

6. ERwin Data Modeler



ERwin Data Modeler

- ◆ 개요
 - <https://erwin.com/products/data-modeler/>
 - Logic Works社에서 개발한 데이터 모델링 도구 (since 1993)
 - 데이터베이스를 빠르고 쉽게 설계할 수 있도록 지원
 - GUI 방식의 쉬운 작업을 통해 개발 기간 단축 효과
 - 운영/유지보수 환경에서 ERD와 Database Schema의 지속적인 관리 지원
 - 다양한 DBMS 지원
 - Oracle, SQL Server, DB2, Sybase, PostgreSQL, Teradata, SAS, MySQL 등
- ◆ 실행
 - erwin → erwin Data Modeler
 - File > New... : 새로운 모델 작성
 - File > Open... : 파일에 저장된 기존 모델을 불러옴

- ◆ CASE(Computer Aided Software Engineering) Tool
 - 개발방법론에 기반한 프로그래밍 자동화 도구
 - 예: UML Tools
- ◆ Data Modeling Tools
 - 데이터베이스의 논리적/물리적 설계를 지원하는 CASE Tool
 - 예
 - erwin Data Modeler (CA → Quest)
 - Toad Data Modeler (Quest)
 - PowerDesigner (SAP)
 - Oracle SQL Developer Data Modeler
 - MySQL Workbench
 - SQL Server Management Studio (Microsoft)

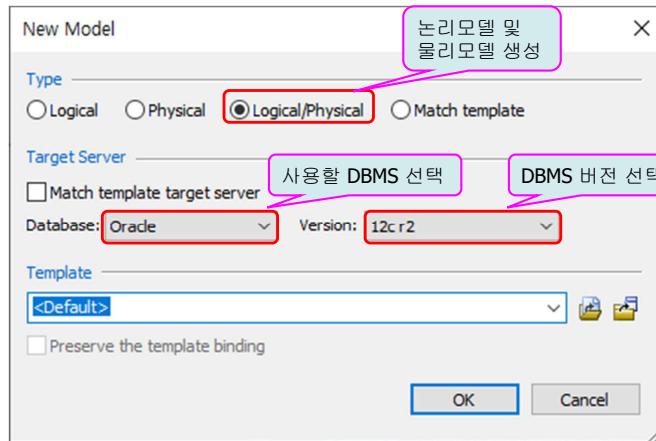
2

ERwin Data Modeler의 특징

- ◆ Method
 - IDEF1X Method
 - 정보 공학 기법
- ◆ Logical Modeling
 - Entity/Attribute/
Relationship
 - 업무 영역별 작업(Subject Area)
 - Domain 정의 및 공유
- ◆ Physical Modeling
 - Validation/Default
 - Index/Trigger
 - Physical Property
- ◆ Dimensional Modeling
- ◆ Forward/Reverse Engineering
 - DDL 자동 생성
 - Reverse 기능
 - Complete-Compare
- ◆ Edit & Report
 - 다양한 편집 및 reporting
 - Reporting format customization

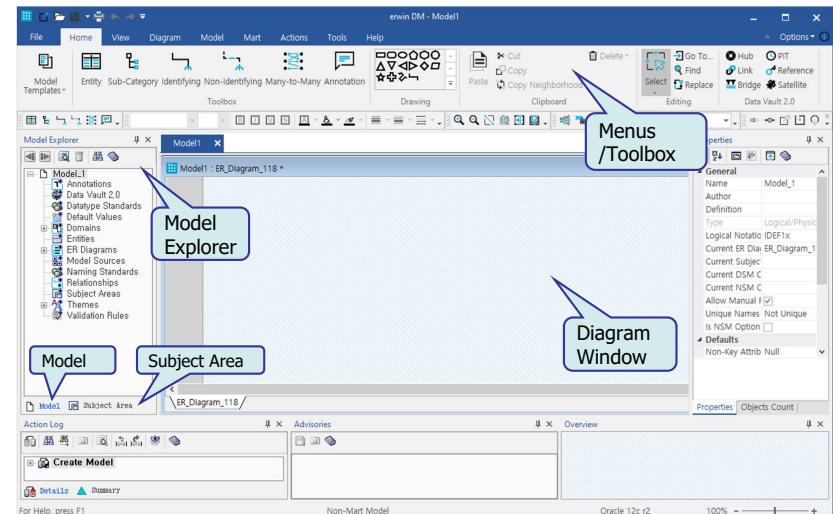
모델 생성

- ◆ Create new model



5

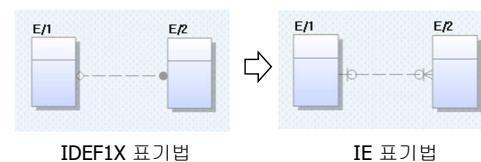
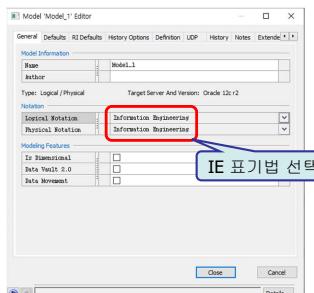
ERwin 화면 구성



6

ERwin의 표기 방식

- ◆ 개체 관계 모델(E-R Model)을 표현하기 위한 표기 방식
 - IDEF1X(Integration DEFinition for Information Modeling) 표기방식
 - 미국방성에서 프로젝트 표준안으로 개발한 표기 방식
 - 기본적으로 이 방식으로 설정되어 있음
 - IE(Information Engineering) 표기방식
 - 정보공학 표기 방식으로 일반적으로 널리 사용됨
- ◆ IDEF1X → IE 방식 변경
 - Model > Model Properties > General 탭에서 변경 가능



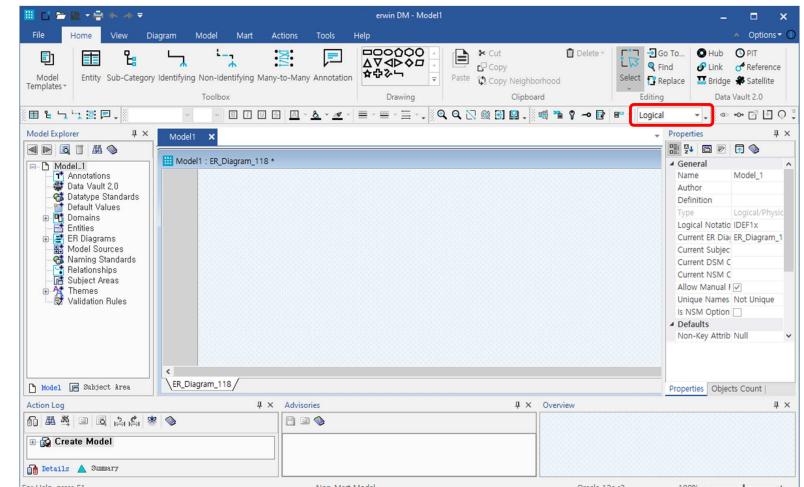
IDEF1X 표기법

IE 표기법

7

논리적 모델 ↔ 물리적 모델

- ◆ Logical Model과 Physical Model 간의 전환
 - ERwin Toolbox의 오른쪽 combo box에서 선택

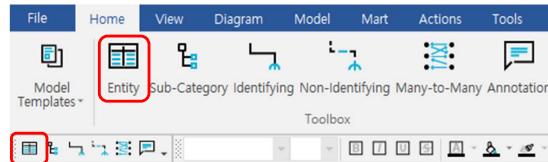


8

개체(Entity) 생성

◆ 개체 생성 방법

1. ERwin Toolbox의 Entity 버튼을 선택한 후 Diagram Window 내에서 클릭하면 새로운 개체가 생성됨

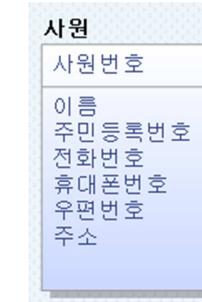


9

개체 생성

2. 개체를 선택한 뒤 Tab 키를 이용하여 영역 간 이동 가능

3. 일반 속성 영역에 속성 추가: 'Enter' key를 치면 새로운 속성을 추가할 자리가 만들어짐



10

관계(Relationship) 생성

◆ 관계(Relationship): 두 개체 간의 업무적 연관성을 표현

◆ 식별 관계(Identifying Relationship)

- 부모 개체(강 개체)에 의해 자식 개체(약 개체)가 식별됨
- 부모 개체의 기본 키가 자식 개체의 기본 키 그룹의 구성원으로 전이됨
- 부모 개체에 대한 1:1 또는 1:N 관계 표현 **N:N 불가!**

◆ 비식별 관계(Non-Identifying Relationship)

- 자식 개체를 독립적으로 식별 가능
- 부모 개체의 기본 키가 자식 개체의 일반 속성으로 전이됨
- 부모 개체에 대한 1:1 또는 1:N 관계 표현

관계 생성

◆ 관계 생성 방법

1. ERwin Toolbox에서 관계 유형에 맞는 관계선 버튼을 선택함



Identifying Relationship
(식별 관계 정의)

Non-Identifying Relationship
(비식별 관계 정의)

11

12

관계 생성

2. 부모 개체를 먼저 선택한 후 자식 개체를 선택함
3. 관계에 관한 property들을 설정
 - 관계선을 더블 클릭하거나, 우클릭한 후 Properties 선택
4. 관계 타입(Relationship Type) 설정
 - Type: 'Identifying' 또는 'Non-Identifying' 선택
 - Null option: 부모의 관계 참여가 필수/선택인지 여부
5. Verb Phrase 설정
 - 부모와 자식(Parent-to-Child), 자식과 부모(Child-to-Parent) 관계에 대한 설명을 입력
6. 관계 차수(Cardinality) 설정
 - Cardinality: 관계에 참여하는 자식 개체의 수

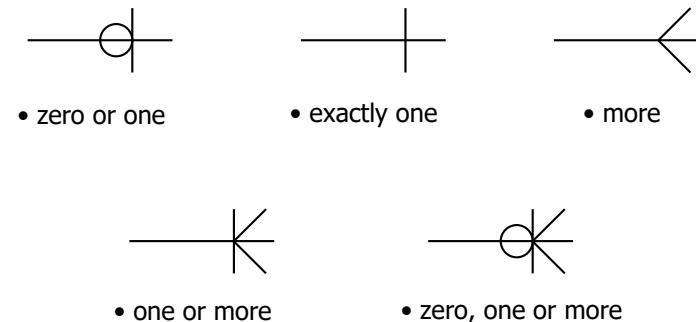
13

관계 생성

◆ 관계 차수(Cardinality)

- 관계를 갖고 있는 개체들 사이의 관계 참여 횟수를 나타냄

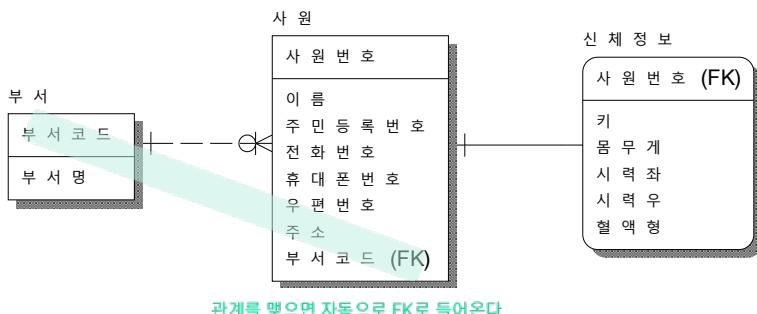
자식의 개수



14

관계 생성: 예

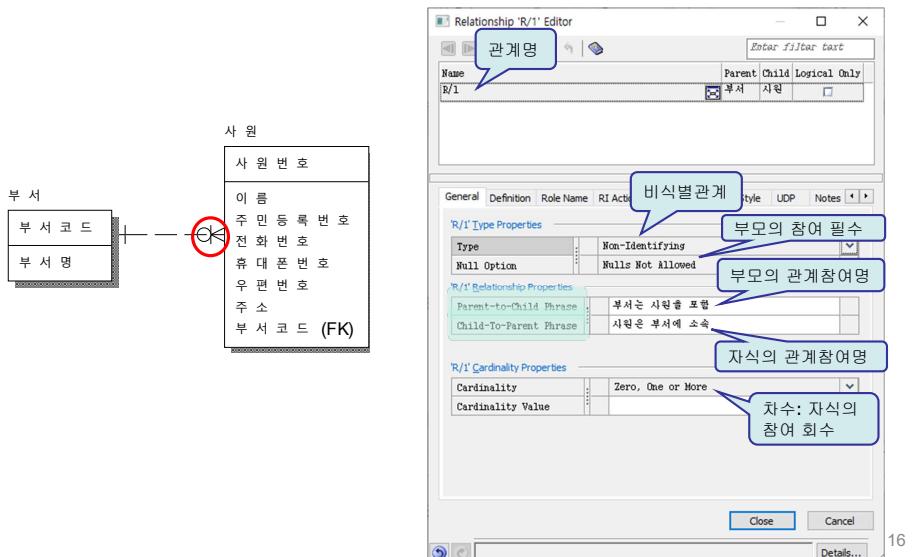
- ◆ 부서와 사원, 신체정보를 정의하기 위해 부서, 사원, 신체정보 개체를 생성하고 개체들 사이에 관계를 설정함
 - 부서-사원 관계: 1:N 비식별 관계
 - 사원-신체정보 관계: 1:1 식별 관계



15

관계 생성: 예

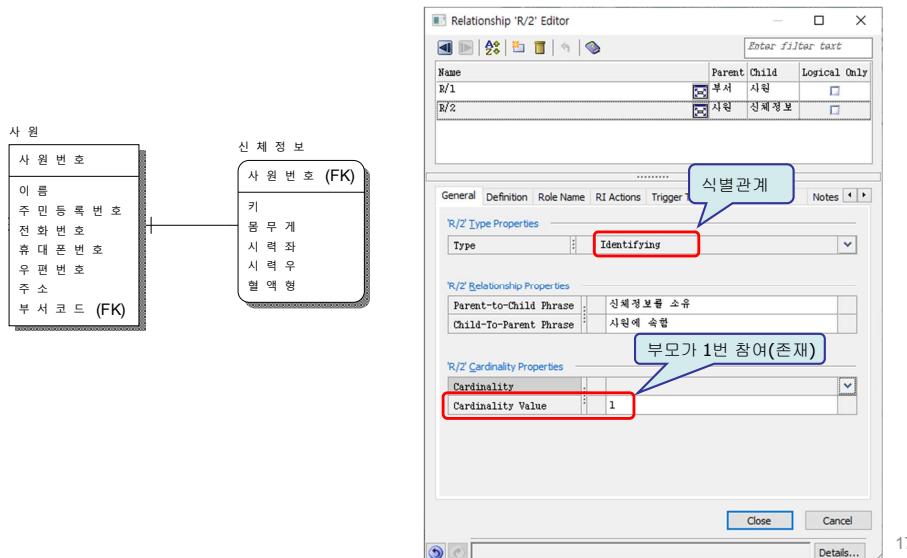
◆ 부서 개체와 사원 개체 간의 비식별 관계 property 설정



16

관계 생성: 예

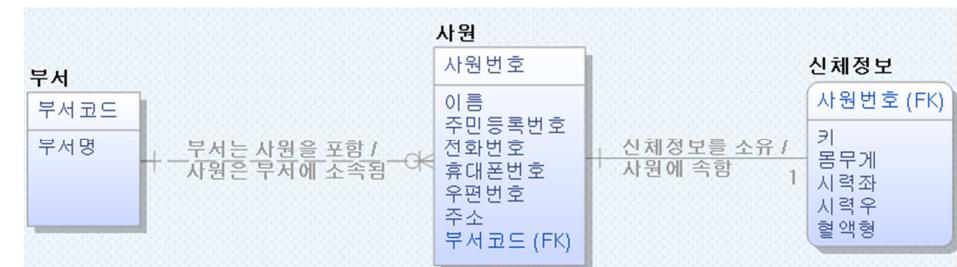
- ◆ 사원 개체와 신체정보 개체 간의 식별 관계 property 설정



관계 생성: 예

- ◆ Cardinality 및 Verb Phrase 화면 표시

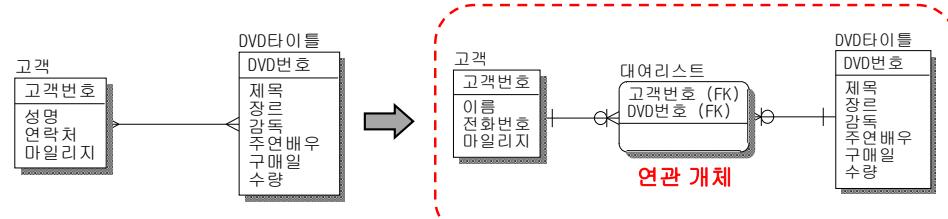
- context menu(diagram window에서 마우스 우클릭)에서 Properties.. 선택
 - Display 탭에서 Display Child-To-Parent Verb Phrase 등을 선택
 - Relationship 탭에서 Display Logical Cardinality 선택



Many-to-Many 관계

- ◆ Many-to-Many(M:N) 관계

- 논리적으로만 존재 가능
 - 논리 모델에서는 표현 가능하나 물리 모델에서는 표현 불가
- 연관 개체(Association Entity)로 변환 수행
 - 두 개체 사이에 관계가 발생할 때마다 이를 기록하는 개체

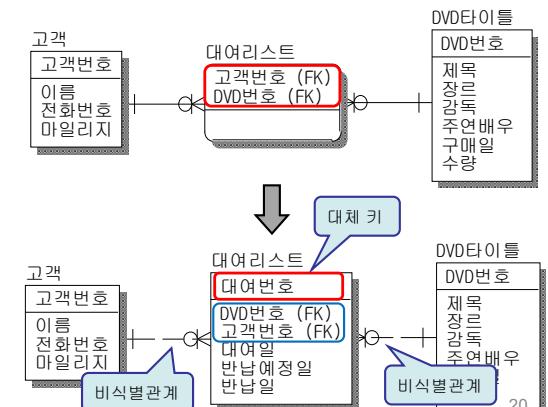


Many-to-Many 관계

- ◆ 연관 개체(Association Entity)의 정제

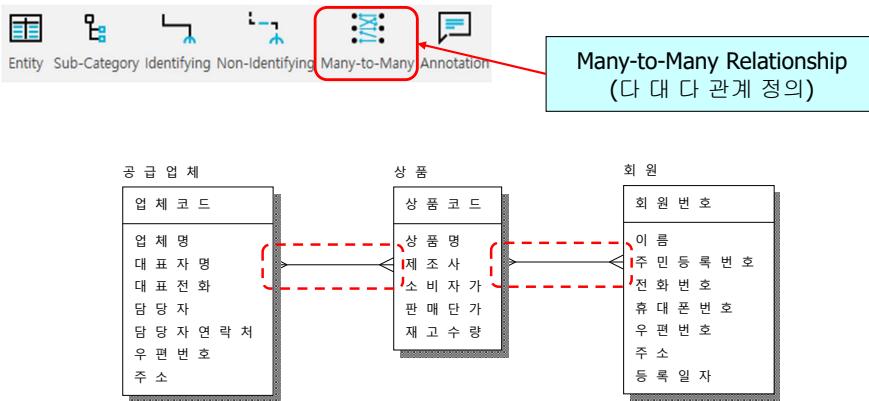
- 기본 키 정제
 - 새로운 식별자 속성 추가
 - ✓ 대체 키(surrogate key)
 - 식별 관계를 비식별 관계로 변환

- 속성 추가
 - 관계와 관련된 부가적인 정보 추가



Many-to-Many 관계 정의: 예

- Many-to-Many 관계에 있는 개체들은 부모와 자식의 관계가 아니므로 ERwin Toolbox에서 Many-to-Many 관계선을 선택한 후 순서에 상관없이 두 개체를 차례대로 선택하면 관계가 형성됨

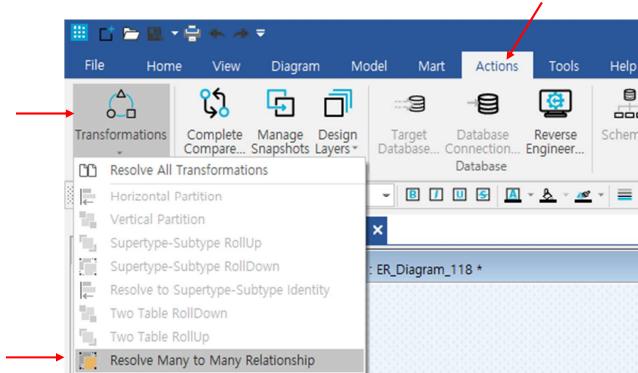


21

Many-to-Many 관계 정의: 예

- Many-to-Many 관계를 연관 개체로 변환

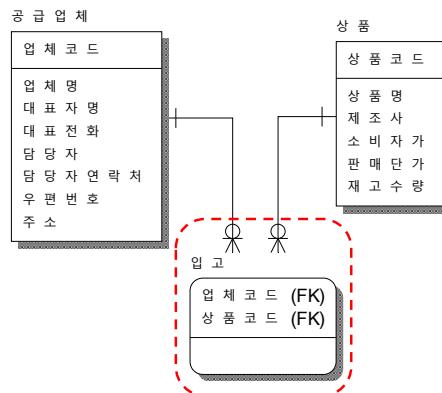
- 공급업체 – 상품 개체 사이의 Many-to-Many 관계선을 선택한 후 Action > Transformations > Resolve Many-to-Many Relationship 선택



22

Many-to-Many 관계 정의: 예

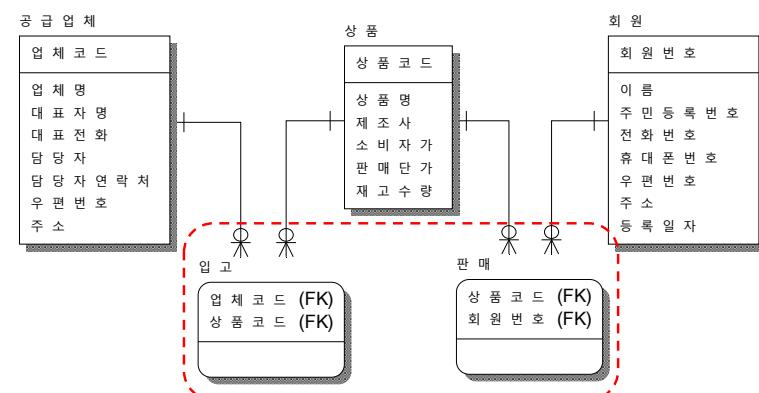
- 생성된 연관 개체의 property 설정을 통해 개체 이름을 '입고'로 변경



23

Many-to-Many 관계 정의: 예

- 같은 방식으로 상품 – 회원 개체 사이의 Many-to-Many 관계에 대해 '판매' 연관 개체를 생성함

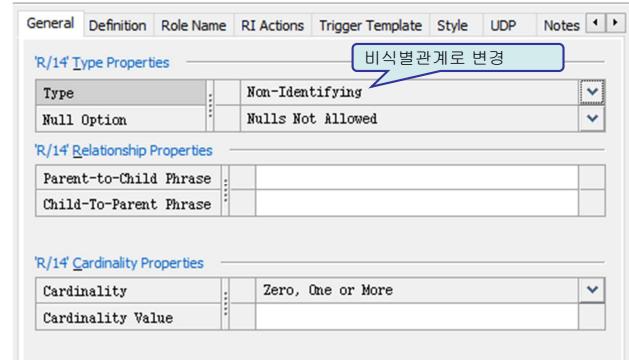


24

Many-to-Many 관계 정의: 예

4. 모든 관계를 비식별 관계(Non-Identifying Relationship)로 변경

- 관계선을 우클릭 > Properties... 선택 > Type의 값을 'Non-Identifying'으로 변경

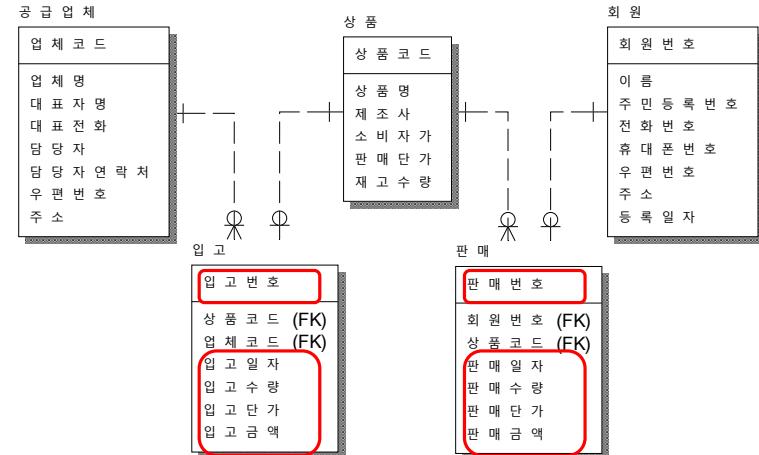


25

Many-to-Many 관계 정의: 예

5. 연관 개체의 기본 키 및 부가적인 속성 정의

- 대체 키 정의 (예: 입고번호, 판매번호)
- 부가 속성 추가 (예: 입고일자, 입고수량 등)



26

재귀적 관계 정의

- 재귀적 관계(Recursive Relationships): 개체 타입이 자기 자신과 관계를 맺는 것

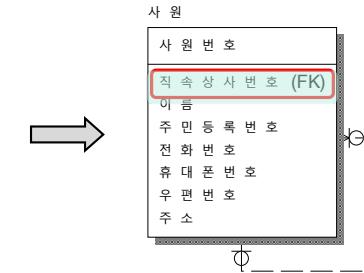
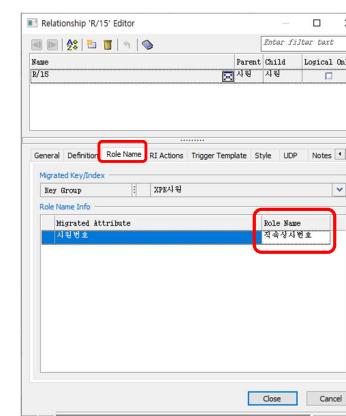


27

재귀적 관계 정의: 예

◆ Role Name 정의

- 관계의 Properties 설정에서 Role Name 탭을 선택한 후 Role Name창에 '직속상사번호' 입력
→ '직속상사번호' 속성이 생성되고 foreign key로 지정됨



28

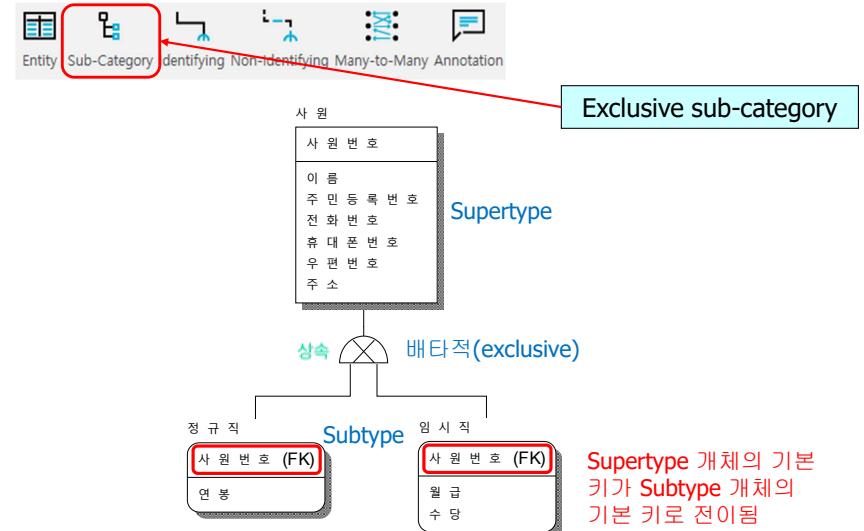
Supertype과 Subtype

- ◆ Supertype 개체란 상호 배타적인 더 작은 그룹으로 분할될 수 있는 개체를 의미
 - 공통되는 속성들을 모아놓은 개체
- ◆ Supertype 내의 분해된 그룹을 Subtype 개체라고 함
 - 개별적인 속성들을 따로 분리해서 정의
- ◆ Supertype – Subtype 사이에는 Generalization/Specialization 관계가 성립

29

Supertype과 Subtype

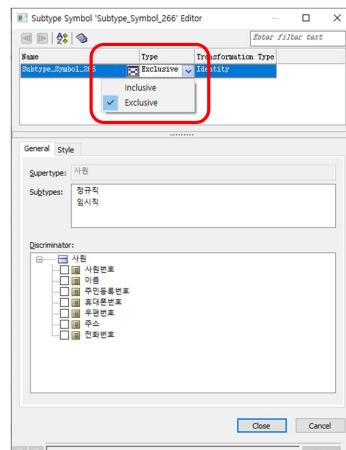
- ◆ Supertype – Subtype 관계 생성



30

Supertype과 Subtype

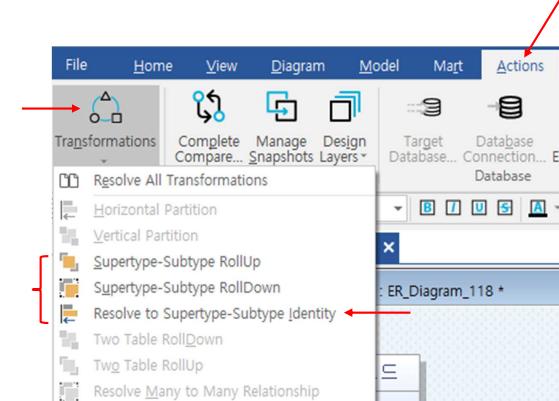
- ◆ Exclusive/Inclusive 관계 정의
 - 중간 교차점을 더블 클릭하거나, 우클릭 후 Properties... 선택
 - Type option 선택



31

Supertype과 Subtype

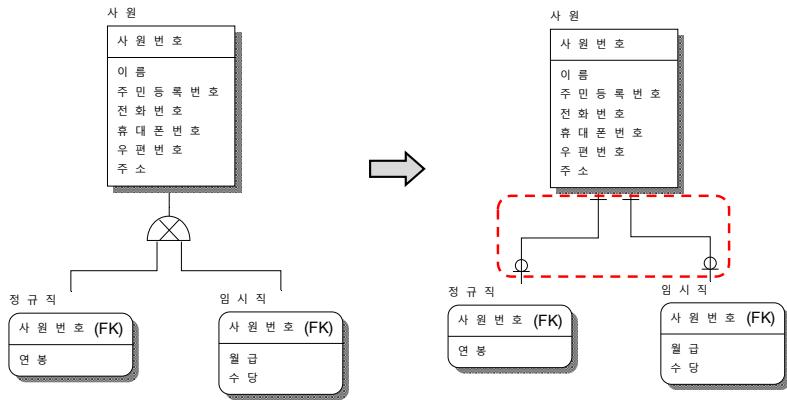
- ◆ 식별관계로 변환
 - 중간 교차점을 선택한 후 Action > Transformations > Resolve to Supertype-Subtype Identity 항목 선택



32

Supertype과 Subtype

- ◆ 식별관계로 변환



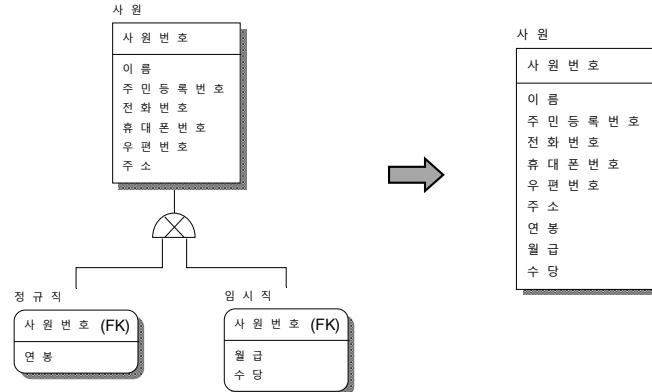
33

Supertype과 Subtype

- ◆ Supertype 개체로 통합

- 중간 교차점을 선택한 후

Action > Transformations > Supertype-Subtype Rollup 항목 선택



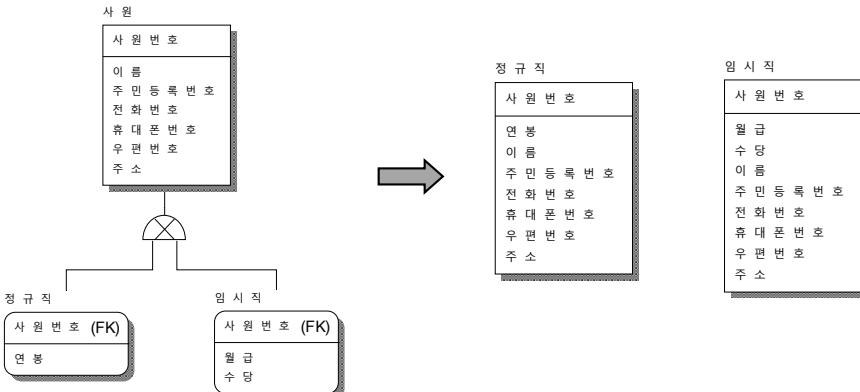
34

Supertype과 Subtype

- ◆ Subtype 개체로 분할

- 중간 교차점을 선택한 후

Action > Transformations > Supertype-Subtype Rolldown 항목 선택



35

Domain 설정

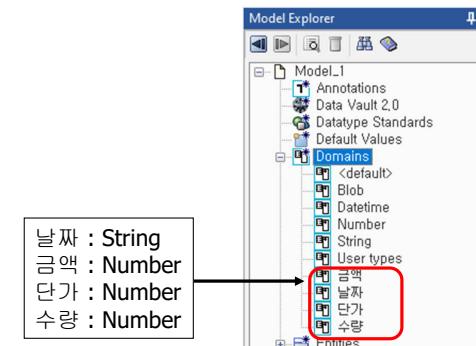
- ◆ 사용자 정의 데이터 타입 생성하고 속성에 binding할 수 있는 기능

- 여러 개체에서 공통적으로 사용되는 속성이 존재하면 Domain을 정의해서 여러 개체에 적용 가능

- ◆ Domain 생성

- Model Explorer > Domains 항목을 우클릭 > New 선택

- 또는 기 정의된 domain 중 하나를 선택해서 New 실행

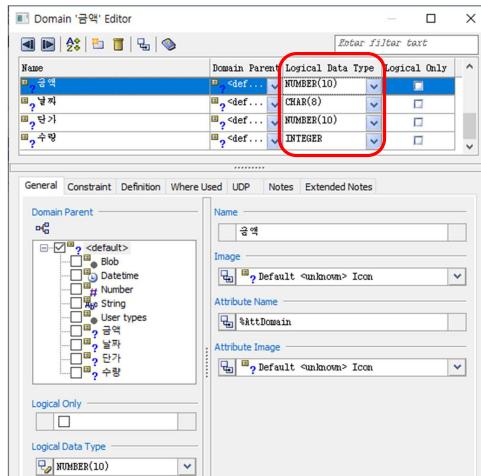


36

Domain 설정

◆ Domain 탑재 정의

- 특정 domain을 우클릭 > Properties... 선택 > Logical Data Type에서 탑재 지정(변경) 가능

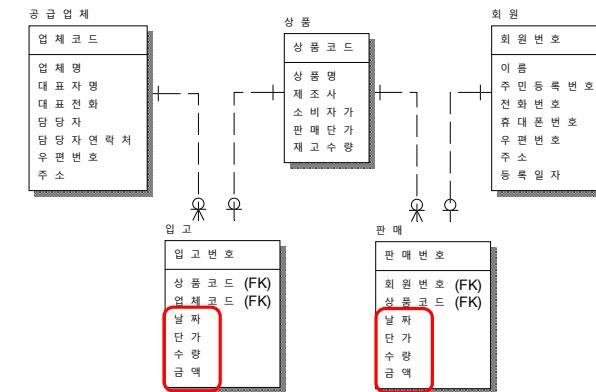


37

Domain 설정

◆ Domain 적용

- 입고 개체의 입고일자, 입고단가 등의 속성과 판매 개체의 판매일자, 판매 단가 등의 속성들을 제거
- Model Explorer > Domains에 정의된 날짜, 단가 등의 domain을 drag해서 입고 개체와 판매 개체 안에 drop

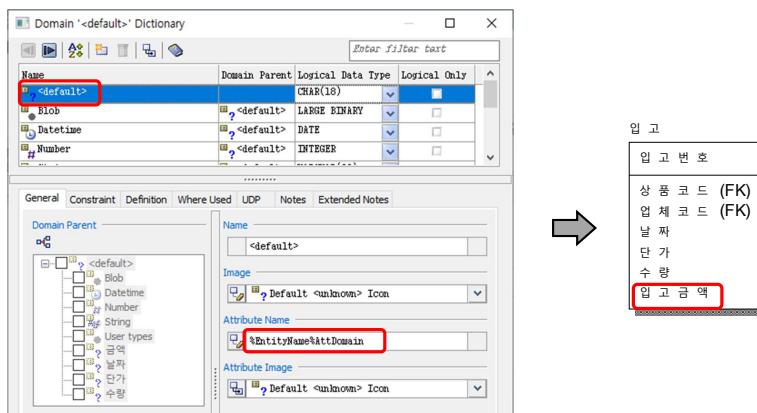


38

Domain 설정

◆ Domain 적용 시 속성명 앞에 개체명 자동 추가

- Model Explorer > Domains > <default> 항목을 우클릭 > Properties... 선택
- Attribute Name 입력 값을 %EntityName%AttDomain으로 정의하면 domain이 적용되는 모든 속성명 앞에 개체명이 붙어서 정의됨



39

Subject Area

◆ Subject Area

- 업무적으로 관련이 있거나 혹은 개발자가 보고자 하는 개체들만으로 화면을 구성해서 좀 더 편리하게 개체와 관계를 작성할 수 있음
- Subject Area에서 내용을 변경하면 전체 디아이어그램에 즉시 반영됨
- 데이터베이스 객체들을 생성할 때 Subject Area 별로 생성 가능

◆ Subject Area 생성 및 선택(이동)

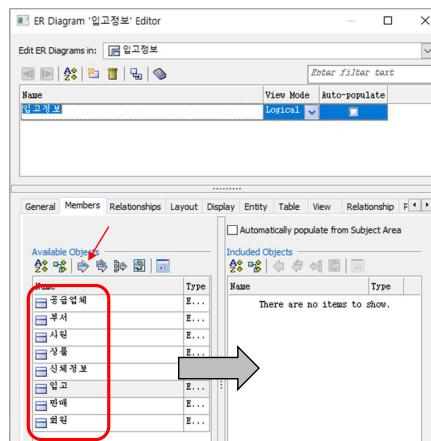
- ERwin Toolbar의 Edit Subject Area 버튼 선택
- Model > Subject Areas 메뉴 선택
- Model Explorer > Subject Areas 메뉴 이용



40

Subject Area: 사용 예

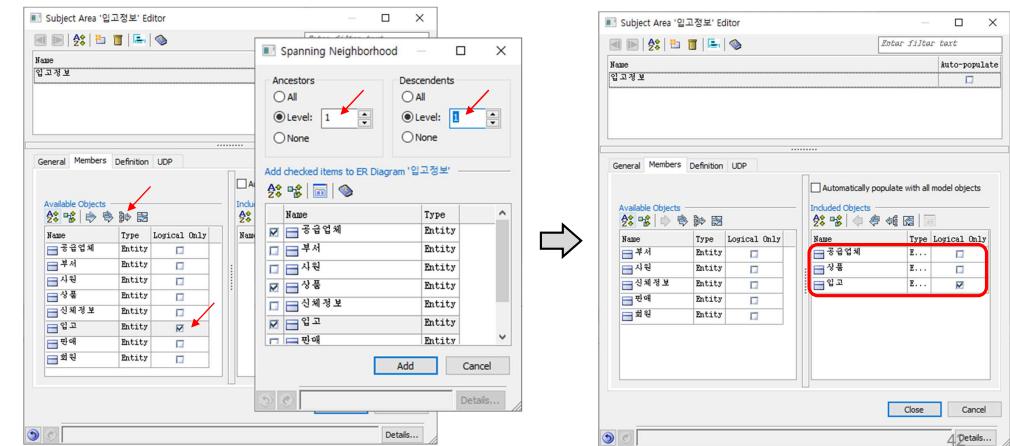
1. ERwin Toolbar에서 Edit Subject Area 버튼 클릭 > New 버튼 선택
2. Subject Area의 이름 변경
3. Members 탭의 Available Objects 리스트의 개체들을 선택하여 Included Objects 리스트로 옮기면 subject area에 포함됨



41

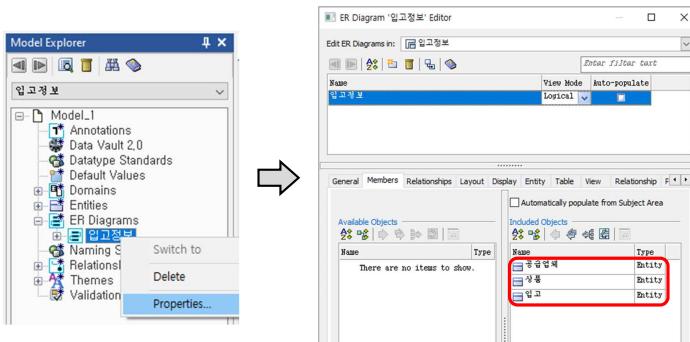
Subject Area: 사용 예

4. Spanning Neighborhood: Available Objects에서 선택된 특정 개체를 중심으로 조상 또는 자손 관계에 있는 개체들을 자동적으로 선택 및 이동
 - 조상/자손 레벨 지정 가능



Subject Area: 사용 예

5. Subject Area 전환 후 Model Explorer > ER Diagrams > 해당 항목 선택 > 마우스 우클릭 > Properties... 선택
 - Available Objects의 개체들을 모두 Included Objects로 옮겨야 화면에 출력됨



43

6. Subject Area에서 작업한 결과는 전체 모델에 자동으로 반영됨

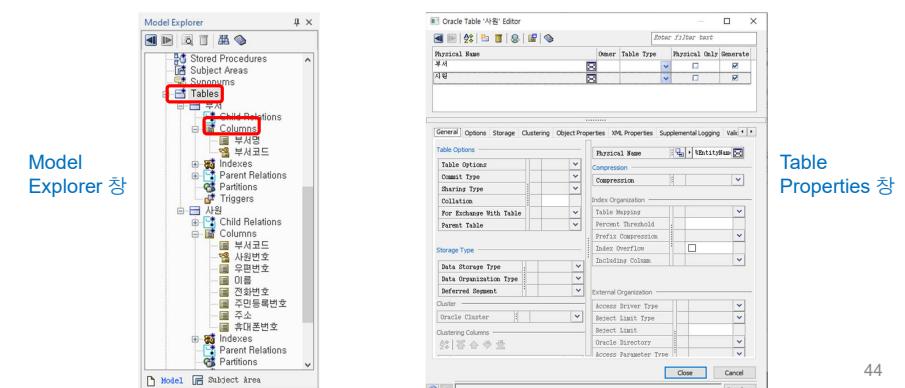
- 예: '입고정보' Subject Area에서 상품 개체에 '등록일자' 속성을 추가한 후 <model>의 다이어그램에도 포함되었는지 확인

물리적 모델링(Physical Modeling)

- ◆ ERwin Toolbox의 오른쪽 콤보박스에서 'Physical'을 선택하면 물리적 모델링으로 전환됨
 - 논리적 모델링에서 물리적 모델링으로 전환하면 사용하는 용어가 바뀜
 - 예: 개체(Entity) → 테이블(Table), 속성(Attribute) → 컬럼(Column)



Model Explorer 창



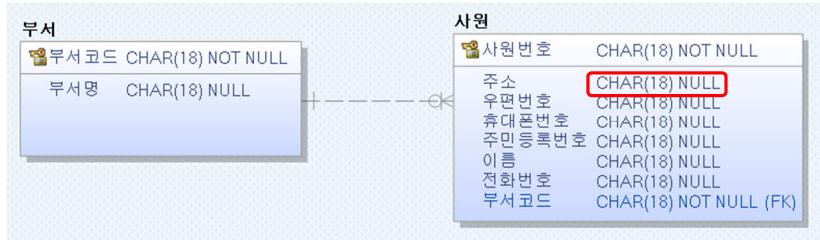
44

Table Properties 창

Column Data Type과 Size 정의

◆ 세부 정보 표시

- Diagram window의 context menu에서 **Properties..**를 선택하고 Table 탭에서 Column Data Type, Null Option, PK Designator 등을 활성화 /비활성화 가능



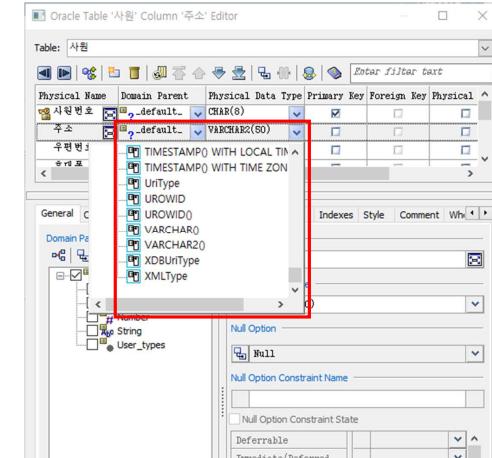
- 주의: 초기에는 각 컬럼의 Data type과 Size가 임의로 설정되어 있으므로 실제 저장될 데이터에 맞게 변경해야 함

45

Column Data Type과 Size 정의

◆ Data type 및 size 지정

- 테이블을 선택한 뒤 우클릭하여 **Column Properties..**를 선택
 - 각 컬럼의 Physical Data Type에서 Type과 Size를 지정(변경) 가능



46

Validation Rule, Default 값 정의

◆ Validation Rule

- 컬럼 값에 대한 CHECK 제약 조건을 정의
- 종류
 - User-defined
 - Min/Max
 - Valid Value List

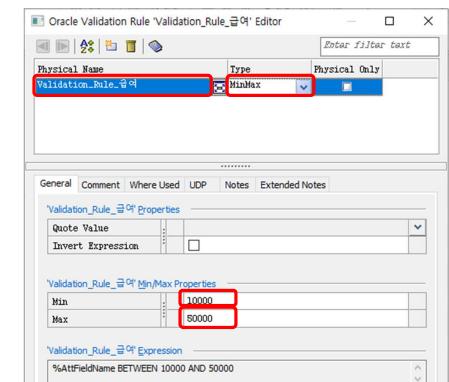
◆ Default

- 컬럼에 저장될 default 값 정의

47

Validation Rule 정의

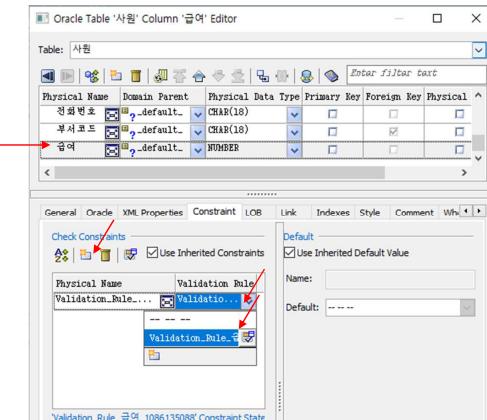
- Model Explorer > Validation Rules 항목을 우클릭 > New 선택 > 생성된 항목을 우클릭 > Properties.. 선택
 - 또는 Model 메뉴에서 Validation Rules... 버튼 선택하여 생성 가능
- Physical Name 변경
- Type 선택 및 세부 조건 입력
 - MinMax: Min, Max 값 입력
 - Valid Values: 값을 하나씩 입력
 - User-Defined: 조건식 입력



48

Validation Rule 적용

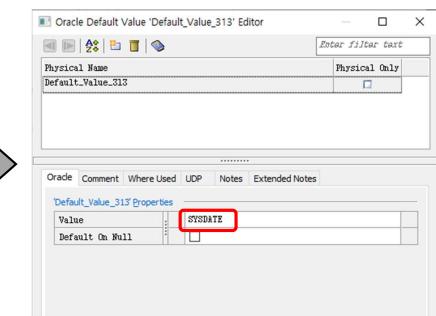
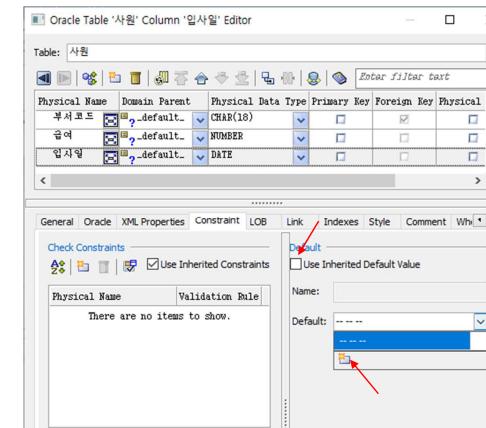
1. 테이블 우클릭 후 **Column Properties...** 선택
2. 특정 컬럼 선택 (예: 급여)
3. Constraint 탭의 Check Constraints 섹션의 new 버튼을 클릭하여 항목 생성
4. 생성된 항목의 Validation Rule 목록에서 기 정의된 특정 rule을 선택
 - new 항목 선택해서 새로운 rule을 생성 및 적용할 수도 있음



49

Default 값 정의

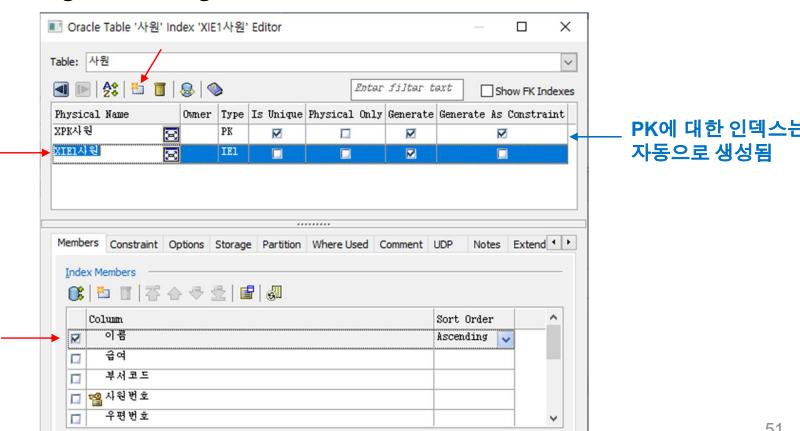
1. 테이블 우클릭 후 **Column Properties...** 선택
2. 특정 컬럼 선택 (예: 입사일)
3. Constraint 탭의 Default 섹션의 Use Inherited Default Value 해제
4. Default 메뉴에서 new 버튼을 클릭 > Oracle 탭의 value 창에 default 값 입력



50

Index 정의

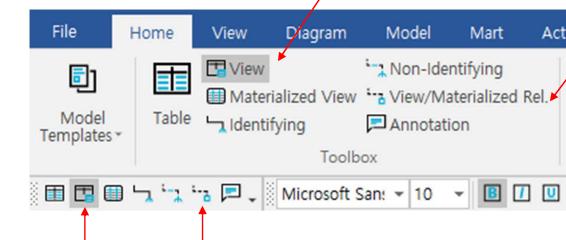
1. 테이블 우클릭 후 **Index Properties...** 선택
2. New 버튼을 클릭하여 새로운 index 항목 생성
3. Members 탭에서 index에 포함시킬 컬럼(들)을 선택
 - ascending/descending 선택 가능



51

View 정의

- ◆ View 생성
 - Home > View 메뉴 또는 ERwin Toolbox에 있는 View 버튼 선택 > diagram window에 클릭하여 view 객체 생성
 - View/Meteralized Relationship 메뉴를 선택하여 base table과 view 연결



52

View 정의

◆ View 정의

- View 객체를 우클릭 > View Properties... 선택
- Select 탭에서 base table에 속한 컬럼 리스트에서 필요한 컬럼들을 선택
- Where 탭에서 선택 조건 입력
- From 탭을 이용하여 join view 정의 가능

base table

view

53

Sequence 정의

◆ Sequence 객체 생성 및 정의

- Model Explorer > Sequences 항목 우클릭 > New 선택
- 생성된 Sequence 객체 우클릭 > Properties... 선택 및 입력

54

Stored Procedure/Function 정의

◆ 객체 생성 및 정의

- Model Explorer > Stored Procedures 또는 Functions 항목을 우클릭 > New 선택 > 생성된 항목을 우클릭 > Properties.. 선택
- 특정 테이블 선택 후 우클릭 > Stored Procedure Properties... 또는 Function Properties... 선택하여 생성 가능

저장 프로시저/함수의 코드 작성 (PL/SQL 이용)

55

역정규화(Denormalization)

- ◆ 시스템의 성능 향상을 위해 정규화에 위배되는 설계를 하는 것
- ◆ Actions > Transformations 의 하위 메뉴 이용

56

Vertical Partition – 컬럼을 기준으로 테이블을 분리

Horizontal Partition – 레코드를 기준으로 테이블을 분리

Two Tables RollDown – 자식 테이블을 기준으로 테이블을 통합

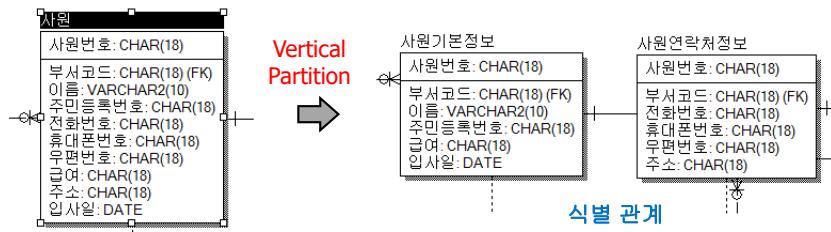
Two Tables RollUp – 부모 테이블을 기준으로 테이블을 통합

Column Denormalization – join 속도 향상을 위해 테이블들에 컬럼을 중복 정의

Table Partitioning

◆ Vertical Partition

- 컬럼을 나누는 가장 중요한 기준은 업무적인 성격
- 업무적인 성격에 따라서 관련된 컬럼들을 서로 다른 테이블에 분리해서 저장하면 하나의 page에 관련된 레코드가 보다 많이 저장되므로 검색 성능을 향상시킬 수 있음

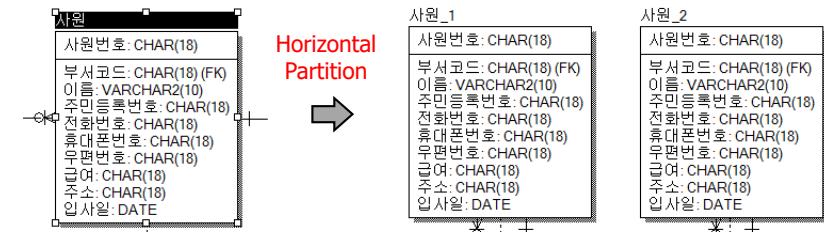


57

Table Partitioning

◆ Horizontal Partition

- 레코드를 기준으로 테이블을 분리하는 경우 사용
- 테이블에 많은 레코드가 저장되어 있고 부분적인 레코드들만 자주 조회할 경우: 전체 레코드를 한 테이블에 저장하는 것보다 여러 테이블로 나누어서 저장 관리하는 것이 검색 성능을 향상시킬 수 있음

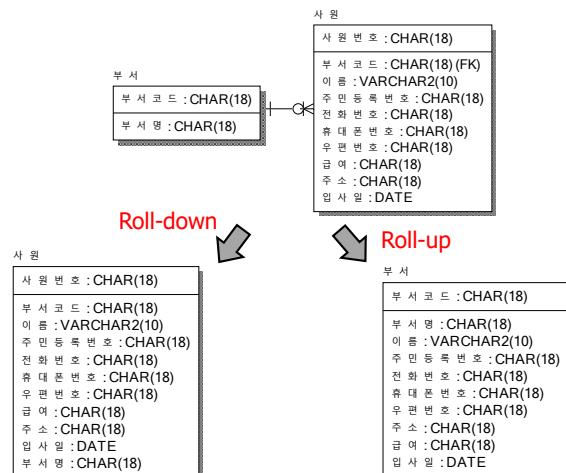


58

Table Roll-Down & Roll-Up

◆ 정규화 과정에서 분리된 두 테이블을 다시 통합하는 방법

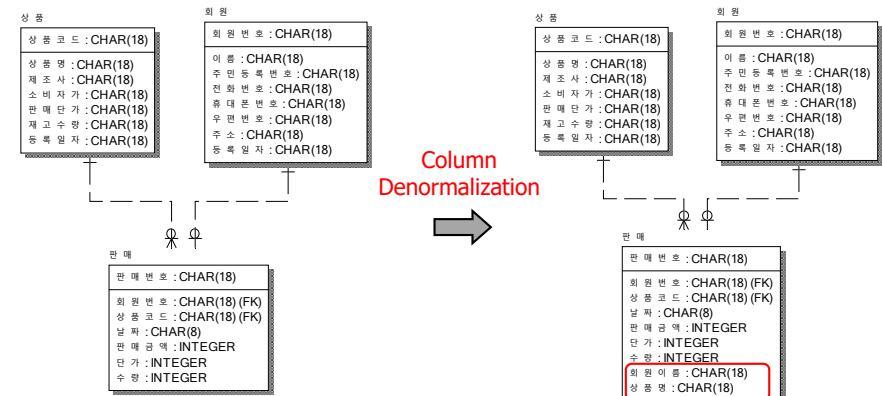
- Roll-Down – 자식 테이블을 기준으로 테이블을 통합
- Roll-Up – 부모 테이블을 기준으로 테이블을 통합



59

Column Denormalization

◆ join의 속도를 향상시키기 위해 대상 테이블들에 컬럼을 중복 정의



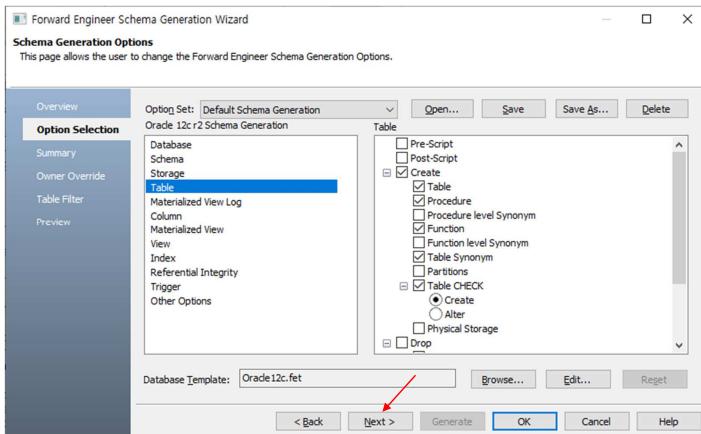
- 데이터가 많을 경우 join을 하기 위해서 많은 부하가 발생하게 되는데 회원 테이블의 이름 컬럼과 상품 테이블의 상품명 컬럼을 판매 테이블에 중복 저장하게 되면 판매 테이블을 조회할 때마다 다른 두 테이블을 join을 하지 않아도 되기 때문에 검색 성능을 향상시킬 수 있음

60

Database Schema 생성

◆ 순공학 (Forward Engineering)

- 물리적 데이터 모델링 결과를 이용하여 특정 DBMS에 대한 데이터베이스 스키마를 생성할 수 있는 **DDL Script 생성**
- Actions > Schema... 메뉴 이용



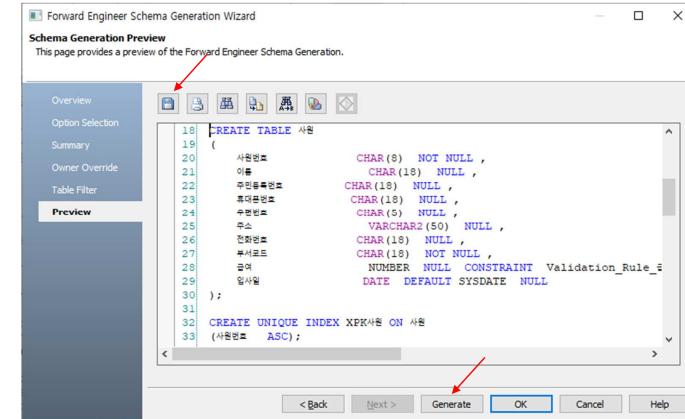
61

Database Schema 생성

- Schema, Table, Column, View, Index 등 종류별로 DDL Script에 포함시킬 대상들을 선택 또는 제외

- 주의: Sequence는 Schema에서 직접 선택, Trigger > erwin Generated 항목은 체크 해제

- Preview에서 자동 생성된 DDL Script 내용 확인 > 파일로 저장하거나 Generate 실행



62

