

2. 데이터베이스 활용과 클라우드 서버 환경 준비

2 . SQL 심화 / 데이터 모델링 / 데이터베이스연동

데이터 모델링



목차

01 데이터 모델링의 개념

02 ER 모델

03 ER 모델을 관계 데이터 모델로 사상

04 MySQL Workbeanch 활용 데이터 모델링 실습

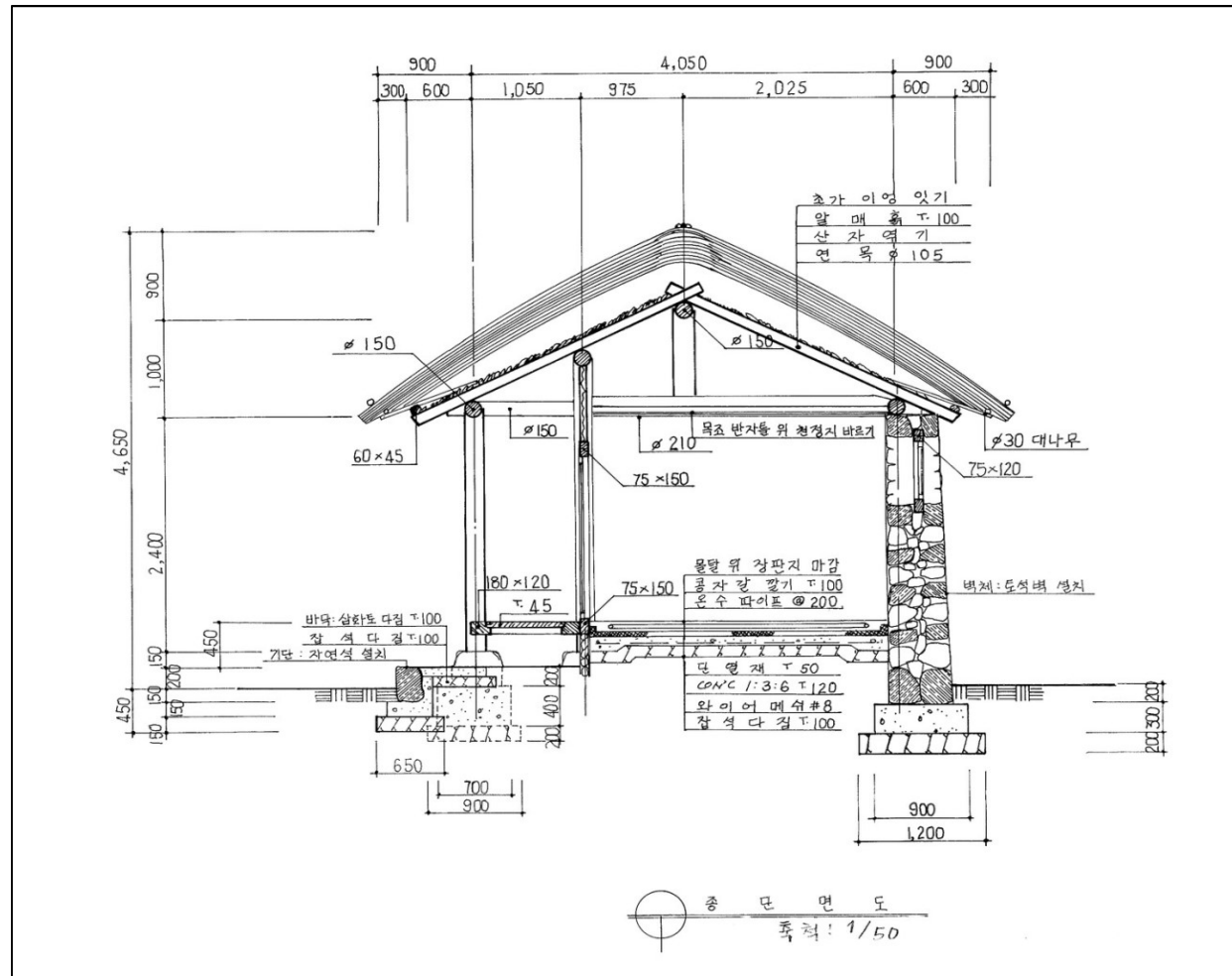
05 모델링 Workshop (대학 데이터베이스)

06 쇼핑몰 모델링 Workshop

01. 데이터 모델링의 개념

1. 데이터베이스 생명주기
2. 데이터 모델링 과정

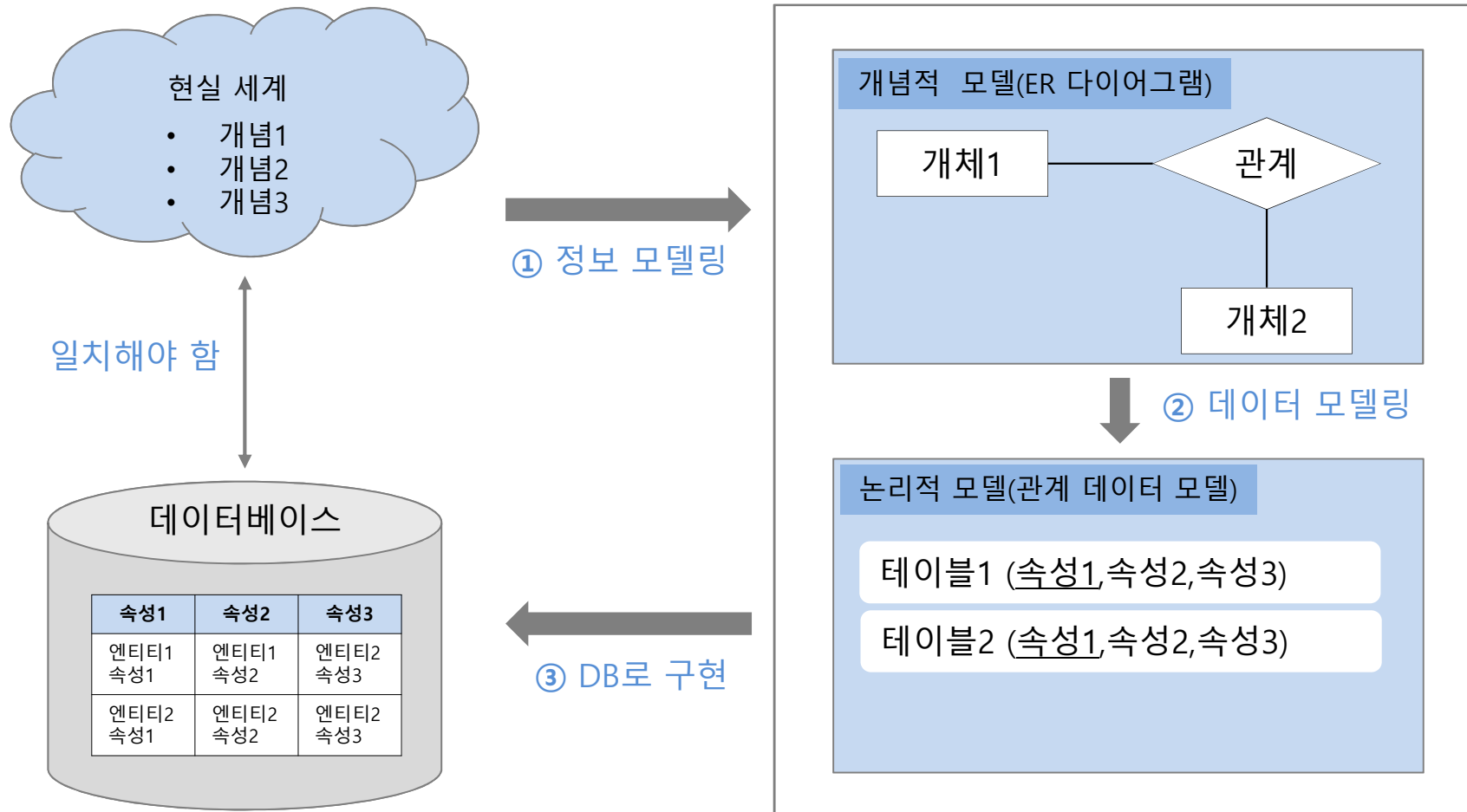
데이터 모델링의 개념



건물 설계
(소프트웨어 설계)

지반 설계
(데이터베이스 설계)

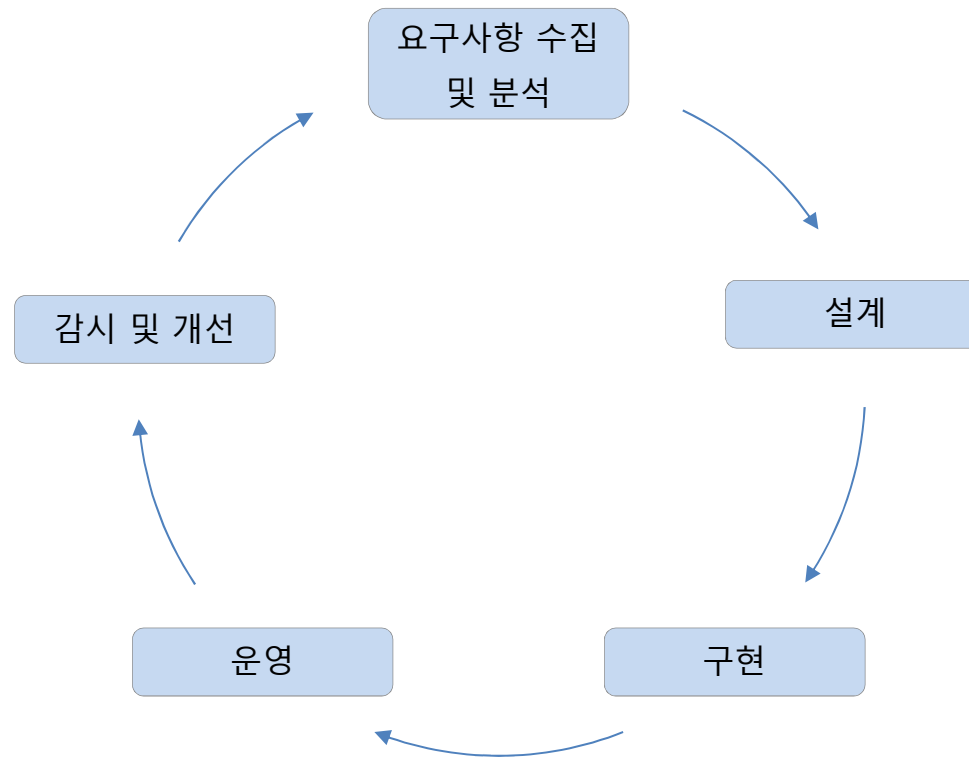
데이터 모델링의 개념



1. 데이터베이스 생명주기

❖ 데이터베이스 생명주기(database life cycle)

데이터베이스의 생성과 운영에 관련된 특징



1. 데이터베이스 생명주기

① 요구사항 수집 및 분석

사용자들의 요구사항을 듣고 분석하여 데이터베이스 구축의 범위를 정함

② 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

③ 구현

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

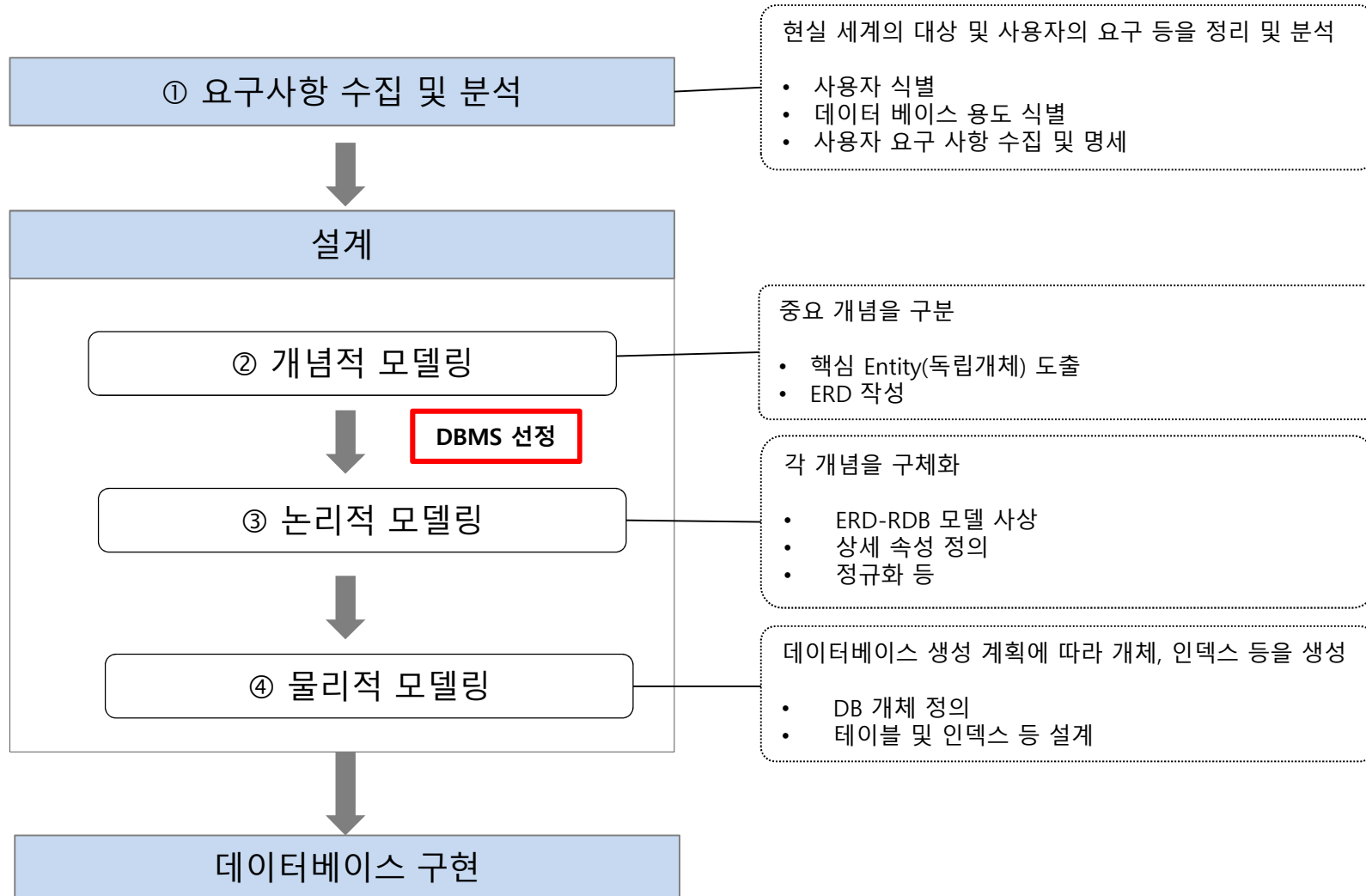
④ 운영

구현된 데이터베이스를 기반으로 소프트웨어를 구축하여 서비스를 제공함

⑤ 감시 및 개선

데이터베이스 운영에 따른 시스템의 문제를 관찰하고 데이터베이스 자체의 문제점을 파악하여 개선함

2. 데이터 모델링 과정



2. 데이터 모델링 과정

❖ 요구사항 수집 및 분석

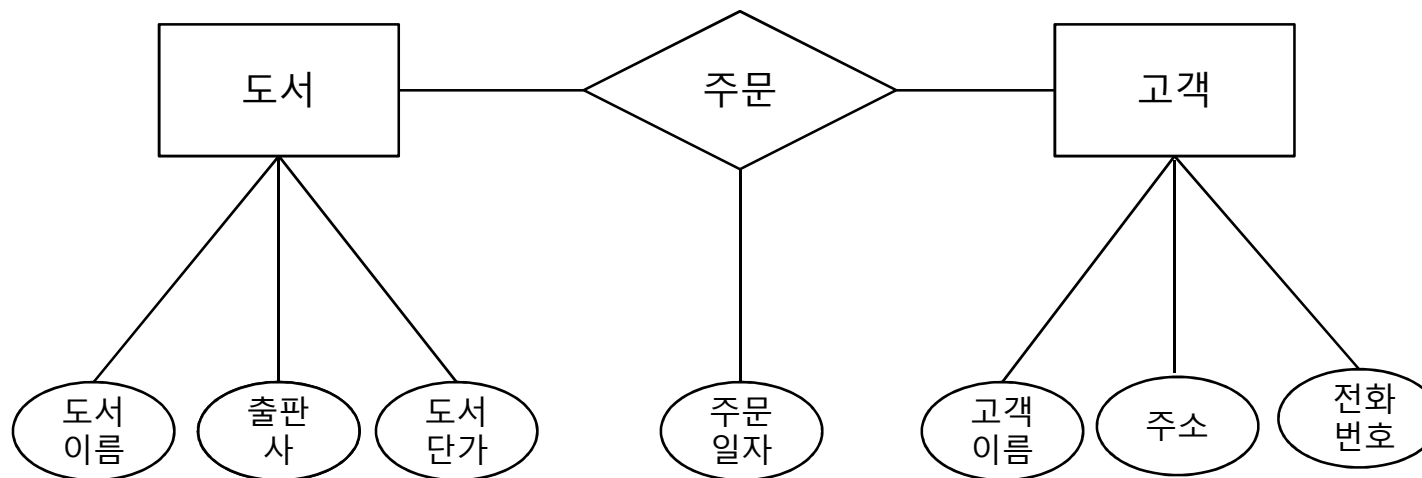
■ 요구사항 수집 방법

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

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



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)

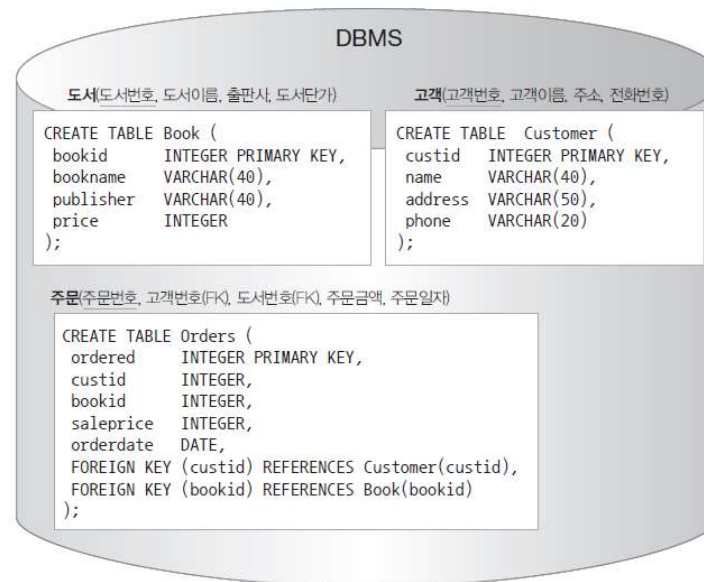
■ 논리적 모델링 과정

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

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



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

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

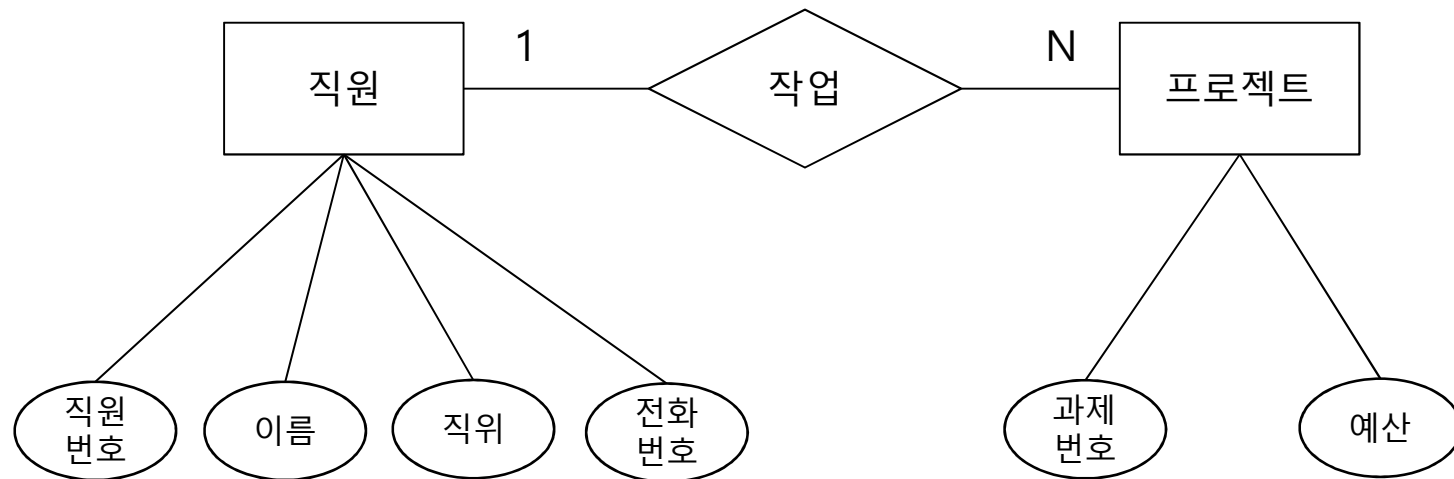
ER 모델

❖ ER 다이어그램

ER 모델은 개체와 개체 간의 관계를 표준화된 그림으로 나타냄

개체-관계

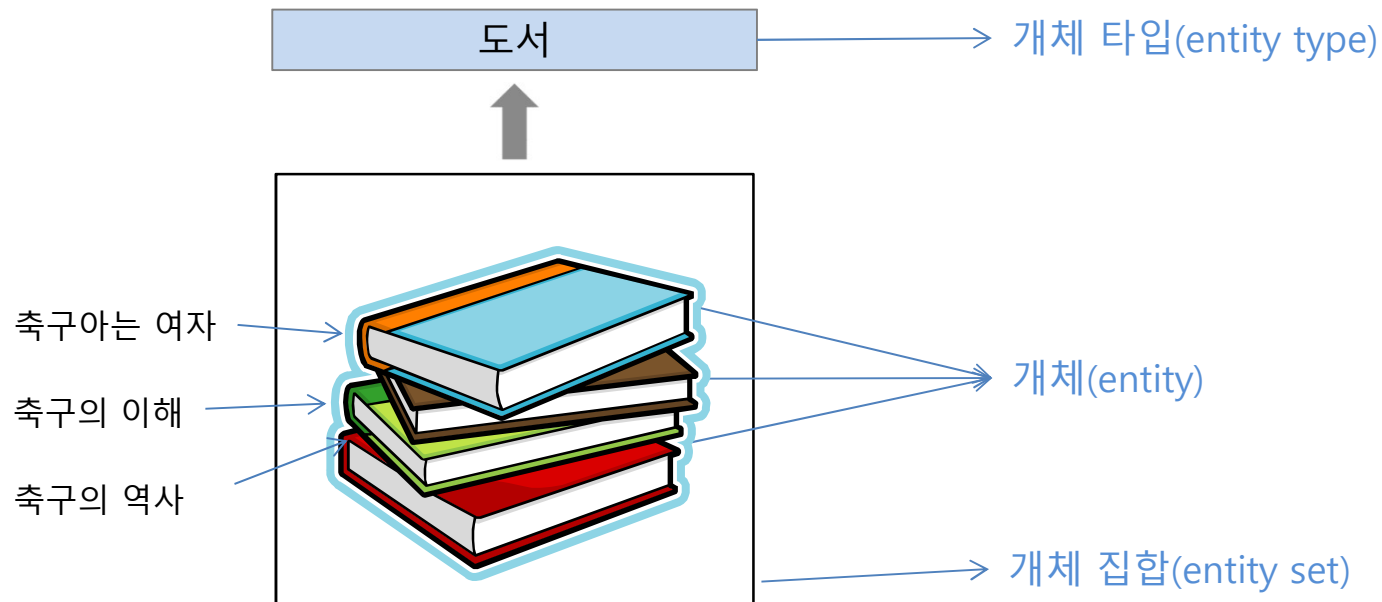
속성



1. 개체와 개체 타입

❖ 개체(entity)란?


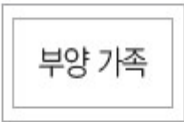
- 사람, 사물, 장소, 개념, 사건과 같이 유무형의 정보를 가지고 있는 독립적인 실체
- 데이터베이스에서 주로 다루는 개체는 낱개로 구성된 것, 낱개가 각각 데이터 값을 가지는 것, 데이터 값이 변하는 것 등이 있음.
- 비슷한 속성의 개체 타입(entity type)을 구성하며, 개체 집합(entity set)으로 묶임.



1. 개체와 개체 타입

❖ 개체 타입의 ER 다이어그램 표현

- ER 다이어그램상에서 개체 타입은 직사각형으로 나타냄

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

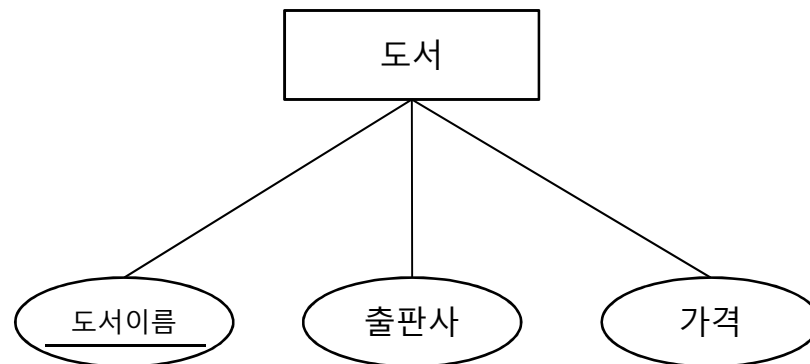
■ 속성(attribute) : 개체가 가진 성질

개체 타입	속성
도서	도서이름, 출판사, 도서단가

표 6-2 개체 타입과 속성

■ 속성의 ER 다이어그램 표현

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



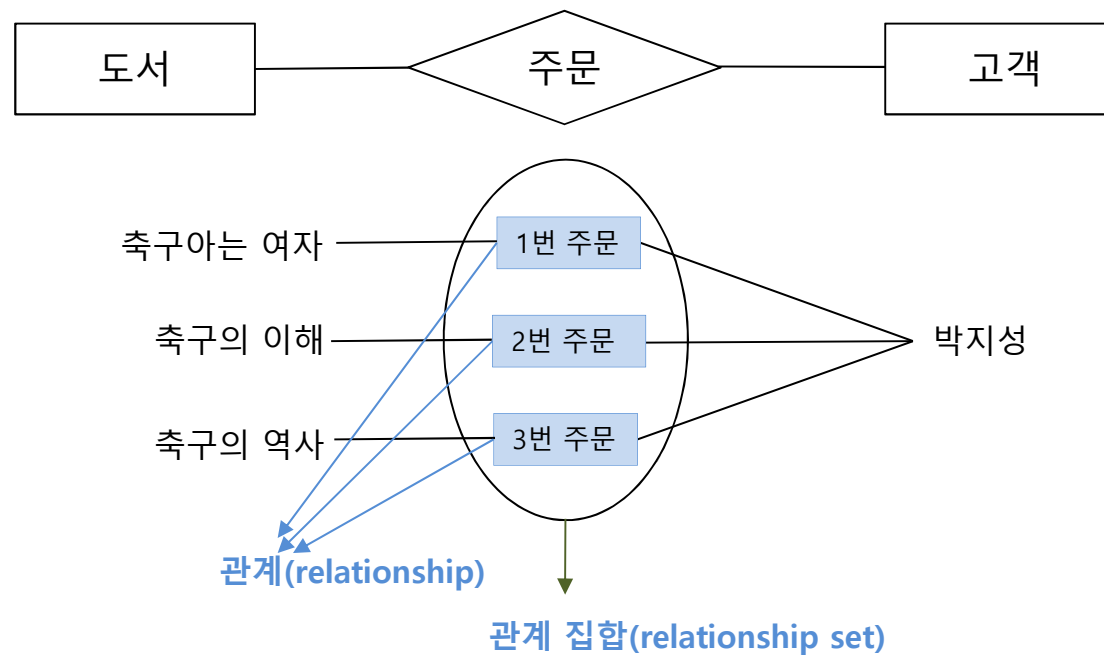
2. 속성

❖ 속성의 유형

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

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

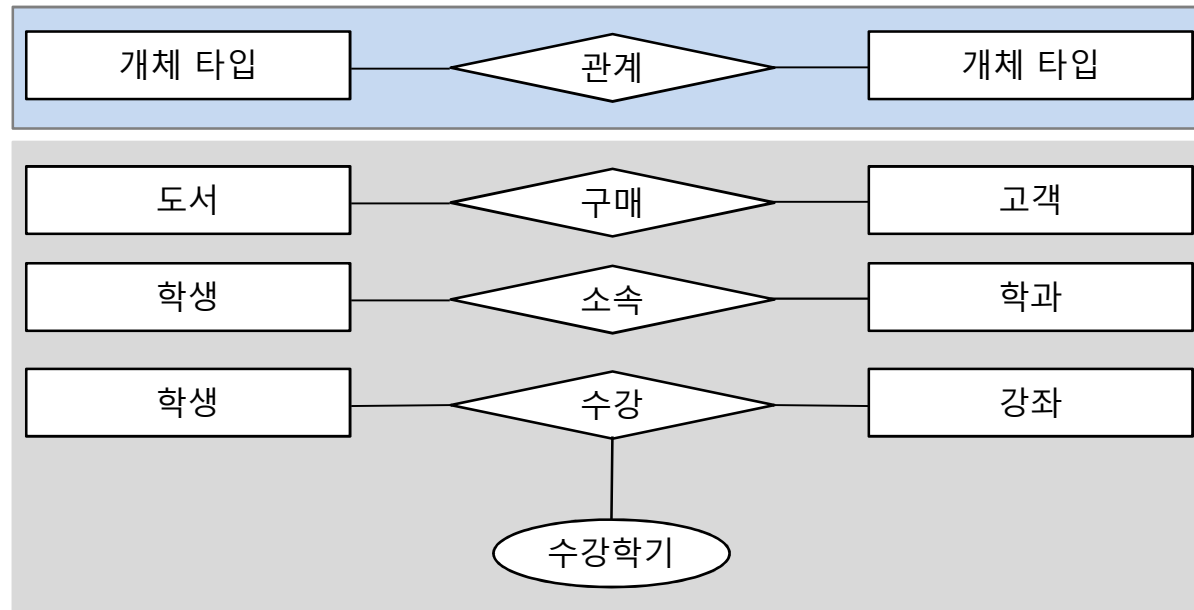


3. 관계와 관계 타입

❖ 관계 타입의 ER 다이어그램 표현

표 6-4 관계 타입의 ER 다이어그램 표현

기호	의미
	관계 타입



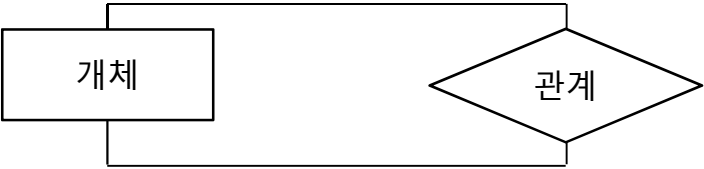
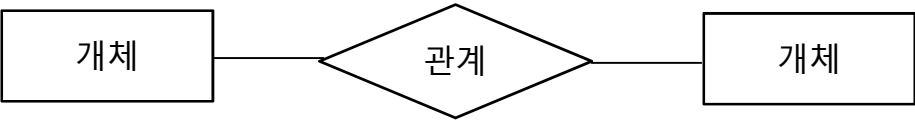
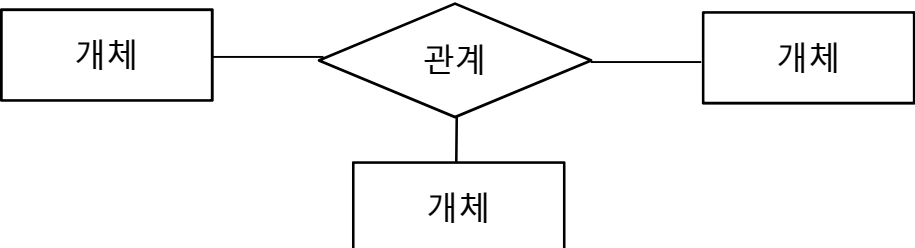
3. 관계와 관계 타입

❖ 관계 타입의 유형

■ 차수에 따른 유형

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

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

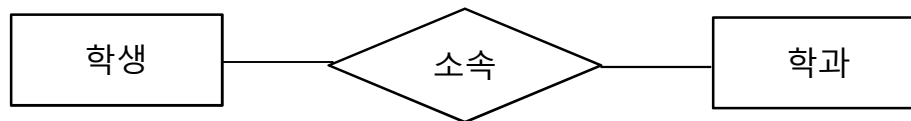
❖ 관계 타입의 유형

- 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

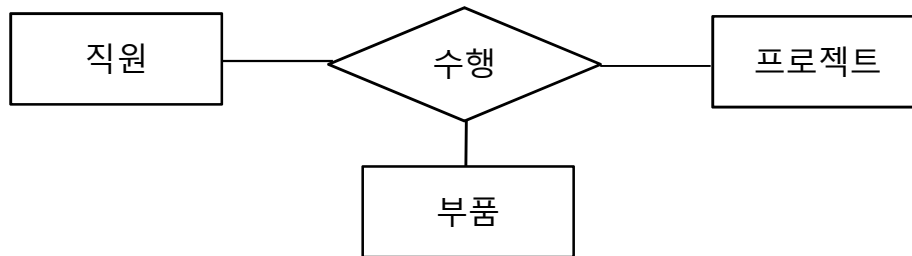


2진 관계의 예

3. 관계와 관계 타입

❖ 관계 타입의 유형

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

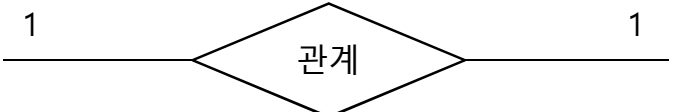
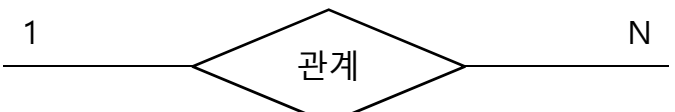
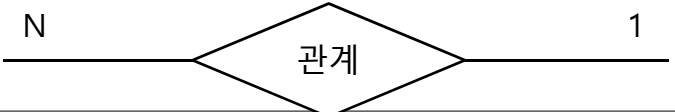
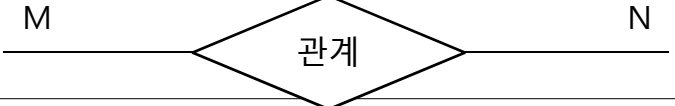


3. 관계와 관계 타입

❖ 관계 타입의 유형

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

관계 대응수에 따른 관계 타입의 유형

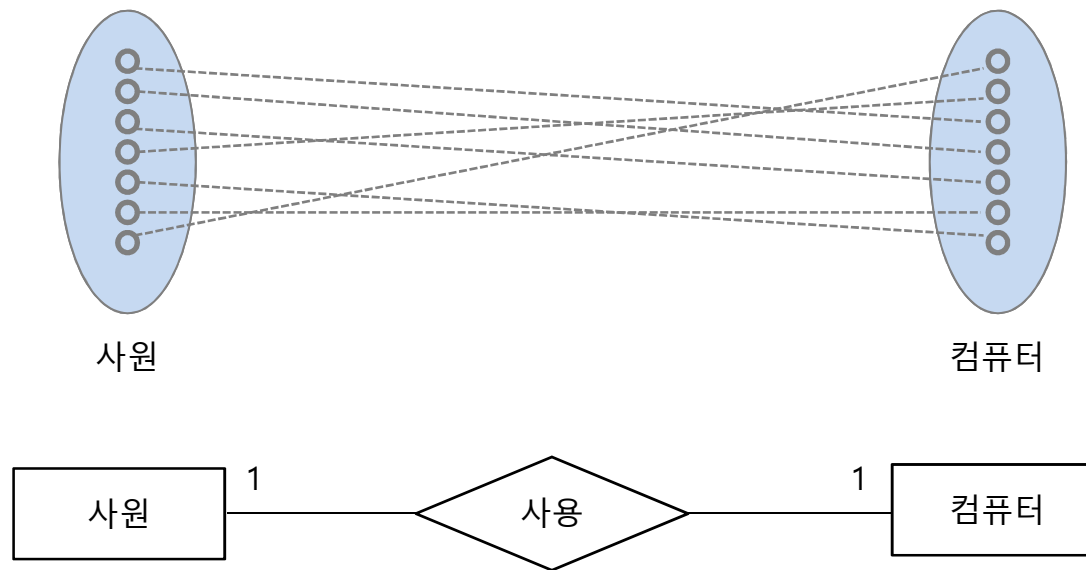
기호	의미	설명
	일대일 관계	하나의 개체가 하나의 개체에 대응
	일대다 관계	하나의 개체가 여러 개체에 대응
	다대일 관계	여러 개체가 하나의 개체에 대응
	다대다 관계	여러 개체가 여러 개체에 대응

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



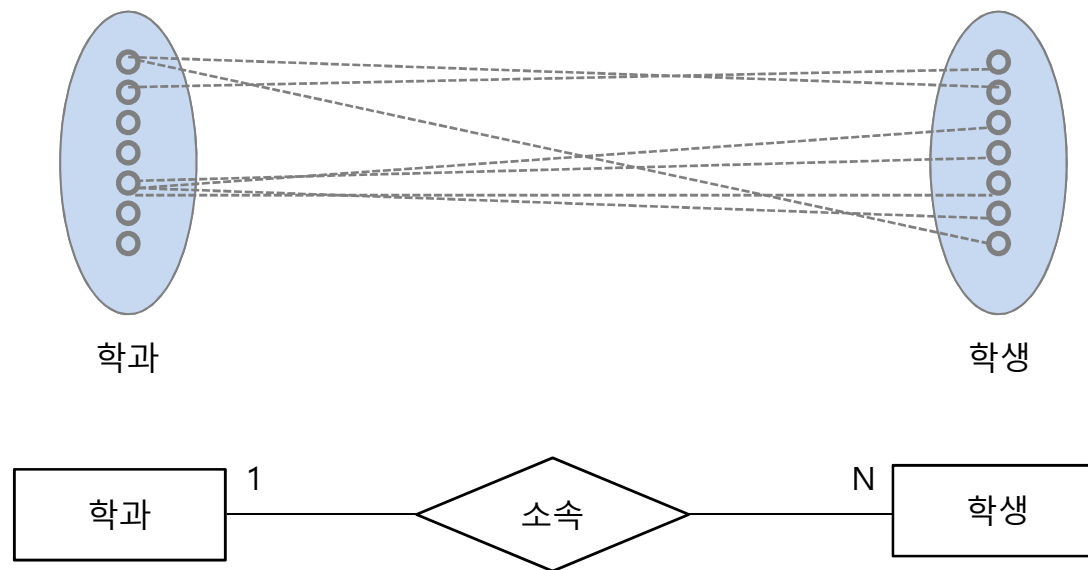
일대일 관계의 예

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



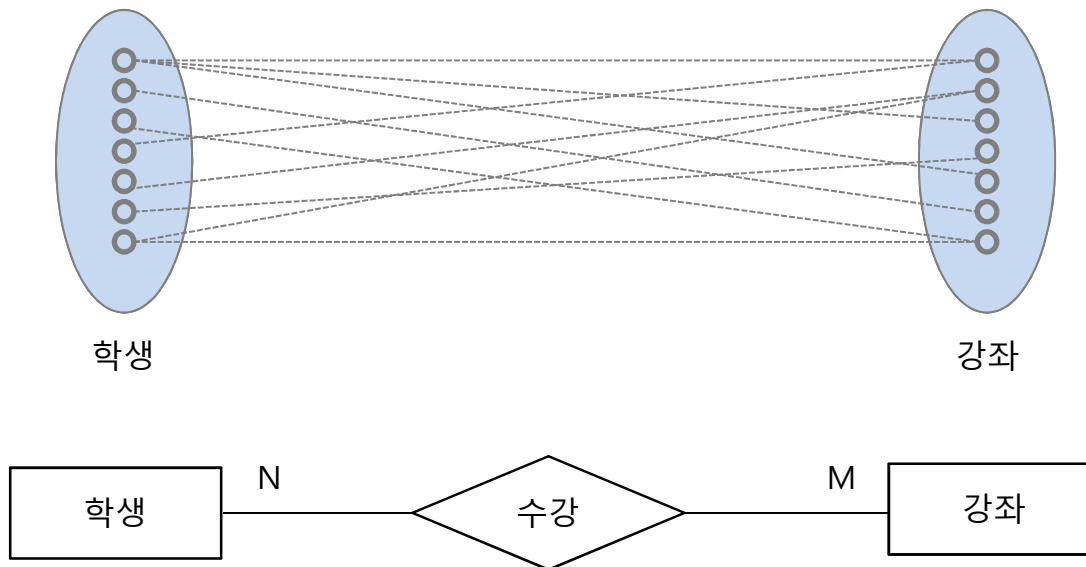
일대다(1:N), 다대일(N:1) 관계의 예

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

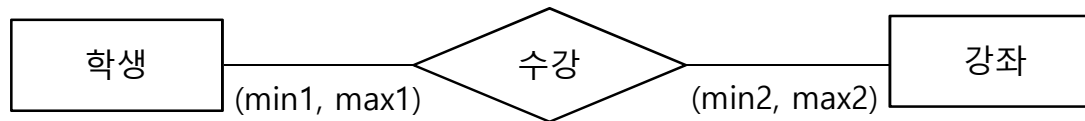


다대다(N:M) 관계의 예

3. 관계와 관계 타입

❖ 관계 대응수의 최솟값과 최댓값

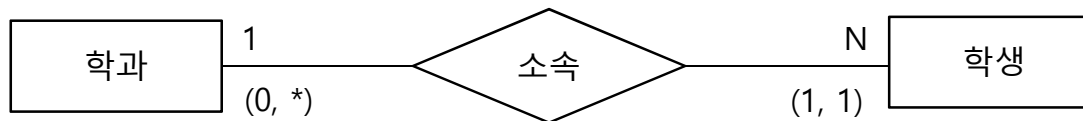
- 관계 대응수 1:1, 1:N, M:N에서 1, N, M은 각 개체가 관계에 참여하는 최댓값을 의미함
- 관계에 참여하는 개체의 최솟값을 표시하지 않는다는 단점을 보완하기 위해 다이어그램에서는 대응수 외에 최솟값과 최댓값을 관계실선 위에 (최솟값, 최댓값)으로 표기함



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, *)




(최솟값, 최댓값) 표기의 예

3. 관계와 관계 타입

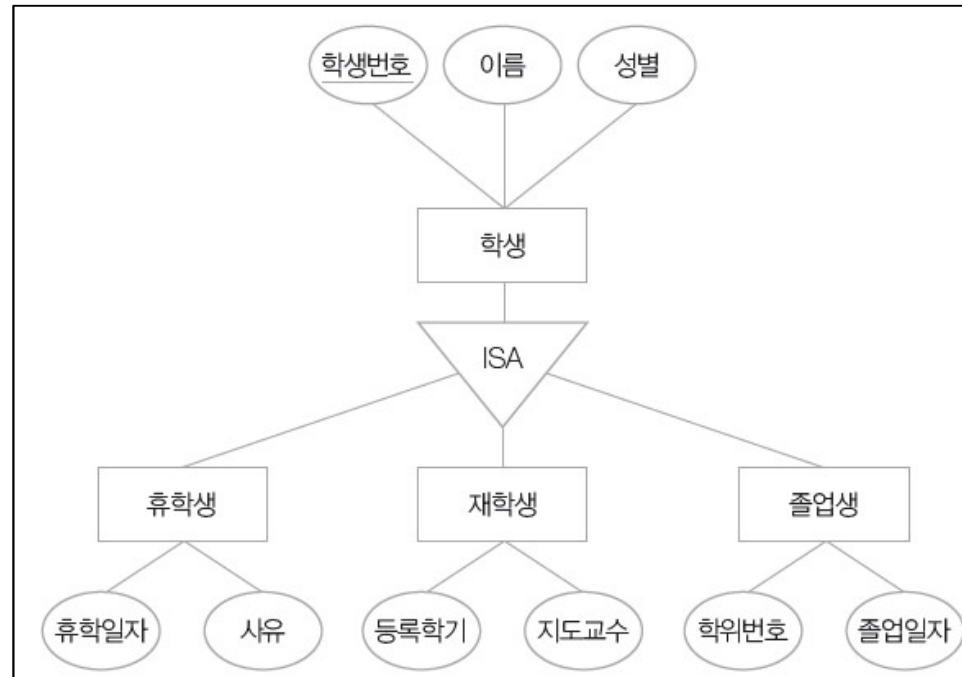
❖ ISA 관계

- 상위 개체 타입의 특성에 따라 하위 개체 타입이 결정되는 형태

ISA 관계 (ISA => is-a)

기호	의미
	개체 간 상하관계

ISA 관계의 예

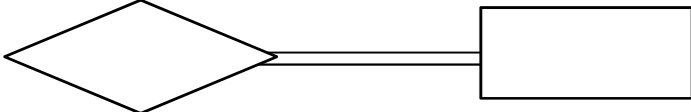
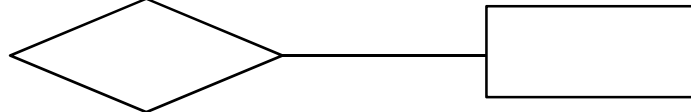


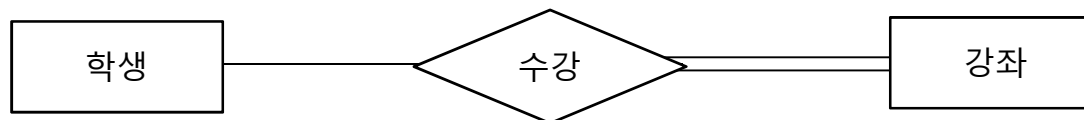
3. 관계와 관계 타입

❖ 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

관계의 참여 제약 조건

기호	의미
	전체 참여
	부분 참여

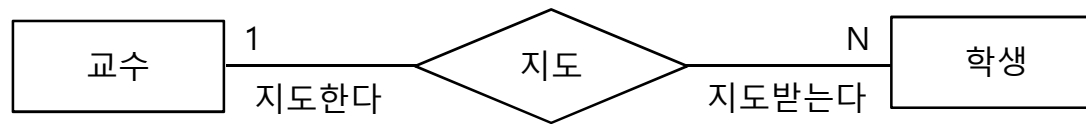


부분 참여와 전체 참여의 예

3. 관계와 관계 타입

❖ 역할

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

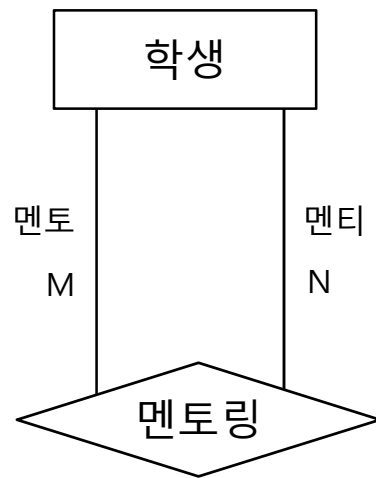


역할의 예

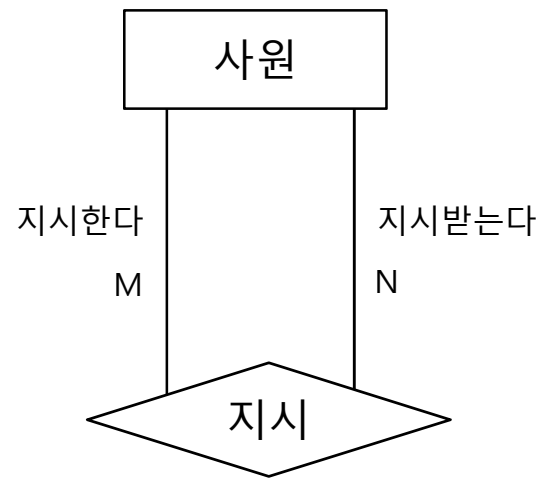
3. 관계와 관계 타입

❖ 순환적 관계

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



(a) 학생의 멘토링 관계



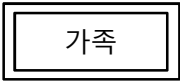
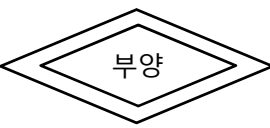
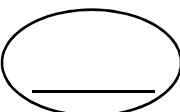

(b) 사원의 지시 관계

순환적 관계의 예

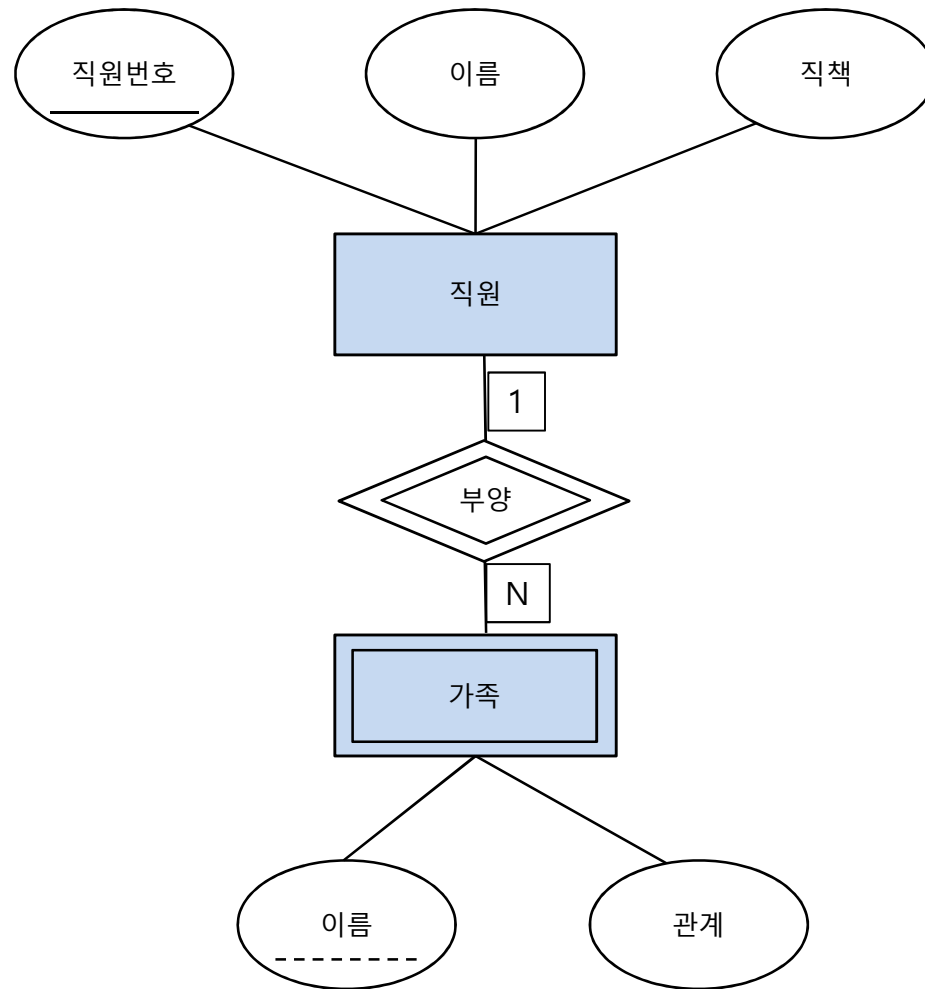
4. 약한 개체 타입과 식별자

- 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

식별자와 약한 개체 타입

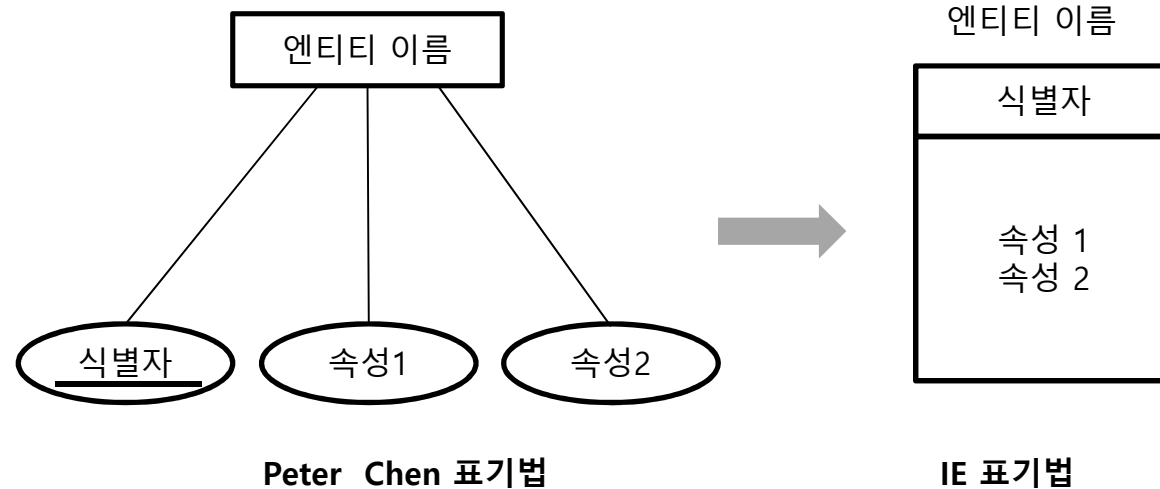
기호	의미	설명
	약한 개체 타입	<ul style="list-style-type: none">• 강한 개체 타입이 있어야 존재할 수 있음• 이중 직사각형으로 표현
	식별 관계 타입	<ul style="list-style-type: none">• 강한 개체 타입과 약한 개체 타입의 관계를 나타냄• 강한 개체 타입의 기본키를 상속받아 사용함• 이중 마름모꼴로 표현
	키	<ul style="list-style-type: none">• 강한 개체 타입의 키 속성
	식별자	<ul style="list-style-type: none">• 약한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

4. 약한 개체 타입과 식별자



5. IE 표기법

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



Peter Chen 표기법과 IE 표기법

5. IE 표기법

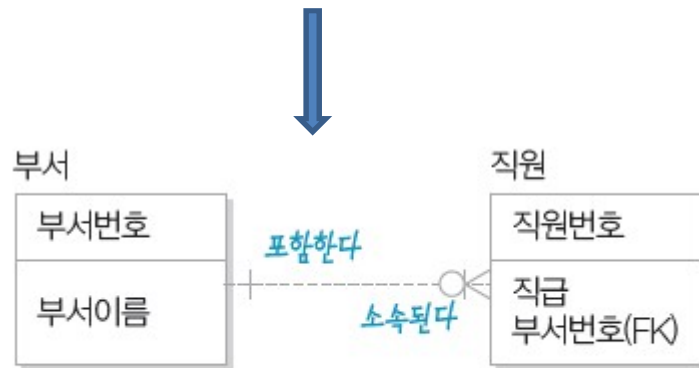
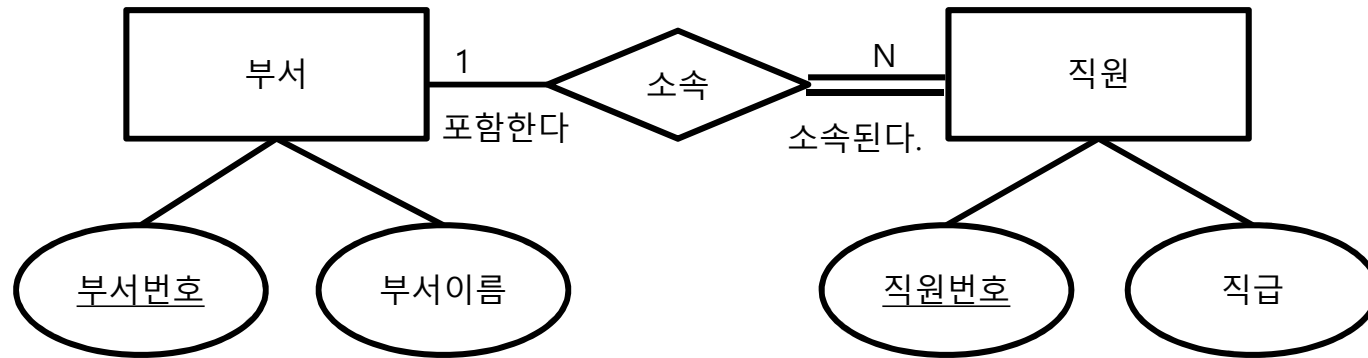
- IE 표기법에서 관계는 실선 혹은 점선으로 표기함

IE 표기법 – 관계와 관계 대응수

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

5. IE 표기법

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



(b) IE 표기법으로 작성한 직원-부서 관계

IE 표기법의 예(비식별자 관계)

- IE 표기법에서 관계(약한관계, 식별자 관계)는 실선으로 표기함



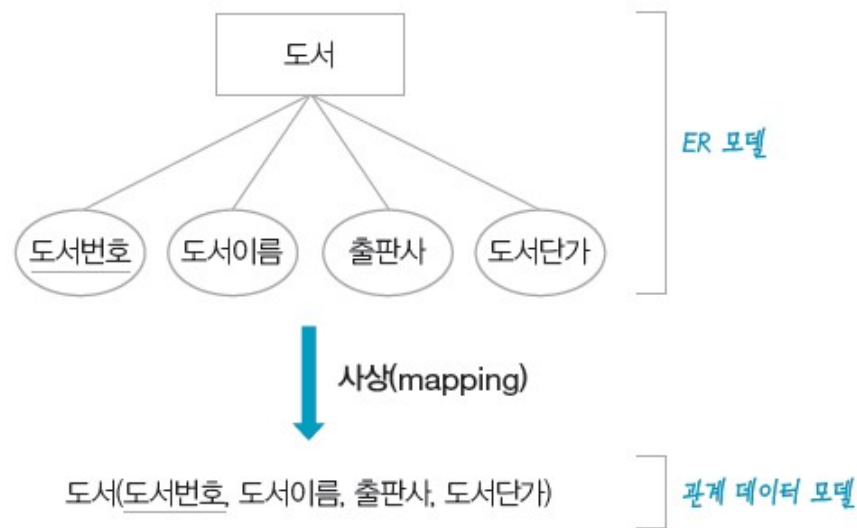
IE 표기법의 예(식별자 관계)

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



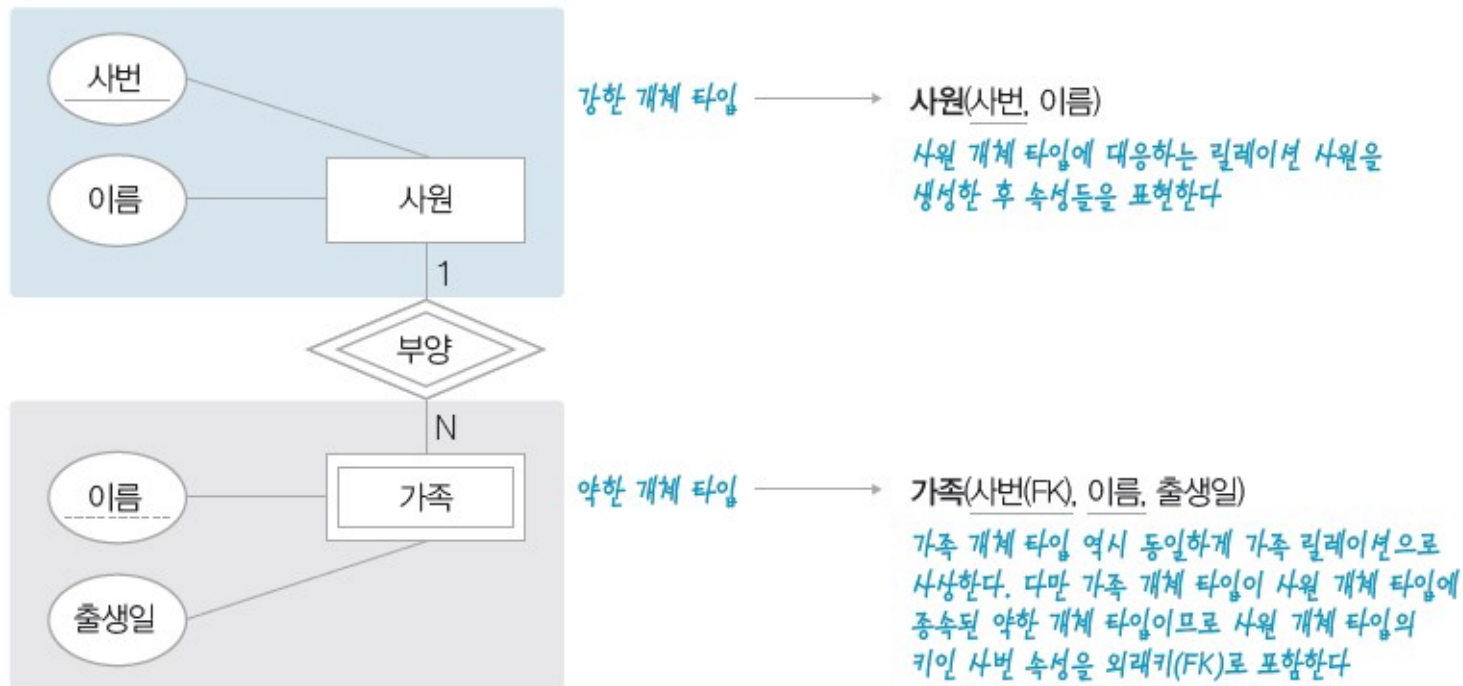
ER 모델을 관계 데이터 모델로 사상

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

ER 모델과 관계 데이터 모델의 사상 알고리즘

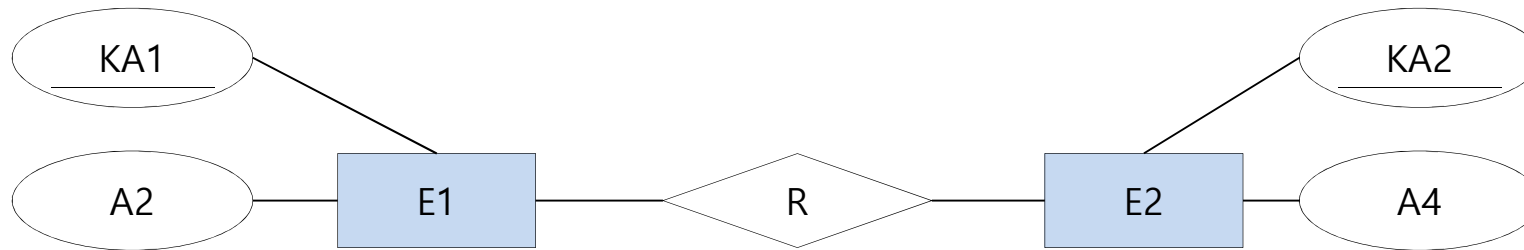
1. 개체 타입의 사상

- [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



개체 타입의 사상

2. 관계 타입의 사상



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.

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

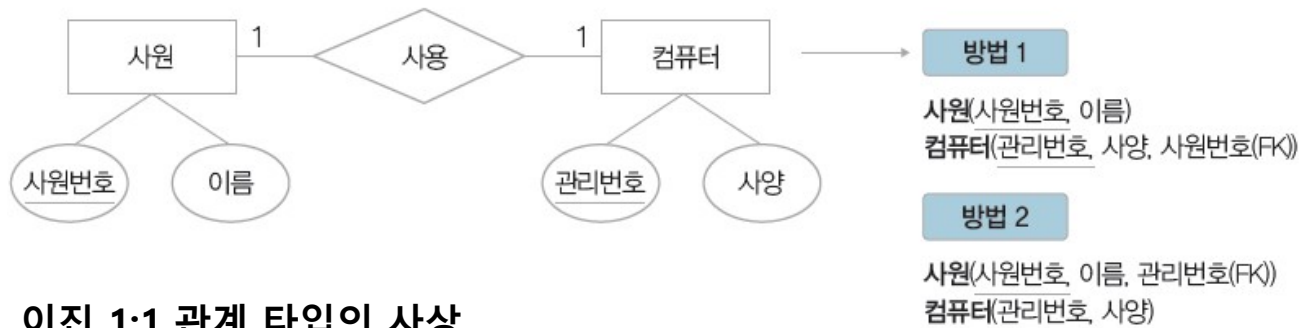
$R(\underline{KA1}, \underline{KA2})$

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

2. 관계 타입의 사상

■ [3단계] 이진 1:1 관계 타입

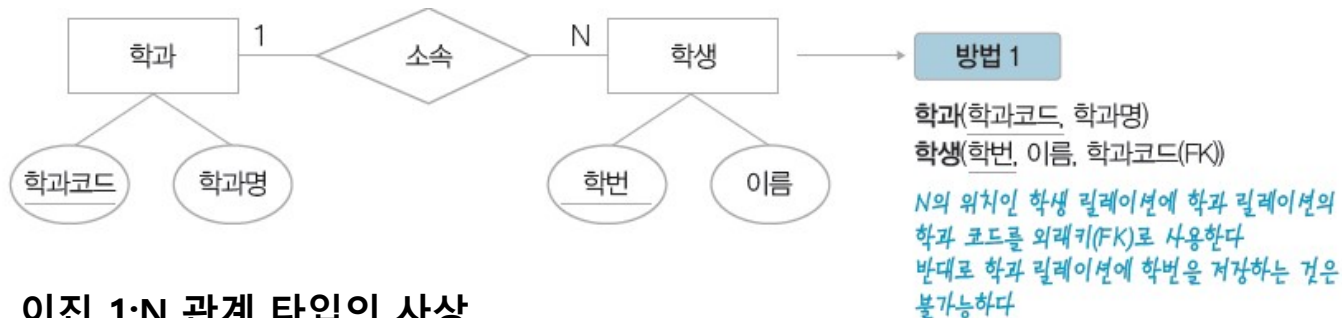
이진 1:1 관계 타입의 경우 [방법1]~[방법4]까지 모든 유형으로 사상 가능, 개체가 가진 정보 유형에 따라 판단



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.

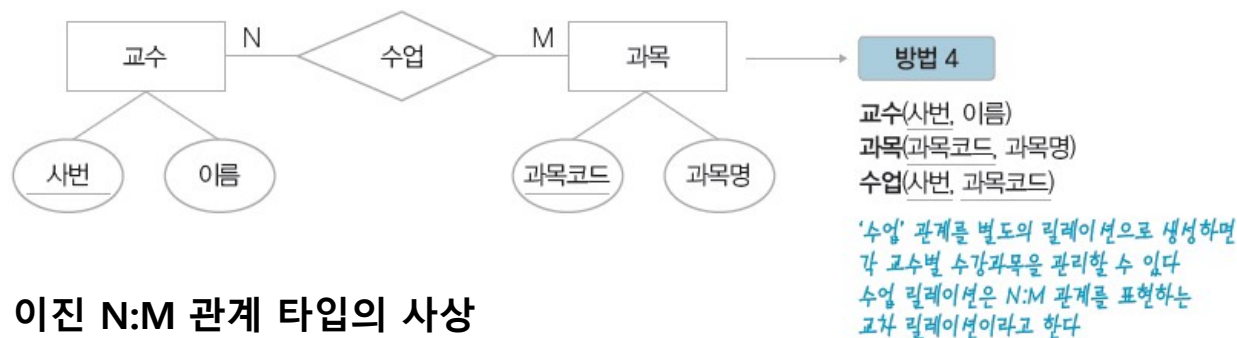


이진 1:N 관계 타입의 사상

2. 관계 타입의 사상

■ [5단계] 이진 M:N 관계 타입

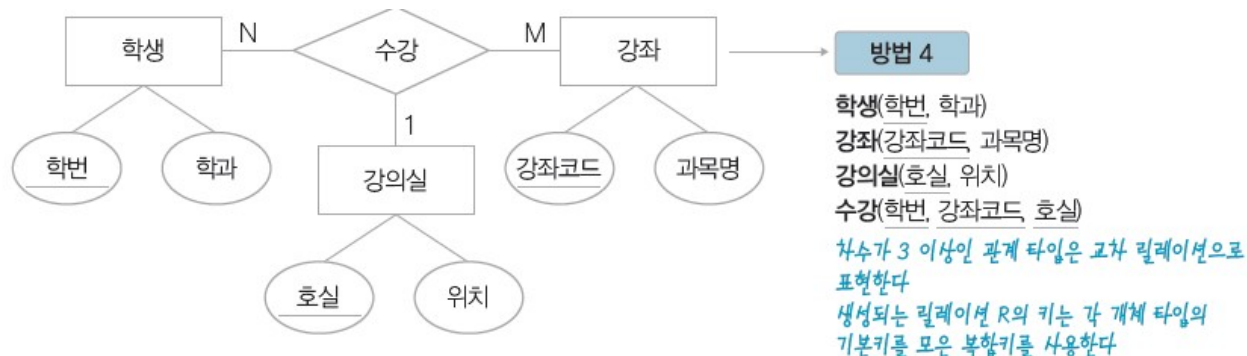
이진 M:N 관계 타입은 [방법4]의 유형으로 사상됨



이진 N:M 관계 타입의 사상

■ [6단계] N진 관계 타입

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



이진 N진 관계 타입의 사상

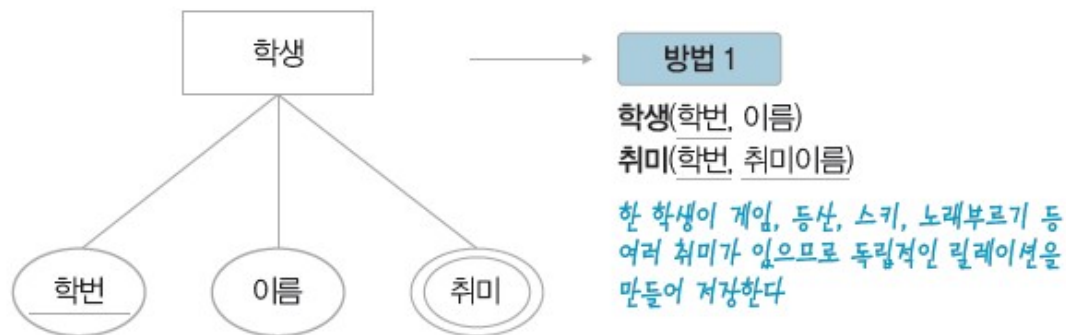
3. 다중값 속성의 사상



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



다중값 속성의 사상

04. 모델링 실습

1. MySQL Workbeanch 기본 화면 및 툴 둘러보기
2. 쇼핑몰 데이터베이스 모델링 실습
3. AA 대학 데이터 Workshop

쇼핑몰 데이터베이스 모델링

쇼핑몰 데이터 예시

- 방문 내역 & 구매내역 데이터
 - 메모장이나 엑셀로 작성되었다 가정
- 기록된 내용에서 물건 구매 내역이 없는 고객 위로 정렬
 - L자형 테이블이 되어 낭비되는 공간 생김
- L자형 테이블을 빈칸이 있는 곳과 없는 곳으로 분류
 - 고객테이블, 구매테이블로 분류하여 공간 절약
 - 고객 테이블 중복 제거
 - 기본 키 (PK, Primary Key)필요
 - 고객 이름을 고객을 구분할 수 있는 구분자로 설정
 - 각 행을 구분하는 유일한 값
 - 기본 키의 조건은 중복되지 않고 비어있지 않아야 함
 - 구매 테이블에 '누가 구매했는지' 표기 위해 고객 이름 필요

쇼핑몰 데이터베이스 모델링

쇼핑몰 데이터 예

- 테이블 간의 업무적인 연관성(Relation) 정의
 - 주 (Master)가 되는 쪽이 부모 테이블
 - ex) 고객이 물건을 소유 (O) , 물건이 고객을 소유 (X)
 - 주가 되는 고객 테이블이 부모, 상세가 되는 구매 테이블이 자식이 됨 (1:N 모델)
 - 기본 키 (PK, Primary Key)
 - 중복되지 않고 비어있지 않아야 함
 - 외래 키 (FK, Foreign Key)
 - 외래 키로 부모 테이블에서 유일하게 하나의 정보를 얻을 수 있음
 - 제약조건
 - 새로운 데이터 들어갈 때는 부모 테이블에 먼저 넣어야 함
 - 데이터 삭제 시에는 자식 테이블에서도 지워야 함

쇼핑몰 데이터베이스 모델링

쇼핑몰 데이터 예시

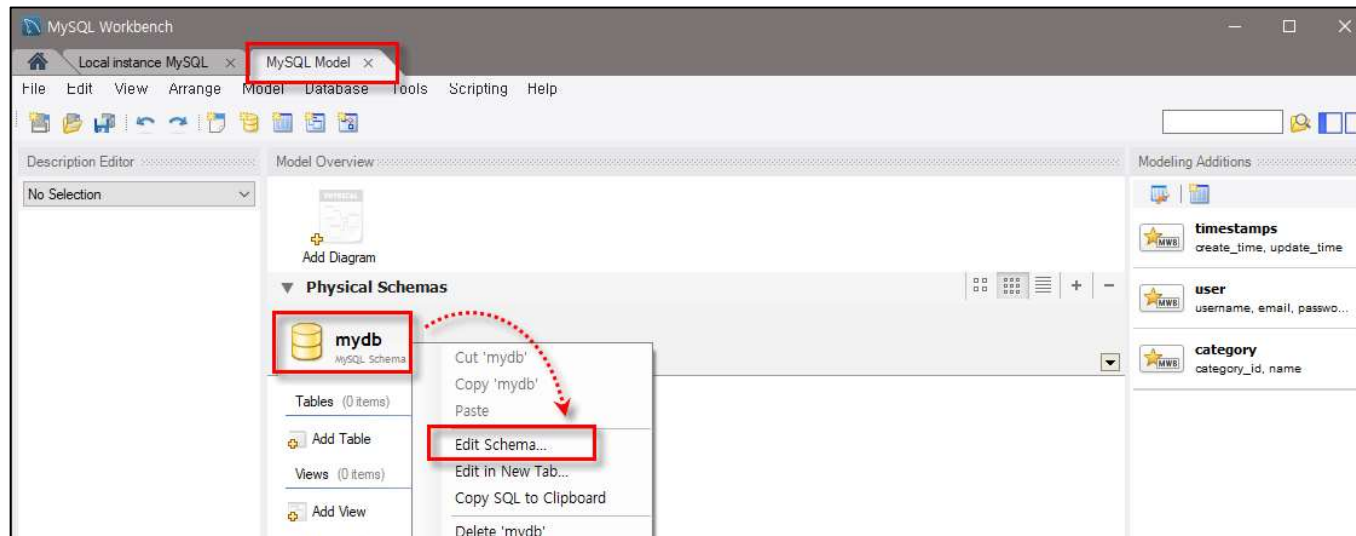
- 완성된 고객 테이블과 구매 테이블의 구조 정의

테이블 이름	열 이름	데이터 형식	Null 허용	기타
고객 테이블 (userTBL)	고객 이름(userName)	문자(최대 3글자)	X	PK
	출생년도(birthYear)	숫자(정수)	X	
	주소(addr)	문자(최대 2글자)	X	
	연락처(mobile)	문자(최대 12글자)	O	
구매 테이블 (buyTBL)	고객 이름(userName)	문자(최대 3글자)	X	FK
	구매한 물건(prodName)	문자(최대 3글자)	X	
	단가(price)	숫자(정수)	X	
	수량(amount)	숫자(정수)	X	

쇼핑몰 데이터베이스 모델링

정의한 테이블을 다이어그램으로 만들기

- Workbench 실행하고 localhost로 접속, 열린 쿼리 창 모두 닫기
- 모델 다이어그램 작성
 - [File] >> [New Model] 선택
 - [MySQL Model] 탭에서 DB 이름 수정
 - 기본적으로 데이터베이스 이름은 'mydb'
 - 데이터베이스에서 마우스 오른쪽 버튼 클릭 후 [Edit Schema], 'modelDB'로 이름 수정



쇼핑몰 데이터베이스 모델링

정의한 테이블을 다이어그램으로 만들기

- 모델 다이어그램 작성
 - [Model Overview]의 [Add Diagram] 더블클릭
 - [EER Diagram] 탭 추가되고 다이어그램 그릴 수 있는 상태

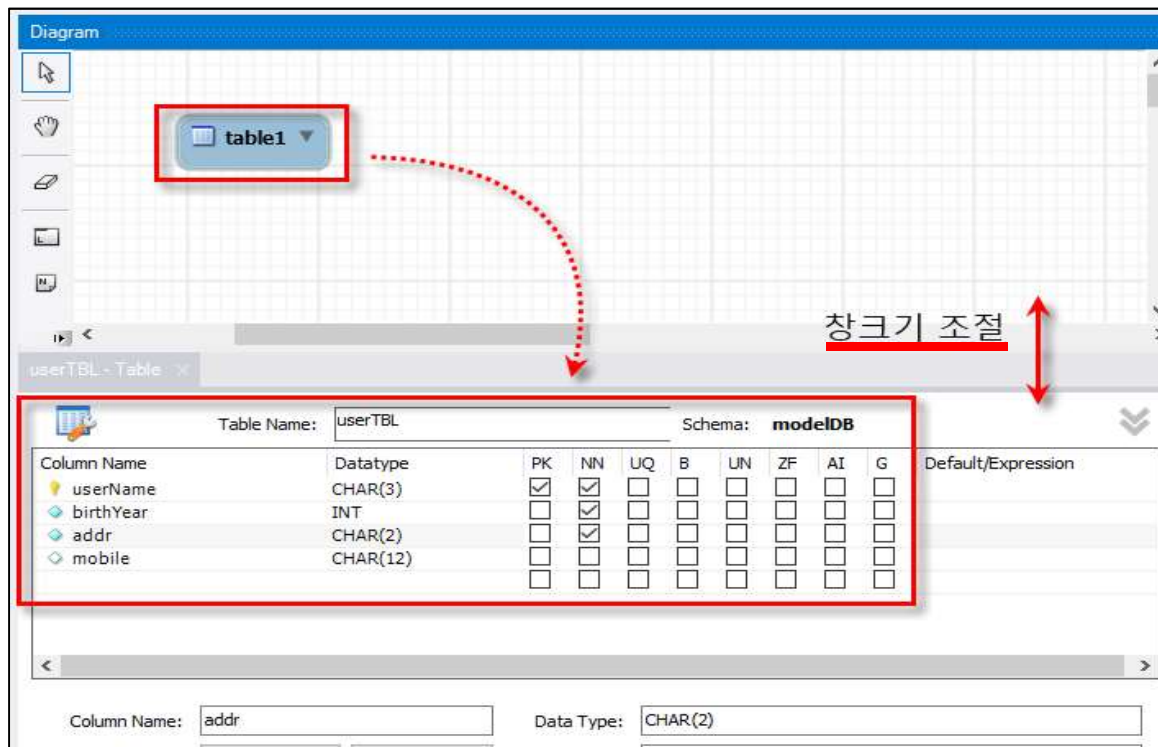


쇼핑몰 데이터베이스 모델링

정의한 테이블을 다이어그램으로 만들기

모델 다이어그램 작성

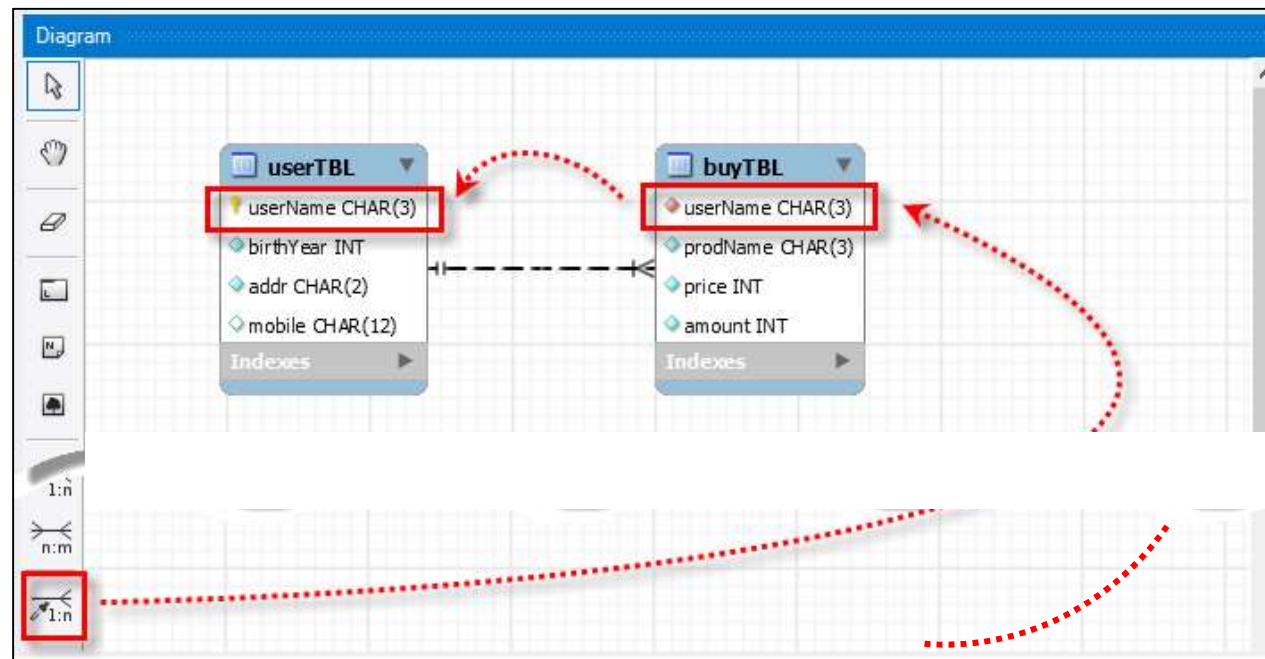
- [Place a New Table] 아이콘 클릭 → 빈 화면에서 다시 마우스 클릭해 테이블 생성
 - 다이어그램의 table1을 더블 클릭 → 고객 테이블(userTBL) 만들기
 - 같은 과정 반복해 구매 테이블(buyTBL) 작성



정의한 테이블을 다이어그램으로 만들기

◦ 모델 다이어그램 작성

- 테이블 간에 1:N 관계 맺어주기
 - <Place a Relationship Using Existing column> 아이콘 클릭
 - buyTBL의 'userName' 열과 userTBL의 'userName' 열을 차례로 클릭



정의한 테이블을 다이어그램으로 만들기

- 모델링 파일 실제 데이터베이스에 적용
 - Workbench 메뉴의 [File] >> [Open Model] → modelDB.mwb 열기
 - [Database] >> [Forward Engineer] 선택
 - [Forward Engineer to Database] 시작되면
 - [Set Parameters for connecting to a DBMS] 기본값
 - [Set Options for Database to be Created] 기본값

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 MySQL 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.

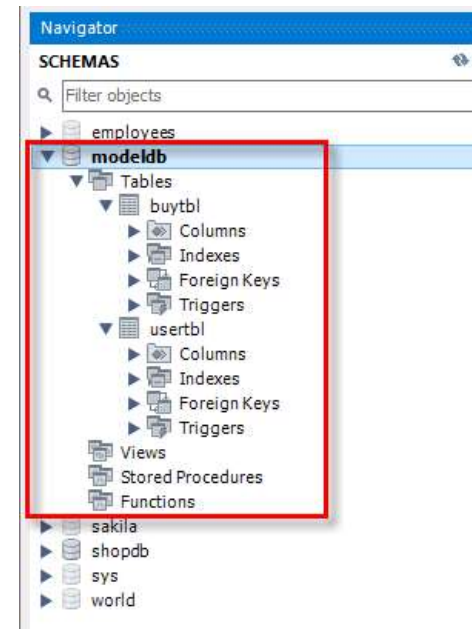
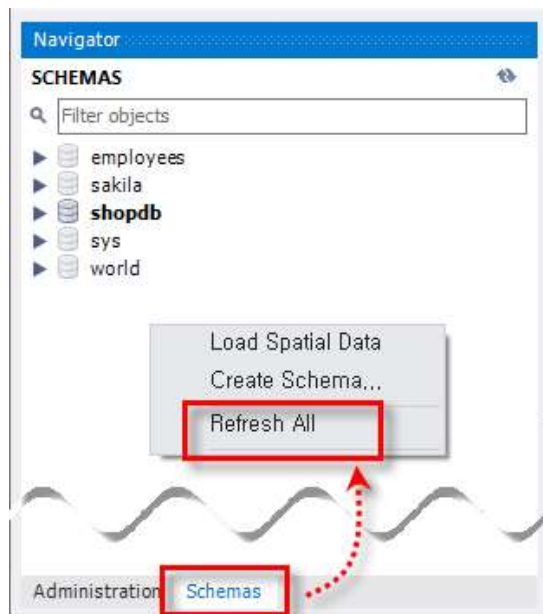
정의한 테이블을 다이어그램으로 만들기

- 모델링 파일 실제 데이터베이스에 적용
 - Root 비밀번호 입력
 - [Select Objects to Forward Engineer]에는 'Export MySQL Table Objects' 체크
 - [Review the SQL Script to be Executed] → 자동 SQL문 생성



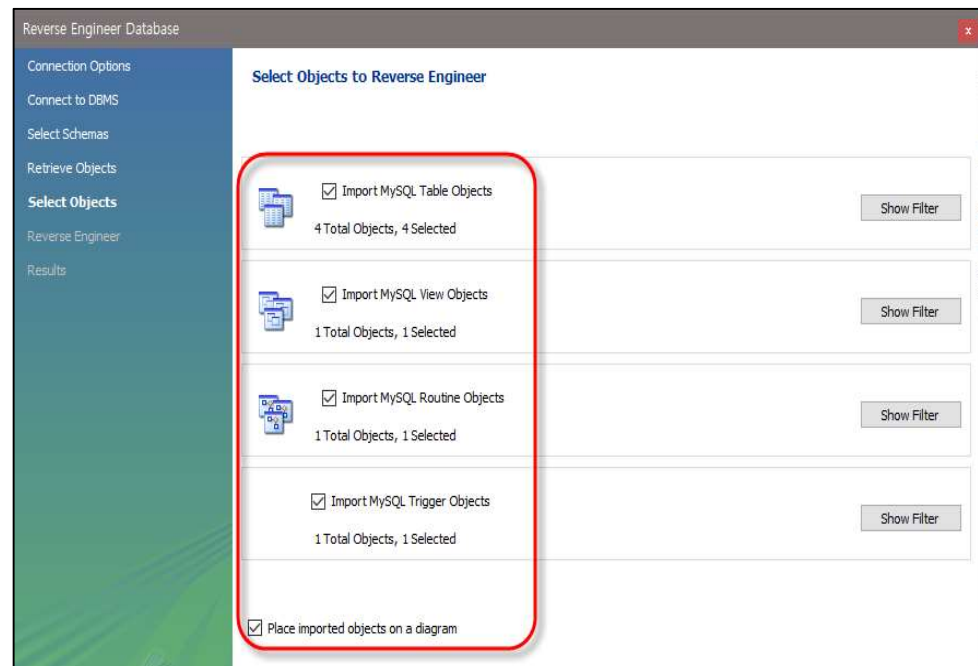
다이어그램에서 데이터베이스로 보내내기한 결과 확인

- [Navigator] >> [Schemas]
 - 빈 곳에서 마우스 오른쪽 버튼 클릭한 후 [Refresh All] 선택하여 새로고침
- modelDB 데이터베이스 확장해 테이블 확인



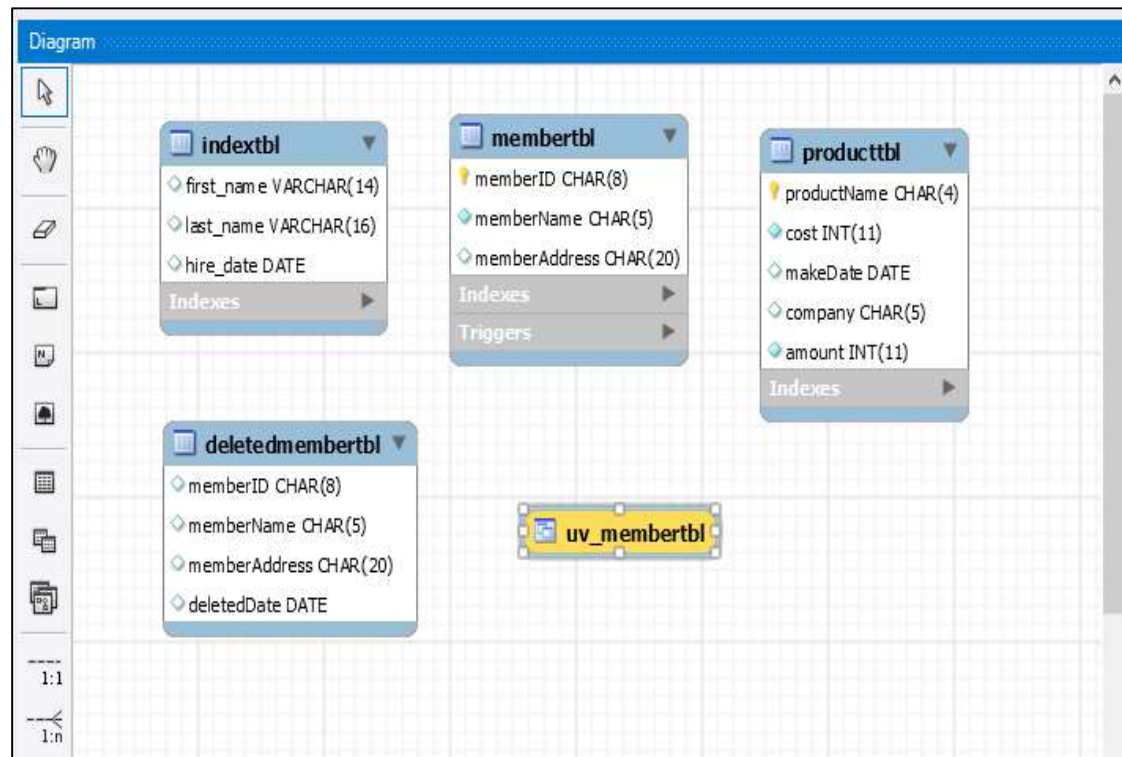
기존 존재하는 데이터베이스 이용해 다이어그램 작성

- ShopDB의 테이블, 인덱스, 스토어드 프로시저, 트리거를 다이어그램으로 변경
- Workbench 메뉴의 [Database] >> [Reverse Engineer] 선택
 - [Set Parameters for connecting to a DBMS]
 - [Connect to DBMS and Fetch Information]
 - [Select the schemas below you want to include:]
 - [Retrieve and Reverse Engineer Schema Objects]
 - [Select Objects to Reverse Engineer]



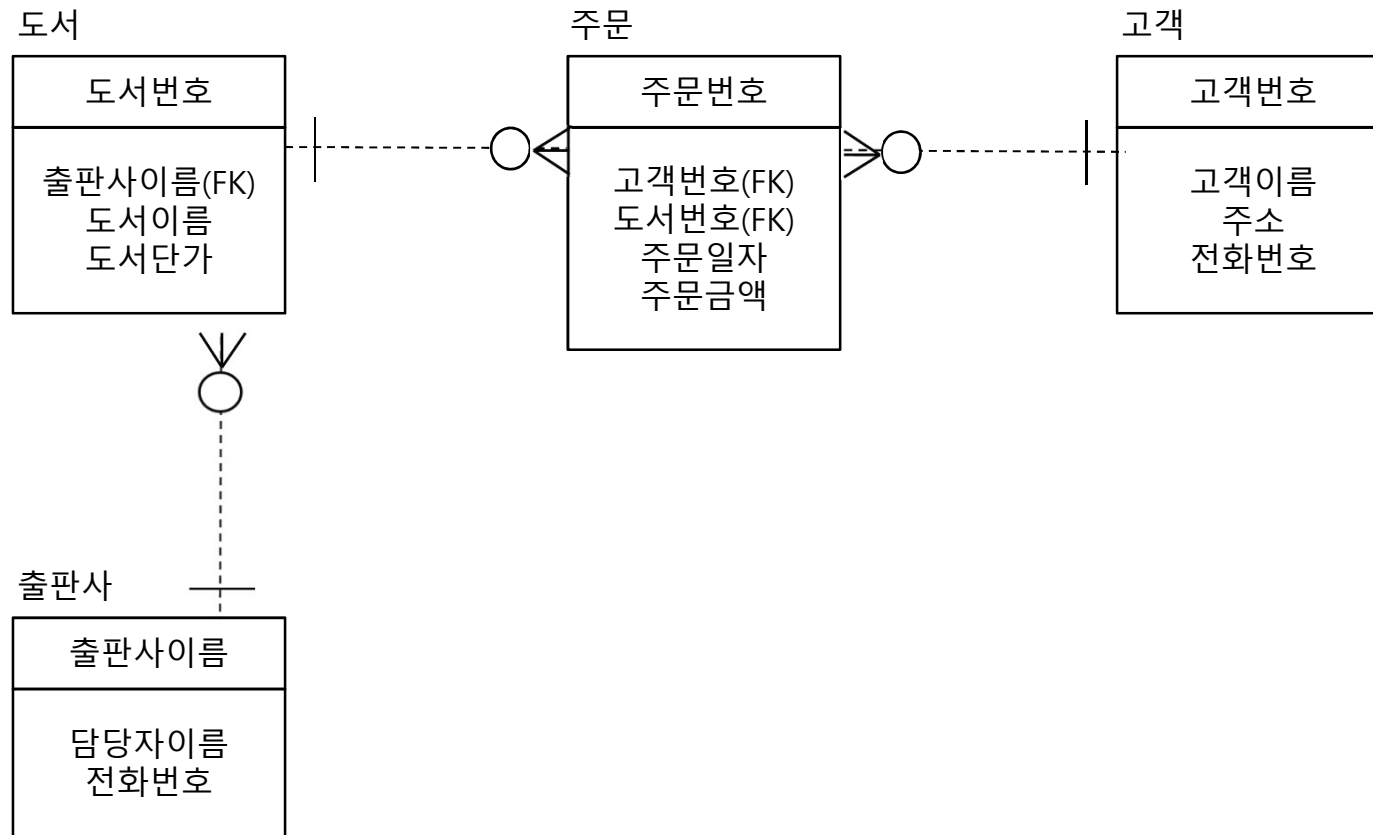
기존 존재하는 데이터베이스 이용해 다이어그램 작성

- ShopDB의 테이블, 인덱스, 스토어드 프로시저, 트리거를 다이어그램으로 변경
- Workbench 메뉴의 [Database] >> [Reverse Engineer] 선택
 - [Reverse Engineering Progress] 의 세부 단계 설정
 - [Reverse Engineering Results]에서
4개에 테이블, 1개 뷰, 1개 루틴(=스토어드 프로시저)
변환 된 것을 확인



실습

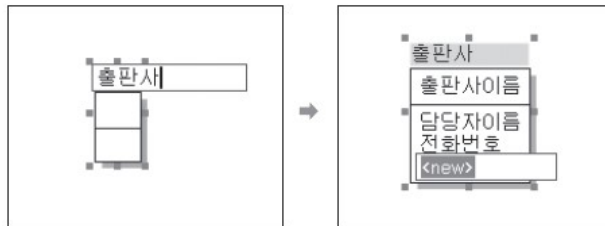
■ 데이터 모델링을 하기 위한 프로그램. IE 표기법을 지원함



서점 ER 다이어그램

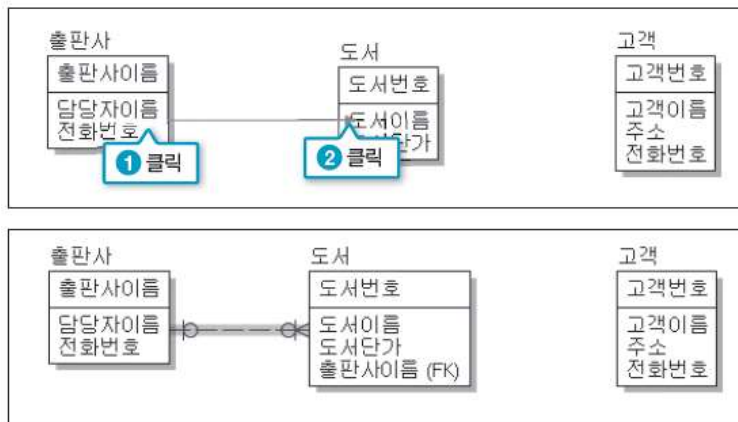
❖ 서점의 논리적 모델링

① 서점의 요구사항 분석 후 개체 만들기

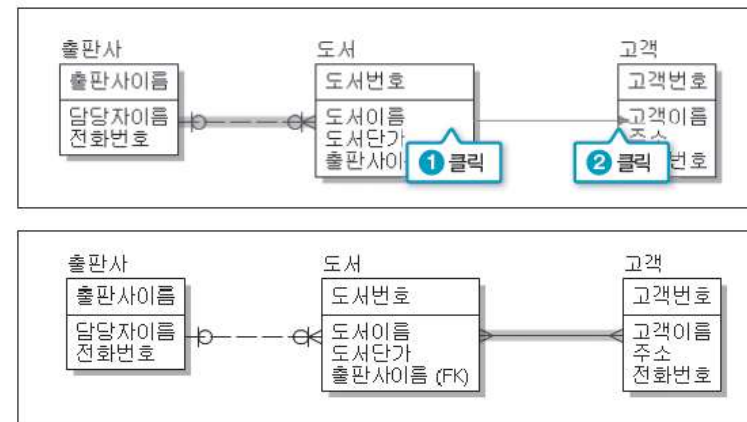


출판사 개체 생성

② 개체 간 관계 표현하기



출판사, 도서 개체의 관계 설정(1:N 비식별)

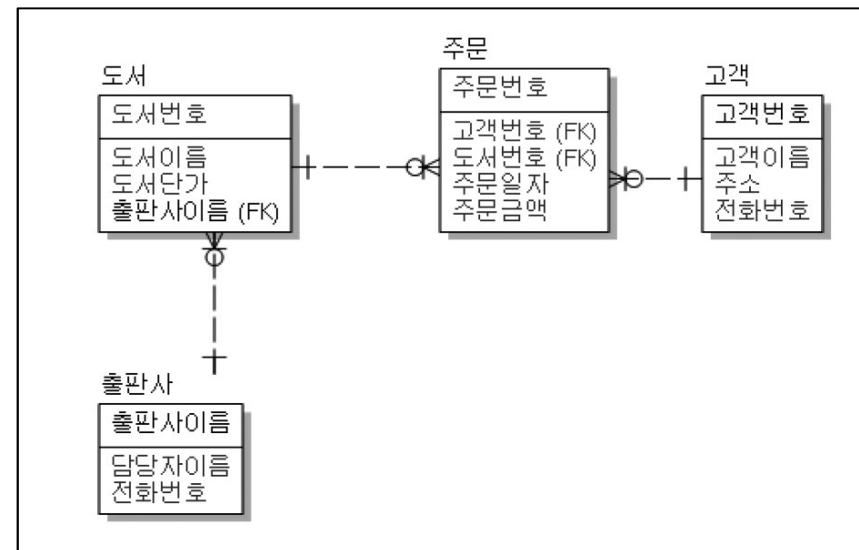


도서, 고객 개체의 관계 설정(N:M 식별)

❖ 도메인 정의하기

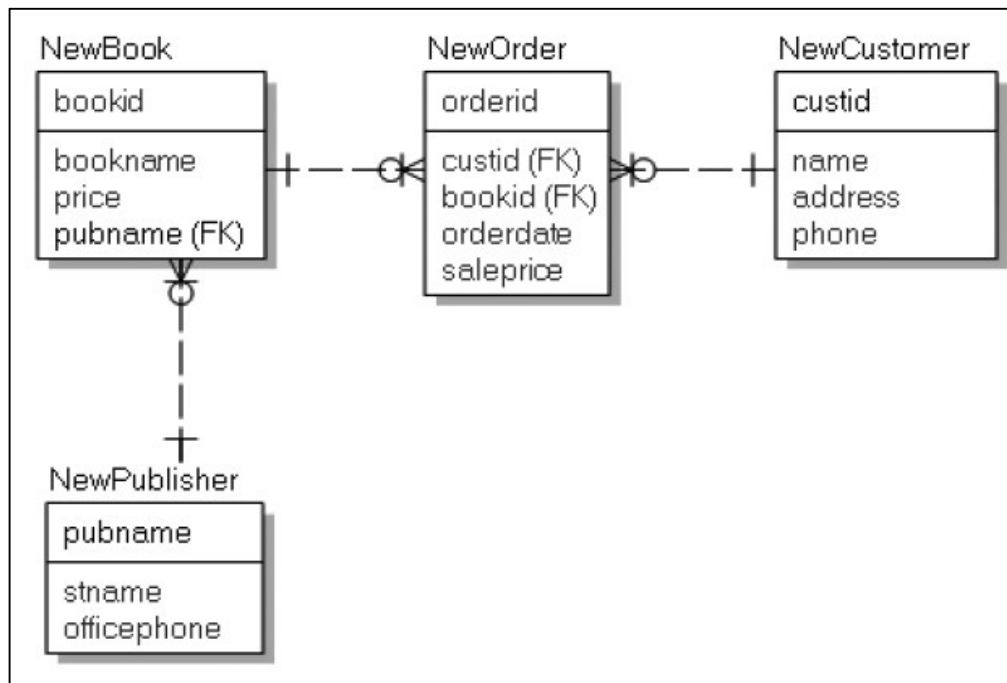
- 도메인이란 속성이 가질 수 있는 값을 정의하는 것. ER 다이어그램이 완성 후 도메인을 정의함

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



❖ 서점의 물리적 모델링

③ 물리적 모델링



05. 모델링 연습(AA 대학 데이터베이스)

1. AA대학의 요구사항
2. AA대학의 ER 다이어그램

1. AA대학의 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.