



## Chapter 06

# 데이터 모델링

MySQL로 배우는 데이터베이스 개론과 실습

# 목차

**01**

데이터 모델링의 개념

**02**

ER 모델

**03**

ER 모델을 관계 데이터 모델로 사상하기

**04**

모델링 실습

**05**

모델링 연습(마당대학 데이터베이스)

# 학습목표

- ❖ 데이터 모델링의 개념을 이해한다.
- ❖ 개념적 모델링 단계에 사용하는 ER 모델을 이해하고 작성 방법을 알아본다.
- ❖ MySQL Workbench를 사용하여 ER 모델을 작성해 본다.
- ❖ 마당대학 데이터베이스를 ER 모델로 직접 설계해 본다.

## 01 데이터 모델링의 개념

1. 데이터베이스 생명주기
2. 데이터 모델링 과정



# 데이터 모델링의 중요성

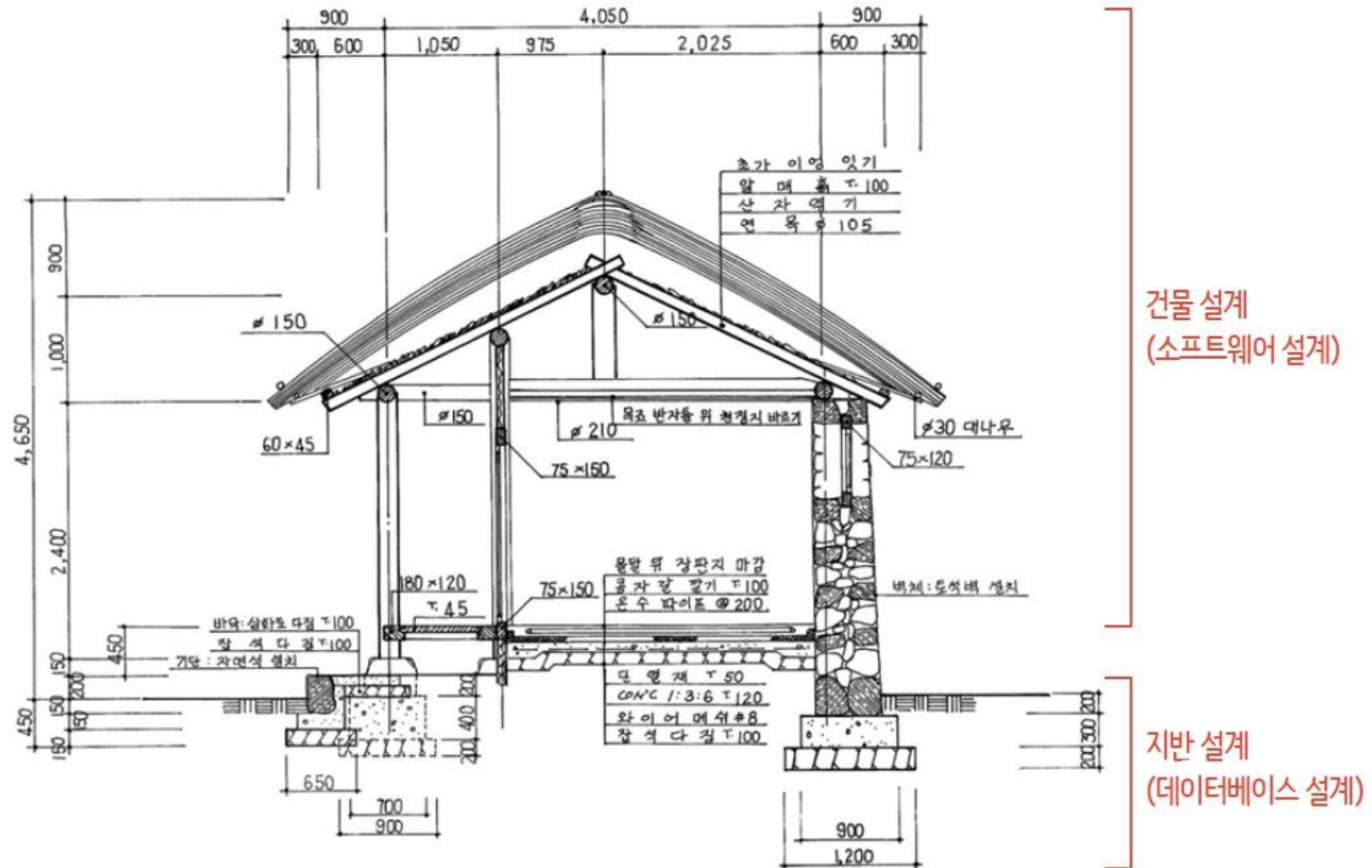


그림 6-1 데이터 모델링의 중요성

# 데이터 모델링의 중요성

- 데이터 모델링은 현실 세계의 복잡한 개념을 단순화·추상화하여 데이터베이스화하는 과정

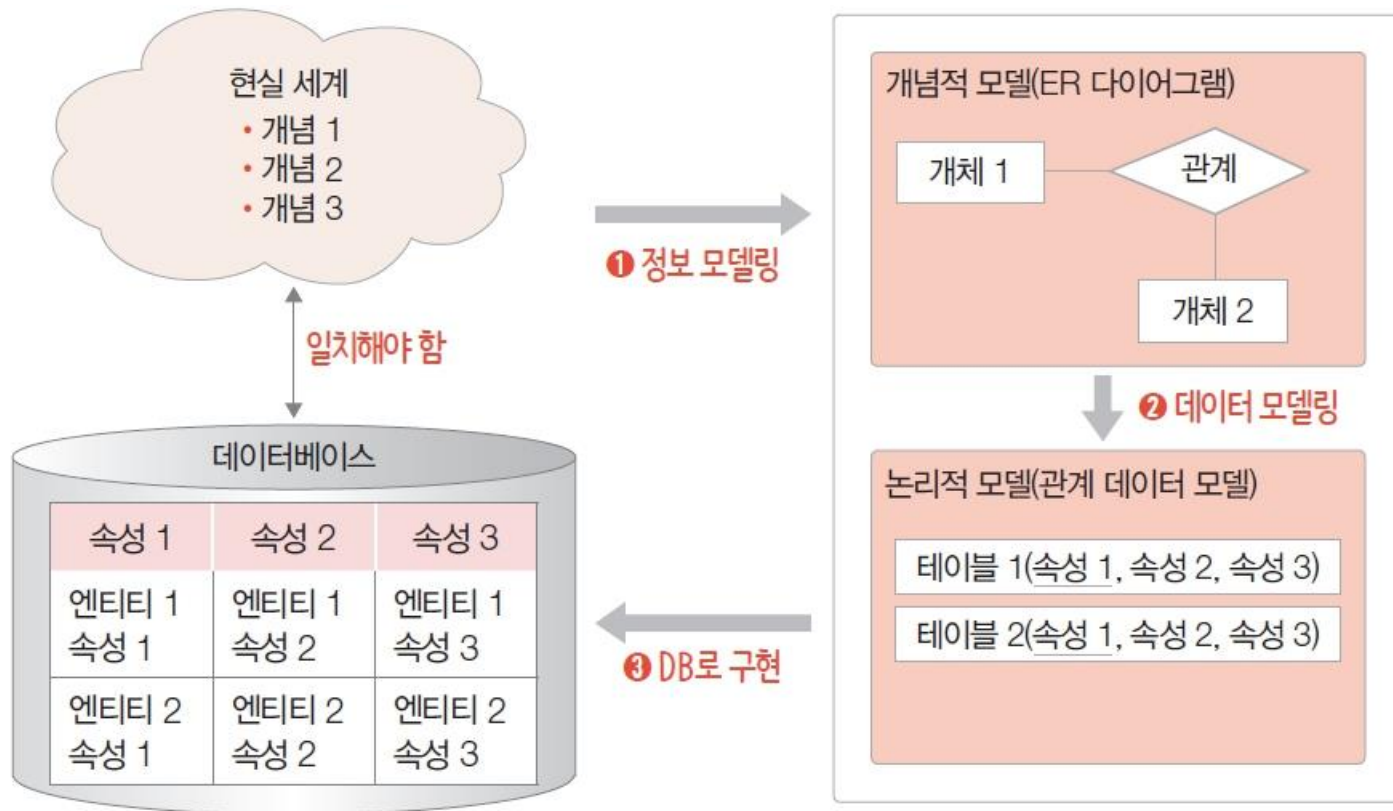


그림 6-2 데이터 모델링의 개념

# 1. 데이터베이스 생명주기

## ❖ 생명주기

- 데이터베이스는 최초 사용자의 요구에 의해 구축되고 사용되다가 필요에 따라 개선 또는 재구축되어 사용됨

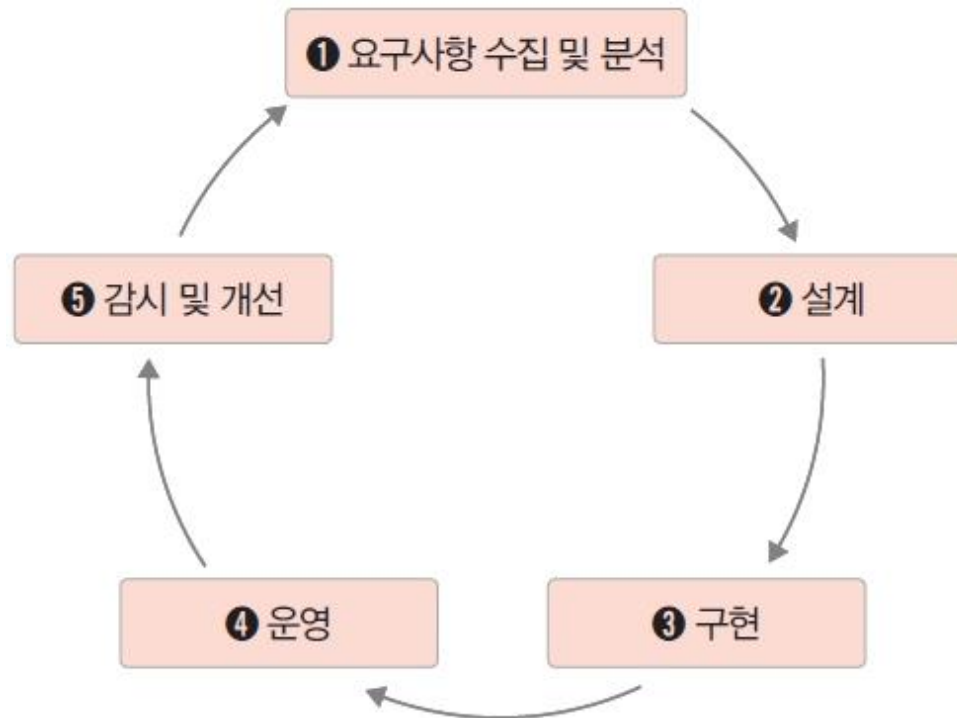


그림 6-3 데이터베이스 생명주기

## 2. 데이터 모델링 과정

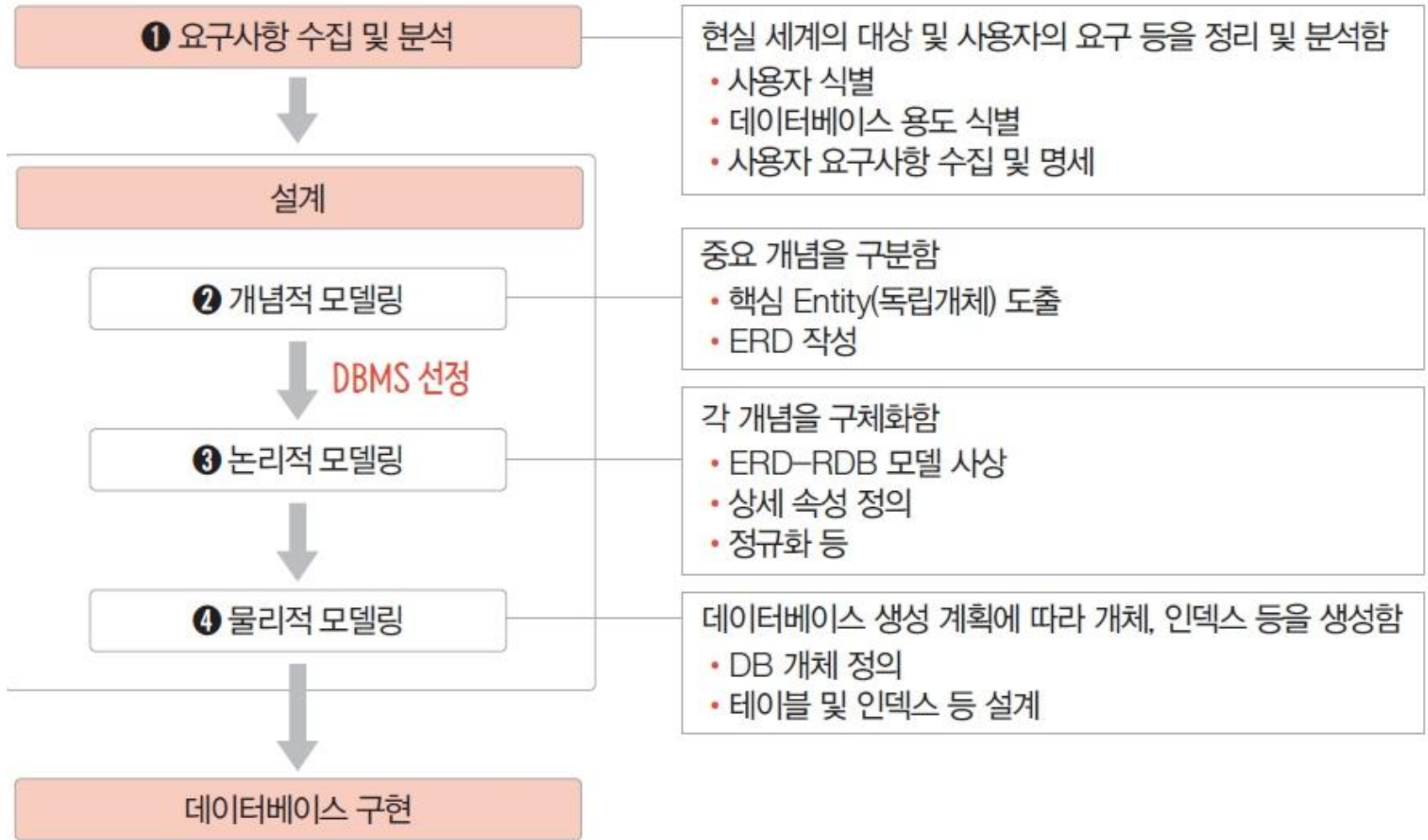


그림 6-4 데이터 모델링 과정



## 2. 데이터 모델링 과정

### ❖ 요구사항 수집 및 분석

- 실제 문서를 수집하고 분석한다
- 담당자와 인터뷰하거나 설문조사하여 요구사항을 직접 수렴한다
- 비슷한 업무를 처리하는 기존 데이터베이스를 분석한다
- 각 업무와 연관된 모든 부문을 살펴본다

### ❖ 개념적 모델링

- 요구사항을 수집하고 분석한 결과를 토대로 업무의 핵심 개념을 구분하고 전체적인 뼈대를 만드는 과정

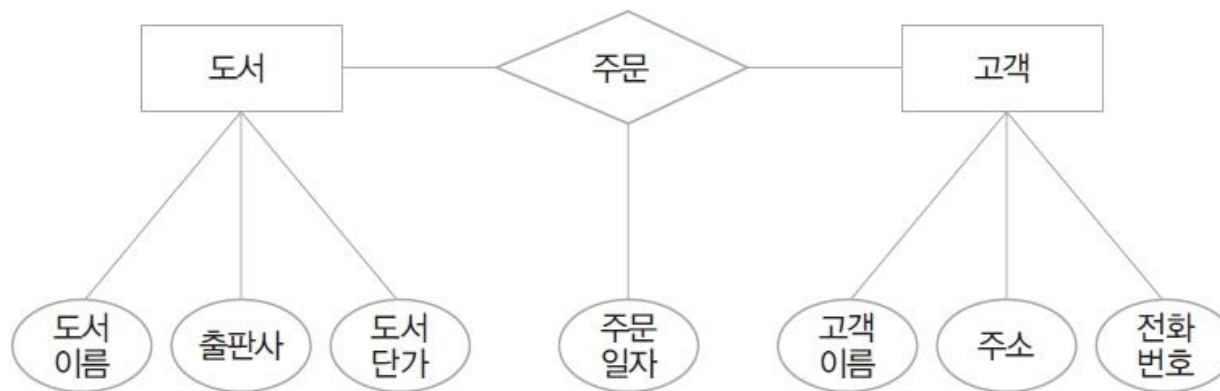


그림 6-5 개념적 모델링의 예

## 2. 데이터 모델링 과정

### ❖ 논리적 모델링

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

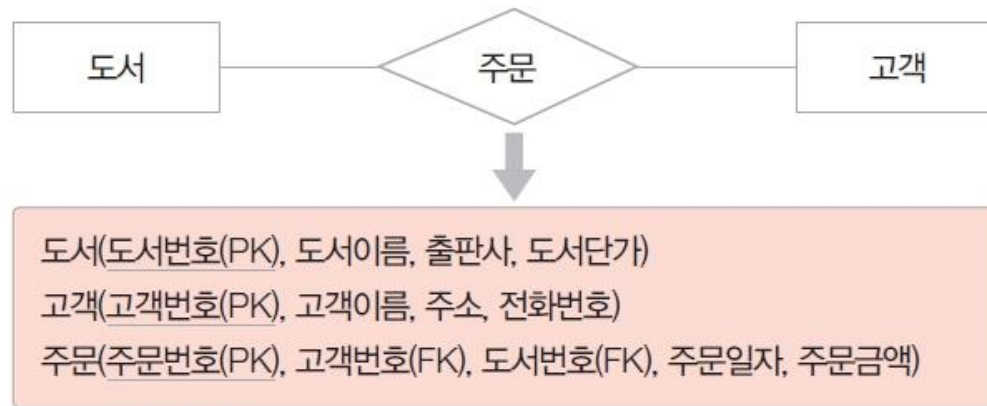


그림 6-6 논리적 모델링의 예

- 논리적 모델링은 다음과 같은 몇 가지 과정을 거쳐 완성됨
  - 개념적 모델링에서 추출하지 않았던 상세 속성들을 모두 추출한다
  - 정규화를 수행한다
  - 데이터의 표준화를 수행한다

## 2. 데이터 모델링 과정

### ❖ 물리적 모델링

- 작성된 논리적 모델을 실제 컴퓨터의 저장 장치에 저장하기 위한 물리적 구조를 정의하고 구현하는 과정

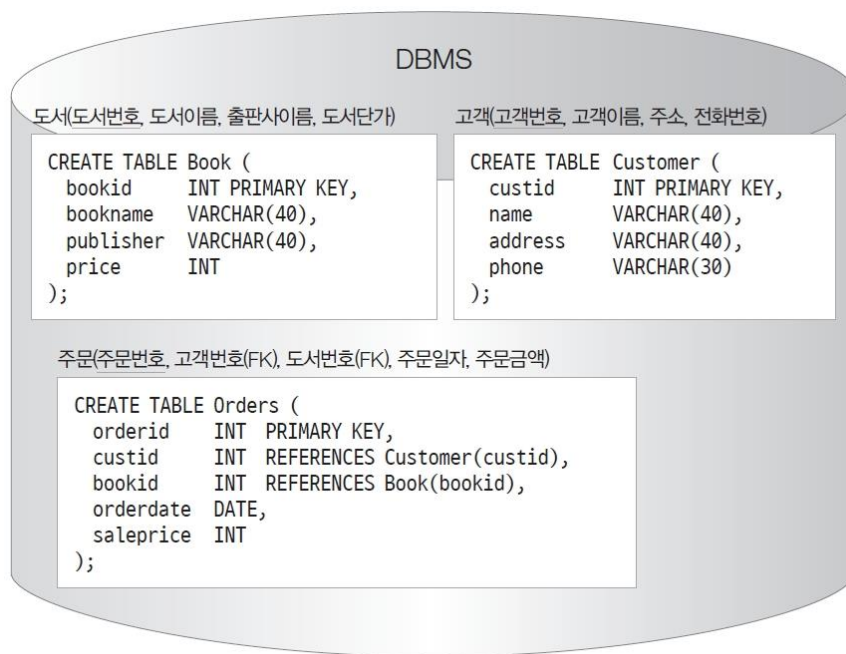


그림 6-7 물리적 모델링의 예

- 물리적 모델링을 할 때 트랜잭션, 저장 공간 설계 측면에서 고려할 사항
  - 응답시간을 최소화해야 한다
  - 얼마나 많은 트랜잭션을 동시에 발생시킬 수 있는지 검토해야 한다
  - 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



# ER 모델

## ❖ ER(Entity Relationship) 모델

- 데이터 모델링 과정 중 개념적 모델링에 사용하는 모델
- 세상의 사물을 개체와 개체 간의 관계로 표현함
- [그림 6-8]은 결혼을 예로 들어 ER 모델을 설명

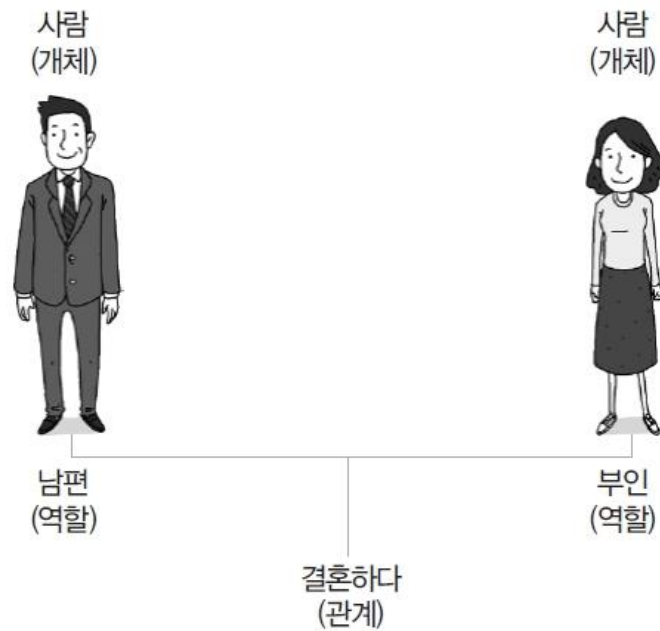


그림 6-8 ER 모델의 기본 개념

# ER 모델

- ER 모델은 개체와 개체 간의 관계를 ER 다이어그램이라는 표준화된 그림으로 표현
- 다음은 직원 개체와 프로젝트 개체 간의 관계를 ER 다이어그램으로 나타낸 것

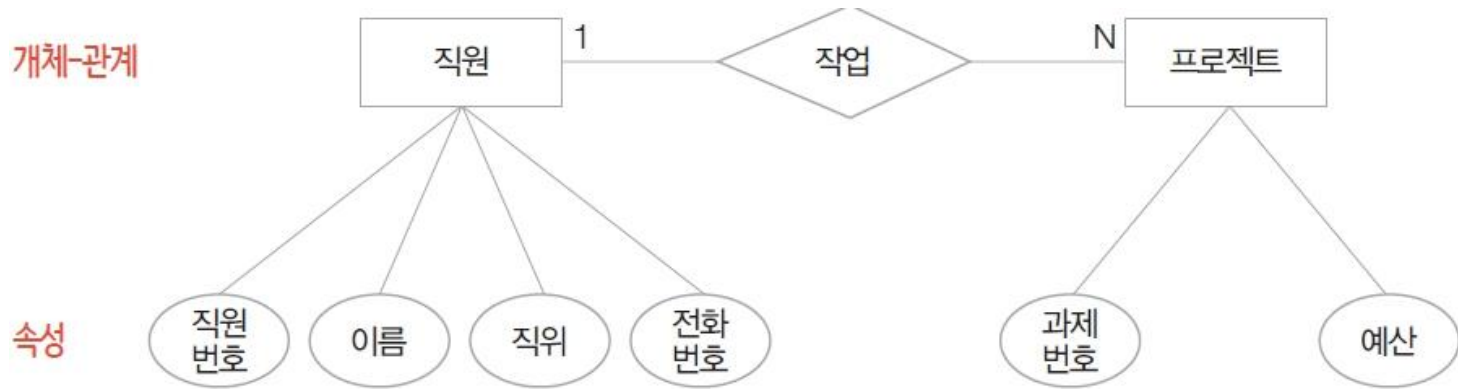


그림 6-9 ER 다이어그램

- ER 모델은 개념적 모델링 단계에서 사용하기 때문에 특정 데이터 모델이나 DBMS와 무관하게 설계할 수 있음

# 1. 개체와 개체 타입

## ❖ 개체

- 사람, 사물, 장소, 개념, 사건과 같이 유·무형의 정보를 가지고 있는 독립적인 실체를 말함
- 개체는 비슷한 속성을 가진 개체 타입을 구성하며, 개체 집합으로 묶임

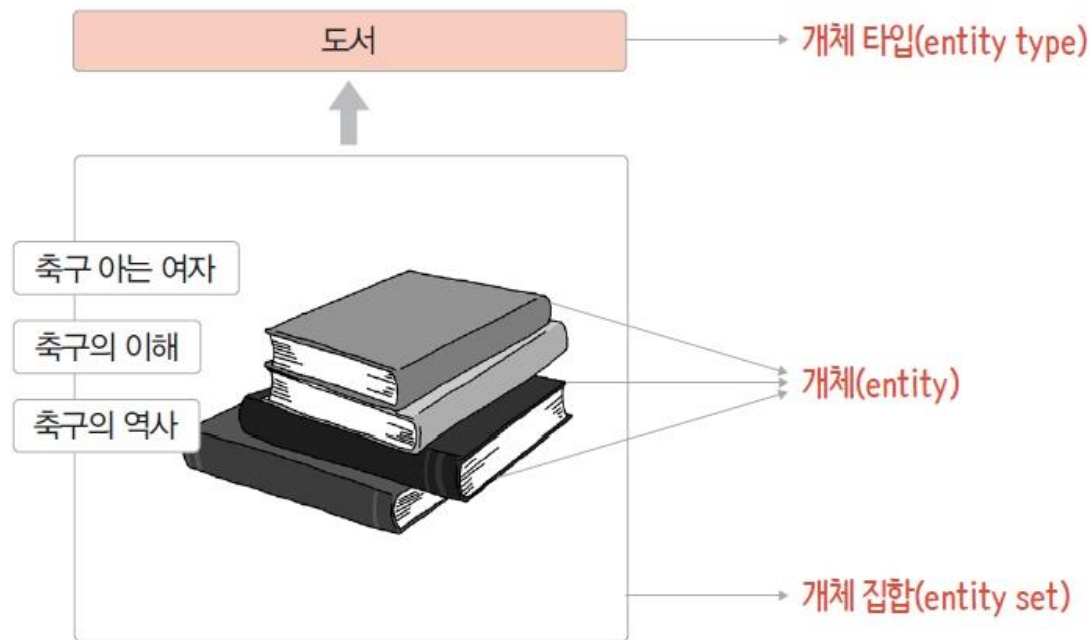


그림 6-10 개체, 개체 타입, 개체 집합


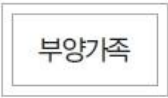
# 1. 개체와 개체 타입

## ❖ 개체의 특징

- 유일한 식별자에 의해 식별이 가능함
- 꾸준한 관리를 필요로 하는 정보
- 두 개 이상 영속적으로 존재함
- 업무 프로세스에 이용됨
- 반드시 자신의 특징을 나타내는 속성을 포함함
- 다른 개체와 최소 한 개 이상의 관계를 맺고 있음

## ❖ 개체 타입의 ER 다이어그램 표현

표 6-1 개체 타입의 ER 다이어그램 표현

기호	의미
	강한 개체 타입(개체 타입이라고 하면 보통 강한 개체 타입을 말함)
	약한 개체 타입



# 1. 개체와 개체 타입

---

## ❖ 개체 타입의 유형

- 강한(일반) 개체 타입
  - 다른 개체의 도움 없이 독자적으로 존재할 수 있음
- 약한 개체 타입
  - 독자적으로 존재할 수 없고 반드시 상위 개체 타입을 가짐

## 2. 속성

- 개별 개체가 가진 속성을 기본으로 개체 타입의 속성을 정의할 수 있음

표 6-2 개체 타입과 속성

개체 타입	속성
도서	도서이름, 출판사, 도서단가

### ❖ 속성의 ER 다이어그램 표현

- 속성은 기본적으로 타원으로 표현하며 개체 타입을 나타내는 직사각형과 실선으로 연결
- 속성의 이름은 타원의 중앙에 표기함
- 속성이 개체를 유일하게 식별할 수 있는 키일 경우 속성 이름에 밑줄을 그음

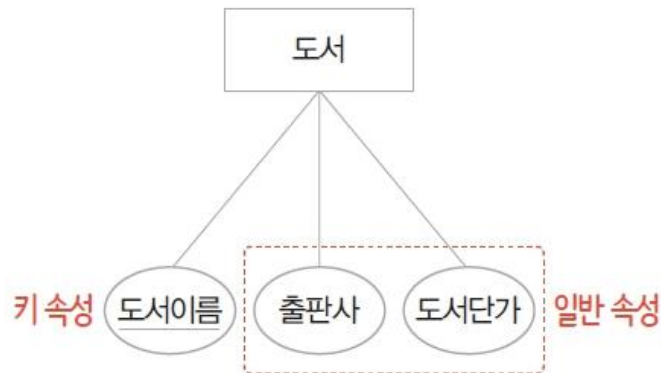


그림 6-11 도서 개체 타입

## 2. 속성 - 속성의 유형

- 단순 속성과 복합 속성

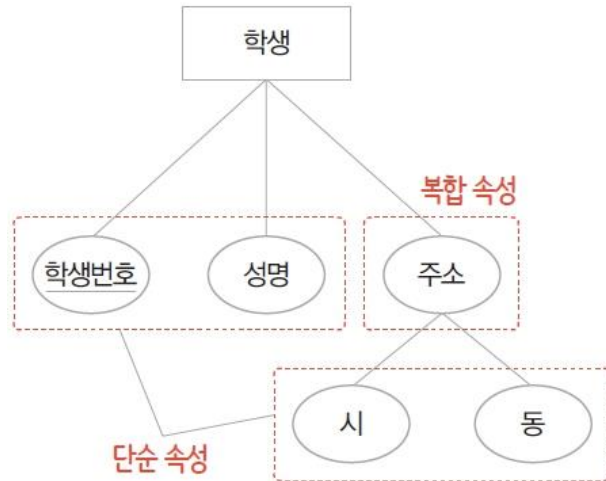


그림 6-12 단순 속성과 복합 속성

- 단일값 속성과 다중값 속성

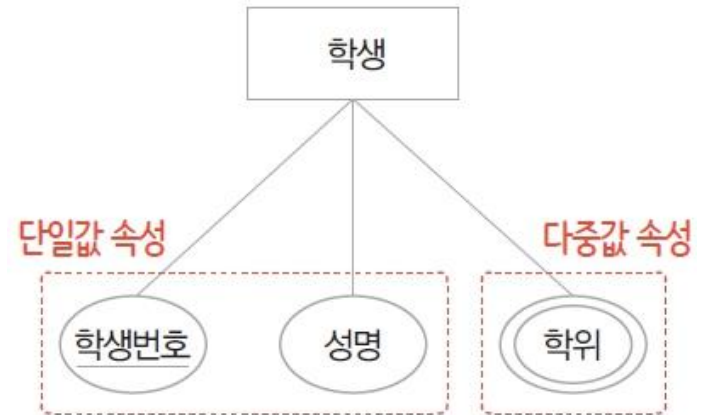


그림 6-13 단일값 속성과 다중값 속성

- 저장 속성과 유도 속성

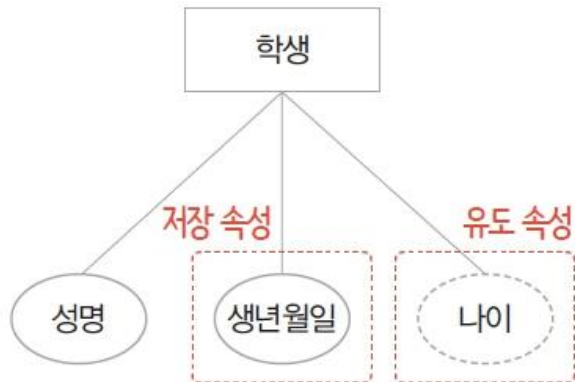


그림 6-14 저장 속성과 유도 속성

## 2. 속성 - 속성의 유형

표 6-3 속성의 ER 다이어그램 표현

기호	의미	설명
	속성	<ul style="list-style-type: none"> <li>일반적인 속성을 나타냄</li> <li>속성의 이름은 타원 중앙에 표시함</li> </ul>
	키(key) 속성	<ul style="list-style-type: none"> <li>속성이 개체를 유일하게 식별할 수 있는 키이면 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>식별자 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 표현함</li> </ul>
	유도 속성	<ul style="list-style-type: none"> <li>나이와 같이 출생 연도로 유도가 가능한 속성</li> <li>점선 타원으로 표현함</li> </ul>
	복합 속성	<ul style="list-style-type: none"> <li>주소(도시, 동, 번지)와 같이 여러 속성으로 구성된 속성</li> <li>큰 타원 아래 작은 타원으로 연결함</li> </ul>

### 3. 관계와 관계 타입

#### ❖ 관계

- 개체 사이의 연관성을 나타내는 개념

#### ❖ 관계 타입

- 개체 타입과 개체 타입 간의 연결 가능한 관계를 정의한 것이며, 관계 집합은 관계로 연결된 집합을 의미함

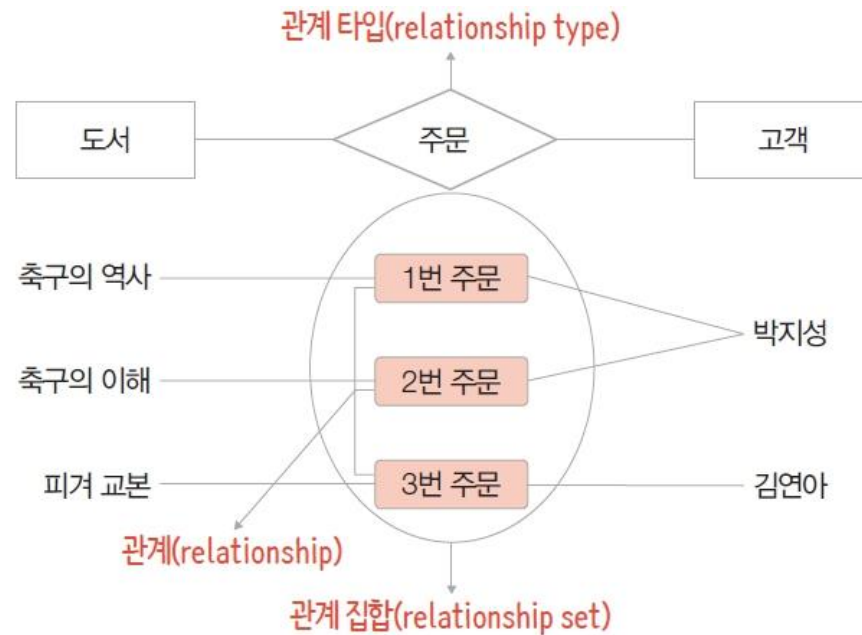


그림 6-15 관계, 관계 타입, 관계 집합

### 3. 관계와 관계 타입

#### ❖ 관계 타입의 ER 다이어그램 표현

표 6-4 관계 타입의 ER 다이어그램 표현

기호	의미
	관계 타입

- [그림 6-16]은 세 가지 관계의 예를 보여줌

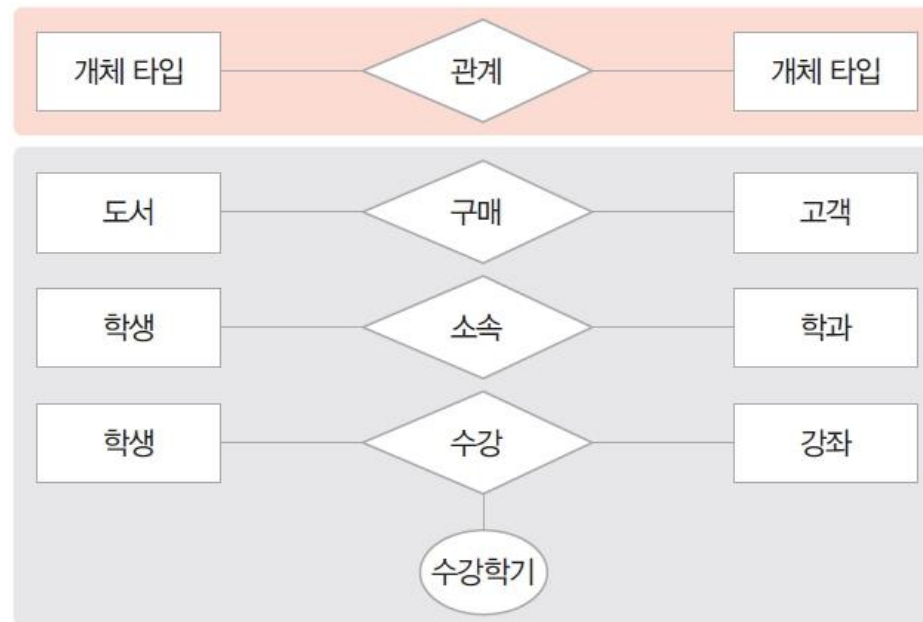


그림 6-16 관계의 예

### 3. 관계와 관계 타입

#### ❖ 관계 타입의 유형

- 차수(degree)에 따른 유형

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

1진 관계(recursive relationship)

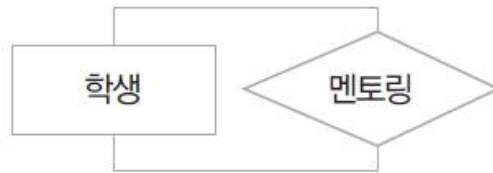


그림 6-17 1진 관계의 예

2진 관계(binary relationship)



그림 6-18 2진 관계의 예

3진 관계(ternary relationship)



그림 6-19 3진 관계의 예



### 3. 관계와 관계 타입

- 관계 대응 수에 따른 유형

표 6-6 관계 대응 수에 따른 관계 타입의 유형

기호	의미	설명
1 — 관계 — 1	일대일 관계	하나의 개체가 하나의 개체에 대응함
1 — 관계 — N	일대다 관계	하나의 개체가 여러 개체에 대응함
N — 관계 — 1	다대일 관계	여러 개체가 하나의 개체에 대응함
N — 관계 — M	다대다 관계	여러 개체가 여러 개체에 대응함

#### 일대일(1:1) 관계

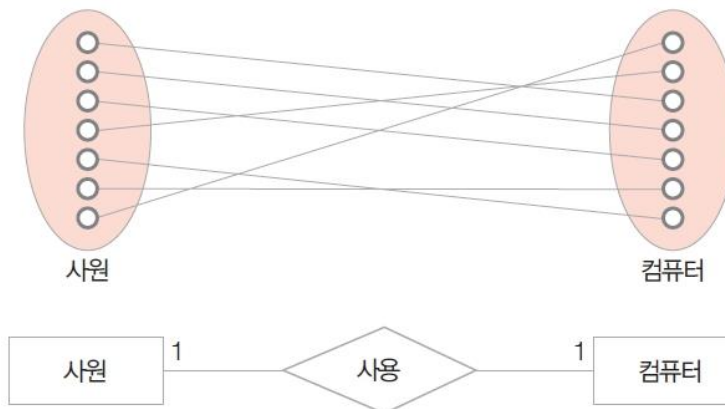


그림 6-20 일대일 관계의 예

### 3. 관계와 관계 타입

일대다(1:N), 다대일(N:1) 관계

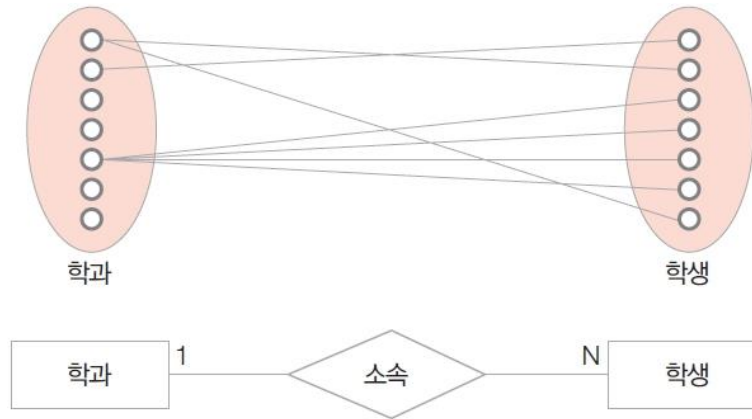


그림 6-21 일대다, 다대일 관계의 예

다대다(N:M) 관계

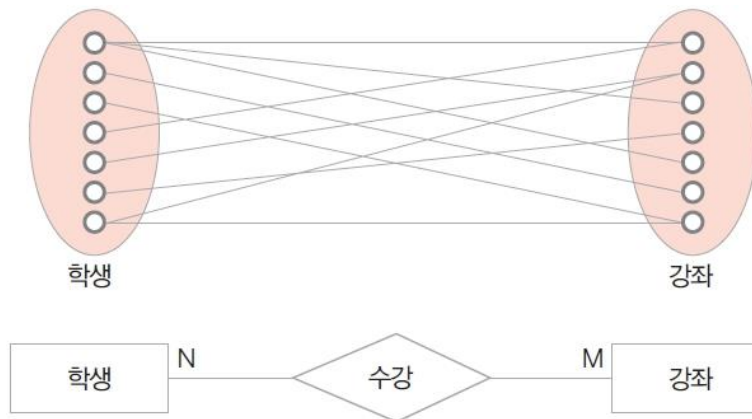


그림 6-22 다대다 관계의 예

### 3. 관계와 관계 타입

- 관계 대응수(cardinality)의 최소값과 최대값
  - 관계 대응 수 1:1, 1:N, N:M에서 1, M, N은 각 개체가 관계에 참여하는 최대값을 의미
  - ER 다이어그램에서는 대응 수 외에 최소값과 최대값을 관계 실선 위에 (최소값, 최대값)으로 표기함



그림 6-23 관계 대응 수의 최소값과 최대값 표기

- [표 6-7]은 1:1, 1:N, N:M 관계를 (min, max)를 사용해 표기한 것

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



그림 6-24 (최소값, 최대값) 표기의 예 1:N 관계 대응수 예

### 3. 관계와 관계 타입

#### ❖ ISA(is-a) 관계

- 상위 개체 타입의 특성에 따라 하위 개체 타입이 결정되는 형태

표 6-8 ISA 관계

기호	의미
	개체 간 상하 관계

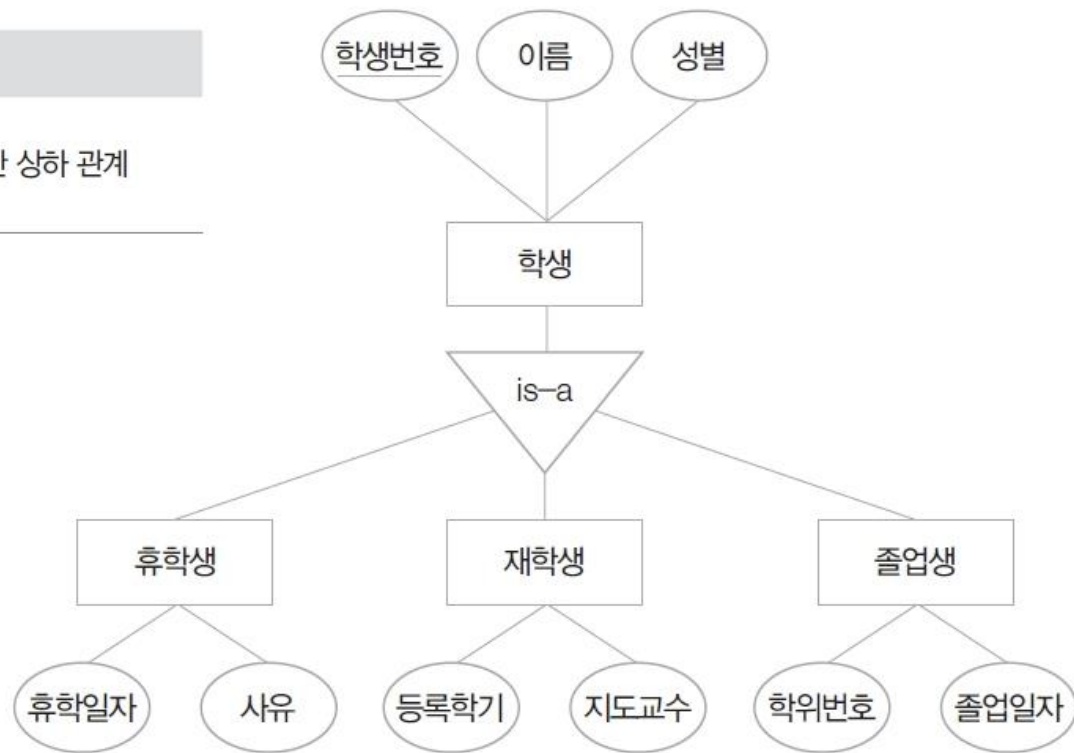


그림 6-25 ISA 관계의 예

### 3. 관계와 관계 타입

#### ❖ 참여 제약조건

- 개체 집합 내 모든 개체가 관계에 참여하는지에 따라 전체 참여와 부분 참여로 구분할 수 있음
- 전체 참여는 개체 집합의 모든 개체가 관계에 참여하고, 부분 참여는 일부만 참여함
- 전체 참여를 (최솟값, 최댓값)으로 표현하면 최솟값이 1 이상으로 모두 참여한다는 뜻이고, 부분 참여는 최솟값이 0 이상

표 6-9 관계의 참여 제약조건


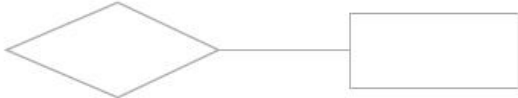
기호	의미
	전체 참여
	부분 참여



그림 6-26 부분 참여와 전체 참여의 예

### 3. 관계와 관계 타입

#### ❖ 역할(role)

- 개체 타입 간의 관계를 표현할 때 각 개체는 고유한 역할을 담당
- 관계에서 역할이 명확하지 않을 때는 반드시 표기해야 함



그림 6-27 역할의 예

#### ❖ 순환적 관계(recursive relationship)

- 하나의 개체 타입이 동일한 개체 타입(자기 자신)과 순환적으로 관계를 맺는 형태

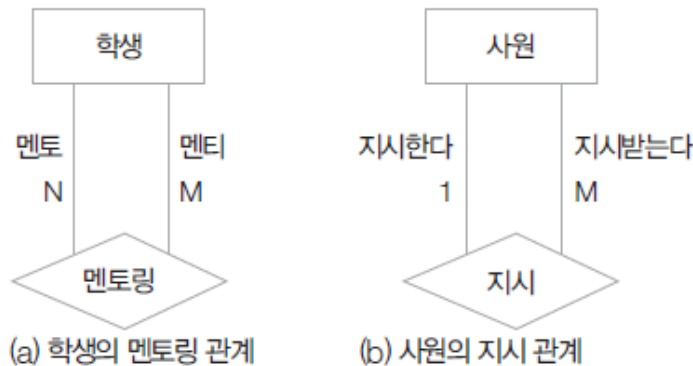


그림 6-28 순환적 관계의 예

## 4. 약한 개체 타입과 식별자

### ❖ 약한 개체(weak entity) 타입

- 가족 개체 타입처럼 상위 개체 타입이 결정되지 않으면 개별 개체를 식별할 수 없는 종속된 개체 타입
- 자신의 키(식별자, discriminator)로 식별하기 어려우므로 상위에 있는 강한 개체(strong entity) 타입의 기본키를 상속받아 사용함

표 6-10 약한 개체 타입과 식별자

기호	의미	설명
	약한 개체 타입	<ul style="list-style-type: none"><li>• 강한 개체 타입이 있어야 존재할 수 있음</li><li>• 이중 직사각형으로 표현함</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 표현함</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

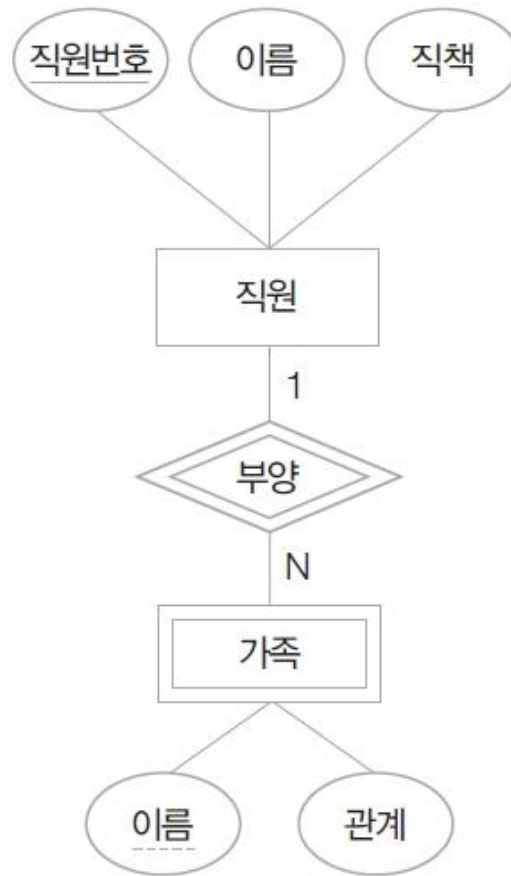


그림 6-29 약한 개체 타입과 식별자의 예



# 5. IE 표기법

- ER 모델 표기법에는 대표적으로 IE 표기법과 바커 표기법이 있음

## ❖ IE(Information Engineering) 표기법

- 관계 대응 수를 새발 모양의 기호로 표현하여 새발 표기법이라고 부르기도 함

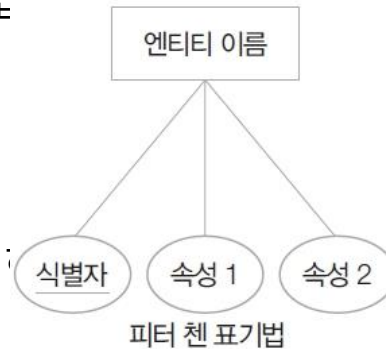
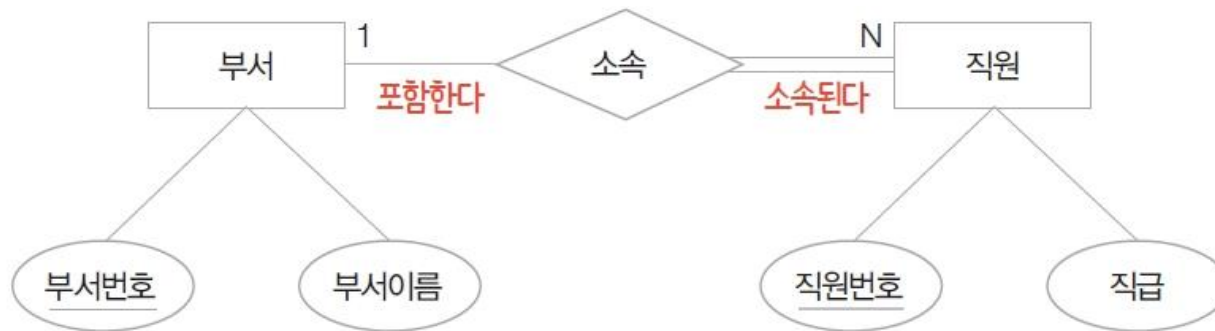


그림 6-30 피터 첸 표기법과 IE 표기법

표 6-11 IE 표기법-관계와 관계 대응 수

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 표시함</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



(a) ER 표기법으로 작성한 부서-직원 관계



(b) IE 표기법으로 작성한 부서-직원 관계

그림 6-31 IE 표기법의 예(비식별자 관계)

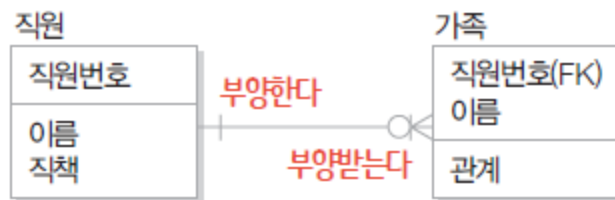


그림 6-32 IE 표기법의 예(식별자 관계)

## 연습문제 (Q6.1)

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 사상하기

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



# ER 모델을 관계 데이터 모델로 사상

- 완성된 ER 모델이 실제 데이터베이스로 구축되려면 논리적 모델링 단계를 거쳐야 하는데, 이 단계에서 사상(mapping)이 이루어짐

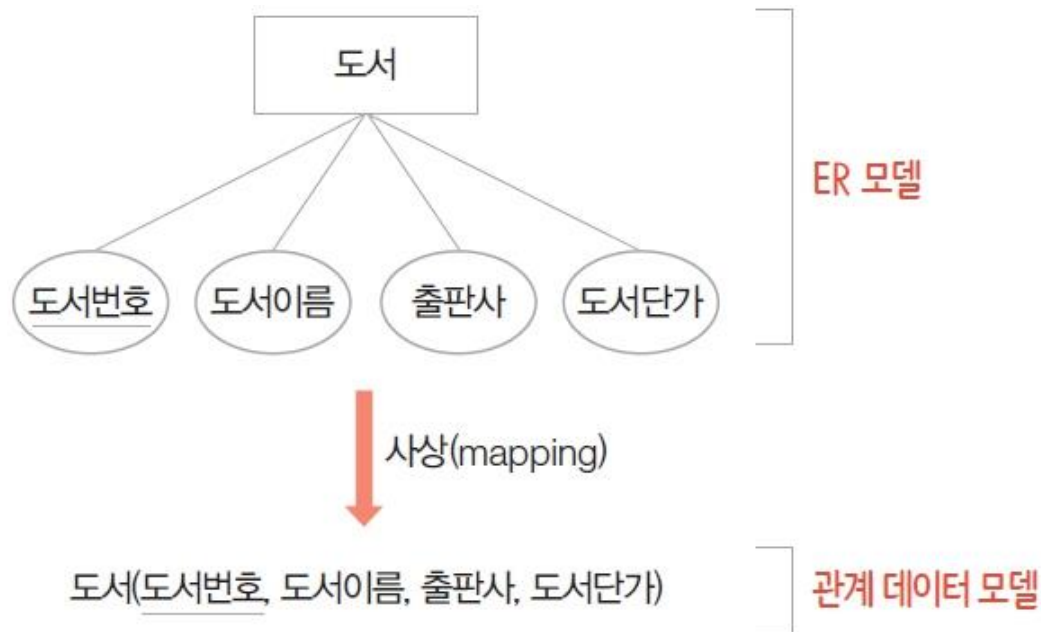


그림 6-33 ER 모델을 관계 데이터 모델로 사상

# ER 모델을 관계 데이터 모델로 사상

- 관계 데이터 모델은 릴레이션 하나로 데이터베이스의 논리적 구조를 표현하지만 ER 모델은 개체 타입, 관계 타입, 속성으로 표현함

표 6-12 ER 모델과 관계 데이터 모델의 사상 알고리즘

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		N진 관계 타입
7단계	속성	다중값 속성

# 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 넣어 자신의 기본키를 구성함

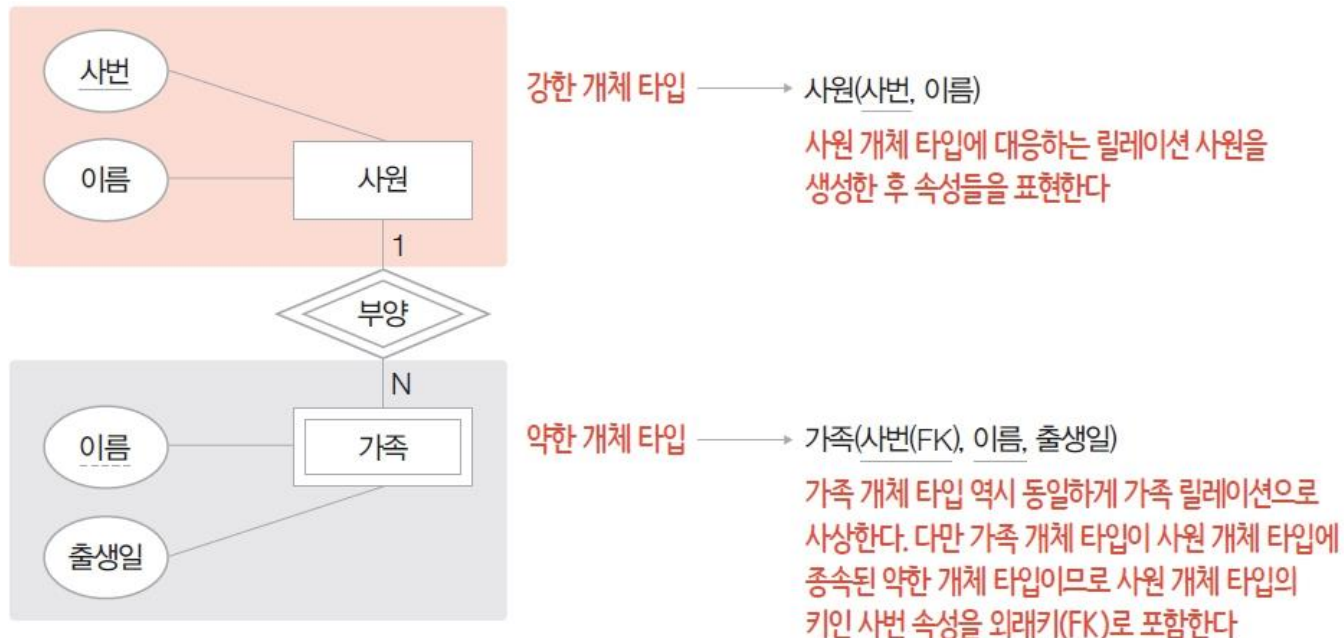


그림 6-34 개체 타입의 사상

## 2. 관계 타입의 사상

- 관계 타입은 각 관계 타입이 맺고 있는 차수와 관계 대응 수에 따라 사상 방식을 구분

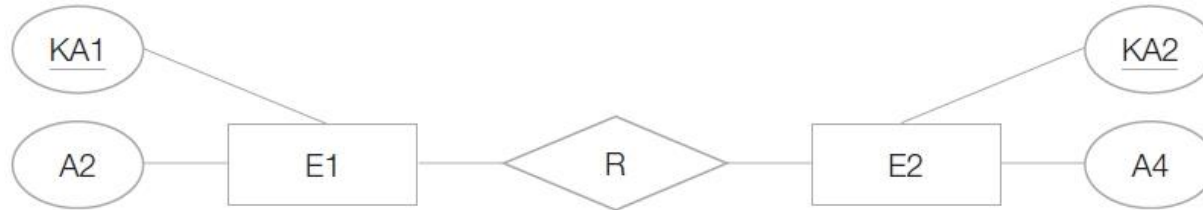


그림 6-35 이진 관계 타입

[방법 1] 오른쪽 개체 타입 E2를 기준으로 관계 R을 표현한다.

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

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

[방법 2] 왼쪽 개체 타입 E1을 기준으로 관계 R을 표현한다.

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

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

[방법 3] 단일 릴레이션 ER로 모두 통합하여 관계 R을 표현한다.

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

[방법 4] 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 [방법 1]~[방법 4]까지 모든 유형으로 사상할 수 있음

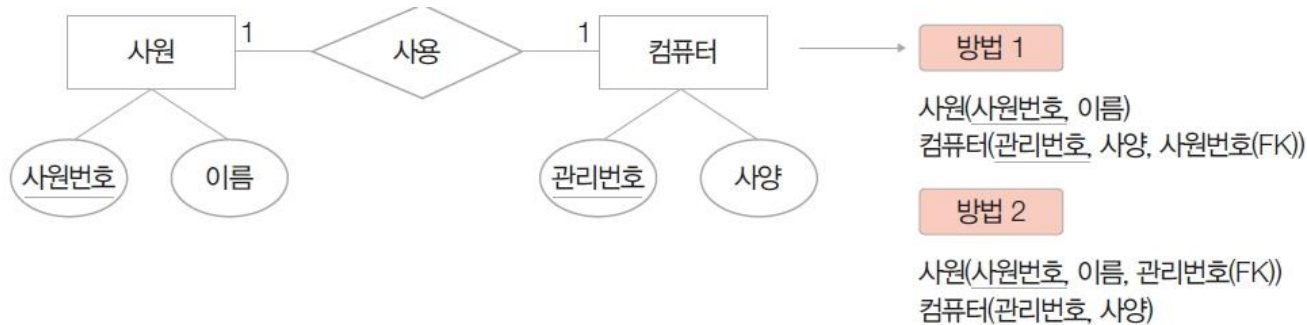


그림 6-36 이진 1:1 관계 타입의 사상

### ❖ [4단계] 이진 1:N 관계 타입

- 이진 1:N 관계 타입의 경우 N의 위치에 따라 [방법 1] 또는 [방법 2]의 유형으로 사상됨

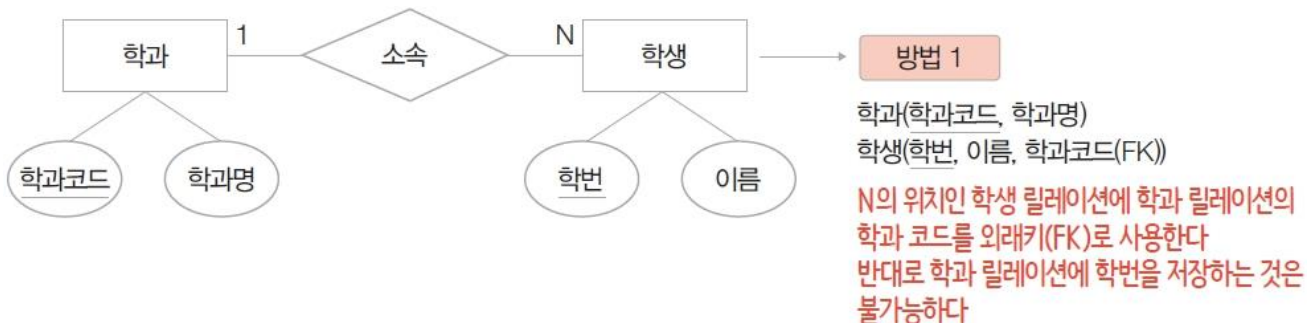


그림 6-37 이진 1:N 관계 타입의 사상

## 2. 관계 타입의 사상

### ❖ [5단계] 이진 N:M 관계 타입

- 이진 N:M 관계 타입은 [방법 4]의 유형으로 사상됨

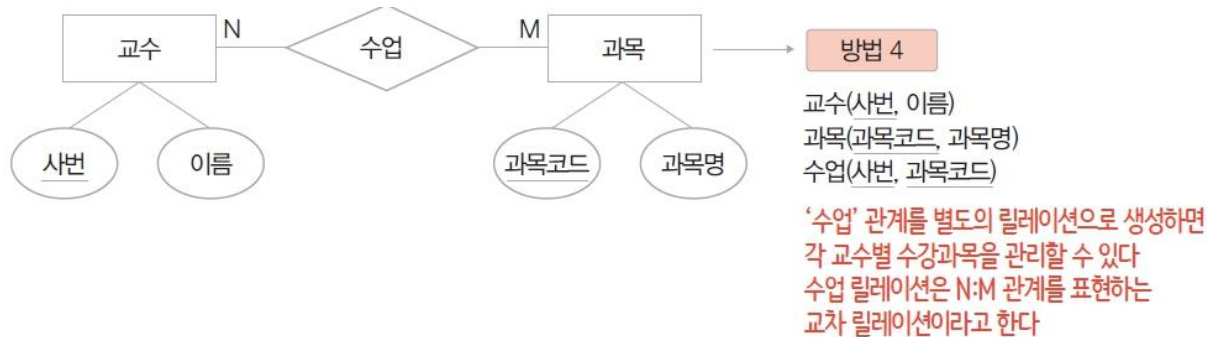


그림 6-38 이진 N:M 관계 타입의 사상

### ❖ [6단계] N진 관계 타입

- ER 모델의 차수가 3 이상인 다진 관계 타입의 경우 [방법 4]의 유형으로 사상됨

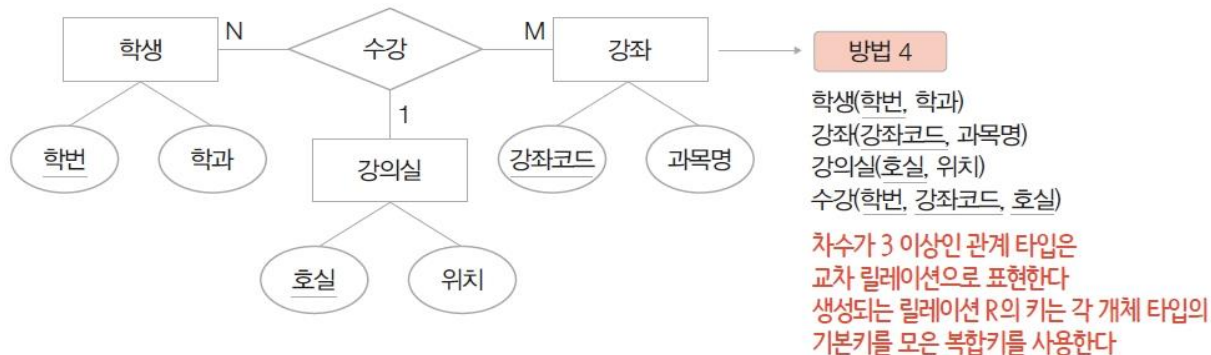


그림 6-39 N진 관계 타입의 사상

### 3. 다중값 속성의 사상

- 다중값 속성은 하나의 속성에 여러 값을 가질 수 있는 속성을 말함
- 다중값 속성을 가진 ER 모델을 관계 데이터 모델로 사상하는 방법에는 다음과 같이 두 가지가 있음

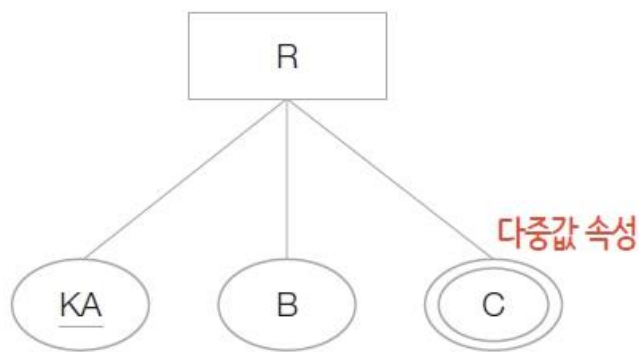


그림 6-40 다중값 속성의 사상

[방법 1] C 값의 수를 알 수 없는 경우

$R(\underline{KA}, B)$   
 $RC(\underline{KA}, \underline{C})$

[방법 2] C 값의 수가 세 개 이하일 경우

$R(\underline{KA}, B, C1, C2, C3)$

#### ❖ [7단계] 다중값 속성

- 속성의 개수를 알 수 없는 경우 [방법 1]을, 속성의 개수가 제한적으로 정해지는 경우 [방법 2]를 사용

[방법 1] 다중값 속성의 개수를 알 수 없는 경우

$R(\underline{KA}, B)$   
 $RC(\underline{KA}, \underline{C})$

[방법 2] 다중값 속성의 개수가 적고 제한 가능한 경우

$R(\underline{KA}, B, C1, C2, C3)$

### 3. 다중값 속성의 사상

- [방법 1]은 새로 생성된 릴레이션 RC에 값이 계속 저장될 수 있어 [방법 2]보다 효율적임
- 학생에 관한 개체를 모델링할 경우 [그림 6-41]과 같이 취미 속성을 앞에서 설명한 [방법 1]로 나타낼 수 있음



그림 6-41 다중값 속성의 사상 예

## 연습문제 (Q6.2)

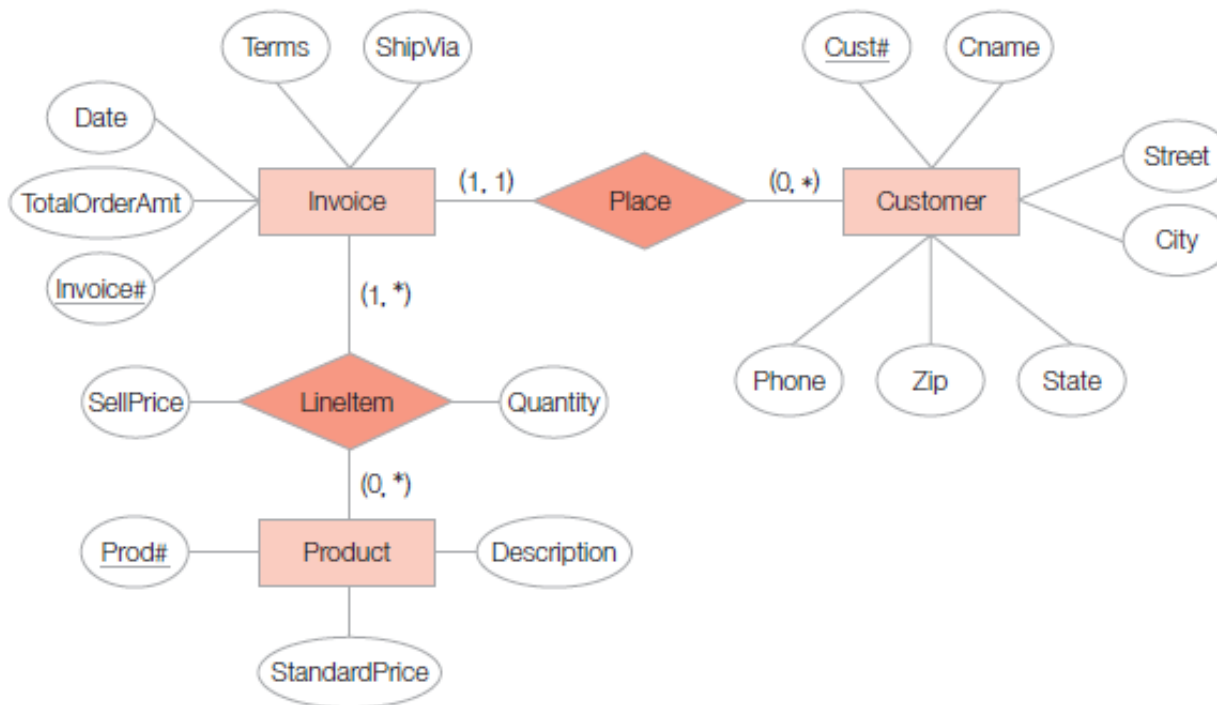
**07** 다음 내용을 모두 포함하는 데이터베이스를 설계하시오. 필요한 경우 몇 가지 가정을 넣을 수 있다.

- 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 테이블로 변환하시오.
- (3) MySQL Workbench를 이용하여 모델링하시오.

## 연습문제 (Q6.3)

- 10** 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 모델링 실습

1. 마당서점 설계 실습
2. 마당서점 ERD 만들기



# ER 다이어그램을 만들기

❖ Workbench를 이용하여 [그림 6-42]의 ER 다이어그램을 만들기

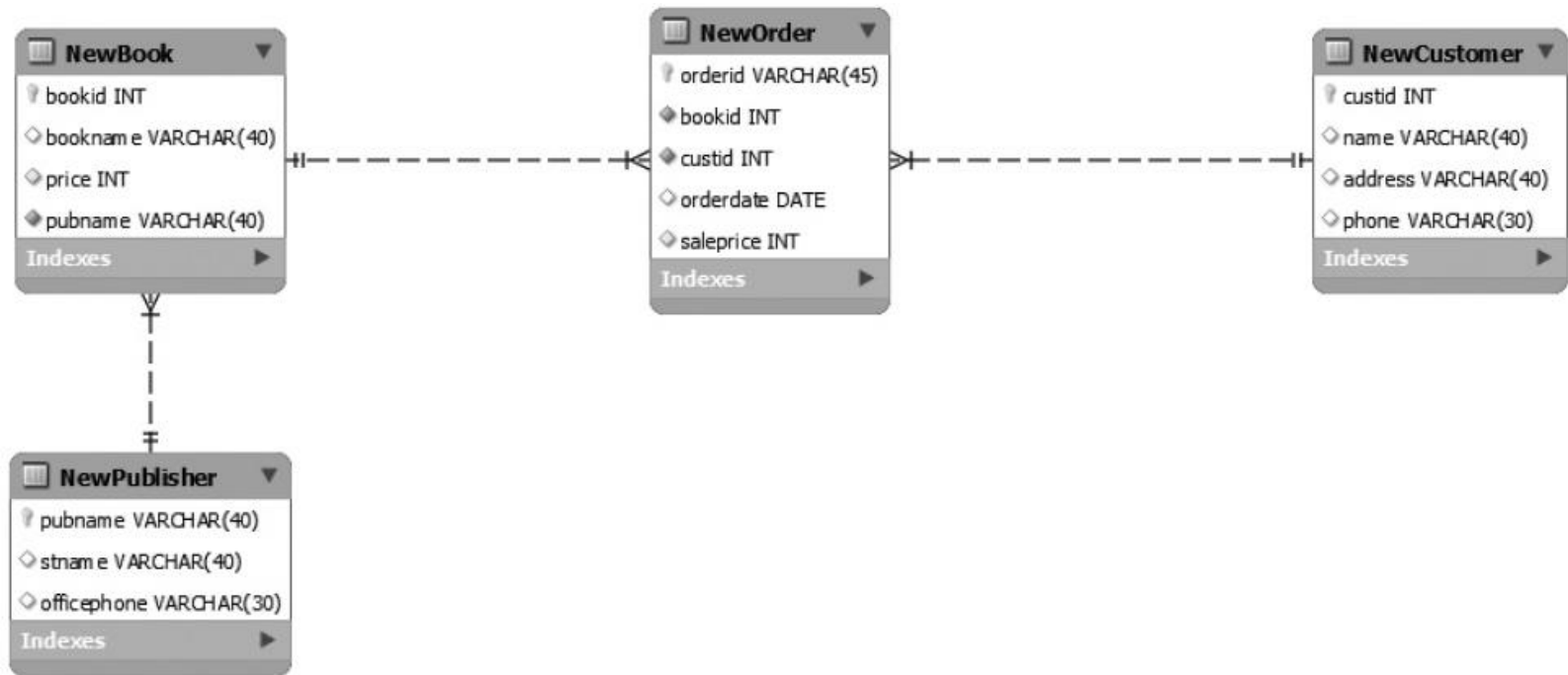


그림 6-42 마당서점의 ER 다이어그램



# 1. 마당서점 설계 실습

## ❖ 마당서점의 개념/논리적 모델링

### ■ 개체

- 마당서점의 요구사항을 살펴보면 개체는 총 3개, 즉 도서 개체, 출판사 개체, 고객 개체가 필요

표 6-13 마당서점의 개체와 속성

개체	키 속성(PK)	속성
출판사	출판사이름	담당자이름, 전화번호
도서	도서번호	도서이름, 도서단가, 출판사이름(FK)
고객	고객번호	고객이름, 주소, 전화번호

### ■ 관계 표현하기

#### 1:N 비식별 관계

- 한 출판사는 여러 권의 도서를 서점에 공급할 수 있으므로 출판사와 도서의 관계는 1:N

#### N:M 식별 관계

- 도서는 여러 명의 고객에게 판매될 수 있고 고객은 여러 권의 도서를 구입할 수 있으므로 도서와 고객 개체는 N:M 관계

# 1. 마당서점 설계 실습

- 식별 관계 및 관계 대응 수 변경하기



그림 6-43 마당서점 개체와 관계

## ❖ 도메인 정의하기

표 6-14 마당서점의 개체와 속성

개체	출판사	도서	주문	고객
기본키	출판사이름	도서번호	주문번호	고객번호
속성	담당자이름 전화번호	도서이름 도서단가 출판사이름(FK)	고객번호(FK) 도서번호(FK) 주문일자 주문금액	고객이름 주소 전화번호

# 1. 마당서점 설계 실습

- 공통으로 사용되는 속성을 도메인으로 미리 정의해 놓으면 향후 데이터 타입 변경의 혼란을 줄일 수 있음

표 6-15 마당서점의 도메인별 데이터 타입 정의표

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

## 2. 마당서점 ERD 만들기

### ❖ 기본 화면 및 툴 둘러보기

- Workbench의 홈 화면에서 [File] - [New Model]을 선택

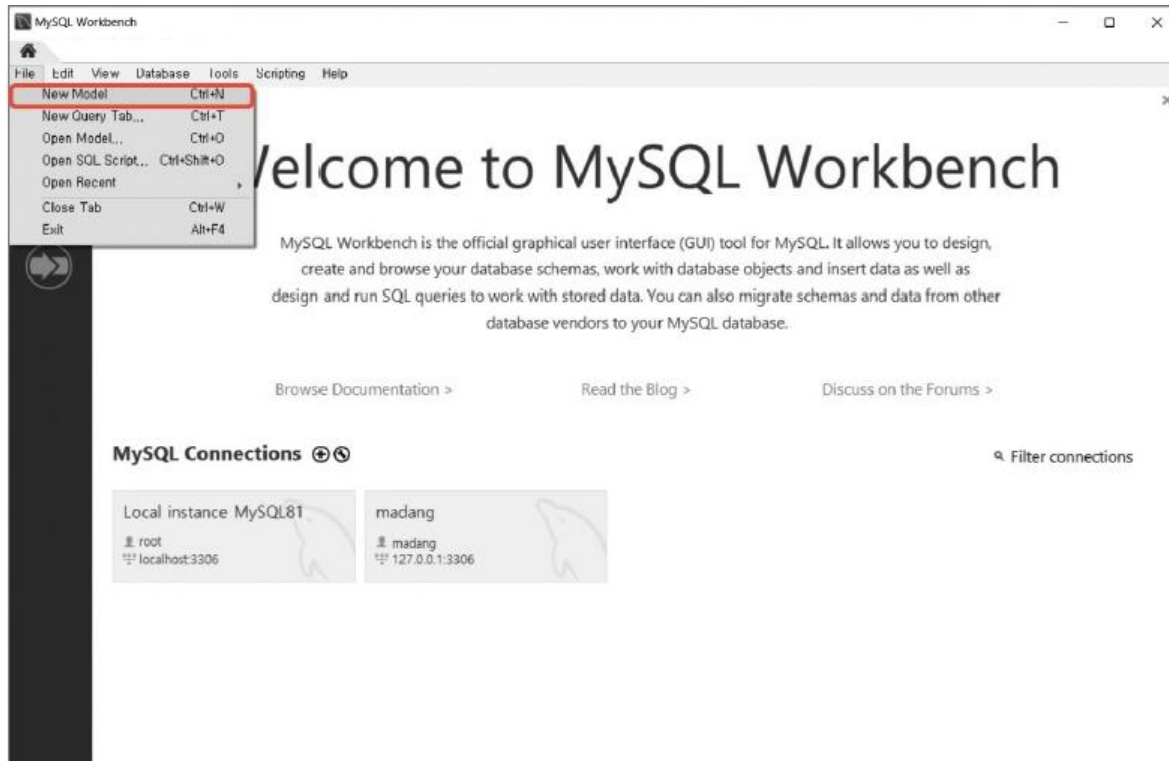


그림 6-44 Workbench New Model 실행

## 2. 마당서점 ERD 만들기

- mydb에서 우클릭하여 Edit Schema를 선택

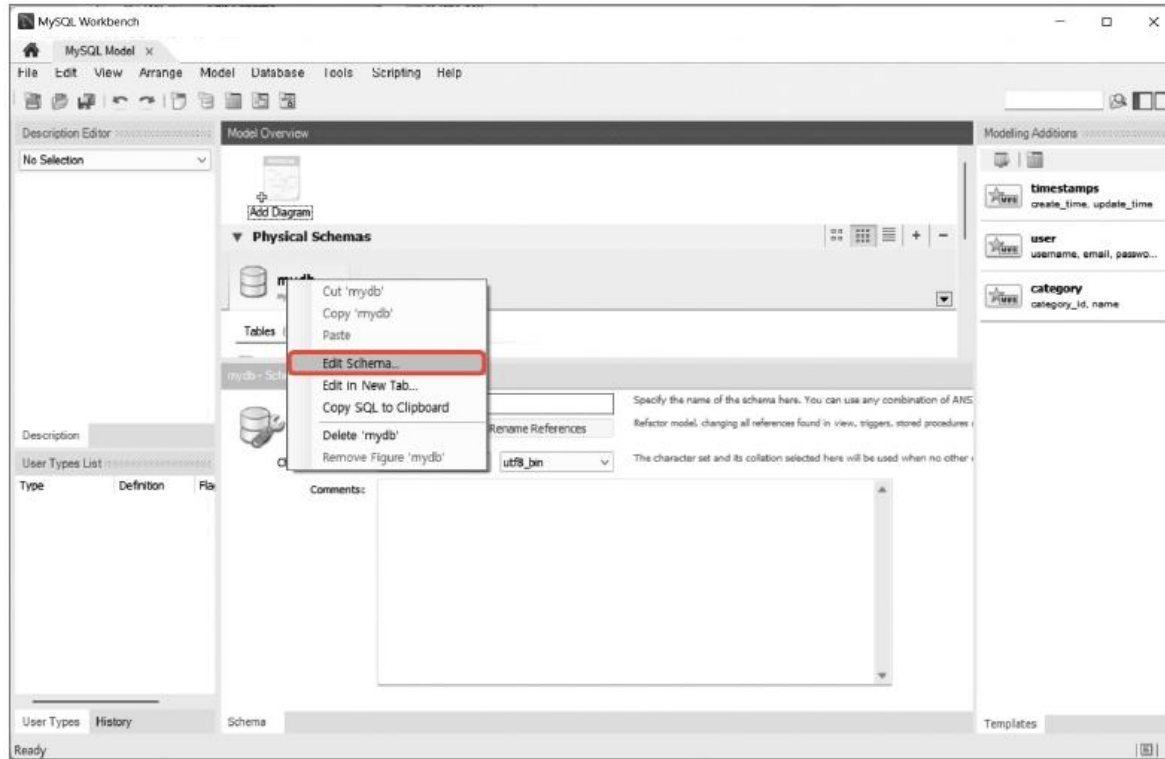


그림 6-45 Edit Schema 선택

## 2. 마당서점 ERD 만들기

- 스키마 이름을 'newmadangdb'로 변경함
- 스키마 이름이 변경되면 'Add Diagram'을 선택(더블클릭)하여 본격적인 ERD 작성 화면에 진입함

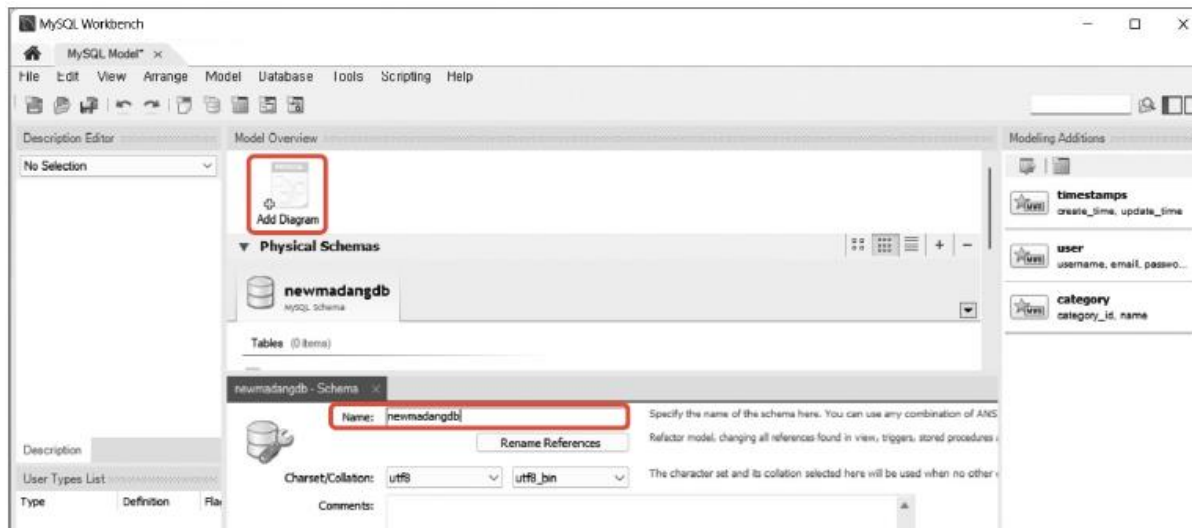


그림 6-46 스키마 이름 변경

## 2. 마당서점 ERD 만들기

- Workbench의 다이어그램 작성을 위한 기본 화면은 다음과 같음

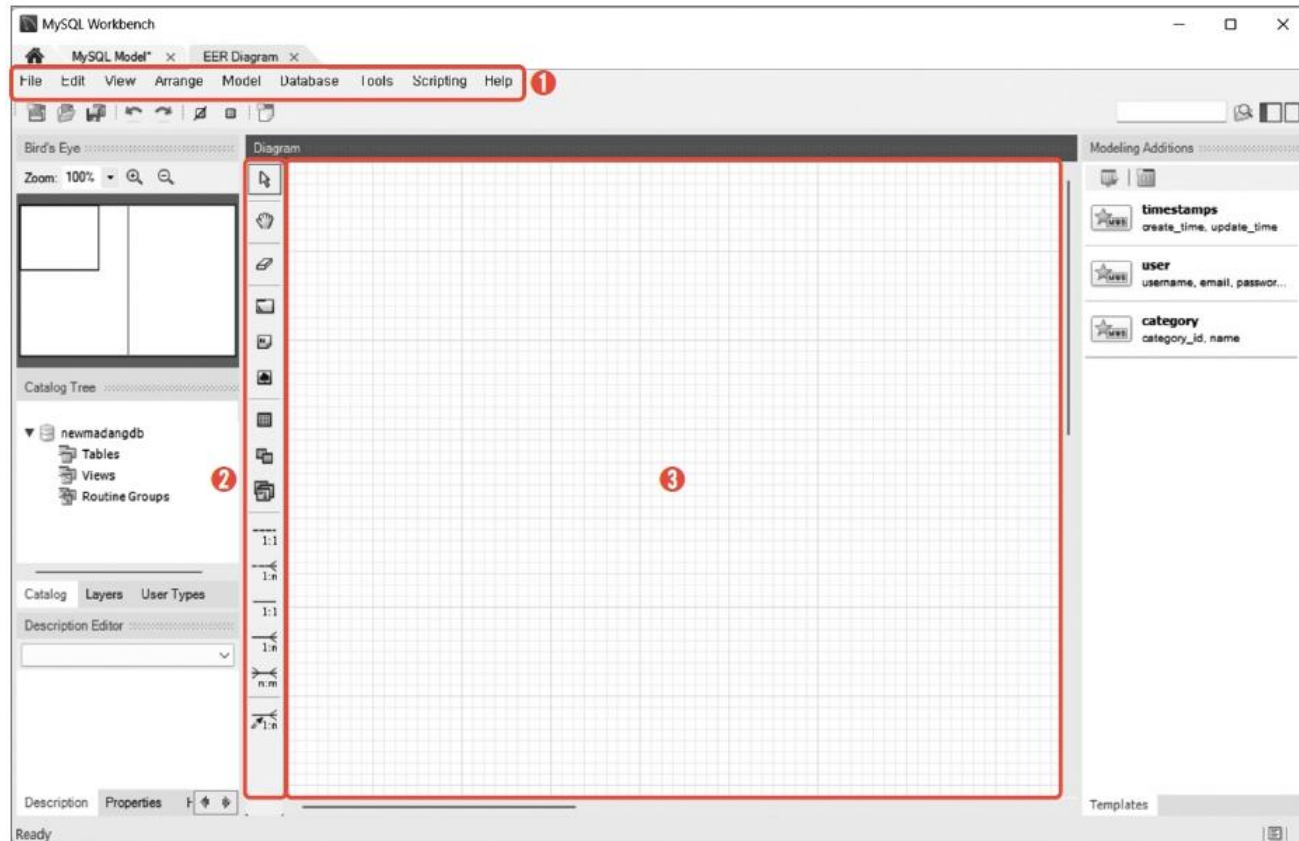


그림 6-47 Workbench의 다이어그램 작성 화면

- ① 메뉴, ② 툴바, ③ 다이어그램 작성 영역

## 2. 마당서점 ERD 만들기

- 다이어그램 작성 시 본 실습에서 사용하는 IE 표기법의 툴 박스는 다음과 같음

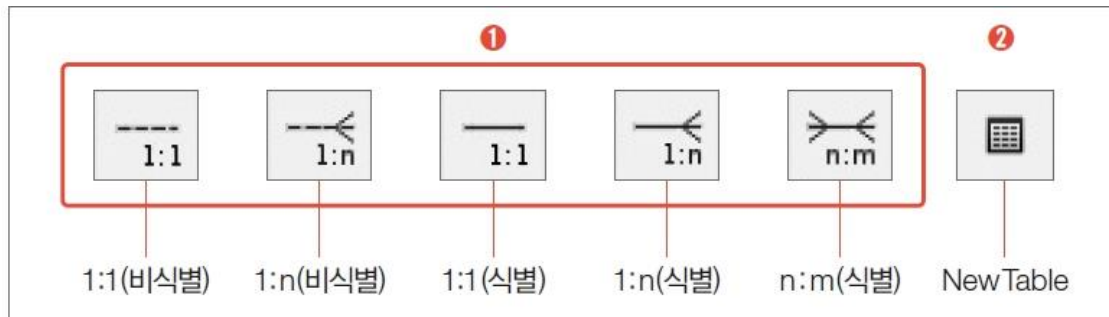


그림 6-48 주요 도구

- 1 1:1(비식별), 1:n(비식별), 1:1(식별), 1:n(식별), n:m(식별)
- 2 New Table



## 2. 마당서점 ERD 만들기

### ❖ 마당서점의 모델링

#### ① 물리적 모델의 ERD 작성

표 6-16 마당서점의 논리적 모델, 물리적 모델에서의 개체별 대응표

개체	출판사(NewPublisher)				
구분	Logical	Physical	Datatype	PK	NN
기본키(PK)	출판사이름	pubname	VARCHAR (40)	✓	✓
속성	담당자이름	stname	VARCHAR (40)		
	전화번호	officephone	VARCHAR (30)		

개체	도서(NewBook)				
구분	Logical	Physical	Datatype	PK	NN
기본키(PK)	도서번호	bookid	INT	✓	✓
속성	출판사이름(FK)	pubname(FK)	VARCHAR (40)		
	도서이름	bookname	VARCHAR (40)		
	도서단가	price	INT		✓

## 2. 마당서점 ERD 만들기

표 6-16 마당서점의 논리적 모델, 물리적 모델에서의 개체별 대응표

개체	출판사(NewPublisher)				
구분	Logical	Physical	Datatype	PK	NN
기본키(PK)	출판사이름	pubname	VARCHAR (40)	✓	✓
속성	담당자이름	stname	VARCHAR (40)		
	전화번호	officephone	VARCHAR (30)		

개체	도서(NewBook)				
구분	Logical	Physical	Datatype	PK	NN
기본키(PK)	도서번호	bookid	INT	✓	✓
속성	출판사이름(FK)	pubname(FK)	VARCHAR (40)		
	도서이름	bookname	VARCHAR (40)		
	도서단가	price	INT		✓

## 2. 마당서점 ERD 만들기

### ② 주요 테이블 그리기

Table Name: table1 Schema: newmadangdb

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type: Charset/Collation: Expression: Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☐ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☒ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

- 편집 화면에서 출판사 엔티티인 NewPublisher 테이블을 아래와 같이 작성

NewPublisher - Table x

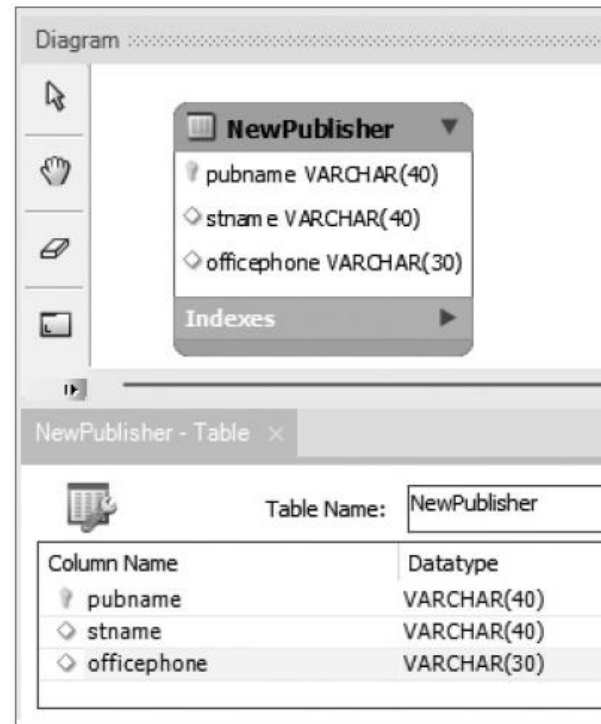
Table Name: NewPublisher

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default
pubname	VARCHAR(40)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
stname	VARCHAR(40)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
officephone	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

## 2. 마당서점 ERD 만들기

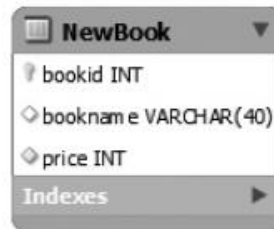
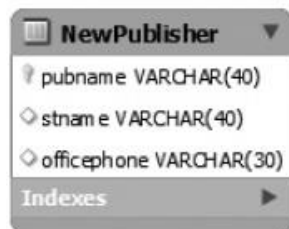
- 모든 항목을 입력하면 다음 이미지와 같이 화면상에 새로 생성된 NewPublisher 테이블을 확인할 수 있음

NewPublisher 개체 생성

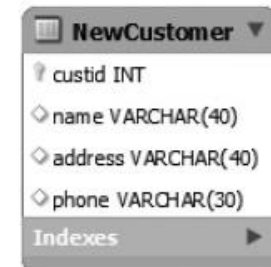


- 같은 방법으로 FK 속성을 제외하고 도서 (NewBook), 고객 (NewCustomer) 테이블을 작성

NewBook 개체 생성

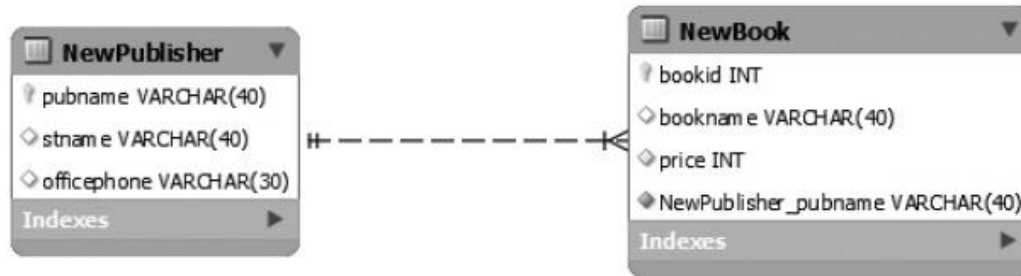


NewCustomer 개체 생성

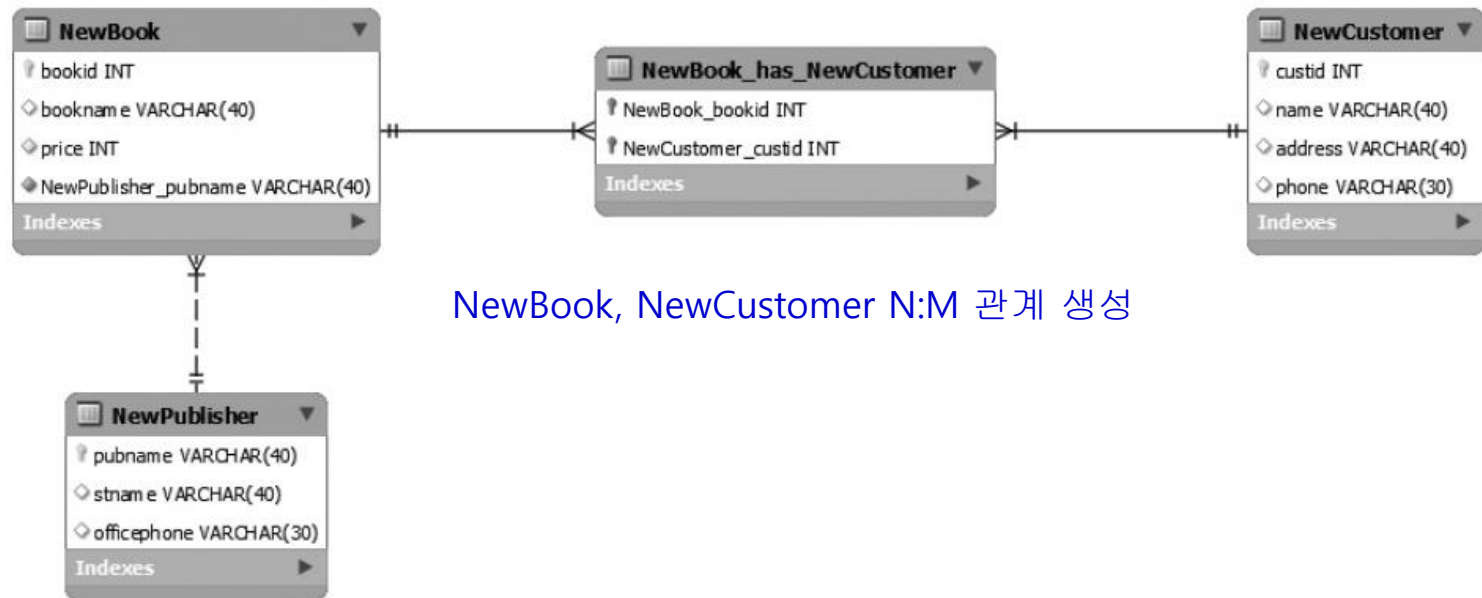


## 2. 마당서점 ERD 만들기

### ③ 관계 선 그리기 및 N:M 관계(주문) 해결

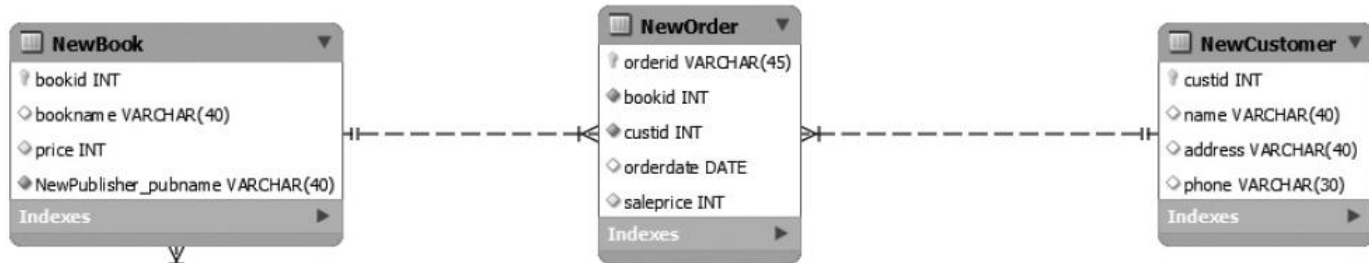


NewPublisher, NewBook 1:N 관계 생성

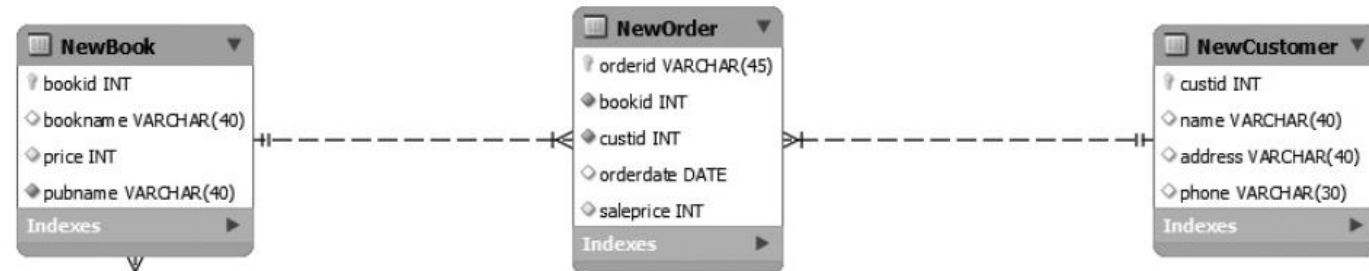


NewBook, NewCustomer N:M 관계 생성

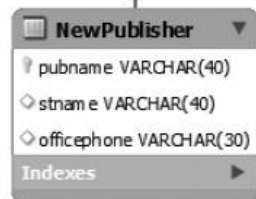
## 2. 마당서점 ERD 만들기



NewBook\_has\_NewCustomer 를 NewOrder로 이름 변경



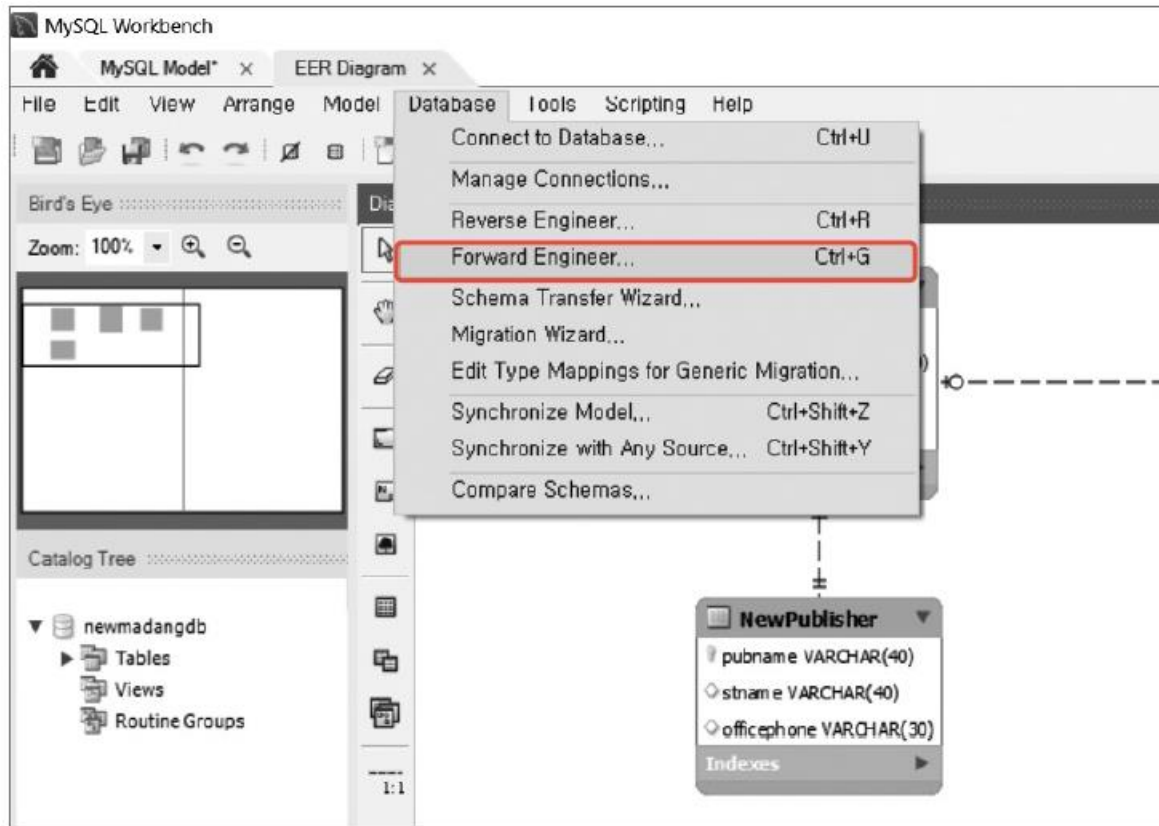
속성 이름과 도메인 정리



## 2. 마당서점 ERD 만들기

### ④ 테이블 생성 (Forward Engineer)

- [Database] - [Forward Engineer] 메뉴를 선택



## 2. 마당서점 ERD 만들기

- Stored Connection에서 Local Instance MySQL80을 선택하고 사용자는 'root'로 입력한 뒤 Next를 누르기

The screenshot shows the 'Forward Engineer to Database' dialog box with the 'Set Parameters for Connecting to a DBMS' tab selected. The 'Stored Connection' dropdown is set to 'Local Instance MySQL80'. The 'Username' field is set to 'root'. The 'Next' button is highlighted.

Forward Engineer to Database

Connection Options

- Options
- Select Objects
- Review SQL Script
- Commit Progress

Set Parameters for Connecting to a DBMS

Stored Connection: Local Instance MySQL80 (Select from saved connection settings)

Connection Method: Standard (TCP/IP) (Method to use to connect to the RDBMS)

Parameters SSL Advanced

Hostname: localhost Port: 3306 (Name or IP address of the server host - and TCP/IP port.)

Username: root (Name of the user to connect with.)

Password: Store in Vault ... Clear (The user's password. Will be requested later if it's not set.)

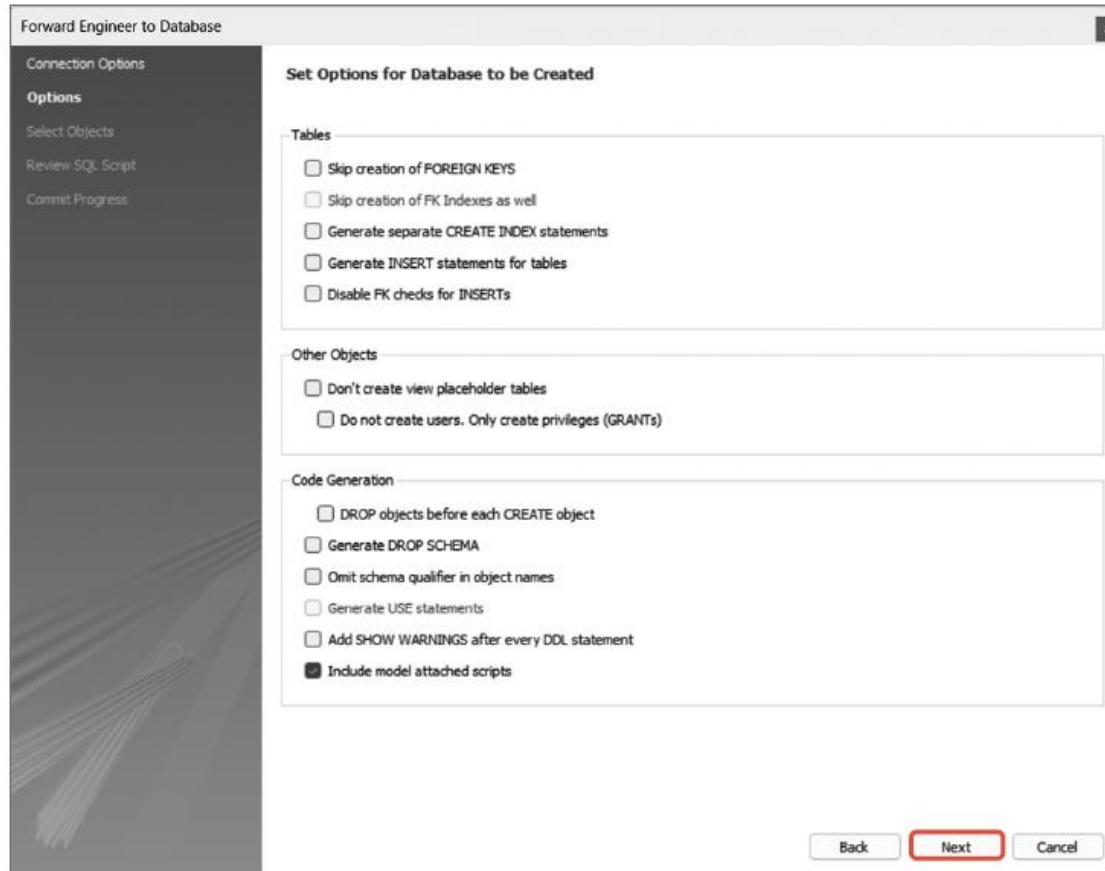
Default Schema: (The schema to use as default schema. Leave blank to select it later.)

Back Next Cancel



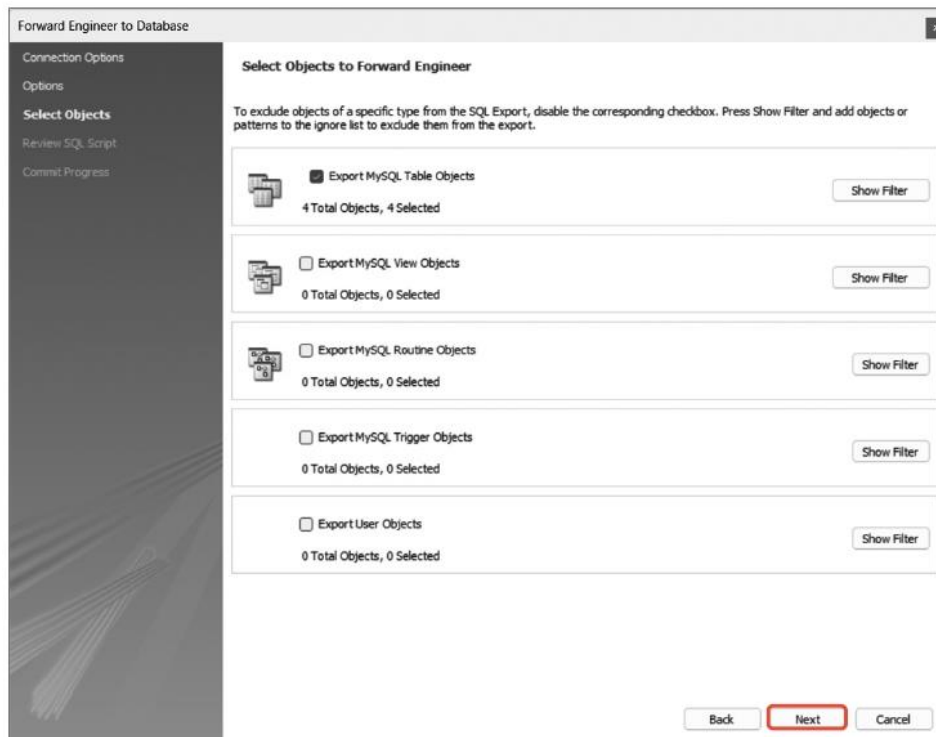
## 2. 마당서점 ERD 만들기

- 아래 화면으로 전환되면 Next를 누르기



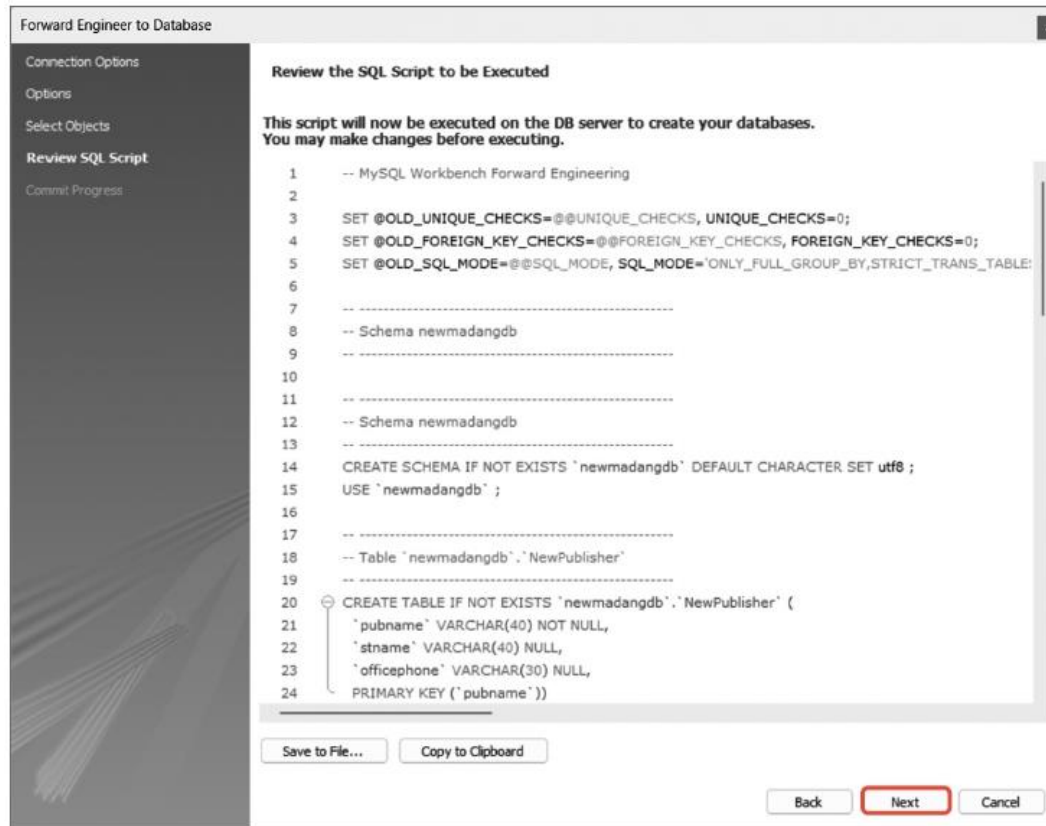
## 2. 마당서점 ERD 만들기

- Connect to MySQL Server 창이 뜨면 Password란에 'root'를 입력하고 Save password in vault를 체크한 후 OK를 누르기
- 접속을 생성할 때 root 비밀번호를 Save해 두었다면 이 창이 나타나지 않을 수 있음
- Next를 눌러 Review SQL Script로 넘어감



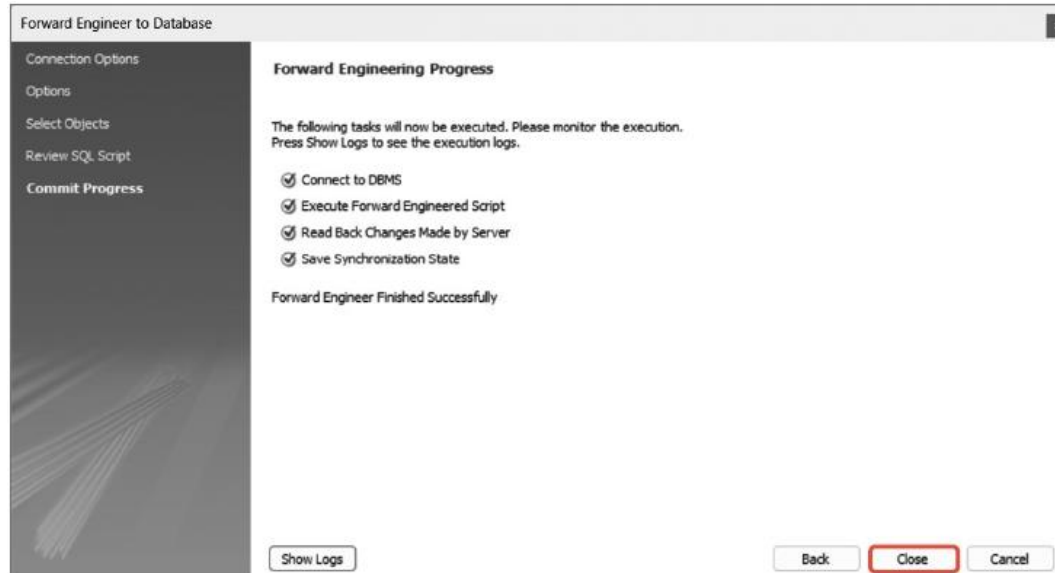
## 2. 마당서점 ERD 만들기

- Next를 눌러 Commit Progress 페이지로 이동함



## 2. 마당서점 ERD 만들기

- Forward Engineer Finished Successfully 메시지를 확인한 후 Close를 선택하여 작업을 완료함



## 2. 마당서점 ERD 만들기

- 기존 madangdb와 별도로 생성된 newmadangdb와 NewBook, NewCustomer, NewOrders, NewPublisher 테이블이 생성된 것을 **확인**할 수 있음

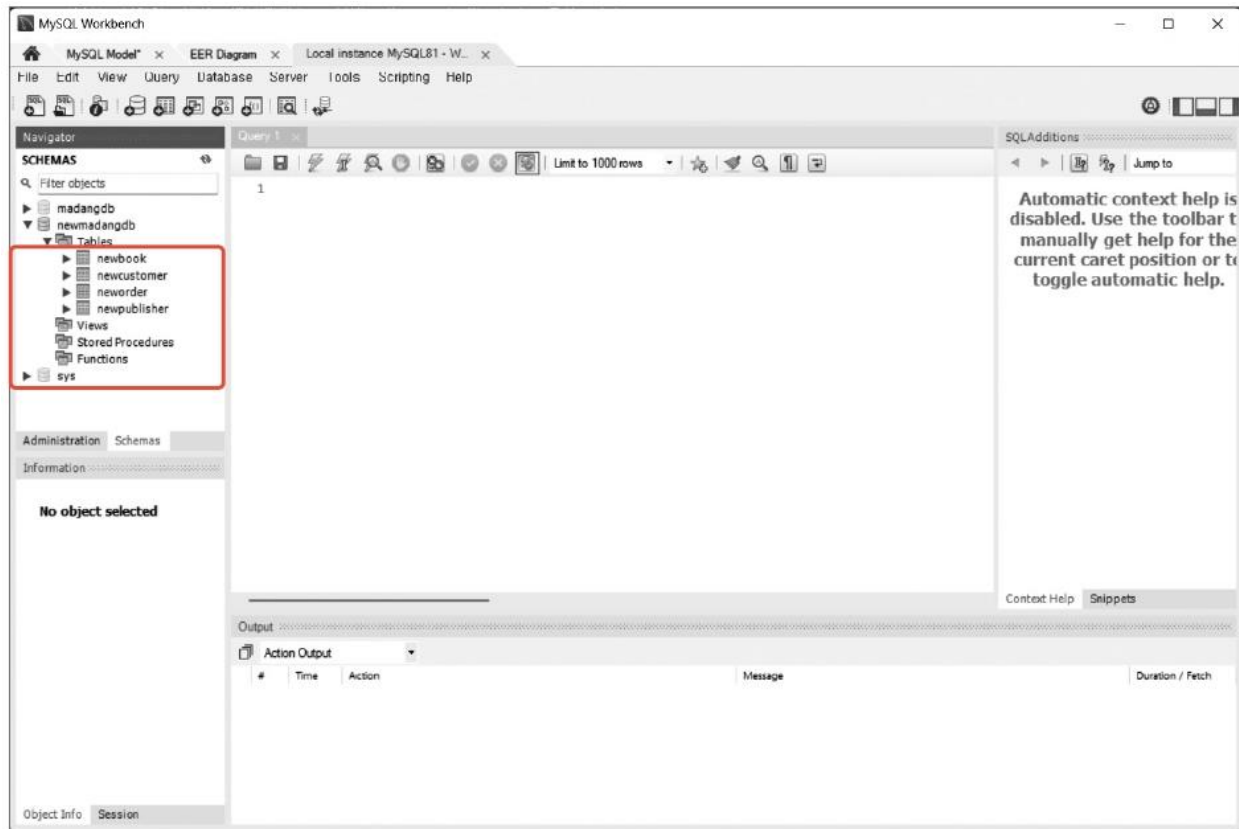


그림 6-49 newmadangdb 데이터베이스와 생성된 테이블

## 05 모델링 연습(마당대학 데이터베이스)

1. 마당대학의 요구사항
2. 마당대학의 ER 다이어그램



# 1. 마당대학의 요구사항

- 마당대학의 요구사항을 분석한 결과 다음과 같은 11개 항목으로 정리됨
- ①~④는 개체에 대한 내용이고, ⑤~⑪은 개체 간의 관계에 대한 내용

- 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)에는 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 학과별로 참여백분율(pct\_time)이 기록된다.
- ⑦ 대학원생에게는 학위 과정을 밟을 전공학과(major)가 하나씩 있다.
- ⑧ 대학원생에게는 어떤 과목을 들으면 좋을지 조언(advisor)해 주는 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.

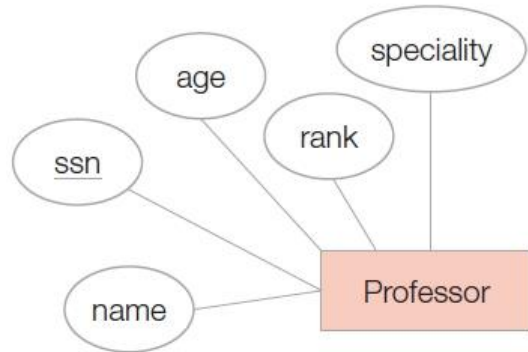


그림 6-50 교수(Professor) 개체

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

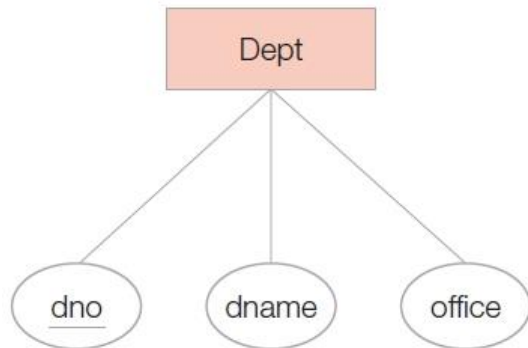


그림 6-51 학과(Department) 개체



## 2. 마당대학의 ER 다이어그램

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

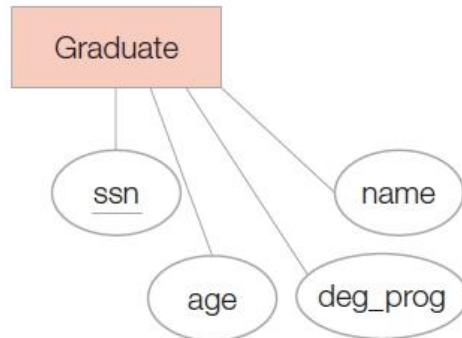


그림 6-52 대학원생(Graduate) 개체

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

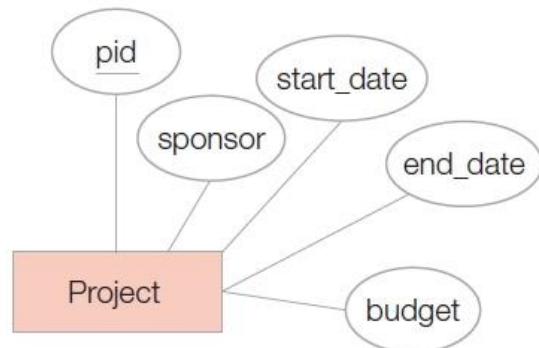


그림 6-53 과제(Project) 개체

## 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.

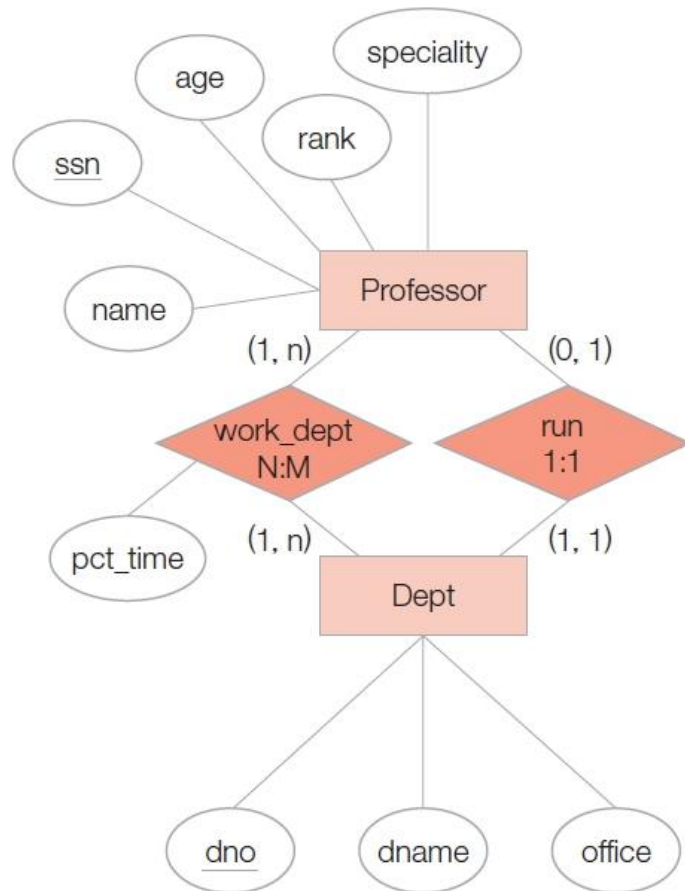


그림 6-54 운영(run) 관계와 근무(work-dept) 관계

## 2. 마당대학의 ER 다이어그램

⑦ 대학원생에게는 학위 과정을 밟을 전공학과(major)가 하나씩 있다.

⑧ 대학원생에게는 어떤 과목을 들으면 좋을지 조언(advisor)해 주는 선임 대학원생(학생조언자라고 한다)이 있다.

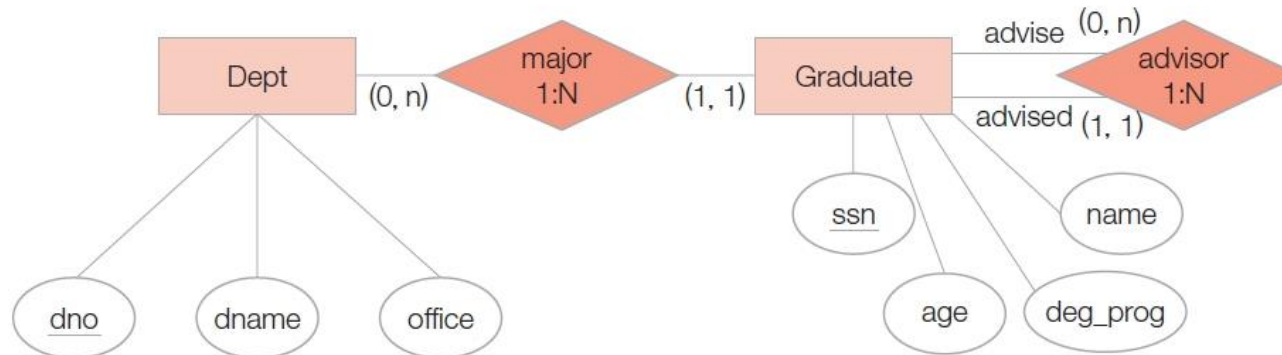


그림 6-55 전공학과(major) 관계와 조언(advisor) 관계

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.

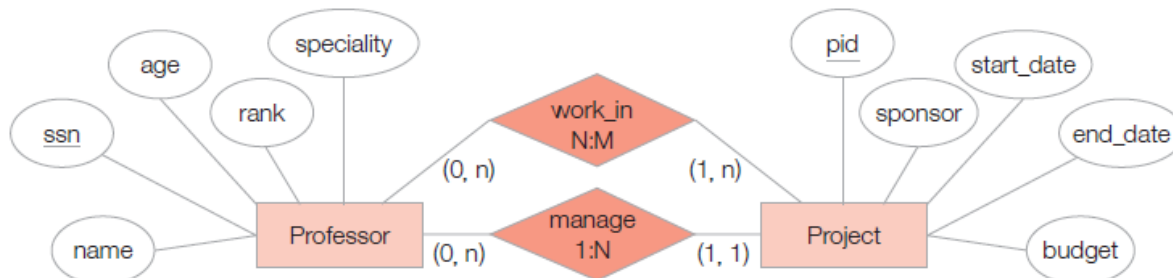


그림 6-56 관리(manage) 관계와 수행(work-in) 관계

## 2. 마당대학의 ER 다이어그램

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

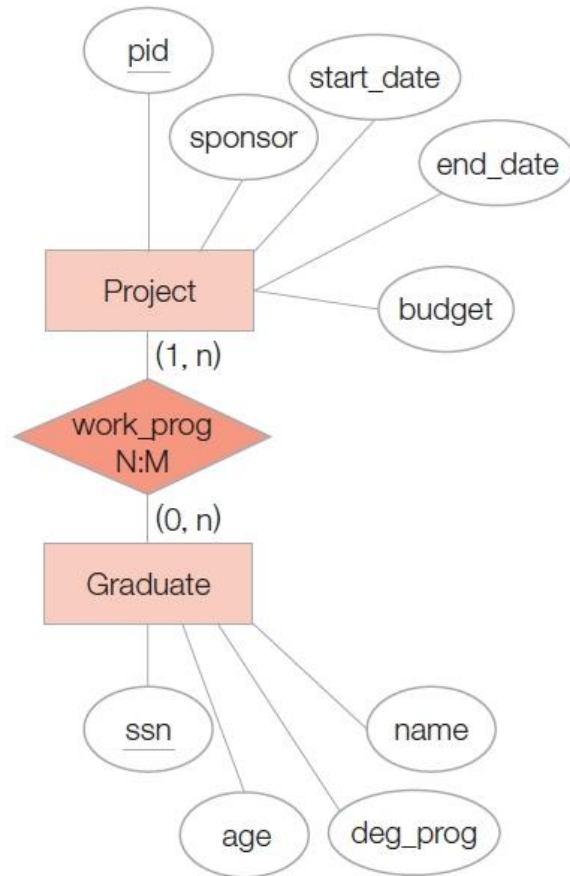


그림 6-57 수행(work-prog) 관계

## 2. 마당대학의 ER 다이어그램

- 요구사항에 맞게 생성된 최종 ER 다이어그램

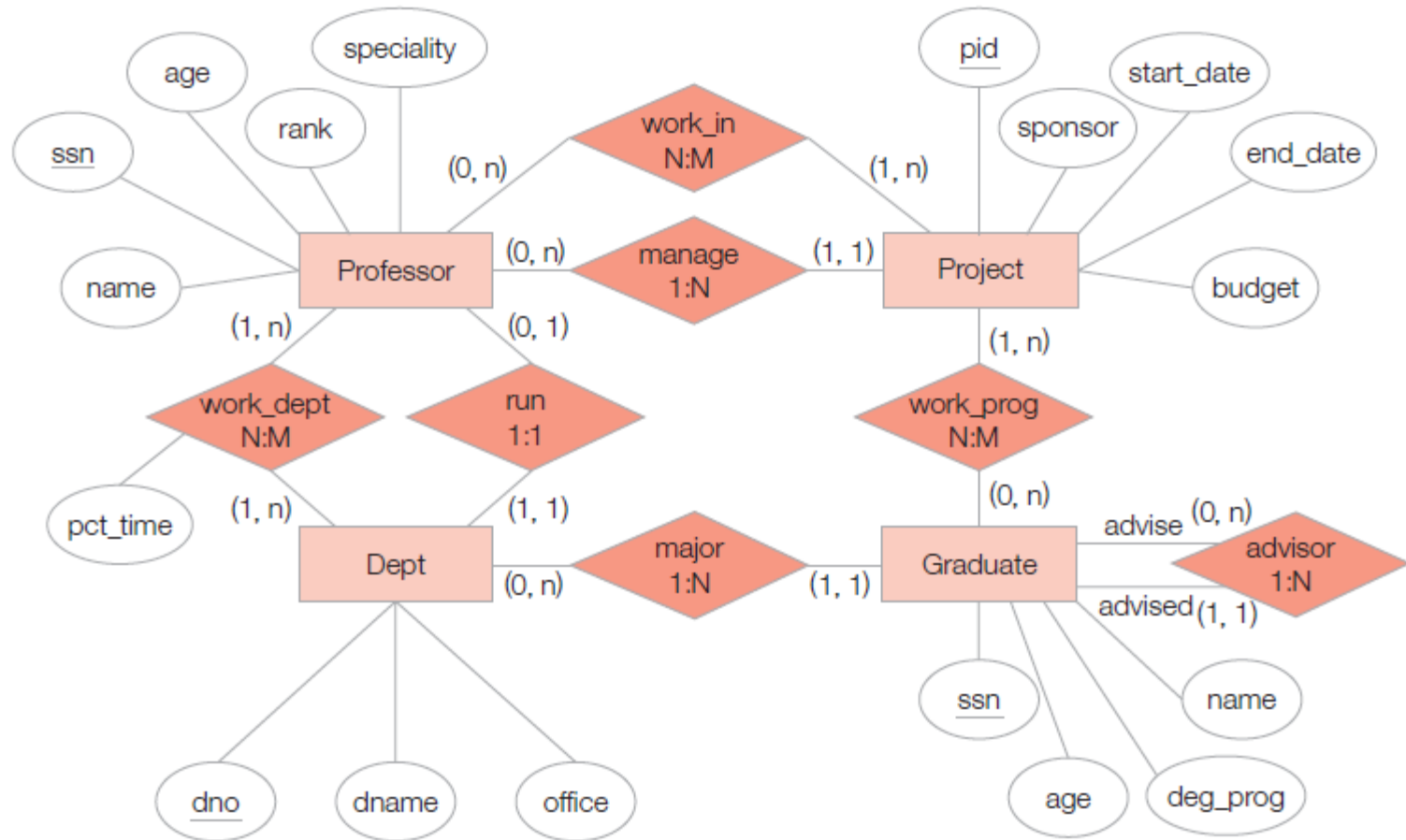


그림 6-58 마당대학의 ER 다이어그램

## 2. 마당대학의 ER 다이어그램

- 마당대학의 ER 다이어그램을 관계 데이터 모델로 변환

### [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, ⑧번 요구사항 advisor(1:N) 관계를 표현하기 위해 추가되었다.

Project(pid, sponsor, start\_date, end\_date, budget, managessn)

managessn : Professor(ssn)을 참조하는 외래키로, ⑨번 요구사항 manage(1:N) 관계를 표현하기 위해 추가되었다.

### [관계]

N:M 관계일 경우 교차 테이블을 생성한다.

work-dept(professorssn, dno, pct\_time) /\* 교수의 학과 참여 \*/

⑥번 요구사항의 work-dept(N:M) 관계를 표현하기 위한 교차 테이블이다. professorssn은 Professor(ssn)을 참조하는 외래키이고, dno는 Dept(dno)를 참조하는 외래키다. pct\_time은 관계에 포함된 속성이다.

work-in(professorssn, pid) /\* 교수의 과제 수행 \*/

⑩번 요구사항의 work-in(N:M) 관계를 표현하기 위한 교차 테이블이다. professorssn은 Professor(ssn)을 참조하는 외래키이고, pid는 Project(pid)를 참조하는 외래키다.

work-prog(graduatessn, pid) /\* 대학원생의 과제 수행 \*/

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

## 2. 마당대학의 ER 다이어그램

- [그림 6-59]의 마당대학 데이터베이스 스키마를 Workbench ERD 로 작성

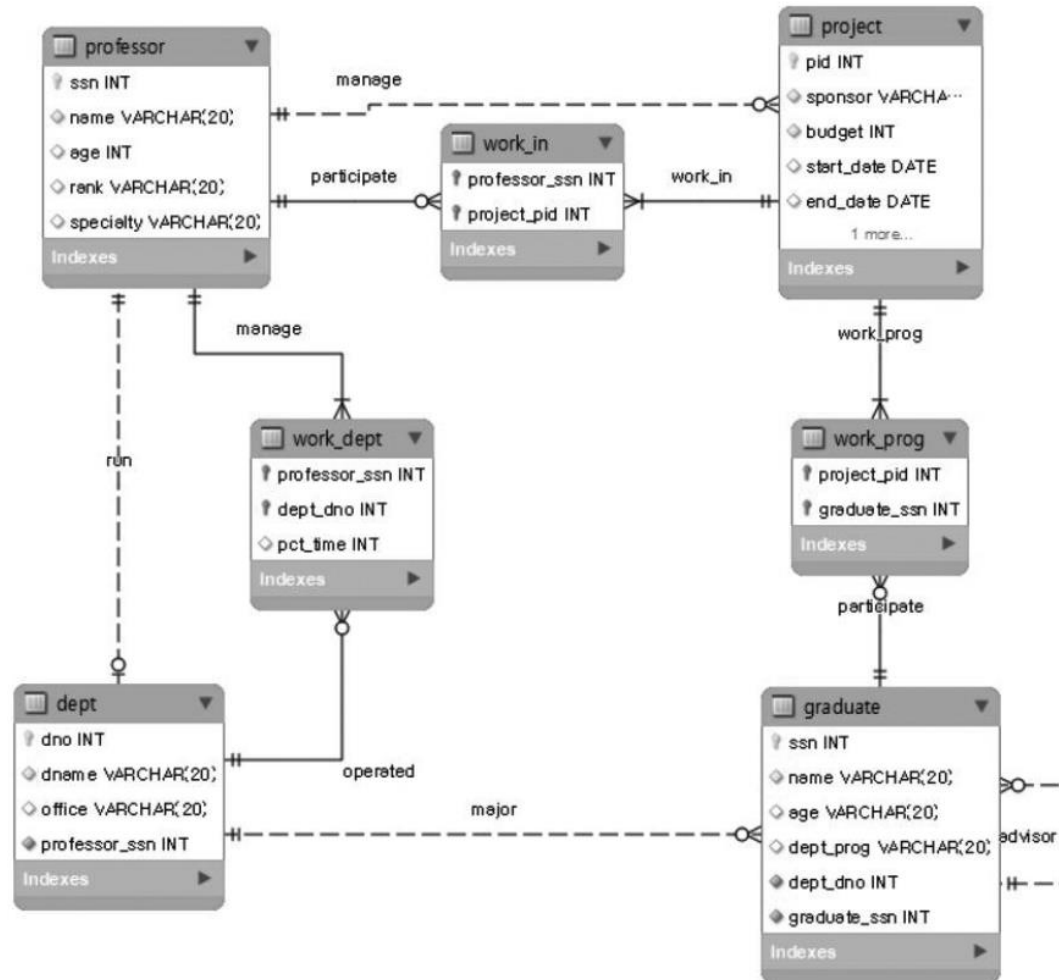
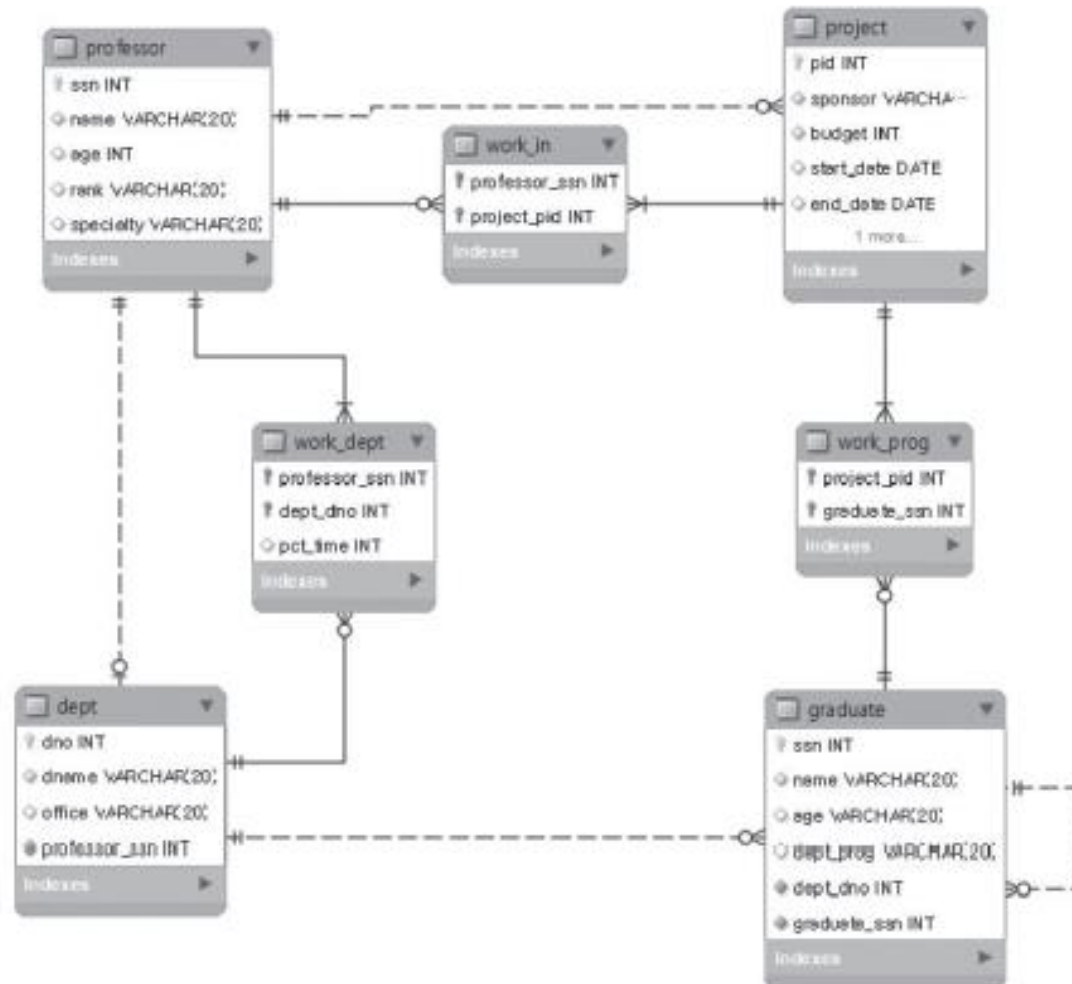


그림 6-60 마당대학 데이터베이스의 ERD 다이어그램(물리적 모델)



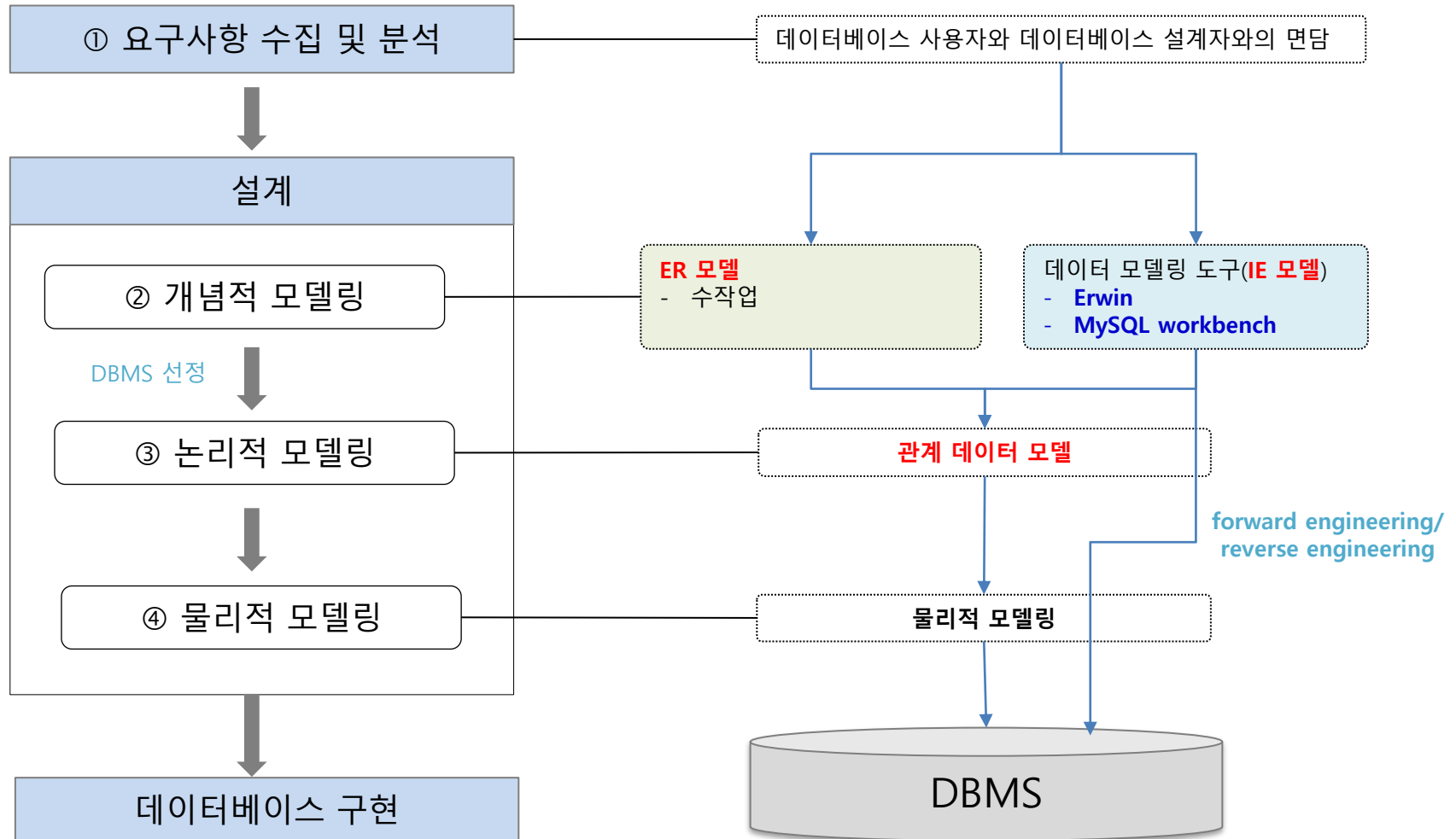
## 연습문제 (Q6.4) erwin 연습 - 마당 대학 모델링

16 5절에서 완성한 마당대학 데이터베이스를 모델링하시오.





# 데이터모델링 - 요약



# 요약

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