

# 관계형 데이터 모델링

1. 릴레이션
2. 무결성 제약

## 관계형 데이터 모델이란

- 개념적 구조를 논리적 구조로 표현하는 논리적 데이터 모델
- 하나의 개체에 대한 데이터를 하나의 릴레이션에 저장

열(속성, 애트리뷰트)

고객아이디	고객이름	나이	등급	직업	적립금
CHAR(20)	CHAR(20)	INT	CHAR(10)	CHAR(10)	INT
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

도메인

행(튜플)

그림 5-1 릴레이션의 예 : 고객 릴레이션

# 관계 데이터 모델의 기본 용어

- 릴레이션(relation)
  - 하나의 개체에 관한 데이터를 2차원 테이블의 구조로 저장한 것
  - 파일 관리 시스템 관점에서 파일(file)에 대응
- 속성(attribute)
  - 릴레이션의 열, 애트리뷰트
  - 파일 관리 시스템 관점에서 필드(field)에 대응
- 튜플(tuple)
  - 릴레이션의 행
  - 파일 관리 시스템 관점에서 레코드(record)에 대응

- 도메인(domain)

- 하나의 속성이 가질 수 있는 모든 값의 집합
- 속성 값을 입력 및 수정할 때 적합성 판단의 기준이 됨
- 일반적으로 속성의 특성을 고려한 데이터 타입으로 정의

- 널(null)

- 속성 값을 아직 모르거나 해당되는 값이 없음을 표현

- 차수(degree)

- 하나의 릴레이션에서 속성의 전체 개수

- 카디널리티(cardinality)

- 하나의 릴레이션에서 튜플의 전체 개수

열(속성, 애트리뷰트)

고객아이디	고객이름	나이	등급	직업	적립금
CHAR(20)	CHAR(20)	INT	CHAR(10)	CHAR(10)	INT
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

도메인

행(튜플)

그림 5-1 릴레이션의 예 : 고객 릴레이션

<고객 릴레이션의 차수는 6, 카디널리티는 4, 도메인은 INT>

고객아이디	고객이름	나이	등급	직업	적립금	릴레이션 스키마
apple	김현준	20	gold	학생	1000	릴레이션 인스턴스
banana	정소화	25	vip	간호사	2500	
carrot	원유선	28	gold	교사	4500	
orange	정지영	22	silver	학생	0	

그림 5-2 릴레이션 구성의 예 : 고객 릴레이션

# 데이터베이스의 구성

- 데이터베이스 스키마(database schema)
  - 데이터베이스의 전체 구조
  - 데이터베이스를 구성하는 릴레이션 스키마의 모음
- 데이터베이스 인스턴스(database instance)
  - 데이터베이스를 구성하는 릴레이션 인스턴스의 모음

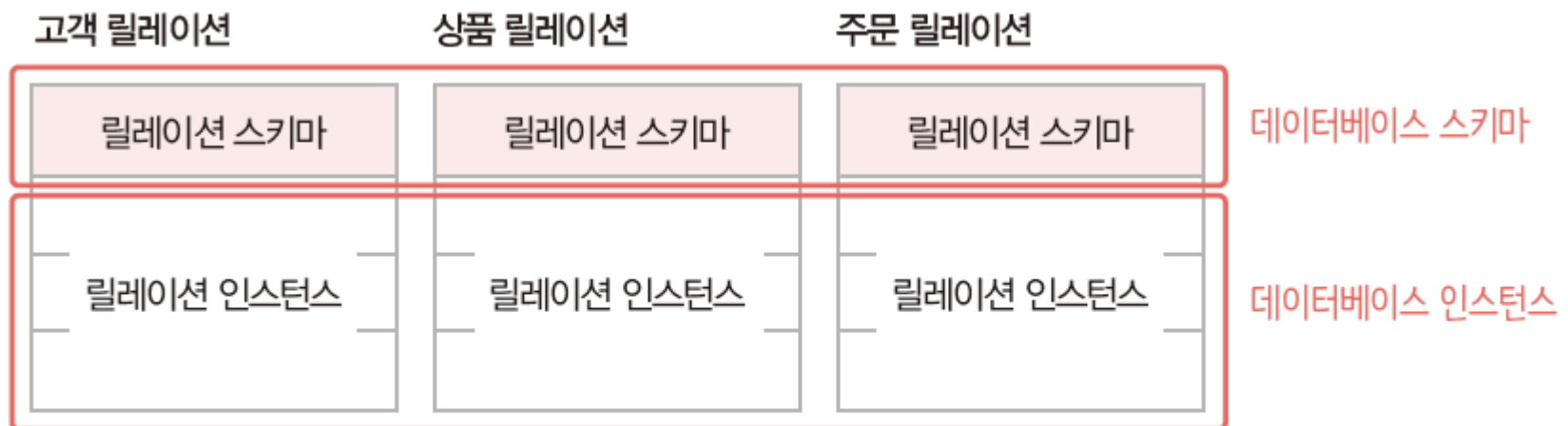


그림 5-3 데이터베이스 구성의 예 : 쇼핑몰 데이터베이스

### 릴레이션의 특성

- 튜플의 유일성
  - 하나의 릴레이션에는 동일한 튜플이 존재할 수 없다.
- 튜플의 무순서
  - 하나의 릴레이션에서 튜플 사이의 순서는 무의미하다.
- 속성의 무순서
  - 하나의 릴레이션에서 속성 사이의 순서는 무의미하다.
- 속성의 원자성
  - 속성 값으로 원자 값만 사용할 수 있다.



고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	회사원, 학생	0

그림 5-4 다중 값 속성을 포함하는 릴레이션의 예 : 고객 릴레이션

## 키(key)

- 릴레이션에서 튜플들을 유일하게 구별하는 속성 또는 속성들의 집합

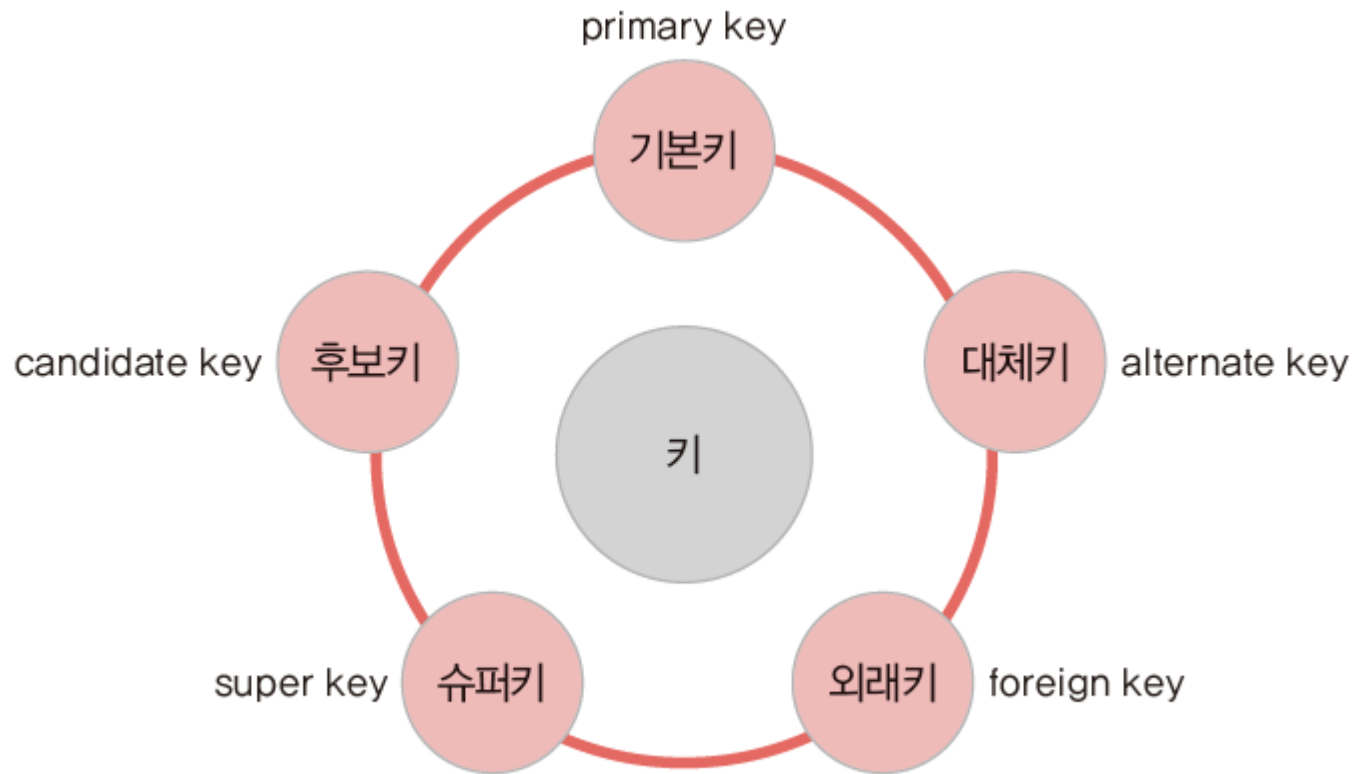


그림 5-5 키의 종류

## 키(Key)의 특성

- 유일성(uniqueness)
  - 하나의 릴레이션에서 모든 튜플은 서로 다른 키 값을 가져야 함
- 최소성(minimality)
  - 꼭 필요한 최소한의 속성들로만 키를 구성함

# 키의 종류

- 슈퍼키(super key)
  - 유일성을 만족하는 속성 또는 속성들의 집합
  - 예) 고객 릴레이션의 슈퍼키 : 고객아이디, (고객아이디, 고객이름), (고객이름, 주소) 등
- 후보키(candidate key)
  - 유일성과 최소성을 만족하는 속성 또는 속성들의 집합
  - 예) 고객 릴레이션의 후보키 : 고객아이디, (고객이름, 주소) 등
- 기본키(primary key)
  - 후보키 중에서 기본적으로 사용하기 위해 선택한 키
  - 예) 고객 릴레이션의 기본키 : 고객아이디

- 대체키(alternate key)
  - 기본키로 선택되지 못한 후보키
  - 예) 고객 릴레이션의 대체키 : (고객이름, 주소)

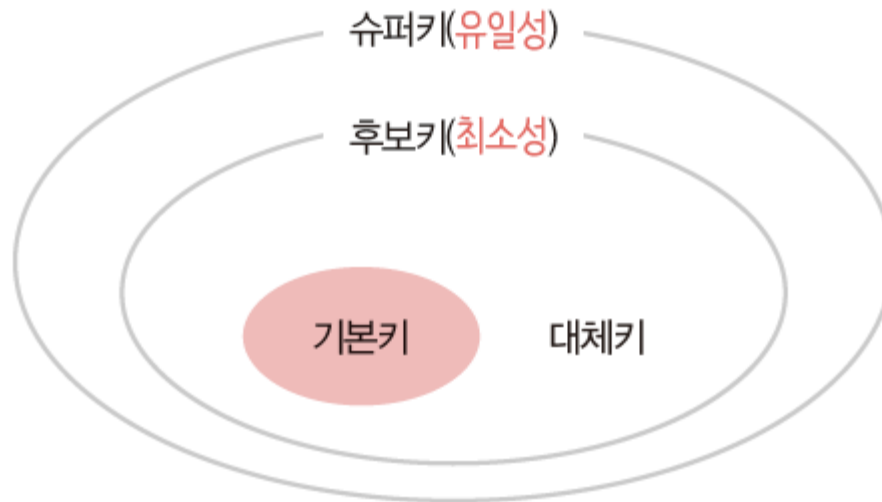


그림 5-8 키의 관계

## ■ 외래키(foreign key)

- 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합
- 릴레이션들 간의 관계를 표현
  - 참조하는 릴레이션 : 외래키를 가진 릴레이션
  - 참조되는 릴레이션 : 외래키가 참조하는 기본키를 가진 릴레이션

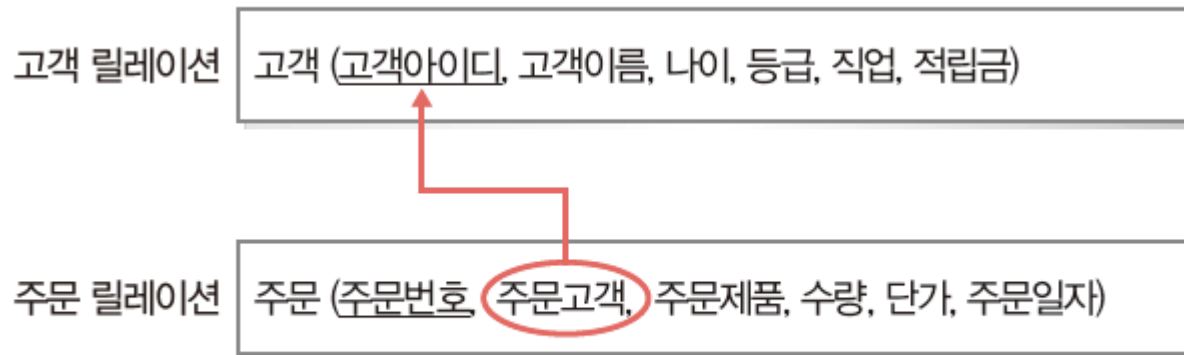


그림 5-9 고객 릴레이션과 주문 릴레이션의 스키마

외래키 속성과 그것이 참조하는 기본키 속성의 이름은 달라도 되지만 도메인은 같아야 한다.



그림 5-10 외래키의 예 : 고객 릴레이션과 주문 릴레이션

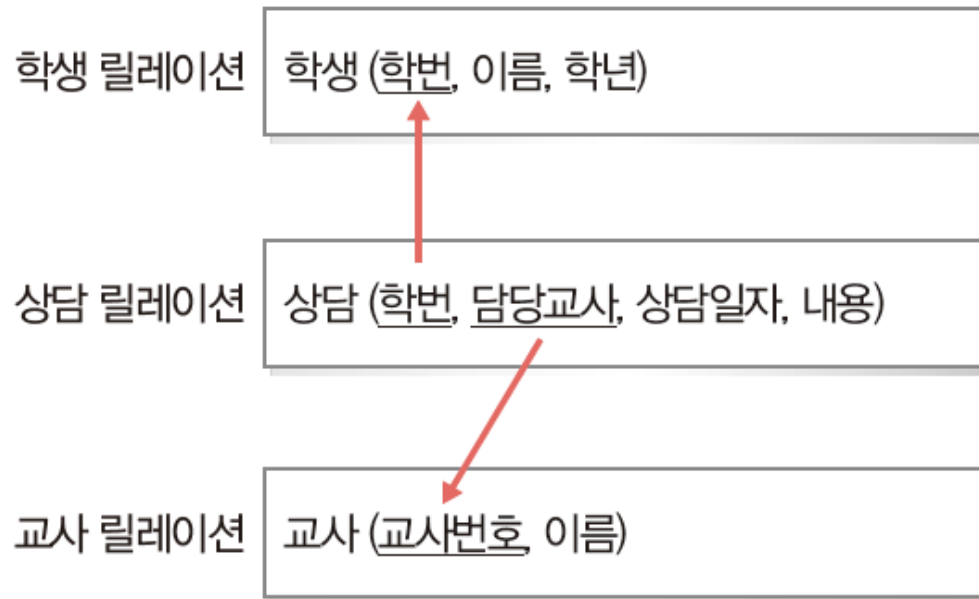


그림 5-11 학생 상담 데이터베이스 스키마

하나의 릴레이션에는 외래키가 여러 개 존재할 수도 있고  
외래키를 기본키로 사용할 수도 있다.



고객 릴레이션의 기본키

고객아이디	고객이름	나이	등급	직업	적립금	추천고객
apple	김현준	20	gold	학생	1000	orange
banana	정소화	25	vip	간호사	2500	orange
carrot	원유선	28	gold	교사	4500	apple
orange	정지영	22	silver	학생	0	NULL

고객 릴레이션의 외래키

그림 5-12 기본키와 외래키의 관계가 함께 정의된 릴레이션의 예 : 고객 릴레이션

같은 릴레이션의 기본키를 참조하는 외래키도 정의할 수 있다.  
그리고 외래키 속성은 널 값을 가질 수도 있다.

특성	<ul style="list-style-type: none"> <li>• 유일성 : 한 릴레이션에서 모든 튜플은 서로 다른 키 값을 가져야 함</li> <li>• 최소성 : 꼭 필요한 최소한의 속성들로만 키를 구성함</li> </ul>
종류	<ul style="list-style-type: none"> <li>• 수퍼키 : 유일성을 만족하는 속성 또는 속성들의 집합</li> <li>• 후보키 : 유일성과 최소성을 만족하는 속성 또는 속성들의 집합</li> <li>• 기본키 : 후보키 중에서 기본적으로 사용하기 위해 선택한 키</li> <li>• 대체키 : 기본키로 선택되지 못한 후보키</li> <li>• 외래키 : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합</li> </ul>

## 무결성 제약조건(integrity constraint)

- 데이터의 무결성을 보장하고 일관된 상태로 유지하기 위한 규칙
- 무결성 : 데이터를 결함이 없는 상태, 즉 정확하고 유효하게 유지하는 것

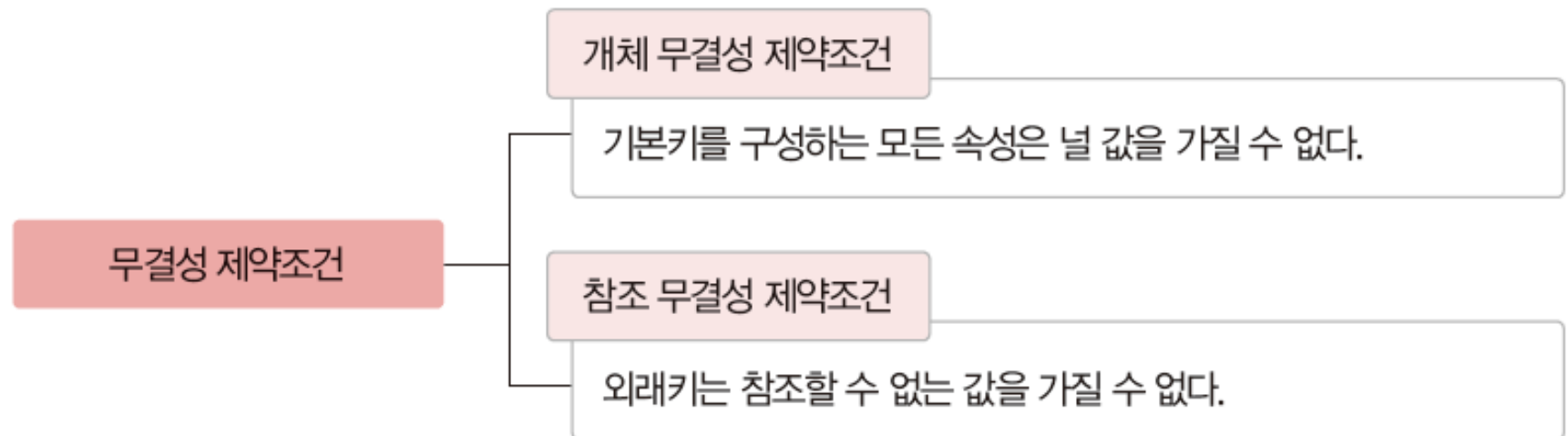


그림 5-13 관계 데이터 모델의 무결성 제약조건

## 개체 무결성 제약조건(entity integrity constraint)

- 기본키를 구성하는 모든 속성은 널 값을 가질 수 없는 규칙

<u>고객아이디</u>	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
NULL	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
NULL	정지영	22	silver	학생	0

그림 5-14 개체 무결성 제약조건을 위반한 릴레이션의 예 : 고객 릴레이션

## 참조 무결성 제약조건(referential integrity constraint)

- 외래키는 참조할 수 없는 값을 가질 수 없는 규칙



- 외래키는 NULL이 될 수 있다



그림 5-16 외래키가 널 값인 릴레이션의 예 : 주문 릴레이션

외래키 속성이 널 값을 가진다고 해서 참조 무결성 제약조건을 위반한 것은 아니다.

# 데이터베이스 설계

1. 데이터베이스 설계 단계
2. 요구 사항 분석
3. 개념적 설계
4. 논리적 설계
5. 물리적 설계

## 데이터베이스 설계

정규화를 이용한 설계

E-R 모델과 릴레이션 변환 규칙을 이용한 설계

1단계

요구 사항 분석

요구 사항 명세서

2단계

개념적 설계

E-R 다이어그램

3단계

논리적 설계

릴레이션 스키마

4단계

물리적 설계

물리적 스키마

5단계

구현





그림 8-2 데이터베이스 설계 과정의 각 단계별 주요 작업과 결과물

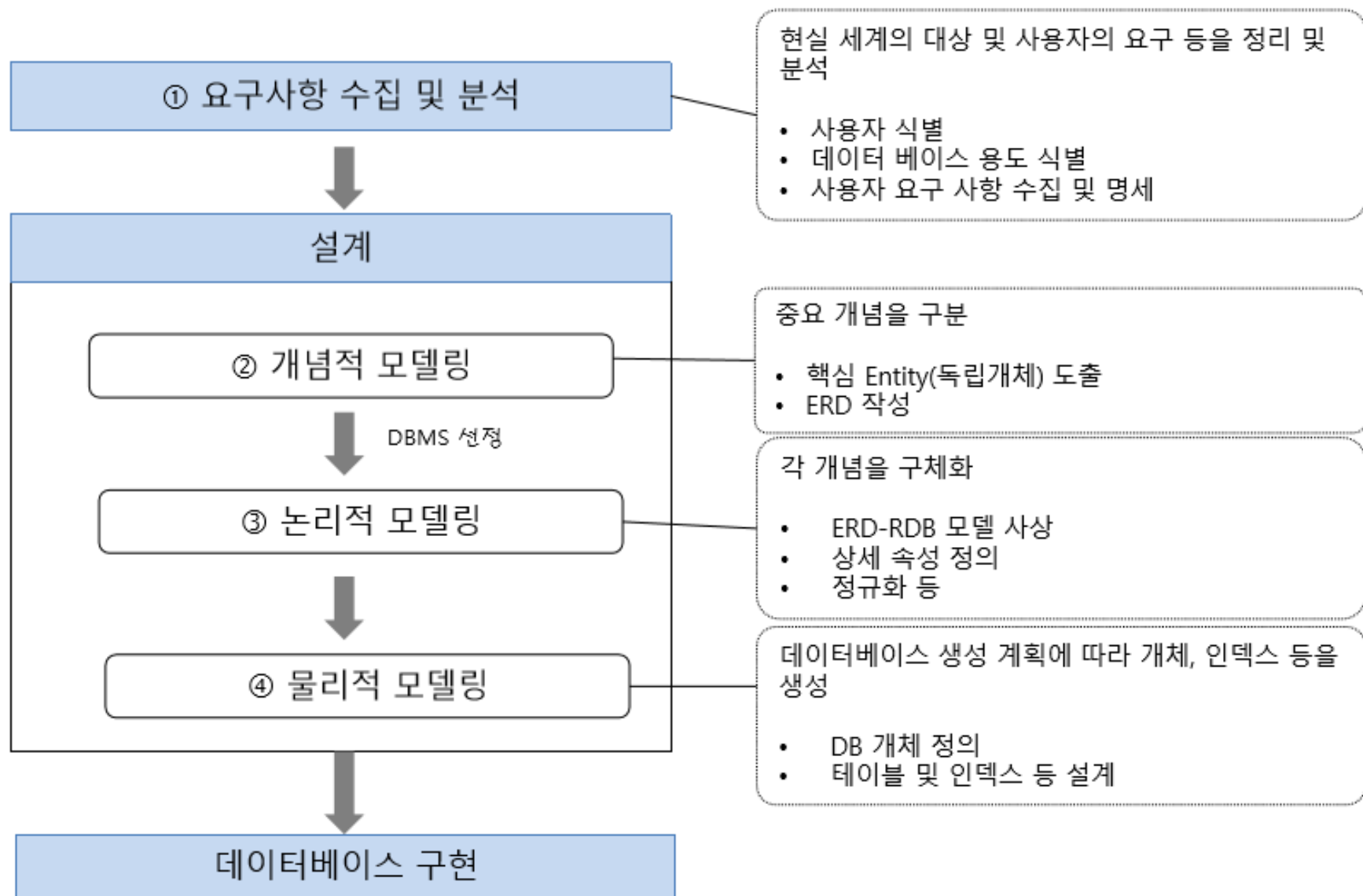


그림 6-4 데이터 모델링 과정

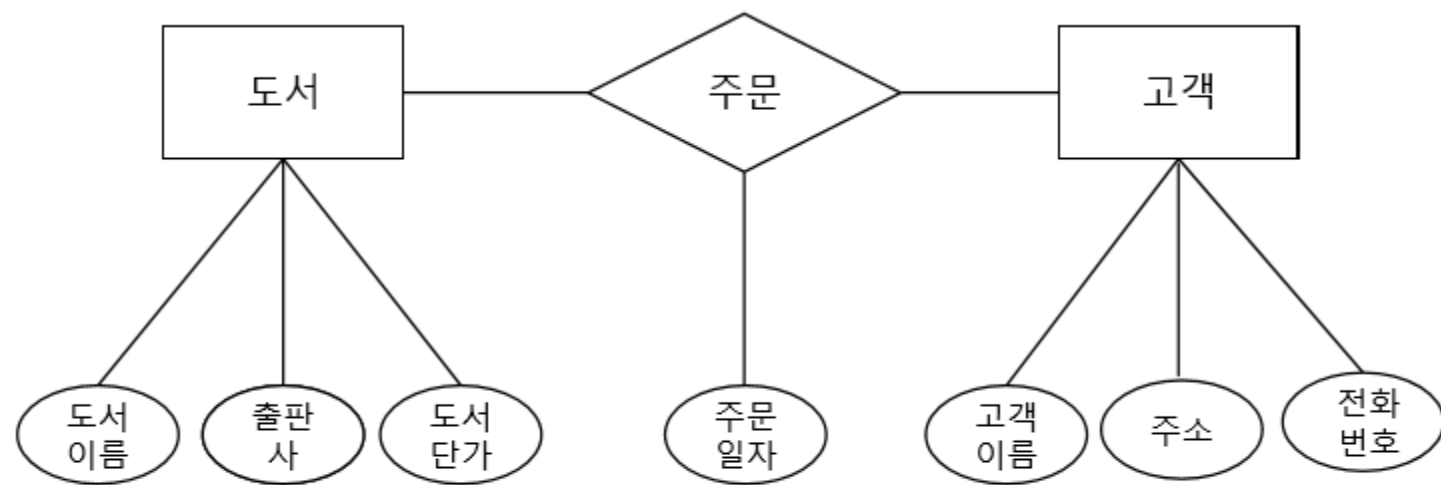


그림 6-5 개념적 모델링의 예



도서 (도서번호, 도서이름, 출판사이름, 도서단가)

고객 (고객번호, 고객이름, 주소, 전화번호)

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

그림 6-6 논리적 모델링의 예

# DBMS

도서 (도서번호, 도서이름, 출판사이름, 도서단가)

```
CREATE TABLE Book (  
  bookid    INT PRIMARY KEY,  
  bookname  VARCHAR(40),  
  publisher VARCHAR(40),  
  price     INT  
);
```

고객 (고객번호, 고객이름, 주소, 전화번호)

```
CREATE TABLE Customer (  
  custid    INT PRIMARY KEY,  
  name      VARCHAR(40),  
  address   VARCHAR(40),  
  phone     VARCHAR(30)  
);
```

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

```
CREATE TABLE Orders (  
  orderid INT PRIMARY KEY,  
  custid  INT REFERENCES Customer(custid),  
  bookid  INT REFERENCES Book (bookid ),  
  orderdate DATE,  
  saleprice INT  
);
```

그림 6-7 물리적 모델링의 예

## 설계 1 단계 : 요구 사항 분석

### ■ 목적

- 사용자의 요구 사항을 수집하고 분석하여 개발할 데이터베이스의 용도를 파악
  - 업무에 필요한 데이터가 무엇인지, 그 데이터에 어떤 처리가 필요한지 등을 고려

### ■ 결과물

- 요구 사항 명세서

### ■ 주요 작업

- 데이터베이스를 실제로 사용할 주요 사용자의 범위를 결정
- 사용자가 조직에서 수행하는 업무를 분석
- 면담, 설문 조사, 업무 관련 문서 분석 등의 방법을 이용해 요구 사항 수집
- 수집된 요구 사항에 대한 분석 결과를 요구 사항 명세서로 작성

- ① 한빛 마트에 회원으로 가입하려면 회원아이디, 비밀번호, 이름, 나이, 직업을 입력해야 한다.
- ② 가입한 회원에게는 등급과 적립금이 부여된다.
- ③ 회원은 회원아이디로 식별한다.
- ④ 상품에 대한 상품번호, 상품명, 재고량, 단가 정보를 유지해야 한다.
- ⑤ 상품은 상품번호로 식별한다.
- ⑥ 회원은 여러 상품을 주문할 수 있고, 하나의 상품을 여러 회원이 주문할 수 있다.
- ⑦ 회원이 상품을 주문하면 주문에 대한 주문번호, 주문수량, 배송지, 주문일자 정보를 유지해야 한다.
- ⑧ 각 상품은 한 제조업체가 공급하고, 제조업체 하나는 여러 상품을 공급할 수 있다.
- ⑨ 제조업체가 상품을 공급하면 공급일자와 공급량 정보를 유지해야 한다.
- ⑩ 제조업체에 대한 제조업체명, 전화번호, 위치, 담당자 정보를 유지해야 한다.
- ⑪ 제조업체는 제조업체명으로 식별한다.
- ⑫ 회원은 게시글을 여러 개 작성할 수 있고, 게시글 하나는 한 명의 회원만 작성할 수 있다.
- ⑬ 게시글에 대한 글번호, 글제목, 글내용, 작성일자 정보를 유지해야 한다.
- ⑭ 게시글은 글번호로 식별한다.

# 설계 2 단계 : 개념적 설계

### ■ 목적

- DBMS에 독립적인 개념적 스키마 설계
- 요구 사항 분석 결과물을 개념적 데이터 모델을 이용해 개념적 구조로 표현
  - ➔ 개념적 모델링
    - 일반적으로 E-R 모델을 많이 이용

### ■ 결과물

- 개념적 스키마 : E-R 다이어그램

### ■ 주요 작업

- 요구 사항 분석 결과를 기반으로 중요한 개체를 추출하고 개체 간의 관계를 결정하여 E-R 다이어그램으로 표현



## ■ 작업 과정

- STEP 1) 개체 추출, 각 개체의 주요 속성과 키 속성 선별
- STEP 2) 개체 간의 관계 결정
- STEP 3) E-R 다이어그램으로 표현

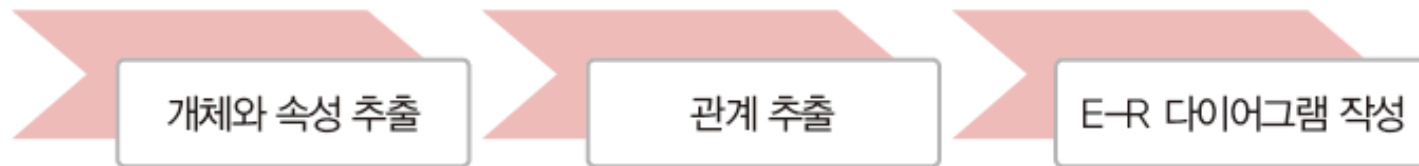


그림 8-6 개념적 모델링 과정

## 개념적 설계 – (STEP 1) 개체와 속성 추출

- 개체 : 저장할 만한 가치가 있는 중요 데이터를 가진 사람이나 사물 등
  - 예) 병원 데이터베이스 개발에 필요한 개체
    - 병원 운영에 필요한 사람 : 환자, 의사, 간호사 등
    - 병원 운영에 필요한 사물 : 병실, 수술실, 의료 장비 등
- 개체 추출 방법
  - 요구 사항 문장에서 업무와 관련이 깊은 의미 있는 **명사**를 찾아라!
    - 업무와 관련이 적은 일반적이고 광범위한 의미의 명사는 제외
    - 의미가 같은 명사가 여러 개일 경우는 대표 명사 하나만 선택
  - 찾아낸 명사를 개체와 속성으로 분류하라!

## (STEP 1) 개체와 속성 추출 예

- ① 한빛 마트에 **회원**으로 가입하려면 회원아이디, 비밀번호, 이름, 나이, 직업을 입력해야 한다.
- ② 가입한 회원에게는 등급과 적립금이 부여된다.
- ③ 회원은 회원아이디로 식별한다.

### [추출 결과]

- 개체 : 회원
- “회원” 개체의 속성 : 회원아이디, 비밀번호, 이름, 나이, 직업, 등급, 적립금
- “회원” 개체의 키 속성 : 회원아이디

## (STEP 1) 개체와 속성 추출 예

- 요구 사항 명세서에서 개체와 속성을 추출한 결과

개체	속성
회원	회원아이디, 비밀번호, 이름, 나이, 직업, 등급, 적립금
상품	상품번호, 상품명, 재고량, 단가
제조업체	제조업체명, 전화번호, 위치, 담당자
게시글	글번호, 글제목, 글내용, 작성일자

그림 8-12 한빛 마트 요구 사항 명세서에서 개체와 개체의 속성을 추출한 최종 결과

## 개념적 설계 (STEP 1) 개체와 속성 추출

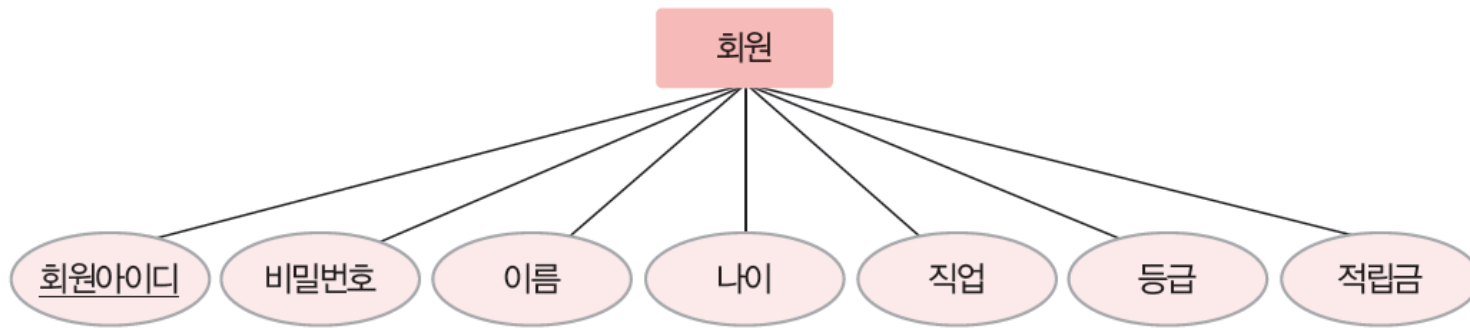


그림 8-13 회원 개체의 E-R 다이어그램

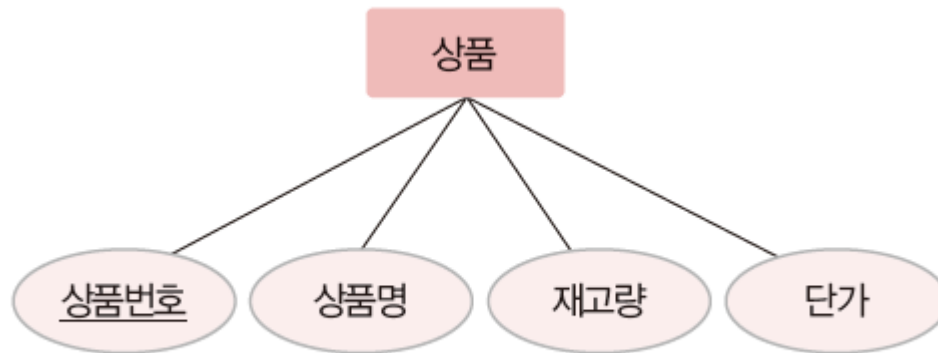


그림 8-14 상품 개체의 E-R 다이어그램

## 개념적 설계 (STEP 1) 개체와 속성 추출

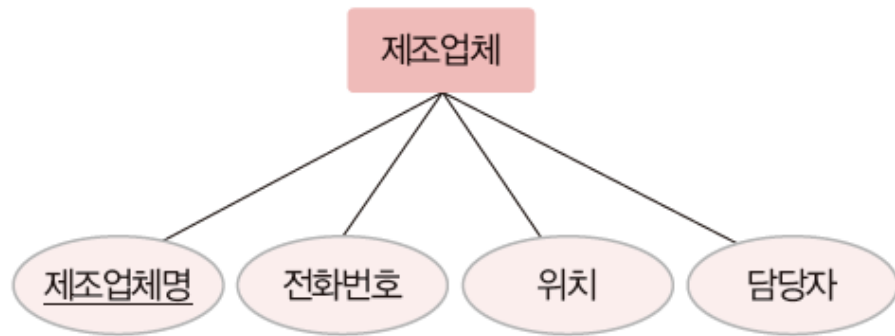


그림 8-15 제조업체 개체의 E-R 다이어그램

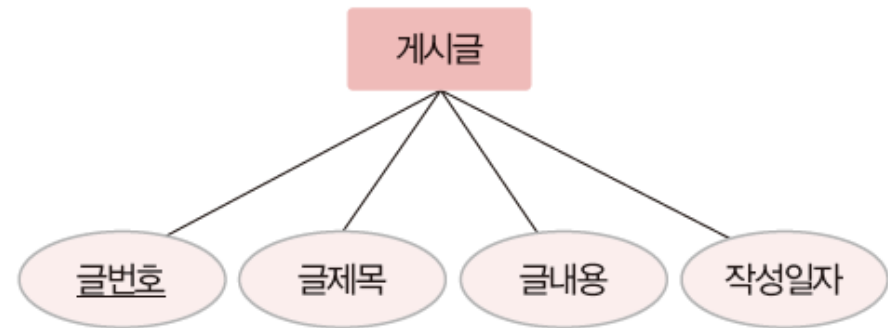


그림 8-16 게시글 개체의 E-R 다이어그램

## 개념적 설계 (STEP 2) 관계 추출

### ■ 관계 추출 방법

- 요구 사항 문장에서 개체 간의 연관성을 의미 있게 표현한 **동사**를 찾아라!
  - 의미가 같은 동사가 여러 개일 경우는 대표 동사 하나만 선택
- 찾아낸 관계에 대해 매핑 카디널리티와 참여 특성을 결정하라!
  - 매핑 카디널리티 : 일대일(1:1), 일대다(1:n), 다대다(n:m)
  - 참여 특성 : 필수적 참여 / 선택적 참여

## (STEP 2) 관계 추출 예

- ⑥ 회원은 여러 상품을 주문할 수 있고, 하나의 상품을 여러 회원이 주문할 수 있다.
- ⑦ 회원이 상품을 주문하면 주문에 대한 주문번호, 주문수량, 배송지, 주문일자 정보를 유지해야 한다.

### [추출 결과]

- 관계: 주문
  - “회원” 개체와 “상품” 개체가 맺는 관계, 다대다(n:m) 관계
  - “회원” 개체는 관계에 선택적으로 참여 / “상품” 개체는 관계에 선택적으로 참여
- “주문” 관계의 속성 : 주문번호, 주문수량, 배송지, 주문일자



## (STEP 2) 관계 추출 예

- 요구 사항 명세서에서 관계를 추출한 결과

관계 ↓	관계에 참여하는 개체 ↓	관계 유형 ↓	속성 ↓
주문	회원(선택) 상품(선택)	다대다	주문번호, 주문수량, 배송지, 주문일자
공급	상품(필수) 제조업체(선택)	일대다	공급일자, 공급량
작성	회원(선택) 게시글(필수)	일대다	

그림 8-22 한빛 마트 요구 사항 명세서에서 관계와 관계의 속성을 추출한 최종 결과

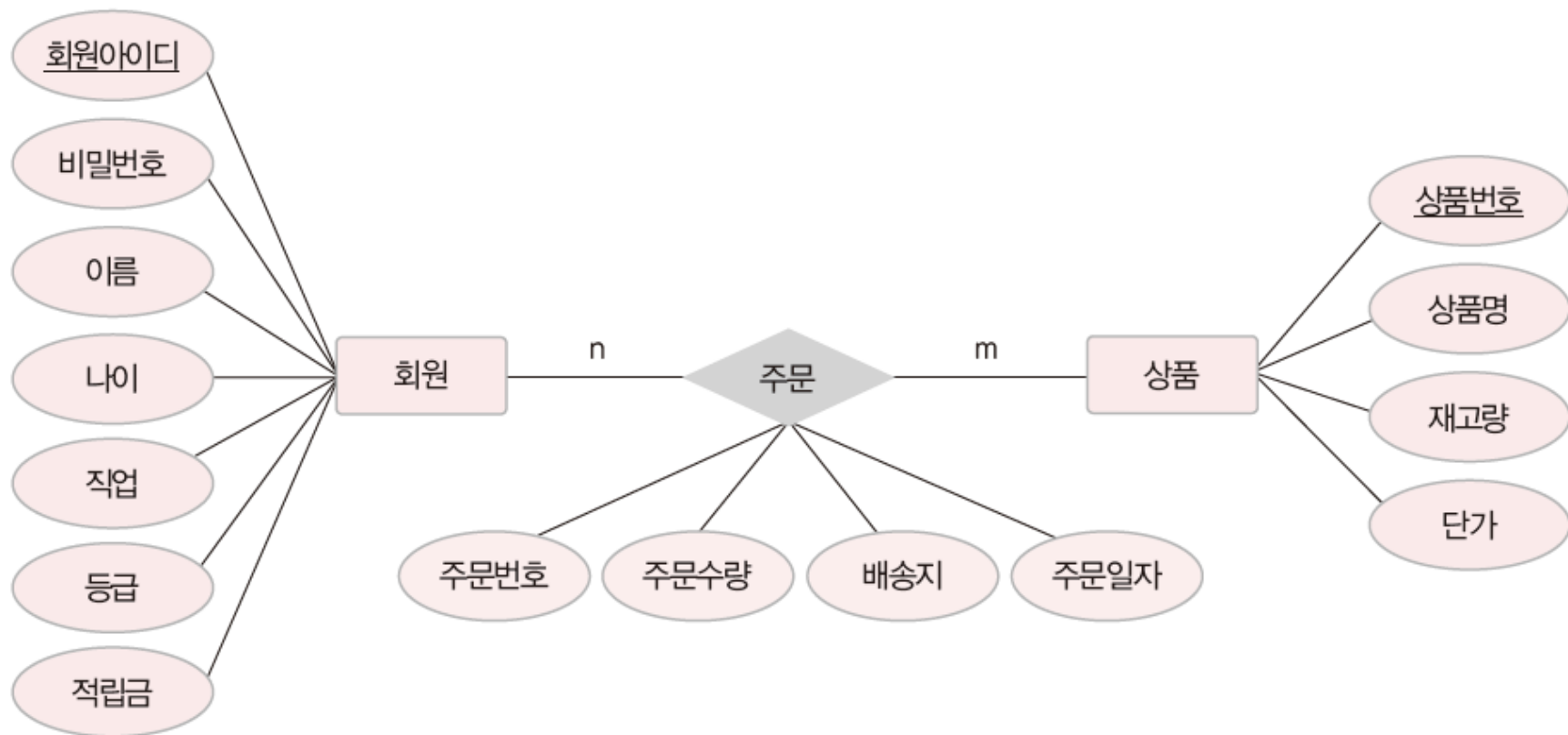


그림 8-23 주문 관계의 E-R 다이어그램

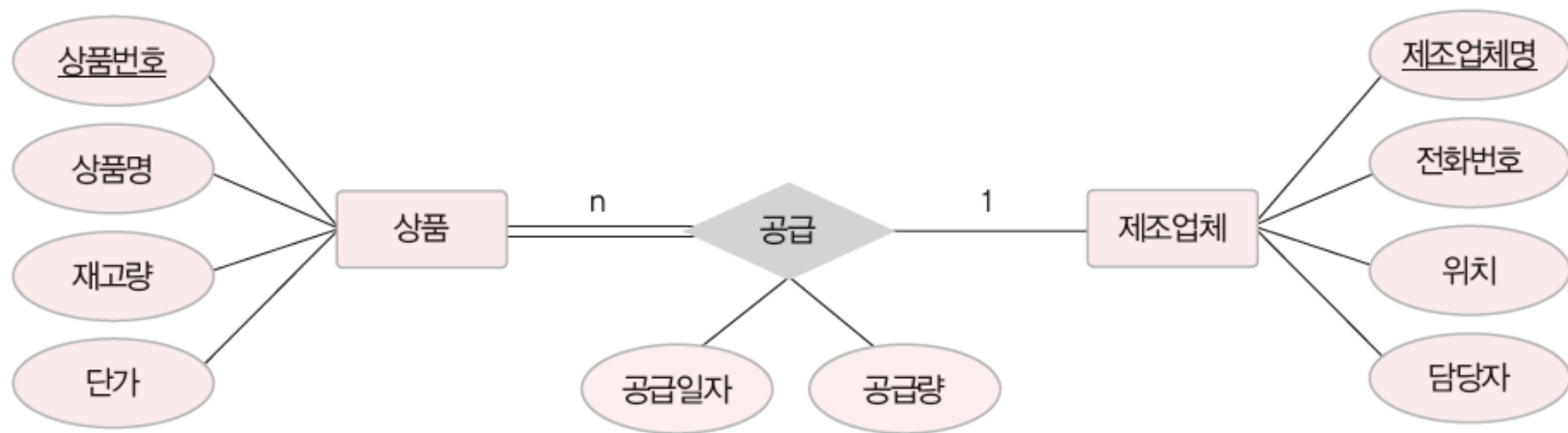


그림 8-24 공급 관계의 E-R 다이어그램

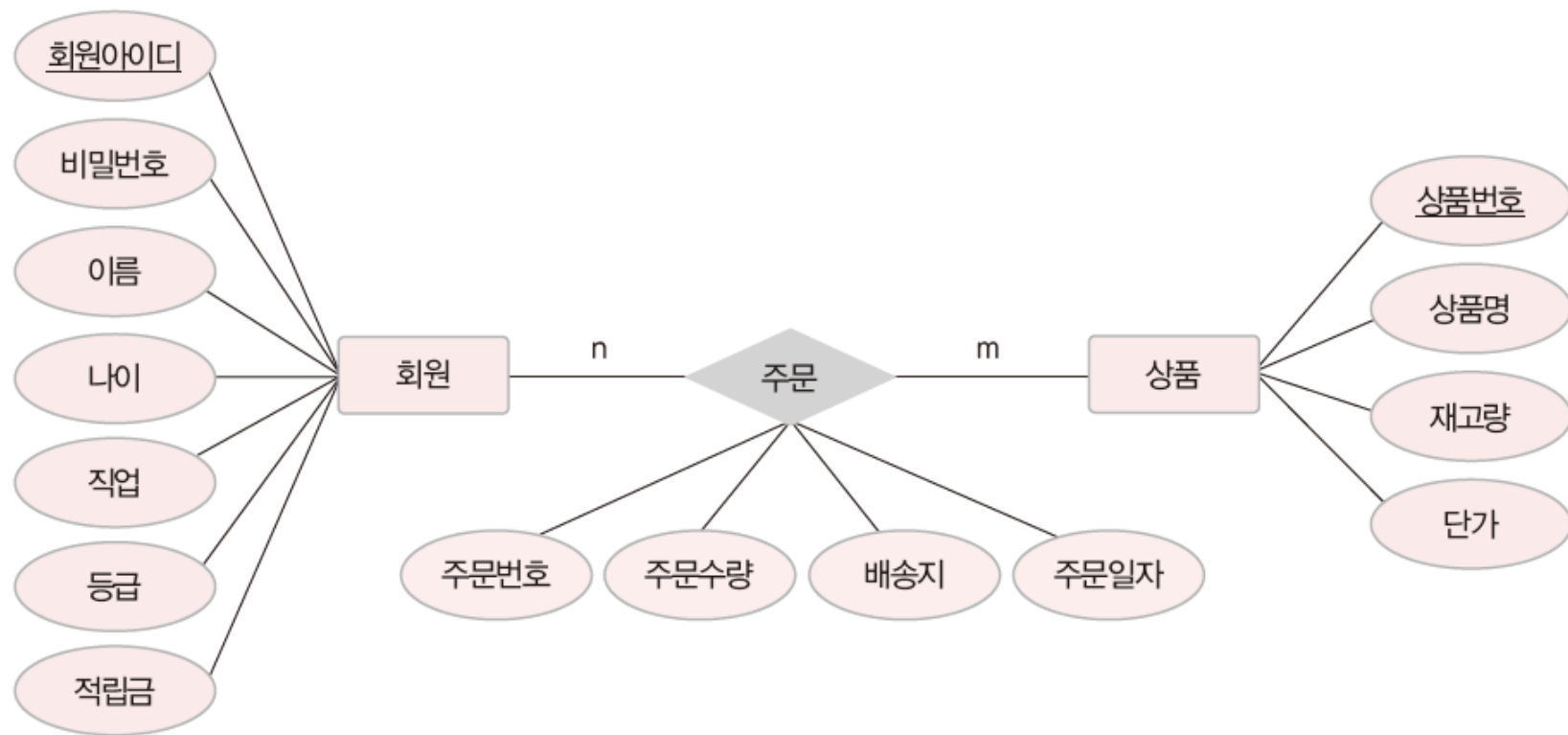


그림 8-23 주문 관계의 E-R 다이어그램

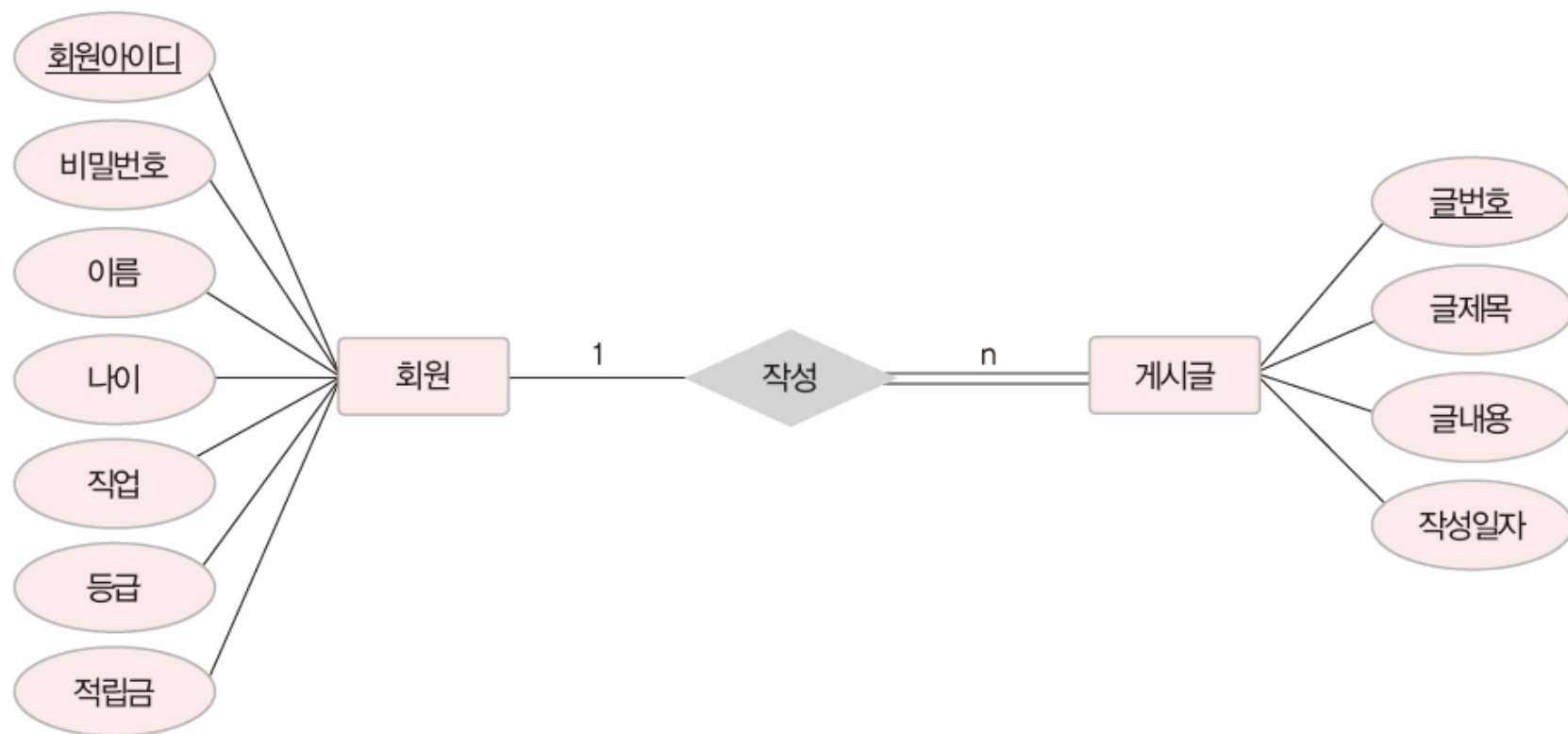
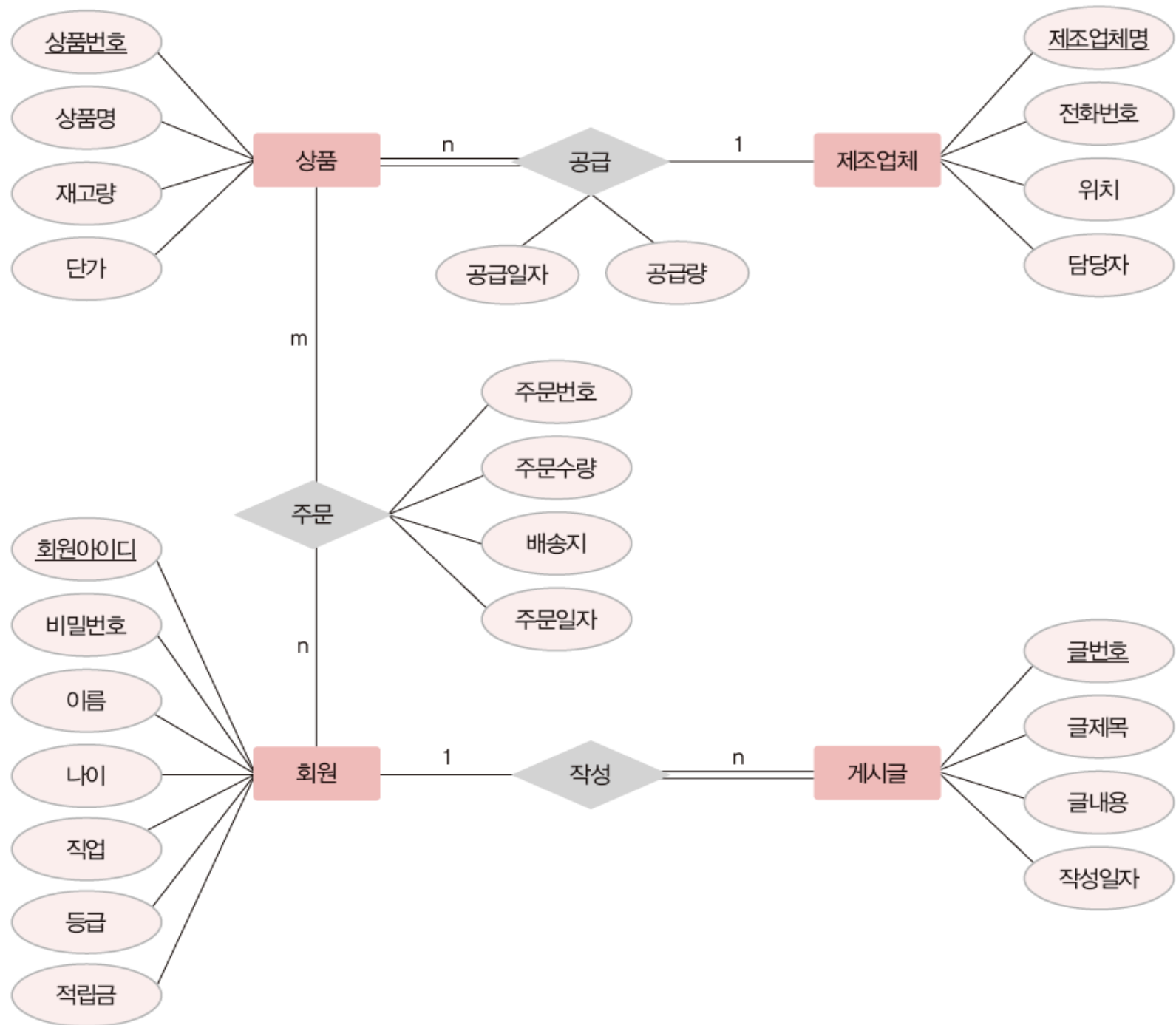


그림 8-25 작성 관계의 E-R 다이어그램



## 설계 3 단계 : 논리적 설계

### ■ 목적

- DBMS에 적합한 논리적 스키마 설계
- 개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현
  - ➔ 논리적 모델링(데이터 모델링)
  - 일반적으로 관계 데이터 모델을 많이 이용

### ■ 결과물

- 논리적 스키마 : 릴레이션 스키마

### ■ 주요 작업

- 개념적 설계 단계의 결과물인 E-R 다이어그램을 릴레이션 스키마로 변환
- 릴레이션 스키마로 변환 후 속성의 데이터 타입, 길이, 널 값 허용 여부, 기본 값, 제약조건 등을 세부적으로 결정하고 결과를 문서화시킴

- E-R 다이어그램을 릴레이션 스키마로 변환하는 규칙
  - 규칙 1 : 모든 개체는 릴레이션으로 변환한다.
  - 규칙 2 : 다대다( $n:m$ ) 관계는 릴레이션으로 변환한다.
  - 규칙 3 : 일대다( $1:n$ ) 관계는 외래키로 표현한다.
  - 규칙 4 : 일대일( $1:1$ ) 관계는 외래키로 표현한다.
  - 규칙 5 : 다중 값 속성은 릴레이션으로 변환한다.
- 변환 규칙을 순서대로 적용하되, 해당되지 않는 규칙은 제외함



## 규칙 1) 모든 개체는 릴레이션으로 변환한다

E-R  
다이아그램



논리적 설계

릴레이션  
스키마

상품 릴레이션

상품번호	상품명	재고량	단가
p01	그냥만두	5000	4500
p02	매운짬면	2500	5500
p03	콩떡파이	3600	2600
p04	맛난초콜릿	1250	2500
p05	얼큰라면	2200	1200

← 릴레이션 스키마

그림 8-28 개체를 릴레이션으로 변환하는 규칙을 적용한 예

■ 복합속성인 경우

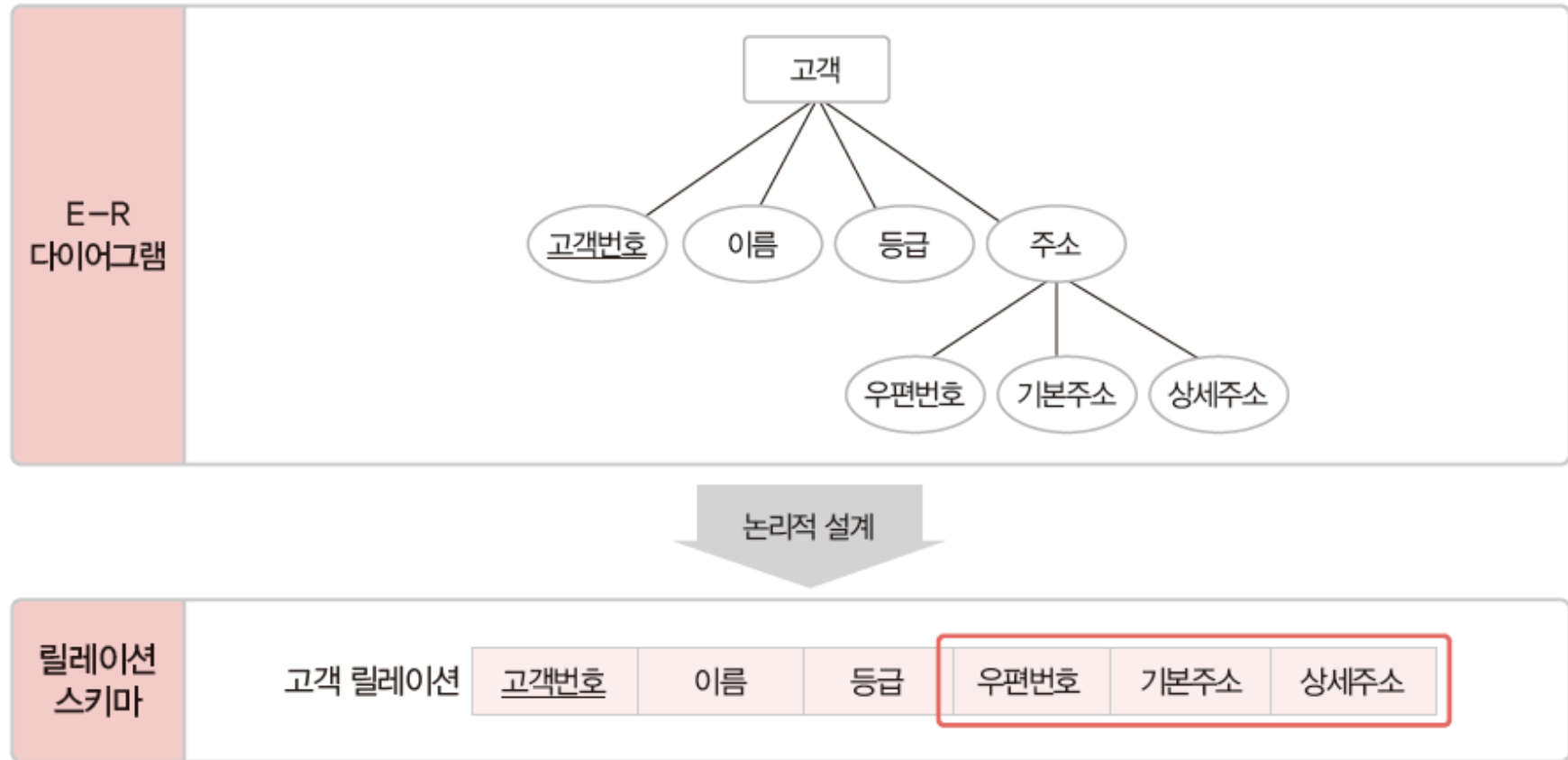
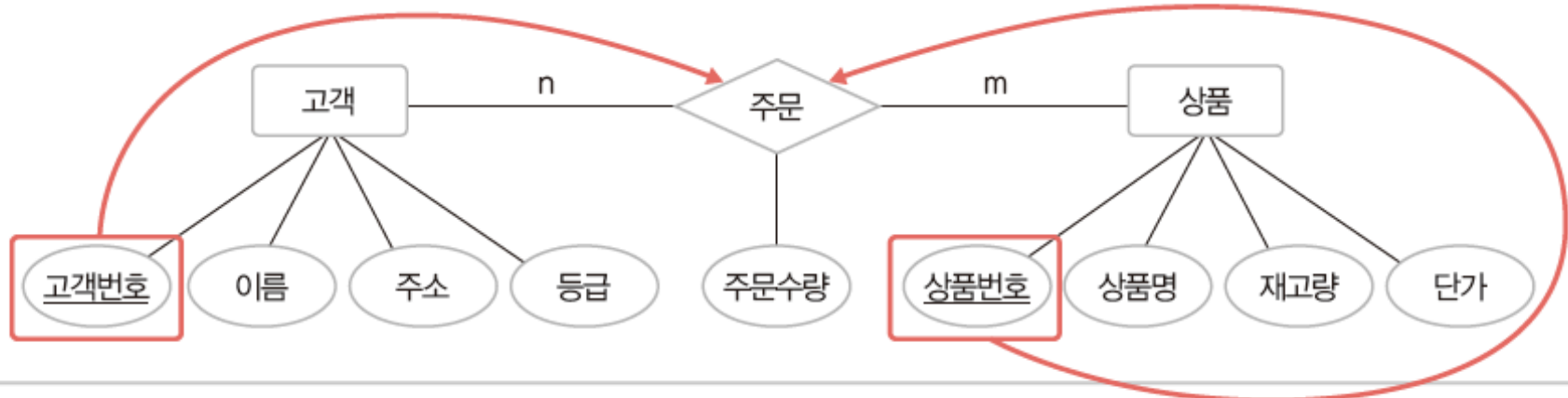


그림 8-29 복합 속성을 가지는 개체를 릴레이션으로 변환하는 예

## 규칙 2) 다대다 관계는 릴레이션으로 변환한다

- 다대다 관계를 하나의 릴레이션으로 변환
  - 관계의 이름 → 릴레이션 이름
  - 관계의 속성 → 릴레이션의 속성
  - 관계에 참여하는 개체를 규칙 1에 따라 릴레이션으로 변환한 후 이 릴레이션의 기본키를 관계 릴레이션에 포함시켜 외래키로 지정하거나, 외래키들을 조합하여 관계 릴레이션의 기본키로 지정

E-R  
다이어그램



논리적 설계

릴레이션  
스키마

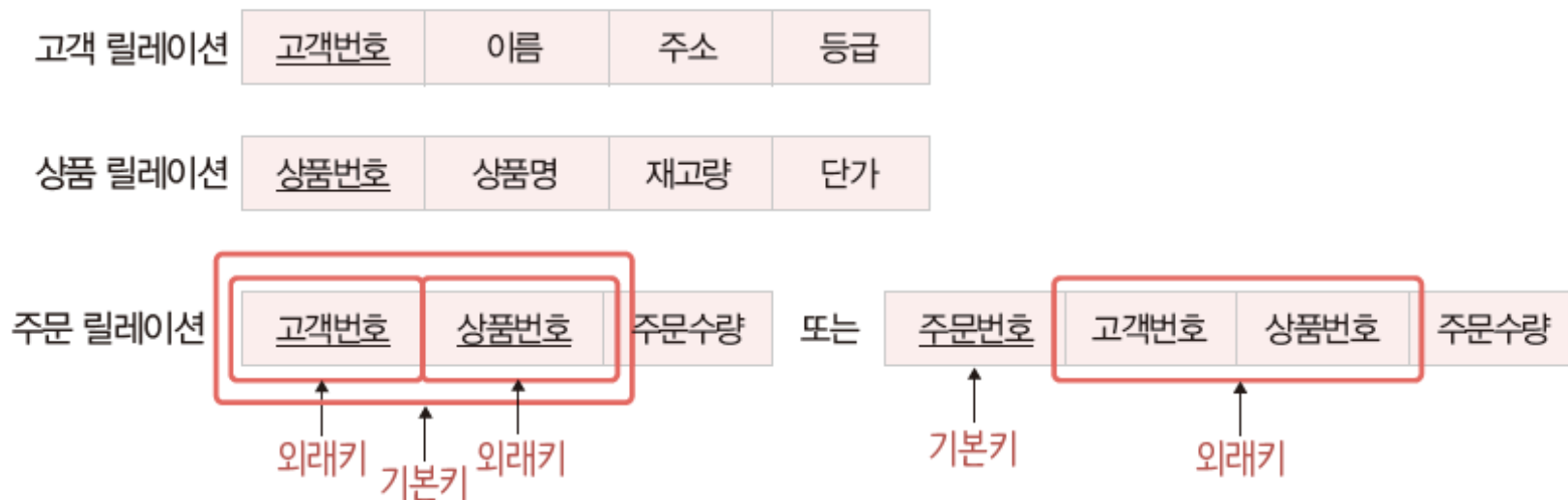


그림 8-30 다대다 관계를 릴레이션으로 변환하는 규칙을 적용한 예

## 규칙 3) 일대다 관계는 외래키로 표현한다

- 3-1) 일반적인 일대다 관계는 외래키로 표현한다.
  - 일대다(1:n) 관계에서 1측 개체 릴레이션의 기본키를  
n측 개체 릴레이션에 포함시켜 외래키로 지정
  - 관계의 속성들도 n측 개체 릴레이션에 포함시킴(다음 그림의 '공급수량'속성)
- 3-2) 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키로 지정한다.
  - 약한 개체: 두 개체가 있을 때 독자적으로 존재할 수 없고, 종속되는 개체를 약한 개체라고 한다.  
예를 들어, 고객과 계좌

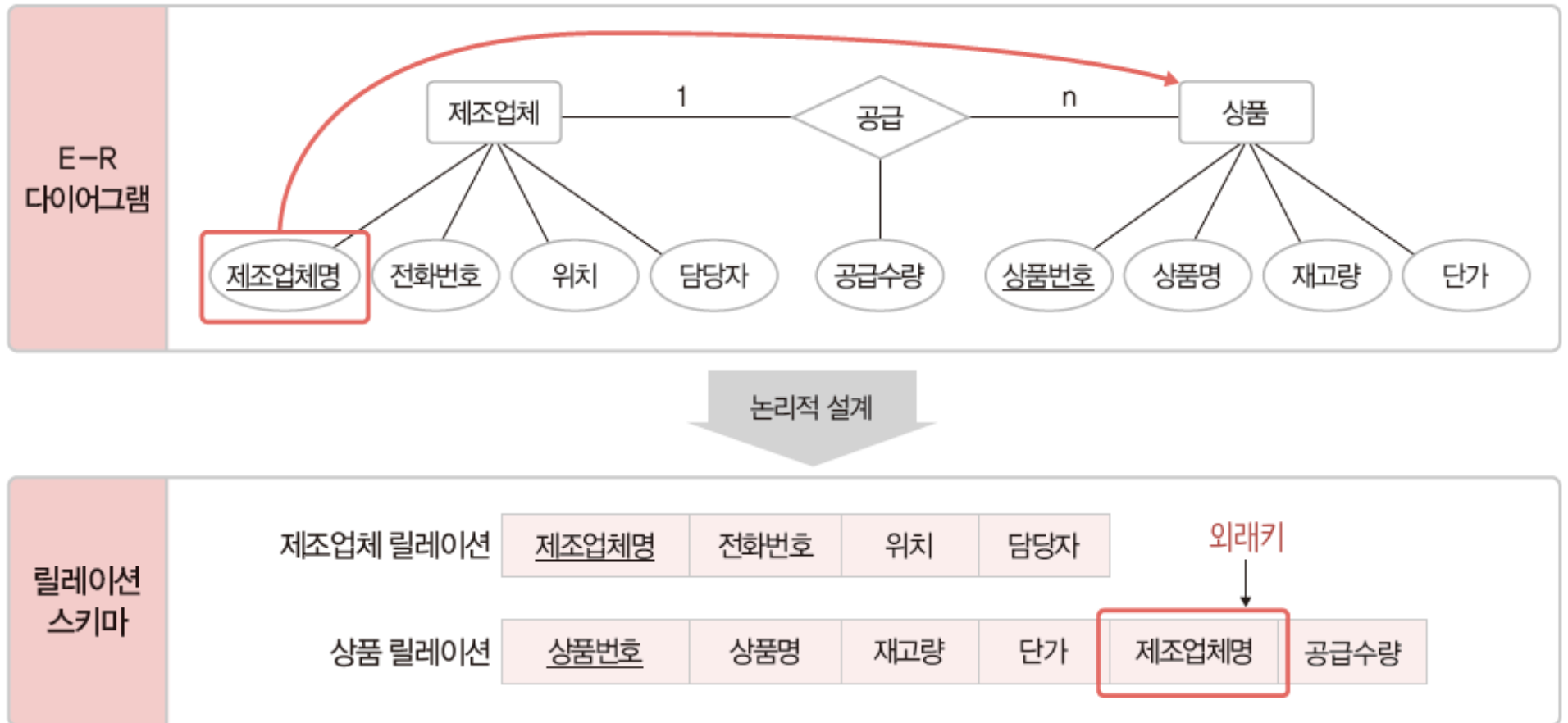


그림 8-31 일반적인 개체가 참여하는 일대다 관계를 외래키로 표현하는 규칙을 적용한 예

- 약한 개체는 강한 개체에 따라 존재 여부가 결정되므로 강한 개체의 기본 키를 이용해 식별해야 함

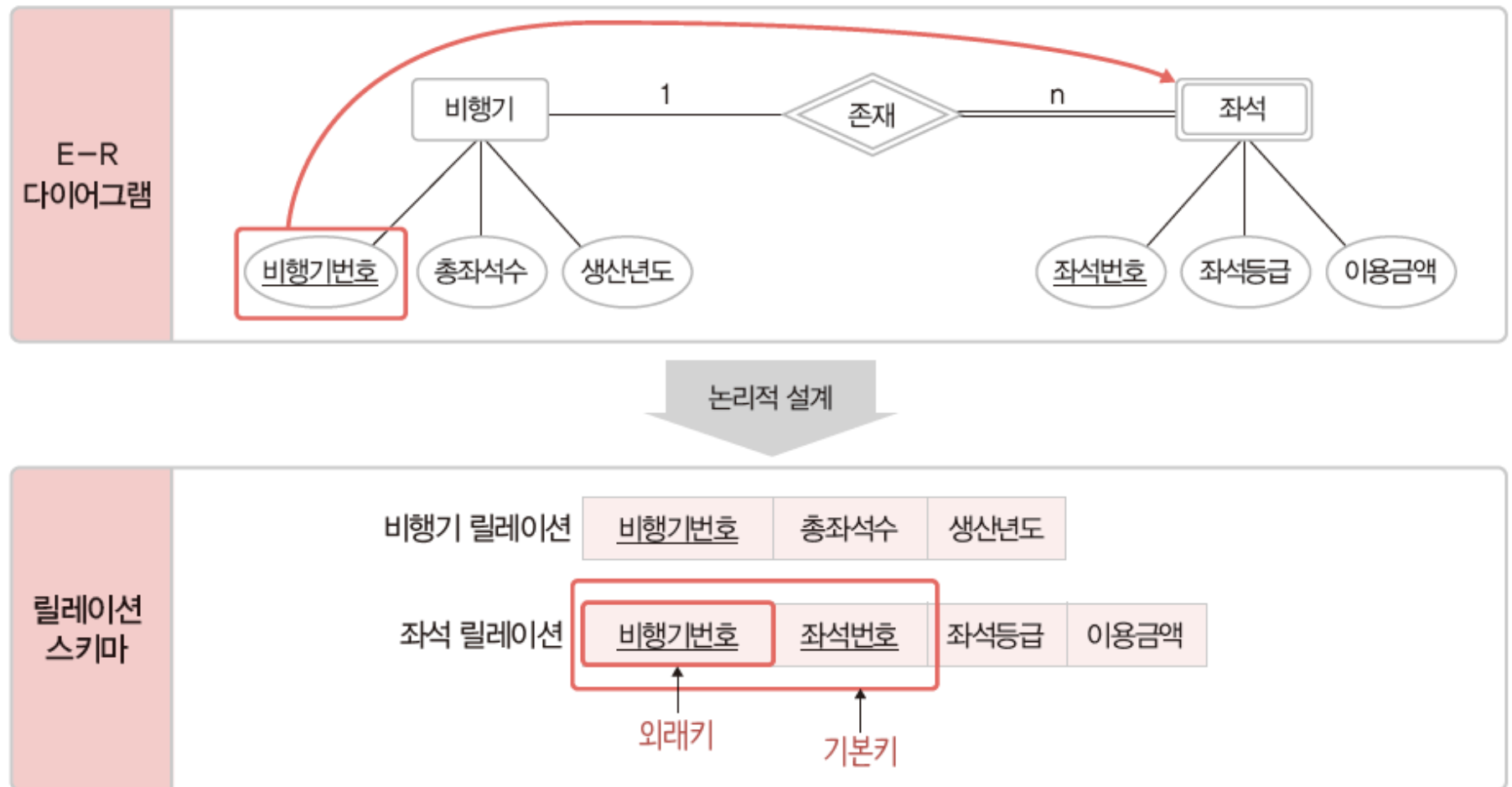


그림 8-32 약한 개체가 참여하는 일대다 관계를 외래키로 표현하는 규칙을 적용한 예

## 규칙 4) 일대일 관계는 외래키로 표현한다

- 4-1) 일반적인 일대일 관계는 외래키를 서로 주고받는다.
- 4-2) 일대일 관계에 필수적으로 참여하는 개체의 릴레이션만 외래키를 받는다.
- 4-3) 모든 개체가 일대일 관계에 필수적으로 참여하면 릴레이션 하나로 합친다.



- 4-1) 일반적인 일대일 관계는 외래키를 서로 주고받는다.
  - 관계에 참여하는 개체 릴레이션들이 서로의 기본키를 주고받아 외래키로 지정
  - 관계의 속성들도 모든 개체 릴레이션에 포함시킴
  - 불필요한 데이터 중복이 발생할 수 있음

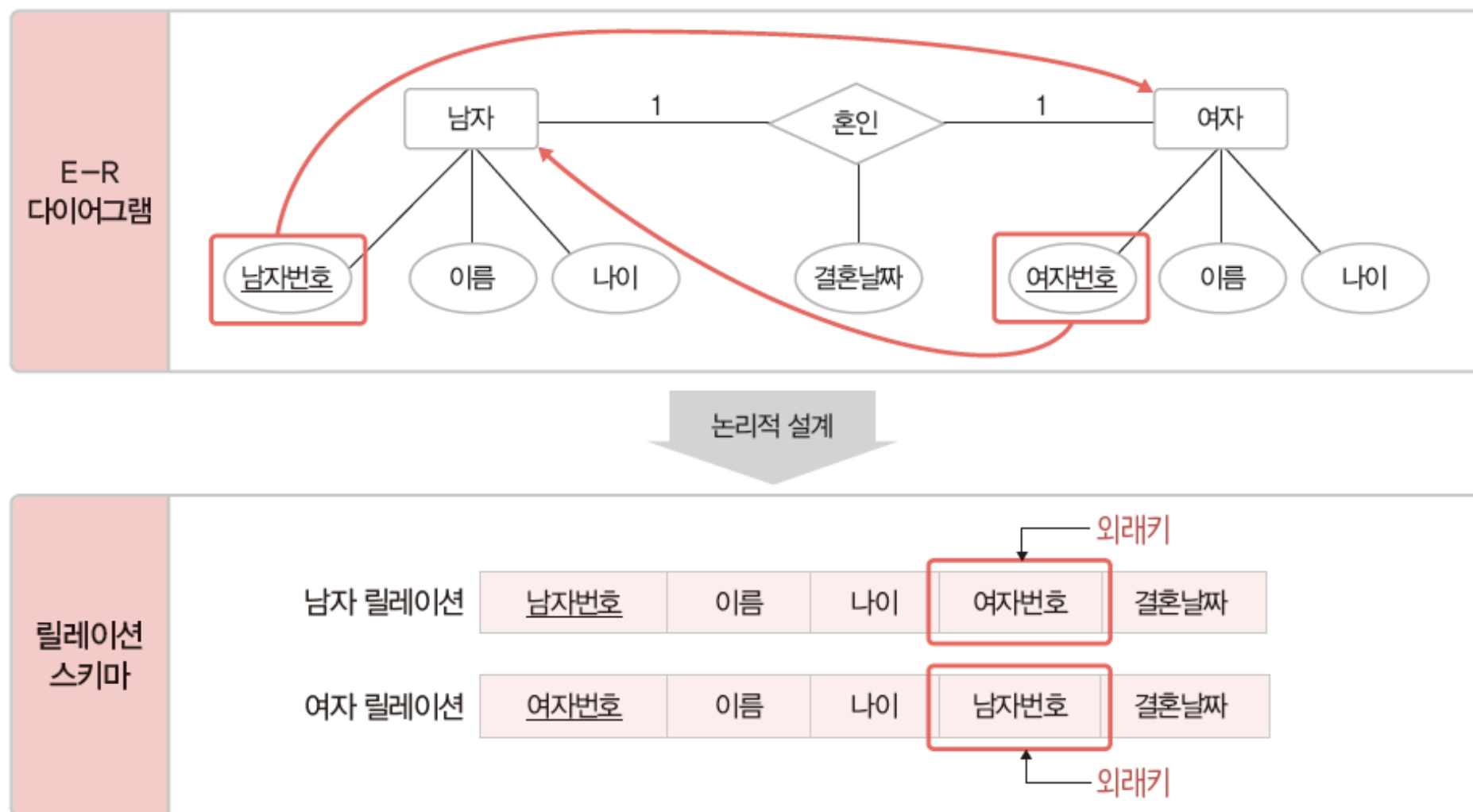


그림 8-33 일반적인 일대일 관계를 외래키로 표현하는 규칙을 적용한 예

- 4-2) 필수적으로 참여하는 개체 릴레이션만 외래키를 받는다.
  - 관계에 필수적으로 참여하는 개체 릴레이션에만 외래키를 포함시킴
  - 관계의 속성들은 관계에 필수적으로 참여하는 개체 릴레이션에 포함시킴

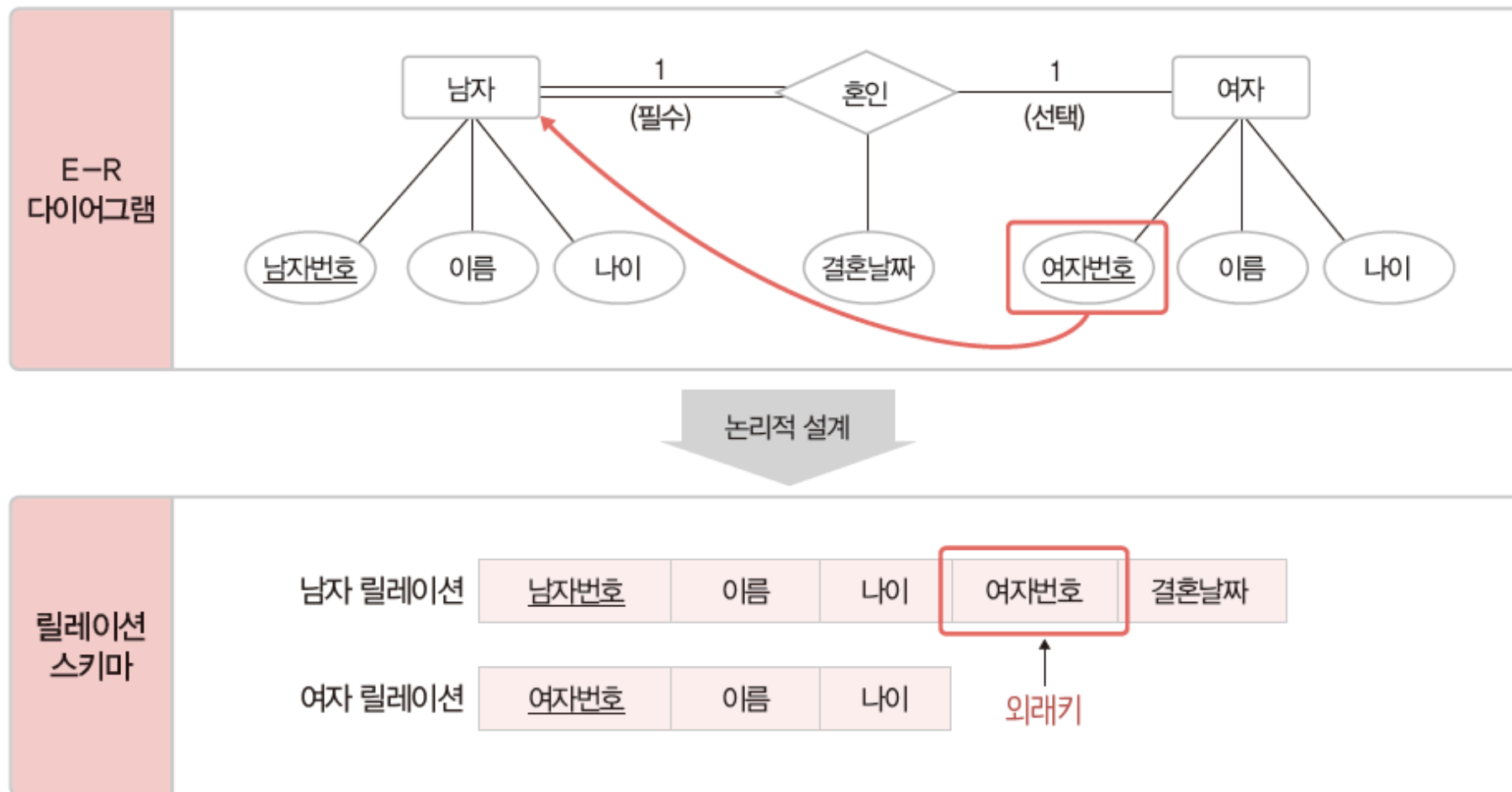


그림 8-34 일대일 관계에 필수적으로 참여하는 개체의 릴레이션이 외래키를 가지는 예

- 4-3) 모든 개체가 필수적으로 참여하면 릴레이션 하나로 합친다.
  - 관계에 참여하는 개체 릴레이션들을 하나의 릴레이션으로 합쳐서 표현
  - 관계의 이름을 릴레이션 이름으로 사용하고, 관계에 참여하는 두 개체의 속성들을 관계 릴레이션에 모두 포함시킴
  - 두 개체 릴레이션의 키 속성을 조합하여 관계 릴레이션의 기본키로 지정

E-R  
다이어그램



논리적 설계

릴레이션  
스키마

혼인 릴레이션

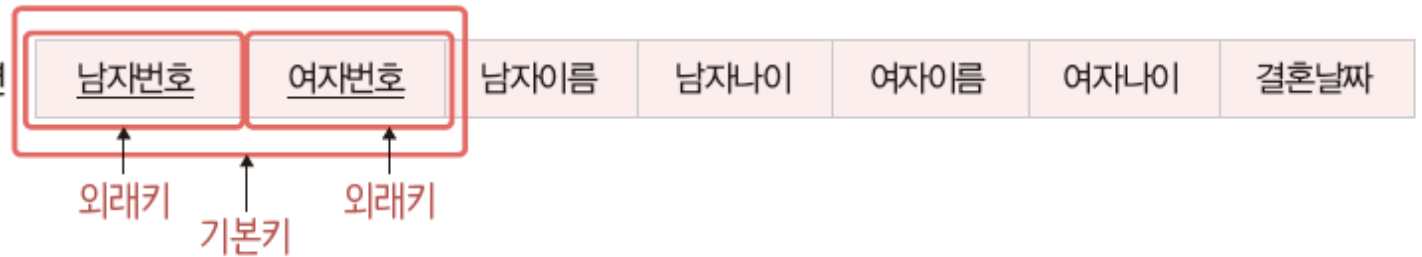


그림 8-35 일대일 관계에 모든 개체가 필수적으로 참여하면 릴레이션을 통합하는 예

## 규칙 5) 다중 값 속성은 릴레이션으로 변환한다

- 다중 값 속성은 독립적인 릴레이션으로 변환
  - 다중 값 속성과 함께 그 속성을 가지고 있던 개체 릴레이션의 기본키를 외래키로 가져와 새로운 릴레이션에 포함시킴
  - 새로운 릴레이션의 기본키는 다중 값 속성과 외래키를 조합하여 지정

<u>사원번호</u>	사원명	직위	부하직원
e001	홍정화	부장	{김정수, 이수연}
e002	김수창	과장	{박영길}
e003	최종민	차장	{이수영, 배길수}

← 다중 값을 가지는 속성

그림 8-37 다중 값 속성인 부하직원 속성을 그대로 포함하는 사원 릴레이션

사원 릴레이션은 "속성에 다중 값을 저장할 수 없다"는 릴레이션 특성을 위반함



<u>사원번호</u>	<u>사원명</u>	<u>직위</u>	<u>부하직원</u>
e001	홍정화	부장	김정수
e001	홍정화	부장	이수연
e002	김수창	과장	박영길
e003	최종민	차장	이수영
e003	최종민	차장	배길수

그림 8-38 릴레이션 특성에 맞게 부하직원 속성을 포함하는 사원 릴레이션

사원 릴레이션은 릴레이션 특성을 위반하지는 않지만  
사원번호, 사원명, 직위 속성의 값이 불필요하게 중복 저장되는 문제가 발생함

사원 릴레이션

사원번호	사원명	직위
e001	홍정화	부장
e002	김수창	과장
e003	최종민	차장

사원-부하직원 릴레이션

사원번호	부하직원
e001	김정수
e001	이수연
e002	박영길
e003	이수영
e003	배길수

그림 8-39 규칙 5를 적용한 후의 사원 릴레이션과 사원-부하직원 릴레이션

(규칙 5)에 따라 다중 값 속성을 독립적인 릴레이션으로 변환하면 불필요한 중복을 제거하면서도 릴레이션의 특성을 만족시킬 수 있다.

E-R  
다이아그램



논리적 설계

릴레이션  
스키마



그림 8-36 다중 값 속성을 릴레이션으로 변환하는 규칙을 적용한 예

## 논리적 설계 최종 결과

회원 릴레이션	<u>회원아이디</u>	비밀번호	이름	나이	직업	등급	적립금
상품 릴레이션	<u>상품번호</u>	상품명	재고량	단가	제조업체명	공급일자	공급량
제조업체 릴레이션	<u>제조업체명</u>	전화번호	위치	담당자			
게시글 릴레이션	<u>글번호</u>	글제목	글내용	작성일자	회원아이디		
주문 릴레이션	<u>회원아이디</u>	<u>상품번호</u>	주문번호	주문수량	배송지	주문일자	

외래키

← 외래키

그림 8-47 규칙 3을 적용한 결과

## 테이블 명세서 작성

- 릴레이션 스키마 변환 후 속성의 데이터 타입과 길이, 널 값 허용 여부, 기본값, 제약조건 등을 세부적으로 결정하고 문서화시킴
- 테이블 명세서 : 릴레이션 스키마에 대한 설계 정보를 기술한 문서

테이블 이름			회원			
속성 이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
회원아이디	VARCHAR(20)	N		PK		
비밀번호	VARCHAR(20)	N				
이름	VARCHAR(10)	N				
나이	INT	Y				0이상
직업	VARCHAR(20)	Y				
등급	VARCHAR(10)	N	silver			silver, gold, vip만 허용
적립금	INT	N	0			

그림 8-48 회원 릴레이션 스키마의 테이블 명세서

## 설계 4 단계 : 물리적 설계

- 하드웨어나 운영체제의 특성을 고려하여 필요한 인덱스 구조나 내부 저장 구조 등에 대한 물리적인 구조를 설계

## 설계 5 단계 : 구현

- SQL로 작성한 명령문을 DBMS에서 실행하여 데이터베이스를 실제로 생성

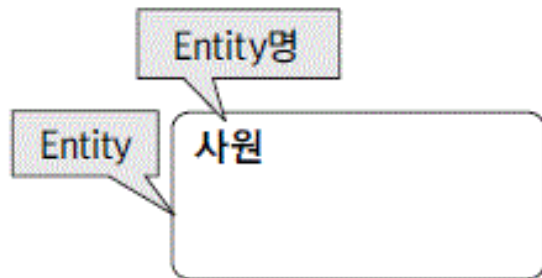
- 테이블 명세서에 따라 SQL 문을 작성한 예

```
CREATE TABLE 회원 (  
    회원아이디    VARCHAR(20)        NOT NULL,  
    비밀번호      VARCHAR(20)        NOT NULL,  
    이름          VARCHAR(10)        NOT NULL,  
    나이          INT,  
    직업          VARCHAR(20),  
    등급          VARCHAR(10)        NOT NULL  DEFAULT 'silver',  
    적립금        INT                NOT NULL  DEFAULT 0,  
    PRIMARY KEY(회원아이디),  
    CHECK (나이 >= 0),  
    CHECK (등급 in ('silver', 'gold', 'vip'))  
);
```

그림 8-50 회원 릴레이션을 구현하기 위한 SQL 문의 예

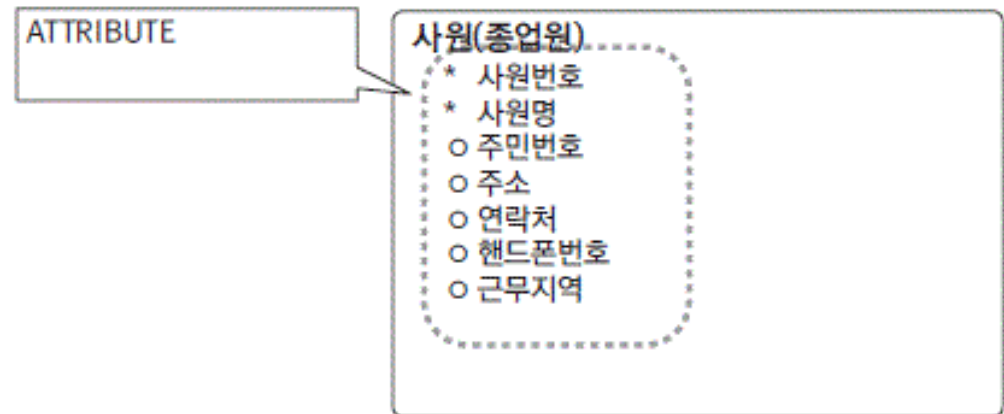
## Barker 표기법

- 엔터티



[그림 4-1-30] 바커 엔터티(Entity) 예

- 속성

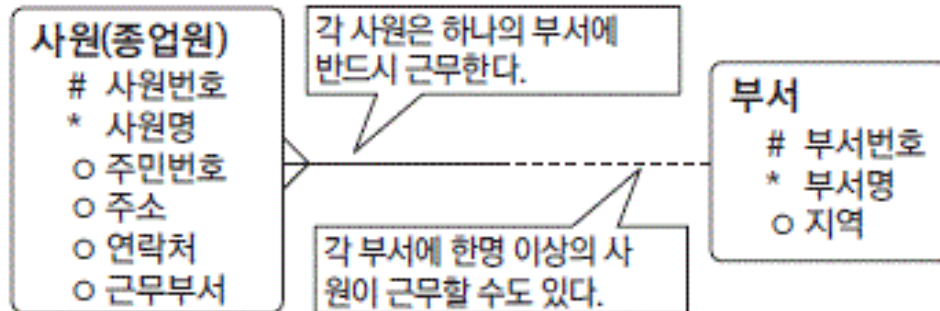


[그림 4-1-31] 바커 속성(Attribute) 예

- # : Primary key, \*: mandatory o: Optional

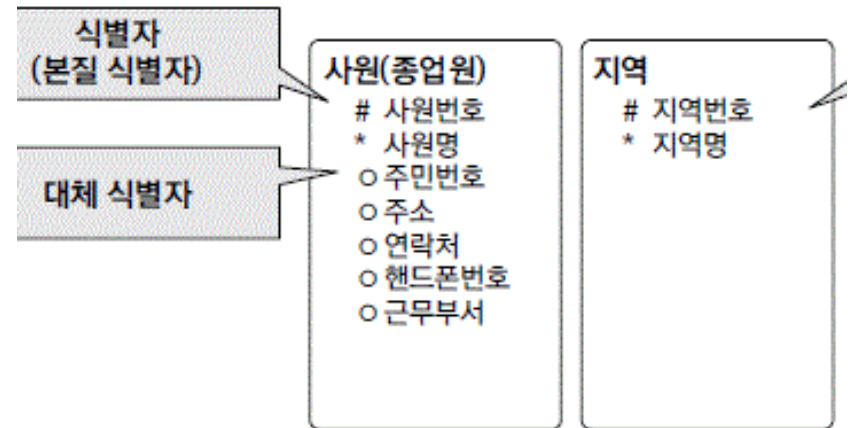


- 관계



[그림 4-1-32] Barker Conditional 작성 단계 예

- 식별자(=Key)



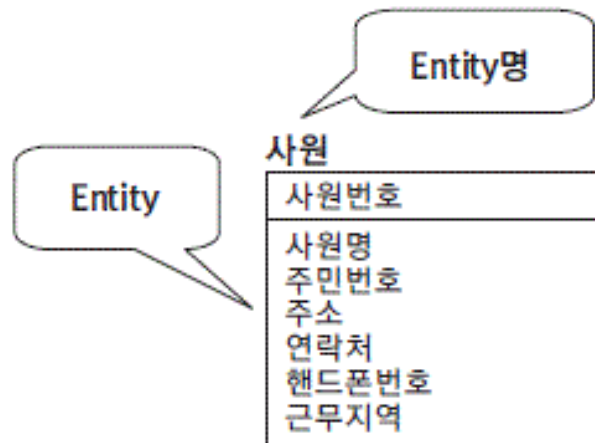
[그림 4-1-34] 바커 식별자(Unique identifier)



[그림 4-1-33] 바커 관계 예

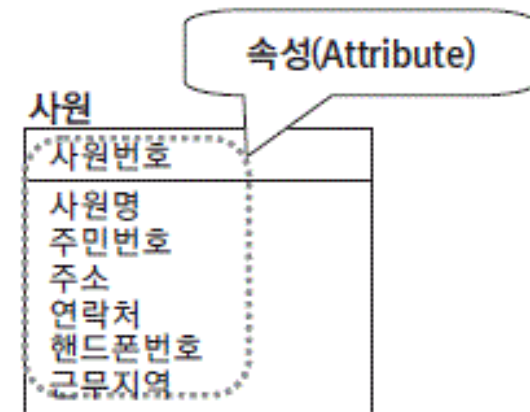
## IE(Information Engineering) 표기법: Crow's foot

- 엔터티



[그림 4-1-37] I/E 엔터티(Entity) 예

- 속성



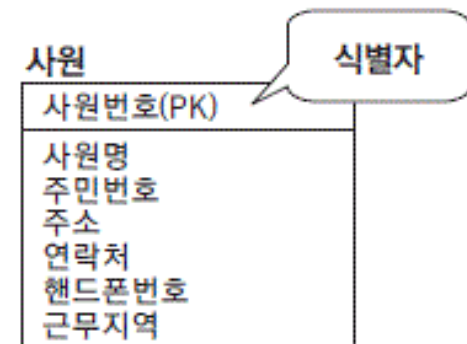
[그림 4-1-38] I/E 속성(Attribute)의 예

- 관계

[표 4-1-3] I/E 관계(Relationship) 예

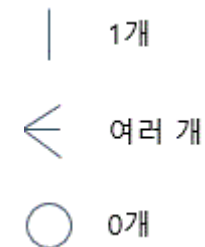
Identifying	Non-identifying	의미
		0,1, 또는 그 이상의 개체 허용
		1 또는 그 이상의 개체 허용
		0 또는 1 개체 허용
		정확히 1 개체 허용

- 식별자(=Key)

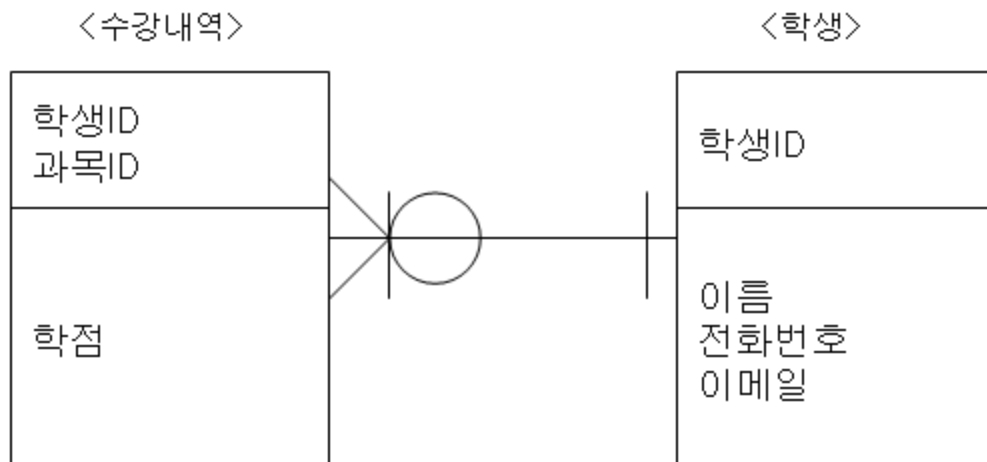


[그림 4-1-39] I/E 식별자(Unique Identifier) 예

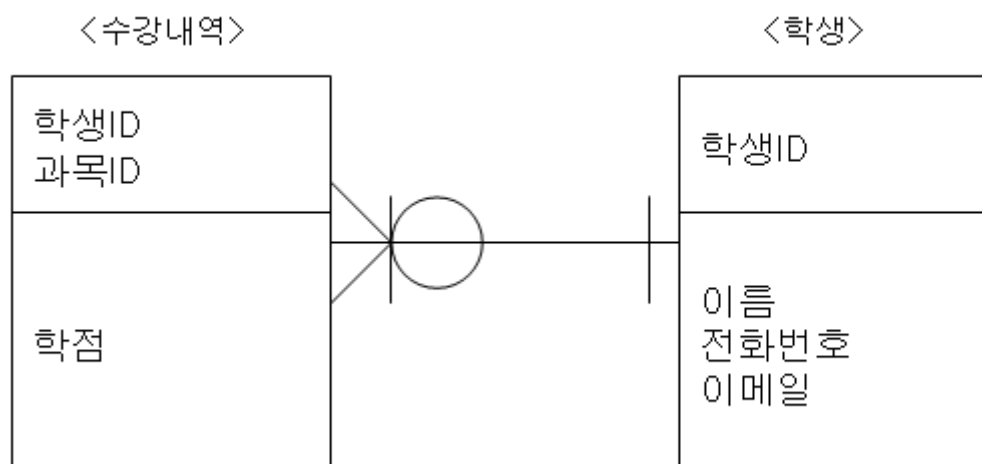
관계	선택성	IE 표기법	Barker 표기법
1 : 1	필수		
1 : 1	선택		
1 : N	필수		
1 : N	선택		



## 표기법 읽기 연습



## 표기법 읽기 연습



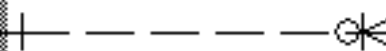
- 수강내역 테이블은 학생 테이블의 PK인 [ 학생ID ]를 FK로 가진다.
- 부모 테이블은 학생 테이블이다.
- 자식 테이블은 수강내역 테이블이다.
- 부모 테이블의 PK를 자식 테이블에서 PK로 사용하고 있다. (식별관계)
- 학생 한 명은 0~N 개의 수강내역을 가진다.
- 수강내역은 하나의 학생을 가진다.

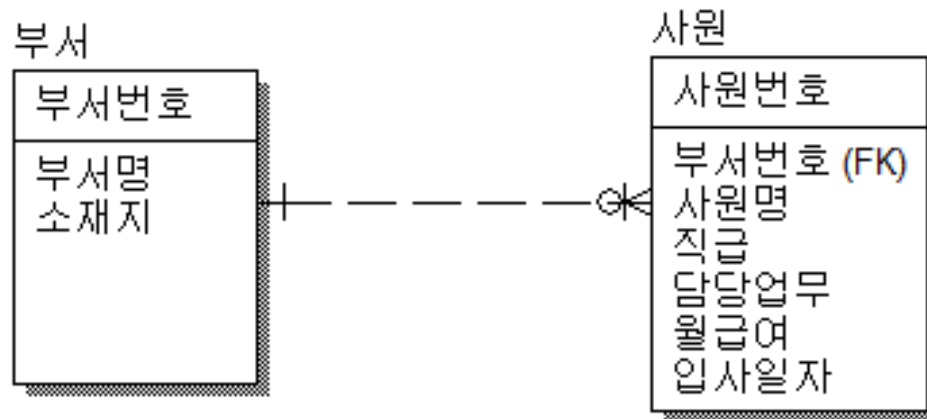
부서

부서번호
부서명
소재지

사원

사원번호
부서번호 (FK)
사원명
직급
담당업무
월급여
입사일자





- 사원 테이블은 부서 테이블의 PK인 부서번호를 FK로 가진다.
- 부모 테이블은 부서 테이블이다.
- 자식 테이블은 사원 테이블이다.
- 자식 테이블이 부모 테이블의 PK를 가지고 있지만 이를 PK로 사용하지는 않는다.(비식별 관계)
- 하나의 부서는 0~N 명의 사원을 가질 수 있다.

## 데이터베이스 설계 도구들

- erwin
- exerd
- erdcloud
- mysql workbench
- oracle-develper
- AQueryTool