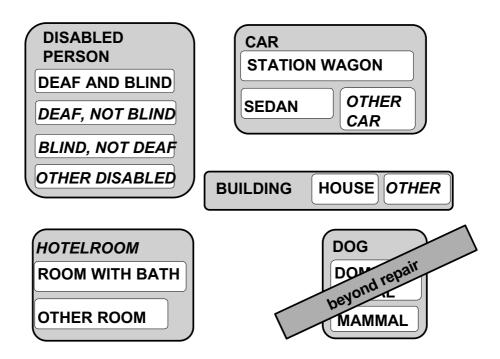
연습 2-3: 판매점: 해답

목록의 정보를 ER 모델의 기초로 사용하십시오. 매우 신중하게 속성을 모두 찾아 내십시오.

두 해답 모두 적절합니다. 두번째 해답은 JFK Airport와 같은 위치를 여러 판매점에 대한 가능한 위치로 간주합니다. 대부분의 위치에는 하나의 판매점만 있을 것입니다.

두번째 해답에서는 한 위치에 반드시 하나의 판매점만 있어야 하는 것은 아닙니다. 이 사항은 향후 판매점이 들어설 위치에도 적용됩니다. 이러한 측면에서, 두번째 모델이 좀더 융통성이 있습니다.

해답: 하위 유형



Copyright © Oracle Corporation, 2002. All rights reserved.

연습 2-4 하위 유형: 해답

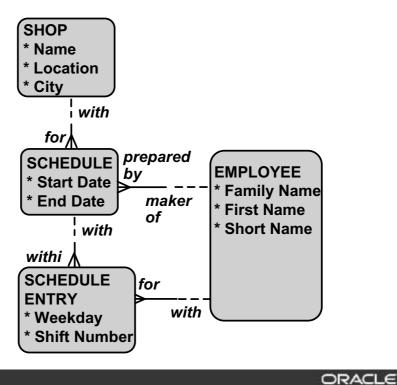
그림에서 잘못된 하위 유형을 모두 찾습니다. 하위 유형이 잘못되었다고 생각하는 이유를 설명하십시오.

더 나은 형태로 모델을 조정하십시오.

모든 예제는 하위 유형이 잘못되어 있습니다.

- DISABLED PERSON의 하위 유형은 상호 배타적이지 않습니다. 즉, DEAF와 BLIND는 중첩될 수 있습니다.
- CAR의 하위 유형은 완전하지 않습니다. 즉, 다용도 자동차도 아니고 고급 세단도 아닌 차가 있습니다.
- BUILDING 엔티티는 적어도 두 개의 하위 유형을 가져야 합니다.
- ROOM WITH BATH가 HOTEL의 하위 유형인 경우, 이것은 모든 ROOM WITH BATH가 HOTEL임을 의미합니다.
- DOMESTIC ANIMAL과 MAMMAL은 상호 배타적이지 않습니다. 실제로, DOG는 한쪽만의 하위 유형이 아니라 두 엔티티 모두의 하위 유형입니다.

해답: 일정(1/2)

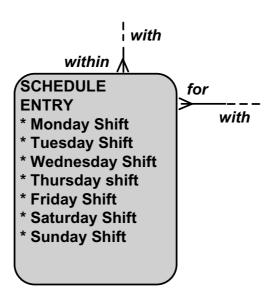


Copyright © Oracle Corporation, 2002. All rights reserved.

연습 2-5 일정: 해답

Amsterdam의 판매점 중 하나에서 사용되는 근무 일정을 엔티티 관계 모델에 대한 기반으로 사용하십시오. 예를 들어 일정에 따르면 10월 12-18일 주중에 Annet B는 월요일, 금요일 및 토요일에 첫번째 교대 근무 일정이 잡혀 있습니다.

해답: 일정(2/2)



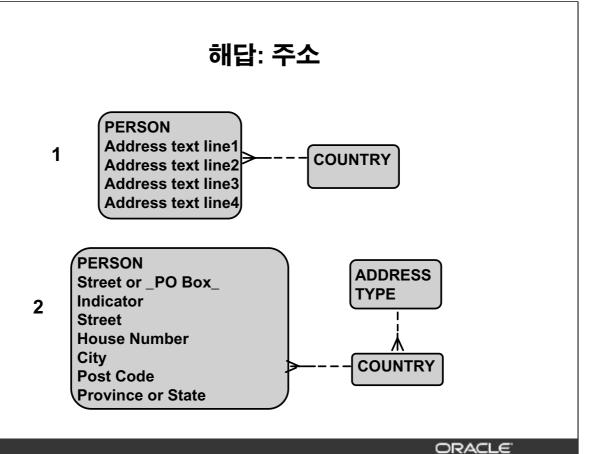
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 2-5 일정: 해답

다른 해답은 다음과 같이 SCHEDULE ENTRY를 다르게 모델링하는 것입니다.

첫번째 모델은 하루에 한 사람이 두 번 이상 교대하도록 허용하지만 두번째 모델은 그렇지 않습니다.



Copyright © Oracle Corporation, 2002. All rights reserved.

연습 2-6 주소: 해답

PERSON(또는 ADDRESS) 엔티티는 아래 예제와 같이 주소를 설명하는 속성을 가질 수 있습니다.

- 1 향후 시스템이 정확한 국제 메일 기능을 생성하도록 요구되는 경우 어떻게 주소 정보를 모델링할 수 있습니까?
- 해답 1은 다양한 주소 형식을 쉽게 처리할 수 있습니다. 해답 1은 하나의 주소가 최대 4 행의 텍스트로 구성될 수 있다는 사실을 간단히 인식합니다. 이러한 사실은 제공된 주소 를 해석해야 할 필요를 없애줍니다. 1045 ST 88은 Street 88에 있는 번지가 1045인 집일 까요 아니면 1045 Street에 있는 번지가 88인 집일까요?
- 해답 2는 주소를 개별 부분으로 자르고 사서함도 구분합니다. 이 해답은 해석에 문제가 없다고 가정합니다. ADDRESS TYPE 엔티티를 추가하여 다양한 주소 부분의 표시 순서를 설정할 수 있습니다.
- 2 이전 연습에서 생성한 모델에서도 아래 주소를 사용할 수 있습니까?
- 3 또한 시스템에 다음 범주의 주소를 검색하기 위한 기능이 요구되는 경우 모델이 달라지는지 확인합니다. 필요에 맞게 변경합니다.

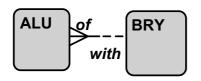
연습 2-6 주소: 해답

모든 주소:

- Kirkland 내에 위치
- Bonn에서 우편 번호가 53111인 경우
- 사서함
- 다음 위치:
 - Oxford Road 또는
 - Oxford Rd 또는
 - OXFORD ROAD 또는
 - OXFORD RD
- 읽기
- 두번째 모델은 필수 질의 대부분을 허용합니다. 단, 마지막 질의에는 약간의 시험 작업이 필요할 수 있습니다. 시스템에서 주소 정보에 대한 빠르고 쉬운 검색을 허용할 필요가 있는 경우 주소 이름 요소를 정렬하고 약어로 나타내기 위한 특정 규칙 세트를 사용하여 주소 정보를 대문자로 저장하는 별도의 속성이 추가됩니다.
- 예를 들어, "President A. B. Collins Road" 같은 거리 이름은 COLLINS, RD PRES A B로 사용됩니다. 이러한 속성은 시스템에 이러한 목적의 내장 구문 분석기가 있어야만 주소 행에서 파생될 수 있습니다.



해답: 관계 읽기

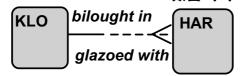


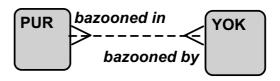
모든 ALU는 정확히 하나의 BRY에 대해 be of 관계여야 합니다.

모든 BRY는 하나 이상의 ALU에 대해 be with 관계일 수 있습니다.

모든 PUR은 하나 이상의 YOK에 대해 be bazooned in 관계일 수 있습니다.

모든 YOK는 하나 이상의 PUR에 대해 be bazooned by 관계일 수 있습니다.





모든 KLO는 하나 이상의 HAR에 대해 be bilought in 관계일 수 있습니다.

모든 HAR은 정확히 하나의 KLO에 대해 be glazoed with 관계일 수 있습니다.

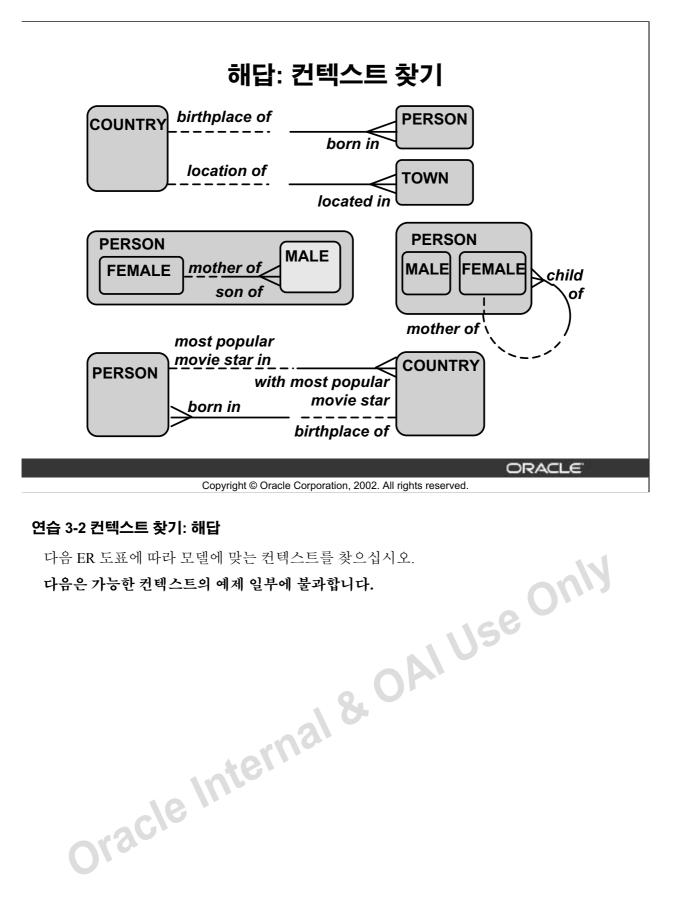
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 3-1 관계 읽기: 해답

도표를 두 시각에서 읽어 보십시오. 해당 업무 영역을 파악하고 있으나 ER 모델을 읽는 방법을 모르는 사람들이 이해하고 확인할 수 있는 문장을 만드십시오.

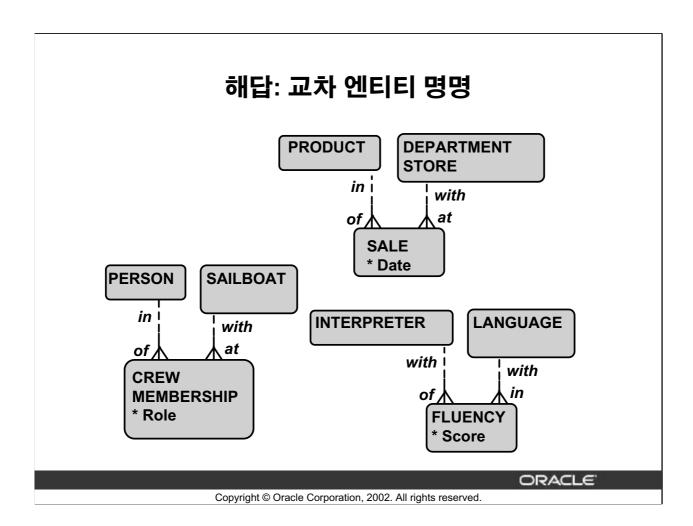
이 어구는 이상한 SF 영화에서 가져온 것처럼 들리지만 읽을 때는 유창하게 읽어야 합니다. 반드시 이 행을 읽은 후에 방금 들은 내용을 생각해보십시오. 실제 상황에서는 때때로 실제로 모델링된 것이 아니라 본인이 생각하기에 표현된 것을 읽게 됩니다. 이와 같이 본인의 자연스 러운 해석 동작을 멈추는 것은 정말 어려운 일입니다.



연습 3-2 컨텍스트 찾기: 해답

다음 ER 도표에 따라 모델에 맞는 컨텍스트를 찾으십시오.

다음은 가능한 컨텍스트의 예제 일부에 불과합니다.

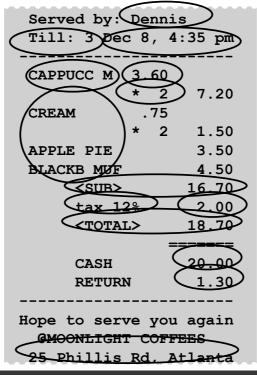


연습 3-3 교차 엔티티 명명: 해답

- 1 다음의 m:m 관계를 분석하십시오. 교차 엔티티에 대해 적용 가능한 이름을 찾으십시오.
- 2 교차 엔티티별로 다소 중요한 업무 컨텍스트에 적합한 하나 이상의 속성을 만드십시오. 해당 속성에 명확한 이름을 지정하십시오.

이것은 원하는 기능에 대한 단지 하나의 가능한 상황과 하나의 가정에 준하는 가능한 해답입니다.

해답: 영수증



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 3-4 영수증: 해답

영수증의 정보를 사용하여 엔티티 및 속성 목록을 만드십시오.

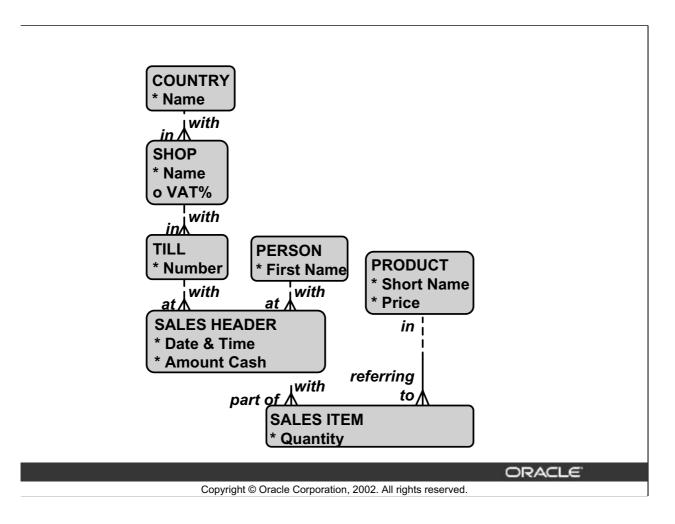
한 가지 가정은 영수증에 표시되는 모든 파생 불가능 정보는 적어도 얼마 동안 저장되어야 한 다는 것입니다.

이 모델은 몇 가지 가정을 기반으로 합니다. 하나는 모든 엔티티가 따로 존재할 수 있다는 것입니다. 예를 들어, 판매점이 없는 국가(시스템에 기록됨), 담당자 번호가 없는 판매점, 판매가 없는 담당자 번호가 있을 수 있습니다. 이것은 한 쪽에서 선택인 모든 1:m 관계를 설명합니다.

이러한 제품 종류에 대한 판매세 또는 부가가치세 비율은 미국처럼 지역 판매점 레벨에서 또는 많은 유럽 국가의 경우처럼 국가 레벨에서 고정된 값입니다. 이 모델에서는 판매점 레벨에서 모델링하는 방식이 선택되었습니다.

많은 국가에서는 일반 제품에 저 부가가치세 비율이 적용되고 사치품에 고 부가가치세 비율이 적용된다는 사실을 알고 있습니다. Moonlight에서는 광천수와 커피가 다른 범주에 속할 수 있으므로 두 가지 부가가치세 비율을 취급해야 할 수 있습니다.

제품의 가격이 시간에 따라 달라질 수 있으므로 PRODUCT의 Attribute Price는 별도의 엔티티로서 모델링될 수 있습니다.

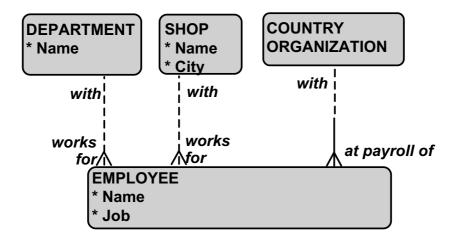


연습 3-5 Moonlight P&O: 해답

- 1 다음의 직원 및 조직 정보를 기준으로 엔티티 관계 모델을 생성하십시오.
- EMPLOYEE에서 DEPARTMENT 및 SHOP으로의 선택적 관계에 유의하십시오. 이 관계는 "... 국가 본점 또는 판매점을 위한..."에서 화생됩니다.

해답: Moonlight P&O(1)

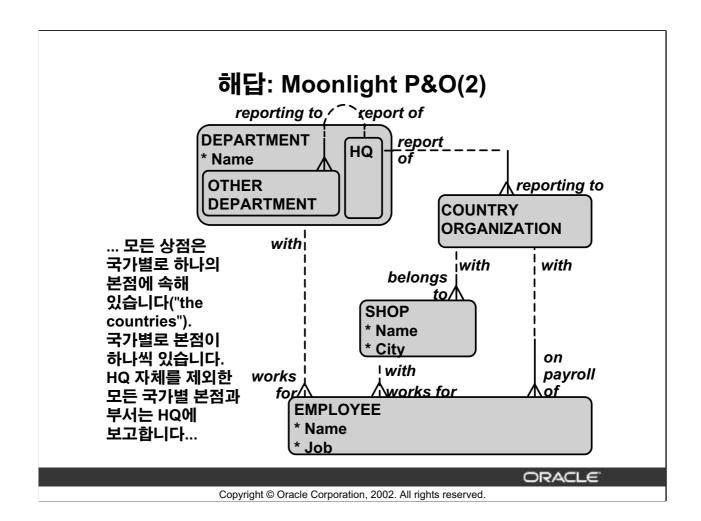
Moonlight Coffee의 모든 사원은 "Global Pricing" 또는 "HQ"와 같은 매장 또는 판매점에서 일합니다. 전사원은 국가 본점 중 하나에서 급료를 받습니다. 예를 들어, Jill은 런던에서 상점 관리자로 일하고, Werner는 독일에서 회계 업무를 맡고 있는 재무 관리자입니다...



Copyright © Oracle Corporation, 2002. All rights reserved.

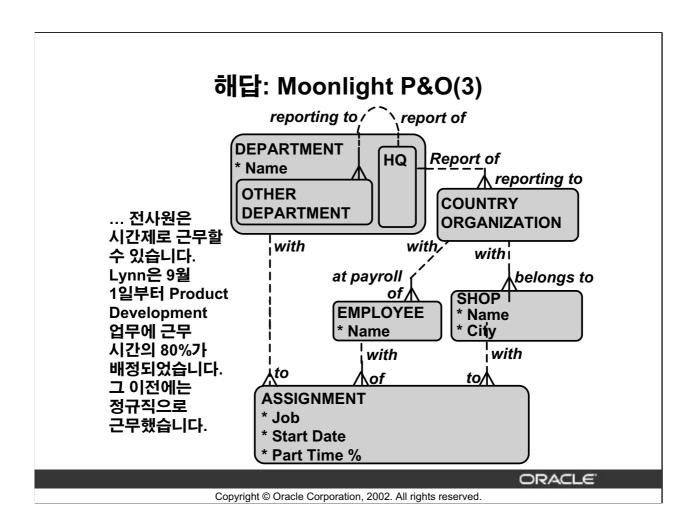
ORACLE

Oracle Internal & OAI Use Only

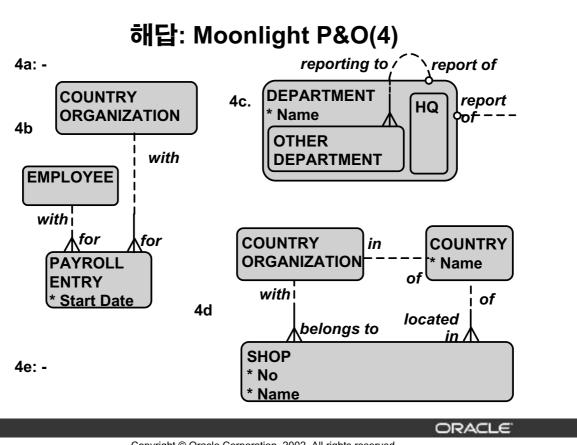


연습 3-5 Moonlight P&O: 해답

Oracle Internal & OAI Use Only 2 이 정보를 기준으로 도표를 확장 또는 수정하십시오.



Oracle Internal & OAI Use Only 연습 3-5 Moonlight P&O: 해답



Copyright © Oracle Corporation, 2002. All rights reserved.

연습 3-5 Moonlight P&O: 해답

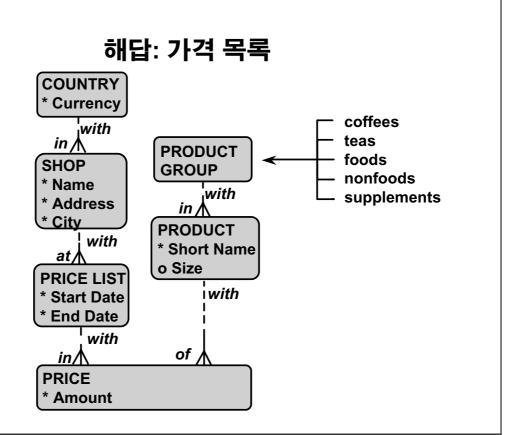
- 4 다음과 같은 새로운 정보를 수용할 수 있도록 필요하고 가능한 경우 모델을 변경합니다.
 - a Jan은 프라하에 잇는 두 개의 상점에서 교대로 근무합니다.
- a. 모델을 변경할 필요는 없습니다. 다양한 장소에서 직원을 고용할 수 있으며, 모델에서 이 것이 허용됩니다.
 - e 권한 충돌을 방지하기 위해 전사원은 동시에 한 부서와 한 상점에서 일할 수 없습니다.
- b. 도표를 참조하십시오.
 - b 지난 해 Tess는 브라질에서 상점 관리자직을 사임하고 토론토로 이주하였습니다. 최근에 그녀는 토론토 공항의 판매점으로 합류했습니다.
- c. 도표를 참조하십시오.
 - c 직접 보고서의 수를 줄이기 위해, 부서와 각 국의 본점들은 본사 대신 다른 부서에 보고할 수 있습니다.
- d. 도표를 참조하십시오.
 - d 룩셈부르크에 있는 상점은 벨기에에 보고합니다.
- e. 지금까지 배운 내용으로는 아직 모델링을 할 수 없습니다.

연습 3-5 Moonlight P&O: 해답

- 1. 당신이 작성한 모델로 다음 질문들에 답변할 수 있습니까?
 - a 현재 Operations 부서에서 근무하는 사람은 누구입니까?
 - b 현재 프랑스 몽마르트르에 있는 Moonlight La Lune에서 근무하는 사람은 누구입니까?
 - a, b. 이전 모델을 사용할 경우 이러한 질문에 대답할 수 있습니다.
 - c 현재 프랑스에서 Marketing 부서에서 근무하는 직원이 있습니까?
 - c. 부서는 국가와 관련이 없으므로 프랑스에는 마케팅 부서가 없습니다. 그러나 프 랑스의 종업원 명부에도 포함되어 있으며 Marketing 부서에서 일하고 있는 사람 을 찾을 수 있습니다.
 - d 사원 수를 기준으로 최대 국가는 어디입니까? 매니저 수를 기준으로 하면 어떻습니까? 시간제 근무자 수를 기준으로 하면 어떻습니까?
 - d. 이 질문은 국가 본점당 별도의 현재 할당량을 계산함으로써 대답할 수 있습니다.
 - e Lynn이 회사에 근무한지 5년째 되는 날은 언제입니까? 마찬가지로 Moonlight에서 근무하는 Tess의 근무 5년째 되는 날은 언제입니까?
 - e 이 모델은 아직 ASSIGNMENT 및 PAYROLL ENTRY에 대한 End Date 속성을 포함하지 않습니다. Moonlight에서 Tess의 경력에 비춰볼 때 기간이 반드시 연속 되는 것은 아니므로 End Date 속성이 필요합니다.
 - f 퇴직자 수가 가장 낮은 국가는 어디입니까?
 f 대답할 수 있습니다.

주:

- 부서와 국가 본점은 조직의 일부이기 때문에 이 모델은 상당히 복잡합니다. 부서는 국가 본점과 달리 위치를 갖지 않습니다. 부서는 '가상의' 조직입니다. 부서를 작업자 그룹보 다는 작업 모음으로 생각하십시오. 작업하는 사람이 없어도 부서는 여전히 존재할 수 있 으며 작업은 여전히 남아 있습니다. 부서의 위치는 '종업원 명단에 현재 해당 부서의 관 리자가 있는 국가 본점의 사무실'로 정의할 수 있습니다.
- 국가 본점은 국가와 다릅니다. 국가 본점은 Moonlight에서 정의할 수 있는 반면 국가는 Moonlight의 통제 범위를 벗어나므로 두 가지를 구분하는 것이 좋습니다. 아이티 공화국의 국가 본점은 아직 존재하지 않을 수 있지만 해당 국가는 존재합니다.



ORACLE!

Copyright © Oracle Corporation, 2002. All rights reserved.

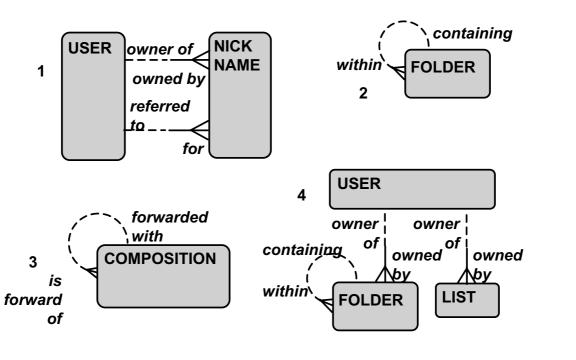
연습 3-6 가격 목록: 해답

Moonlight Coffee Store 중 한 매장의 가격표를 기준으로 ER 모델을 만드십시오.

여기에서 발생되는 질문 중 하나는 PRODUCT가 카페인이 제거되었는가 입니다. 여러 가지 측면에서, 이 제품은 다른 제품과 다르게 작동합니다. 별도로 또는 애플 파이와 같은 제품과 함께 구입할 수 없습니다. 한편 이 제품은 구입할 수 있으며 다른 제품들과 같이 정해진 가격이 있습니다.

이 해답에서 카페인 제거는 보조 식품(더 나은 이름으로 지칭할 수도 있음)과 같은 특수한 제품 그룹의 한 제품으로 간주됩니다. 이 그룹의 제품들은 특별한 주의가 필요하며, 그렇지 않으면 일반적인 제품으로 취급될 수 있습니다. PRODUCT GROUP 엔티티에도 주의하십시오. PRODUCT GROUP 인스턴스는 가격 목록의 다양한 제품들을 구분하는 선으로 표시됩니다.

해답: 전자 메일



Copyright © Oracle Corporation, 2002. All rights reserved.

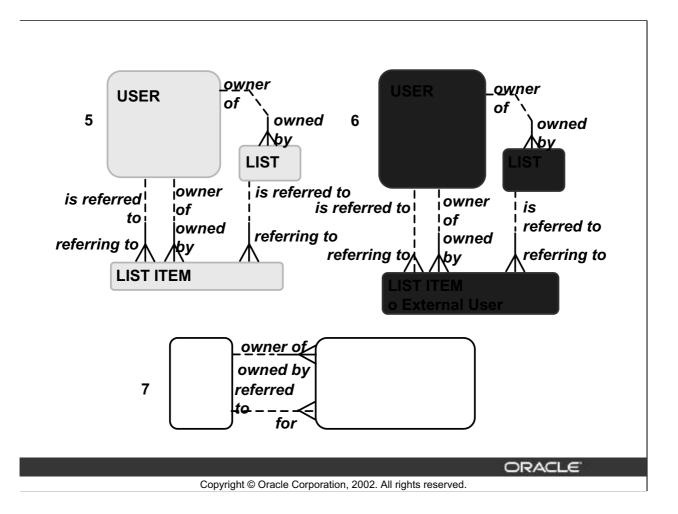
ORACLE

연습 3-7 전자 메일: 해답

제공된 모델을 시작점으로 사용합니다. 다음 기능을 편리하게 사용할 수 있도록 엔티티, 속성 및 관계를 추가, 삭제 또는 변경합니다.

그림에서 번호는 다양한 할당 방식을 나타냅니다.

- 1 사용자는 다른 사용자에 대한 애칭(별칭)을 생성할 수 있어야 합니다.
- 2 폴더는 다른 폴더를 포함할 수 있습니다.
- 3 사용자는 composition을 전달할 수 있어야 합니다. forward는 전달된 메시지와 함께 자동 으로 전송되는 새 메시지입니다.
- 4 모든 폴더와 목록은 사용자가 소유합니다.



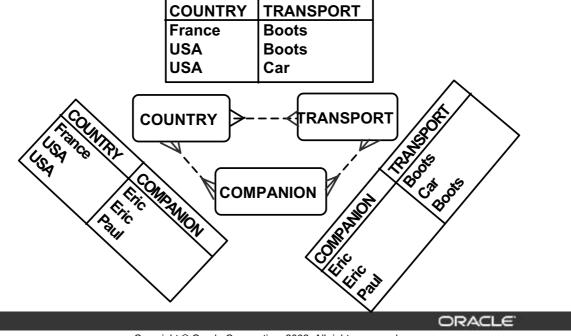
연습 3-7 전자 메일: 해답

해결 과제:

- 5 메일 목록은 사용자와 다른 목록을 모두 포함할 수 있습니다.
- oracle Internal 8. OAI USE 6 메일 목록은 "giovanni_papini@yahoo.com"과 같은 외부 주소를 포함할 수 있습니다
- 7 애칭은 외부 주소에 대한 별칭일 수 있습니다.

해답: 휴일(1)

"Paul과 나는 미국에서 여행을 했습니다. Eric과 나는 프랑스에 서 여행을 하고 작년에 미국에서 차를 한 대 빌렸습니다."



Copyright © Oracle Corporation, 2002. All rights reserved.

연습 3-8 휴일: 해답

Oracle Internal & OAI Use Only 시나리오 텍스트를 기준으로 하는 아래 제공된 모델에 주석을 다십시오.

해답: 휴일(2)

COUNTRY	COMPANION	TRANSPORT	_
France	Eric	Boots V	
France	Eric	Car	_
USA	Eric	Boots	7
USA	Eric	Car 🗸	,
USA	Paul	Boots	/

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

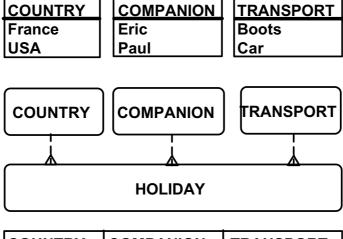
COUNTRY-COMPANION 테이블 및 COMPANION-TRANSPORT 테이블의 정보를 결합하면 결과는 다음과 같이 됩니다.

이것은 명백히 잘못된 것입니다. 이 모델은 입력된 내용 이상의 결과를 파생시킵니다.

그 이유는 무엇일까요? Eric, Hiking, France의 조합이 중요한 것처럼 Paul, Hiking, USA의 조합 또한 중요합니다. 문제가 되는 것은 이 세 값의 조합입니다. 다음 그림에서 해당 모델과 테이 블 구조를 참조하십시오.



COUNTRY

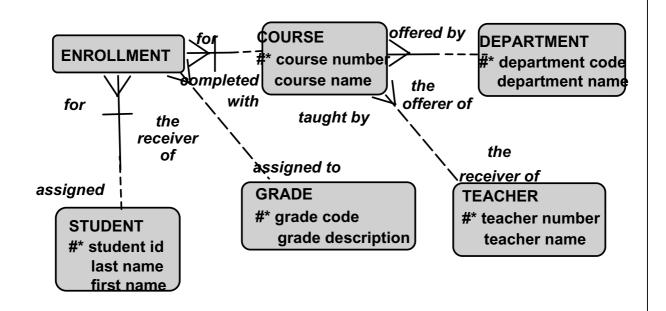


COUNTRY	COMPANION	TRANSPORT
France	Eric	Boots
USA	Eric	Car
USA	Paul	Boots

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

Practice: ER 모델 정규화: 해답



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 3-9: ER 모델 정규화: 해답

목표

이 연습의 목적은 정규화되지 않은 ER 모델을 세번째 정규 폼에 배치하는 것입니다

과제

• 다음 ER 모델에 대해, 정규화 규칙에 따라 각 엔티티를 평가하고, 잘못 배치된 속성을 식별하고, 잘못 배치된 각 속성이 위반하는 정규화 규칙이 무엇인지 설명하십시오.

STUDENT 엔티티

• 이 엔티티는 모든 속성이 제대로 배치되어 있는 상태로 3NF에 있습니다.

COURSE 엔티티

- 이 엔티티는 2NF에만 있습니다. 부서 이름, 부서 코드, 교사 번호 및 교사 이름이 3NF를 위반하며 잘못 배치되었습니다.
- 부서 이름과 부서 코드는 서로 종속됩니다. 이 두 항목을 COURSE와 관계가 있는 별도 의 DEPARTMENT 엔티티로 이동하십시오. 교사 이름과 교사 번호는 서로 종속됩니다. 이 두 항목을 COURSE와 관계가 있는 별도의 TEACHER 엔티티로 이동하십시오.

연습 3-9: ER 모델 정규화: 해답

ENROLLMENT 엔티티

- 이 엔티티는 1NF에만 있습니다. 교사 번호와 과정 이름이 2NF를 위반하며 잘못 배치되 었습니다. 등급 코드와 등급 설명이 3NF를 위반하며 잘못 배치되었습니다.
- 교사 번호와 과정 이름은 UID의 COURSE 부분 집합에만 종속되므로 COURSE 엔티티 로 이동하십시오.
- 등급 설명과 비 UID 속성 등급 코드는 상호 종속됩니다. 두 항목을 ENROLLMENT와 관 계가 있는 GRADE 엔티티로 이동하십시오.
- 세번째 정규화 폼에 ER 도표를 다시 그리십시오. (선택 사항)

목표

이 연습의 목적은 정규화되지 않은 ER 모델을 세번째 정규 폼에 배치하는 것입니다.

과제

• 다음 ER 모델에 대해, 정규화 규칙에 따라 각 엔티티를 평가하고, 잘못 배치된 속성을 식별하고, 잘못 배치된 각 속성이 위반하는 정규화 규칙이 무엇인지 설명하십시오.

STUDENT 엔티티

• 이 엔티티는 모든 속성이 제대로 배치되어 있는 상태로 3NF에 있습니다.

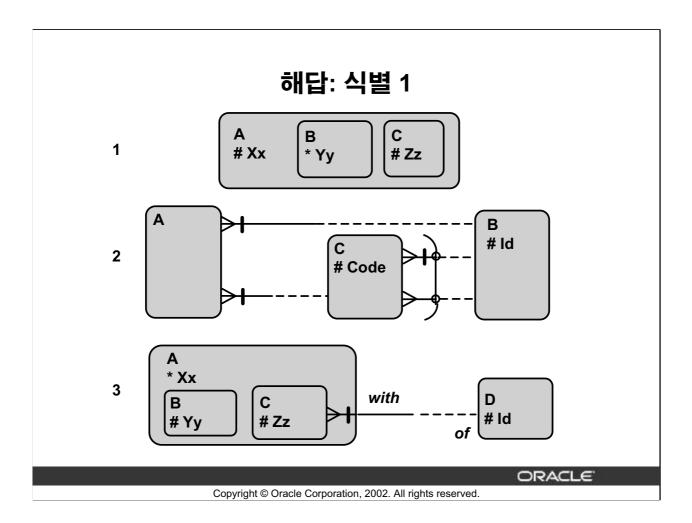
COURSE 엔티티

- 이 엔티티는 2NF에만 있습니다. 부서 이름, 부서 코드, 교사 번호 및 교사 이름이 3NF를 위반하며 잘못 배치되었습니다.
- 부서 이름과 부서 코드는 서로 종속됩니다. 이 두 항목을 COURSE와 관계가 있는 별도 의 DEPARTMENT 엔티티로 이동하십시오. 교사 이름과 교사 번호는 서로 종속됩니다. 이 두 항목을 COURSE와 관계가 있는 별도의 TEACHER 엔티티로 이동하십시오.

ENROLLMENT 엔티티

- 이 에티티는 1NF에만 있습니다. 교사 번호와 과정 이름이 2NF를 위반하며 잘못 배치되 었습니다. 등급 코드와 등급 설명이 3NF를 위반하며 잘못 배치되었습니다.
- 교사 번호와 과정 이름은 UID의 COURSE 부분 집합에만 종속되므로 COURSE 엔티티 로 이동하십시오.
- 등급 설명과 비 UID 속성 등급 코드는 상호 종속됩니다. 두 항목을 ENROLLMENT와 관 계가 있는 GRADE 엔티티로 이동하십시오.

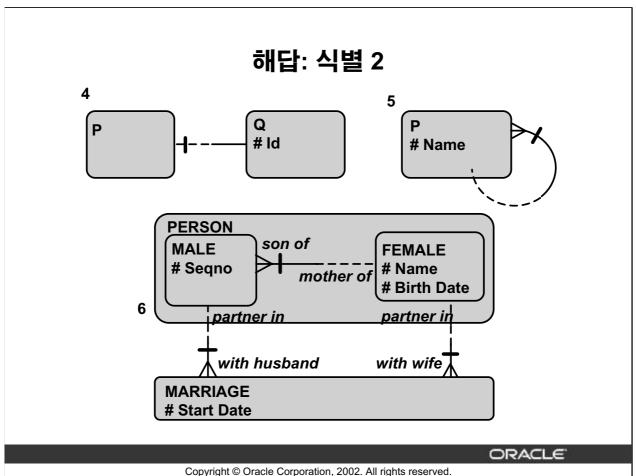
 - 세번째 정규화 폼에 ER 도표를 다시 그리십시오. (선택 사항)



연습 4-2 식별: 해답

다음 도표의 엔티티는 식별 가능합니까?

- 1. 모든 A는 Xx 속성으로 식별 가능하기 때문에 모든 B와 C도 마찬가지입니다.
- 2. B는 ID로 식별할 수 있습니다. A는 B와 C를 식별할 수 있는 경우에만 식별할 수 있습니다. C는 식별되지 않은 상태로 남아 있습니다. 호 구조 때문에, 모든 C가 B와의 세로줄 표시 관계를 갖는 것은 아닙니다. 하지만 이로 인해 C는 식별되지 않은 상태가 됩니다.
- 3. D는 ID로 식별할 수 있습니다. 모든 C는 ZZ 및 D로 식별할 수 있습니다. 모든 B는 Yy로 식별할 수 있습니다. 모든 B와 모든 C를 식별할 수 있으므로 모든 A도 식별할 수 있습니다.



연습 4-2 식별: 해답

- 1. 모든 Q는 ID로 식별할 수 있습니다. Q의 인스턴스와 관련된 P의 인스턴스는 해당 인스턴스에 의해 식별될 수 있지만 모든 P가 관련되어 있는 것은 아닙니다.
- 2. 개념적으로 볼 때 여기에서 문제는 없습니다. 모든 P는 해당 Name 및 "이전 엔티티"에 대한 참조로 식별됩니다. 첫번째 P의 처리 방법에 대해 문제를 제기할 수 있지만, 이 경우에는 먼저 "첫번째 P가 있다는 사실을 어떻게 알 수 있는지"에 대한 질문이 나올 수 있습니다. 더욱 문제가 되는 것은 n번째로 생성된 인스턴스의 식별에는 n+1의 이름을 사용한다는 사실입니다. 이것은 특히 이러한 유형의 식별을 매우 비현실적인 것으로 만듭니다.

주: 다음 모델은 사용자에게 익숙하지 않을 수 있는 상황을 설명합니다.

- 3. FEMALE 엔티티는 Name과 Birth Date에 의해 식별됩니다. 이러한 사실은 전세계 모든 인구에 대해서는 적용되지 않을 수 있지만 업무에 관련된 소규모의 사람들 에게는 적용될 수 있습니다.
- 4. MALE 엔티티는 "나는 내 어머니의 유일한 둘째 아들입니다."로도 식별될 수 있습니다.
- 5. 모든 사람은 남자 아니면 여자이므로 모든 PERSON은 식별이 가능합니다. MARRIAGE 엔티티는 관련된 MALE과 FEMALE의 조합과 결혼 Date에 의해 식별 할 수 있습니다.

연습 4-2 식별: 해답

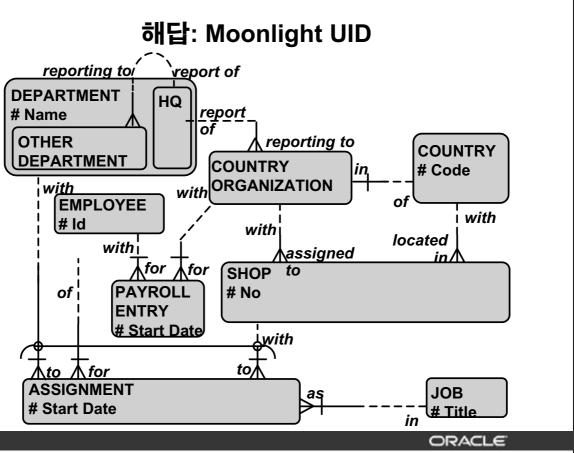
- 7. 위의 모델을 전제로 다음 질문에 답하십시오.
 - a A라는 사람은 두 번 결혼할 수 있습니까?

예

- b A라는 사람은 같은 날에 두 번 결혼할 수 있습니까?
- c A라는 사람은 B라는 사람과 두 번 결혼할 수 있습니까?
- d A라는 사람은 B라는 사람과 같은 날 두 번 결혼할 수 있습니까? 아니오
- e A라는 사람은 B라는 사람 및 C라는 사람과 동시에 결혼할 수 있습니까? 예
- f A라는 사람은 A라는 사람과 결혼할 수 있습니까?

아니오. 결혼은 항상 남자와 여자 사이에 이루어져야 합니다. 이것은 사람의 독특한 하위 유형이므로 A가 남자도 되고 여자도 될 수는 없습니다.





Copyright © Oracle Corporation, 2002. All rights reserved.

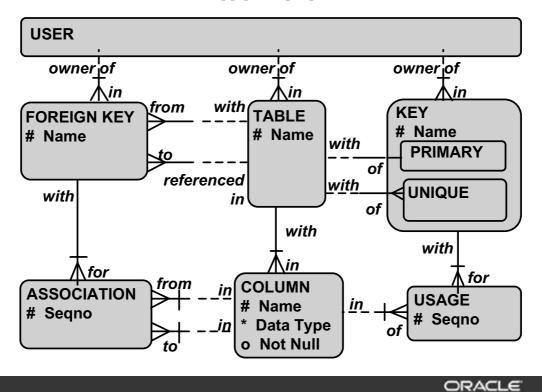
연습 4-3 Moonlight UID: 해답

지금까지 Moonlight Coffees에 대해 알고 있는 지식을 사용하고, 특히 상상력을 동원하십시오.

- 1. 위 모델을 전제로 다양한 엔티티에 대한 UID를 확인하십시오. 적합하다고 간주되는 속성을 모두 추가합니다. 국가 본점은 해당 국가에서 고유한 "사업자 등록 번호"를 가집니다.
- 2. 누락된 호가 있습니까?

투명성을 위해, 식별에 필요한 속성만 이 모델에 추가되었습니다.

해답: 테이블



Copyright © Oracle Corporation, 2002. All rights reserved.

연습 4-4 테이블: 해답

ISO 관계형 테이블의 텍스트를 읽어 보십시오.

따옴표로 묶인 텍스트와 이 주제에 대해 본인이 알고 있는 내용을 기준으로 ER 모델의 품질을 확인하십시오. 또한 텍스트에 언급되어 있지만 모델링되지 않은 제약 조건을 나열하십시오.

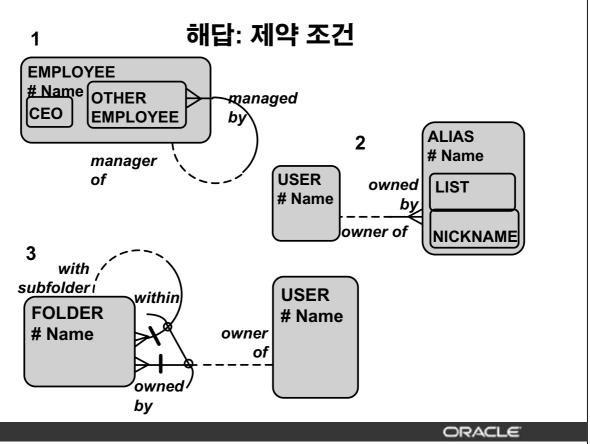
오류

- FOREIGN KEY에서 TABLE로의 두 관계는 세로줄이 표시되면 안됩니다.
- Table은 Name만으로 식별되지 않으며 소유하는 USER에 의해서도 식별됩니다. 동일한 사항이 KEY 및 FOREIGN KEY에도 적용됩니다.
- 모든 KEY, PRIMARY, UNIQUE 및 FOREIGN은 USER마다 고유하게 명명되어야 합니다.

위 도표에서는 이러한 오류가 제거되었습니다.

표시되지 않은 사항

- KEY의 COLUMN은 KEY가 속하는 TABLE의 COLUMN이어야 합니다.
- FOREIGN KEY의 COLUMN은 FOREIGN KEY가 참조하는 TABLE의 KEY에 있는 COLUMN을 참조해야 합니다.
- 이러한 제약 조건은 표시될 수 없으며 따로 나열해야 합니다.

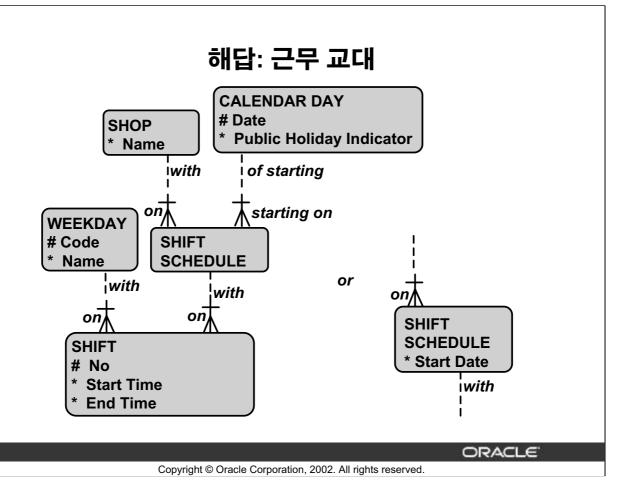


Copyright © Oracle Corporation, 2002. All rights reserved.

연습 4-5 제약 조건: 해답

도표를 변경하여 제공된 제약 조건을 모델링하십시오.

- 1. CEO(최고 경영권자)를 제외한 모든 EMPLOYEE에게는 매니저가 있어야 합니다.
 - EMPLOYEE라는 CEO 하위 유형을 생성하여 이 제약 조건을 쉽게 모델링할 수 있습니다. 이 제약 조건을 모델링하는 이유가 단지 CEO 하위 유형을 존재하도록 하기 위한 것이라면 이 CEO 하위 유형의 사용에 의의를 제기할 수도 있습니다.
- 2. 사용자는 NICKNAME 및 LIST 이름 둘 다에 대해 동일한 이름을 사용하지 않아도 됩니다.
 - LIST와 NICKNAME은 동일한 이름 공간을 공유합니다. 이것은 상위 유형을 사용 하여 모델링될 수 있습니다. Name 속성의 재배치 결과에 대해서도 유의하십시오.
- 3. 최상위 레벨의 FOLDER는 사용자별로 고유한 이름을 가져야 합니다. 하위 폴더는 해당 폴더가 위치하는 폴더 내에서 고유한 이름을 가져야 합니다.
 - 호 및 세로줄 표시 관계를 추가하는 것으로 충분합니다. 많은 끝에서 FOLDER에서 FOLDER로의 순환 관계가 필수적으로 만들어졌습니다. 순환 식별에 대한 연습 4- 식별 5: 해답의 정보도 참조하십시오.

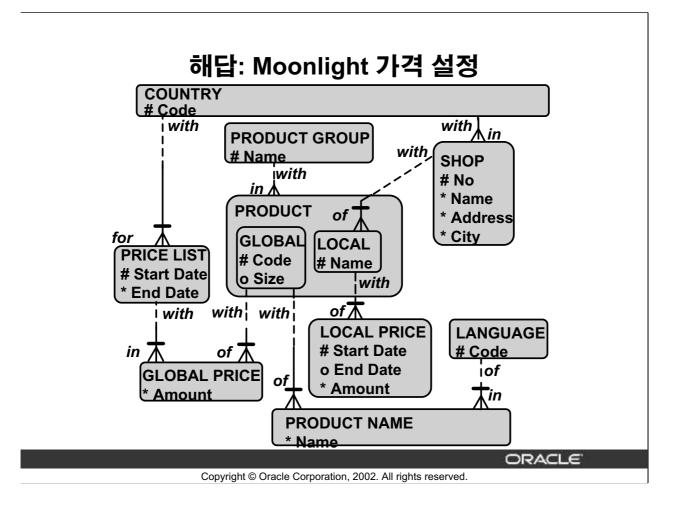


연습 5-1 근무 교대: 해답

이 근무 교대표에서 찾은 다양한 날짜/시간 요소를 나열하고 개념적 데이터 모델을 만드십시

이 해답에서는 CALENDAR DAY 엔티티가 사용됩니다. 이 엔티티는 일반적으로 공휴일이 중요한 영향을 미치는 인력 계획 환경에서 민감할 수 있습니다. 공휴일은 국가별로 완전히 다르다는 사실에 주의하십시오. 즉, 공휴일 표시자는 국제적으로 단일 값이 아닙니다.

글로벌 상황에서는 일광 절약 시간(일반적이지 않으며 전세계적으로 동일한 날짜에 시작되지 않음) 및 TIME ZONE 문제가 영향을 미칩니다.



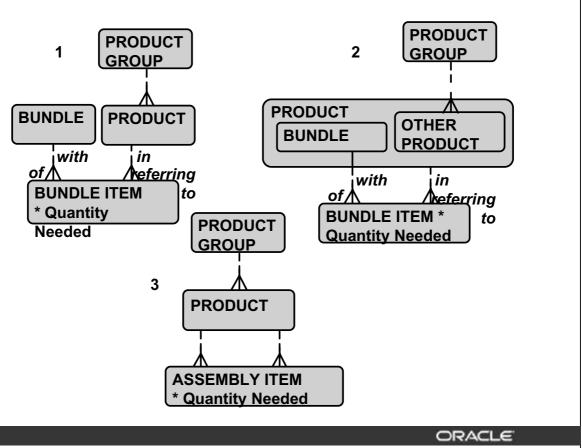
연습 5-2 딸기 와퍼: 해답

다음의 추가 정보가 제공된다고 가정할 경우 모델을 재방문하고 필요에 맞게 변경하십시오.

이 모델에서는 앞에 제시된 모델(연습 3 – 가격 목록: 해답)에 비교해 볼 때 몇 가지 중요한 사항 및 일반적인 사항이 변경되었습니다.

중요한 변경 사항은 국가 레벨 또는 판매점 레벨에서 결정된 가격 문제입니다. 이로 인해 GLOBAL 및 LOCAL PRICE의 두 가지 가격 엔티티가 생성되었습니다. 이러한 두 엔티티의 여러 다른 속성과 관계에 유의하십시오.

또한 가격 목록은 이전 것과 동일한 제품 범위를 나타내지만 현재는 제품을 여러 언어로 명명할 수 있습니다.



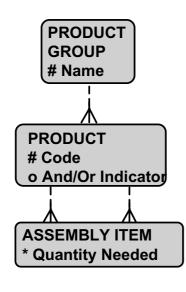
Copyright © Oracle Corporation, 2002. All rights reserved.

연습 5-3 번들: 해답

Oracle

- 1. 원하는 계산을 수행할 수 있는 방식으로 모델의 제품 부분을 수정하십시오.
 - 첫번째 모델이 아마 여러분이 처음 생각해 낸 것일 것입니다. 하나의 번들은 여러 제품으로 구성됩니다. 이 모델은 BUNDLE ITEM 엔티티를 사용하여 모델링한 것입니다. 그러나 번들이 SuperSweetTreat와 같은 다른 번들로 구성될 수 있으므로 이해답은 올바르지 않습니다. 모델 2는 이러한 점을 고려한 것입니다. 이제 BUNDLE ITEM은 하나 이상의 BUNDLE 인스턴스 및 OTHER PRODUCT 인스턴스로 구성될 수 있습니다.
 - 모델 2는 제품과 번들이 하위 유형으로 모델링되므로 이 두 가지 항목이 항상 명확하게 구분된다고 가정합니다. 모델 3은 이러한 사실을 가정하지 않습니다. 여기에서 번들은 임의의 제품으로 취급됩니다. 생크림을 바른 애플 파이를 번들이나 단일 제품 중 어느 것으로 간주해야 할지에 대해서는 걱정할 필요가 없습니다.

DecafPunch (DP) = {Coffee (C) or Tea (T)} and {Blackberry Muffin (BM)}



Consider this as: AS1 = (C or T) DP = (AS1 and BM)

PRODUCTS

Code	And or	Pg name
С		
Т		
AS1	OR	
DP	AND	

ASSEMBLY ITEMS

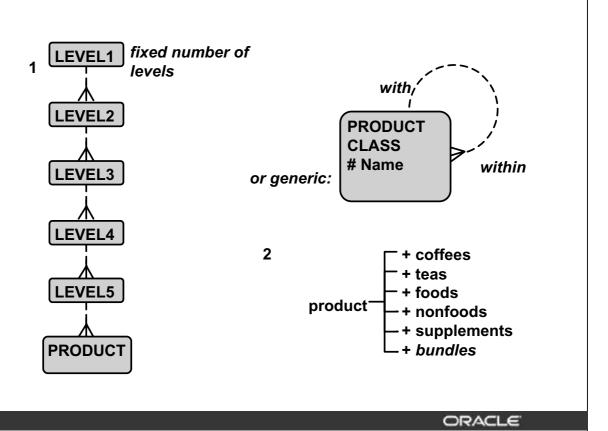
Prod code	Using code	Quantity
AS1	С	1
AS1	Т	1
DP	AS1	1
DP	ВМ	1

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 5-3 번들: 해답

- 2. 모델에 허용되는 방식으로 모델을 변경합니다.
 - 이것은 까다로운 문제입니다. DecafPunch를 가격이 동일한 두 제품 DecafPunch Coffee 및 DecafPunch Tea의 제품 그룹으로 간주할 수 있습니다(or 때문에 가능). 이 러한 관점은 이미 PRODUCT GROUP / PRODUCT 모델에서 다루었습니다.
 - 제시된 해답에서는 이러한 측면이 무시되었습니다. 이 해답은 구성의 and or or 유형을 모델링하는 방법에 초점을 맞추고 있습니다.
 - DecafPunch가 and or or로 연결된 일련의 기본 그룹으로 간주될 수 있음을 알 수 있습니다. 기본 그룹은 단일 제품 또는 and or or로 연결된 제품 쌍으로 구성된 그룹으로 보입니다. 제품은 항상 이러한 방식으로 설명될 수 있습니다. (이것은 수학적 논리에서 파생되었습니다.)
 - 테이블 구조를 살펴 보고 모델의 작동 내용을 확인하십시오.
 - 이 모델의 복잡성은 Moonlight의 상황에서는 별로 적합하지 않습니다. 하지만 이 러한 유형의 모델은 많은 생산 업체에서 대체 구성을 포함하여 제품의 구성 정보를 저장하는 데 사용하는 BOM 구조에 적합합니다.



Copyright © Oracle Corporation, 2002. All rights reserved.

연습 5-4 제품 구조: 해답

Olacle

- 1. 제품 분류 구조에 대한 모델을 생성하십시오.
 - 레벨 번호를 알고 있으며 이 번호가 고정되어 있으면 왼쪽 모델을 사용할 수 있습니다. 이 모델을 사용하면 모든 제품에 대해 강제로 다섯 개의 레벨을 사용해야 합니다. 이를 피하기 위해 전체 간에 호 관계를 형성하는 것은 물론, PRODUCT에서 모든 LEVEL 엔티티로의 관계를 생성할 수 있습니다.
 - 왼쪽 모델은 상당히 자유롭습니다. 최소 레벨 수나 최대 레벨 수의 제한이 없습니다.
 다. 시스템 수명 동안 분류를 자유롭게 변경할 수 있습니다.
 - 2 (선택 사항) 번들 제품을 어떤 방식으로 취급할 것입니까?
 - 번들의 문제점은 엄격히 말해서, 번들이 Drink와 Food 모두로 구성될 수 있으므로 Drink나 Food 중 어느 것으로도 분류할 수 없다는 것입니다. 이러한 문제를 처리하는 가장 쉬운 방법은 번들을 PRODUCT CLASS의 인스턴스로 간주하는 것입니다.

패턴

•	체스 게임에서의 게임 동작 모델	체인 또는 네트워크
•	입찰 모델(견적)	M/D 또는 바구니
•	조리법 모델	ВОМ
•	대학에 속한 모든 사람 모델: 학생, 교사, 학부모	롤
•	비디오 가게에서의 대여	M/D 또는 바구니

ORACLE

체인 또는 네트워크

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-1 패턴: 해답

제공된 컨텍스트에서 어떤 패턴을 기대할 수 있습니까? 기대한 패턴을 찾을 수 없으면 모델을 개략적으로 그려보십시오. 상상력과 상식을 동원하십시오.

체스 게임에서의 게임 동작

프로세스의 단계 모델

MOVE는 체스 하나를 한 FIELD에서 다른 FIELD로 이동합니다. 이것은 일반적인 네트워크 패턴입니다. 또한 MOVE를 체스 게임에서 순서대로 진행된 단계의 체인으로 볼 수도 있습니 다.

견적내기

견적은 특정 수량의 제품을 제공하기 위한 제안서입니다. 이 구조는 ORDER와 유사하며 마 스터-디테일 또는 바구니 패턴을 갖습니다.

조리법은 일반적으로 어떤 것을 만드는 방법을 설명합니다: 예를 들어, ...을(를) 만들려면 두 개의 하나의 ...이(가) 필요하다 등과 같이 말할 수 있습니다. 이것은 일반적인 BOM 패턴입 니다.

패턴

• 체스 게임에서의 게임 동작 모델 체인 또는 네트워크 • 입찰 모델(견적) M/D 또는 바구니

. 조리법 모델

· 프 _ _ 대학에 속한 모든 사람 모델: 학생, 교사, 학부모...

비디오 가게에서의 대여 M/D 또는 바구니

BOM

롴

프로세스의 단계 모델 체인 또는 네트워크

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-1 패턴: 해답

대학에 속한 사람들

패턴: 롤. 교사는 학생의 부모가 될 수도 있고 부모는 이전의 학생일 수도 있습니다.

비디오 가게에서의 대여

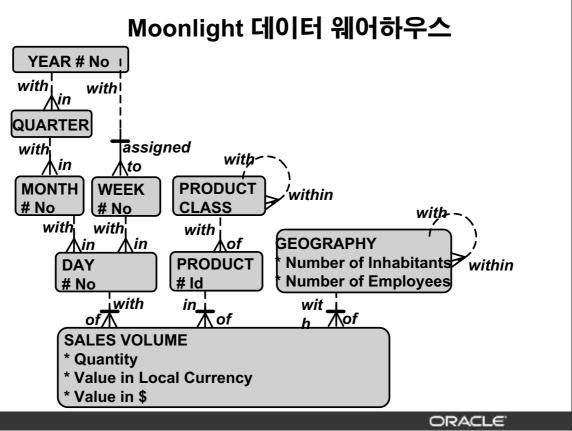
RENTAL은 판매 또는 견적내기(마스터-디테일 또는 바구니 패턴)와 같은 이벤트입니다.

프로세스의 단계

PROCESS와 PHASE는 *마스터-디테일* 패턴을 갖습니다. 이 점에 주의하십시오. 프로세스의 단계에 대한 중요한 사항은 단계가 일반적으로 정렬된다는 것입니다.

특정 단계 앞뒤에 항상 하나의 단계만 있으면 단계는 체인을 형성할 수 있습니다.

한편, 단계 앞뒤에 임의의 단계(자기 자신 포함)가 올 수 있는 경우 구조는 네트워크가 됩니다.

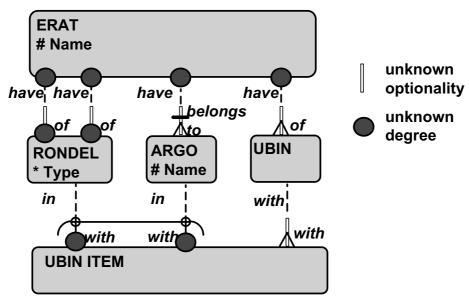


Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-2 데이터 웨어하우스: 해답

- 1. 지금까지 생성한 Moonlight 모델을 확인하십시오. 이러한 모델을 통해 나열된 질문에 충분히 대답할 수 있습니까? 그렇지 않다면 모델을 적절히 변경하십시오.
 - 정해진 해답은 없습니다.
- 2 데이터 웨어하우스 데이터 모델에 대해 중앙의 "facts" 엔티티를 제시하십시오.
 - 이 도표는 연습에서 요구하는 것 이상의 정보를 제공합니다. 중앙 엔티티는 SALES VOLUME입니다.
 - 이것은 스타 모델이 아닌 일반적인 스노우플레이크 모델입니다. 세 개의 차원인 시간, 지리 및 제품은 모두 자체의 내부 구조를 갖습니다.
 - 주는 월에 할당되지 않습니다. 모든 주가 달력 연도에 정확히 일치하는 것은 아니지만 주는 연도에 할당됩니다. 주는 일 및 연도와 같이 고정 길이를 가지지만 천문학적 현상을 기반으로 하지 않으므로 다르게 동작합니다.
 - 앞서 정의한 제품 구조를 여기에서 사용할 수 있습니다. 로컬 제품과 글로벌 제품 간의 구분은 사라졌습니다.
 - 지리는 중요한 영역(예: 국가의 일부 또는 국가나 대륙)에 대한 일반적인 단어입니다.

해답: Argos 및 Erats



제약 조건은 표시되지 않음: 하나의 ubin은 항상 *erat*의 하나 이상의 argos와 하나 이상의 ...(으)로 구성됩니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 6-3 Argo 및 Erat: 해답

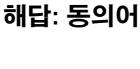
텍스트의 정보를 기준으로 개념적 데이터 모델을 만드십시오. 도표에서 텍스트를 통해 확인할 수 있는 모든 부분에 표시하십시오.

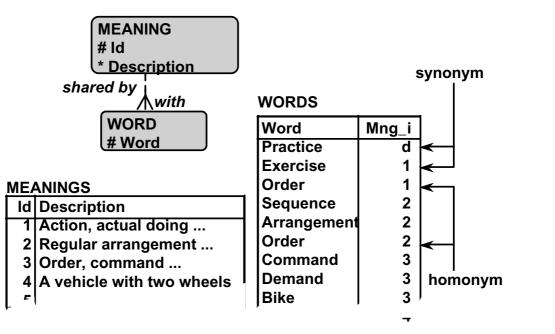
제공된 텍스트에 필요한 모든 정보가 있는 것은 아닙니다. 실제로 확인해야 할 정보의 양이 놀랄 만큼 많을 수 있습니다.

제공된 "하나의 ubin은 항상 하나 이상의 ...(으)로 구성됩니다"를 모델링하기 위해 확신할 수 있는 유일한 부분은 UBIN - UBIN ITEM입니다. 이것이 일반적인 바구니 구성입니다.

주

이 텍스트는 일반 텍스트에서 찾기/바꾸기 명령을 사용하여 생성되었습니다. 강사에게 원본 텍스트를 물어 보십시오.





Copyright © Oracle Corporation, 2002. All rights reserved.

ORACLE!

연습 6-4 동의어: 해답

동의어 사전의 기초가 될 수 있는 개념적 데이터 모델을 만드십시오.

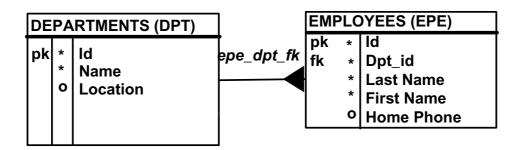
이 해답은 놀랍고 당혹스러울 정도로 쉬울 것입니다.

동의어를 동일한 의미를 공유하는 단어로 간주하십시오. *의미*라는 단어와 *단어*라는 개념을 혼동하지 마십시오. 하나의 단어 또는 여러 개의 단어를 사용하여 하나의 의미를 나타내지만 이것은 하나의 단어처럼 동일하지 않습니다. 단어는 의미의 표현입니다.

SYNONYM 엔티티가 없다는 점에 유의하십시오. 동의어는 상황의 영향을 받습니다. 둘 이상의 단어로 공유되는 하나의 MEANING이 있는 경우 동의어에 해당합니다.

반대로, 둘 이상의 MEANING을 갖는 하나의 WORD가 있는 경우 이것은 동음 이의어입니다.

해답: 기본 엔티티, 속성 및 관계 매핑



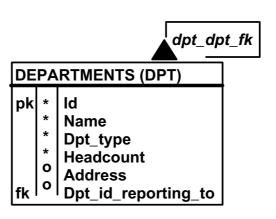
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-1 기본 엔티티, 속성 및 관계 매핑: 해답

테이블 도표에서 이 구현 다음에 생성해야 하는 모든 요소의 이름을 지정하십시오. 이 단원에서 설명하는 대로 이름 지정 규칙을 사용하거나 사용자 자체 규칙을 사용하십시오. 열 및 외래 키 제약 조건에 적절한 이름을 지정하십시오.

해답: 상위 유형 매핑



ORACLE

only

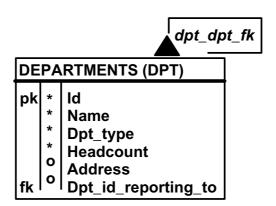
Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-2 상위 유형 매핑: 해답

Olacle

- 1 이렇게 선택한 이유는 무엇입니까?
 - 가능한 고려 사항은 다음과 같습니다.
 - 부서의 수는 별로 크지 않을 것이며 대개 100개 정도일 것입니다.
 - 하위 유형에 대해 소수의 특정 속성만 있습니다.
 - 한 가지 하위 유형인 HQ는 하나의 인스턴스만 가질 수 있습니다.
 - HQ는 HQ를 제외한 모든 부서에서 한 부서로 보고해야 함을 나타내는 방식으로만 모델링된 것 같습니다.
- 2 테이블 도표에서 이 상위 유형을 구현한 후에 생성해야 하는 모든 요소의 이름을 지정하십시오. 이 단원에서 설명하는 대로 이름 지정 규칙을 사용하거나 사용자 자체 규칙을 사용하십시오. 열 및 외래 키 제약 조건에 적절한 이름을 지정하고 검사 제약 조건이 있는 경우 식별하십시오.

해답: 상위 유형 매핑



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-2 상위 유형 매핑: 해답

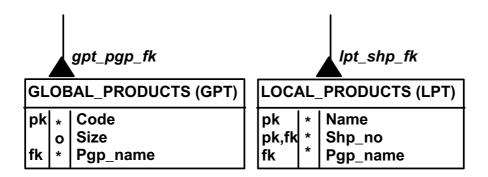
선택적인 Address 열에 유의하십시오. 선택적인 외래 키 열과 해당 이름에 대해서도 유의하십 시오. 규칙에 따르면 이름은 Dpt_id가 될 수 있습니다. 많은 사람들이 이 이름을 기본 키 열 이 름으로 가정할 수 있으므로 이 이름 뒤에 외래 키가 나타내는 관계의 이름이 나옵니다. Al USE

다음과 같은 두 개의 제약 조건이 필요합니다.

하위 유형을 보호하려면 다음과 같이 지정합니다.

```
(
         Dpt type = 'HQ'
  and Address is not null
  and Dpt id reporting to is null)
또는
        Dpt type <> 'HQ'
  and Address is null
  and Dpt id reporting to is not null)
```

해답: 하위 유형 구현 품질 검사



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-3 하위 유형 구현 품질 검사: 해답

". Use only PRODUCT 엔티티의 제안된 하위 유형 구현에 대해 품질 검사을 수행하십시오.

오류:

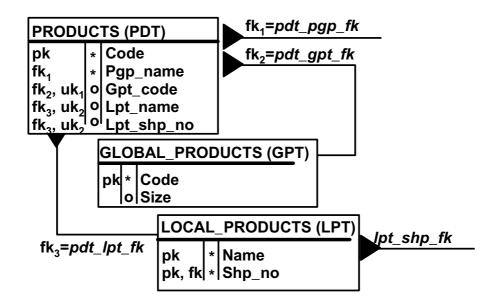
GLOBAL PRODUCTS

• Pgp name 외래 키 열이 fk로 표시되어 있지 않습니다.

LOCAL PRODUCTS

- Shop_no 열의 이름은 Shp_no여야 합니다.
- Shp no 열 또한 기본 키의 일부입니다.
- Oracle Intern SHOPS 테이블에 대한 외래 키 이름은 lpt_shp_fk여야 합니다.

해답: 상위 유형 및 하위 유형 구현의 품질 검사



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-4 호 구현 품질 검사: 해답

PRODUCT 엔티티 및 해당 하위 유형의 제안된 상위 유형 및 하위 유형 구현에 대해 품질 검사 ise on 를 수행하십시오. 또한 선택된 이름을 확인하십시오.

오류:

PRODUCTS

- Gpt_code와 Lpt_name은 모두 고유키 uk1의 일부입니다.
- 외래 키 이름의 마지막 문자는 모두 fk여야 합니다.
- LOCAL PRODUCTS의 기본 키가 두 개의 열로 구성되어 있어서 fk 열이 누락되었 습니다. Lpt_shp_no 열을 추가해야 합니다. 이 열은 fk3의 일부이며 uk2의 일부입 니다.

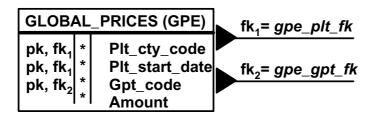
GLOBAL PRODUCTS

- 이것은 사용하기 부적절합니다.
- GLOBAL PRODUCTS가 PRODUCT GROUPS에 대한 자체의 외래 키를 갖지 않으므로 도표의 외래 키 행은 제거해야 합니다.

LOCAL PRODUCTS

Pgp name 열은 제거해야 합니다. Shp n

해답: 기본 키 및 열



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 7-5 기본 키 및 열: 해답

GLOBAL PRICE 엔티티를 변형하여 만들어지는 기본 키 열 및 이름을 식별하십시오. 간단한 이름을 지정합니다.

GLOBAL_PRICES에는 해당 기본 키의 일부인 세 개의 열이 있습니다. 이러한 모든 열 또한 외래 키의 일부입니다. 처음 두 열은 PRICE_LISTS에서 가져온 것이며 Pct_ 접두어가 붙습니 다. 세번째 열은 GLOBAL_PRODUCTS에서 가져온 것이며 Gpt_ 접두어가 붙습니다.

Pct_cty_code 열 이름은 자기 자신이 외래 키 열인 PRICELISTS의 Cty_code를 참조하므로 접두어가 이중으로 붙습니다.

연습 8-1 비정규화 이름 지정: 해답

다음 테이블 도표의 경우 사용된 비정규화 유형을 판단하고 도표가 사용자가 나열한 비정규화를 나타내는 이유를 설명합니다.

다음 중 하나를 사용하십시오.

- 테이블 사전 조인
- 하드 코딩된 값
- 마스터로 디테일 유지
- 마스터에 단일 디테일 반복
- 단락 키

유형 – 테이블 사전 조인

이유: WEEKDAYS 테이블의 Name 열이 Wdy_name 열로서 SHIFTS 테이블에 추가 되었습니다.

유형 - 단락키

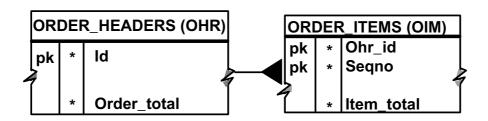
이유: PROD_GRPS 테이블에 조인하기 위해 PROD_NAMES 테이블에 새로운 외래 키제약 조건과 열이 추가되었습니다.

유형 - 올바르지 않음(End Date 열 또는 현재 표시자일 수 있음)

이유: 날짜 검색이나 삽입과 관련된 문제를 용이하게 하기 위해 테이블에 열을 추가할 때 end-date 열 또는 현재 표시자 열 중에서 하나만 추가합니다.



해답: 트리거(1/6)



ORACLE

- 내답

 DER_HEADERS의 비정규화된 Order_total 열을 처리하는 데 필요리거가 어떤 작업을 수행하는지 확인하십시오.

 ORDER_HEADERS에 대해 Order_total을 처리하는 방법 1. ORDER HEADERS의 비정규화된 Order total 열을 처리하는 데 필요한 트리거와 이 트

해답: 트리거(2/6)

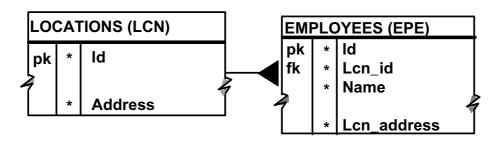
Table	Trg Type	Column	Needed?	What should it do?
OHR	Insert		Υ	Order_total := 0
	Delete		N	
	Update	ld	N	
		Order_total	Υ	prevent update
OIM	Insert		Υ	recalculate Order_total
	Delete		Y	recalculate Order_total
	Update	Ohr_id	Υ	recalculate Order_total
		Item_total	Υ	recalculate Order_total

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



해답: 트리거(3/6)



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-2 트리거: 해답

- uternal & OAL USE OAL OAL 2 EMPLOYEES의 비정규화된 Lcn address 열을 처리하는 데 필요한 트리거와 이 트리거

해답: 트리거(4/6)

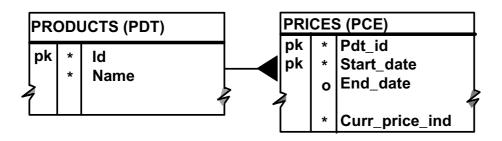
Table	Trg Type	Column	Needed?	What should it do?
LCN	Insert		N	
	Delete		N	
	Update	Address	Υ	Cascade to Employees
		other cols	Y	If pk updated than extended cascade
EPE	Insert		Υ	Set correct Lcn_address
	Delete		N	
	Update	Lcn_id	Y	Set correct Lcn_address
		Lcn_address	Υ	Prevent update

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-2 트리거: 해답

- ea한t
 oracle Internal & OAI USE PRICES 테이블의 비정규화된 Curr price ind 열을 처리하는 데 필요한 트리거와 이 트리

해답: 트리거(5/6)



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



해답: 트리거(6/6)

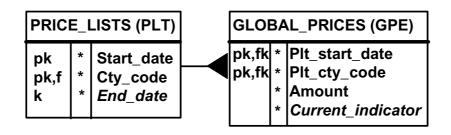
Table	Trg Type	Column	Needed?	What should it do?
PDT	Insert		N	
	Delete		N	
PCE	Insert		Υ	Prevent overlap in price periods
	Delete		N	
	Update	Pdt_id	Υ	Set Curr_price_ind to NULL
		Start_date	Υ	Re-evaluate Curr_price_ind
		End_date	Υ	Re-evaluate Curr_price_ind
		Curr_price_Ind	Υ	Prevent update by user

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

이 해결 방법은 불완전합니다. Curr_price_ind가 실제로 Start_date≤Sysdate≤End_date인 가격에 대해 설정되었다는 보장은 없습니다. 이를 위해 현재 상태로 표시된 가격의 End_date가 만기 되었는지 하루에 한 번 검사하는 타이머를 설정해야 합니다.

해답: 가격 목록 비정규화



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-3 가격 목록 비정규화: 해답

구현해야 할 비정규화 유형과 데이터베이스의 무결성 유지를 위해 추가해야 하는 코드를 설명하십시오. 다음 도표는 현재의 테이블 스키마를 보여줍니다. 구현할 비정규화 유형을 결정할 때 위에서 설명한 두 가지 문제점을 고려하십시오.

두 가지 비정규화 기법을 사용하여 설계를 처리할 수 있습니다. 성능 문제를 위해서는 *현재 열 표시자* 기법을 사용하고, 매니저가 가격 목록을 미리 입력할 수 있게 하려는 경우에는 *End Date* 열을 사용할 수 있습니다.

느린 성능

GLOBAL_PRICES 테이블에 비정규화된 열을 추가하여 조회에 대한 성능을 높일 수 있습니다. 이 테이블은 극히 일부만 현재 데이터를 포함하는 대량의 레코드를 저장하게 되므로 테이블에 Current indicator 열을 추가하는 것을 고려할 수 있습니다.

열을 올바르게 채우기 위해 일부 추가 코드가 요구됩니다.

연습 8-3 가격 목록 비정규화: 해답

가격 목록 미리 입력

추가해야 하는 코드는 PRICE_LISTS 테이블에서 Start_date 열을 읽고 현재의 금액을 반영하도록 현재 레코드 표시자를 설정해야 합니다. 이 코드는 현재의 레코드 표시자를 채웁니다. 하지만 이 작업이 언제 발생할까요? 제품의 가격이 하루에 한 번만 변경될 수 있다고 가정해 봅시다. 이 시나리오에서는 간단히 24시간용 타이머를 설정하고 이 타이머가 만기되면 가격 목록이 변경되었을 때 현재의 레코드 표시자를 업데이트하도록 해당 코드를 발생시킬 수 있습니다.

발효일이 되기 전에 가격 목록 정보를 미리 입력하기 위해서는 또 다른 유형의 비정규화가 필 요합니다

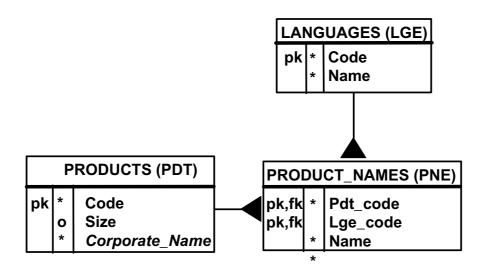
가격 목록이 변경될 때까지 기다리지 않는 회사 정책을 구현하려면 유사하지만 다른 유형의비정규화 작업이 필요합니다. PRICE_LISTS 테이블에 End Date 열을 추가해야 합니다. 이 새열을 추가하는 경우 이 열을 사용하여 가격의 만기일에 도달하기 전에 가격의 만기 날짜를 표시할 수 있습니다. 따라서 가격의 만기일을 다음 순서의 가격에 대한 시작일과 동일하게 계산하지 않고, 원본 레코드를 삽입할 때 시작일, 가격 및 종료일을 입력할 수 있습니다. 레코드가삽입될 때 종료일을 알 수 없는 일부 경우에는 해당 열을 선택적 열로 지정하게 됩니다.

추가해야 하는 코드는 가격 목록이 삽입 또는 업데이트될 때마다 실행되어야 합니다. 이 코드는 현재 가격 목록의 시작일을 읽은 다음 이전 가격 목록 레코드의 종료일을 업데이트합니다. DML 중에 약간의 손해는 있지만 이것은 최소한의 손해입니다.

아래 도표는 두 가지 비정규화 기법이 구현된 후의 테이블 도표를 보여줍니다.



해답: 전역 이름 지정 비정규화



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 8-4 전역 이름 지정: 해답

아래의 설계를 사용하여 테이블 설계를 비정규화하고 이러한 요구 사항을 구현할 수 있도록 하는 추가 코드를 설명하십시오.

이러한 요구에 대한 한 가지 해결 방법으로 PRODUCT에 대한 Name을 English 언어로 PRODUCTS 테이블의 비정규화된 새로운 열에 저장할 수 있습니다. 그런 후 이 Corporate name 열은 모든 전역 제품에 대한 회사 이름을 포함할 수 있습니다.

또한 PRODUCT_NAMES 테이블에 삽입된 데이터가 있고 언어로 English가 사용되는 PRODUCT_NAMES 테이블에서 Name 열을 변경한 경우 PRODUCTS 테이블의 Corporate name 열을 업데이트하기 위한 응용 프로그램 코드를 추가해야 합니다.

아래 도표는 테이블 설계가 구성되는 방식을 보여줍니다.

해답: 데이터 유형(1)

Table	Column	Suggested Data Type	Your Choice Data Type?
COUNTRIES CURRENCIES EXCHANGE_RATES PRICE_LISTS PRODUCT_GROUPS PRODUCTS	Code Code Month Rate Start_date End_date Name Code Size Pdt_type	Varchar2(2) Varchar2(3) Date Number(8,4) Date Date Char(8) Char(10) Number(4,2) Number(1)	Char(2) Char(3) Number(2) Number(12,4) Date Date Varchar2(15) Varchar2(15) Char(1) Char(3)

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 9-1 데이터 유형: 해답

oracle Internal 8. OAI USE 1 다음은 테이블 이름 및 열 이름과 제안된 데이터 유형입니다. 이러한 항목에 대해 품질



해답: 데이터 유형(2)

Table	Column	Your Choice Data Type
GLOBAL_PRICES LOCAL_PRICES SHOPS	Amount Start_date End_date Amount Name Address City	Number(15,3) Date Date Number(15,3) Varchar2(50) Varchar2(50) Varchar2(50)

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 9-1 데이터 유형: 해답

국가 및 통화에 대한 코드와 같은 고정 길이 값은 Char 데이터 유형을 사용하여 저장하는 것이 가장 좋습니다. "Month"는 날짜가 아니며 숫자를 사용하여 저장하는 것이 바람직하며 정렬도 가능합니다. 일반적으로는 긴 문자 이름 및 기타 값을 허용하는 데 주저하지 마십시오 너무 데이터를 제한하는 경우에는 나중에 후회할 수도 있습니다.

- 1. 다음 열에 대한 데이터 유형을 제안하십시오. 열들은 모두 이전 연습을 기반으로 합니다.
- 2. 시간만 포함하는 열에는 어떤 데이터 유형을 사용합니까?
 - SHIFT 엔티티의 Start_time과 End_time 속성과 같은 시간은 여러 가지 방법으로 구현될 수 있습니다. 한 가지 방법은 08:55와 같이 CHAR(5)를 사용하여 시간을 기록하는 것입니다. 계산이 필요한 경우에는 자주, 시간에 대한 열과 분에 대한 열의 두개의 Number(2,0) 열을 사용하여 구현되기도 합니다. 세 번째 방법은 01/01/01과 같은 고정된 날짜로 DATE 열에 시간을 저장하는 것입니다. 하지만 이 방법은 나중에 고려하는 것이 좋습니다.
 - 대부분의 국가에서는 지역 통화의 화폐 금액에 대해 소수점 두 자리 수를 사용합니다. 다른 국가에서는 소수점을 사용하지 않습니다. 지금까지 알려진 바로는 한국가에서만 소수점 세 자리 수를 사용하고 있습니다.

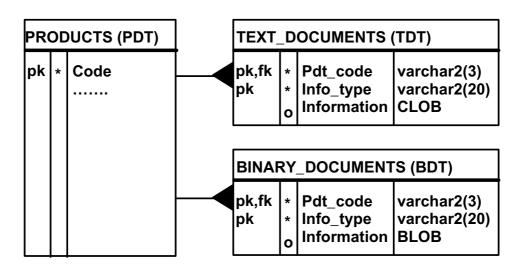
연습 9-2 인공 키: 해답

각 테이블에 대해 인공 키를 생성하면 유용하다고 생각되는지 지적하고 그 이유를 설명하십 시오.

- COUNTRIES
- GLOBAL PRICES
- PRICE LISTS
- COUNTRIES는 국제적으로 사용되는 세 문자 코드를 가지며 이 코드는 기본 키로 사용될 수 있습니다. 이 코드는 약어와 같으며 기억하기 쉽습니다. 국가가 자체적으로 이름을 변경하면 어떤 결과가 발생하는지 정확히 알 수 없습니다. 이 경우 새코드가 생성될 수 있으며 인공 키가 될 수 있는 가장 강력한 후보는 아닙니다.
- GLOBAL_PRICES는 현재 이것을 참조하는 테이블이 없으므로 인공 키가 필요하지 않습니다. 그러나 앞으로는 참조가 있을 가능성이 있으므로 인공 키가 필요할수 있습니다.
- PRICE_LISTS는 UID가 두 구성요소로 구성되며 이 중 하나가 날짜이므로 인공 키에 대한 적절한 후보인 것처럼 보입니다.
- Moonlight 모델을 기반으로 하는 테이블(있는 경우)에 인공 키를 생성하는 것이 적합하지 않은 테이블은 무엇입니까?
- EXCHANGE_RATES, GLOBAL_PRICES(위 참조), LOCAL_PRICES, PRODUCT_NAMES 테이블의 인공 키는 이러한 테이블의 기본 키가 다른(앞으로 사용될) 테이블에서 참조되었을 때만 적절한 대안이 될 수 있습니다.



해답: 제품 그림



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

연습 9-3 제품 그림: 해답

- 1 각 열에 사용하면 좋을 데이터 유형을 결정하십시오.
 - 각 열에 대해 어떤 데이터 유형을 사용할 것입니까?
 - Picture LONG RAW
 - Html doc LONG

Olacle

2 사용자는 이전 Oracle 버전에서는 테이블당 두 개 이상의 long 유형 열을 허용하지 않는 다는 사실을 알고 있습니다. 아직도 이러한 제한이 적용되는지는 정확히 모릅니다. 구현에 대한 의견을 제시하십시오.

구현에 대한 의견을 제시하십시오. 테이블 구조의 도표를 참조하십시오. 한 제품에 대해 여러 개의 LONG 열이 있는 하나의 테이블이 있을 수 있습니다. LONG RAW 열에 대한 두번째 테이블이 있어야 합니다. 실제로 이 모델은 제품 레코드를 세로로 분할한 것입니다. 또한테이블에 둘 이상의 long 유형 열이 허용되면 적절한 구조입니다. 작은 열에서 큰 열을 분리하면 읽기 속도가 크게 향상됩니다.



Oracle Internal & OAI Use Only

개요

- 테이블 정규화
- 테이블의 정규 폼

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

소개

단원 목적

고배치 용 Oracle Internal B. 이 단원에서는 전체 데이터 분석을 수행할 가능성이 없을 때 테이블 데이터를 세번째 정규 폼 으로 정규화하기 위해 필요한 단계에 대해 설명합니다.

테이블

목표

- 이 단원을 마치면 다음을 수행할 수 있습니다.
 - 정규화 정의 및 그 이점 설명
 - 테이블을 세번째 정규 폼으로 배치

정규화 및 이점

테이블을 정규화하는 경우와 그 이유는 무엇입니까?

정규화의 이유를 논의하기에 앞서 우선 정규화해야 하는 경우를 고려해 봅시다. 응용 프로그 램을 개발하는 중에 엔티티 관계(ER) 모델링 기법을 사용하는 경우 정규화가 필요하지 않습 니다. 엔티티 관계 모델링의 이점 중 하나는 ER 모델에 명백한 오류가 없다면 그 결과로 생성 되는 테이블 설계가 이미 정규화되어 있다는 것입니다.

데이터를 정규화해야 하는 유일한 경우는 엔티티 모델을 구축할 시간이 없으며 이미 일부 테 이블을 사용할 수 있는 경우입니다. 이때 기존 데이터베이스 무결성을 확인하기 위한 마지막 기회로서, 초기 데이터베이스 설계 이후에 정규화 기법을 사용할 수 있습니다.

정규화의 역사

정규화는 관계형 모델의 창시자인 E.F. Codd에 의해 확립된 기법입니다. 전체 정규화 기법에 는 진정한 정규화를 위해 데이터베이스가 따라야 하는 12가지 규칙이 포함되어 있습니다. 이 것은 에티티 관계 모델링이 개발되기 수년 전에 관계형 이론을 토대로 생성된 기법입니다. 엔 티티 관계 모델링 프로세스에는 정규화된 엔티티 관계 도표를 생성하기 위한 많은 정규화 기 법이 통합되었습니다.

정규화 기법에 기원을 두고 있는 두 가지 용어가 여전히 널리 사용되고 있는데, 바로 정규화된 데이터와 비정규화입니다.

정규화의 목적

정규화의 주요 목적은 기존 테이블 세트나 테이블 정의에서 중복된 데이터를 제거함으로써 데이터베이스 설계의 무결성을 높이고 데이터 저장의 유연성을 극대화하는 것입니다. 중복된 데이터를 제거하면 업데이트 이상 현상을 없애는 데 도움이 됩니다. 처음 세 개의 정규 폼은 이러한 목적을 달성하도록 체계적으로 발전되고 있습니다.

이 세 개의 정규 폼 외에 많은 다른 정규 폼이 있으며 이러한 정규 폼은 좀더 미묘한 이상 현상 을 처리합니다. 일반적으로 IT 업계에서는 세번째 폼으로의 정규화를 중복성 제거에 합당한 oracle Internal 8. OAI Use 레벨로 간주합니다. 몇 가지 예외를 제외하고, 더 높은 정규화 레벨을 사용하는 경우는 흔치 않습니다.

정규화의 주요 대상은 엔티티가 아니라 테이블입니다.



왜 정규화를 합니까?

- 엔티티 모델을 설계의 시작점으로 항상 사용할 수 있는 것은 아닙니다.
- 기존 설계에서 중복 데이터를 줄일 수 있습니다.
- 데이터의 무결성과 설계의 안정성을 높일 수 있습니다.
- 누락된 테이블, 열 및 제약 조건을 식별할 수 있습니다.

주: 세번째 정규 폼은 중복성을 제거하는 데이터베이스 설계를 위해 일반적으로 적용되는 목표입니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

정규화 및 이점

정규화된 데이터와 정규화 비교

정규화된 데이터는 중복된 내용을 포함하지 않는 데이터입니다. 데이터의 중복성은 무결성 문제를 유발할 수 있기 때문에 이는 중요한 사항입니다. 정규화는 엔티티 관계 모델링과 마찬 가지로 정규화된 데이터 구조를 가져오는 작업이자 프로세스입니다.

정규화된 데이터의 이젂

정보 시스템 관점에서 볼 때 올바르게 정규화된 데이터베이스의 주요 이점은 다음과 같습니다.

- 테이블을 생성하고 키를 선택하기 위한 전략의 구체화
- 일반 사용자의 응용 프로그램 작업과의 커뮤니케이션 기능 향상
- 데이터 삽입 및 삭제와 관련된 문제점 감소
- 데이터 구조 변경과 관련된 개선 및 수정 작업 시간 단축
- 물리적 데이터베이스 설계와 관련된 의사 결정에 사용할 수 있는 정보 향상
- 분석 중 간과될 수 있는 잠재적 문제점 식별

정규화되지 않은 데이터 인식

USER	USER	MSE	REC_			SRVR	SERVER
_ID	_NAME	_ID	DATE	SUBJECT	TEXT	_ID	_NAME
2301	Smith	54101	05/07	Meeting Today	There is	3786	IMAP05
2301	Smith	54098	07/12	Promotions	I like to.	3786	IMAP05
2301	Smith	54445	10/06	Next Assignment	Your next.	3786	IMAP05
5607	Jones	54101	05/07	Meeting Today	There is	6001	IMAP08
			•	Lunch?	Can you		
5607	Jones	54660	12/01	Jogging Today?	Can you	6001	IMAP08
7773	Walsh	54101	05/07	Meeting Today	There is	9988	EMEA01
7773	Walsh	54554	03/17	Stock Quote	The latest	9988	EMEA01
0022	Patel	54101	05/07	Meeting Today	There is	2201	EMEA09
0022	Patel	54512	06/07	Lunch?	Can we	2201	EMEA09

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

정규화되지 않은 데이터

"정규화"를 하지 않은 데이터는 "정규화되지 않은" 데이터 또는 0번 정규 폼으로 간주됩니다. 이 데이터를 비정규화된 데이터와 혼동하면 안됩니다. 데이터베이스 설계 프로젝트 초기에 ER 모델을 생성하지 않은 경우에는 비정규화된 데이터가 아닌 정규화되지 않은 데이터가 있 을 수 있습니다. 빠른 성능이나 기타 이유로 중복성을 추가하려면 비정규화 프로세스 중에 정 의된 규칙을 따르십시오. 하지만 데이터를 비정규화하려면 정규화된 데이터로 작업을 시작해 야 합니다. 얼음이 없는 차에서 얼음을 제거할 수 없는 것처럼 정규화되지 않은 설계를 비정 규화할 수 없습니다.

정규화 규칙

정규 폼 규칙	설명
첫번째 정규 폼(1NF)	테이블은 정렬되지 않은 일련의 2차원 테이블로 표시되어야 합니다. 테이블은 반복 그룹을 포함할 수 없습니다.
두번째 정규 폼(2NF)	테이블은 1NF 형태여야 합니다. 키가 아닌 모든 열은 기본 키의 모든 부분에종속되어야 합니다.
세번째 정규 폼(3NF)	테이블은 2NF 형태여야 합니다. 키가 아닌 열은 키가 아닌 다른 열에 기능적으로 종속될 수 없습니다.

"기본 키가 아닌 각 키 값은 전적으로 해당 키에만 종속되어야 합니다".

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

정규화

정규화는 정규화되지 않은 데이터 세트에서 정규화된 구조로 이동하기 위해 적용되어야 하는 일련의 규칙으로 구성됩니다. 이 프로세스는 "고급" 레벨의 정규화를 수행하는 다양한 단계 에 따라 설명됩니다. 이러한 레벨을 정규 폼이라고 합니다.



첫번째 정규 폼으로 변환

USER	USER	MSE	REC_			SRVR	SERVER
_ID	_NAME	_ID	DATE	SUBJECT	TEXT	_ID	_NAME
2301	Smith	54101	05/07	Meeting Today	There is	3786	IMAP05
2301	Smith	54098	07/12	Promotions	I like to.	3786	IMAP05
2301	Smith	54445	10/06	Next Assignment	Your next.	3786	IMAP05
5607	Jones	54512	06/07	Lunch?	Can you	6001	IMAP08
5607	Jones	54101	05/07	Meeting Today	There is	6001	IMAP08
5607	Jones	54660	12/01	Jogging Today?	Can you	6001	IMAP08
7773	Walsh	54101	05/07	Meeting Today	There is	9988	EMEA01
7773	Walsh	54554	03/17	Stock Quote	The latest	9988	EMEA01
0022	Patel	54101	05/07	Meeting Today	There is	9988	EMEA01
0022	Patel	54512	06/07	Lunch?	Can we	9988	EMEA01

- 1. 기본 테이블에서 반복 그룹을 제거합니다.
- 2. 기본 테이블 및 반복 그룹의 PK로 새 테이블을생성합니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

첫번째 정규 폼

첫번째 정규 폼(1NF)의 정의

테이블은 정렬되지 않은 일련의 2차원 테이블 구조로 표시되어야 합니다. 테이블에 반복 그룹 이 없으면 해당 테이블이 첫번째 정규 폼으로 간주됩니다.

반복 그룹 제거 단계

- 원래의 정규화되지 않은 테이블에서 반복 열을 제거합니다. 기보 테이브이 기보 기사 자꾸
- 기본 테이블의 기본 키와 반복 열로 새 테이블을 생성합니다.
- 기본 키에 고유성이 보장되는 적절한 다른 열을 추가합니다.
- oracle internal • 새 테이블에 외래 키를 생성하여 원래의 정규화되지 않은 테이블로 다시 연결합니다.

첫번째 정규 폼 – 단일 레코드

USERS	U	IS	E	R	S
-------	---	----	---	---	---

USER	USER	MSE	REC_					SRVR	SERVER
_ID	_NAME	_ID	DATE	SUBJECT		TEXT		_ID	_NAME
2301	Smith	54101	05/07	Meeting	Today	There	is	3786	IMAP05
5607	Jones	54512	06/07	Lunch?		Can yo	ou	6001	IMAP08
7773	Walsh	54101	05/07	Meeting	Today	There	is	9988	EMEA01
0022	Patel	54101	05/07	Meeting	Today	There	is	9988	EMEA01

	USER	USER	SRVR	SERVER
	_ID	_NAME	_ID	_NAME
USERS				
	2301	Smith	3786	IMAP05
	5607	Jones	6001	IMAP08
				EMEA01
	0022	Patel	9988	EMEA01

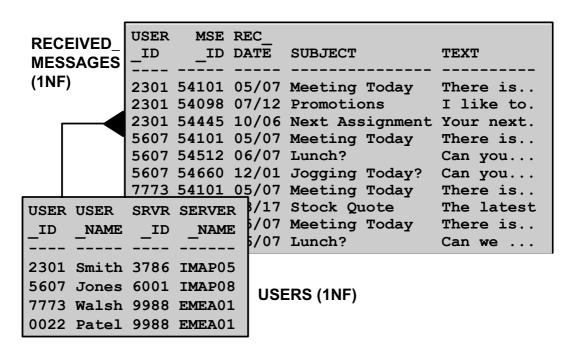
ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

첫번째 정규 폼

먼저 두번째 테이블을 생성하여 반복 그룹 열이 포함되도록 합니다. 그런 다음 정규화되지 않은 테이블의 기본 키와 고유한 다른 열로 구성되는 기본 키를 생성합니다. 마지막으로 하나의 외래 키를 생성하여 첫번째 테이블로 다시 연결합니다.

첫번째 정규 폼 – 반복 그룹



Copyright © Oracle Corporation, 2002. All rights reserved.

ORACLE

두번째 정규 폼으로 변환

- 1. 테이블의 전체 기본 키에 종속되지 않는 키가 아닌 열을 확인합니다.
- 2. 기본 테이블에서 해당 열을 제거합니다.
- 3. 해당 열과 이 열이 종속되는 PK의 열로 두번째 테이블 을 생성합니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

두번째 정규 폼

두번째 정규 폼(2NF)의 정의

테이블이 첫번째 정규 폼 형태이며 기본 키가 아닌 모든 키 열이 전체 기본 키에 기능적으로 종속된 경우 해당 테이블은 두번째 정규 폼 형태입니다. 기본 키가 아닌 키 열은 기본 키 일부에 기능적으로 종속될 수 없습니다.

종속 관계에 따르면 B 열이 A 열에 종속될 경우 A가 변경될 때마다 B를 재평가해야 합니다. 다음 중 하나라도 적용되면 첫번째 정규 폼의 테이블은 두번째 정규 폼 형태가 됩니다.

- 기본 키는 하나의 열로만 구성됩니다.
- 테이블에 키로 지정되지 않은 열이 없습니다.
- 키로 지정되지 않은 모든 속성은 기본 키에 포함된 모든 열에 종속됩니다.

부분적 종속성을 제거하기 위한 단계

- 테이블의 전체 기본 키에 종속되는 키가 아닌 열을 확인합니다.
- 기본 테이블에서 해당 열을 제거합니다. 키로 지정되지 않은 열과 해당 열이 종속된 기본 키의 열 복사본으로 두번째 테이블을 생성합니다.
- 원래의 기본 테이블에서 새 테이블로 외래 키를 생성하고 새 기본 키에 연결합니다.

테이블이 이미 두번째 정규 폼 형태임

USERS



USER _ID	USER _NAME		SERVER _NAME
			IMAP05
7773	Walsh	9988	IMAP08 EMEA01
0022	Patel	9988	EMEA01

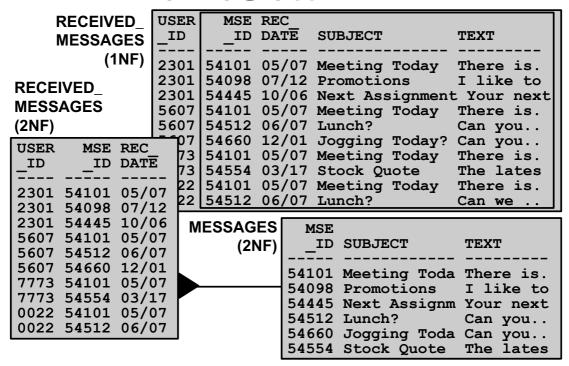
USERS 테이블은 이미 2NF 형태입니까?

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



두번째 정규 폼으로 변환



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

세번째 정규 폼으로 변환

키가 아닌 다른 열에 종속되는 모든 열을 제거합니다.

- 1. 키가 아닌 다른 열에 종속된 열을 확인합니다.
- 2. 기본 테이블에서 해당 열을 제거합니다.
- 3. 해당 열과 이 열이 종속되는 키가 아닌 열로 두번째 테이블을 생성합니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

세번째 정규 폼

세번째 정규 폼(3NF)의 정의

키로 지정되지 않은 모든 열이 기본 키에 직접 종속되며 키로 지정되지 않은 다른 열에 종속되지 않는 경우 테이블은 세번째 정규 폼입니다. 테이블이 두번째 정규 폼 형태이고 모든 "전이적 종속성"이 제거된 경우, 키로 지정되지 않은 모든 열은 "전적으로 해당 키에만 종속되어 있다"고 할 수 있습니다.

전이적 종속성을 제거하기 위한 단계

- 키로 지정되지 않은 다른 열에 종속된 열을 확인합니다.
- 기본 테이블에서 해당 열을 제거합니다.
- 해당 열과 이 열이 종속되는 키가 아닌 열로 두번째 테이블을 생성합니다.
- 원본 테이블에 새 테이블의 기본 키에 연결되는 외래 키를 생성합니다.

테이블이 이미 세번째 정규 폼 형태임

키가 아닌 열은 키가 아닌 다른 열에 기능적으로 종속될 수 없습니다.

RECEIVED_ MESSAGES (2NF)

USER	MSE	REC
_ID	_ID	DATE
2301	54101	05/07
2301	54098	07/12
2301	54445	10/06
5607	54101	05/07
5607	54512	06/07
5607	54660	12/01
7773	54101	05/07
7773	54554	03/17
0022	54101	05/07
0022	54512	06/07

MESSAGES (2NF)

ID	SUBJECT	TEXT	
54101		There is.	
	Promotions	I like to	
54445	Next Assignment Your		
next			
	Lunch?	Can you	
		Can you	
54554	Stock Quote	The lates	

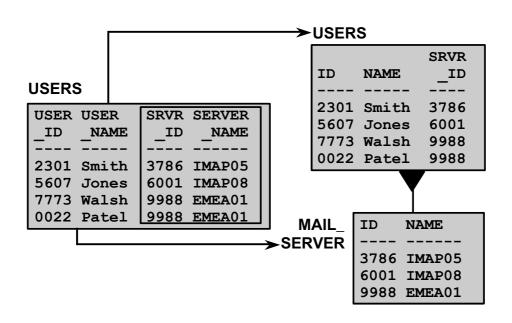
세번째 정규 폼에 이러한 두 개의 테이블이 있습니까? 그 이유는 무엇입니까?

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.



세번째 정규 폼으로 변환



ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

세번째 정규 폼

.. 付下答问。ONLUSE ONLUSE ONLUSE ONLUSE ONLUSE ONLUSE 정규화 이론은 잔존할 수 있는 몇 가지 불확실한 구성을 해결하기 위한 세번째 정규 폼 이상의 개념으로 확대되었습니다. 이러한 정규 폼은 다른 과정에서 다룹니다.

요약

- 1NF 테이블은 정렬되지 않은 2차원 테이블 세트로 표시되어야 합니다. 테이블은 반복 그룹을 포함할 수 없습니다.
- 2NF 테이블은 1NF 형태여야 합니다. 키가 아닌 모든 열은 기본 키의 모든 부분에 종속되어야 합니다.
- 3NF 테이블은 2NF 형태여야 합니다. 키가 아닌 열은 키가 아닌 다른 열에 기능적으로 종속될 수 없습니다.

엔티티 관계 모델은 정규화된 데이터 설계로 변경됩니다.

ORACLE

Copyright © Oracle Corporation, 2002. All rights reserved.

