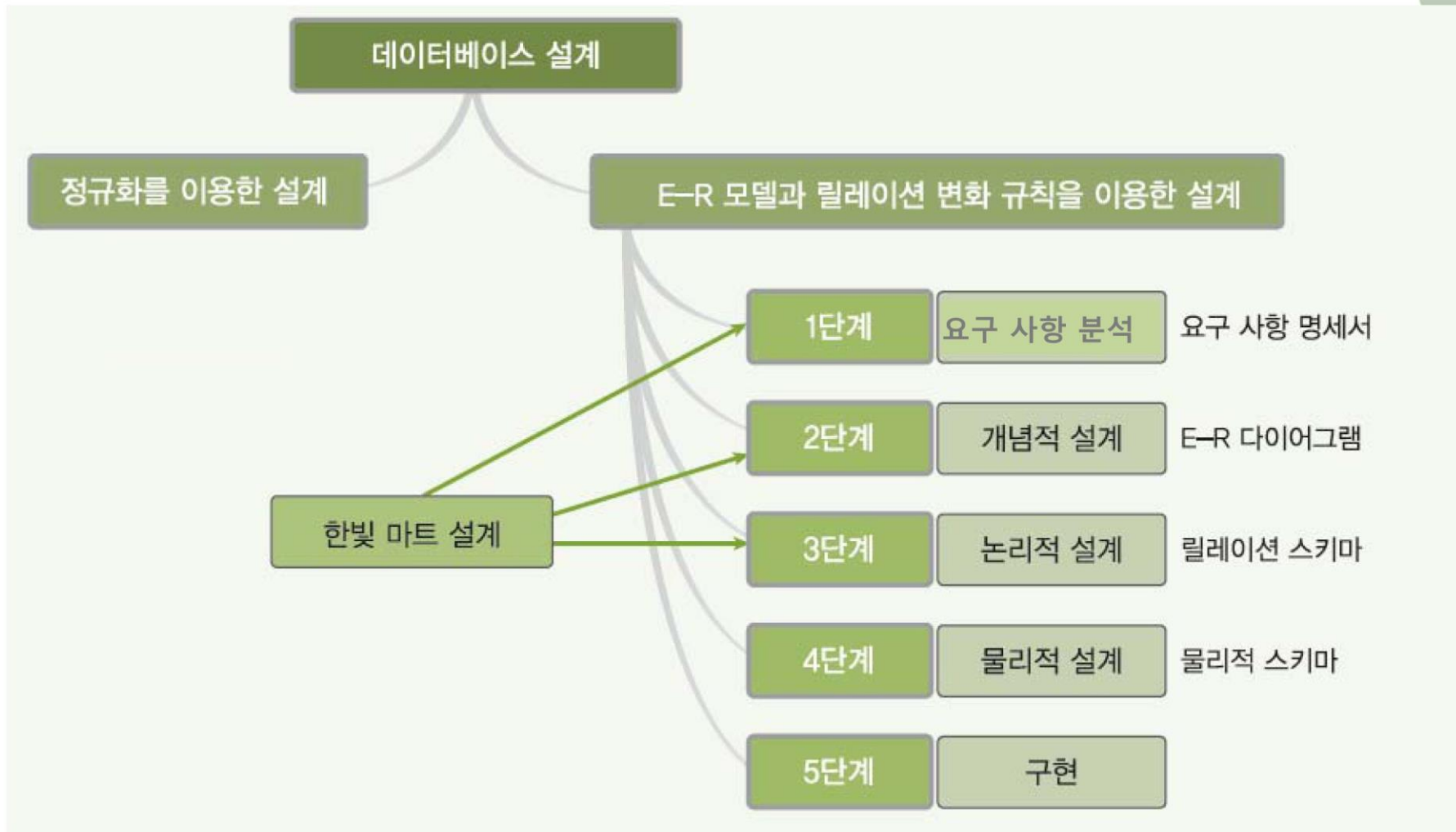


데이터베이스 설계

- 데이터베이스 설계 단계
- 요구 사항 분석
- 개념적 설계
- 논리적 설계
- 물리적 설계와 구현

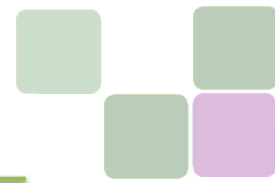
학습목표



- ▶ 데이터베이스 설계의 중요성과 목표를 이해한다.
- ▶ 데이터베이스 설계 5단계를 학습한다.
- ▶ 요구 사항 분석, 개념적 설계, 논리적 설계의 과정을 실제 예를 통해 연습해본다.



01 데이터베이스 설계 단계



❖ 데이터베이스 설계

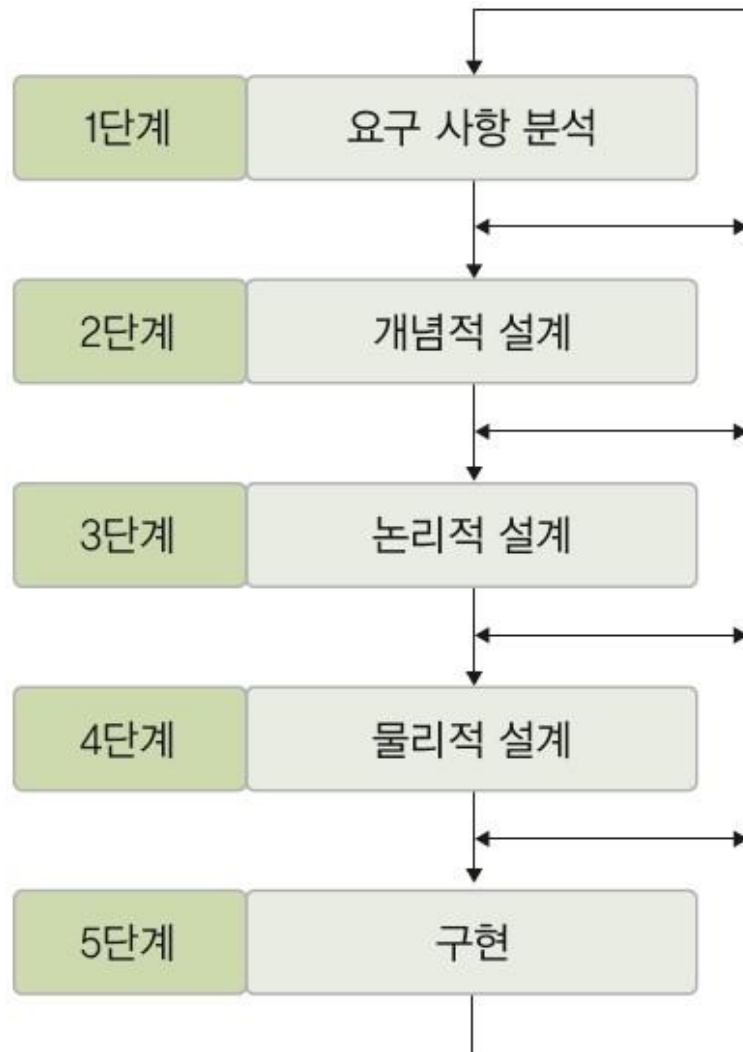
- 사용자의 다양한 요구 사항을 고려하여 데이터베이스를 생성하는 과정

❖ 관계 데이터베이스의 대표적인 설계 방법

- E-R 모델과 변환 규칙을 이용한 설계
- 정규화를 이용한 설계

01 데이터베이스 설계 단계

❖ E-R 모델과 릴레이션 변환 규칙을 이용한 설계의 과정



설계 과정 중에 오류가 발견되어
변경이 필요하면 이전 단계로
되돌아가 설계 내용을 변경 가능

그림 8-1 데이터베이스 설계의 과정

01 데이터베이스 설계 단계

❖ E-R 모델과 릴레이션 변환 규칙을 이용한 설계의 과정



그림 8-2 데이터베이스 설계 과정의 각 단계별 주요 작업과 결과물

❖ 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

02 요구사항 분석

❖ 요구 사항 분석 예 – [한빛 마트 데이터베이스]

■ 인터넷으로 회원들에게 상품을 판매하는 한빛 마트의 데이터베이스 개발

- 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.

02 요구사항 분석



그림 8-3 데이터베이스 설계 : 요구 사항 분석 단계

❖ 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

03 개념적 설계

❖ 설계 2 단계 : 개념적 설계

■ 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

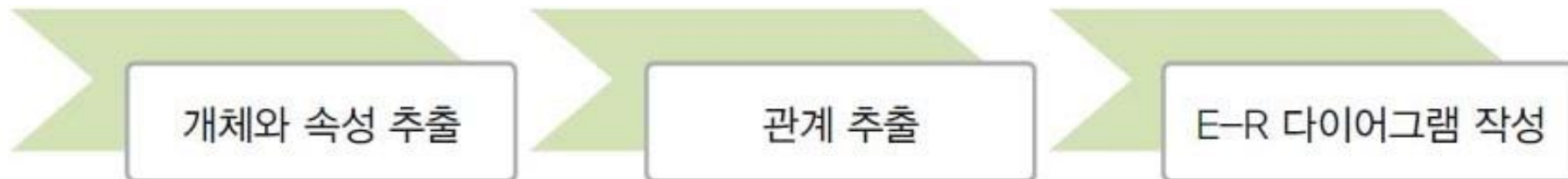


그림 8-6 개념적 모델링 과정

❖ 개념적 설계 – (STEP 1) 개체와 속성 추출

- 개체 : 저장할만한 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!

03 개념적 설계

❖ 개념적 설계 – (STEP 1) 개체와 속성 추출 예

- 요구 사항 명세서에서 개체와 속성을 추출하는 과정

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

그림 8-7 요구 사항 문장에서 명사를 선별한 예 (A)

- "한빛 마트"는 일반적이고 광범위한 의미의 명사이므로 제외
- "회원아이디", "비밀번호", "이름", "나이", "직업", "등급", "적립금"은 회원의 속성으로 분류
- "회원아이디"는 키 속성으로 분류

03 개념적 설계

❖ 개념적 설계 – (STEP 1) 개체와 속성 추출 예

■ 요구 사항 명세서에서 개체와 속성을 추출하는 과정

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

그림 8-8 요구 사항 문장에서 개체와 속성을 추출한 예(A)

[추출 결과]

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

03 개념적 설계

❖ 개념적 설계 – (STEP 1) 개체와 속성 추출 예

- 요구 사항 명세서에서 개체와 속성을 추출하는 과정

⑦ 회원이 상품을 주문하면 주문에 대한 주문번호, 주문수량, 배송지, 주문일자 정보를 유지해야 한다.

그림 8-10 요구 사항 문장에서 개체와 속성을 추출한 예(B)

[추출 결과]

- 개체 : 회원, 상품
- 속성 : 주문번호, 주문수량, 배송지, 주문일자
 - 회원이 상품을 주문을 해야 생기는 중요한 정보이기 때문에 회원이나 상품 개체의 속성으로 보기는 어렵고 이후 추출할 특정 관계의 속성일 가능성이 높음

❖ 개념적 설계 – (STEP 1) 개체와 속성 추출 예

■ 요구 사항 명세서에서 개체와 속성을 추출하는 과정

- 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 대해 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 명의 회원만 작성할 수 있다.
- ⑬ 게시글에 대한 글번호, 글제목, 글내용, 작성일자 정보를 유지해야 한다.
- ⑭ 게시글은 글번호로 식별한다.

03 개념적 설계

❖ 개념적 설계 – (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

03 개념적 설계

❖ 개념적 설계 – (STEP 1) 개체와 속성 추출 예

- 요구 사항 명세서에서 개체와 속성을 추출한 결과

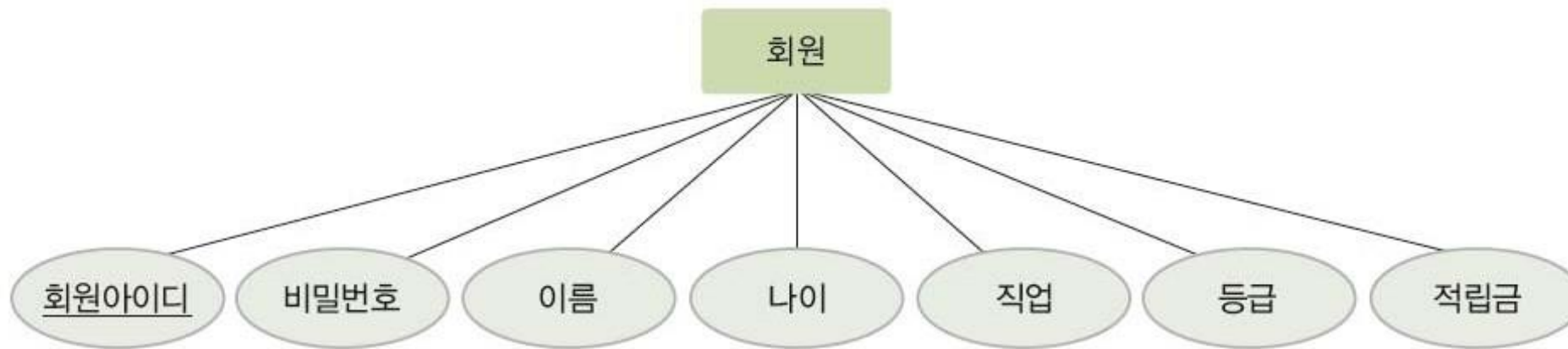


그림 8-13 회원 개체의 E-R 다이어그램

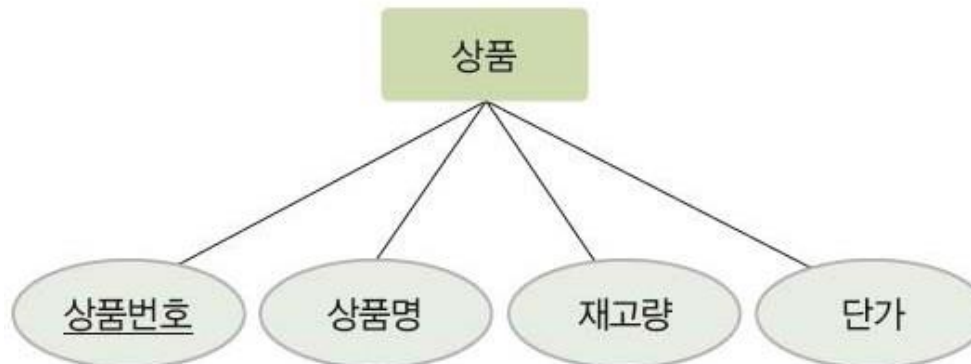


그림 8-14 상품 개체의 E-R 다이어그램

03 개념적 설계

❖ 개념적 설계 – (STEP 1) 개체와 속성 추출 예

- 요구 사항 명세서에서 개체와 속성을 추출한 결과

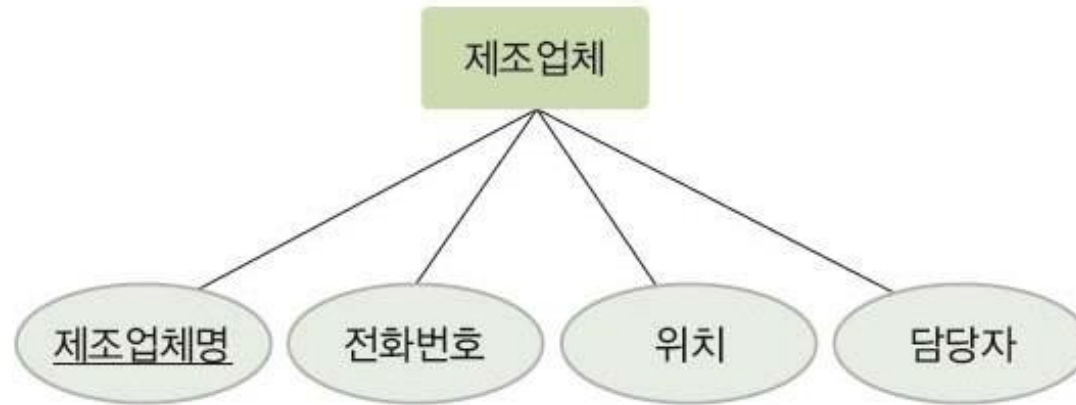


그림 8-15 제조업체 개체의 E-R 다이어그램

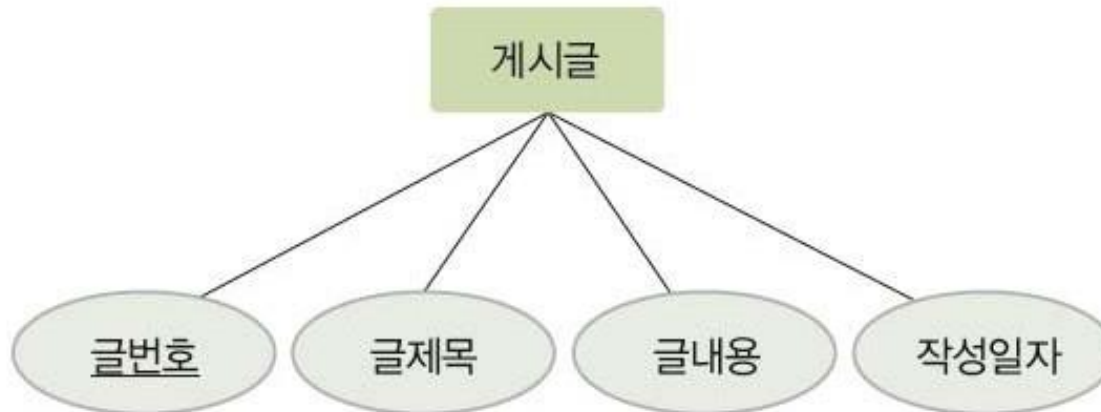
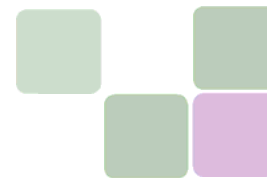


그림 8-16 게시글 개체의 E-R 다이어그램



데이터베이스 설계

- 데이터베이스 설계 단계
- 요구 사항 분석
- 개념적 설계
- 논리적 설계
- 물리적 설계와 구현

❖ 개념적 설계 – (STEP 2) 관계 추출

- 관계 : 개체 간의 의미 있는 연관성
- 관계 추출 방법
 - 요구 사항 문장에서 개체 간의 연관성을 의미 있게 표현한 **동사**를 찾아라!
 - 의미가 같은 동사가 여러 개일 경우는 대표 명사 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

03 개념적 설계

❖ 개념적 설계 – (STEP 2) 관계 추출 예

■ 요구 사항 명세서에서 관계를 추출하는 과정

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

그림 8-17 요구 사항 문장에서 동사를 선별한 예(A)

- “입력해야 한다”는 개체와 개체의 관계를 표현하는 동사로 볼 수 없으므로 제외
- “부여된다”는 개체와 개체의 관계를 표현하는 동사로 볼 수 없으므로 제외
- “식별한다”는 개체와 개체의 관계를 표현하는 동사로 볼 수 없으므로 제외

03 개념적 설계

❖ 개념적 설계 – (STEP 2) 관계 추출 예

- 요구 사항 명세서에서 관계를 추출하는 과정

- 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.

그림 8-18 요구 사항 문장에서 동사를 선별한 예(B)

[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

❖ 개념적 설계 – (STEP 2) 관계 추출 예

- 요구 사항 명세서에서 관계를 추출하는 과정

- ⑧ 각 상품은 한 제조업체가 공급하고, 제조업체 하나는 여러 상품을 공급할 수 있다.
- ⑨ 제조업체가 상품을 공급하면 공급일자와 공급량 정보를 유지해야 한다.

그림 8-19 요구 사항 문장에서 동사를 선별한 예(C)

[추출 결과]

- 관계: 공급
 - “상품” 개체와 “제조업체” 개체가 맺는 관계, 일대다(1:n) 관계
 - “상품” 개체는 관계에 필수적으로 참여 / “제조업체” 개체는 관계에 선택적으로 참여
- “공급” 관계의 속성: 공급일자, 공급량

03 개념적 설계

❖ 개념적 설계 – (STEP 2) 관계 추출 예

- 요구 사항 명세서에서 관계를 추출하는 과정

⑫ 회원은 게시글을 여러 개 작성할 수 있고, 게시글 하나는 한 명의 회원만 작성할 수 있다.

그림 8-20 요구 사항 문장에서 동사를 선별한 예 (D)

[추출 결과]

- 관계: 작성
 - “회원” 개체와 “게시글” 개체가 맺는 관계, 일대다(1:n) 관계
 - “회원” 개체는 관계에 선택적으로 참여 / “게시글” 개체는 관계에 필수적으로 참여

03 개념적 설계

❖ 개념적 설계 – (STEP 2) 관계 추출 예

■ 요구 사항 명세서에서 관계를 추출하는 과정

- 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 하나가 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.

03 개념적 설계

❖ 개념적 설계 – (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

03 개념적 설계

❖ 개념적 설계 – (STEP 2) 관계 추출 예

■ 요구 사항 명세서에서 관계를 추출한 결과

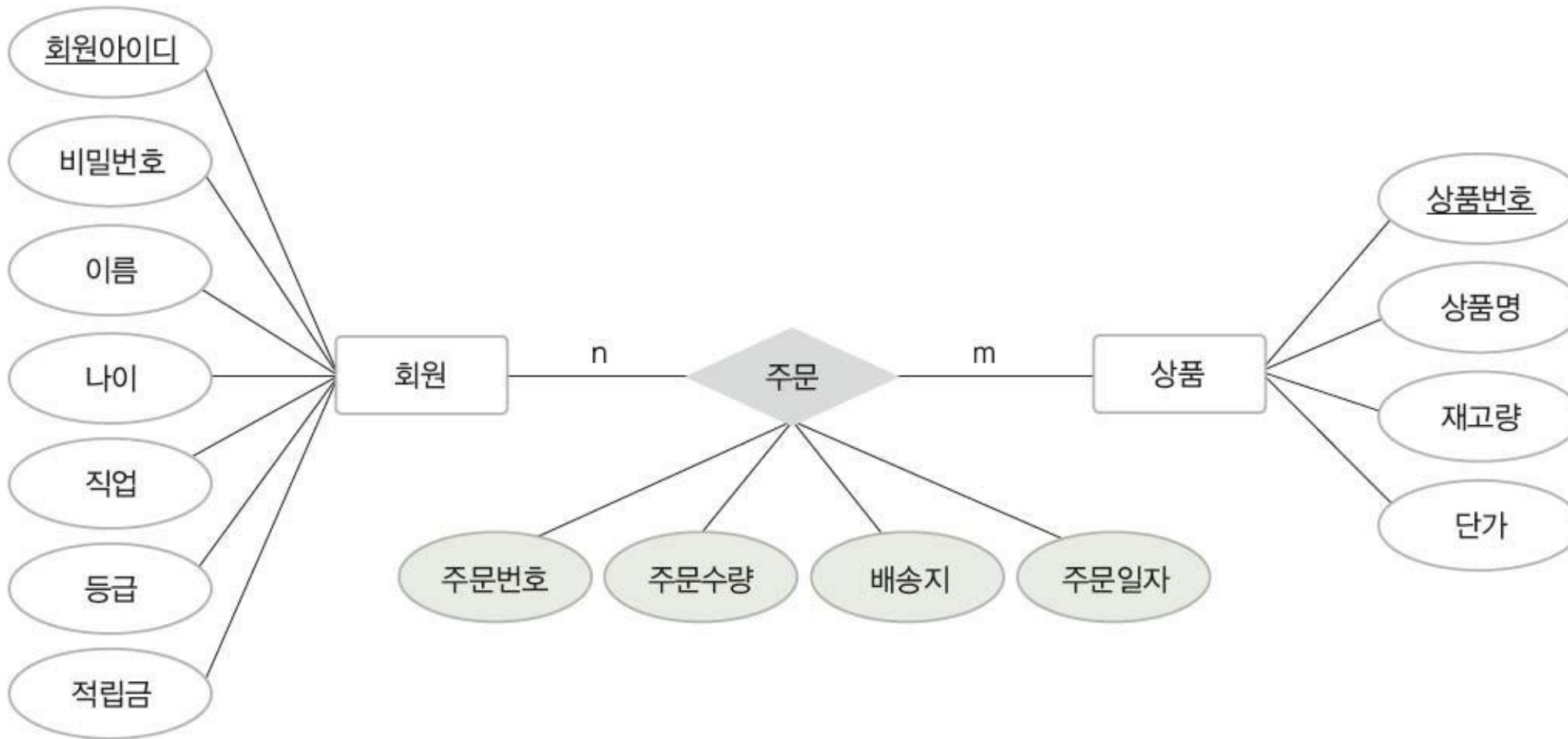


그림 8-23 주문 한빛 마트 요구 사항 명세서의 E-R 다이어그램

03 개념적 설계

❖ 개념적 설계 – (STEP 2) 관계 추출 예

■ 요구 사항 명세서에서 관계를 추출한 결과

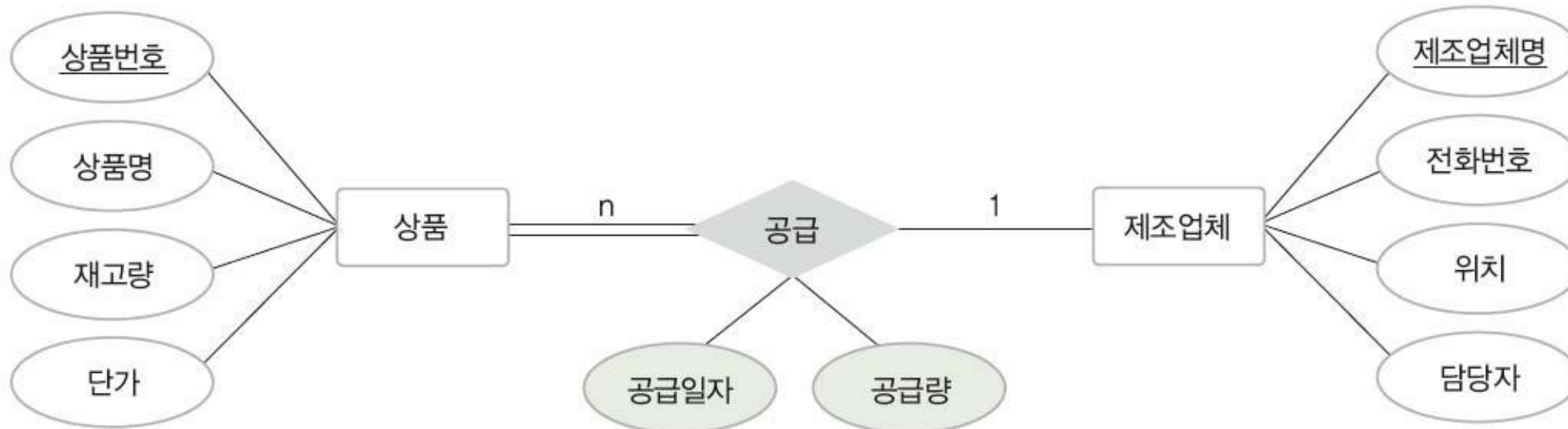


그림 8-24 공급 관계의 E-R 다이어그램

03 개념적 설계

❖ 개념적 설계 – (STEP 2) 관계 추출 예

■ 요구 사항 명세서에서 관계를 추출한 결과

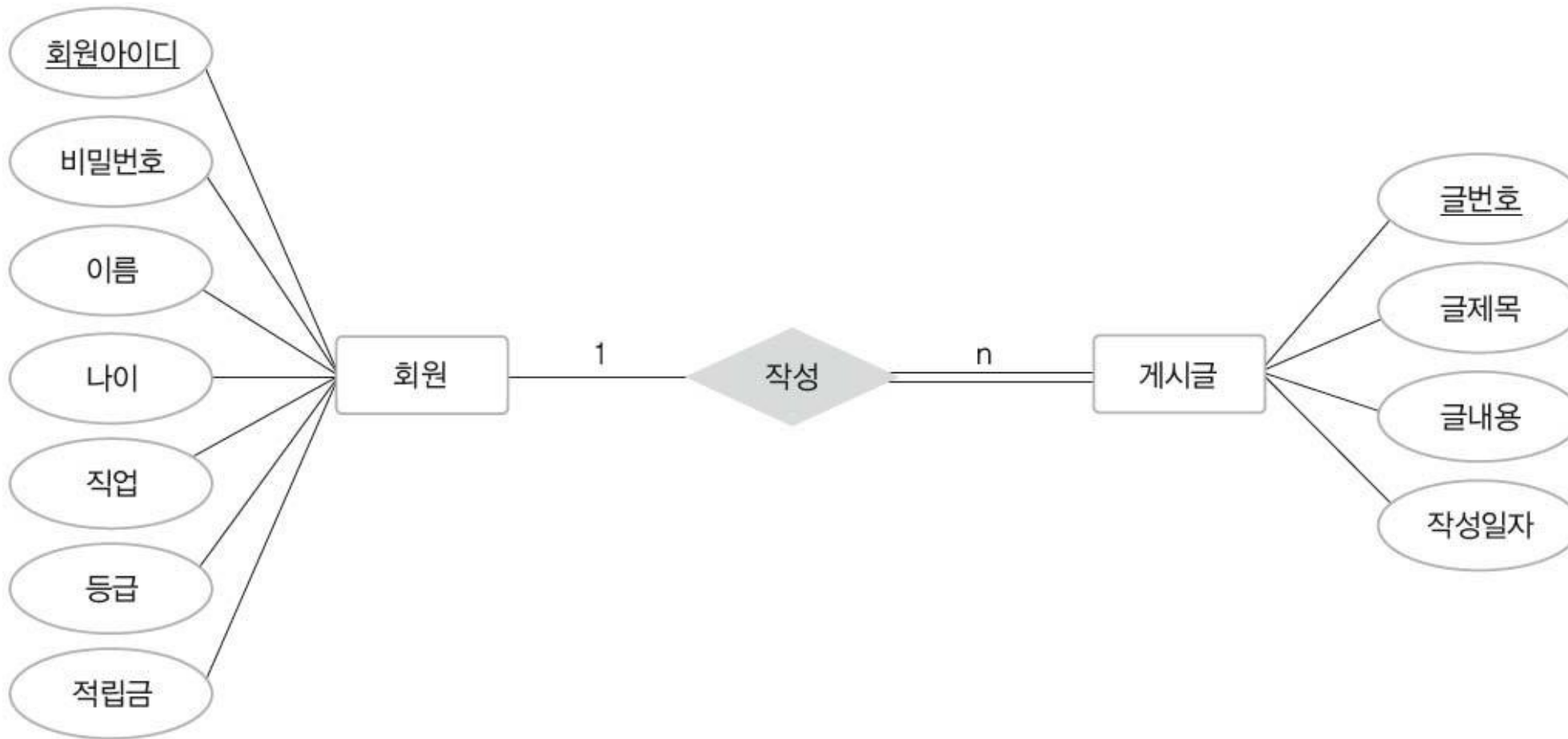


그림 8-25 작성 관계의 E-R 다이어그램

03 개념적 설계

❖ 개념적 설계 – (STEP 3) E-R 다이어그램 작성

- 요구 사항 명세서를 개념적 스키마로 작성한 결과

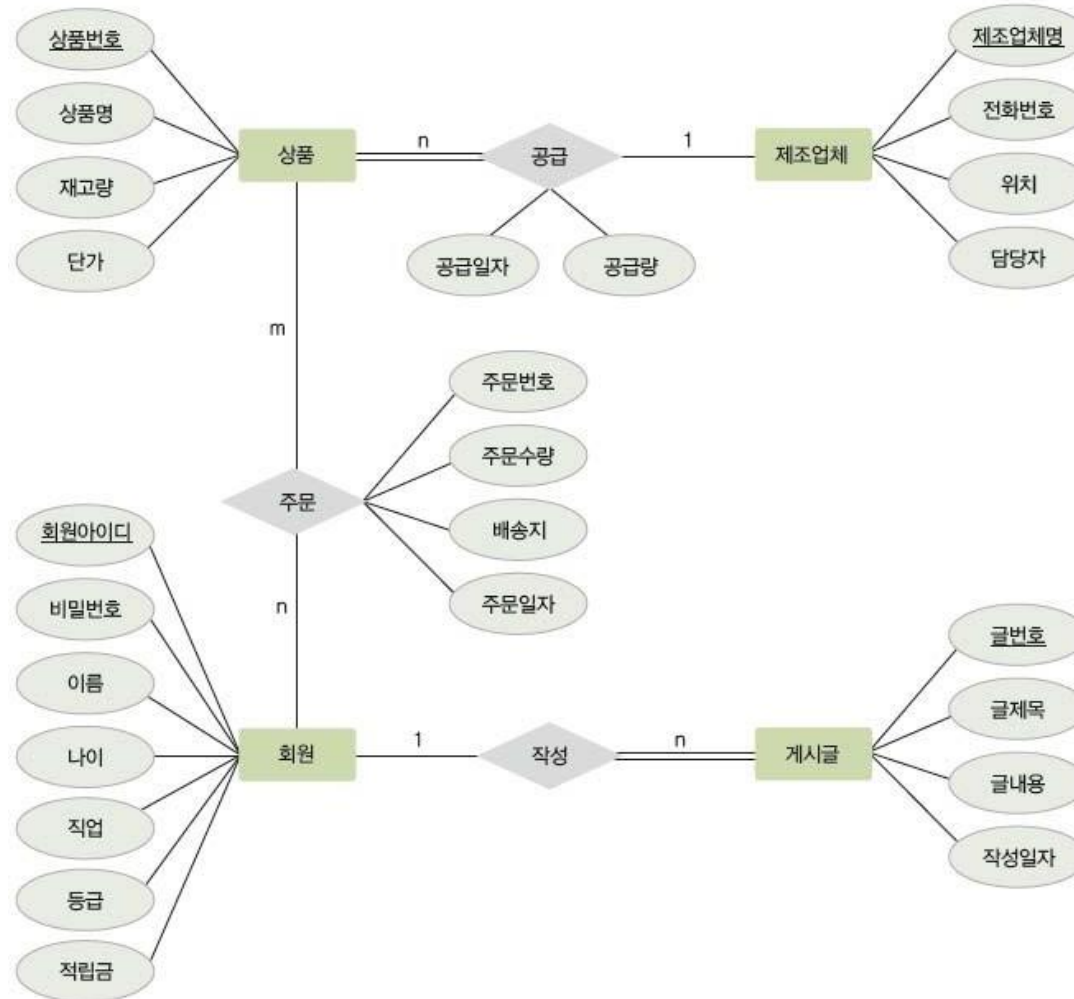


그림 8-26 한빛 마트 요구 사항 명세서의 E-R 다이어그램

1. 요구사항 분석

1) 한국건설의 구조

- 한국건설은 10대 건설회사 중 하나로 수십 개의 사업장에 직원들이 근로하며 한국건설은 수백 개의 하청업체를 가지고 있다. 직원으로 총당할 수 없는 인원은 하청업체를 두어서 관리한다.
- 하지만 이러한 상관관계는 생략하고 사업장 관리 부분만 개체로 표현하기로 한다.

2) 서비스와 제한점

- 사원이 근무하는 사업장을 확인할 수 있으며 한 명의 사원은 어느 기간 동안에는 하나의 사업장에만 근무할 수 있으며, 그 기간이 지나면 다른 사업장에서 근무할 수 있다.
- 구입한 사업장자재는 한 사업장에서만 사용할 수 있으며, 한 사업장에서 관리하는 사업장 자재는 많다.

1. 요구사항 분석

3) 사용자 요구 사항을 분석한 결과

- 사원은 (사원번호, 사원명, 주소, 전화번호, 직급, 부서명)의 속성을 갖는다.
- 사업장은 (사업장번호, 사업장명, 주소, 전화번호, 공사금액, 투입인원, 시공일자, 예상완공일, 완공일, 비고)의 속성을 갖는다.
- 사업장의 비고는 공사중과 공사완료로 구분한다.
- 사업장자재는 (자재명코드, 자재명, 수량, 구입가격, 구입일)의 속성을 갖는다.
- 한 사원은 일정 기간 동안 하나의 사업장에서 근무하며 그 기간이 지나면 다른 사업장에서 근무한다.
- 구입한 사업장자재는 하나의 사업장에서만 관리할 수 있다.

과제)

<<자동차수리 전문점의 데이터베이스 구축을 위한 요구 사항>>

자동차수리 전문점 ‘다수리’ 는 사업확장을 위해 자동차수리 서비스와 직원을 관리할 수 있도록 데이터베이스를 구축하려고 한다. ‘다수리’ 에는 여러명의 **직원**이 근무하고 있으며 직원번호(key), 이름, 주소, 연락처 및 월급을 관리한다. 새로운 **고객**이 자동차 수리를 요청하면 고객정보를 등록하며 이때 고객번호(key), 고객명, 주소, 연락처를 입력한다. **자동차**에 대해서는 자동차번호(key), 제조사, 연식, 주행거리 정보를 관리한다. 한 명의 고객은 여러대의 자동차를 소유할 수 있다. 고객이 자동차 수리를 요청하면 한명의 전담직원이 할당되고 이때 서비스 번호가 부여된다. 수리 후 수리비와 수리시간 정보를 기록한다. 수리 요청은 한번에 한 자동차에 대해서만 가능하다.

※ 각 엔티티, 속성, 관계 및 데이터 타입은 수험자가 알아서 결정, 정의한다.

03 개념적 설계



그림 8-5 데이터베이스 설계 : 개념적 설계 단계

❖ 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 스키마 변환 후 속성의 데이터 타입, 길이, 널 값 허용 여부, 기본 값, 제약조건 등을 세부적으로 결정하고 결과를 문서화시킴

❖ 설계 3 단계 : 논리적 설계

- 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

04 논리적 설계

❖ 논리적 설계 – (규칙 1) 모든 개체는 릴레이션으로 변환한다

- E-R 다이어그램의 각 개체를 하나의 릴레이션으로 변환
 - 개체의 이름 → 릴레이션 이름
 - 개체의 속성 → 릴레이션의 속성
 - 개체의 키 속성 → 릴레이션의 기본키
 - 개체의 속성이 복합 속성인 경우에는 복합 속성을 구성하고 있는 단순 속성만 릴레이션의 속성으로 변환

04 논리적 설계

❖ 논리적 설계 – (규칙 1) 모든 개체는 릴레이션으로 변환한다



그림 8-28 개체를 릴레이션으로 변환하는 규칙을 적용한 예

상품(상품번호, 상품명, 재고량, 단가)

04 논리적 설계

❖ 논리적 설계 – (규칙 1) 모든 개체는 릴레이션으로 변환한다

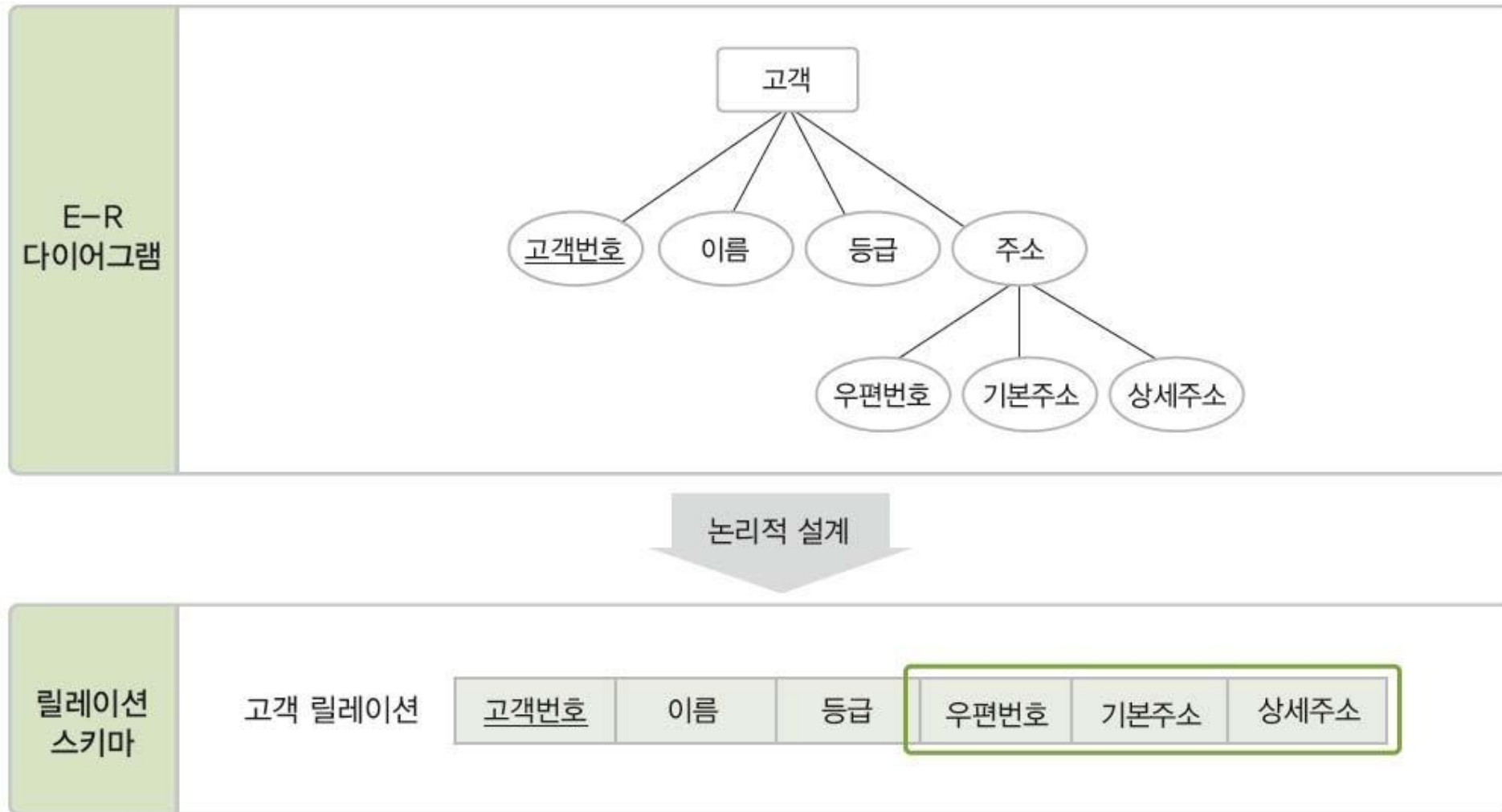


그림 8-29 복합 속성을 가지는 개체를 릴레이션으로 변환하는 예

❖ 논리적 설계 – (규칙 2) 다대다 관계는 릴레이션으로 변환한다

- E-R 다이어그램의 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 지정하고 외래키들을 조합하여 관계 릴레이션의 기본키로 지정

04 논리적 설계

❖ 논리적 설계 – (규칙 2) 다대다 관계는 릴레이션으로 변환한다

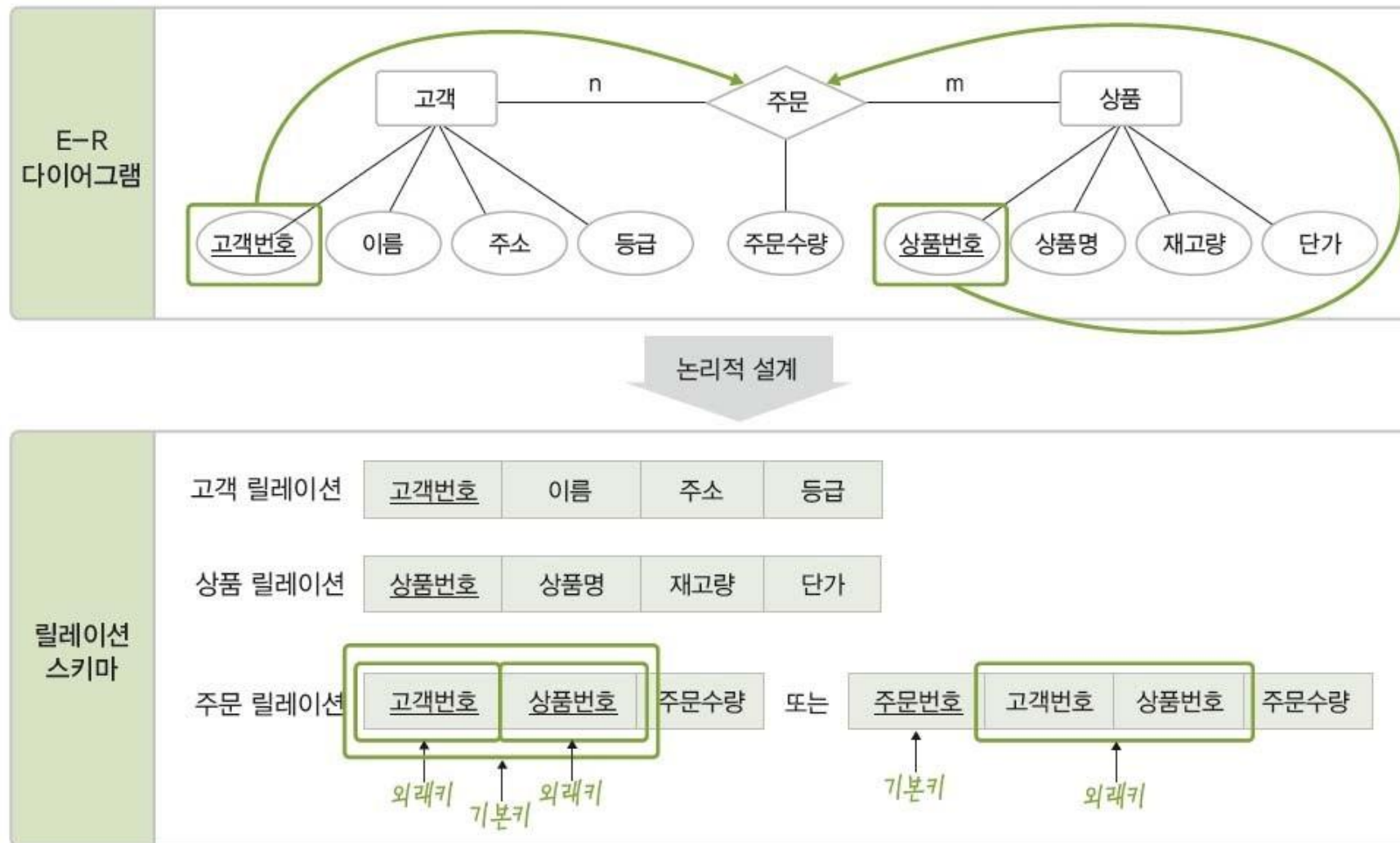


그림 8-30 다대다 관계를 릴레이션으로 변환하는 규칙을 적용한 예

04 논리적 설계

❖ 논리적 설계 – (규칙 3) 일대다 관계는 외래키로 표현한다

- E-R 다이어그램의 일대다 관계는 외래키로만 표현
 - (규칙 3-1) 일반적인 일대다 관계는 외래키로 표현한다.
 - (규칙 3-2) 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키로 지정한다.

04 논리적 설계

❖ 논리적 설계 – (규칙 3) 일대다 관계는 외래키로 표현한다

- (규칙 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

04 논리적 설계

❖ 논리적 설계 – (규칙 3) 일대다 관계는 외래키로 표현한다

- (규칙 3-1) 일반적인 일대다 관계는 외래키로 표현한다.

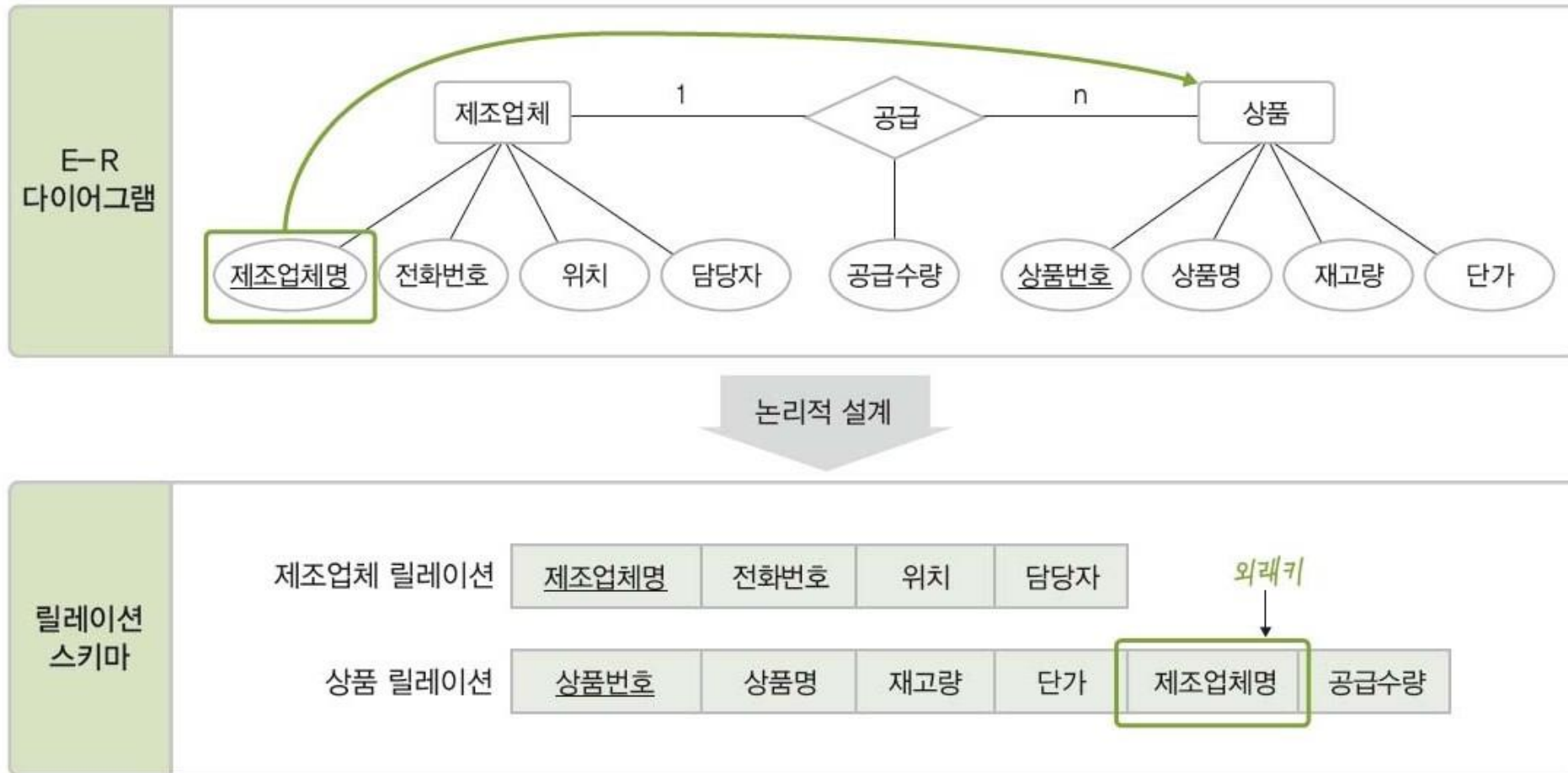


그림 8-31 일반적인 개체가 참여하는 일대다 관계를 외래키로 표현하는 규칙을 적용한 예

04 논리적 설계

❖ 논리적 설계 – (규칙 3) 일대다 관계는 외래키로 표현한다

- (규칙 3-2) 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키를 지정한다.
 - 일대다(1:n) 관계에서 1측 개체 릴레이션의 기본키를 n측 개체 릴레이션에 포함시켜 외래키로 지정
 - 관계의 속성들도 n측 개체 릴레이션에 포함시킴
 - n측 개체 릴레이션은 외래키를 포함하여 기본키를 지정함
 - 약한 개체는 오너 개체에 따라 존재 여부가 결정되므로 오너 개체의 기본키를 이용해 식별해야 함

04 논리적 설계

❖ 논리적 설계 – (규칙 3) 일대다 관계는 외래키로 표현한다

- (규칙 3-2) 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키를 지정한다.

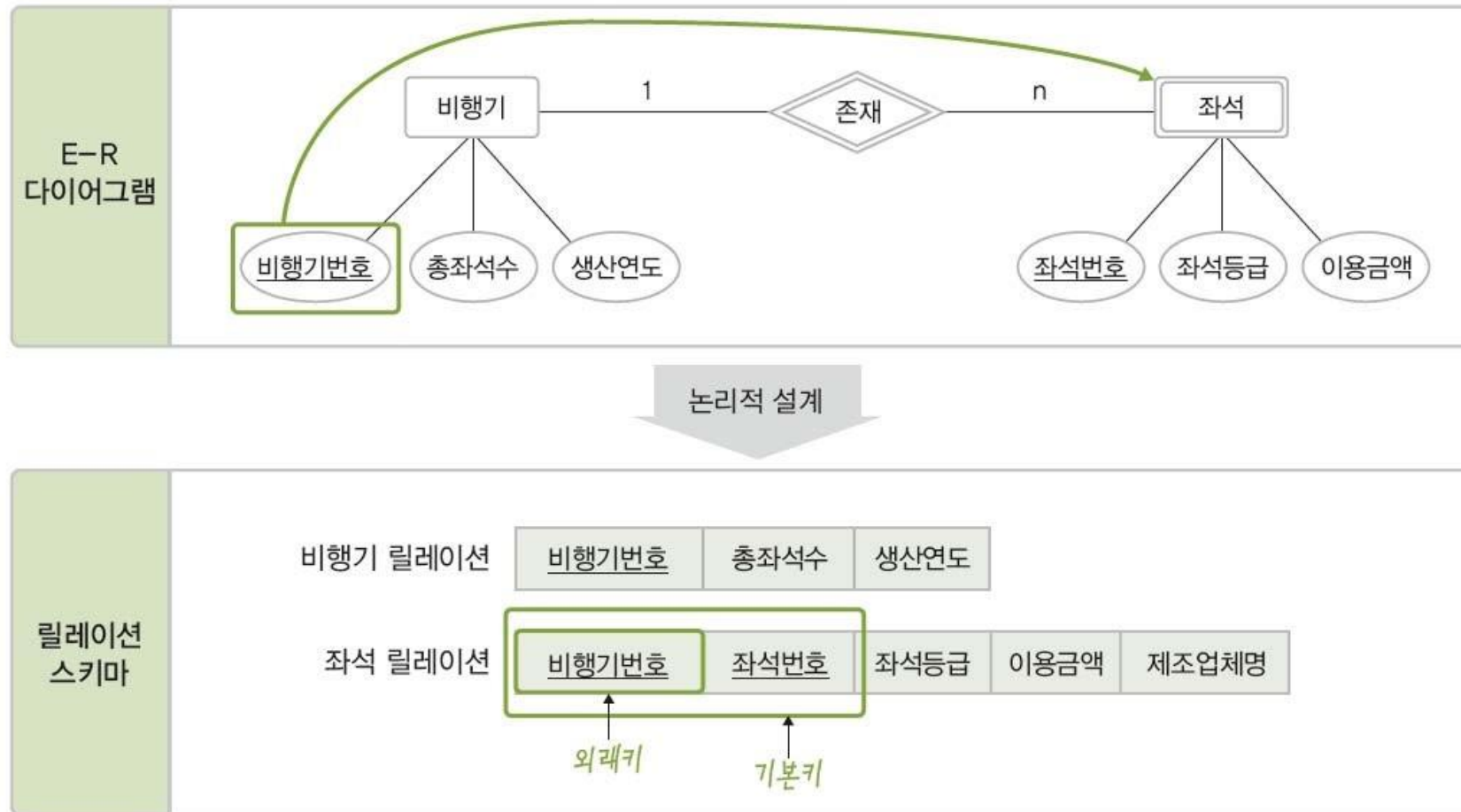


그림 8-32 약한 개체가 참여하는 일대다 관계를 외래키로 표현하는 규칙을 적용한 예

04 논리적 설계

❖ 논리적 설계 – (규칙 4) 일대일 관계는 외래키로 표현한다

- E-R 다이어그램의 일대일 관계는 외래키로만 표현
 - (규칙 4-1) 일반적인 일대일 관계는 외래키를 서로 주고 받는다.
 - (규칙 4-2) 일대일 관계에 필수적으로 참여하는 개체의 릴레이션만 외래키를 받는다.
 - (규칙 4-3) 모든 개체가 일대일 관계에 필수적으로 참여하면 릴레이션을 하나로 합친다.

04 논리적 설계

❖ 논리적 설계 – (규칙 4) 일대일 관계는 외래키로 표현한다

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

04 논리적 설계

❖ 논리적 설계 – (규칙 4) 일대일 관계는 외래키로 표현한다

- (규칙 4-1) 일반적인 일대일 관계는 외래키를 서로 주고 받는다.

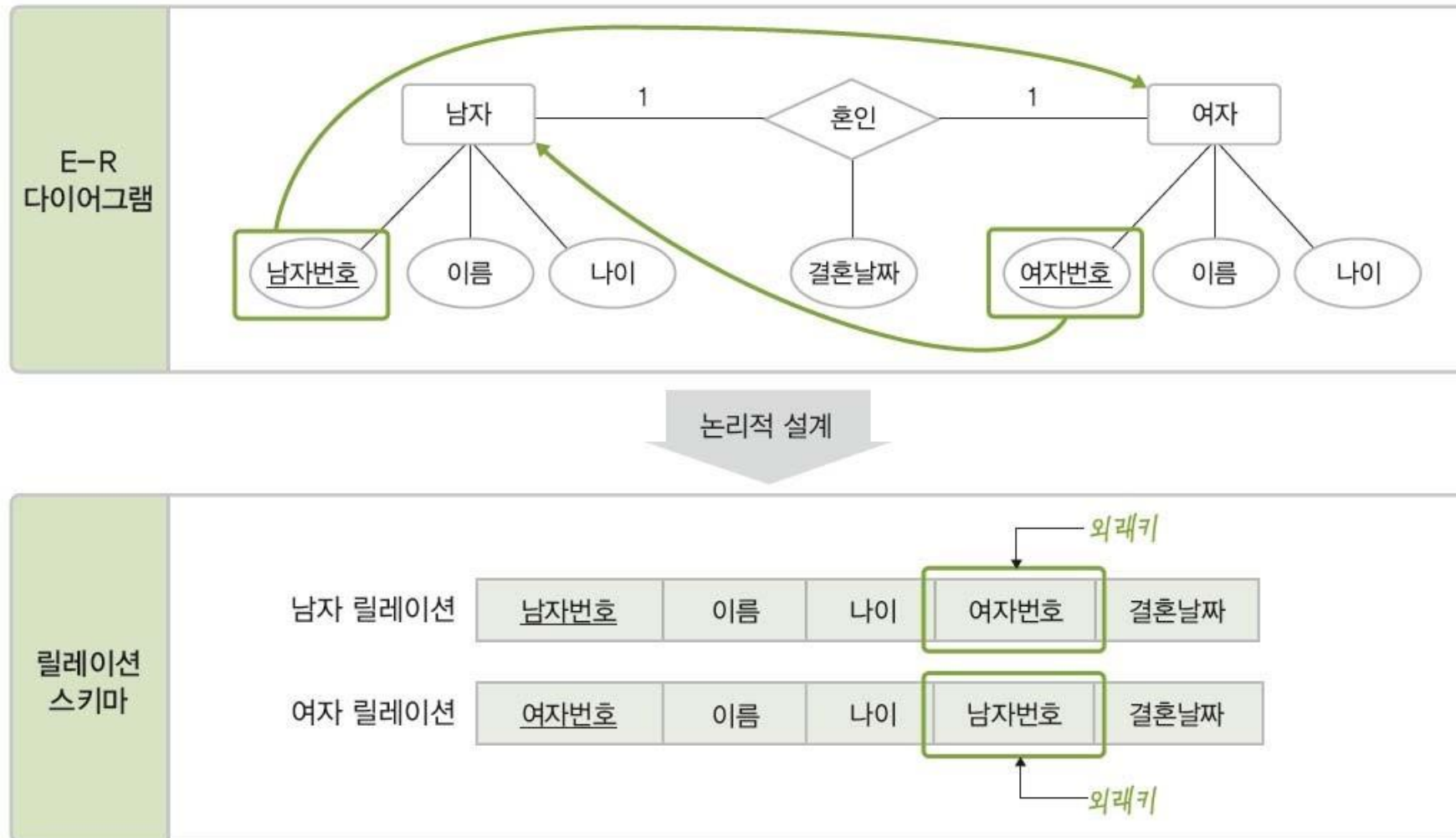


그림 8-33 일반적인 일대일 관계를 외래키로 표현하는 규칙을 적용한 예

04 논리적 설계

❖ 논리적 설계 – (규칙 4) 일대일 관계는 외래키로 표현한다

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

04 논리적 설계

- ❖ 논리적 설계 – (규칙 4) 일대일 관계는 외래키로 표현한다
 - (규칙 4-2) 필수적으로 참여하는 개체 릴레이션만 외래키를 받는다.

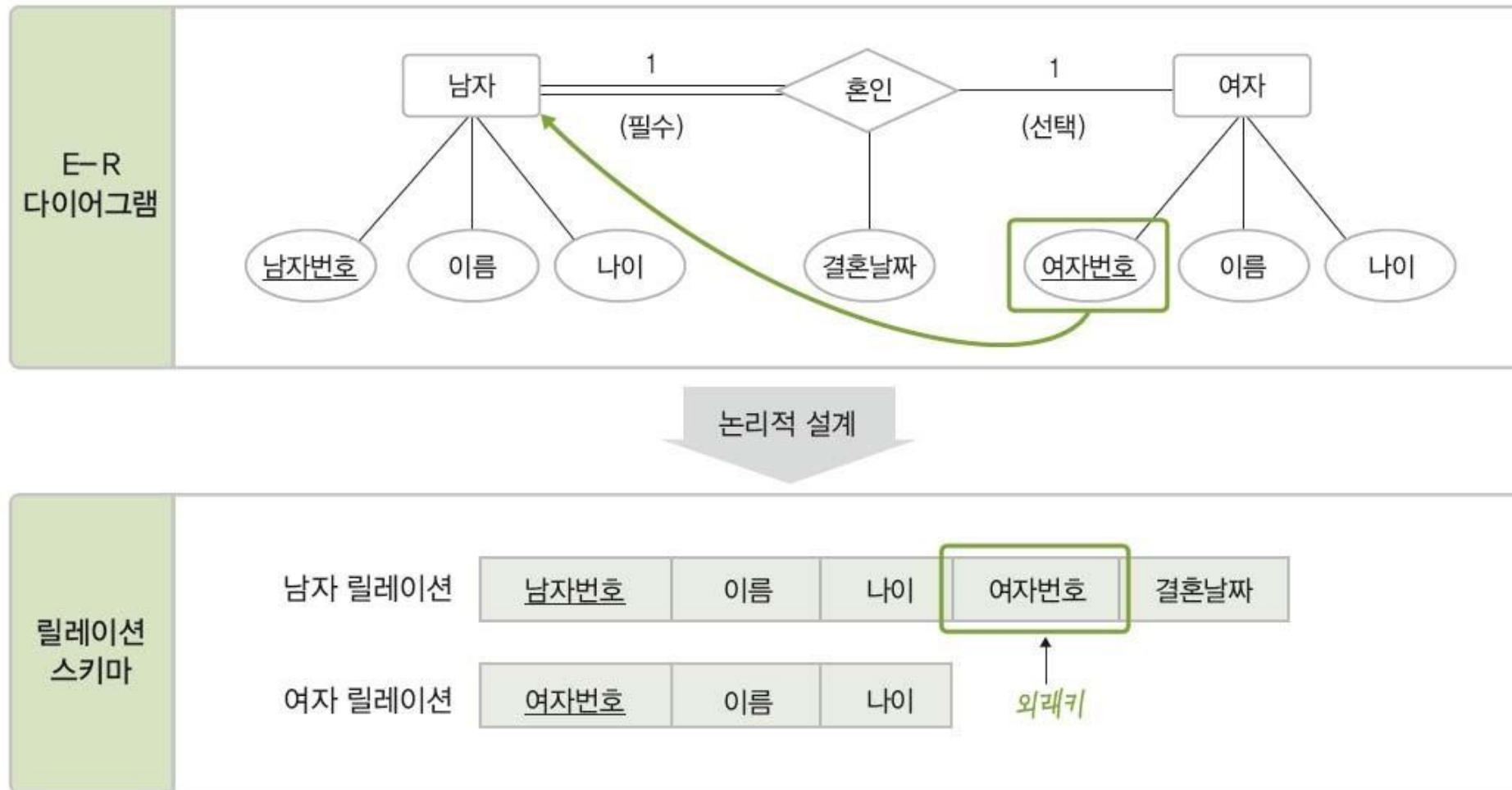


그림 8-34 일대일 관계에 필수적으로 참여하는 개체의 릴레이션이 외래키를 가지는 예

04 논리적 설계

❖ 논리적 설계 – (규칙 4) 일대일 관계는 외래키로 표현한다

- (규칙 4-3) 모든 개체가 필수적으로 참여하면 릴레이션을 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

04 논리적 설계

❖ 논리적 설계 – (규칙 4) 일대일 관계는 외래키로 표현한다

- (규칙 4-3) 모든 개체가 필수적으로 참여하면 릴레이션을 하나로 합친다.

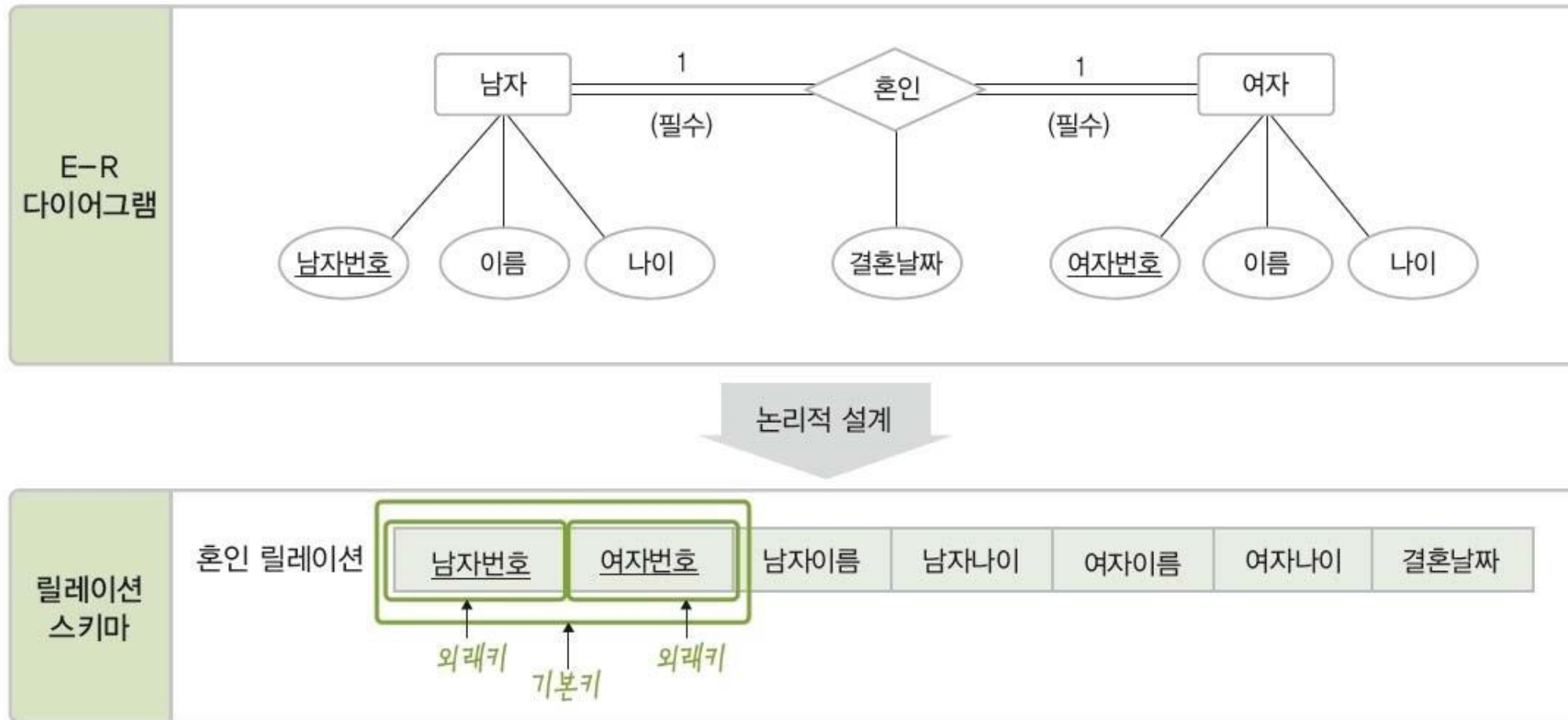


그림 8-35 일대일 관계에 모든 개체가 필수적으로 참여하면 릴레이션을 통합하는 예

04 논리적 설계

❖ 논리적 설계 – (규칙 5) 다중 값 속성은 릴레이션으로 변환한다

- E-R 다이어그램의 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

04 논리적 설계

❖ 논리적 설계 – (규칙 5) 다중 값 속성은 릴레이션으로 변환한다

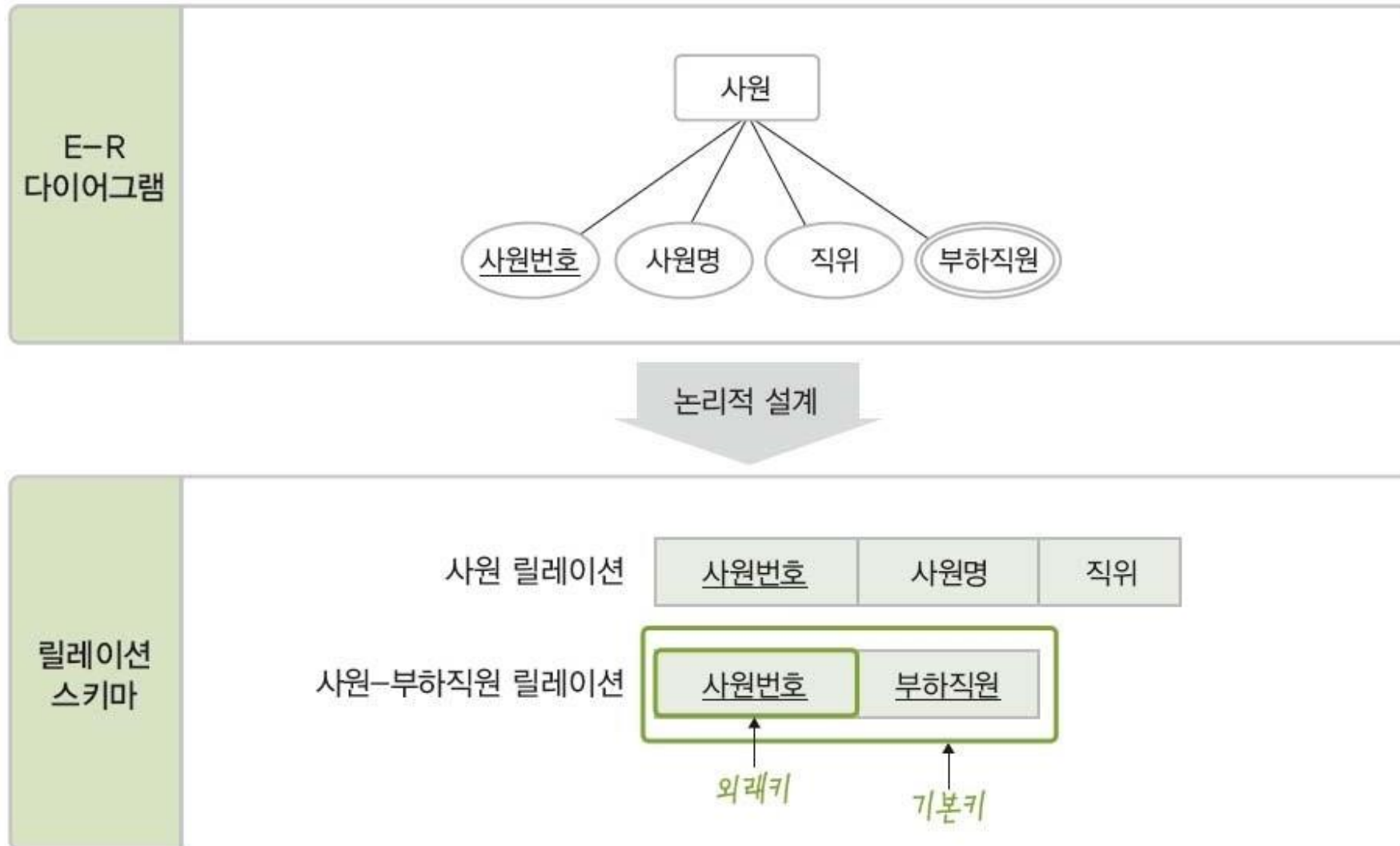


그림 8-36 다중 값 속성을 릴레이션으로 변환하는 규칙을 적용한 예

04 논리적 설계

❖ 논리적 설계 – (규칙 5) 다중 값 속성은 릴레이션으로 변환한다

사원번호	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 없는 릴레이션 특성을 위반함

04 논리적 설계

❖ 논리적 설계 – (규칙 5) 다중 값 속성은 릴레이션으로 변환한다

사원번호	사원명	직위	부하직원
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

04 논리적 설계

❖ 논리적 설계 – (규칙 5) 다중 값 속성은 릴레이션으로 변환한다

사원 릴레이션

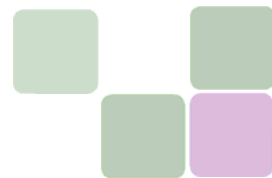
<u>사원번호</u>	사원명	직위
e001	홍정화	부장
e002	김수창	과장
e003	최종민	차장

사원-부하직원 릴레이션

<u>사원번호</u>	<u>부하직원</u>
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.



데이터베이스 설계

- 데이터베이스 설계 단계
- 요구 사항 분석
- 개념적 설계
- 논리적 설계
- 물리적 설계와 구현

04 논리적 설계

❖ 논리적 설계 – 기타 고려 사항

- 모든 관계를 독립적인 릴레이션으로 변환할 수 있다.
- 속성이 많은 관계는 유형에 상관없이 릴레이션으로의 변환을 고려할 수 있음

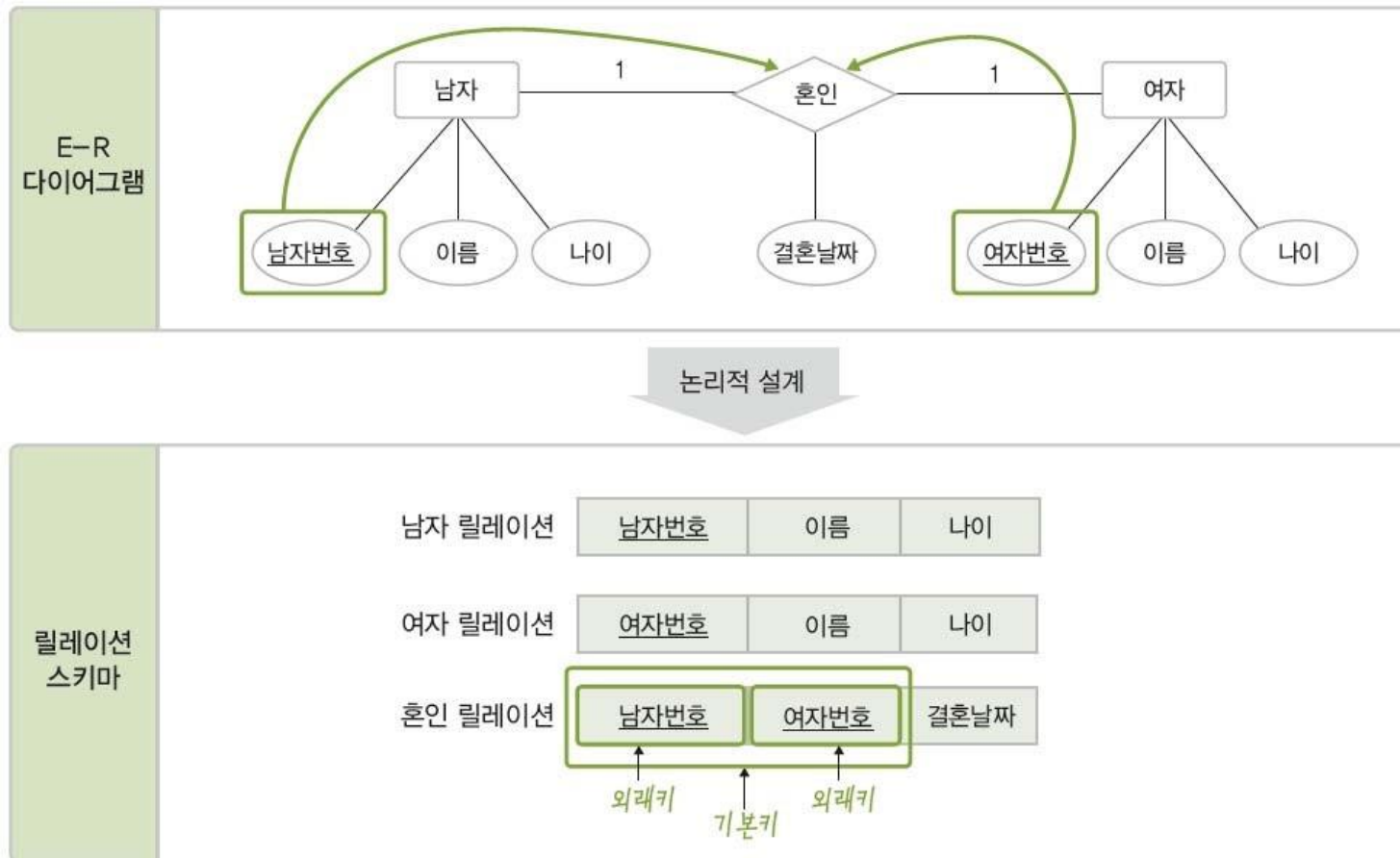


그림 8-40 일대일 관계를 릴레이션으로 변환하는 예

04 논리적 설계

❖ 논리적 설계 – 기타 고려 사항

- 개체가 자기 자신과 관계를 맺는 순환 관계도 기본 규칙을 그대로 적용

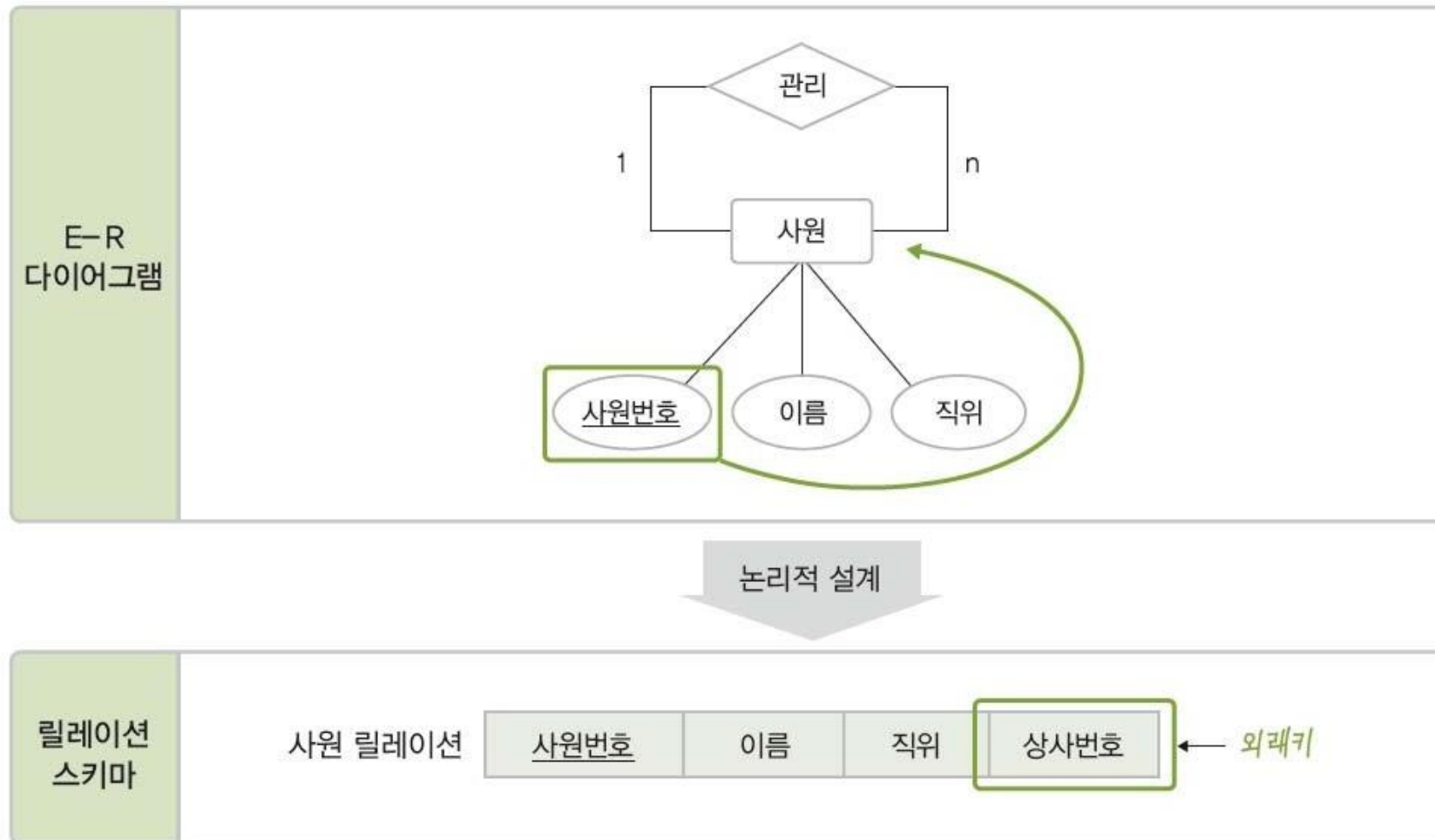


그림 8-41 순환 관계를 변환하는 예

04 논리적 설계

❖ 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예

- E-R 다이어그램을 릴레이션으로 변환하는 과정
- STEP 1) 규칙 1 적용

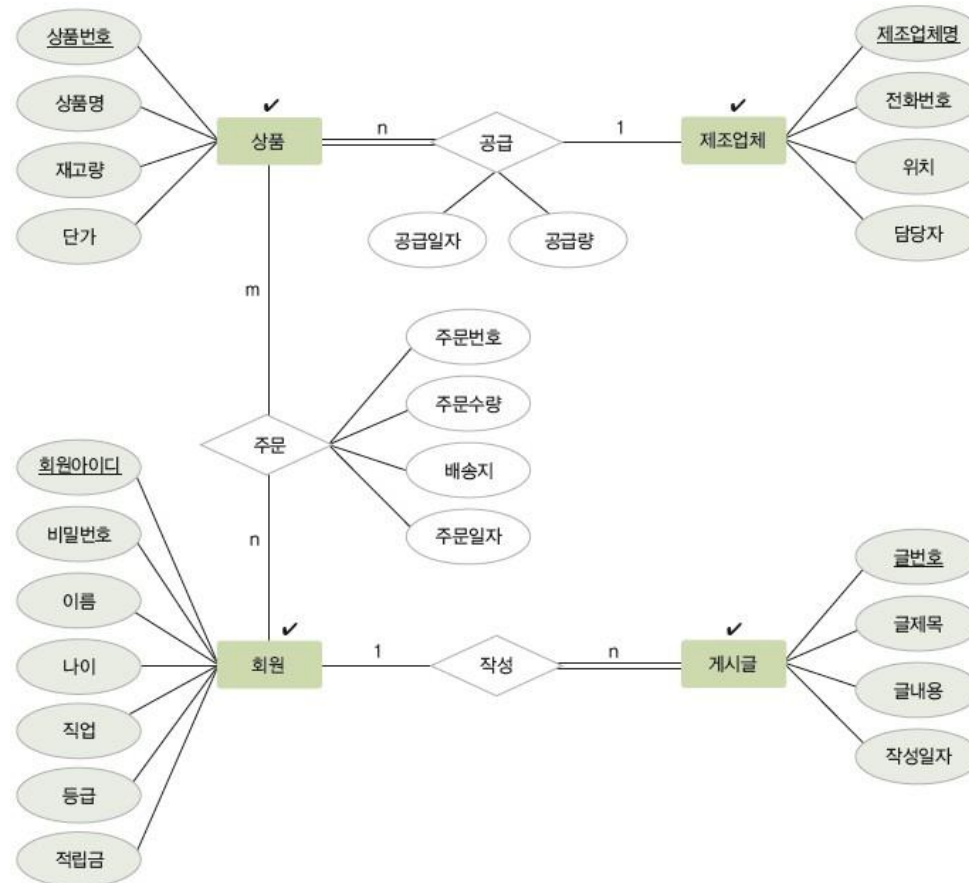


그림 8-42 릴레이션 스키마로 변환할 한빛 마트 E-R 다이어그램

04 논리적 설계

❖ 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예

- E-R 다이어그램을 릴레이션으로 변환하는 과정
- (규칙 1) 적용 결과

회원 릴레이션	<u>회원아이디</u>	비밀번호	이름	나이	직업	등급	적립금
---------	--------------	------	----	----	----	----	-----

상품 릴레이션	<u>상품번호</u>	상품명	재고량	단가
---------	-------------	-----	-----	----

제조업체 릴레이션	<u>제조업체명</u>	전화번호	위치	담당자
-----------	--------------	------	----	-----

게시글 릴레이션	<u>글번호</u>	글제목	글내용	작성일자
----------	------------	-----	-----	------

그림 8-43 규칙1을 적용한 결과

04 논리적 설계

❖ 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예

- E-R 다이어그램을 릴레이션으로 변환하는 과정
- STEP 2) 규칙 2 적용

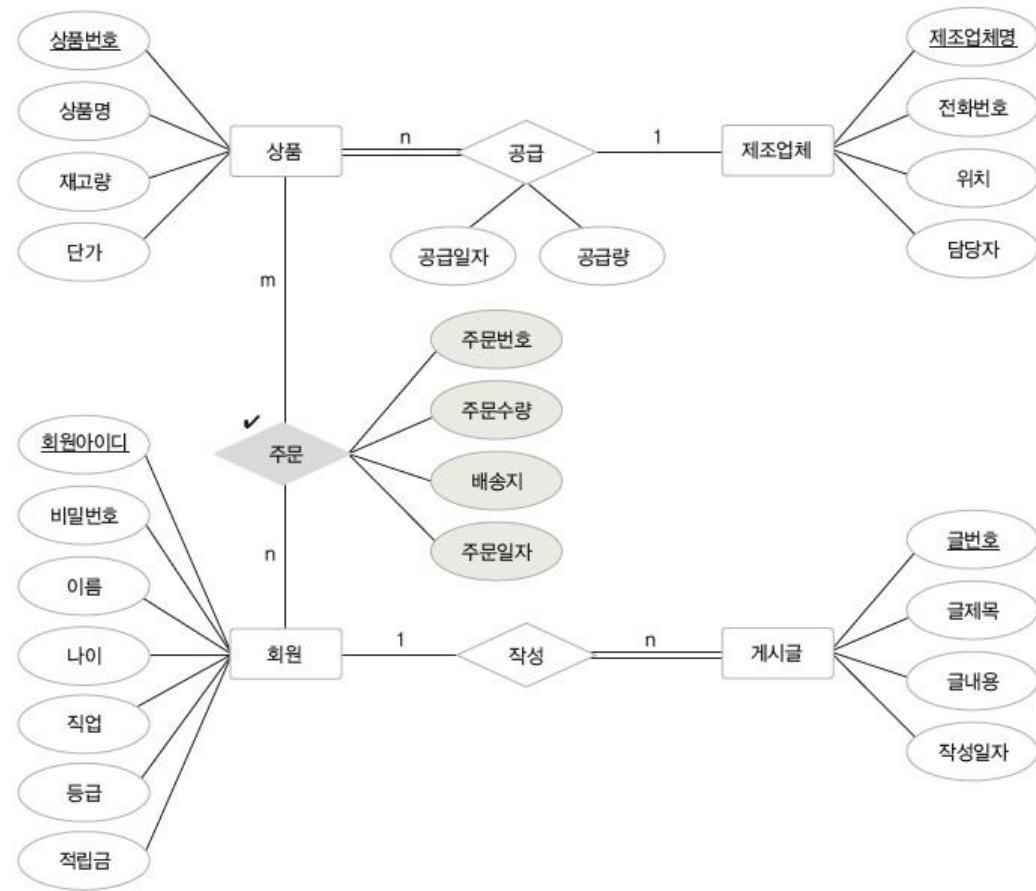


그림 8-44 한빛 마트의 E-R 다이어그램에 표현된 다대다 관계

04 논리적 설계

❖ 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예

- E-R 다이어그램을 릴레이션으로 변환하는 과정
- (규칙 2) 적용 결과

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

외래키 기본키 외래키

그림 8-45 규칙2를 적용한 결과

04 논리적 설계

❖ 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예

- E-R 다이어그램을 릴레이션으로 변환하는 과정
- STEP 3) 규칙 3 적용

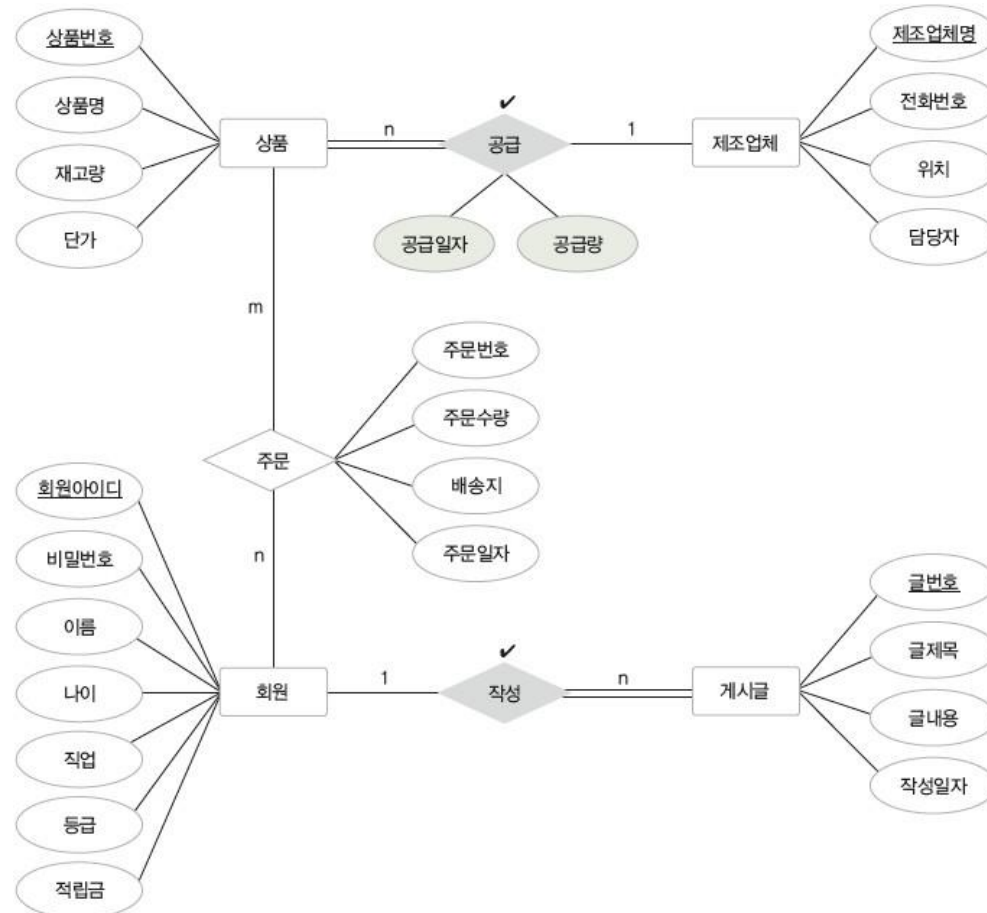


그림 8-46 한빛 마트의 E-R 다이어그램에 표현된 일대다 관계

04 논리적 설계

❖ 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예

- E-R 다이어그램을 릴레이션으로 변환하는 과정
- (규칙 3) 적용 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 규칙3을 적용한 결과

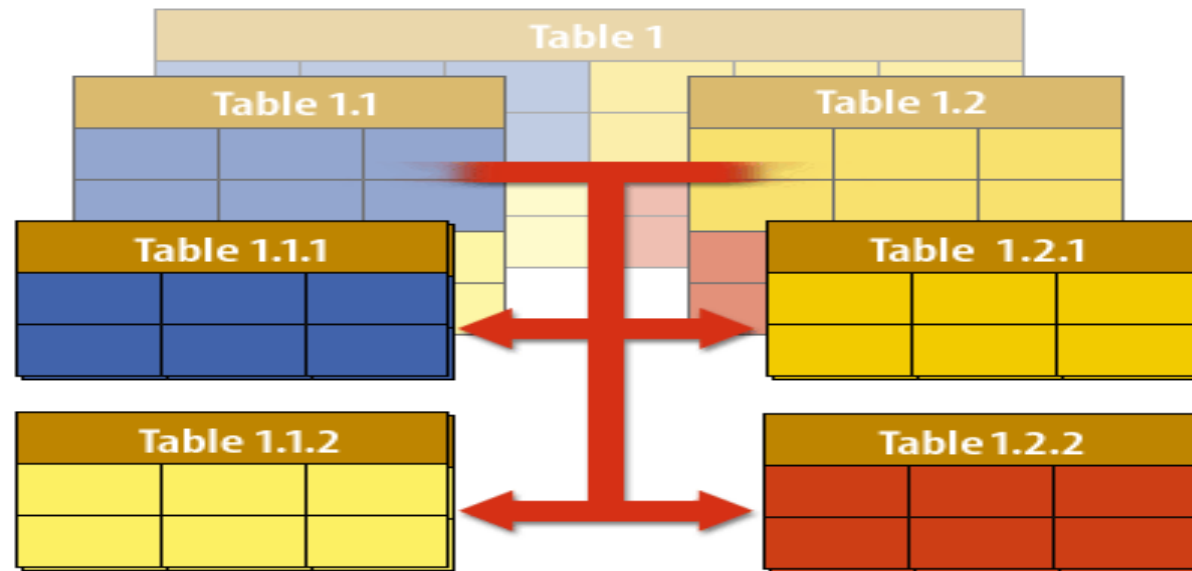
04 논리적 설계

❖ 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예

- E-R 다이어그램을 릴레이션으로 변환하는 과정
- STEP 4) 규칙 4 적용
 - 일대일 관계가 없으므로 규칙 4는 적용할 필요가 없음
- STEP 5) 규칙 5 적용
 - 다중 값 속성이 없으므로 규칙 5는 적용할 필요가 없음

❖ 정규화란?

- 관계형 데이터베이스 설계 시 중복을 최소화하도록 데이터를 구조화 하는 작업
- 정규화를 하는 목적 – 이상(anomaly) 이 있는 관계를 재구성함으로써 바람직한 스키마로 구성



...until you have broken the single table into a series of smaller tables that reflect the business objects represented in your database.

❖ 함수종속

- 데이터의 종속성에는 함수종속/ 다가종속/ 조인종속/ 파생종속 등이 존재
- 함수종속은 릴레이션 내에 존재하는 속성 간의 종속성 의미
- 즉, 릴레이션에서 A속성의 값이 B속성의 값을 유일하게 식별할 수 있다면 B속성은 A속성에 함수적으로 종속 되었다고 함

❖ 결정자와 종속자

- 결정자(Determinant) – 종속성 분류시 기준이 되는 값
- 종속자(Dependent) – 결정자의 값에 의해 정해질 수 있는 값

04 논리적 설계

- 속성 Y 가 속성 x 에 함수적으로 종속되었다는 것은 \Rightarrow 속성 x 의 값을 이용하여 속성 Y 의 값을 유일하게 식별할 수 있다는 의미(x 를 결정자, Y 를 종속자라 함)

$X \rightarrow Y$, 또는 $y=f(x)$ 라 표현

예) 주민번호, 이름, 휴대전화번호, 주소 속성이 있는 경우 “주민번호” 속성이 결정자 임

❖ 이행종속

- 만약 $X \rightarrow Y \rightarrow Z$ 종속이 있다면 Z 는 X 에 —간접적으로 종속되어 있음
- 이를 이행종속(Transitive Dependency)이라하며 같은 엔터티로 설계해서는 안됨
- 직접 종속인 X 와 Y , Y 와 Z 를 별도의 엔터티로 설계(3정규화)

❖ 종속성과 폐포(Closure)

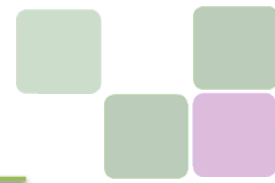
- 릴레이션 R 의 속성 X 의 폐포는 X 에 종속되었다고 추론되는 모든 속성의 집합

$X \rightarrow (Y, Z)$ 이면

$X^+ = X, Y, Z$ 이다

❖ 종속성 추론 규칙

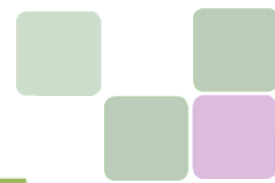
- $Y \subseteq X$ 이면 $X \rightarrow Y$ 관계가 성립
- $X \rightarrow Y$ 이면 $XZ \rightarrow YZ$ 가 성립
- $X \rightarrow Y$ 이고 $Y \rightarrow Z$ 이면 $X \rightarrow Z$ 가 성립(이행종속)
- $X \rightarrow YZ$ 이면 $X \rightarrow Y$ 이고 $X \rightarrow Z$ 가 성립
- $X \rightarrow Y$ 이고 $X \rightarrow Z$ 이면 $X \rightarrow YZ$ 가 성립
- $X \rightarrow Y$ 이고 $YZ \rightarrow W$ 이면 $XZ \rightarrow W$ 가 성립



❖ 아노말리(Anomaly)

- 데이터의 이상 현상
- Update Anomaly - 릴레이션에서 업데이트 할 때 발생하는 데이터 이상 현상
- Insert Anomaly - 릴레이션에서 새로운 인스턴스를 삽입 할 때 발생하는 데이터 이상 현상
- Delete Anomaly - 릴레이션에서 인스턴스를 삭제 할 때 발생하는 데이터 이상 현상

04 논리적 설계

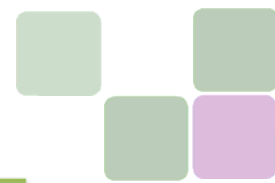


❖ 아노말리(Anomaly)

#선수번호	선수이름	포지션	#팀번호	팀이름	리그구분	리그이름
10	홍길동	유격수	25	라이거스	ML	메이저리그
14	강길동	투수	25	라이거스	ML	메이저리그
15	박길동	중견수	25	라이거스	ML	메이저리그
13	이길동	포수	20	라이언스	ML	메이저리그
14	강길동	투수	10	타이언스	SL	싱글리그
12	김길동	1루수	05	타이거스	SL	싱글리그
10	홍길동	유격수	10	타이언스	SL	싱글리그
11	최길동	2루수	10	타이언스	SL	싱글리그
12	김길동	1루수	15	엘리펀츠	DL	더블리그
10	홍길동	유격수	30	갯츠	DL	더블리그

❖ 아노말리(Anomaly)

- 업데이트 아노말리
 - '홍길동' 이 포지션이 '2루수' 로 바뀌면->3개의 인스턴스 모두 바뀌어야 함
 - 만일 홍길동의 포지션이 '2루수' 로 남아있다면 데이터 이상현상이 발생
- 삭제 아노말리
 - '김길동' 의 인스턴스 삭제시 '타이거즈'와 '엘리펀츠' 팀의 데이터도 삭제
- 삽입 아노말리
 - 새로운 선수 '장길동'이 팀이 정해지지 않으면 삽입 불가능(팀번호가 식별자)



❖ 정규화의 종류

- 정규화는 1 정규화(First Normal Form), 2 정규화(Second Normal Form), 3 정규화(Third Normal Form), 보이스 코드 정규화(Boyce-codd Normal Form), 4 정규화(Fourth Normal Form), 5 정규화(Fifth Normal Form)이 있다.

04 논리적 설계

❖ 1 정규화

- 모든 속성은 반드시 하나의 값을 가져야함-다가속성(Multivalued Attributes), 복합 속성(Composite Attributes)과 관련

#고객번호	고객명	주민번호	취미코드
100	홍길동	123456-2345678	10,11,12
101	김길동	234567-3456789	10,11
102	박길동	345678-4567890	12

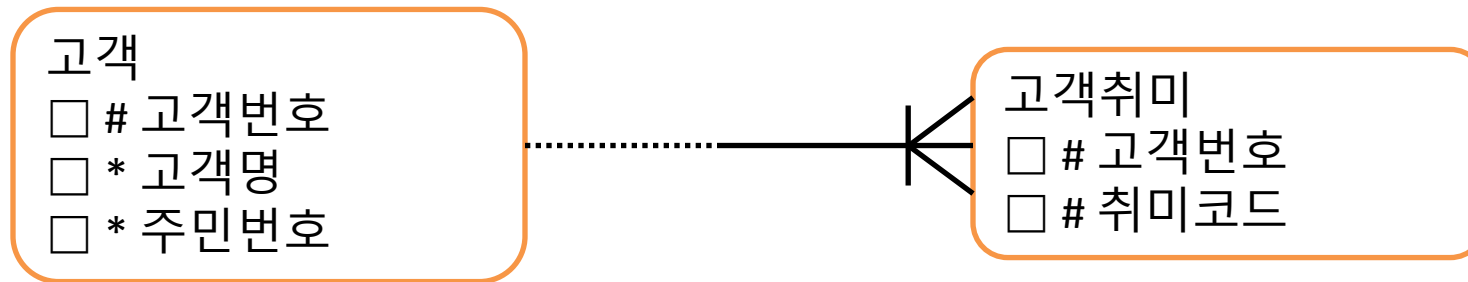
다가속성

#고객번호	고객명	주민번호
100	홍길동	123456-2345678
101	김길동	234567-3456789
102	박길동	345678-4567890

#고객번호	#취미코드
100	10
100	11
100	12
101	10
101	11
102	12

04 논리적 설계

- 중첩된 릴레이션을 새로운 릴레이션으로 만들고 주 식별자는 원래 릴레이션의 주 식별자와 중첩된 릴레이션의 주 식별자를 조합하여 사용



- 1 정규화 대상
 - 다가 속성이 사용된 릴레이션
 - 복합속성이 사용된 릴레이션
 - 유사속성이 반복되는 릴레이션
 - 중첩 릴레이션
 - 동일 속성이 여러 릴레이션에 사용된 경우

04 논리적 설계

- 예) 다음 릴레이션을 1 정규화 하라

1)

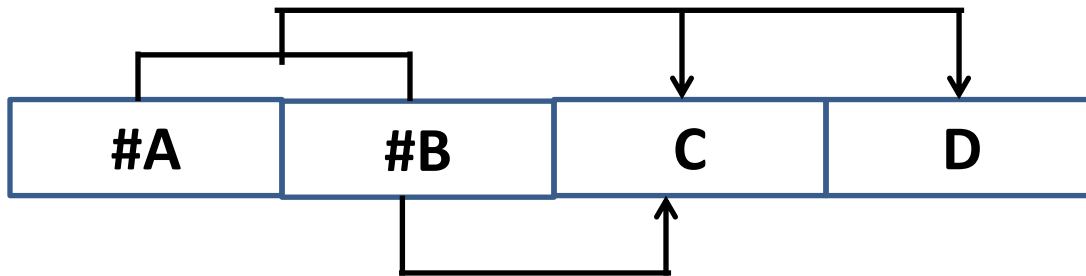
주문
☐ # 주문번호
☐ * 고객번호
☐ * 주문일자
☐ * 상품번호
☐ * 주문수량

2)

주문
☐ # 주문번호
☐ * 주문금액
☐ * 주문일자
☐ * 고객번호
☐ * 연락처번호
☐ * 배송주소
☐ * 상품번호1
☐ * 주문수량1
☐ * 상품번호2
☐ * 주문수량2
☐ * 상품번호3
☐ * 주문수량3

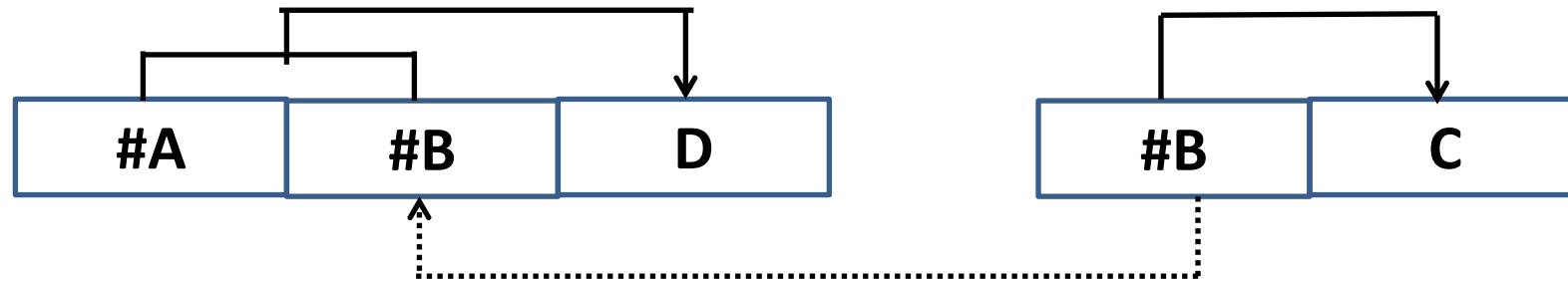
❖ 2 정규화

- 주 식별자가 두 개 이상인 릴레이션에서 발생
- 모든 비 식별자는 주 식별자에 완전 함수종속 되어야 한다.
- 일반속성 중 주 식별자 전체에 종속적이지 않은 속성을 찾아 기본 엔터티에서 제거하고 그 속성의 결정자를 주 식별자로 하는 새로운 상위 엔터티를 생성



(부분 종속이 발생한 릴레이션)

04 논리적 설계



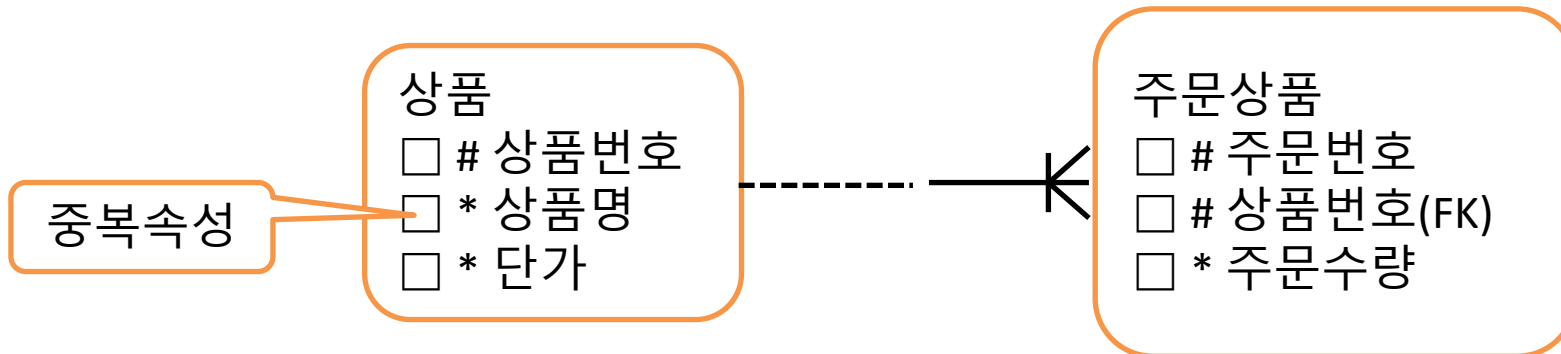
(2 정규화를 한 릴레이션)

04 논리적 설계

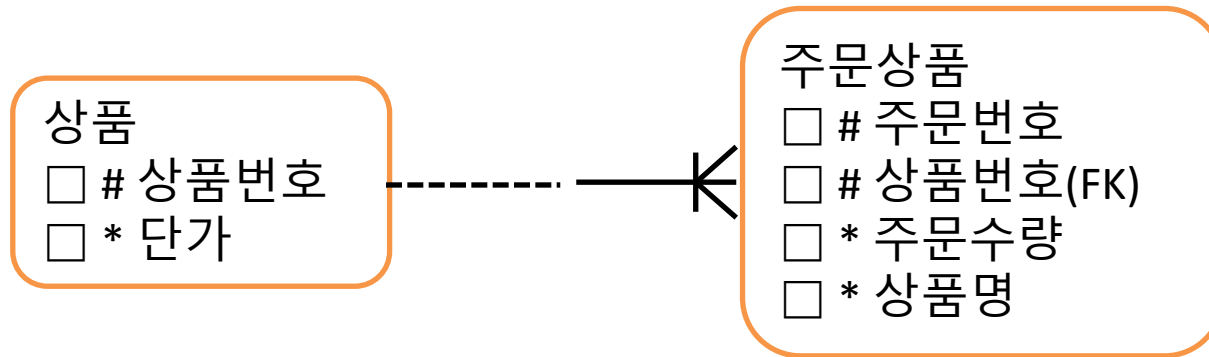
- 아래 엔터티에서 상품명과 단가가 상품번호에만 종속 - 2정규형 위배

주문상품
□ # 주문번호
□ # 상품번호
□ * 상품명
□ * 단가
□ * 주문수량

=> 상품번호를 주 식별자로하는 상품 엔터티를
상위 엔터티로 생성

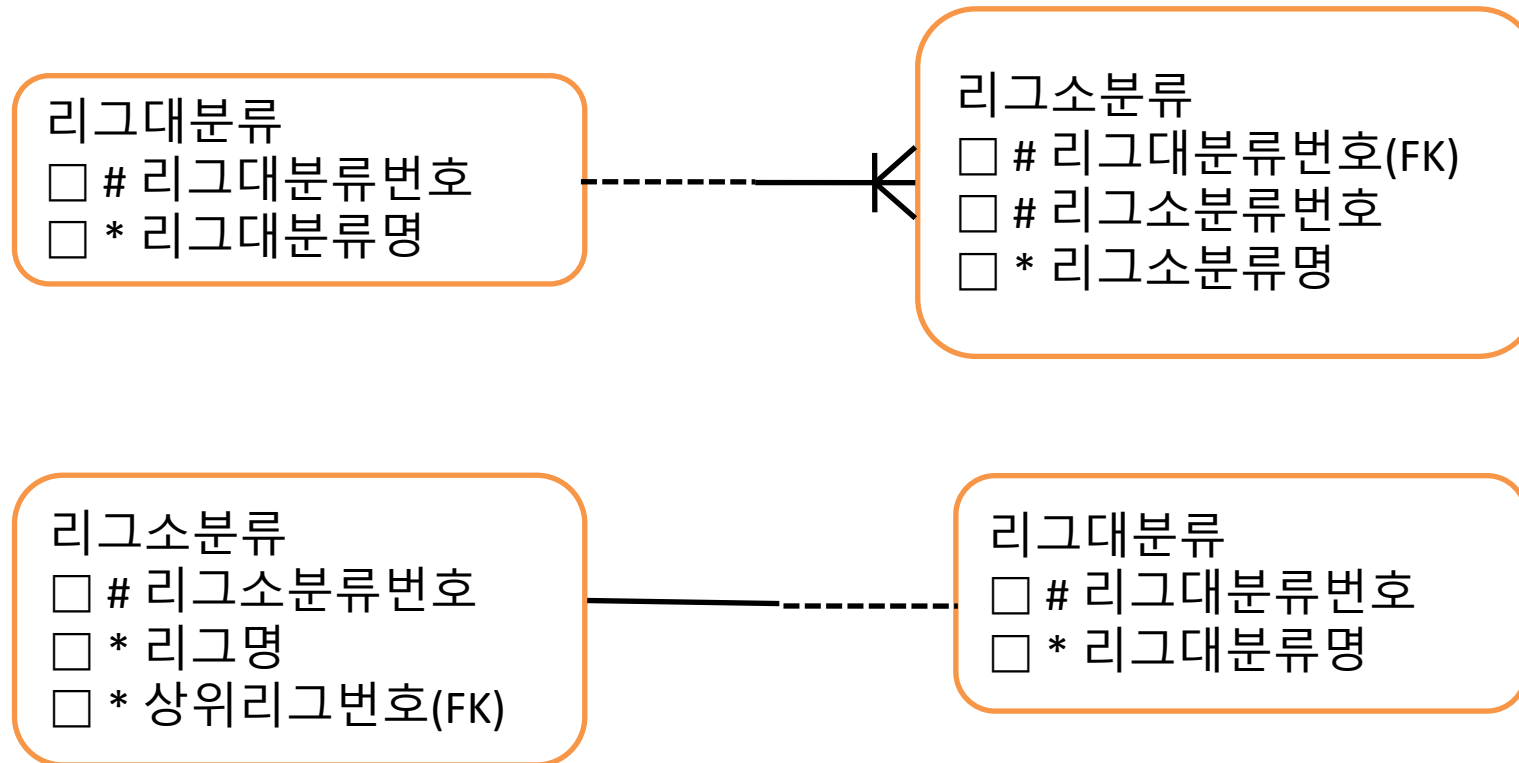


04 논리적 설계



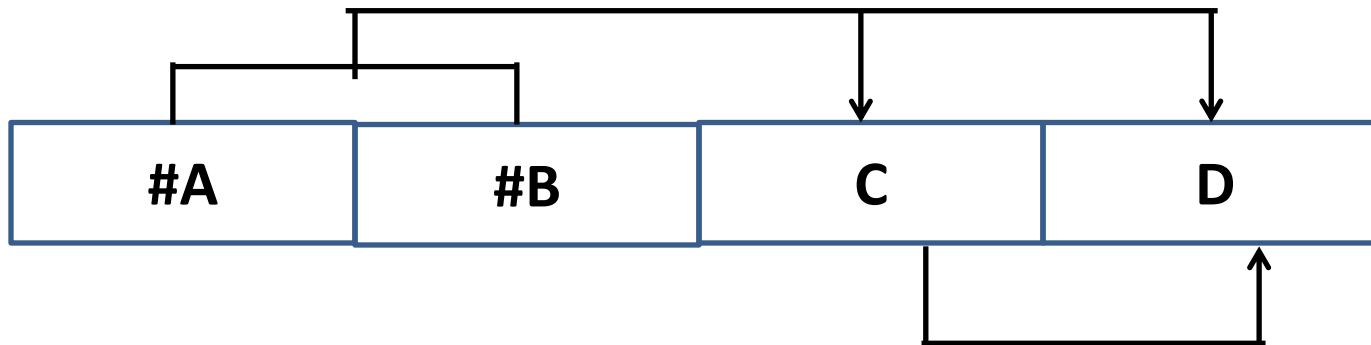
04 논리적 설계

- 대분류, 소분류를 관리하는 모델이다. 2정규화를 수행한 결과를 제시하라

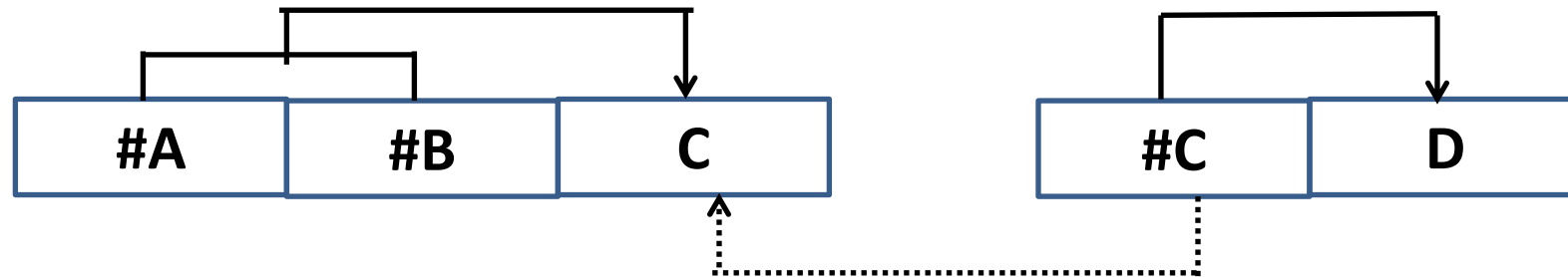


❖ 3 정규화

- 비 식별자(일반 속성)간에 발생하는 이행적 종속성과 관련된 정규화
- $X \rightarrow Y$ 이고 $Y \rightarrow Z$ 이면 $X \rightarrow Z$ 가 성립한다. 이때 Y가 릴레이션 식별자가 아닌 일반 속성일때



04 논리적 설계



(3 정규화를 한 릴레이션)

04 논리적 설계

주문

- ☐ # 주문번호
- ☐ * 고객번호
- ☐ * 고객명
- ☐ * 주문일자
- ☐ * 배송요청일자

고객번호를 알면 고객이름이 결정되
므로 고객명은 고객번호에 종속

(3 정규형을 위배한 모델)

고객

- ☐ # 고객번호
- ☐ * 고객명

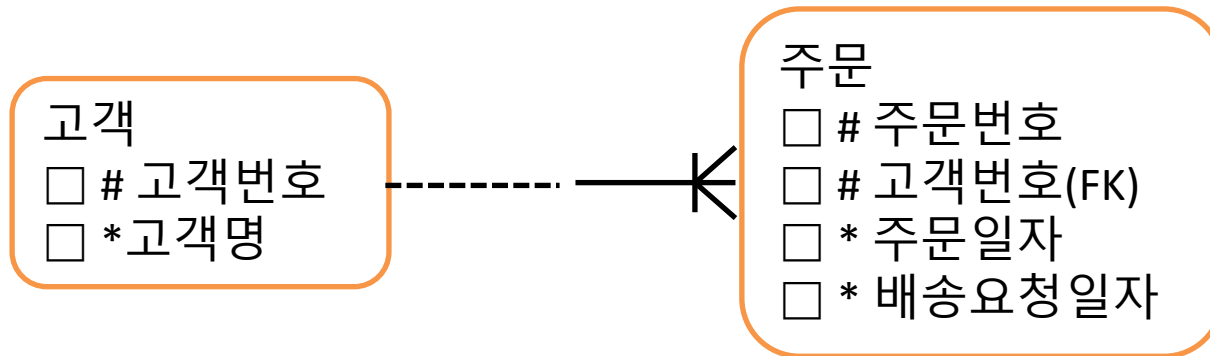
주문

- ☐ # 주문번호
- ☐ # 고객번호(FK)
- ☐ * 주문일자
- ☐ * 배송요청일자

(3 정규화를 적용한 모델)

❖ BC 정규화

- 3 정규형을 보강한 정규형
- 모든 BC 정규형은 모두 3 정규형 릴레이션이지만 3 정규형이 모두 BC 정규형 릴레이션이 아님
- 모든 결정자는 주 식별자이어야 함-릴레이션에 존재하는 종속자는 후보식별자자 아니어야 함



(3 정규화를 적용한 모델)

04 논리적 설계

❖ 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예

- E-R 다이어그램을 릴레이션으로 변환한 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

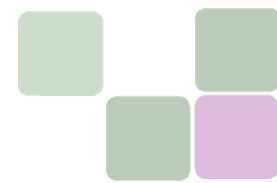
데이터베이스 설계 사례

□기업에서 흔히 볼 수 있는 작은 세계에 관한 요구사항

- ✓ 회사에는 다수의 직원들이 재직
- ✓ 각 직원에 대해서 직원번호(고유함), 이름, 직책, 급여, 주소를 저장. 주소는 시, 구, 동으로 세분하여 나타냄
- ✓ 각 직원은 0명 이상의 부양가족을 가질 수 있음. 한 부양가족은 두 명 이상의 직원에게 속하지 않음. 각 부양가족에 대해서 부양가족의 이름과 성별을 저장
- ✓ 회사는 여러 개의 프로젝트들을 진행. 각 프로젝트에 대해서 프로젝트번호(고유함), 이름, 예산, 프로젝트가 진행되는 위치를 나타냄. 한 프로젝트는 여러 위치에서 진행될 수 있음. 각 프로젝트마다 여러 명의 직원들이 일함. 각 직원은 여러 프로젝트에서 근무할 수 있음. 각 직원이 해당 프로젝트에서 어떤 역할을 수행하고, 얼마 동안 근무해 왔는가를 나타냄. 각 프로젝트마다 한 명의 프로젝트 관리자가 있음. 한 직원은 두 개 이상의 프로젝트의 관리자가 될 수는 없음. 프로젝트 관리자 임무를 시작한 날짜를 기록

데이터베이스 설계 사례

- ✓ 각 사원은 한 부서에만 속함. 각 부서에 대해서 부서번호(고유함), 이름, 부서가 위치한 층을 나타냄
- ✓ 각 프로젝트에는 부품들이 필요. 한 부품이 두 개 이상의 프로젝트에서 사용될 수 있음. 하나의 부품은 다른 여러 개의 부품들로 이루어질 수 있음. 각 부품에 대해서 부품번호(고유함), 이름, 가격, 그 부품이 다른 부품들을 포함하는 경우에는 그 부품들에 관한 정보도 나타냄
- ✓ 각 부품을 공급하는 공급자들이 있음. 한 명의 공급자는 여러 가지 부품들을 공급할 수 있고, 각 부품은 여러 공급자들로부터 공급될 수 있음. 각 공급자에 대해서 공급자번호(고유함), 이름, 신용도를 나타냄. 각 공급자에 대해서 그 공급자가 어떤 부품을 어떤 프로젝트에 얼마나 공급하는가를 나타냄



Thank You

A light gray semi-circle is positioned above the text, and another identical one is positioned below the text, framing the central message.