



## Chapter 06

# 데이터 모델링

# 목차

**01**

**데이터 모델링의 개념**

**02**

**ER 모델**

**03**

**ER 모델을 관계 데이터 모델로 사상**

**04**

**ERwin 실습**

**05**

**모델링 연습(마당대학 데이터베이스)**

# 학습목표

- ❖ 데이터 모델링의 개념을 이해한다.
- ❖ 개념적 모델링 단계에 사용하는 ER 모델을 이해하고 작성 방법을 알아본다.
- ❖ ERwin 프로그램을 사용하여 ER 모델을 작성해본다.
- ❖ 마당대학 데이터베이스를 ER 모델로 직접 설계해본다.

# 01. 데이터 모델링의 개념

1. 데이터베이스 생명주기
2. 데이터 모델링 과정



# 데이터 모델링의 개념

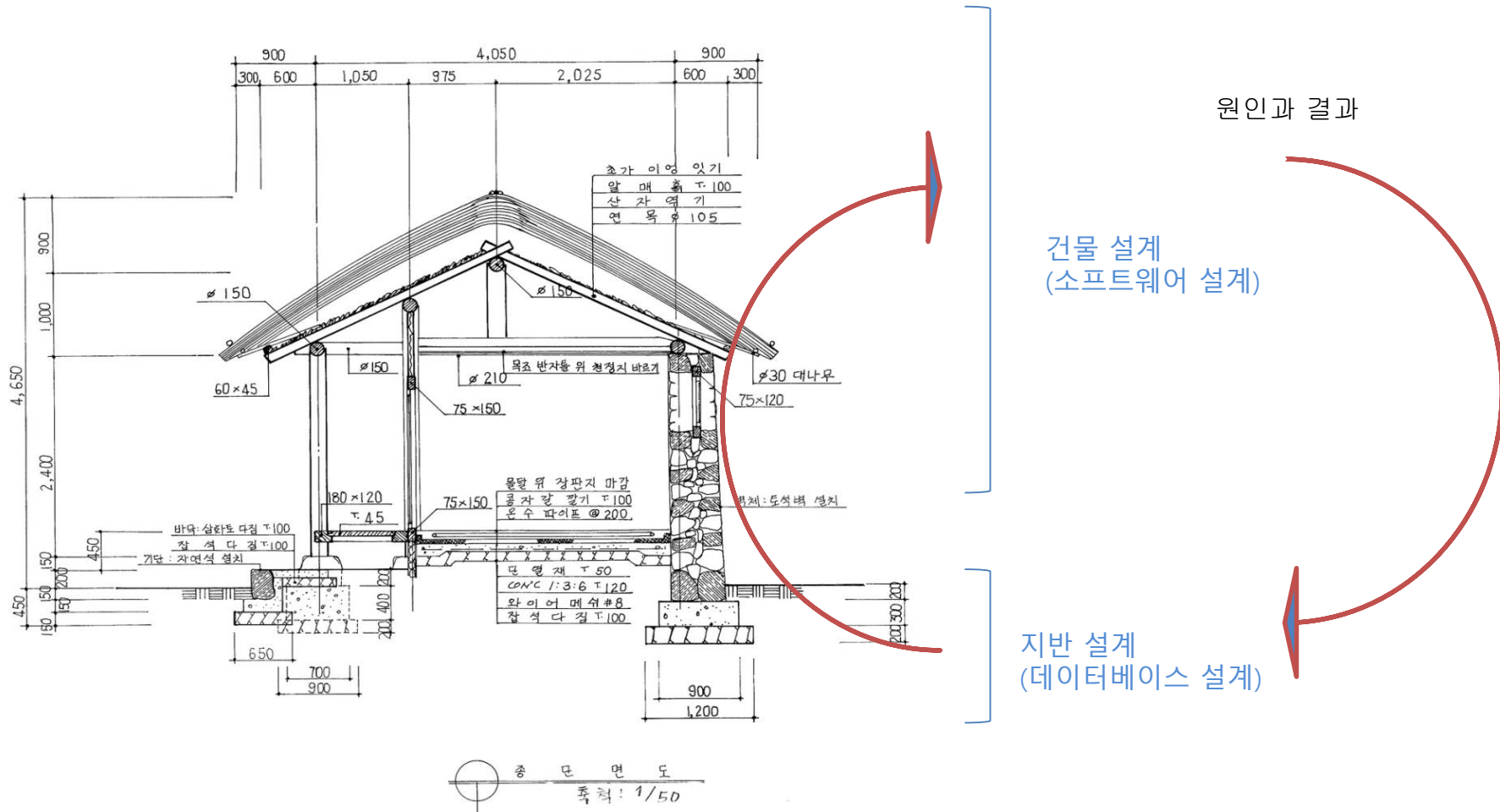


그림 6-1 데이터 모델링의 중요성 --> 순차적으로 점검해 가면서 좋은 모델링을 만들어가야 한다.

# 데이터 모델링의 개념

\* MODEL - 특정 목적을 달성하기 위해 실제 세계를 모방해서 만든 것

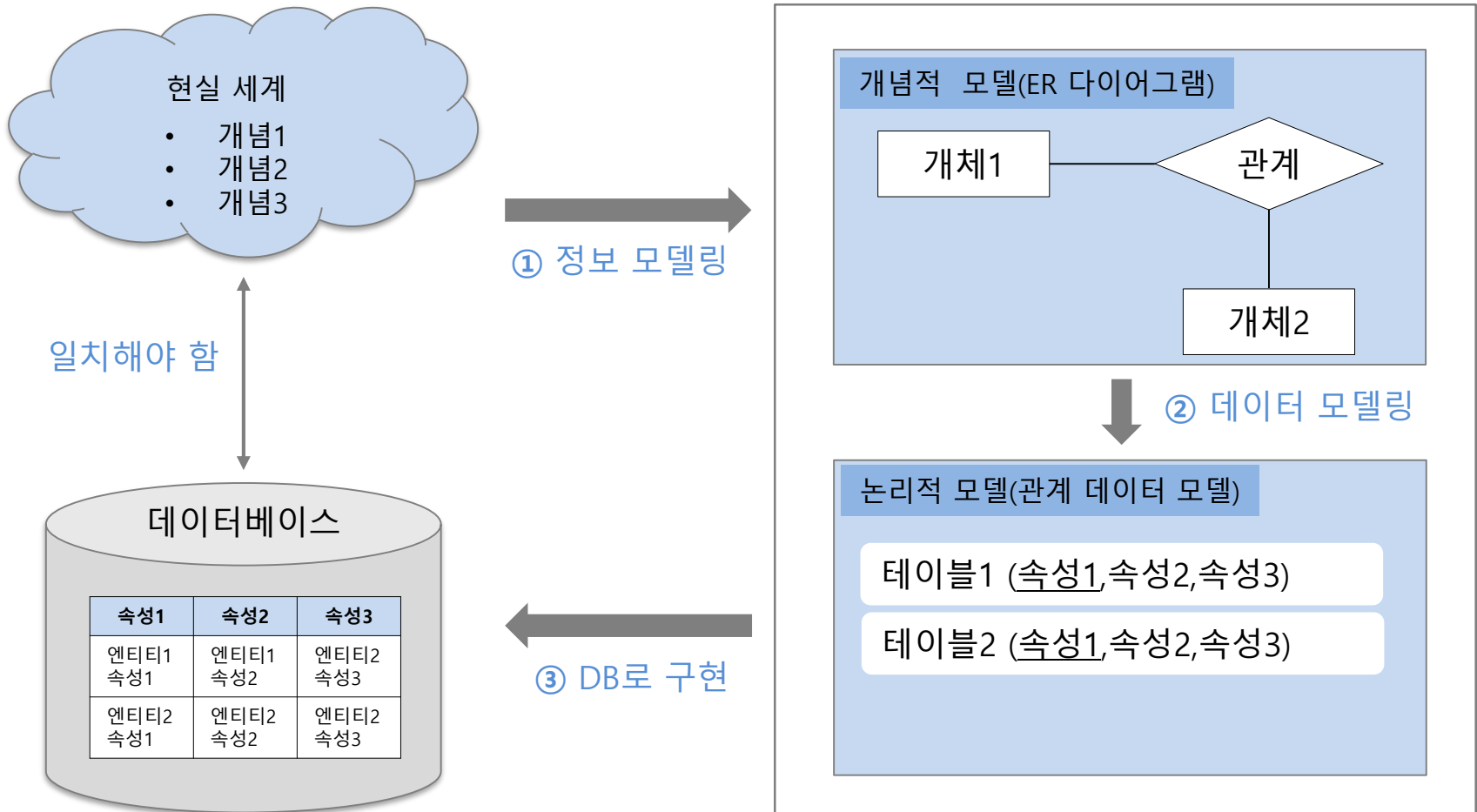


그림 6-2 데이터 모델링의 개념

\* 현실의 복잡성을 컴퓨터에 담을 수 있는 방법론이 데이터 모델링이다.

# 1. 데이터베이스 생명주기

## ❖ 데이터베이스 생명주기(database life cycle)

데이터베이스의 생성과 운영에 관련된 특징

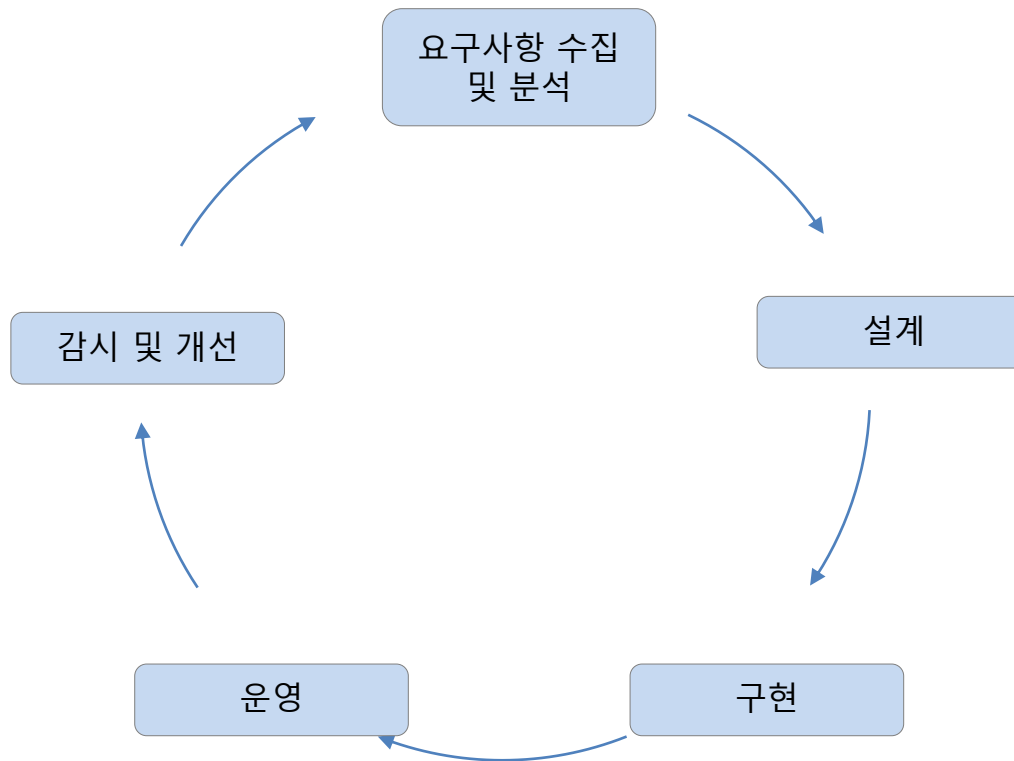


그림 6-3 데이터베이스 생명주기

# 1. 데이터베이스 생명주기

## ❶ 요구사항 수집 및 분석

사용자들의 요구사항을 듣고 분석하여 데이터베이스 구축의 범위를 정함

## ❷ 설계

분석된 요구사항을 기초로 주요 개념과 업무 프로세스 등을 식별하고(개념적 설계), 사용하는 DBMS의 종류에 맞게 변환(논리적 설계)한 후, 데이터베이스 스키마를 도출(물리적 설계)함

## ❸ 구현

설계 단계에서 생성한 스키마를 실제 DBMS에 적용하여 테이블 및 관련 객체(뷰, 인덱스 등)를 만듦

## ❹ 운영

구현된 데이터베이스를 기반으로 소프트웨어를 구축하여 서비스를 제공함

## ❺ 감시 및 개선

데이터베이스 운영에 따른 시스템의 문제를 관찰하고 데이터베이스 자체의 문제점을 파악하여 개선함



## 2. 데이터 모델링 과정

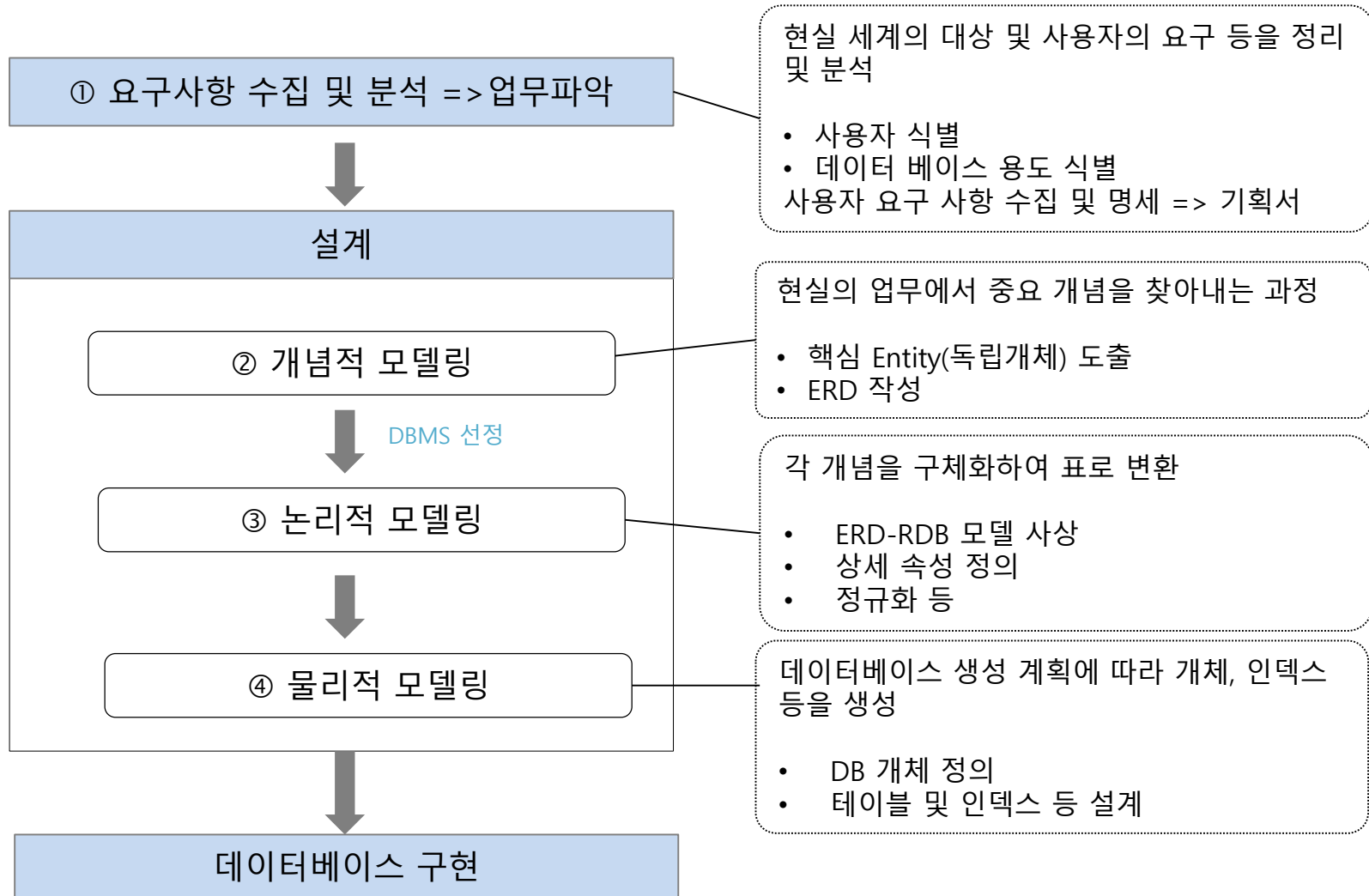


그림 6-4 데이터 모델링 과정

## 2. 데이터 모델링 과정

### ❖ 요구사항 수집 및 분석 => 업무 파악(과정)과 기획서(결과)

#### ■ 요구사항 수집 방법

1. 실제 문서를 수집하고 분석함
  2. 담당자와의 인터뷰나 설문조사를 통해 요구사항을 직접 수렴함
  3. 비슷한 업무를 처리하는 기존의 데이터베이스를 분석함
  4. 각 업무와 연관된 모든 부분을 살펴봄
- 해당분야의 실무자들과 정확하게 의사소통하는 것이 중요하다.
  - 현실적 조언 - SW 엔지니어는 기술적인 공부도 많이 해야 하고, 인간관계도 등한시 할 수 없고, 인문학적 소양도 갖추어야 한다.
  - 업무파악할 때 하는 방법중 좋은 방법은 의뢰인과 UI를 같이 그리는 것이다.

## 2. 데이터 모델링 과정

### ❖ 개념적 모델링

- 요구사항을 수집하고 분석한 결과를 토대로 업무의 핵심적인 개념을 구분하고 전체적인 뼈대를 만드는 과정
- 개체(entity)를 추출하고 각 개체들 간의 관계를 정의하여 ER 다이어그램(ERD, Entity Relationship Diagram)을 만드는 과정까지를 말함

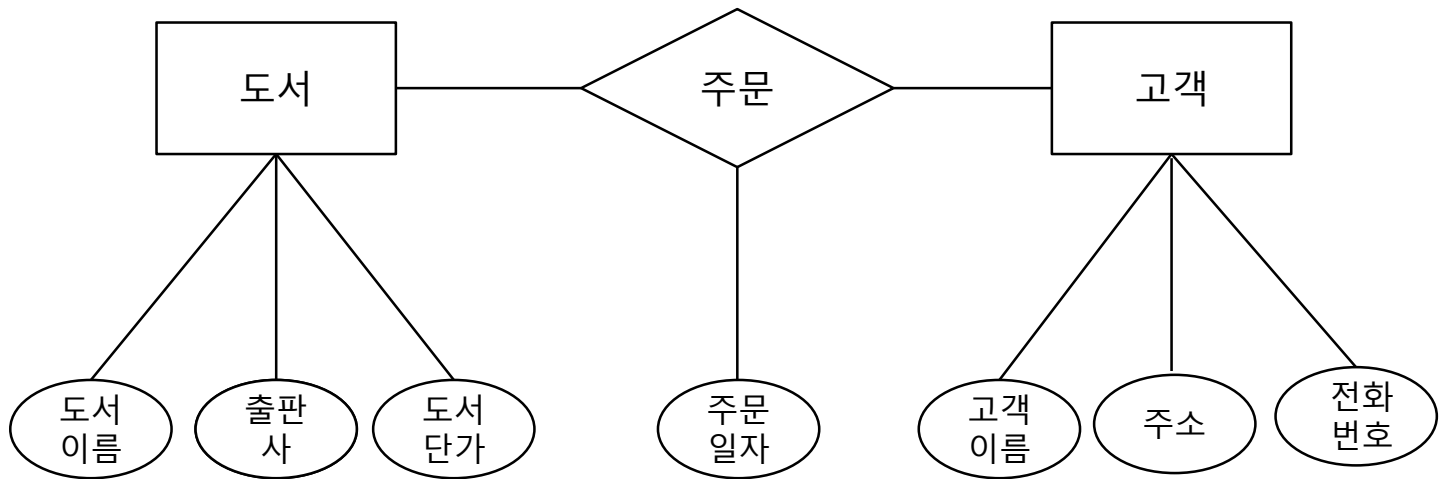


그림 6-5 개념적 모델링의 예

---

□	<b>Entity</b>	➔	<b>Table</b>
○	<b>Attribute</b>	➔	<b>Column</b>
◇	<b>Relation</b>	➔	<b>PK, FK</b>
	<b>Tuple</b>	➔	<b>Row</b>

## 2. 데이터 모델링 과정

### ❖ 논리적 모델링

- 개념적 모델링에서 만든 ER 다이어그램을 사용하려는 DBMS에 맞게 사상(매핑, mapping)하여 실제 데이터베이스로 구현하기 위한 모델을 만드는 과정



도서 (도서번호, 도서이름, 출판사이름, 도서단가)

고객 (고객번호, 고객이름, 주소, 전화번호)

주문 (주문번호, 고객번호(FK), 도서번호(FK), 주문일자, 주문금액)

그림 6-6 논리적 모델링의 예

### ■ 논리적 모델링 과정

1. 개념적 모델링에서 추출하지 않았던 상세 속성들을 모두 추출
2. 정규화 수행
3. 데이터 표준화 수행

□ Entity	➡ Table
○ Attribute	➡ Column
◇ Relation	➡ PK, FK

## 2. 데이터 모델링 과정

### ❖ 물리적 모델링

- 작성된 논리적 모델을 실제 컴퓨터의 저장 장치에 저장하기 위한 물리적 구조를 정의하고 구현하는 과정
- DBMS 특성에 맞게 저장 구조를 정의해야 데이터베이스가 최적의 성능을 낼 수 있음

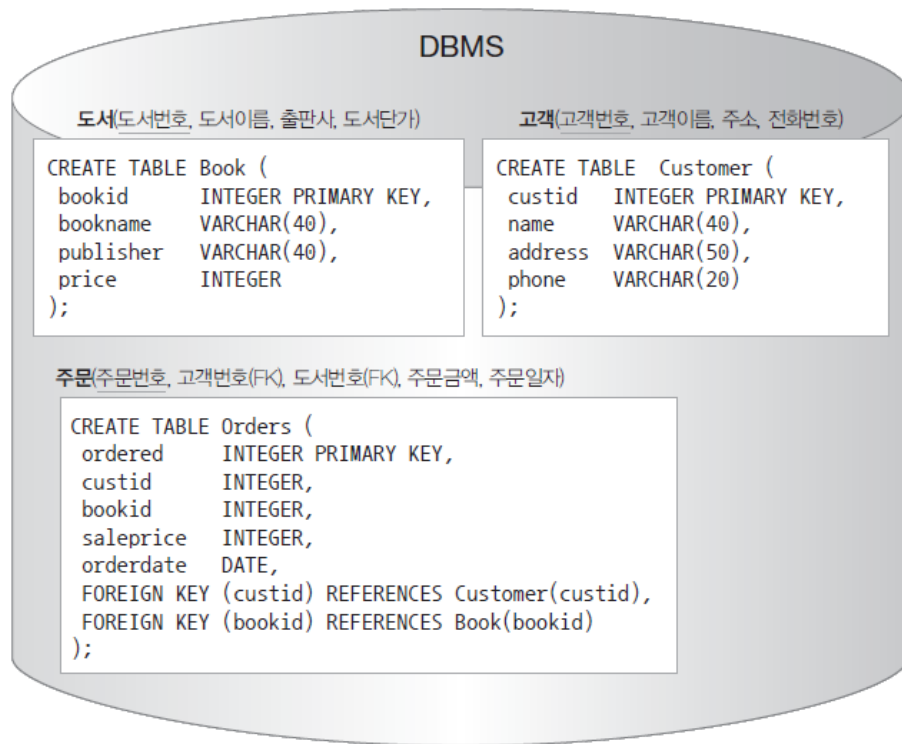


그림 6-7 물리적 모델링의 예

## 2. 데이터 모델링 과정

### ❖ 물리적 모델링

#### ■ 물리적 모델링 시 트랜잭션, 저장 공간 설계 측면에서 고려할 사항

1. 응답시간을 최소화해야 한다.
2. 얼마나 많은 트랜잭션을 동시에 발생시킬 수 있는지 검토해야 한다.
3. 데이터가 저장될 공간을 효율적으로 배치해야 한다.

## 02. ER 모델

1. 개체와 개체 타입
2. 속성
3. 관계와 관계 타입
4. 약한 개체 타입과 식별자
5. IE 표기법





# 기획서에서 ERD를 만드는 과정

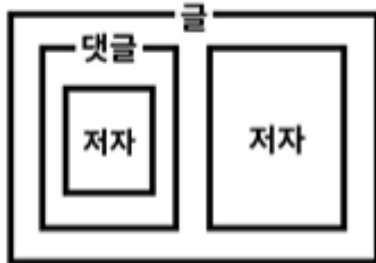


RDB는 내포관계를 허용하지 않습니다.

글 제목			글 내용			글 저자			댓글					
글 제목	글 내용								댓글 내용		작성일		저자	
									댓글 1 내용	댓글 1 작성일				
											저자 이름	저자 소개	저자 가입일	
			저자 이름	저자 소개	저자 가입일							저자 1	저자 1 소개	저자 1 가입일
			저자 1	저자 1 소개	저자 1 가입일									
			저자 2	저자 2 소개	저자 2 소개									

거대 단일 테이블로 표현을 하면 중복이 발생합니다.

글 제목	글 내용	글 저자 이름	글 저자 소개	글 저자 가입일	댓글 내용	댓글 저자	댓글저자소개	댓글저자가가입일
글 1 제목	글 1 내용	저자 1 이름	저자 1 소개	저자 1 가입일	댓글 1 내용	저자 1 이름	저자 1 소개	저자 1 가입일
글 2 제목	글 2 내용	저자 1 이름	저자 1 소개	저자 1 가입일	댓글 2 내용	저자 1 이름	저자 1 소개	저자 1 가입일
글 3 제목	글 3 내용	저자 2 이름	저자 2 소개	저자 2 가입일	댓글 3 내용	저자 2 이름	저자 2 소개	저자 2 가입일



글				저자			댓글				
아이디	제목	내용	저자 아이디	아이디	이름	소개	아이디	제목	내용	작성일	저자 아이디
1	제목 1	내용 1	①	1	이름 1	소개 1	1	제목 1	내용 1	작성일 1	①
2	제목 2	내용 2	①	2	이름 2	소개 2	2	제목 2	내용 2	작성일 2	①
3	제목 3	내용 3	①	3	이름 3	소개 3	3	제목 3	내용 3	작성일 3	②

SELECT 댓글.내용, 댓글.작성일, 저자.이름, 저자.소개 FROM 댓글 LEFT JOIN 저자 ON 댓글.저자 아이디 = 저자.아이디



댓글 내용	댓글 작성일	저자	저자 소개
댓글 1 내용	댓글 1 작성일	저자 1 이름	저자 1 자기소개
댓글 2 내용	댓글 2 작성일	저자 1 이름	저자 1 자기소개
댓글 3 내용	댓글 3 작성일	저자 1 이름	저자 1 자기소개

# ER 모델

## ■ ER(Entity Relationship) 모델

- 세상의 사물을 개체(entity)와 개체 간의 관계(relationship)로 표현함

## ■ 개체

- 독립적인 의미를 지니고 있는 유무형의 사람 또는 사물
- 개체의 특성을 나타내는 속성(attribute)에 의해 식별됨. 개체끼리 서로 관계를 가짐

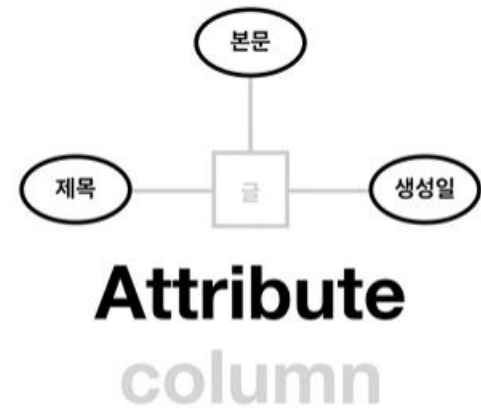
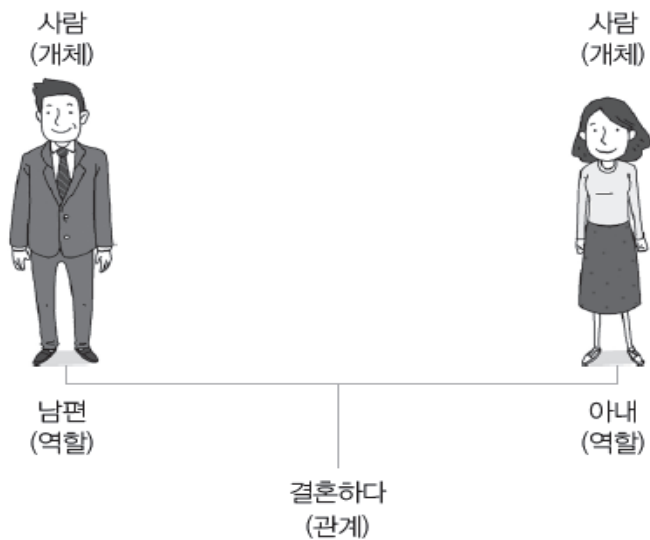


그림 6-8 ER 모델의 기본 개념

# ER 모델

## ❖ ER 다이어그램

ER 모델은 개체와 개체 간의 관계를 표준화된 그림으로 나타냄

개체-관계

속성

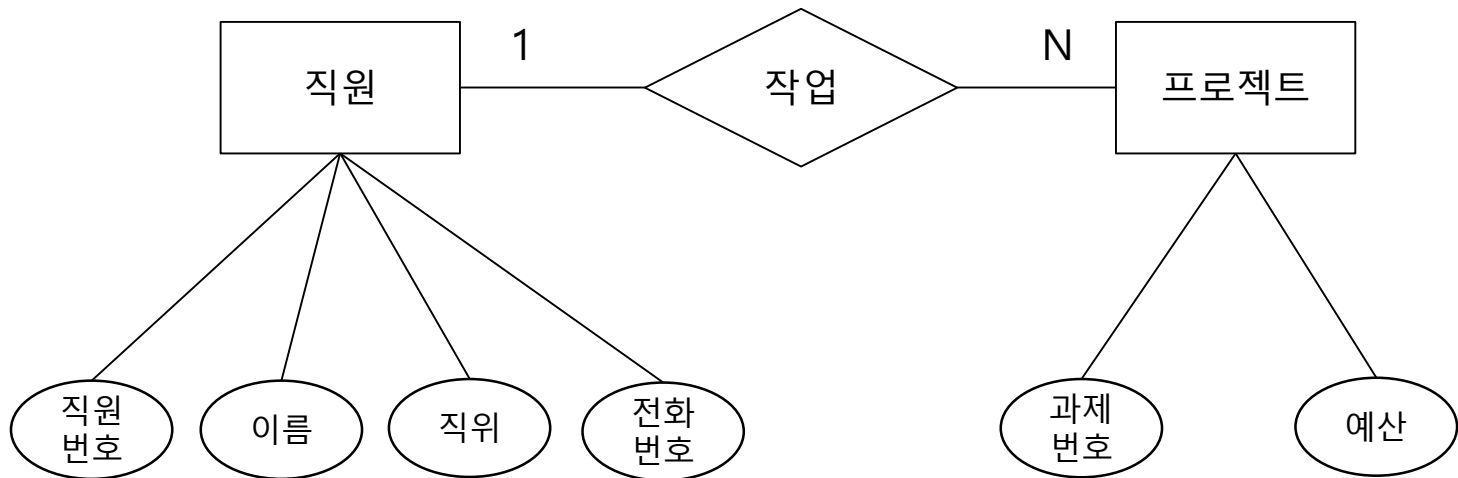


그림 6-9 ER 다이어그램

# 1. 개체와 개체 타입

## ❖ 개체(entity)란?

- 사람, 사물, 장소, 개념, 사건과 같이 유무형의 정보를 가지고 있는 독립적인 실체
- 데이터베이스에서 주로 다루는 개체는 낱개로 구성된 것, 낱개가 각각 데이터 값을 가지는 것, 데이터 값이 변하는 것 등이 있음.
- 비슷한 속성의 개체 타입(entity type)을 구성하며, 개체 집합(entity set)으로 묶임.

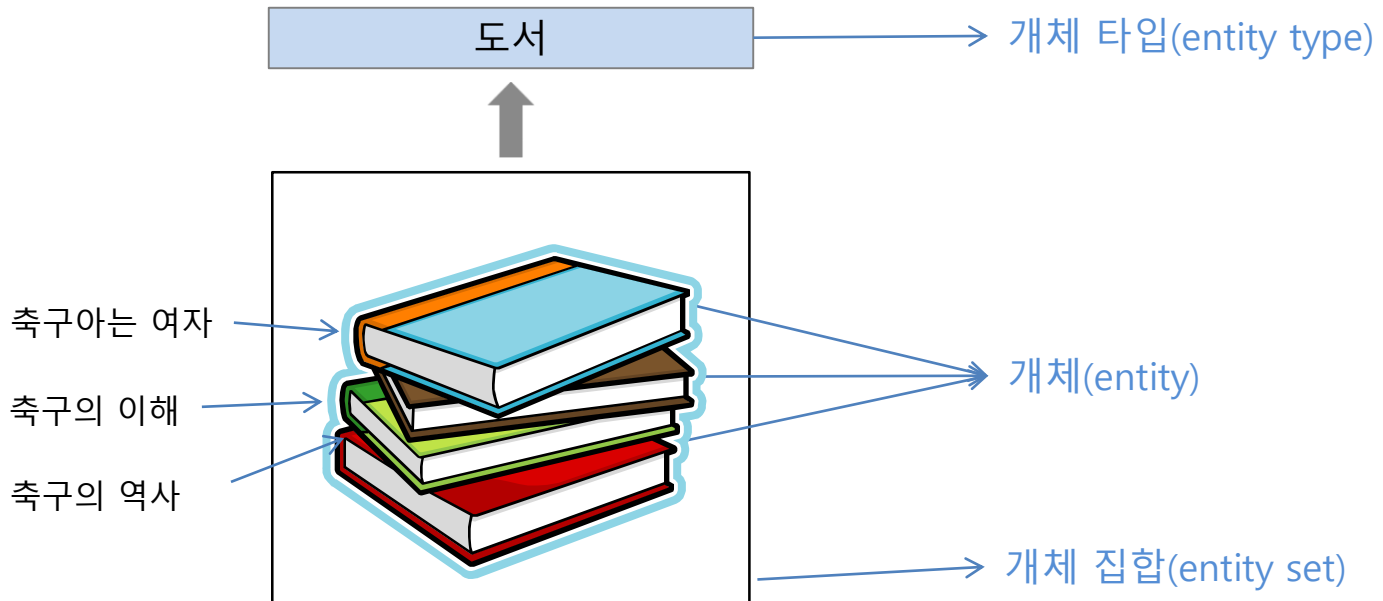


그림 6-10 개체, 개체 타입, 개체 집합

# draw.io 도구 활용

이름  
소속  
저자등록

저자

제목  
분류  
저자등록  
글등록

글

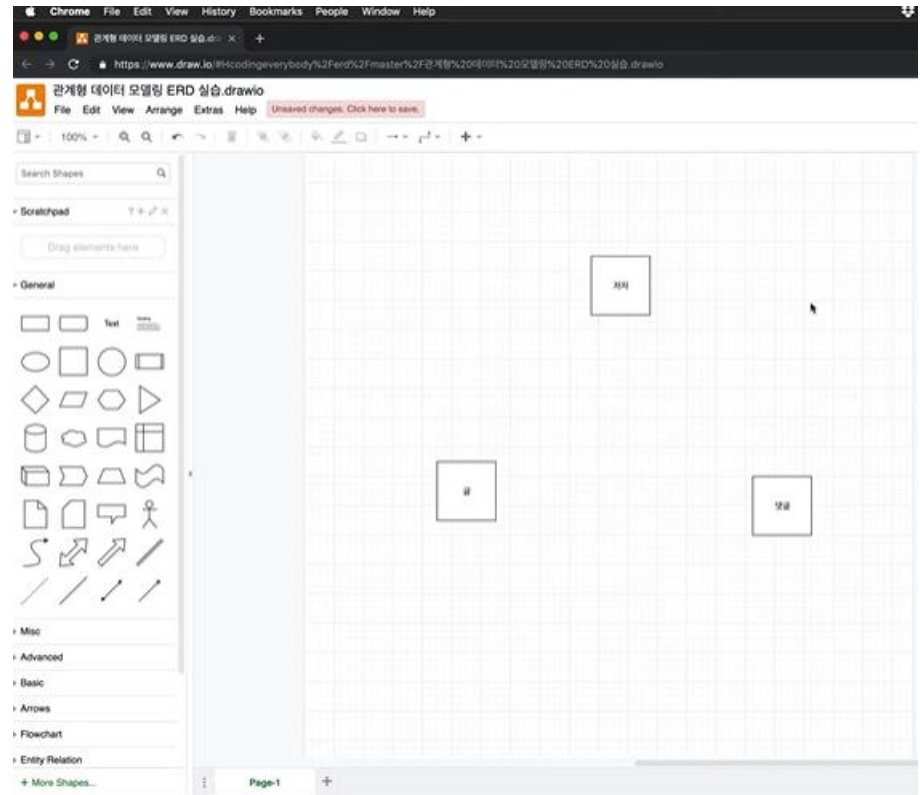
내용을 입력해주세요  
내용을 입력해주세요  
댓글작성

댓글

저자

글

댓글


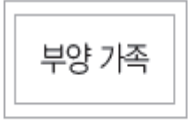


# 1. 개체와 개체 타입

## ❖ 개체 타입의 ER 다이어그램 표현

- ER 다이어그램상에서 개체 타입은 직사각형으로 나타냄

표 6-1 개체 타입의 ER 다이어그램 표현

기호	의미
 직원	강한 개체 타입(보통 개체 타입이라고 하면 강한 개체 타입을 말한다)
 부양 가족	약한 개체 타입

### ■ 개체 타입의 유형

- 강한 개체(strong entity) : 다른 개체의 도움 없이 독자적으로 존재할 수 있는 개체
- 약한 개체(weak entity) : 독자적으로는 존재할 수 없고 반드시 상위 개체 타입을 가짐

## 2. 속성

### ■ 속성(attribute) : 개체가 가진 성질

개체 타입	속성
도서	도서이름, 출판사, 도서단가

표 6-2 개체 타입과 속성

### ■ 속성의 ER 다이어그램 표현

- 속성은 기본적으로 **타원**으로 표현. 개체 타입을 나타내는 직사각형과 **실선**으로 연결됨
- 속성의 이름은 타원의 중앙에 표기함
- 속성이 개체를 유일하게 식별할 수 있는 키일 경우 속성 이름에 밑줄을 그음

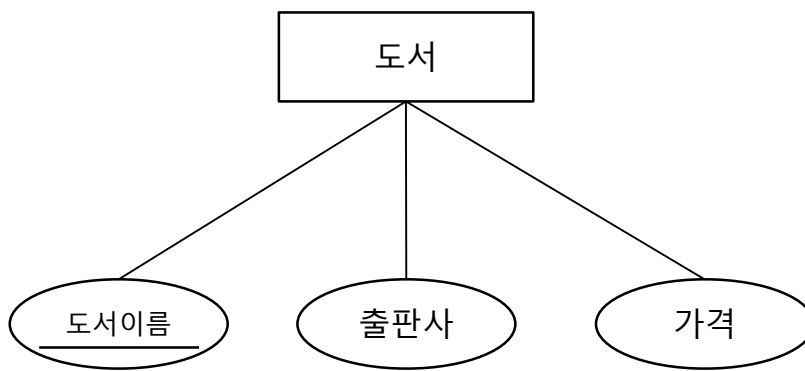


그림 6-11 도서 개체 타입

## 2. 속성

### ❖ 속성의 유형

기호	의미	설명
	속성	<ul style="list-style-type: none"> <li>일반적인 속성을 나타냄</li> <li>속성의 이름은 타원 중앙에 표시</li> </ul>
	키(key) 속성	<ul style="list-style-type: none"> <li>속성이 개체를 유일하게 식별할 수 있는 키일 경우 속성 이름에 밑줄을 그음</li> </ul>
	약한 개체의 식별자	<ul style="list-style-type: none"> <li>약한 개체는 키를 갖지 못하고 대신 식별자를 가짐</li> <li>식별자의 아래에 점선을 그음</li> </ul>
	다중값 속성	<ul style="list-style-type: none"> <li>취미(수영, 자전거)와 같이 여러 개의 값을 갖는 속성</li> <li>이중 타원으로 표현</li> </ul>
	유도 속성	<ul style="list-style-type: none"> <li>나이와 같이 출생년도로 유도가 가능한 속성</li> <li>점선 타원으로 표현</li> </ul>
	복합 속성	<ul style="list-style-type: none"> <li>주소(시, 동, 번지)와 같이 여러 속성으로 구성된 속성</li> <li>큰 타원 아래 작은 타원으로 연결</li> </ul>

표 6-3 속성의  
ER 다이어그램 표현



### 3. 관계와 관계 타입

- 관계(relationship) : 개체 사이의 연관성을 나타내는 개념
- 관계 타입(relationship type) : 개체 타입과 개체 타입 간의 연결 가능한 관계를 정의한 것이며, 관계 집합(relationship set)은 관계로 연결된 집합을 의미함

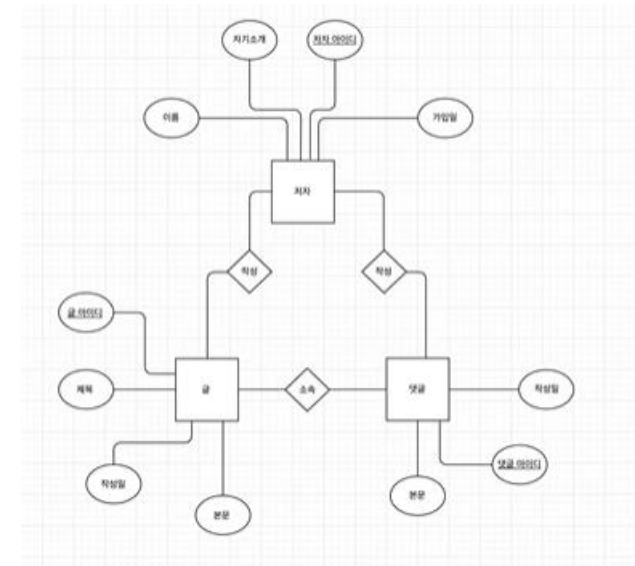
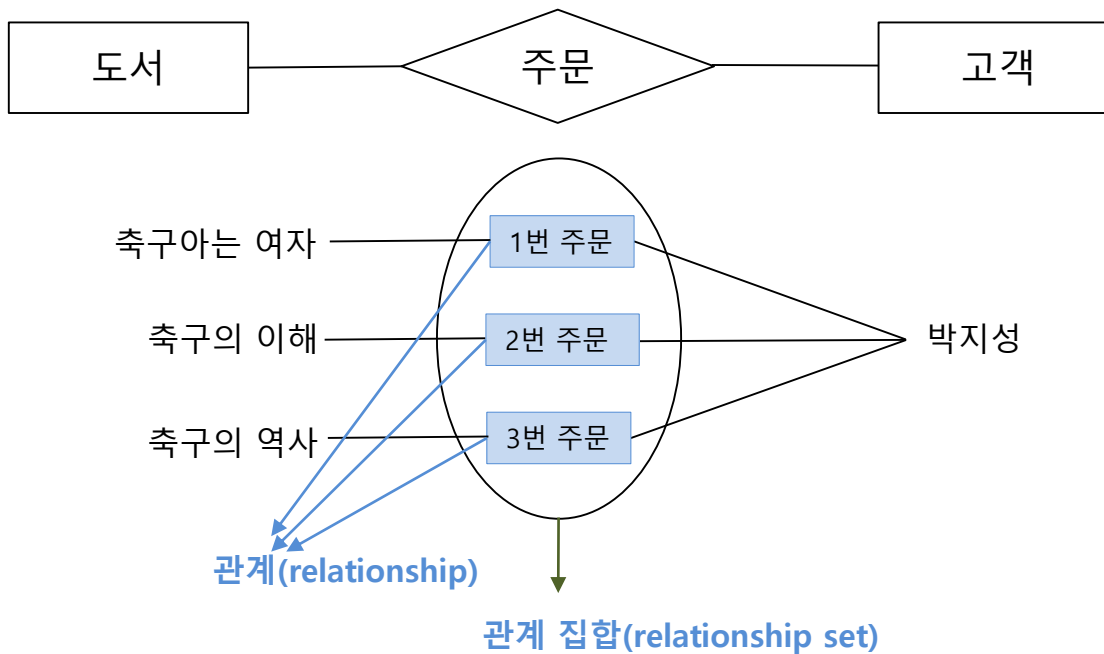



그림 6-15 관계, 관계 타입, 관계 집합

### 3. 관계와 관계 타입

#### ❖ 관계 타입의 ER 다이어그램 표현

표 6-4 관계 타입의 ER 다이어그램 표현

기호	의미
	관계 타입

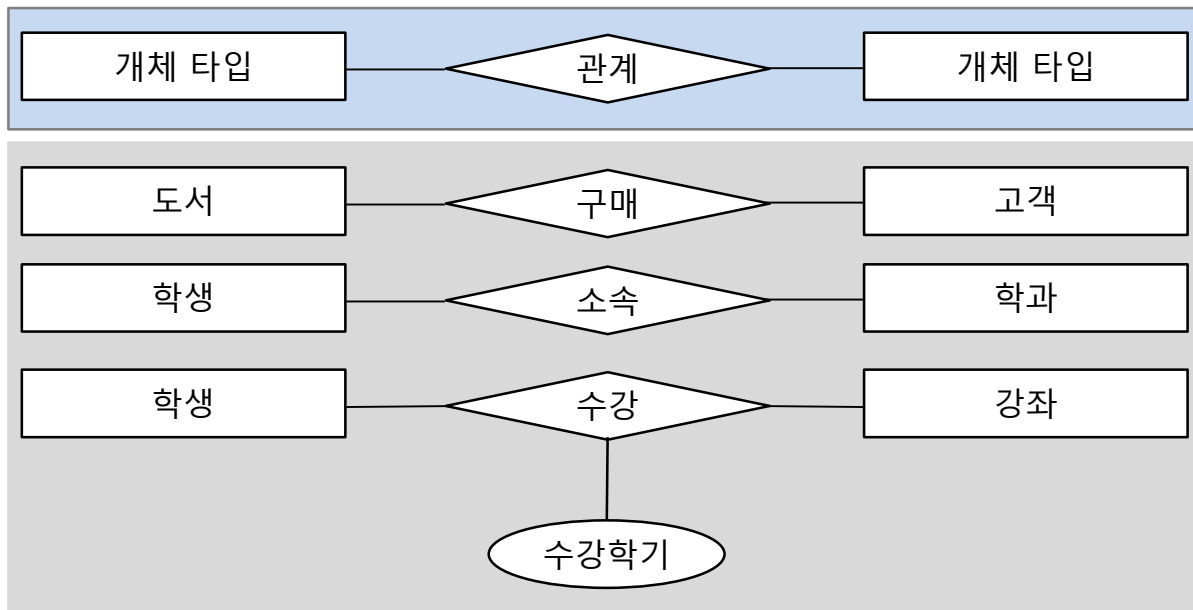


그림 6-16 관계의 예


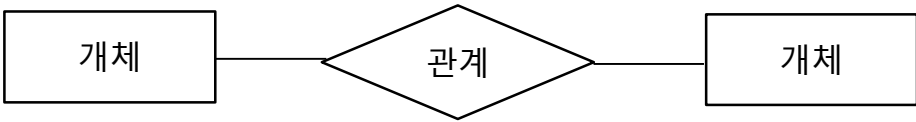
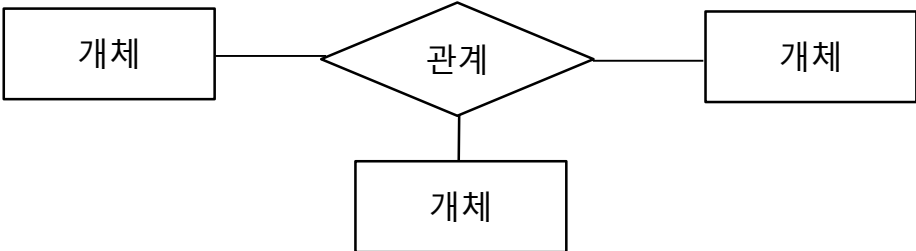
### 3. 관계와 관계 타입

#### ❖ 관계 타입의 유형

##### ■ 차수에 따른 유형

관계 집합에 참가하는 개체 타입의 수를 관계 타입의 차수(degree)라고 함

표 6-5 차수에 따른 관계 타입의 유형

기호	의미	설명
	1진 관계	한 개의 개체가 자기 자신과 관계를 맺음
	2진 관계	두 개의 개체가 관계를 맺음
	3진 관계	세 개의 개체가 관계를 맺음

### 3. 관계와 관계 타입

#### ❖ 관계 타입의 유형

❶ 1진 관계(recursive relationship) : 한 개의 개체가 자기 자신과 관계를 맺는 경우



그림 6-17 1진 관계의 예

❷ 2진 관계(binary relationship) : 두 개의 개체가 관계를 맺는 경우

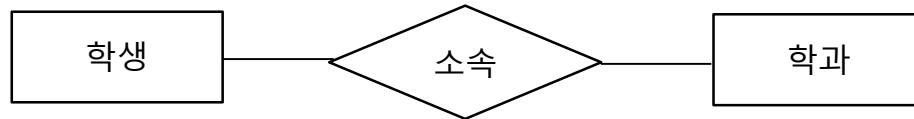


그림 6-18 2진 관계의 예

### 3. 관계와 관계 타입

#### ❖ 관계 타입의 유형

③ 3진 관계(ternary relationship) : 세 개의 개체가 관계를 맺는 경우

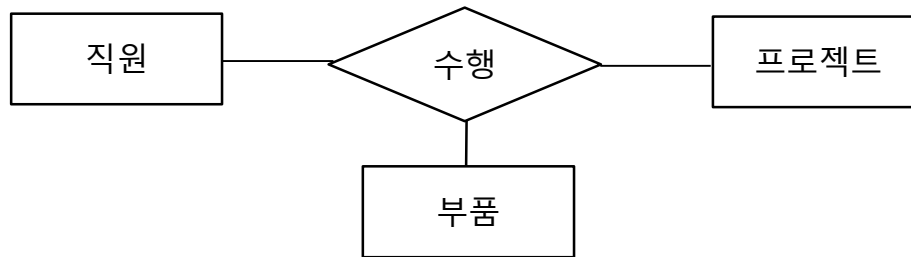


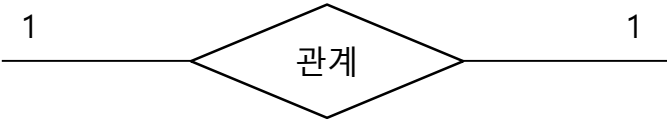
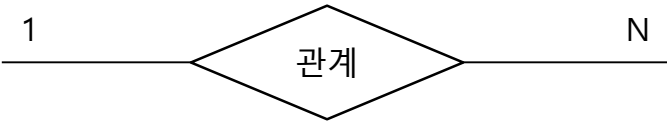
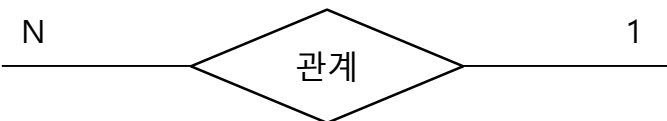
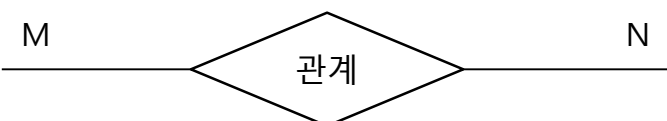
그림 6-19 3진 관계의 예

### 3. 관계와 관계 타입

#### ❖ 관계 타입의 유형

- 관계 대응수(cardinality) : 두 개체 타입의 관계에 실제로 참여하는 개별 개체 수

표 6-6 관계 대응수에 따른 관계 타입의 유형

기호	의미	설명
	일대일 관계	하나의 개체가 하나의 개체에 대응
	일대다 관계	하나의 개체가 여러 개체에 대응
	다대일 관계	여러 개체가 하나의 개체에 대응
	다대다 관계	여러 개체가 여러 개체에 대응

### 3. 관계와 관계 타입

#### ❖ 관계 타입의 유형

##### ❶ 일대일(1:1)관계

좌측 개체 타입에 포함된 개체가 우측 개체 타입에 포함된 개체와 일대일로 대응하는 관계

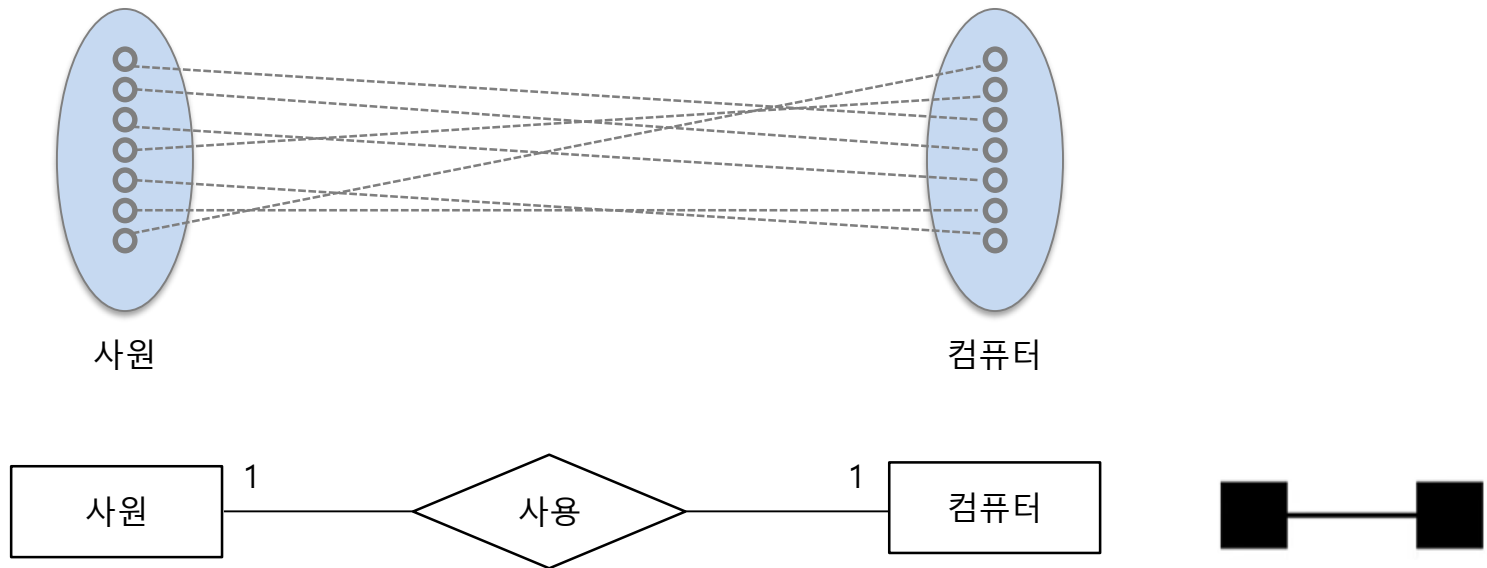


그림 6-20 일대일 관계의 예

### 3. 관계와 관계 타입

#### ❖ 관계 타입의 유형

##### ② 일대다(1:N), 다대일(N:1) 관계

실제 일상생활에서 가장 많이 볼 수 있는 관계로, 한쪽 개체 타입의 개체 하나가 다른 쪽 개체 타입의 여러 개체와 관계를 맺음

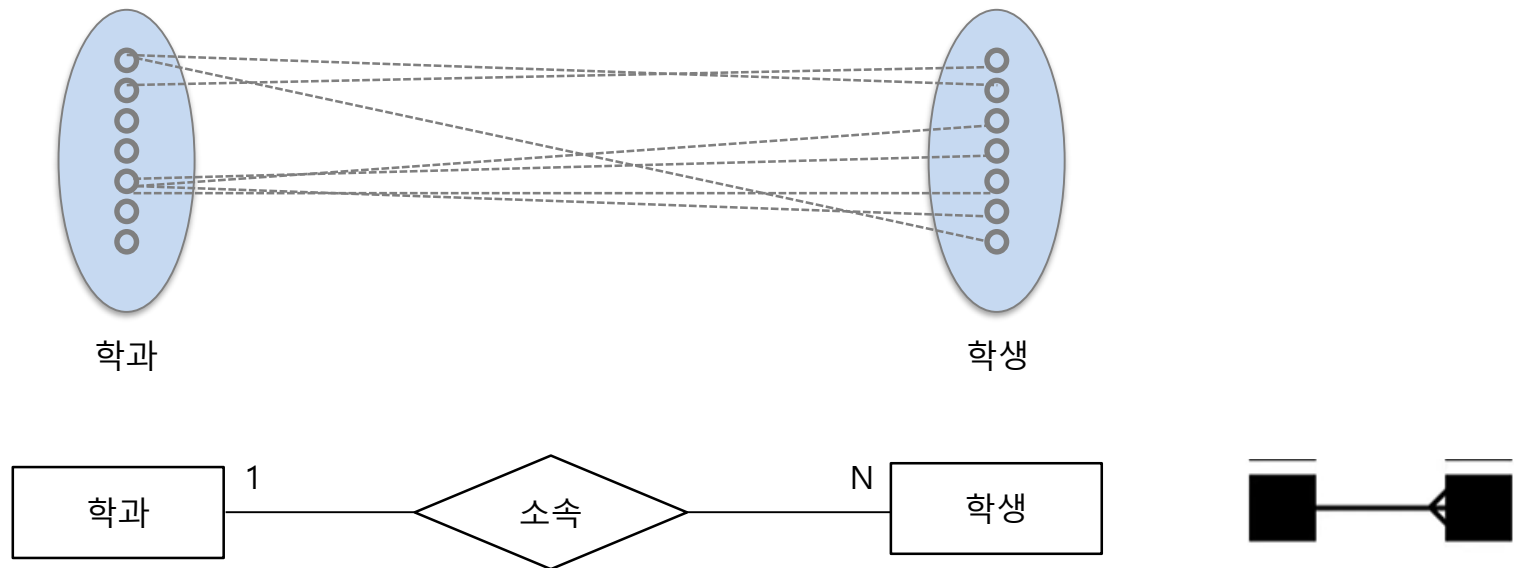


그림 6-21 일대다(1:N), 다대일(N:1) 관계의 예



### 3. 관계와 관계 타입

#### ❖ 관계 타입의 유형

##### ③ 다대다(N:M) 관계

각 개체 타입의 개체들이 서로 임의의 개수의 개체들과 서로 복합적인 관계를 맺고 있는 관계

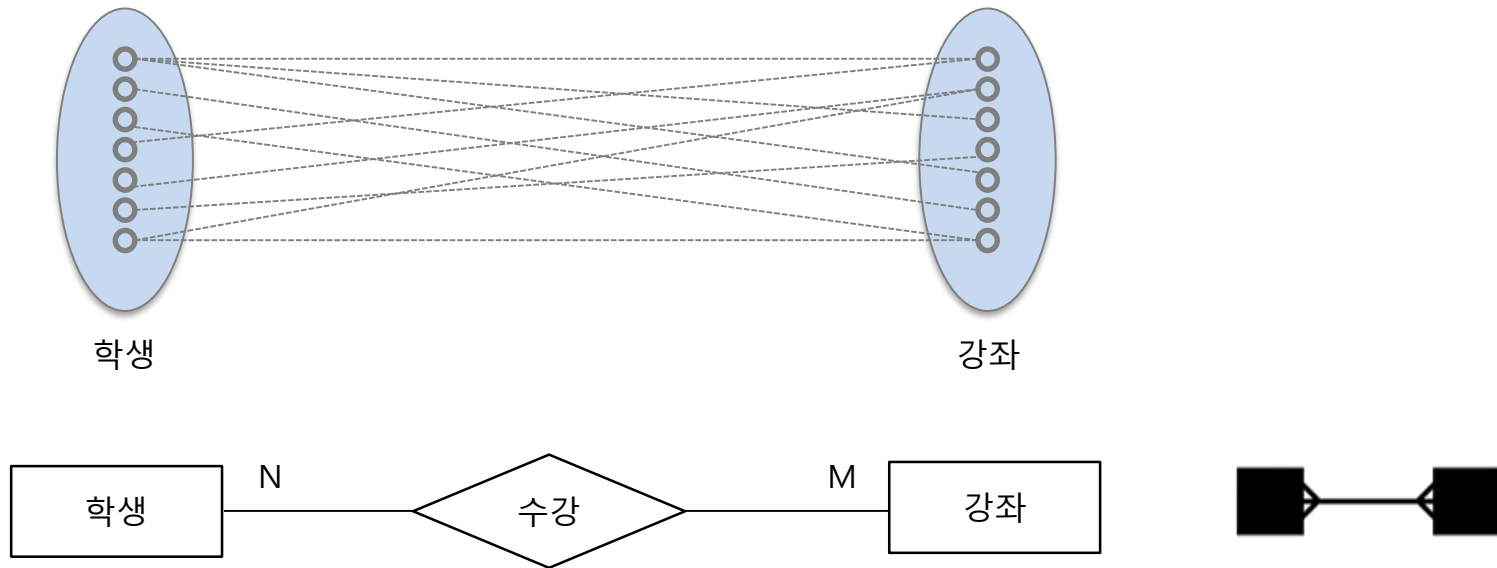


그림 6-22 다대다(N:M) 관계의 예

다대다 관계는 현실의 db 테이블의 세계에서는 표현할 수 없어서  
중간에 연결테이블이라는 특별한 테이블을 만들어서

1:N 으로 컨버트 해서 표현한다.

### 3. 관계와 관계 타입

#### ❖ 관계 대응수의 최솟값과 최댓값(Optionality)

- 관계 대응수 1:1, 1:N, M:N에서 1, N, M은 각 개체가 관계에 참여하는 최댓값을 의미함
- 관계에 참여하는 개체의 최솟값을 표시하지 않는다는 단점을 보완하기 위해 다이어그램에서는 대응수 외에 최솟값과 최댓값을 관계실선 위에 (최솟값, 최댓값)으로 표기함

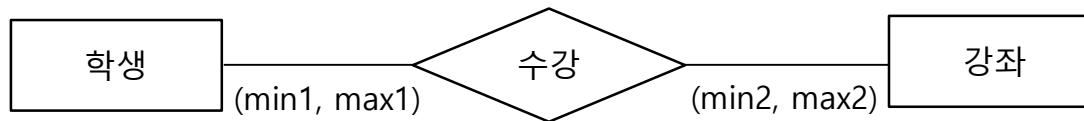


그림 6-23 관계 대응수의 최솟값과 최댓값의 표기

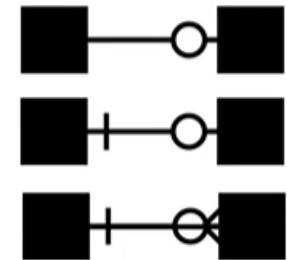


표 6-6 관계 대응수에 따른 관계 타입의 유형

관계	(min1,max1)	(min2,max2)
1:1	(0, 1)	(0, 1)
1:N	(0, *)	(0, 1)
M:N	(0, *)	(0, *)

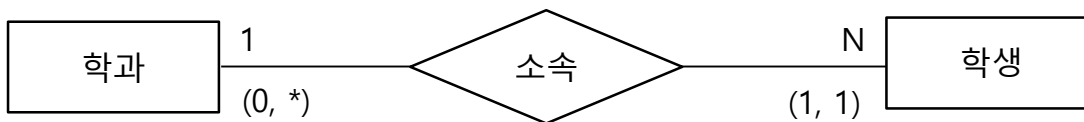



그림 6-24 (최솟값, 최댓값) 표기의 예

### 3. 관계와 관계 타입

#### ❖ ISA 관계

- 상위 개체 타입의 특성에 따라 하위 개체 타입이 결정되는 형태

표 6-8 ISA 관계 (ISA => is-a )

기호	의미
	개체 간 상하관계

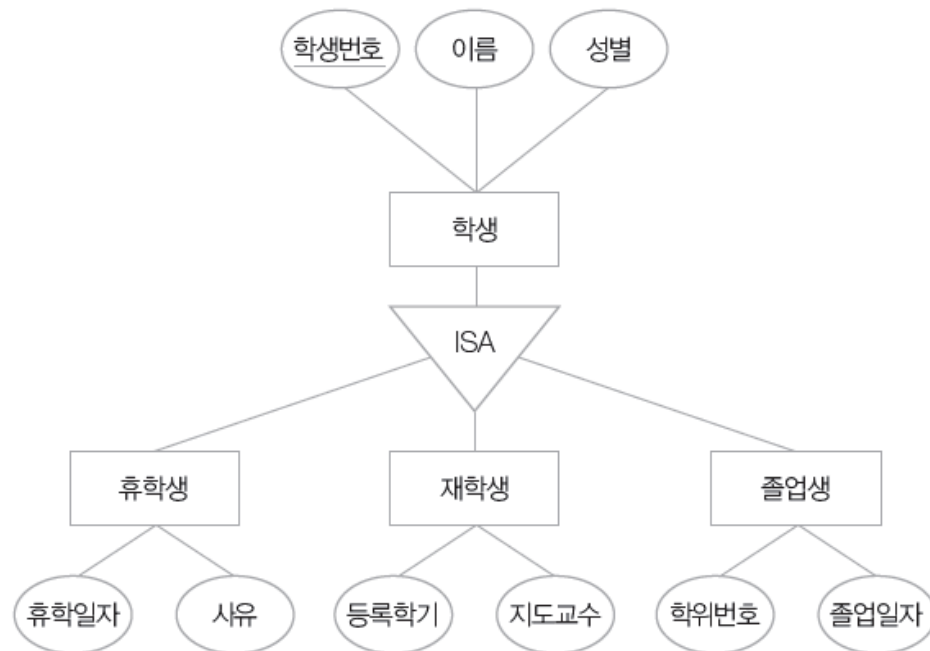


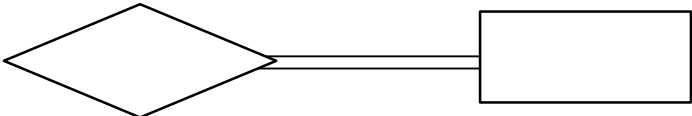
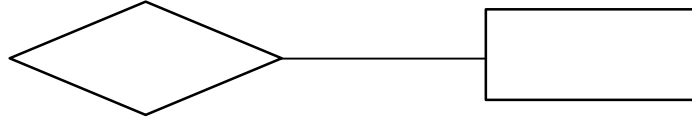
그림 6-25 ISA 관계의 예

### 3. 관계와 관계 타입

#### ❖ 참여 제약 조건

- 개체 집합 내 모든 개체가 관계에 참여하는지 유무에 따라 전체 참여와 부분 참여로 구분 가능
- 전체 참여는 개체 집합의 모든 개체가, 부분 참여는 일부만 참여함
- 전체 참여를 (최솟값, 최댓값)으로 표현할 경우 최솟값이 1 이상으로 모두 참여한다는 뜻이고, 부분 참여는 최솟값이 0 이상임

표 6-9 관계의 참여 제약 조건

기호	의미
	전체 참여
	부분 참여

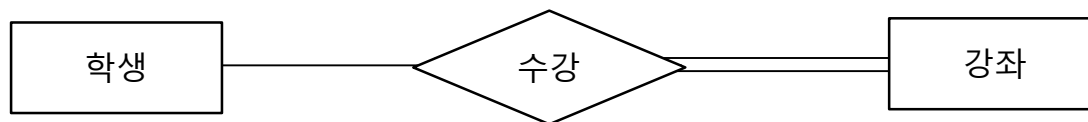


그림 6-26 부분 참여와 전체 참여의 예

### 3. 관계와 관계 타입

#### ❖ 역할

- 개체 타입 간의 관계를 표현할 때 각 개체들은 고유한 역할(role) 담당

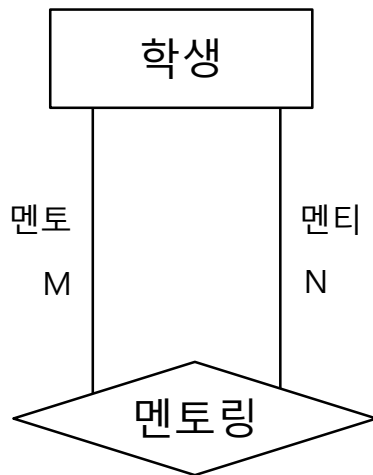


그림 6-27 역할의 예

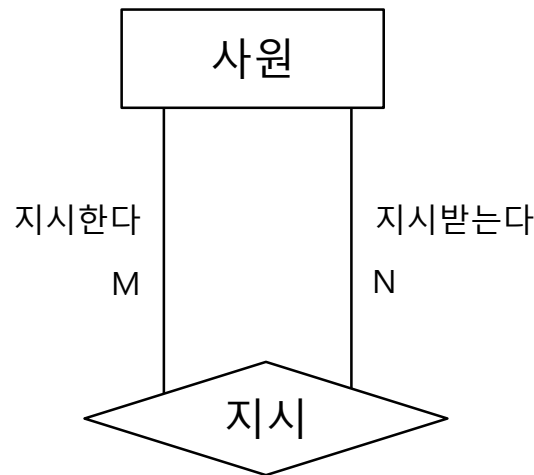
### 3. 관계와 관계 타입

#### ❖ 순환적 관계

- 순환적 관계(recursive relationship) : 하나의 개체 타입이 동일한 개체 타입(자기 자신)과 순환적으로 관계를 가지는 형태.



(a) 학생의 멘토링 관계



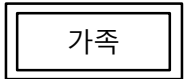
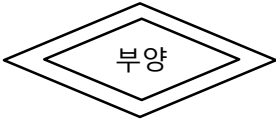
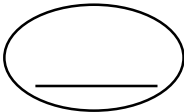
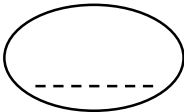
(b) 사원의 지시 관계

그림 6-28 순환적 관계의 예

## 4. 약한 개체 타입과 식별자

- 약한 개체(weak entity) 타입 : 상위 개체 타입이 결정되지 않으면 개별 개체를 식별할 수 없는 종속된 개체 타입
- 약한 개체 타입은 독립적인 키로는 존재할 수 없지만 상위 개체 타입의 키와 결합하여 약한 개체 타입의 개별 개체를 고유하게 식별하는 속성을 식별자(discriminator) 혹은 부분키(partial key)라고 함

표 6-10 식별자와 약한 개체 타입

기호	의미	설명
	약한 개체 타입	<ul style="list-style-type: none"><li>• 강한 개체 타입이 있어야 존재할 수 있음</li><li>• 이중 직사각형으로 표현</li></ul>
	식별 관계 타입	<ul style="list-style-type: none"><li>• 강한 개체 타입과 약한 개체 타입의 관계를 나타냄</li><li>• 강한 개체 타입의 기본키를 상속받아 사용함</li><li>• 이중 마름모꼴로 표현</li></ul>
	키	<ul style="list-style-type: none"><li>• 강한 개체 타입의 키 속성</li></ul>
	식별자	<ul style="list-style-type: none"><li>• 약한 개체 타입에서 개별 개체를 구분하는 속성</li><li>• 키라고 하지 않고 식별자라고 부름</li></ul>

## 4. 약한 개체 타입과 식별자

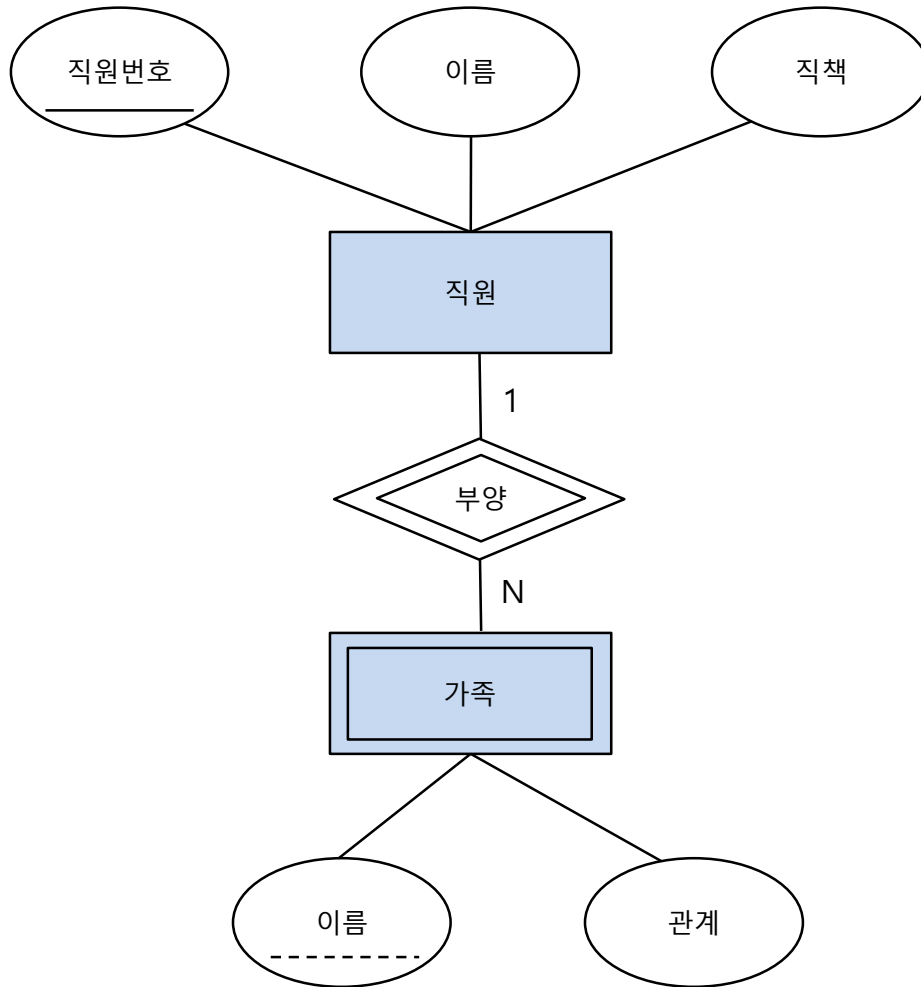


그림 6-29 약한 개체 타입과 식별자의 예



## 5. IE 표기법

- ER 다이어그램을 더 축약하여 쉽게 표현하면 Erwin 등 소프트웨어에서 사용함
- IE(Information Engineering) 표기법에서 개체 타입과 속성은 직사각형으로 표현함

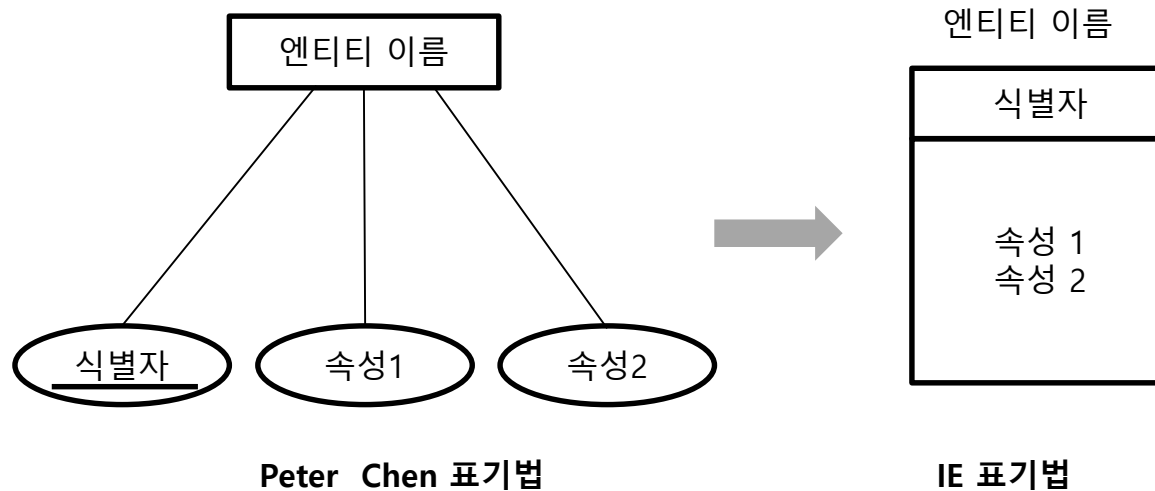


그림 6-30 Peter Chen 표기법과 IE 표기법

## 5. IE 표기법

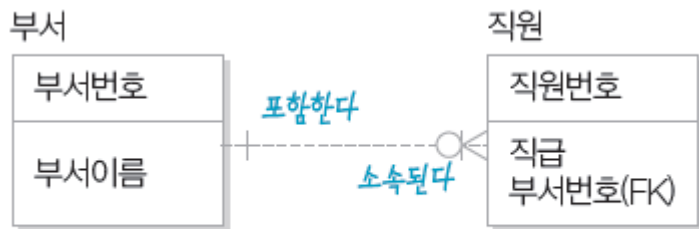
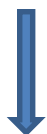
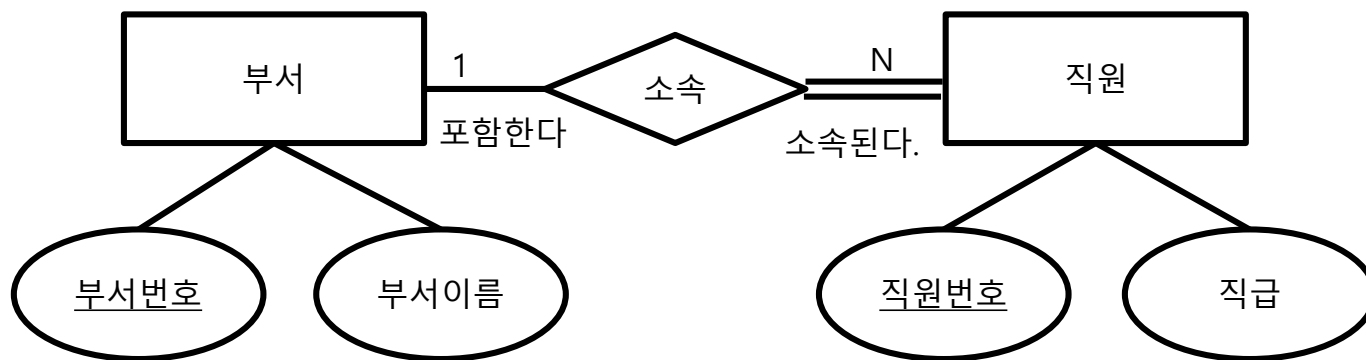
### ■ IE 표기법에서 관계는 실선 혹은 점선으로 표기함

그림 6-11 IE 표기법 - 관계와 관계 대응수

기호	의미
-----	<ul style="list-style-type: none"><li>• 비식별자 관계(non-identifying relationship): 강한 개체 타입</li><li>• 부모 개체의 키가 일반 속성으로 포함되는 관계</li></ul>
————	<ul style="list-style-type: none"><li>• 식별자 관계(identifying relationship): 약한 개체 타입</li><li>• 부모 개체의 키가 주식별자로 포함되는 관계</li></ul>
————<	<ul style="list-style-type: none"><li>• 일대다(1:N)의 관계: N 쪽에 새발을 표시</li></ul>
————○	<ul style="list-style-type: none"><li>• 0(선택 참여), 최소 참여가 0일 경우</li></ul>
————+	<ul style="list-style-type: none"><li>• 1(필수 참여), 최소 참여가 1일 경우</li></ul>

## 5. IE 표기법

- IE 표기법에서 관계(강한관계, 비식별자 관계)는 점선으로 표기함



(b) IE 표기법으로 작성한 직원-부서 관계

그림 6-31 IE 표기법의 예(비식별자 관계)

- IE 표기법에서 관계(약한관계, 식별자 관계)는 실선으로 표기함



그림 6-32 IE 표기법의 예(식별자 관계)

# 연습문제





## 02 ER 모델의 표현 방법으로 옳지 않은 것은?

- ① 개체 집합-사각형
- ② 관계 집합-마름모
- ③ 속성-오각형
- ④ 다중값 속성-이중타원형

## 03 ER 모델에 대한 설명으로 옳지 않은 것은?

- ① ER 다이어그램으로 표현하며 피터 첸이 제안하였다.
- ② 일대일(1:1) 관계 유형만 표현할 수 있다.
- ③ 개체 타입과 이들 간의 관계 타입을 이용해 현실 세계를 개념적으로 표현한다.
- ④ ER 다이어그램은 ER 모델을 그래프 방식으로 표현한 것이다.

## 04 ER 표기법에 대한 설명 중 옳지 않은 것은?

- ①  도서번호 키(key) 속성
- ②  부양가족 약한 개체의 식별자
- ③  취미 복합 속성
- ④  나이 유도 속성

## 03. ER 모델을 관계 데이터 모델로 사상

1. 개체 타입의 사상
2. 관계 타입의 사상
3. 다중값 속성의 사상



# ER 모델을 관계 데이터 모델로 사상

- 완성된 ER 모델은 실제 데이터베이스로 구축하기 위해 논리적 모델링 단계를 거치는데, 이 단계에서 사상(mapping)이 이루어짐

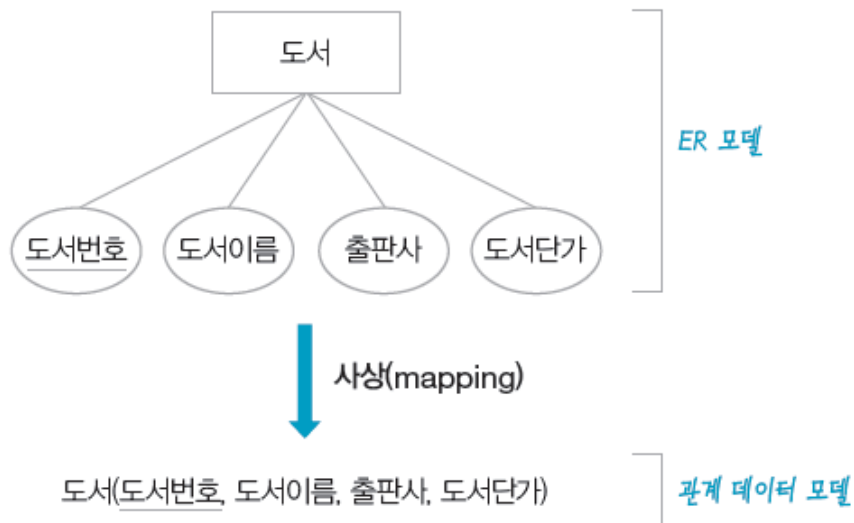


그림 6-33 ER 모델을 관계 데이터 모델로 사상

단계	사상할 대상	구분
1단계	개체 타입	강한 개체 타입
2단계		약한 개체 타입
3단계	관계 타입	이진 1:1 관계 타입
4단계		이진 1:N 관계 타입
5단계		이진 N:M 관계 타입
6단계	속성	N진 관계 타입
7단계		다중값 속성

표 6-12 ER 모델과 관계 데이터 모델의 사상 알고리즘

# 1. 개체 타입의 사상

- [1단계] 강한(정규) 개체 타입 정규 개체 타입 E의 경우 대응하는 릴레이션 R을 생성함
- [2단계] 약한 개체 타입 : 약한 개체 타입에서 생성된 릴레이션은 자신의 키와 함께 강한 개체 타입의 키를 외래키로 사상하여 자신의 기본키를 구성함

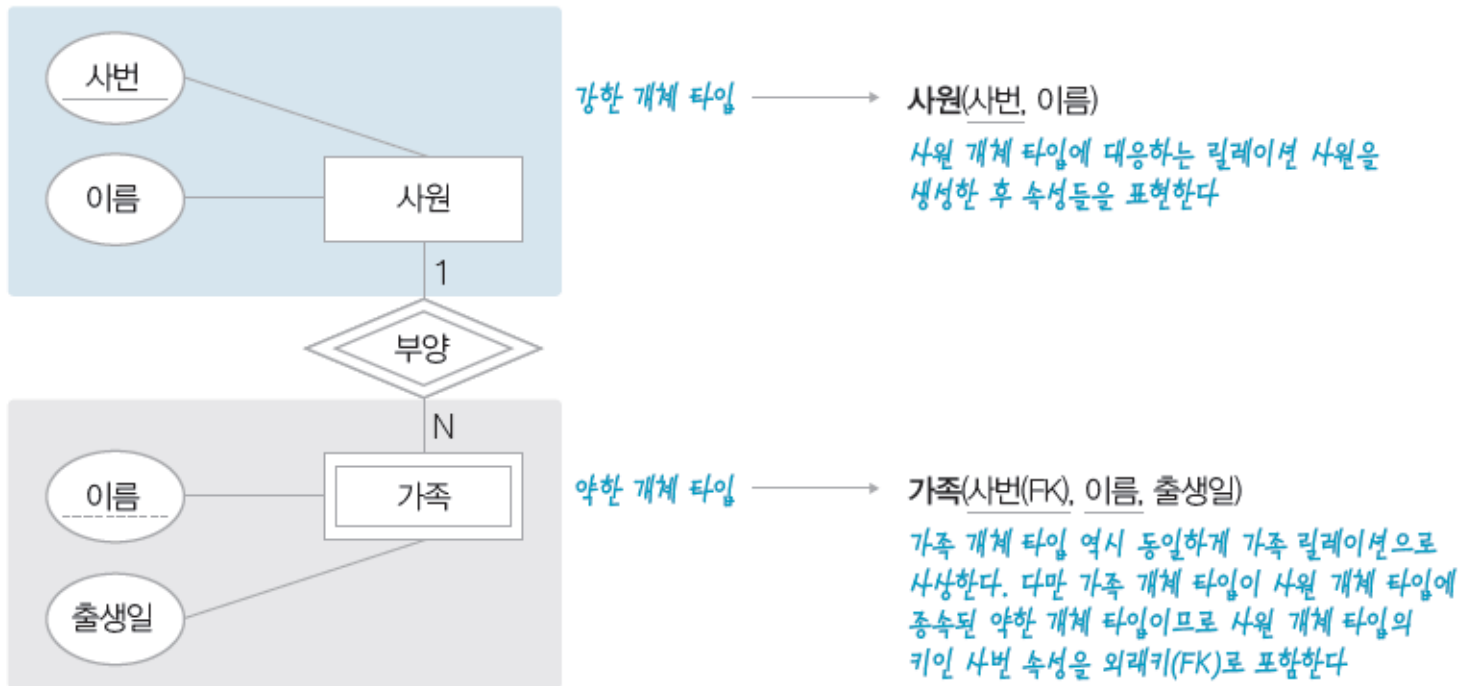


그림 6-34 개체 타입의 사상

## 2. 관계 타입의 사상

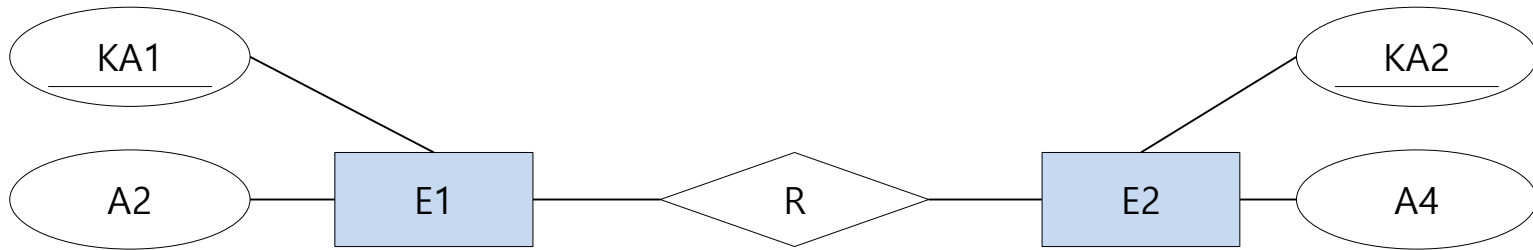


그림 6-35 이진 관계 타입

**[방법1]** 오른쪽 개체 타입 E2를 기준으로 관계 R을 표현한다.

$E1(\underline{KA1}, A2)$

$E2(\underline{KA2}, A4, \underline{KA1})$

**[방법2]** 왼쪽 개체 타입 E1을 기준으로 관계 R을 표현한다.

$E1(\underline{KA1}, A2, \underline{KA2})$

$E2(\underline{KA2}, A4)$

**[방법3]** 단일 릴레이션 ER로 모두 통합하여 관계 R을 표현한다.

$ER(\underline{KA1}, A2, \underline{KA2}, A4)$

**[방법4]** 개체 타입 E1, E2와 관계 타입 R을 모두 독립된 릴레이션으로 표현한다.

$E1(\underline{KA1}, A2)$

$R(\underline{KA1}, \underline{KA2})$

$E2(\underline{KA2}, A4)$



## 2. 관계 타입의 사상

### ■ [3단계] 이진 1:1 관계 타입

이진 1:1 관계 타입의 경우 [방법1]~[방법4]까지 모든 유형으로 사상 가능, 개체가 가진 정보 유형에 따라 판단

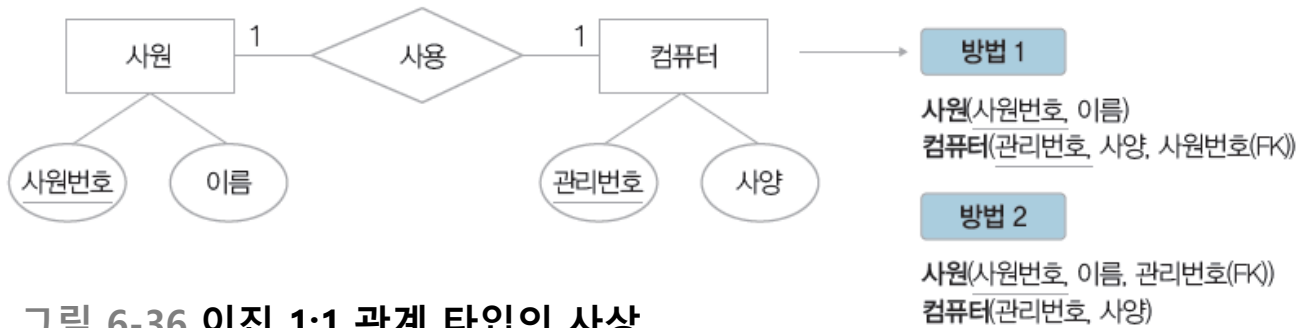


그림 6-36 이진 1:1 관계 타입의 사상

### ■ [4단계] 이진 1:N 관계 타입

이진 1:N 관계 타입의 경우 N의 위치에 따라 [방법1] 또는 [방법2]의 유형으로 사상됨.

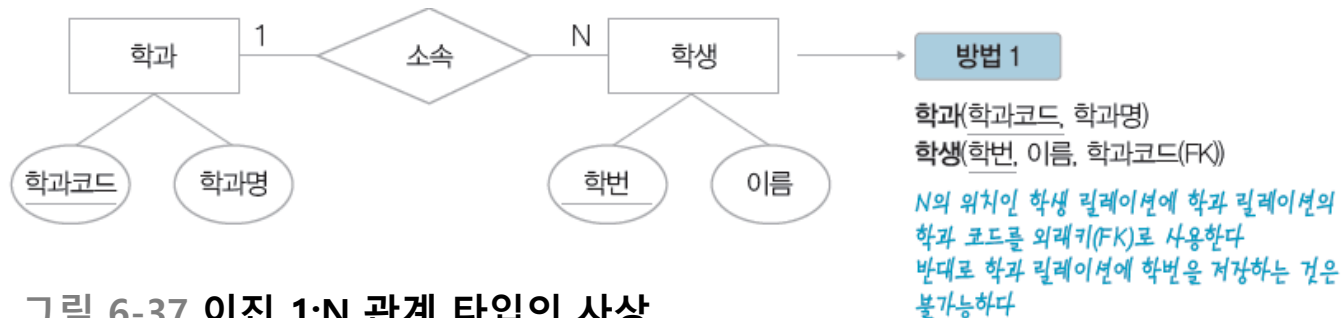


그림 6-37 이진 1:N 관계 타입의 사상

## 2. 관계 타입의 사상

### ■ [5단계] 이진 M:N 관계 타입

이진 M:N 관계 타입은 [방법4]의 유형으로 사상됨

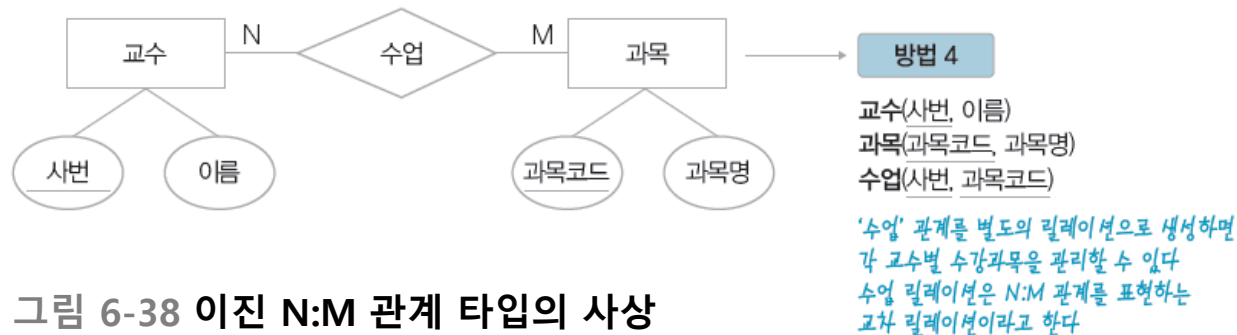


그림 6-38 이진 N:M 관계 타입의 사상

### ■ [6단계] N진 관계 타입

ER 모델의 차수가 3 이상인 다진 관계 타입의 경우 [방법4]의 유형으로 사상됨

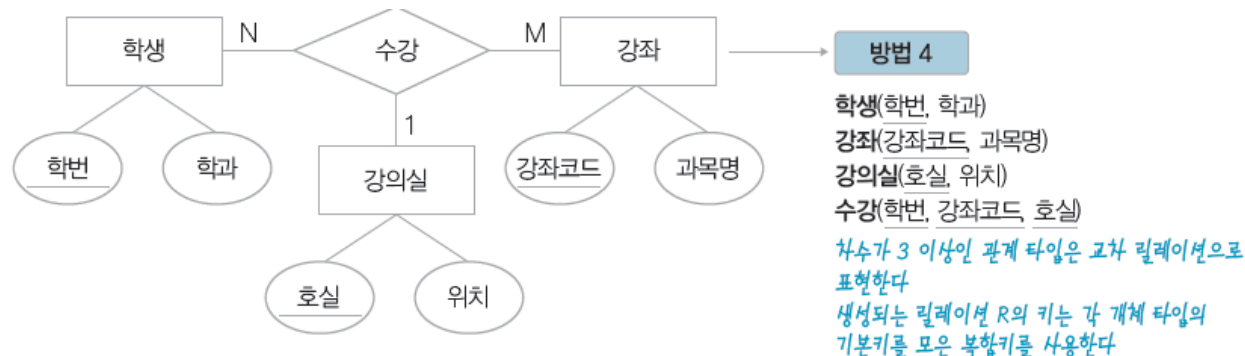


그림 6-39 이진 N진 관계 타입의 사상

### 3. 다중값 속성의 사상



그림 6-40 다중값 속성의 개수에 따른 사상 방법

#### ■ [7단계] 다중값 속성

속성의 개수를 알 수 없는 경우 [방법1]을, 속성의 개수가 제한적으로 정해지는 경우 [방법2]를 사용함

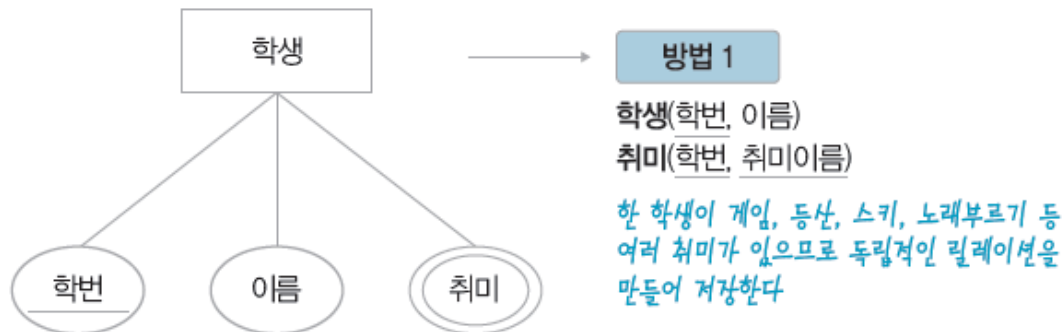


그림 6-41 다중값 속성의 사상

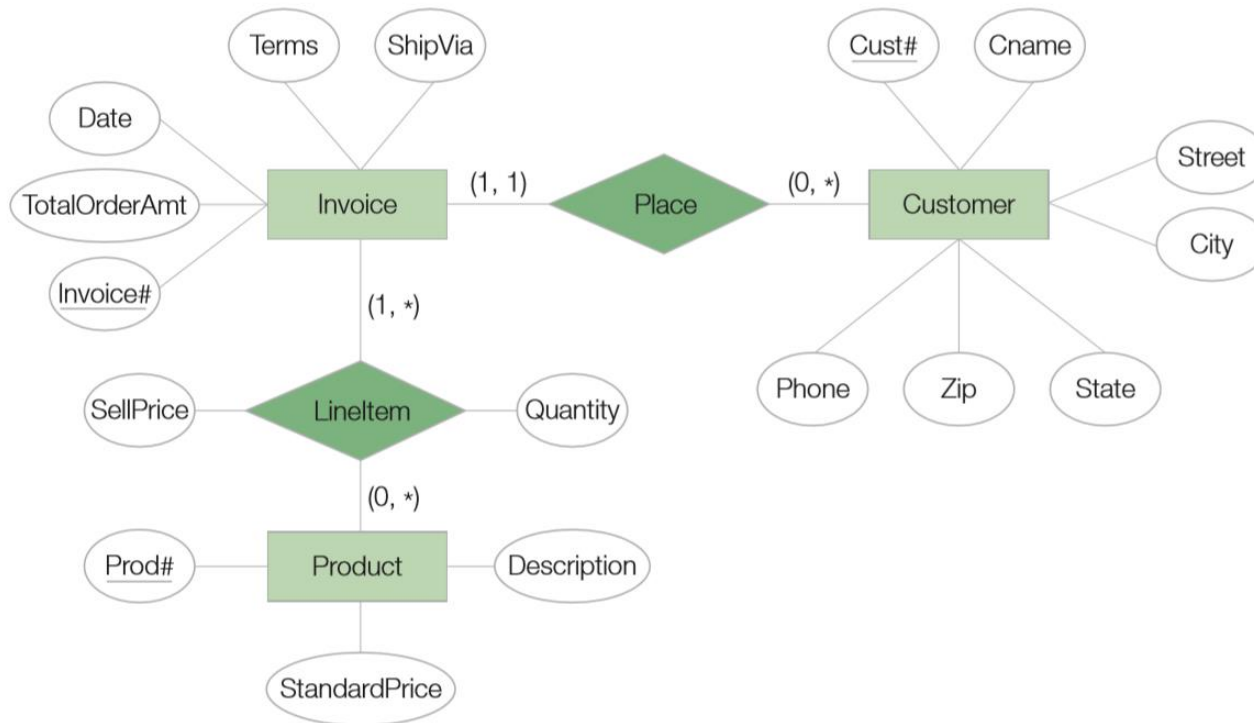
**06** 다음 내용을 모두 포함하는 데이터베이스를 설계하시오. 필요한 경우 몇 가지 가정을 넣을 수 있다.

- ① 회사는 네 개의 부서를 운영한다. 부서는 (부서번호, 부서이름)을 저장한다.
- ② 부서는 1명 이상의 직원(직원번호, 직원이름, 직책)을 두고 있다. 각 직원은 하나의 부서에 소속된다.
- ③ 직원은 부양가족(이름, 나이)이 있을 수 있다.
- ④ 각 직원은 근무했던 부서에 대한 근무기록(기간, 직책)이 있다.

- (1) ER 다이어그램을 그리시오.
- (2) ER 다이어그램을 IE 표기법으로 변환하여 그리시오.
- (3) ER 다이어그램을 테이블로 변환하시오.

# 연습문제

**08** 다음은 고객과 주문에 관한 ER 다이어그램이다. 개체는 고객(Customer), 제품(Product), 주문(Invoice)으로 구성된다. Place 관계는 '주문한다'를, LineItem은 '주문 항목'을 의미한다. 그림에 해당하는 테이블을 작성하시오(변환된 테이블의 기본키는 밑줄 실선, 외래키는 밑줄 점선으로 표시한다. 기본키인 동시에 외래키일 경우에는 밑줄 실선으로 표시한다. 테이블 변환을 위하여 필요한 사항 중 설명되지 않은 것은 임의로 정하여 설계한다).



### 04. ERwin 실습

1. ERwin 기본 화면 및 툴 둘러보기
2. ERwin 실습을 위한 기본 환경 설정하기
3. 마당서점 설계 실습
4. DBMS에 접속하여 테이블 생성하기



# 여기서 잠깐 ERwin Data Modeler 시험판 받기

- ERwin Data Modeler의 Academic License(교육용 시험판)이나 Trial 버전은 ERWIN사의 웹사이트(<https://erwin.com>)에서 받을 수 있다.
- Academic License는 ERWIN사의 웹사이트에 접속한 후 [Services]- [erwin Academic Program]- [erwin DM ACademic Edition]- [Apply Now]을 클릭하고 순서에 따라 본인 메일, 수업 과목명, 교수 이름 등을 입력하면 바로 다운받을 수 있다.
- 신청 후 수신받은 메일 을 따라 라이선스 신청 메일을 보내면 1~3일 후에 1년 동안 유효한 License Key(18자리 숫자)를 메일로 보 내준다. 이후 ERwin을 실행하고 License 키를 입력하면 된다.
- Trial 버전은 ERWIN사의 웹사이트 메인 페이지의 Try for Free을 클릭한 후 절차에 따라 신청하면 다운로드 받을 수 있다. 신청 시 Acadmic License는 학교 메일을, Trial 버전은 기업, 단체 등의 이메일을 사용해야만 한다(naver, daum, gmail 등은 신청할 수 없다).

# ERwin 실습

- ERwin : 데이터 모델링을 하기 위한 프로그램. IE 표기법을 지원함

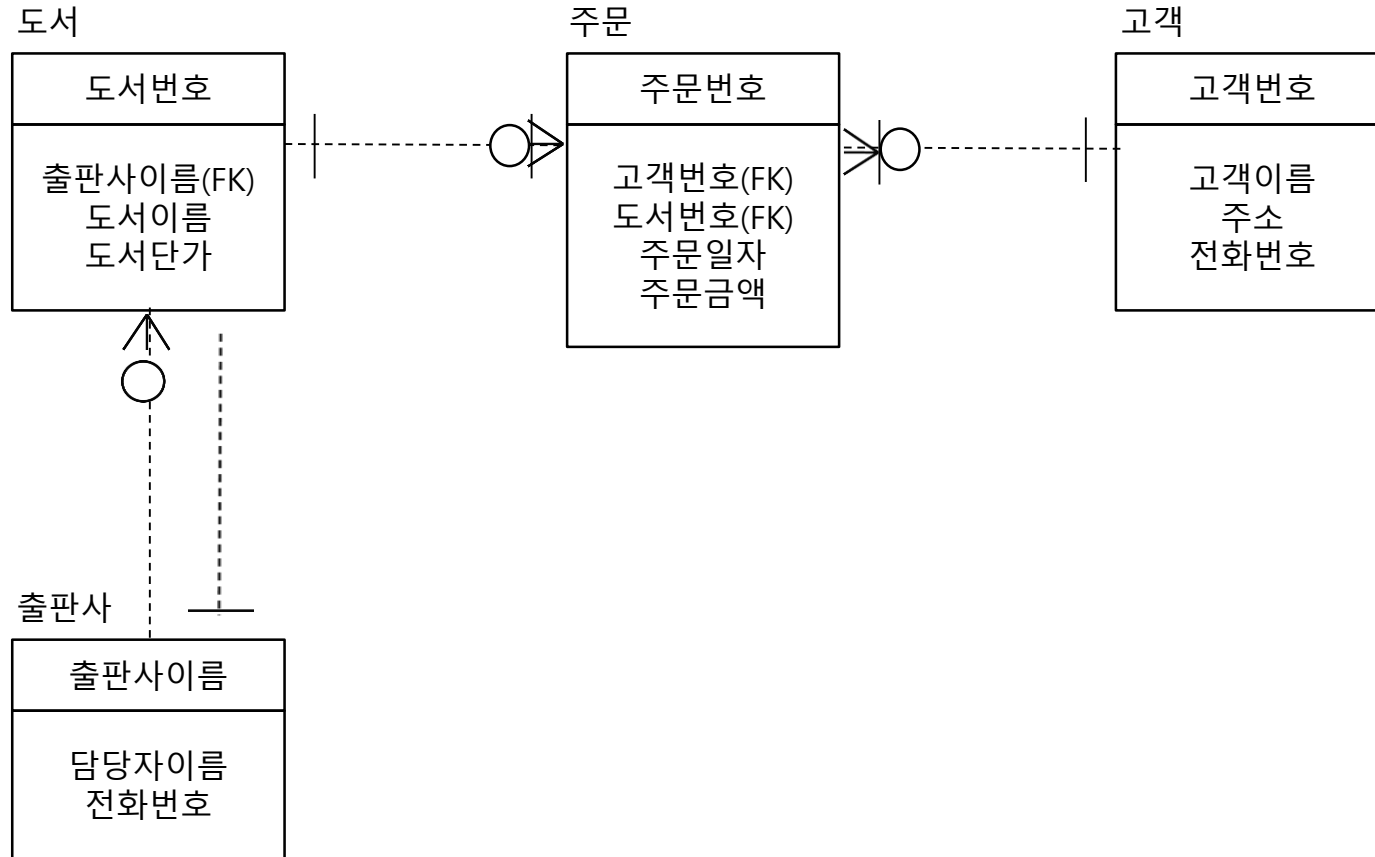
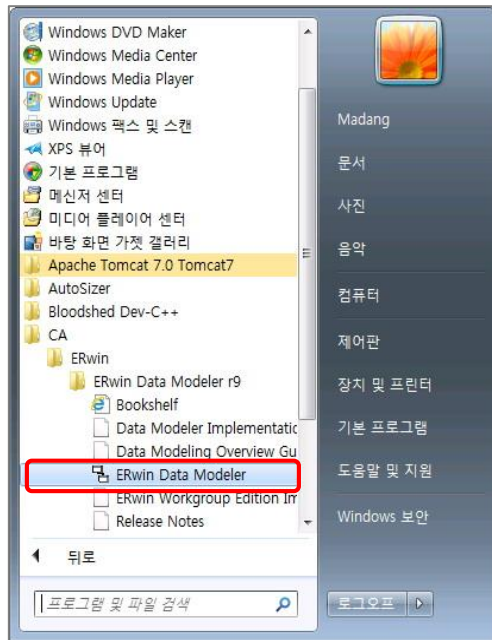


그림 6-42 마당서점의 ER 다이어그램

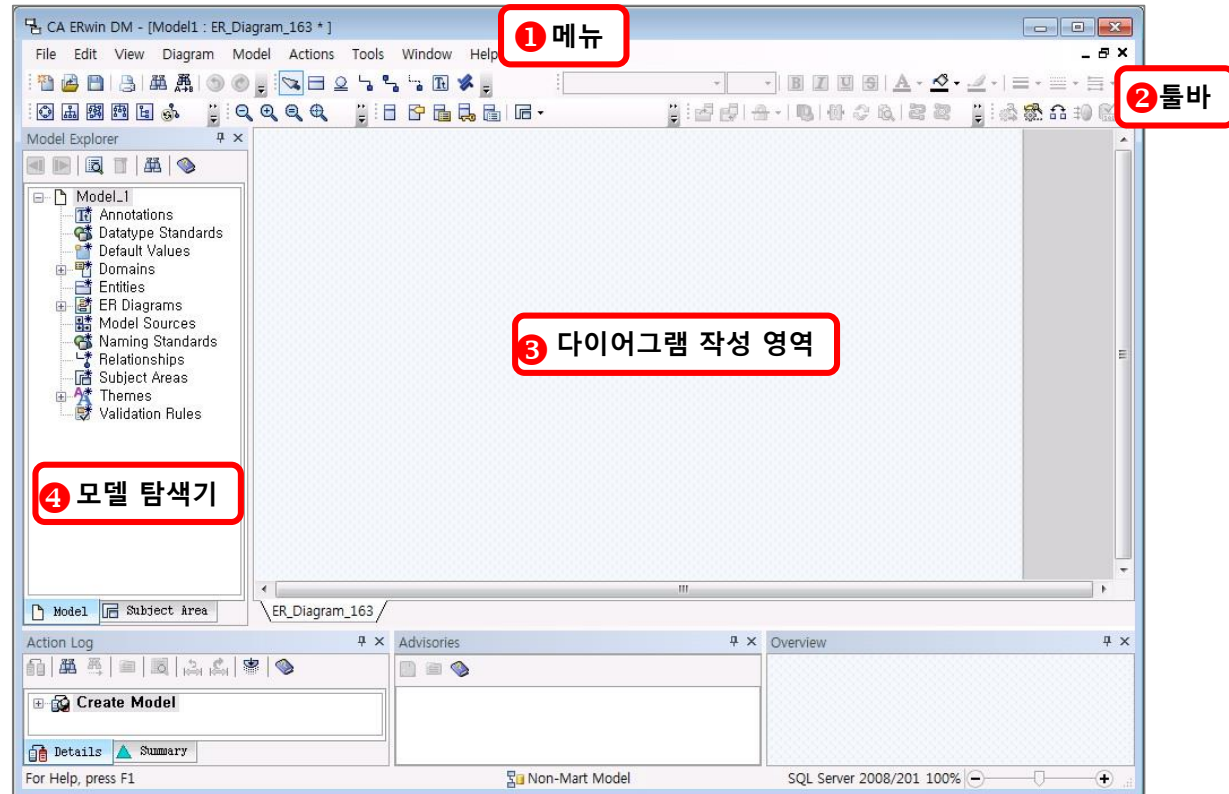


# 1. ERwin 기본 화면 및 툴 둘러보기

## ① ERwin Data Modeler 실행

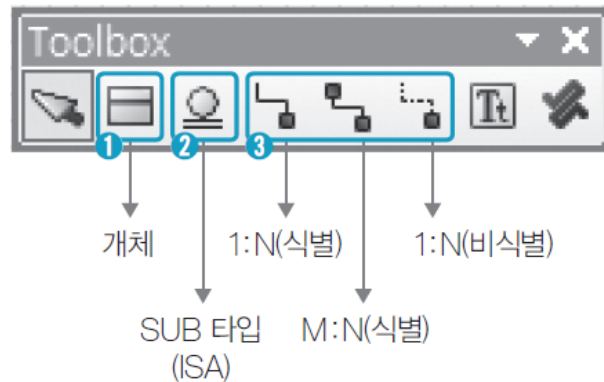


## ② ERwin의 기본 화면



# 1. ERwin 기본 화면 및 툴 둘러보기

## ■ 툴바



- ① **개체** : 개체 타입의 이름, 식별자, 속성을 표현함.
- ② **SUB 타입** : ISA 모델의 슈퍼클래스와 서브클래스처럼 부모, 자식 관계에서 자식 개체가 서로 배타적인 관계를 가지는 여러 서브 개체 타입을 표현함.
- ③ **1:N(식별), N:M(식별), 1:N(비식별)** : 1, N, M은 두 개체 간의 관계에서 관계 대응수를 말함. 식별 관계는 두 개체가 부모(1), 자식(N) 관계일 때 부모의 기본키가 자식의 기본키가 되거나 기본키의 구성원으로 사용되는 관계로 실선으로 나타냄. 비식별 관계는 부모의 기본키가 자식의 기본키가 아닌 속성의 일부로 전이되는 관계로 점선으로 나타냄. 관계의 필수(1)와 선택(0)은 관계선의 옵션을 통해 선택할 수 있음.

## 2. ERwin 실습을 위한 기본 환경 설정하기

- 목적과 대상에 맞는 모델, DBMS, 표기법을 선택해야 함.

표 6-13 Erwin 실습을 위한 기본 환경 설정

모델 타입	Logical/Physical	(새로만들때) File -> New -> Type
DBMS	MySQL 5.x	(새로만들때) File -> New -> Type
표기법	IE 표기법	Model -> Properties -> Notation 에서 변경

- 기본 환경 설정 순서

1. 모델 타입, DBMS 선택하기
2. IE 표기법으로 변경하기
3. 툴바에 메뉴 추가하기(선택사항)

### 3. 마당서점 설계 실습

#### ❖ 마당서점의 논리적 모델링

① 마당서점의 요구사항 분석 후 개체 만들기

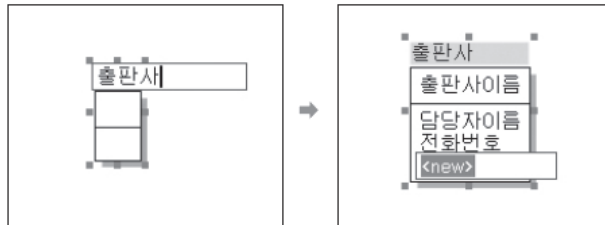


그림 6-52 출판사 개체 생성

② 개체 간 관계 표현하기

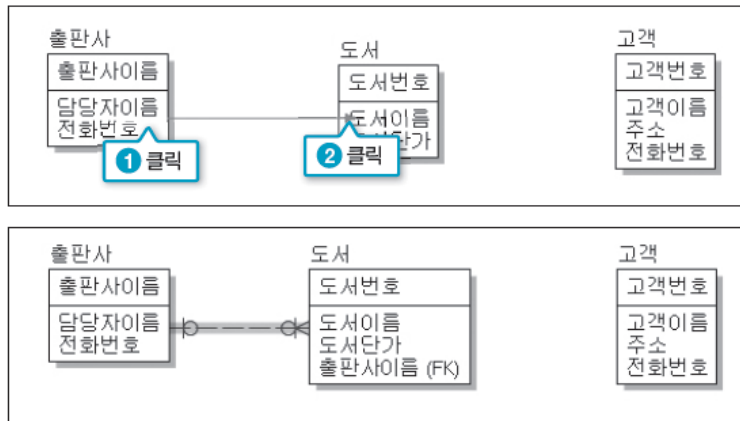


그림 6-53  
출판사, 도서 개체의 관계 설정(1:N 비식별)

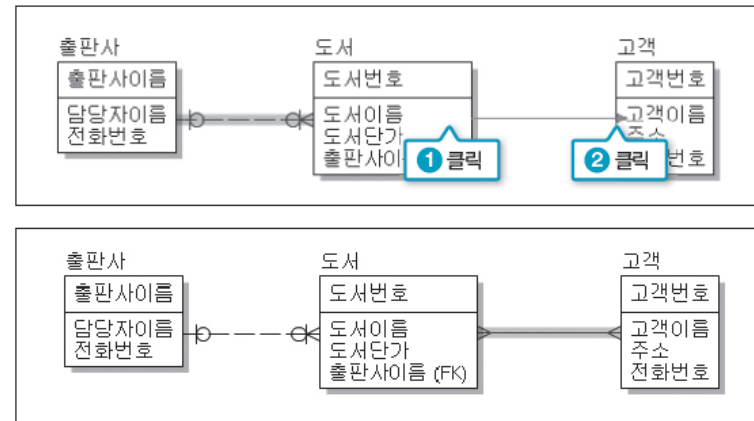


그림 6-54  
도서, 고객 개체의 관계 설정(N:M 식별)

### 3. 마당서점 설계 실습

#### ③ N:M 관계 해소하기

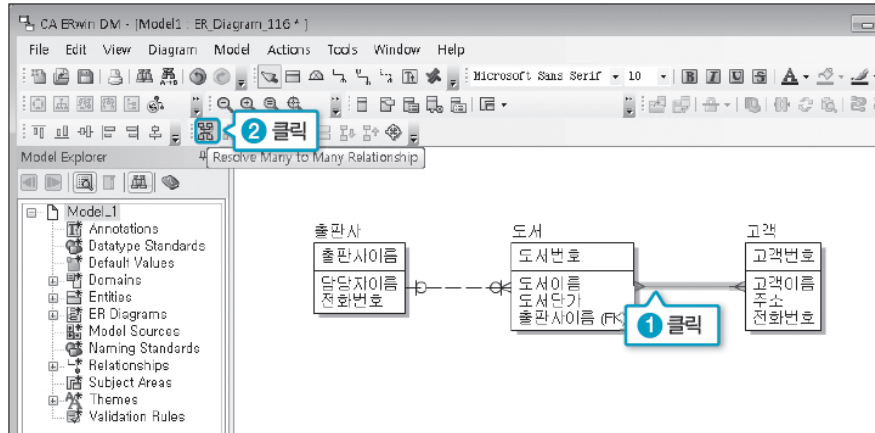


그림 6-55 N:M 관계 해소

#### ④ 식별 관계 및 관계 대응 수 변경하기



그림 6-57 개체 간 식별 관계 변경

Relationship 'OrdCustomer' Editor

Name	Parent	Child	Logical Only
BookPublisher	출판사	도서	<input type="checkbox"/>
OrdBook	도서	주문	<input type="checkbox"/>
OrdCustomer	고객	주문	<input type="checkbox"/>

General | Definition | Role Name | RI Actions | Style | UDP | Notes

'OrdCustomer' Type Properties

Type: Non-Identifying

Full Option: Nulls Not Allowed

'OrdCustomer' Relationship Properties

Parent-to-Child Phrase:

Child-to-Parent Phrase:

'OrdCustomer' Cardinality Properties

Cardinality: Zero, One or More

Cardinality Value:

3 설정 후 클릭

Close Cancel

### 3. 마당서점 설계 실습

#### ❖ 도메인 정의하기

- 도메인이란 속성이 가질 수 있는 값을 정의하는 것. ER 다이어그램이 완성 후 도메인을 정의함

표 6-17 마당서점의 도메인별 데이터 타입 정의표

구분(기본 도메인)	도메인	데이터 타입
숫자(Number)	번호	INTEGER
	금액	INTEGER
	단가	INTEGER
문자(String)	이름	VARCHAR(40)
	주소	VARCHAR(40)
	전화번호	VARCHAR(30)
날짜시간(Datetime)	일자	DATE

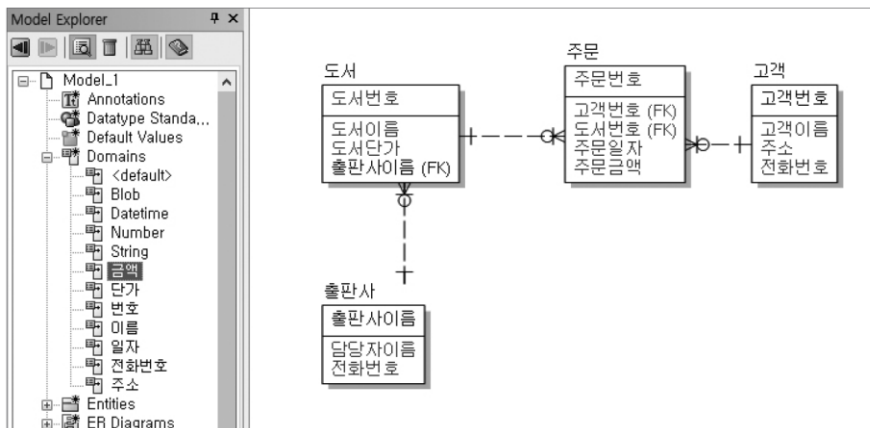


그림 6-60 도메인 정의표에 따라 생성한 도메인

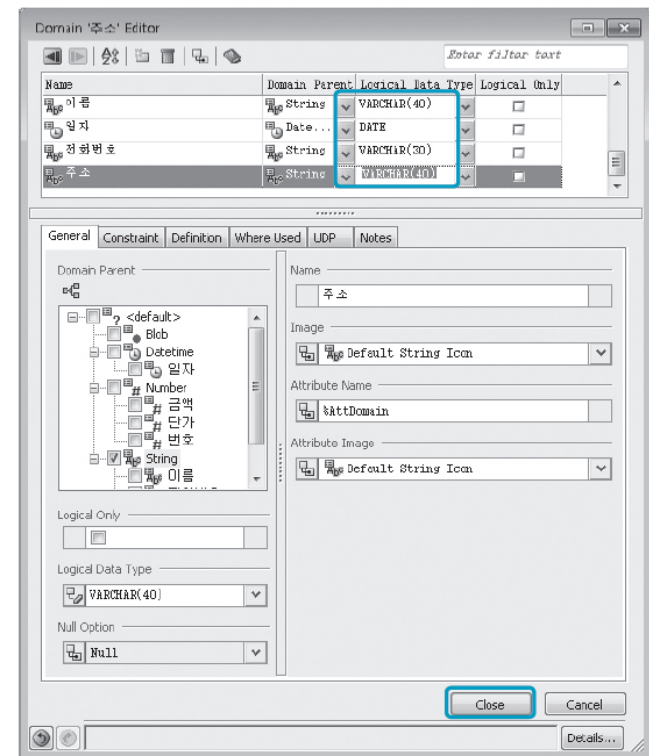


그림 6-61 도메인별 데이터 타입 설정

### 3. 마당서점 설계 실습

#### ❖ 마당서점의 물리적 모델링

- ① ER 다이어그램 불러오기 : [File] -[Open]
- ② Physical 타입으로 변경하기 : [View] - [Physical Model]

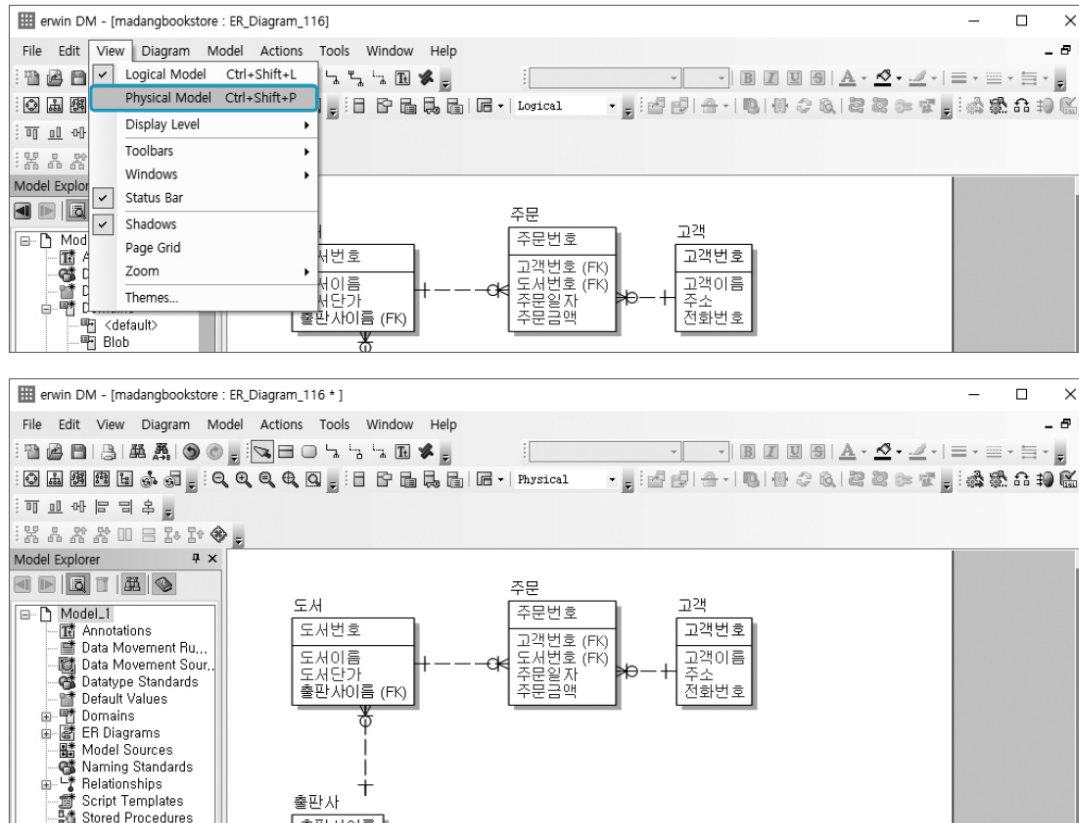


그림 6-63 모델 타입의 변경

### 3. 마당서점 설계 실습

#### ❖ 마당서점의 물리적 모델링

##### ③ 물리적 모델링

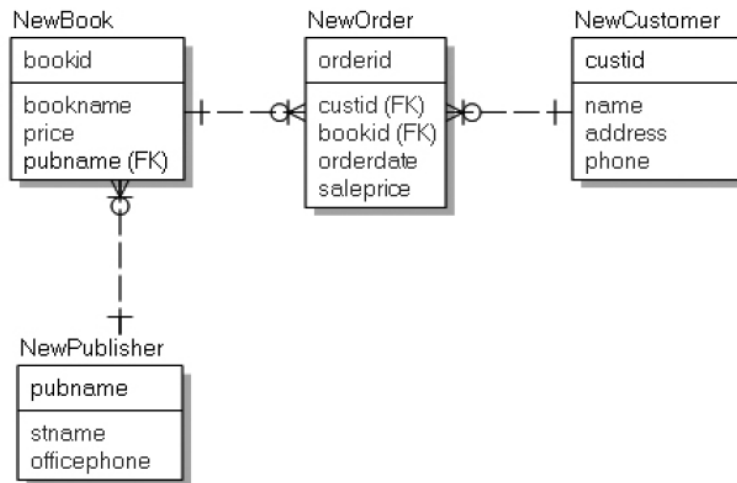


그림 6-64 마당서점의 테이블



### 3. 마당서점 설계 실습

#### ❖ 마당서점의 물리적 모델링

##### ④ 컬럼의 속성 확인하기

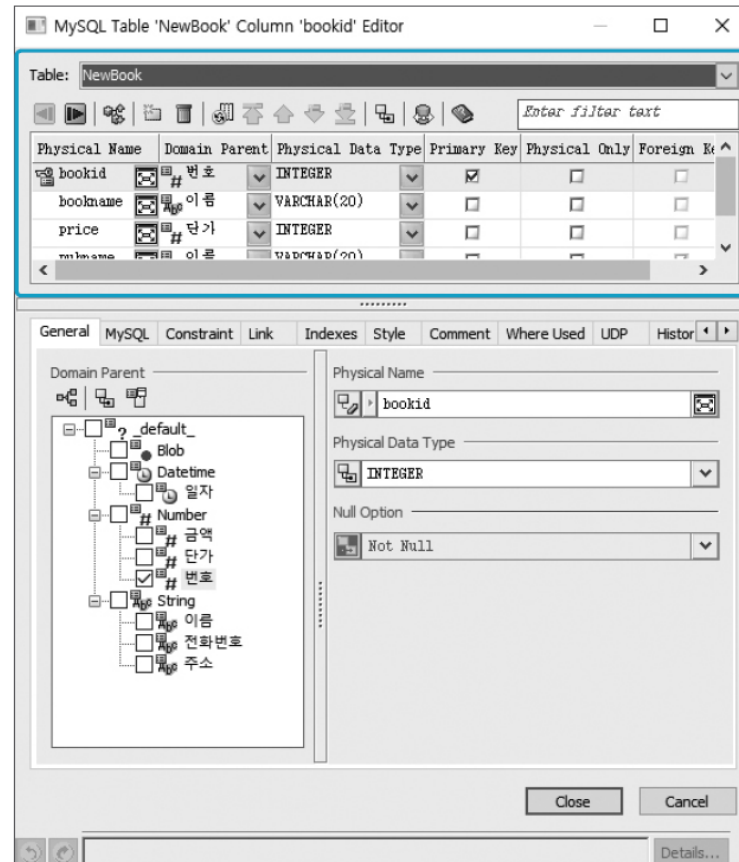
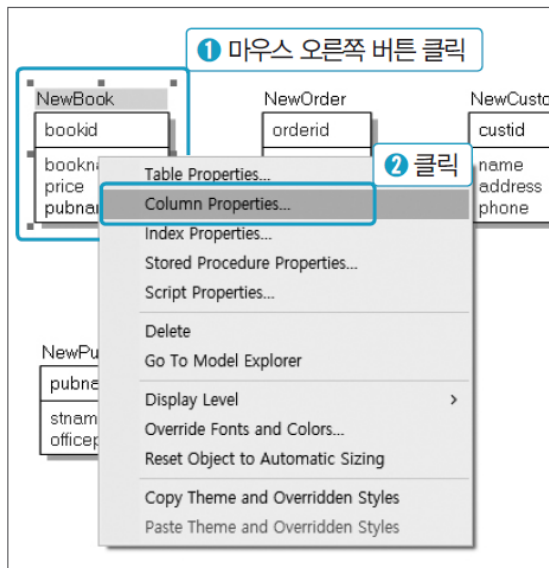


그림 6-65 컬럼 속성 확인

# 여기서 잠깐 Erwin 모델링 팁

- 최소 관계를 0->1로 변경하려면,  
1:N의 관계는 관계를 오른쪽 마우스로 클릭한 후
  - > Properties -> Cardinality Properties
  - > One or More로 수정(1쪽), Nulls Not Allowed(N쪽)
- 관계 이름을 표시하려면  
모델 그림의 바탕화면에서 오른쪽 마우스 클릭한 후
  - > Properties -> Relationship -> Display Logical Relationship Name(check)
  - > Properties -> Relationship -> Physical Logical Relationship Name(check)
- 1:1 순환 관계의 표현  
자기 자신으로 바탕화면을 한번 클릭해가면서 자기자신으로 클릭  
1:N 카디널리티 표현을 위한 외래키는 별도로 생성해야 함  
(관계를 오른쪽마우스선택 -> properties -> Role Name -> (이름 입력))

## 4. DBMS에 접속하여 테이블 생성하기

### ❶ DBMS에 접속하기

[Actions]-[Database Connection] 메뉴 선택 후 [Oracle Connection] 창에서 다음과 같이 설정

Database : MySQL 5.x  
Authentication : Database Authentication  
User Name : madnag  
Password : madang  
ODBC Data Source : madang

(ODBC 설정 시 사용한 이름 = madang)

## 4. DBMS에 접속하여 테이블 생성하기

### ② 테이블 생성하기

[Actions]–[Forward Engineer]–[Schema] 메뉴 선택

-> 각 대상별로 오라클에 적용할 내용 설정함

-> 테이블과 인덱스를 제외한 모든 부분의 체크 해제

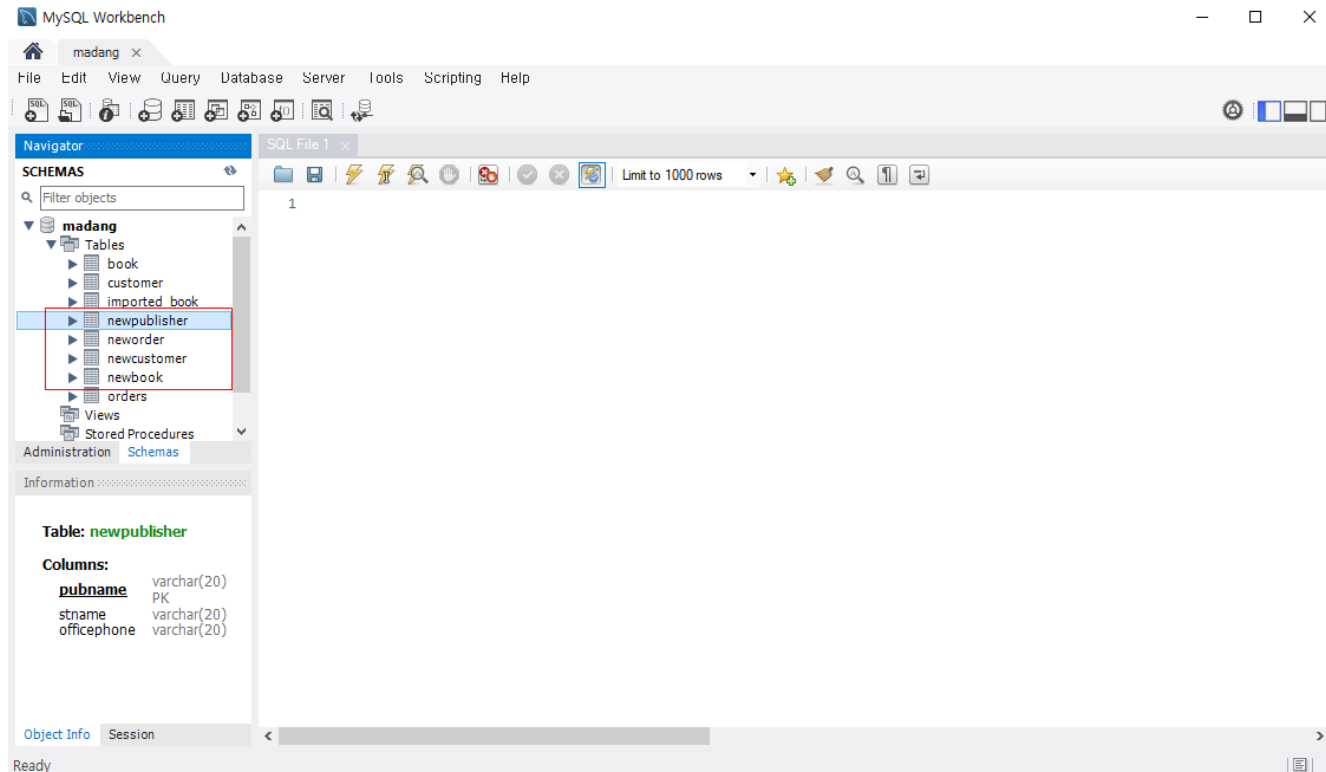


그림 6-70 madang 데이터베이스에 추가된 테이블

## 05. 모델링 연습(마당대학 데이터베이스)

1. 마당대학의 요구사항
2. 마당대학의 ER 다이어그램



# 1. 마당대학의 요구사항

- ① 교수(Professor)는 아이디(ssn), 이름(name), 나이(age), 직위(rank), 연구 분야(speciality)를 가진다.
- ② 학과(Department)에는 학과번호(dno), 학과이름(dname), 학과사무실(office)이 있다.
- ③ 대학원생(Graduate)은 아이디(ssn), 이름(name), 나이(age), 학위과정(deg\_prog, 석사/박사)을 가진다.
- ④ 과제(Project)는 과제번호(pid), 지원기관(sponsor), 개시일(start\_date), 종료일(end\_date), 예산액(budget)이 있다.
- ⑤ 학과마다 그 학과를 운영(run)하는 교수(학과장이라고 한다)가 한 명씩 있다.
- ⑥ 한 교수가 여러 학과에서 근무(work-dept)할 수 있는데, 이때 각 학과별로 참여백분율(pct\_time)이 기록된다.
- ⑦ 대학원생에게는 학위 과정을 밟을 전공학과(major)가 하나씩 있다.
- ⑧ 대학원생에게는 어떤 과목을 들으면 좋을지 조언(advisor)해주는 선임 대학원생(학생조언자라고 한다)이 있다.
- ⑨ 과제는 한 교수(연구책임자라고 한다)에 의해 관리(manage)된다.
- ⑩ 과제는 한 사람 이상의 교수(공동연구책임자라고 한다)에 의해 수행(work-in)된다.
- ⑪ 한 과제는 한 명 이상의 대학원생(연구조교라고 한다)에 의해 수행(work-prog)된다.

## 2. 마당대학의 ER 다이어그램

① 교수(Professor)는 아이디(ssn), 이름(name), 나이(age), 직위(rank), 연구 분야(speciality)를 가진다.

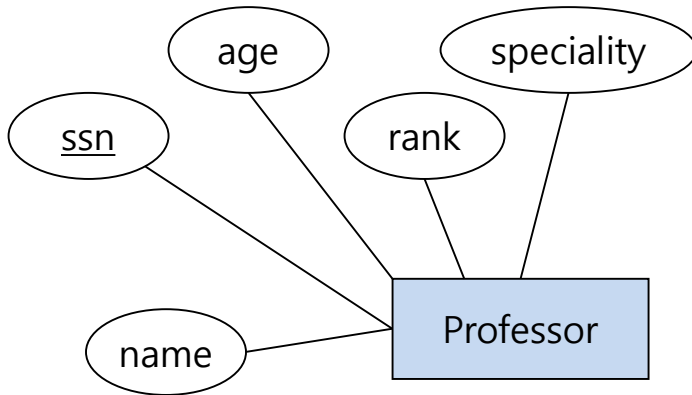


그림 6-71 교수(Professor) 개체

② 학과(Department)에는 학과번호(dno), 학과이름(dname), 학과사무실(office)이 있다.

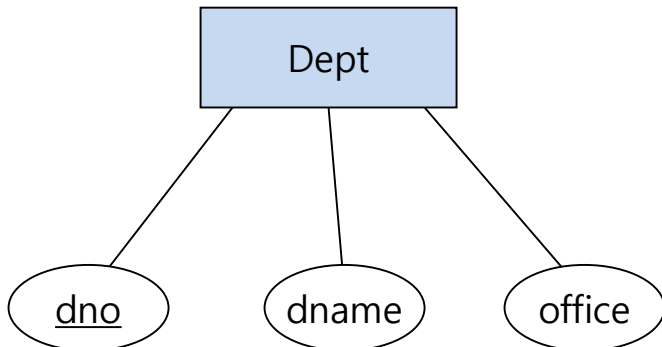


그림 6-72 학과(Dep) 개체

## 2. 마당대학의 ER 다이어그램

③ 대학원생(Graduate)은 아이디(ssn), 이름(name), 나이(age), 학위과정(deg\_prog, 석사/박사)을 가진다.

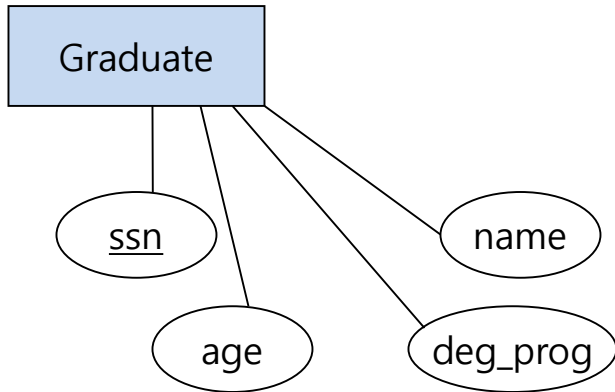


그림 6-73 대학원생(Graduate) 개체

④ 과제(Project)는 과제번호(pid), 지원기관(sponsor), 개시일(start\_date), 종료일(end\_date), 예산액(budget)이 있다.

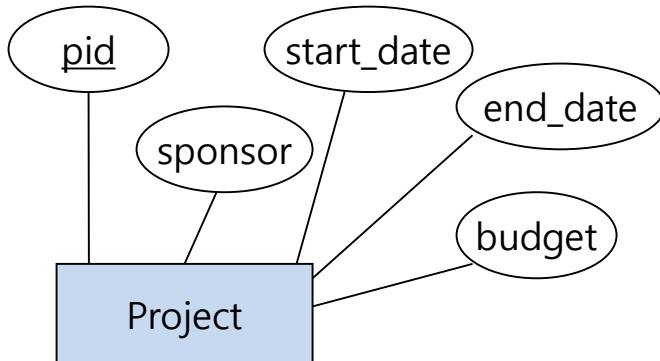


그림 6-74 과제(Project) 개체



## 2. 마당대학의 ER 다이어그램

- ⑤ 학과마다 그 학과를 운영(run)하는 교수(학과장이라고 한다)가 한 명씩 있다.
- ⑥ 한 교수가 여러 학과에서 근무(work-dept)할 수 있는데, 이때 각 학과별로 참여백분율(pct\_time)이 기록된다.

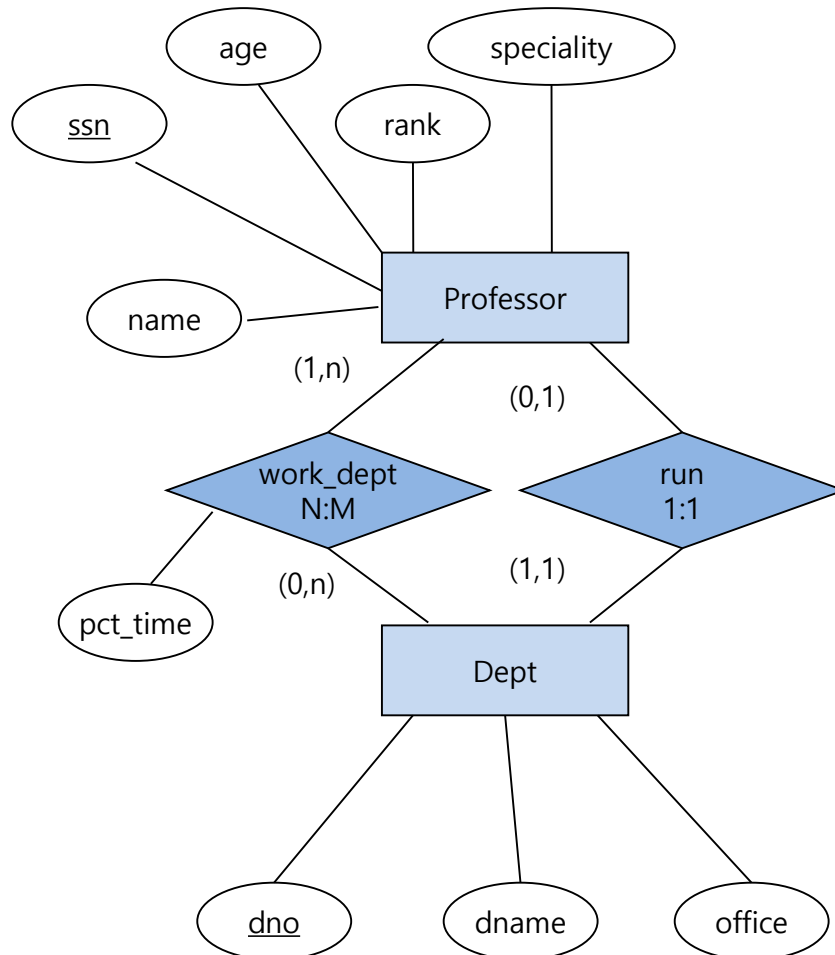


그림 6-75 운영(run) 관계와 근무(work-dept) 관계

## 2. 마당대학의 ER 다이어그램

- ⑦ 대학원생에게는 학위 과정을 밟을 전공학과(major)가 하나씩 있다.
- ⑧ 대학원생에게는 어떤 과목을 들으면 좋을지 조언(advisor)해주는 선임 대학원생(학생조언자라고 한다)이 있다.

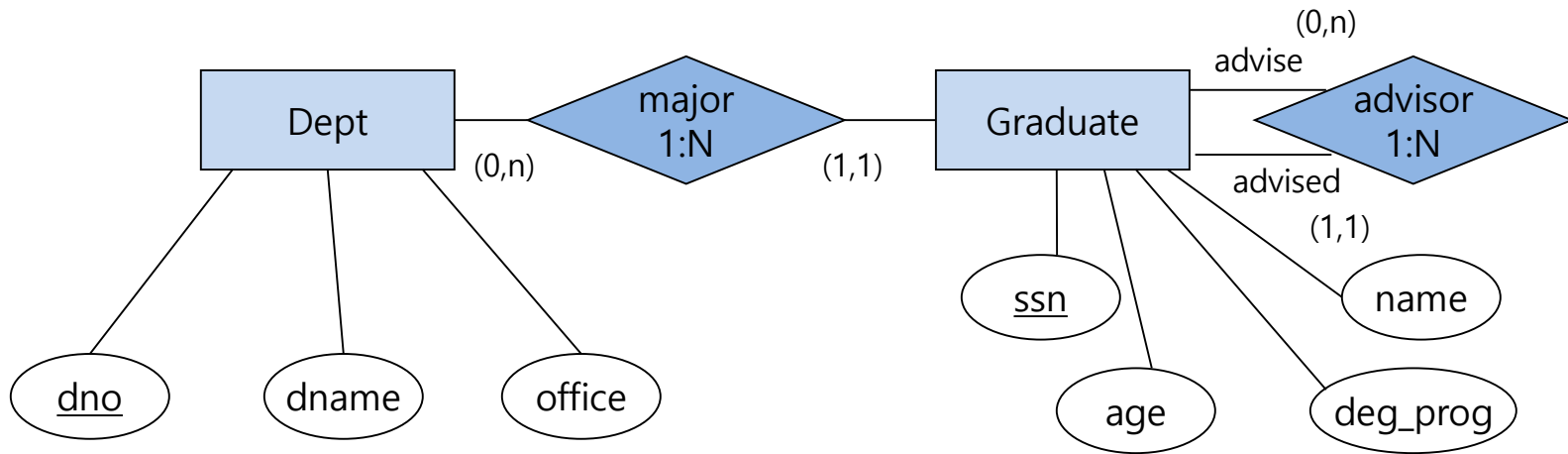


그림 6-76 전공학과(major) 관계와 조언(advisor)관계

## 2. 마당대학의 ER 다이어그램

- ⑨ 과제는 한 교수(연구책임자라고 한다)에 의해 관리(manage)된다.  
⑩ 과제는 한 사람 이상의 교수(공동연구책임자라고 한다)에 의해 수행(work-in)된다.

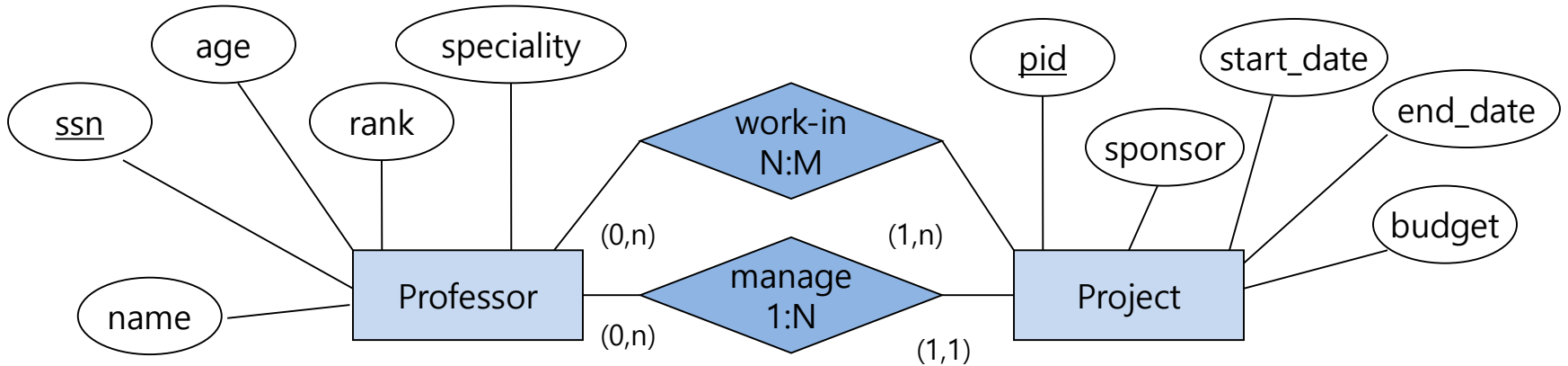


그림 6-77 관리(manage) 관계와 수행(work-in) 관계

## 2. 마당대학의 ER 다이어그램

⑪ 한 과제는 한 명 이상의 대학원생(연구조교라고 한다)에 의해 수행(work-prog)된다.

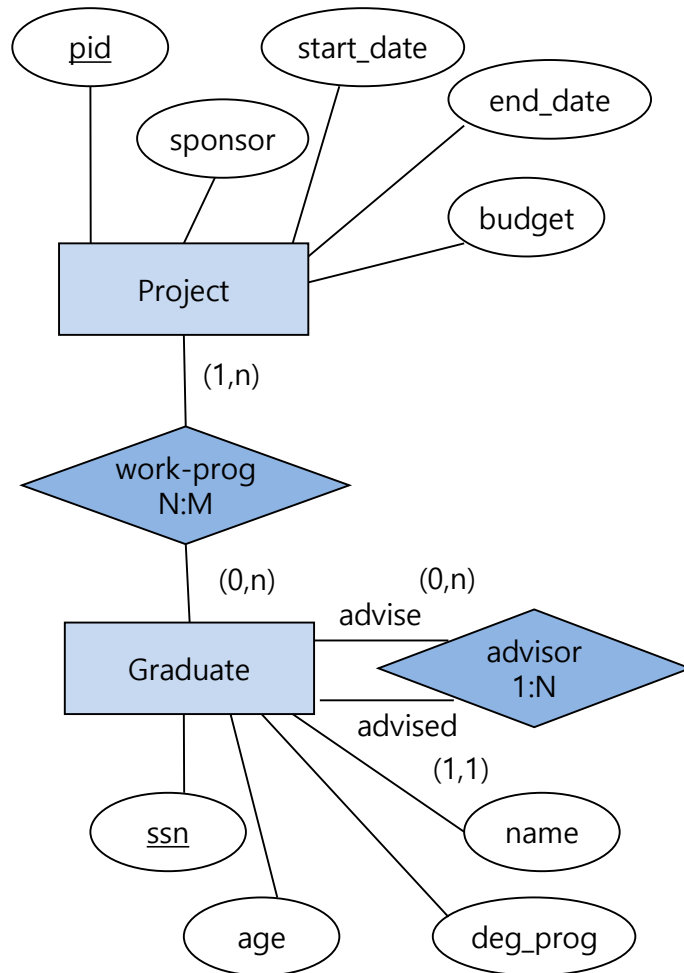


그림 6-78 수행(work-prog) 관계

## 2. 마당대학의 ER 다이어그램

### ❖ 요구사항에 맞게 생성된 최종 ER 다이어그램

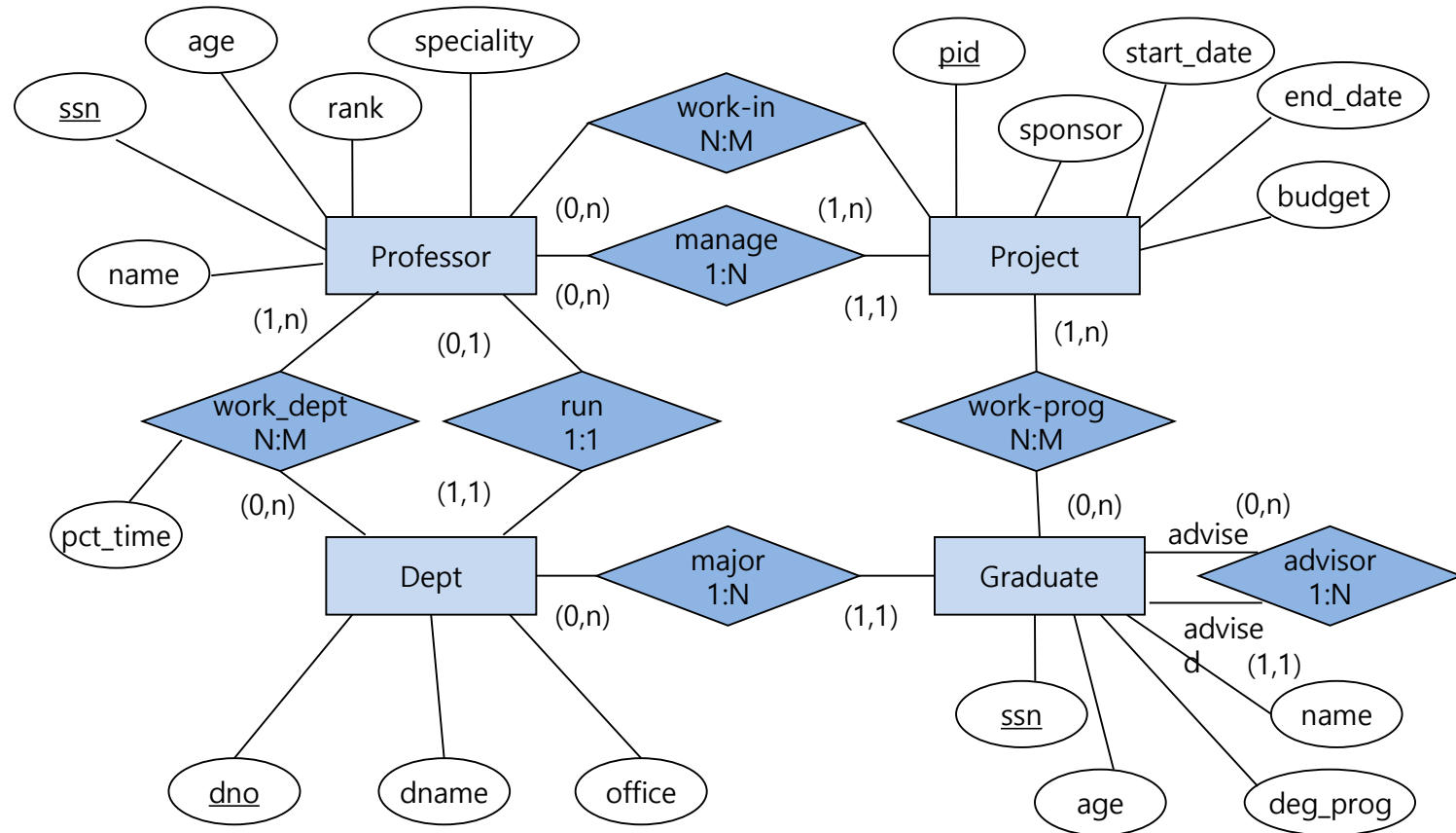


그림 6-79 마당대학의 ER 다이어그램

## 2. 마당대학의 ER 다이어그램

## ❖ 관계 데이터 모델로 변환한 마당대학의 ER 다이어그램

[개체]

Professor(ssn, name, age, rank, speciality)

Dept(dno, dname, office, runprofessorssn)

runprofessorssn : Professor(ssn)을 참조하는 외래키로, ⑤번 요구사항의 run(1:1) 관계를 표현하기 위해 추가되었다.

Graduate(ssn, name, age, deg\_prog, dno, graduatessn)

dno : Dept(dno)를 참조하는 외래키로, ⑦번 요구사항 major(1:N) 관계를 표현하기 위해 추가되었다.

graduatessn : Graduate(ssn)을 참조하는 외래키로 @버 요구사항 advisor(1:N) 관계를 표현하기 위해 추가되었다

```
Project(pid, sponsor, start_date,
```

managessn : Professor(ssn)을 참

[관계]

M:N관계일 경우 교차 테이블을 생성

```
work-dept(professorssn, dno,
```

⑥번 요구사항의 work-dept(M:N) :

래키고, dno는 Dept(dno)를 참조하는

```
work-in(professorssn, pid) /*
```

⑩번 요구사항의 work-in(M:N) 관계

고, pid는 Project(pid)를 참조하는 외래키다.

```
work-prog(graduate_ssn, pid) /* 대학원생의 과제 수행 */
```

⑪번 요구사항의 work-prog(M:N) 관계를 표현하기 위한 교차 테이블이다. graduatessn은 Graduate(ssn)을 참조하는 외래 키고, pid는 Project(pid)를 참조하는 외래키다.

kim	MySQL	SQL Server	ORACLE
lee	MySQL	SQL Server	
author			topic
id	name	profile	created
1	kim	developer	2011
2	lee	designer	2012
3	park	planner	2013
3	park	planner	2013
author			write
id	name	profile	created
1	kim	developer	2011
2	lee	designer	2012
3	park	planner	2013
			topic
id	title	description	created
1	MySQL	...	2011
2	ORACLE	...	2012
3	SQL Server	...	2013
3	SQL Server	...	2013

그림 6-80 마당대학 데이터베이스 스키마

## 2. 마당대학의 ER 다이어그램

- 관계 데이터 모델이 완성되면 실제 DBMS에 테이블을 생성하기 위한 물리적 모델링 과정을 거침

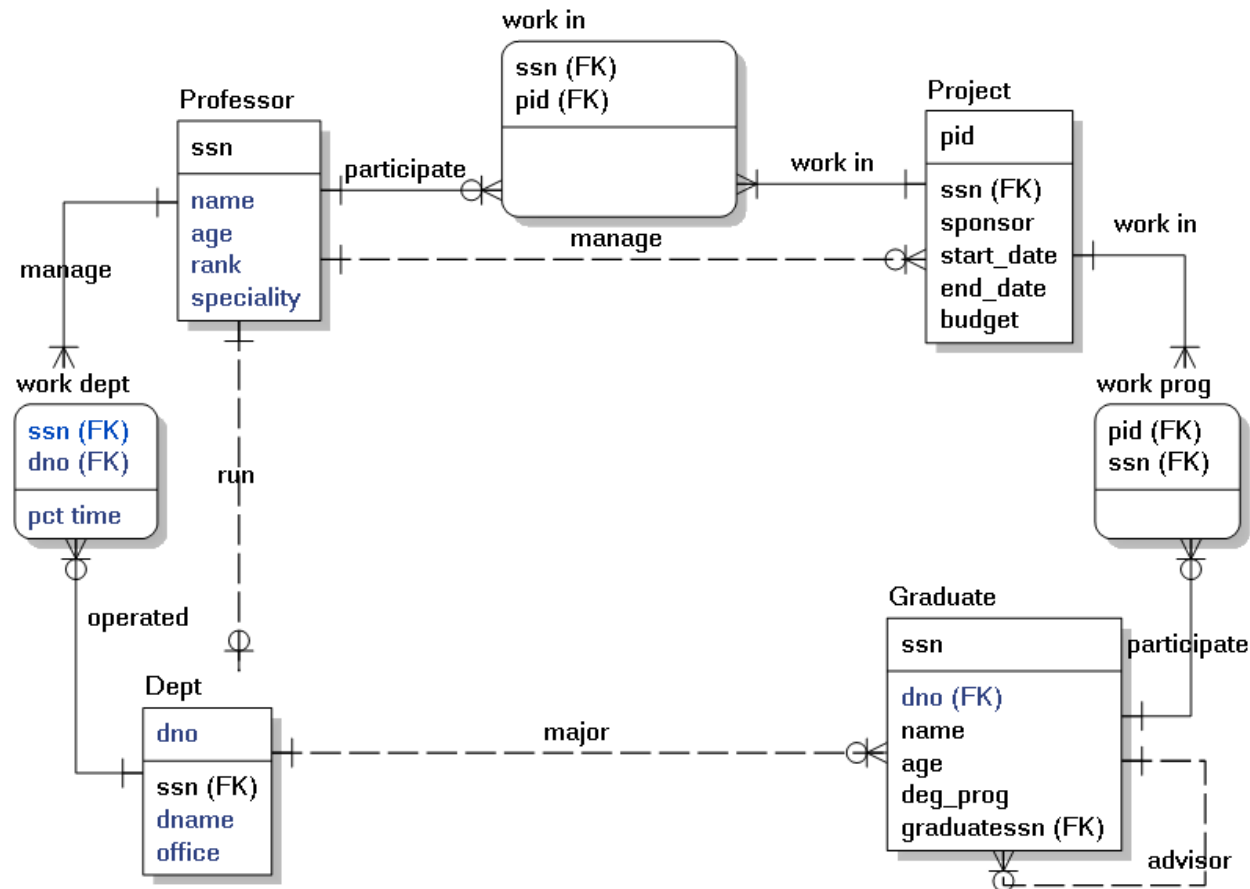
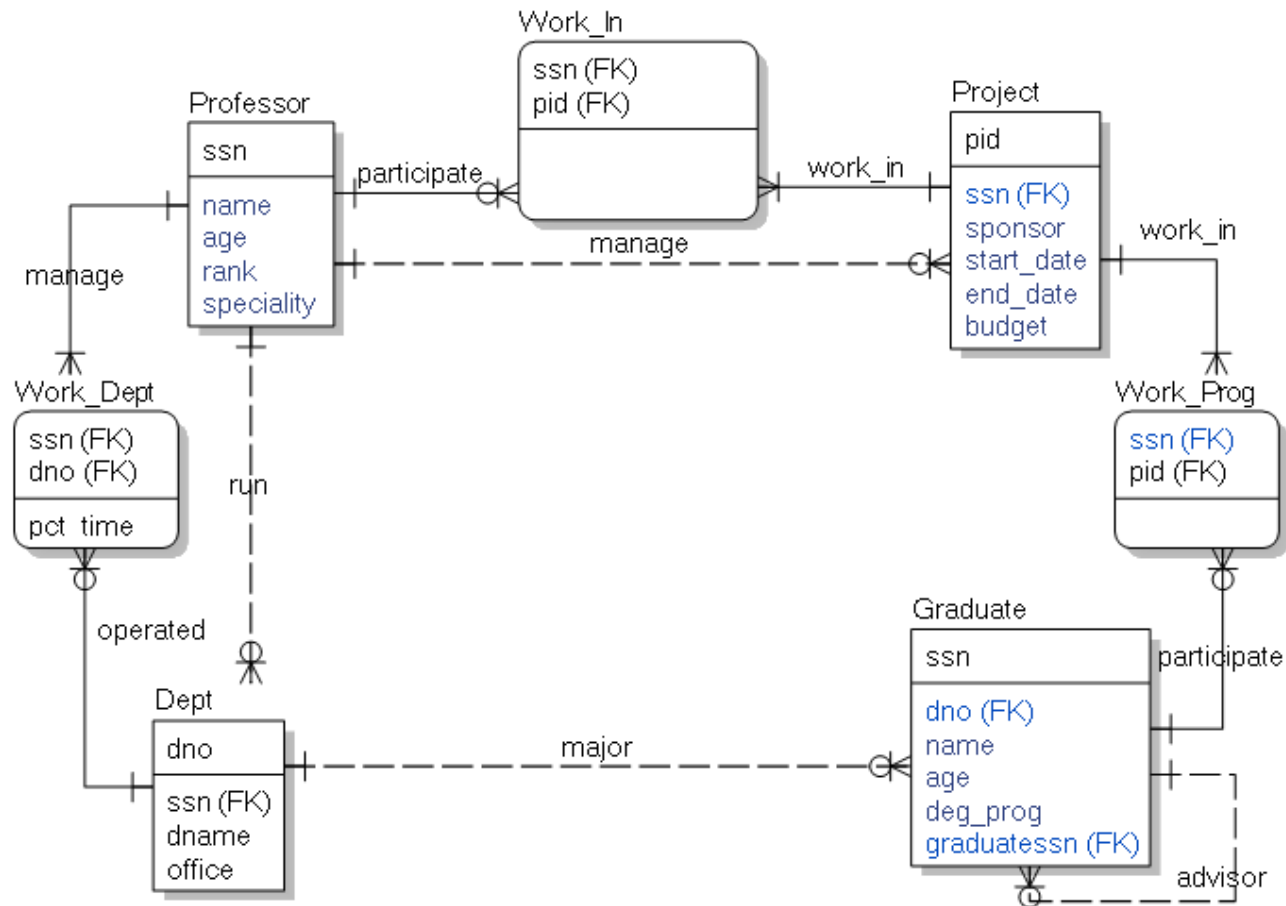


그림 6-81 마당대학 데이터베이스의 ER 다이어그램

# erwin 연습 - 마당 대학을 erwin으로 모델링해보자

- 관계 데이터 모델이 완성되면 실제 DBMS에 테이블을 생성하기 위한 물리적 모델링 과정을 거침





1. 데이터베이스 생명주기
2. 개념적 모델링
3. 논리적 모델링
4. 물리적 모델링
5. ER 모델
6. 개체와 개체 타입
7. 개체 타입의 종류
8. 속성
9. 속성의 종류
10. 관계와 관계 타입
11. 관계 대응 수
12. ISA(수퍼클래스와 서브클래스) 관계
13. 전체 참여와 부분 참여
14. 순환적 관계
15. 식별자
16. IE 표기법
17. 사상(매핑)