

# DBMS

## 9일차

강사 : 김근형

## ◆ 설계 순서

- 1. 요구사항 수집 및 분석
- 2. 개념적 설계
- 3. 논리적 설계
- 4. 물리적 설계
- 5. 구현

## ◆ 데이터베이스 설계 중요성

- 가장 중요한 것은 분석
- 내가 구현할 시스템이 무엇인지 정확하게 파악하고 이에 따른 데이터베이스 설계가 이루어져야 함
- 만일, 데이터베이스 설계가 제대로 되어 있지 않은 상태에서 시스템이 구현된다면
- 사소한 문제로 인해 데이터베이스를 수정할 때 어렵게 해결해야 하는 문제점 발생

# ☒ 요구사항 수집 및 분석

## ◆ 목적

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

## ◆ 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

## ◆ 결과물

- 요구 사항 명세서

## ◆ 기능적 요구사항

- 시스템이 무엇을 하는지, 어떤 기능을 하는지 등 기능의 수행과 관련된 요구사항
- 시스템의 입력과 출력으로 무엇이 포함되어야 하는지에 대한 사항
- 시스템이 어떤 데이터를 저장하거나 연산을 수행해야 하는지에 대한 사항
- 시스템이 반드시 수행해야 하는 기능
- 사용자가 시스템을 통해 제공 받기를 원하는 기능

## ◆ 비기능적 요구사항

- 품질이나 제약사항과 관련된 요구사항
- 시스템 장비 구성 요구사항
- 성능 요구사항
- 인터페이스 요구사항
- 데이터를 구축하기 위해 필요한 요구사항
- 테스트, 보안을 위한 요구사항
- 품질 요구 사항 : 가용성, 정합성, 상호 호환성, 대응성, 이식성, 확장성, 보안성 등...
- 프로젝트 관리 요구사항
- 프로젝트 자원 요구사항

## ◆ 예시1 → 보드 구독 커뮤니티 DB

- 보드를 구독하려면 아이디, 구독 여부가 있어야합니다.
- 유저는 여러 보드를 구독할 수 있습니다.
- 보드를 구독했다면 글 작성, 댓글 작성, 좋아요 뿐만 아니라 여러 부가기능을 사용할 수 있습니다.
- 유저는 또한 여러 글, 댓글을 작성할 수 있습니다.
- 보드에 대한 정보 보드설명, 보드추가기능여부를 저장하고 있습니다.
- 유저에 대한 정보 아이디,실명, 비밀번호, 구독여부를 가지고 있습니다.
- 글에 대한 정보 아이디, 글 내용, 글 쓴 날짜, 어떤 보드에 해당하는 지 가지고 있습니다.

## ◆ 예시2 → 커뮤니티 기능을 도입한 마트 관리

- Roach 마트에 가입하려면 고객은 회원 아이디, 비밀번호, 나이, 직업을 입력 해야한다.
- 가입한 회원에게는 등급과 적립금이 부여된다.
- 회원은 회원 아이디로 식별한다.
- 상품에 대한 상품번호, 상품명, 재고량, 단가정보를 유지해야 한다.
- 상품은 상품번호로 식별한다.
- 회원은 여러 상품을 주문할 수 있고, 하나의 상품을 여러 회원이 주문할 수 있다.
- 회원이 상품을 주문하면 주문에 대한 주문번호, 주문수량, 배송지, 주문일자 정보를 유지해야 한다.



## ◆ 예시2 → 커뮤니티 기능을 도입한 마트 관리

- 각 상품은 한 제조업체가 공급하고, 제조업체 하나는 여러 상품을 공급할 수 있다.
- 제조업체가 상품을 공급하면 공급일자와 공급량 정보를 유지해야 한다.
- 제조업체에 대한 제조업체명, 전화번호, 위치, 담당자 정보를 유지해야 한다.
- 제조업체는 제조업체명으로 식별한다.
- 회원은 게시글을 여러개 작성할 수 있고, 게시글 하나는 한 명의 회원만 작성할 수 있다.
- 게시글에 대한 글번호, 글제목, 글내용, 작성일자 정보를 유지해야 한다.
- 게시글은 글 번호로 식별한다.

## ◆ 예시3 → 쇼핑몰 구축을 위한 요구사항

- 회원이 상품을 구매한다.
- 회원은 상품을 카트(장바구니)에 담을 수 있다.
- 회원정보: 회원아이디, 비밀번호, 성명, 주소, 연락처
- 상품: 상품아이디, 상품명, 이미지, 가격
- 회원이 요구하는 배송지에 배송을 지원한다.
- 회원은 0또는 1곳 이상의 배송지를 등록할 수 있다.
- 배송지: 배송지 아이디, 회원아이디, 우편번호, 연락처, 주소, 기본배송지등록여부

## ◆ 예시3 → 쇼핑몰 구축을 위한 요구사항

- 회원은 구매하기 위해 상품을 카트에 담는다.
- 카트(장바구니) 회원아이디, 상품아이디, 수량, 금액, 날짜
- 회원은 구매시 신용카드로 구매한다.
- 신용카드: 신용카드 아이디, 회원 아이디, 카드사, 카드번호, 유효기간
- 회원은 0또는 1장 이상의 신용카드를 등록한다.
- 주문: 회원아이디, 주문 일자, 상품명, 수량, 금액, 배송지 아이디, 카드 아이디

## ◆ 예시4 → 대학 요구사항

- 교수(Professor)는 아이디(ssn), 이름(name), 나이(age), 직위(rank), 연구 분야(speciality)를 가진다.
- 학과(Department)에는 학과번호(dno), 학과이름(dname), 학과사무실(office)이 있다.
- 대학원생(Graduate)은 아이디(ssn), 이름(name), 나이(age), 학위과정(deg\_prog, 석사/박사)을 가진다.
- 과제(Project)는 과제번호(pid), 지원기관(sponsor), 개시일(start\_date), 종료일(end\_date), 예산액 (budget)이 있다.
- 학과마다 그 학과를 운영(run)하는 교수(학과장이라고 한다)가 한 명씩 있다.

## ◆ 예시4 → 대학 요구사항

- 한 교수가 여러 학과에서 근무(work-dept)할 수 있는데, 이 때 각 학과별로 참여 백분율(pct\_time)이 기록된다.
- 대학원생에게는 학위 과정을 밟을 전공학과(major)가 하나씩 있다.
- 대학원생에게는 어떤 과목을 들으면 좋을지 조언(advisor)해주는 선임대학원생(학생조언자라고 한다)이 있다.
- 과제는 한 교수(연구책임자라고 한다)에 의해 관리(manage)된다.
- 과제는 한 사람이상의 교수(공동연구책임자라고 한다)에 의해 수행(work-in)된다.
- 한 과제는 한 명이상의 대학원생(연구조교라고 한다)에 의해 수행(work-prog)된다.

## ◆ 예시5 → 항공사 예약 프로그램의 요구사항

- 한빛 항공사에 회원으로 가입하려면 회원 아이디, 비밀번호, 성명, 신용카드 정보를 입력해야 한다
- 회원의 신용카드 정보는 여러 개를 저장할 수 있는데, 세부적으로는 신용카드번호, 유효기간을 저장할 수 있다
- 한빛 항공사에서는 보유한 비행기에 대해 비행기번호, 출발날짜, 출발시간 정보를 저장하고 있다
- 한빛 항공사에서는 좌석에 대한 좌석번호, 등급 정보를 저장하고 있다
- 회원은 좌석을 예약하는데, 회원 한 명은 좌석을 하나만 예약할 수 있고, 한 좌석은 회원 한 명만 예약할 수 있다
- 비행기에는 좌석이 존재하는데, 비행기 하나에는 좌석이 여러 개 존재할 수 있고
- 한 좌석은 반드시 하나의 비행기에만 존재해야 한다.
- 좌석은 비행기가 없으면 의미가 없다.

# 요구사항 수집 및 분석

## ◆ 요구사항 분석을 마치면

- 요구사항 명세서 작성
- 요구사항 내에 존재하는 개체와 속성 추출

RQ-ID	화면명	요구사항명	요구사항 내용	날짜	작성자	진행사항	버전명
RQ-0001	관리자	통계	매물의 [조회수], [장비등록일], [장비판매일]을 기록한다.	2/3	송미경	반영	0.2.0
RQ-0002	공통	오른쪽 영역	등급 모델 표시를 해준다.	12/28	송미경	다음 버전	-
RQ-0003	공통	오른쪽 영역	최근 본 매물이 화면 오른쪽에 있다. (아래) - 3개씩 5페이지 - 총 15개	12/28	송미경	반영	0.5.5
RQ-0004	공통	오른쪽 영역	많이 본 매물이 화면 오른쪽에 있다. (위) - 3개씩 1페이지 - 총 3개 - 판매완료된 매물은 제외한다. - 7일 기준이며, 월요일 날 기록을 리셋시킨다. - 기록이 없는 경우, 랜덤으로 3개 띄운다. - 많이 본 매물에서 선택시 카운트 제거 - 조회수 기반이다.	12/28	송미경	반영	0.5.5
RQ-0005	공통	상단 메뉴 구성	딜러와 관련된 메뉴를 구성한다. 방법1) 상단을 사용자를 위한 메뉴로만 구성, 푸터에 딜러관련 메뉴 배치 -> 웹사이트 이용자 중에서 사용자가 다수일 때 선택 방법2) 상단에 매물등록 등을 배치 -> 사용자에게 여기 딜러가 있다는것을 암시	12/28	송미경	미반영	-
RQ-0006	공통	명칭 변경	추천 매물 -> 스페셜 매물, 일반 매물 -> 추천 매물로 변경한다.	1/2	송미경	반영	0.4.0
RQ-0007	공통	명칭 변경	딜러 -> 판매자 제휴딜러 -> 제휴업체	1/9	송미경	반영	0.5.0
RQ-0008	공통	마이페이지	사용자 마이페이지 - 판매자 마이페이지와 동일	1/9	송미경	반영	0.5.0
RQ-0101	인트로	화면 개설	[인트로] 화면이 필요하다. - 중고장비매매서비스, 장치매칭서비스 - 예시 : <a href="http://line25.com/articles/interesting-web-design-trend-vertical-split-layouts">http://line25.com/articles/interesting-web-design-trend-vertical-split-layouts</a>	12/16	송미경	반영	0.2.0
RQ-0102	인트로	언어	추후 다국어를 지원할 계획이다.	12/16	송미경	다음 버전	-

# 개념적 설계

## ◆ 목적

- DBMS에 독립적인 개념적 스키마 설계
- 요구사항 분석 결과물을 개념적 데이터 모델을 이용해 개념적 구조로 표현
- 일반적으로 E-R 모델을 이용

## ◆ 주요 작업

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

## ◆ 작업 과정

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

## ◆ 결과물

- E-R 다이어그램



## ◆ 개체

- 저장할 만한 가치가 있는 주요 데이터를 가진 사람이나 사물 등..
- EX) 병원 DB 개발에 필요한 개체
  - 병원 운영에 필요한 사람 : 의사, 환자, 간호사 등..
  - 병원 운영에 필요한 사물 : 병실, 수술실, 의료 장비 등...
- 개체 추출 방법
  - 요구 사항 문장에서 업무와 관련이 깊은 의미 있는 명사를 찾기
  - 업무와 관련이 적은 일반적으로 광범위한 의미의 명사는 제외
  - 의미가 같은 명사가 여러 개일 경우는 대표 명사 하나만 선택
  - 찾아낸 명사를 개체와 속성으로 분류

# ☐ 개념적 설계 – 개체와 속성 추출

## ◆ 개체 추출

- 개체 : 회원
- 회원 개체의 속성 : 회원 아이디, 비밀번호, 이름, 나이, 직업, 등급, 적립금
- 회원 개체의 키 속성 : 회원 아이디

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

## ◆ 개체 추출

- 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.

# 개념적 설계 – 개체와 속성 추출

## ◆ 개체 추출

개체	속성
회원	<u>회원 아이디</u> , 비밀번호, 이름, 나이, 직업, 등급, 적립금
상품	<u>상품번호</u> , 상품명, 재고량, 단가
제조업체	<u>제조업체명</u> , 전화번호, 위치, 담당자
게시글	<u>글 번호</u> , 글 제목, 글 내용, 작성일자

## ◆ 개체의 E-R 다이어그램

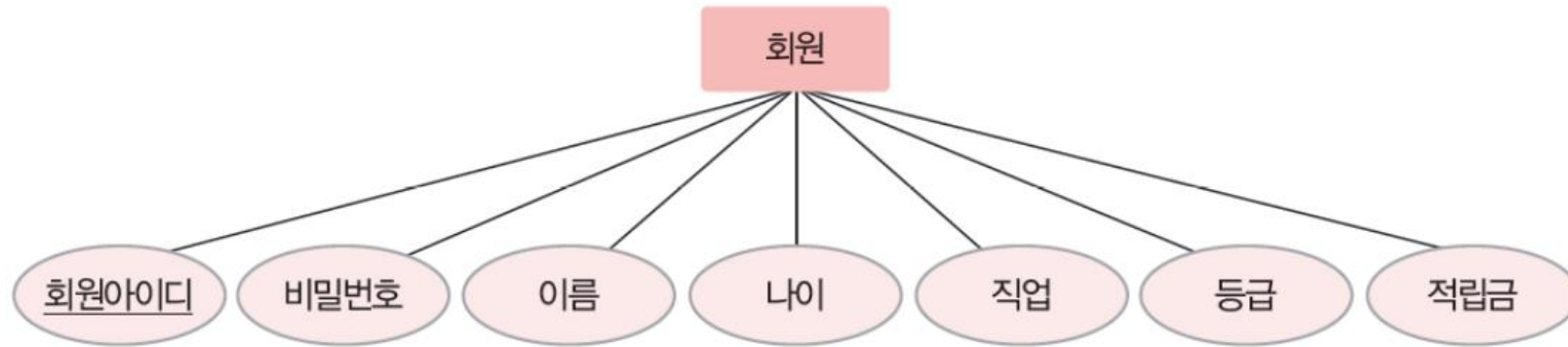


그림 8-13 회원 개체의 E-R 다이어그램

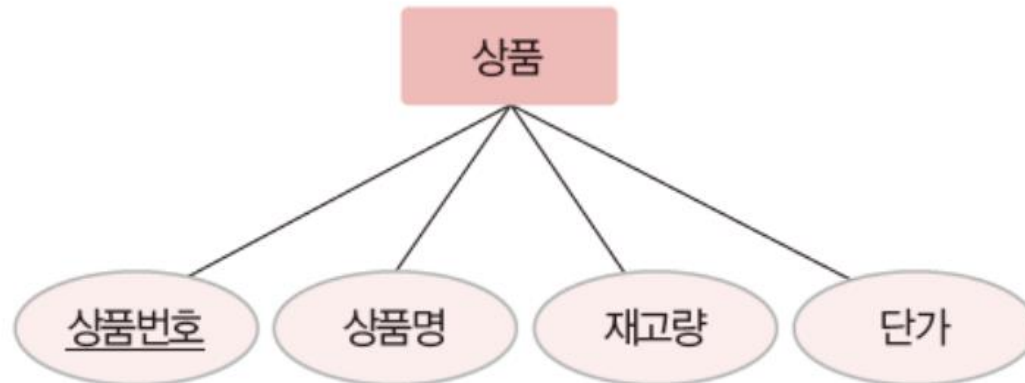


그림 8-14 상품 개체의 E-R 다이어그램

## ◆ 개체의 E-R 다이어그램

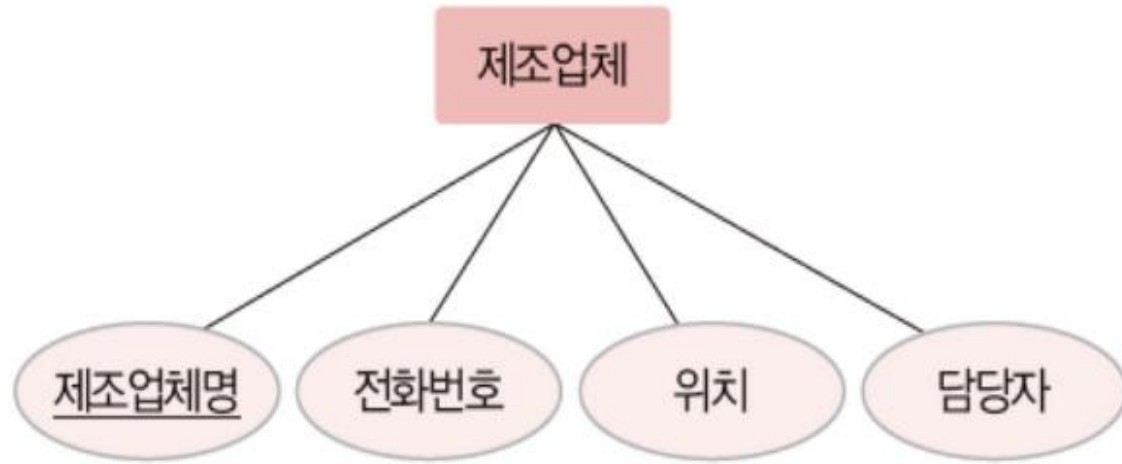


그림 8-15 제조업체 개체의 E-R 다이어그램

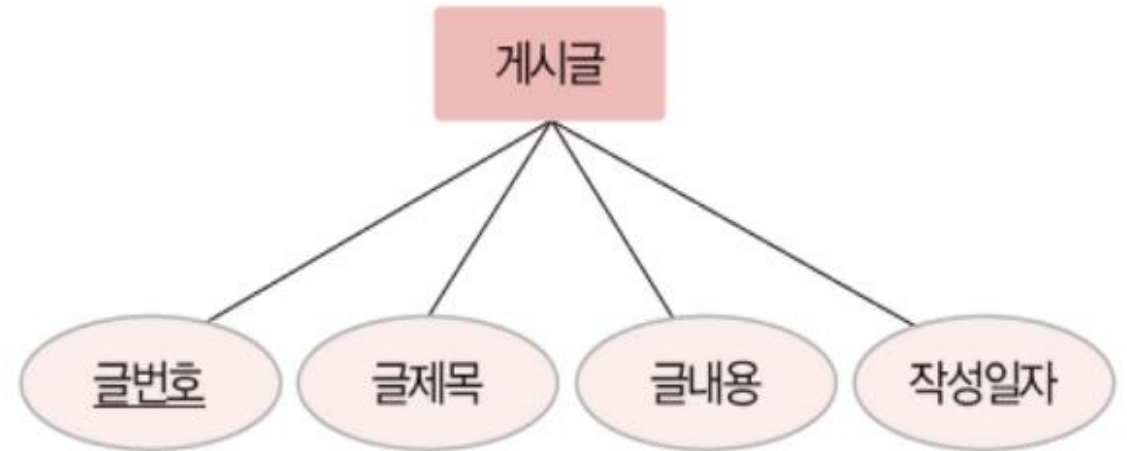


그림 8-16 게시글 개체의 E-R 다이어그램

## ◆ 관계

- 개체 간의 의미 있는 연관성
- 관계 추출 방법
  - 요구 사항 문장에서 개체 간의 연관성을 의미 있게 표현한 동사를 찾아야 함
  - 의미가 같은 동사가 여러 개일 경우는 대표 동사 하나만 선택
  - 찾아낸 관계에 대해 매핑 카디널리티와 참여 특성을 결정
  - 매핑 카디널리티 : 일대일(1:1), 일대다(1:N), 다대다(N:M)
  - 참여 특성 : 필수적 참여, 선택적 참여

## ◆ 관계 추출

- “입력해야 한다”는 개체와 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

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



## ◆ 관계 추출

- 관계 : 주문
- "회원" 개체와 "상품" 개체가 맺는 다대다(M:N) 관계
- "회원" 개체는 관계에 선택적으로 참여
- "상품" 개체는 관계에 선택적으로 참여
- "주문" 관계의 속성 : 주문 번호, 주문 수량, 배송지, 주문일자

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.

## ◆ 관계 추출

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

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

## ◆ 관계 추출

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

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

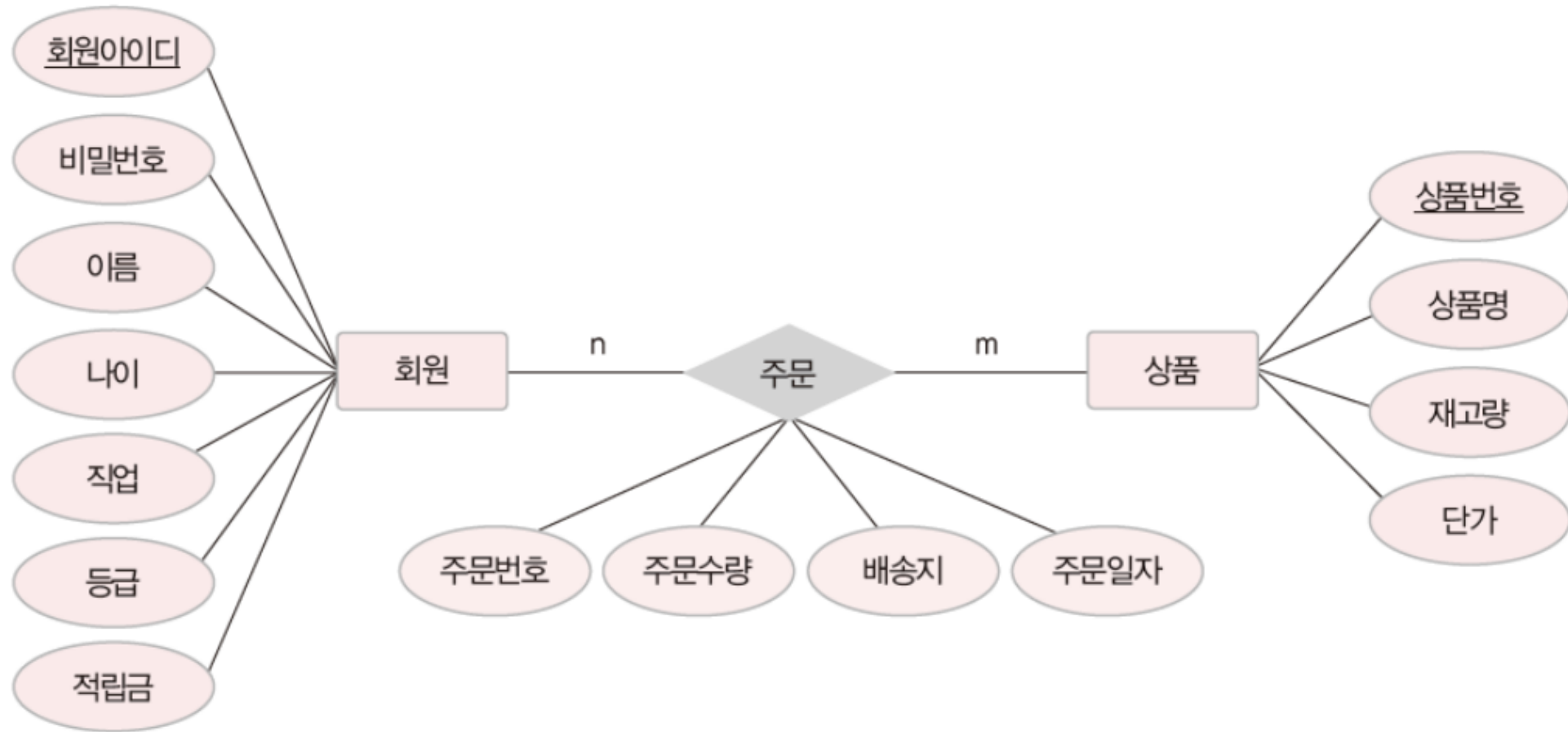
# ☐ 개념적 설계 – 관계 추출

- 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.

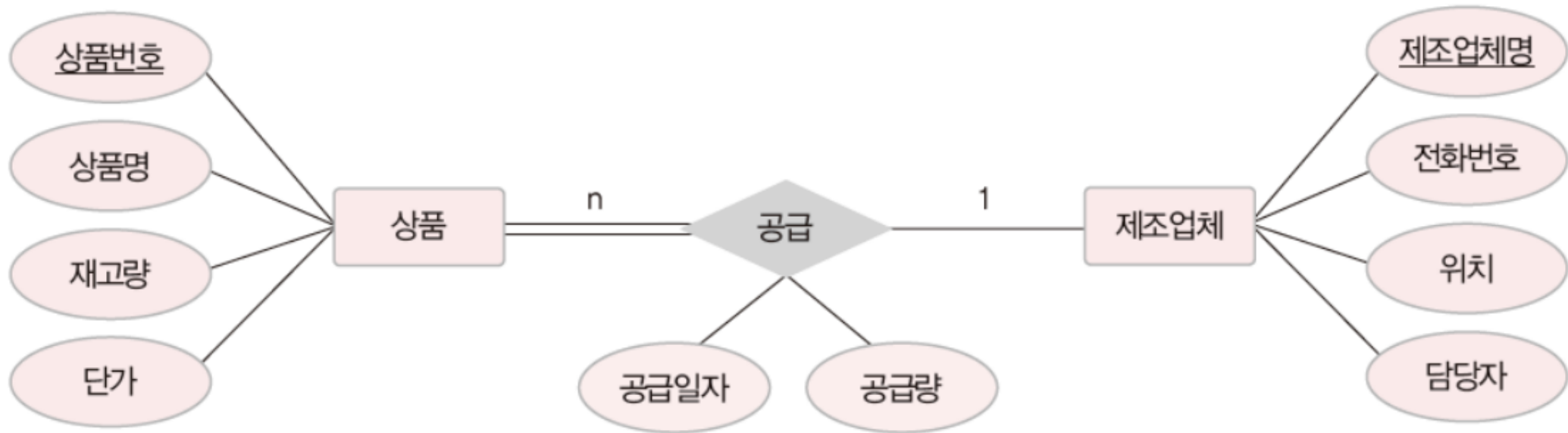
## 개념적 설계 - 관계 추출

관계	관계를 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	

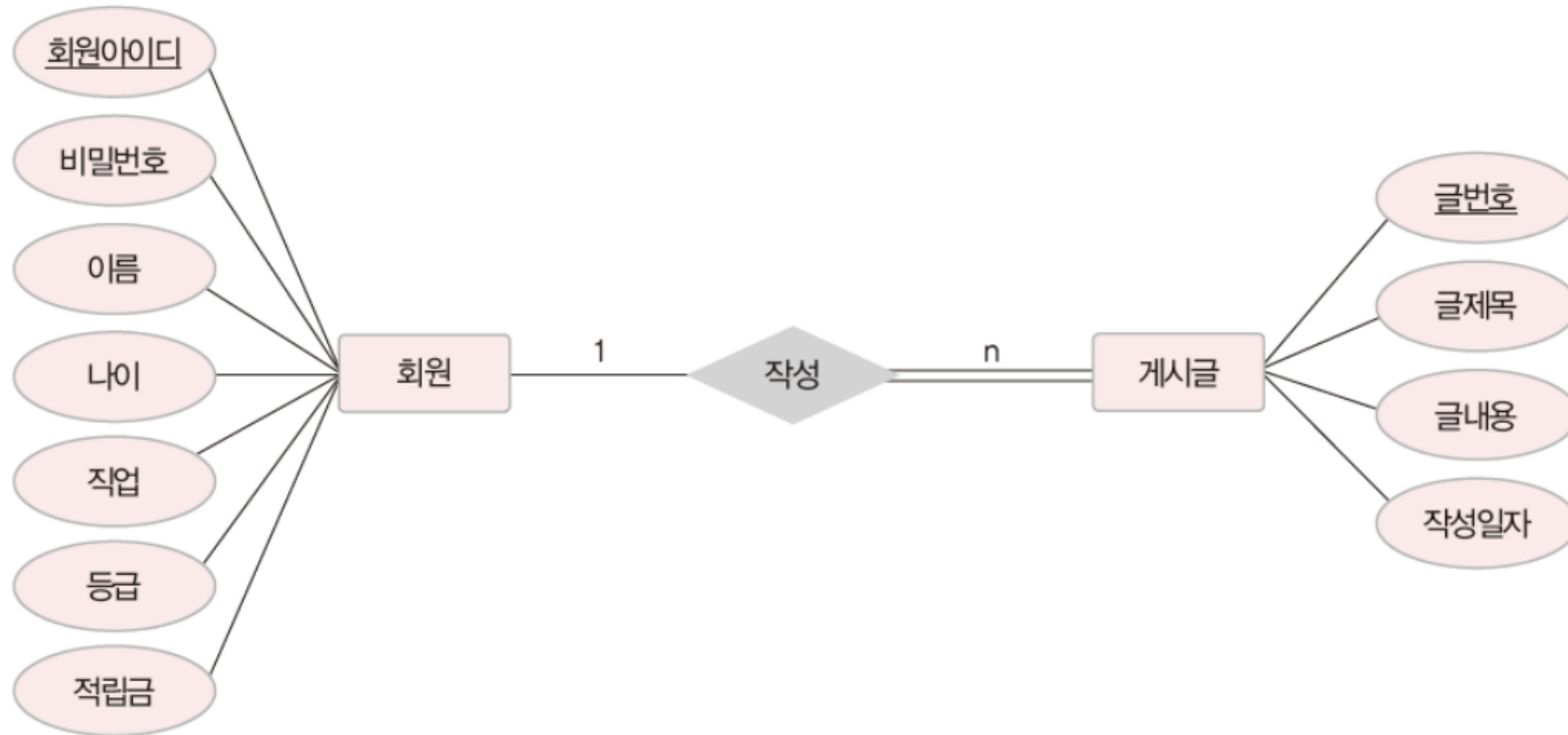
## ◆ 주문 속성의 ERD



## ◆ 공급 관계의 ERD

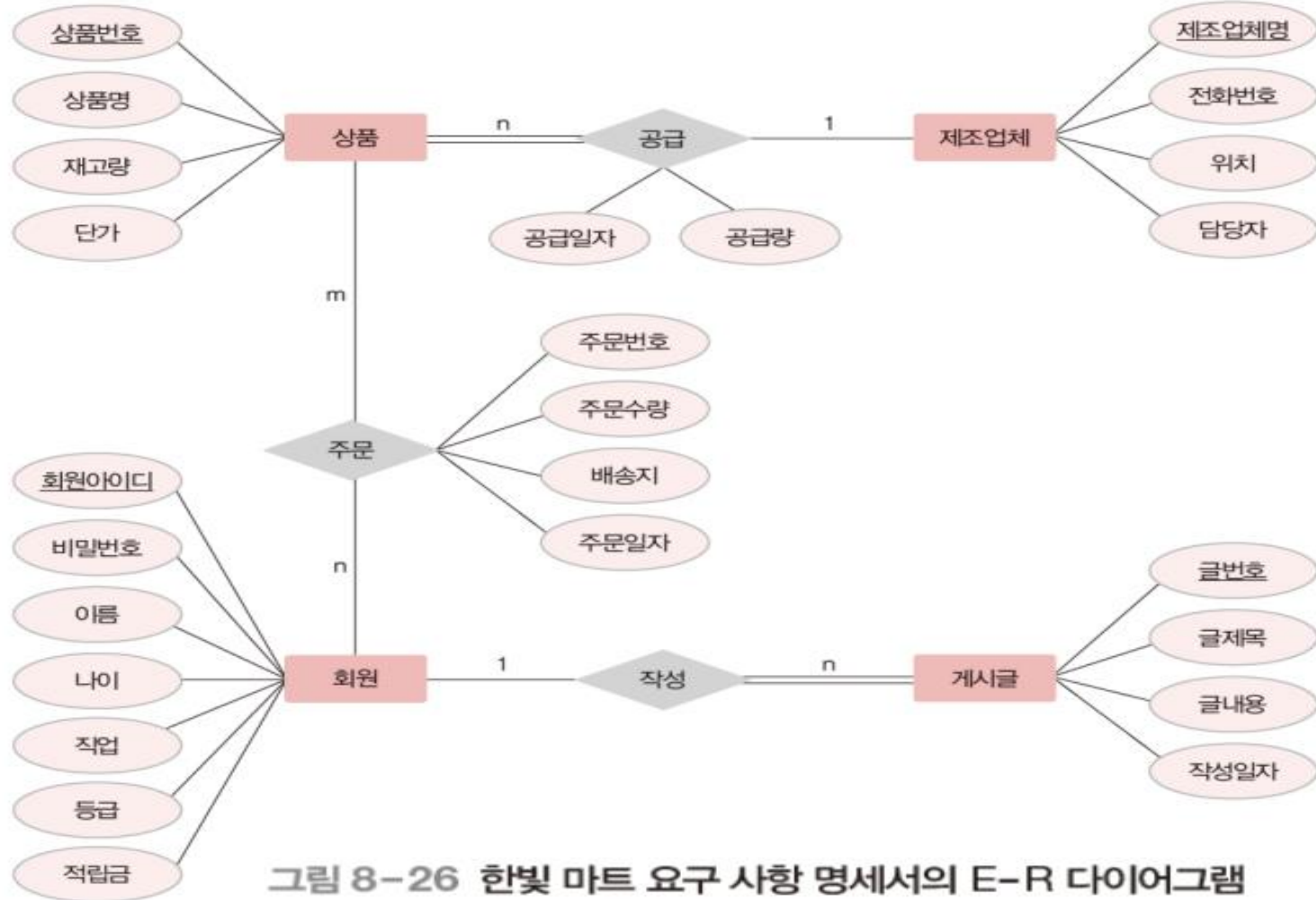


## ◆ 작성 관계의 ERD

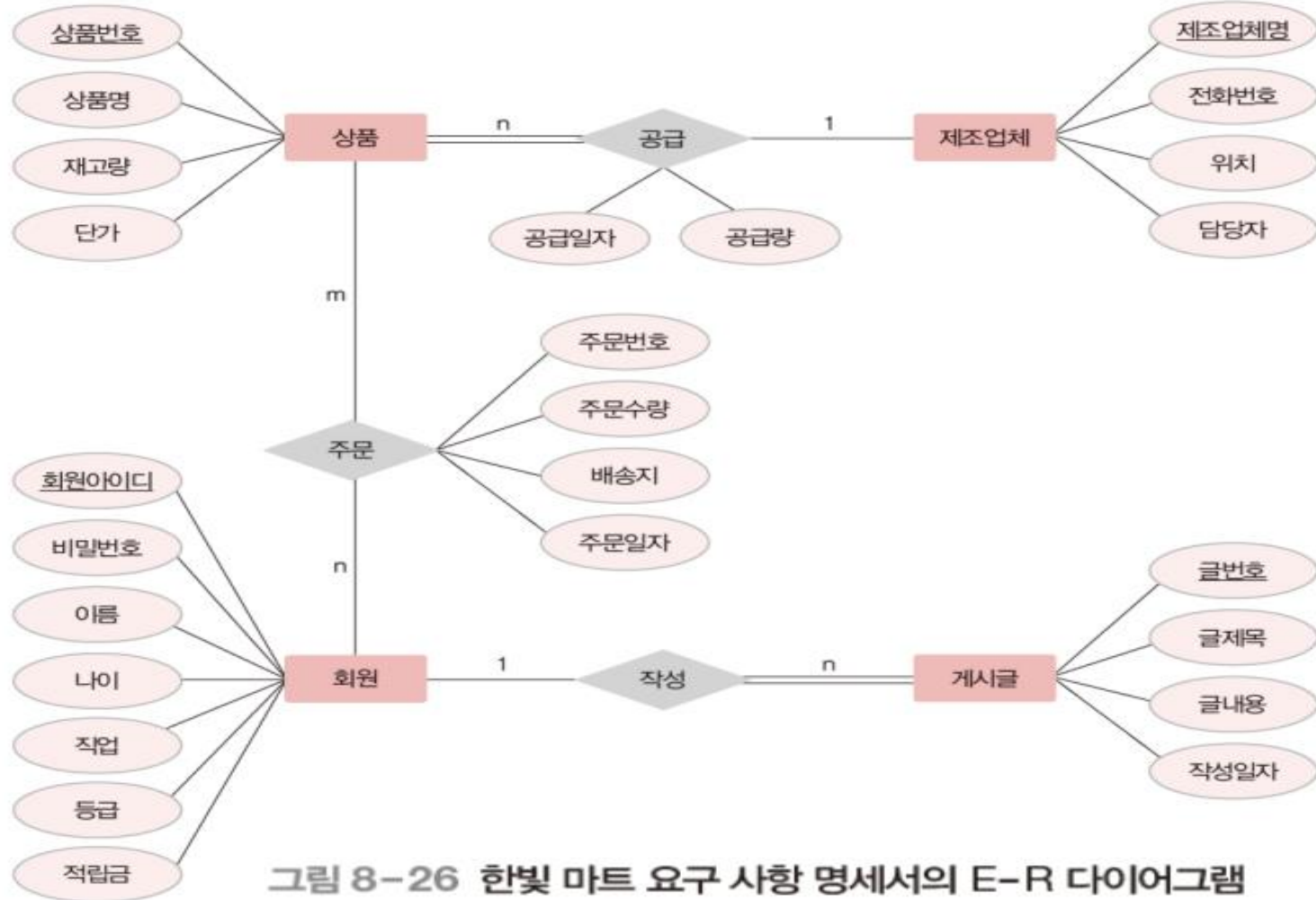




## ◆ 전체 ERD



## ◆ 전체 ERD



# ☐ 논리적 설계

## ◆ 목적

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

## ◆ 주요 작업

- ERD(개념적 설계의 결과물)를 릴레이션 스키마로 변환
- 변환 후 속성의 데이터 타입, 길이, 널 값 허용 여부, 기본 값(디폴트), 제약 조건 등을 세부적으로 결정하고 결과를 문서화

## ◆ 결과물

- 논리적 스키마 : 릴레이션 스키마

## ◆ ERD를 릴레이션 스키마로 변환하는 규칙

- 규칙1) 모든 개체는 릴레이션으로 변환
  - 규칙2) 다대다(N:M) 관계는 릴레이션으로 변환
  - 규칙3) 일대다(1:N) 관계는 외래키로 표현
  - 규칙4) 일대일(1:1) 관계는 외래키로 표현
  - 규칙5) 다중 값 속성은 릴레이션으로 변환
- 
- 변환 규칙을 순서대로 적용하되, 해당되지 않는 규칙은 제외

# 🖥 논리적 설계 - 규칙1(모든 개체 → 릴레이션)

## ◆ ERD의 각 개체를 하나의 릴레이션으로 변환

- 개체의 이름 → 릴레이션 이름
- 개체의 속성 → 릴레이션의 속성
- 개체의 키 속성 → 릴레이션의 기본 키
- 개체의 속성이 복합 속성인 경우에는 복합 속성을 구성하고 있는 단순 속성만 릴레이션의 속성으로 변환

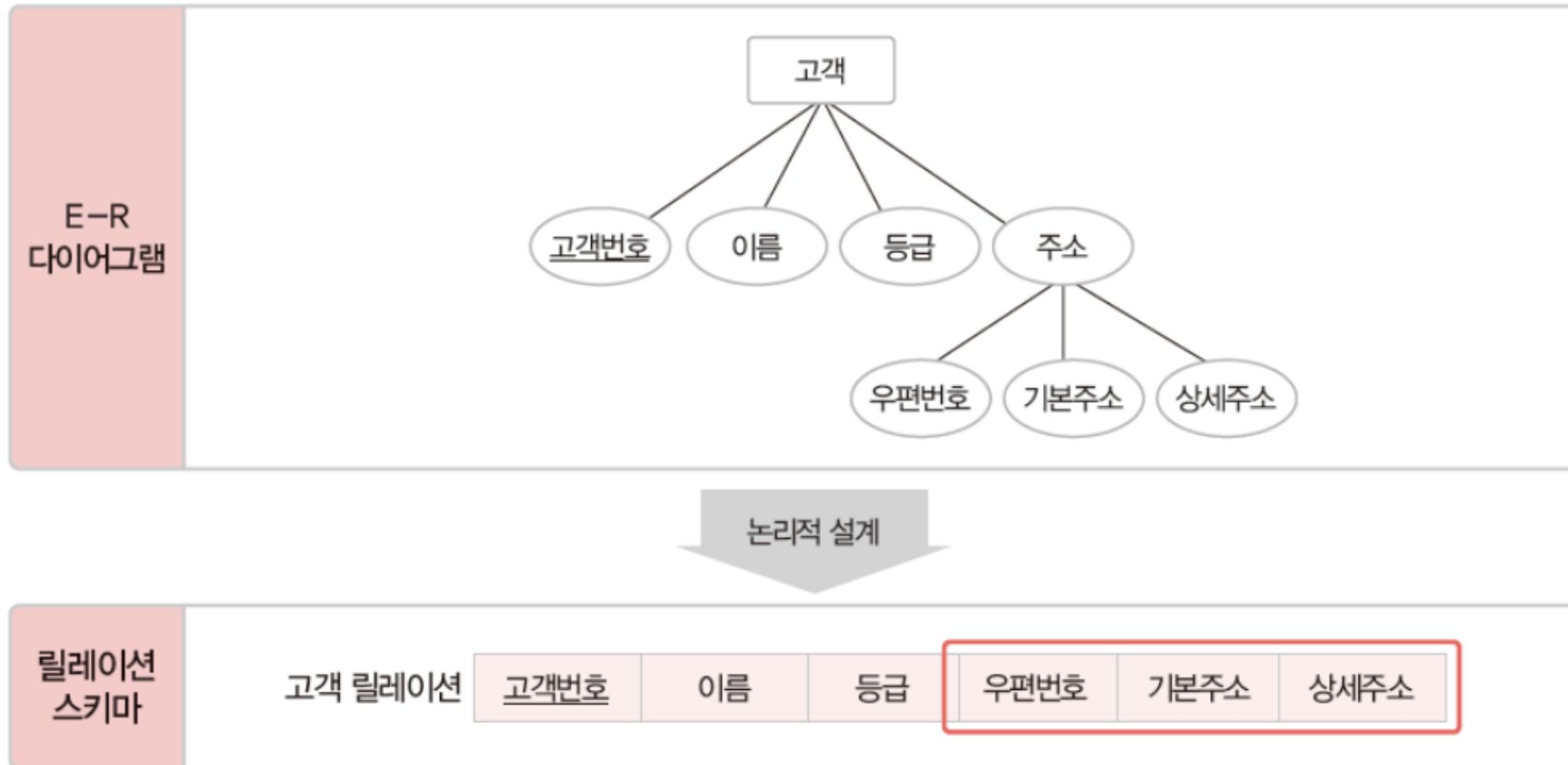
# ☐ 논리적 설계 - 규칙1(모든 개체 → 릴레이션)

## ◆ ERD의 각 개체를 하나의 릴레이션으로 변환



# 논리적 설계 - 규칙1(모든 개체 → 릴레이션)

## ◆ ERD의 각 개체를 하나의 릴레이션으로 변환



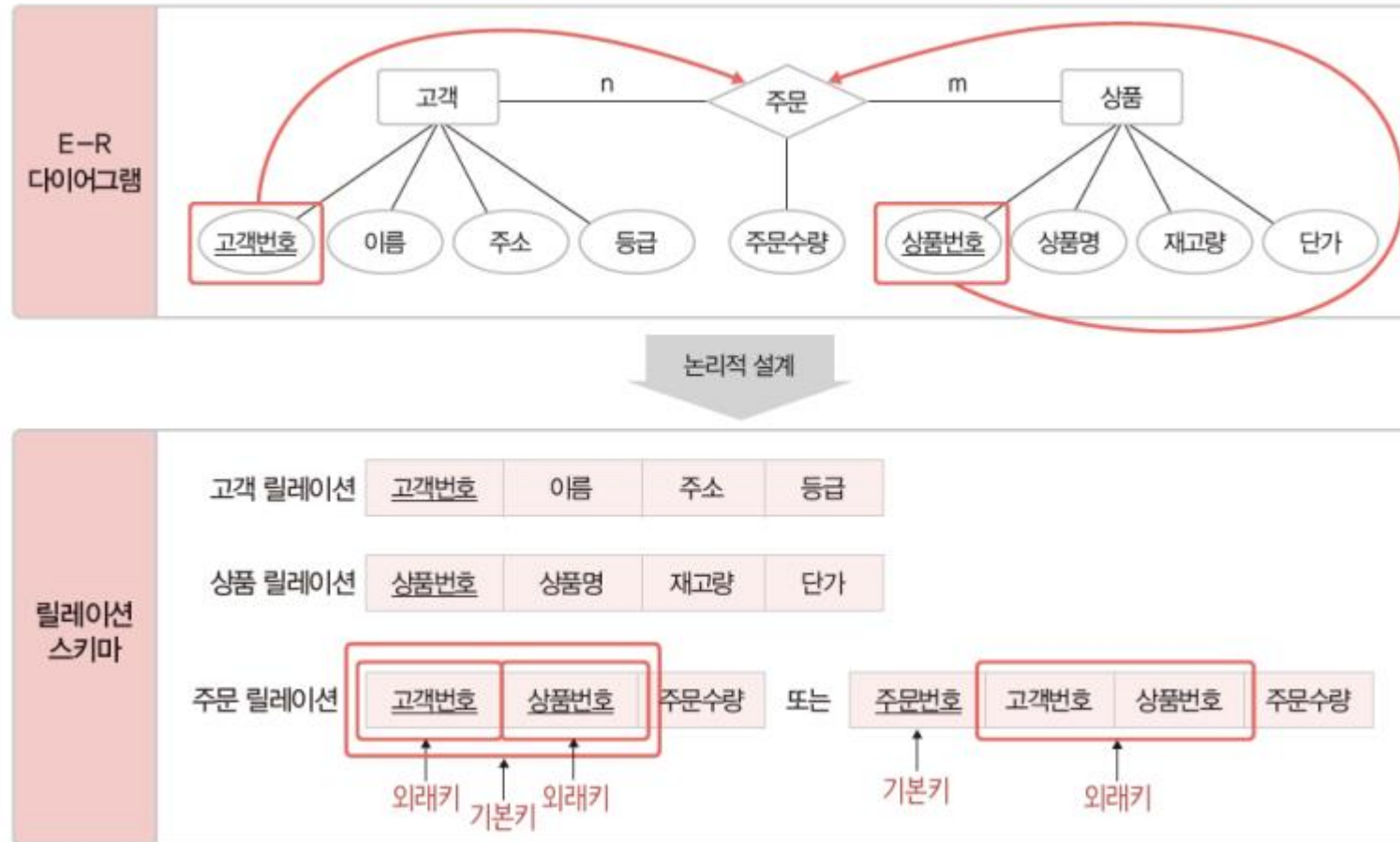
## ◆ ERD 다대다 관계를 하나의 릴레이션으로 변환

- 관계의 이름 → 릴레이션 이름
- 관계의 속성 → 릴레이션의 속성
- 관계에 참여하는 개체를 규칙1에 따라 릴레이션으로 변환한 후 이 릴레이션의 기본키를 관계 릴레이션에 포함시켜 외래키를 지정하고, 외래키들을 조합하여 관계 릴레이션의 기본키로 설정



# 논리적 설계 - 규칙2(다대다 관계 → 릴레이션)

## ◆ ERD 다대다 관계를 하나의 릴레이션으로 변환

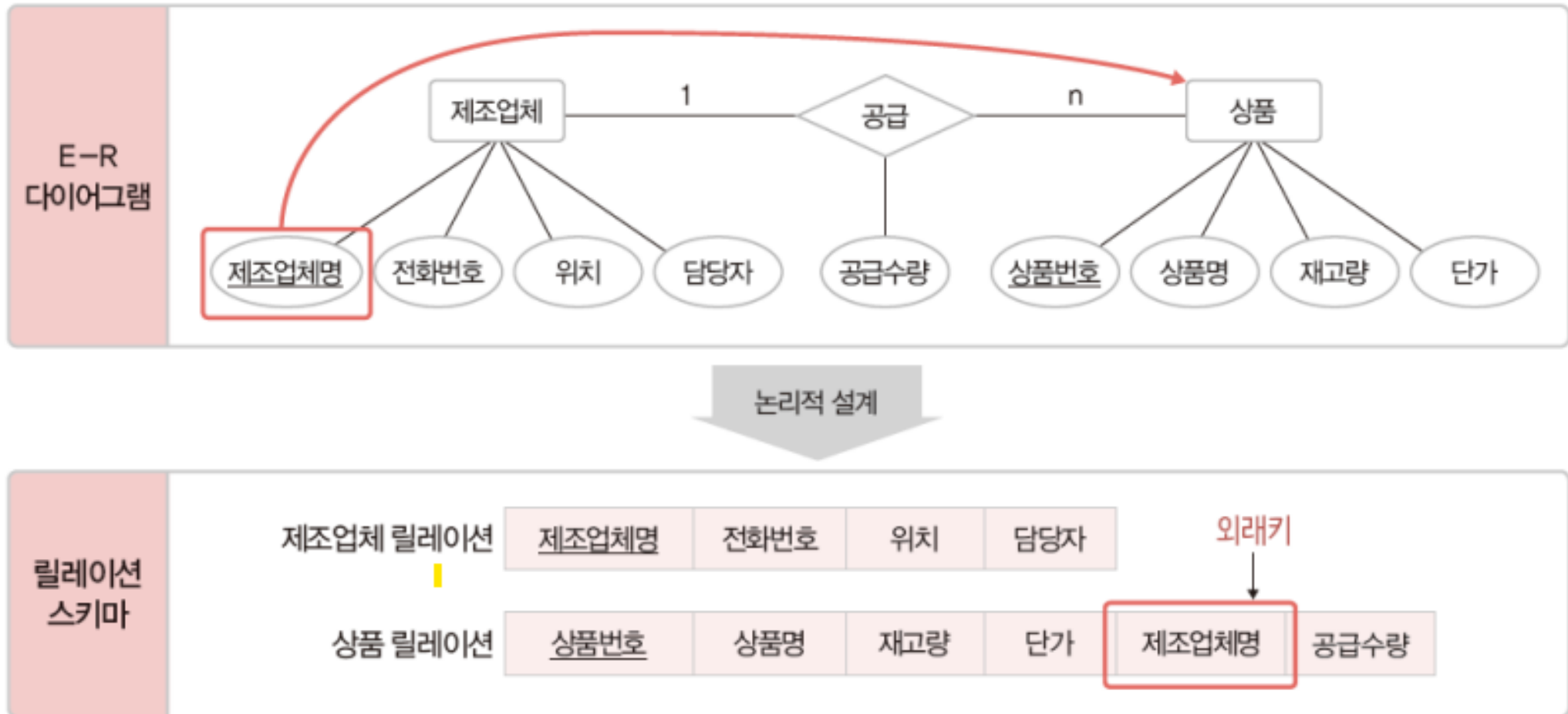


## ◆ ERD의 일대다 관계는 외래키로만 표현

- 규칙 3-1 일반적인 일대다 관계는 외래키로 표현
  - 일대다 관계에서 1측 개체 릴레이션의 기본키를 N측 개체 릴레이션에 포함시켜 외래키로 지정
  - 관계의 속성들도 N측 개체 릴레이션에 포함시킴
- 규칙 3-2 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키로 지정
  - 일대다 관계에서 1측 개체 릴레이션의 기본키를 N측 개체 릴레이션에 포함시켜 외래키로 지정
  - 관계의 속성들도 N측 개체 릴레이션에 포함시킴
  - N측 개체 릴레이션은 외래키를 포함하여 기본키를 지정
- 약한 개체는 강한 개체에 따라 존재 여부가 결정되므로 강한 개체의 기본키를 이용해 식별

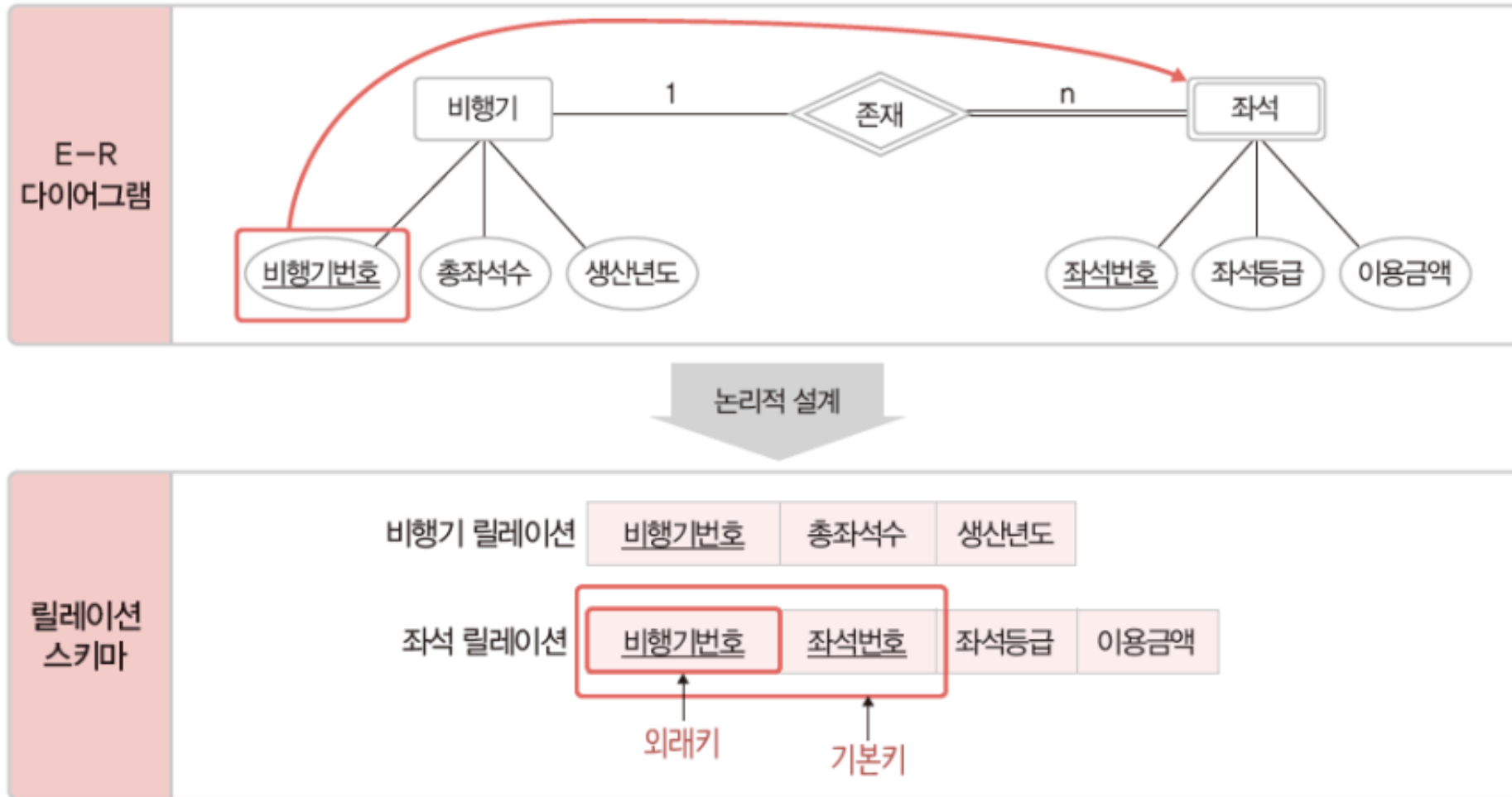
# 논리적 설계 - 규칙3(일대다 관계 → 외래키)

## ◆ 규칙 3-1) 일반적인 개체가 참여하는 일대다 관계를 외래키로 표현



# 논리적 설계 - 규칙3(일대다 관계 → 외래키)

## ◆ 규칙 3-2) 약한 개체가 참여하는 일대다 관계를 외래키로 표현

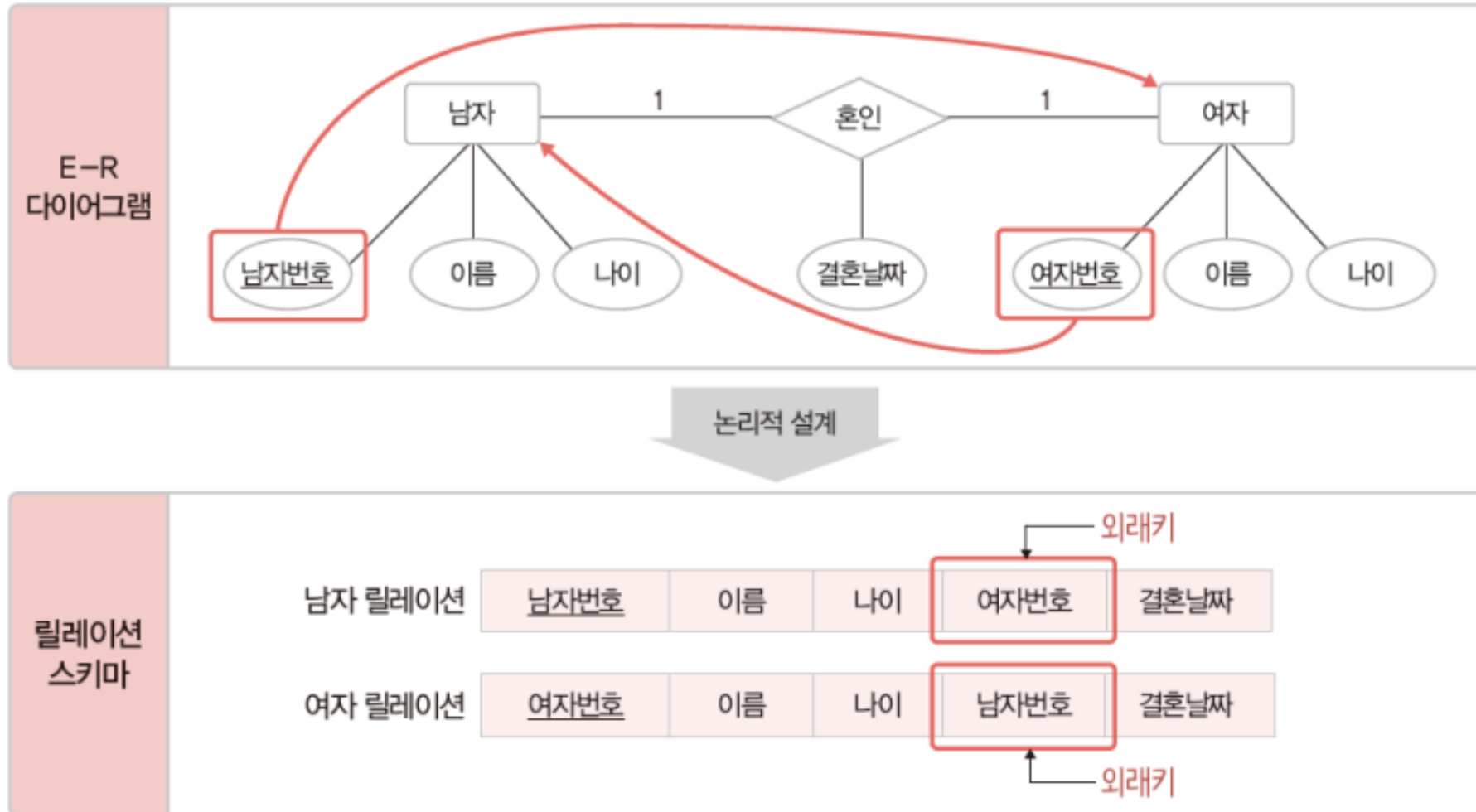


## ◆ ERD의 일대일 관계는 외래키로만 표현

- 규칙 4-1 일반적인 일대일 관계는 외래키를 서로 주고 받음
  - 관계에 참여하는 개체 릴레이션들이 서로의 기본키를 주고 받아 외래키로 지정
  - 관계의 속성들도 모든 개체 릴레이션에 포함시킴
  - 불필요한 데이터 중복이 발생할 수 있음
- 규칙 4-2 일대일 관계에 필수적으로 참여하는 개체의 릴레이션만 외래키를 받음
  - 관계에 필수적으로 참여하는 개체 릴레이션만 외래키를 포함시킴
  - 관계의 속성들은 관계에 필수적으로 참여하는 개체 릴레이션에 포함시킴
- 규칙 4-3 모든 개체가 일대일 관계에 필수적으로 참여하면 릴레이션 하나로 합침
  - 관계에 참여하는 관계 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

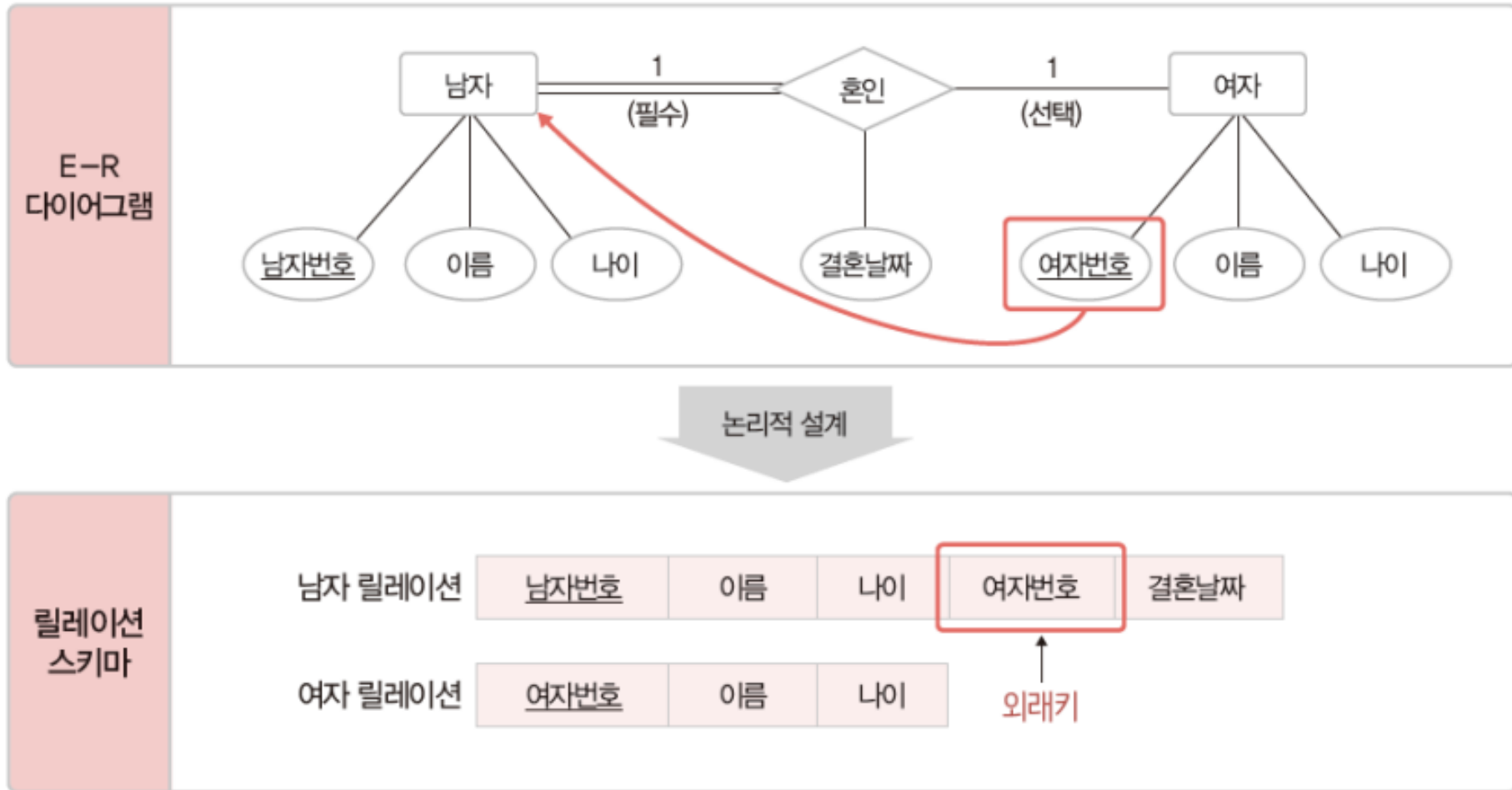
# 논리적 설계 - 규칙4(일대일 관계 → 외래키)

## ◆ 규칙 4-1) 일반적인 일대일 관계를 외래키로 표현



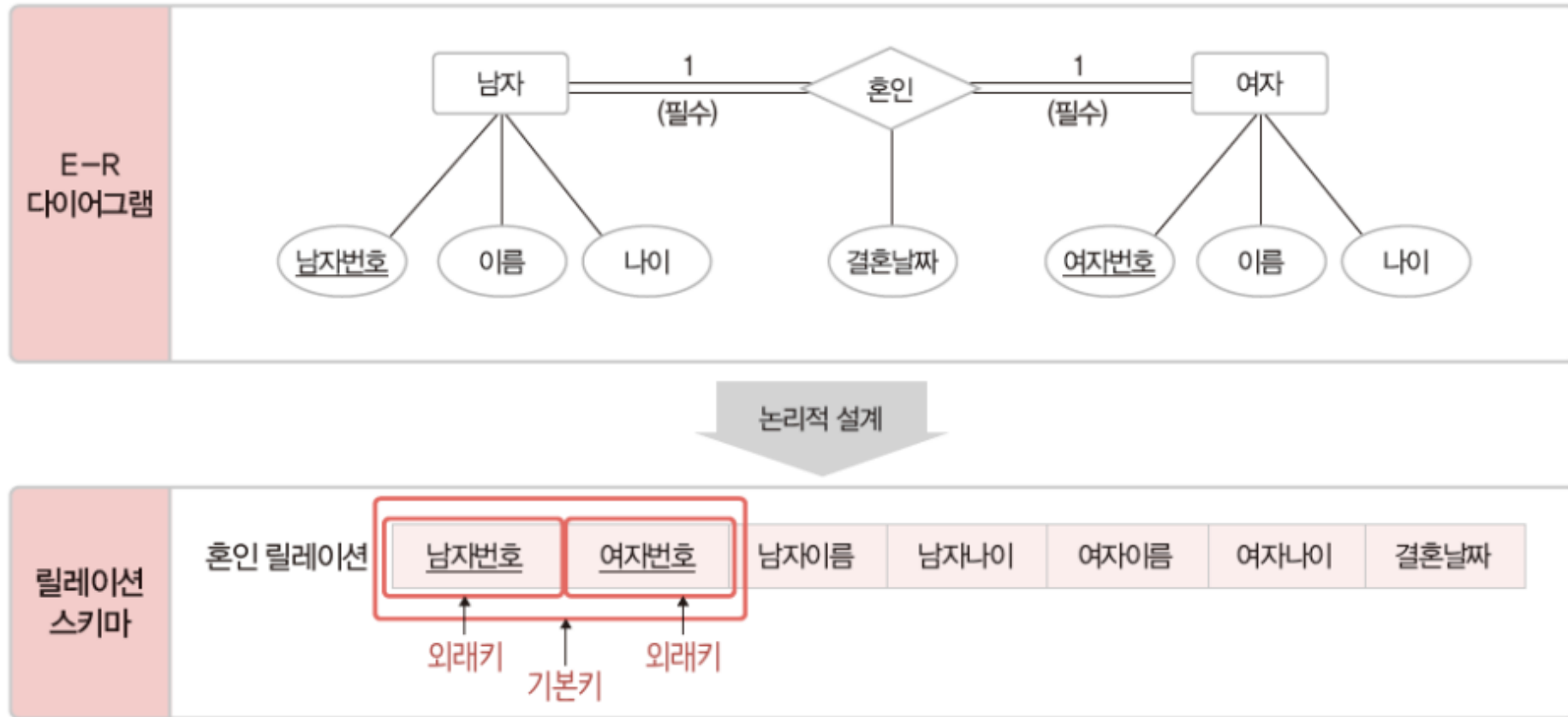
# 논리적 설계 - 규칙4(일대일 관계 → 외래키)

## ◆ 규칙 4-2) 일대일 관계에 필수적으로 참여하는 개체의 릴레이션



# 논리적 설계 - 규칙4(일대일 관계 → 외래키)

## ◆ 규칙 4-3) 일대일 관계에 모든 개체가 필수적으로 참여하는 릴레이



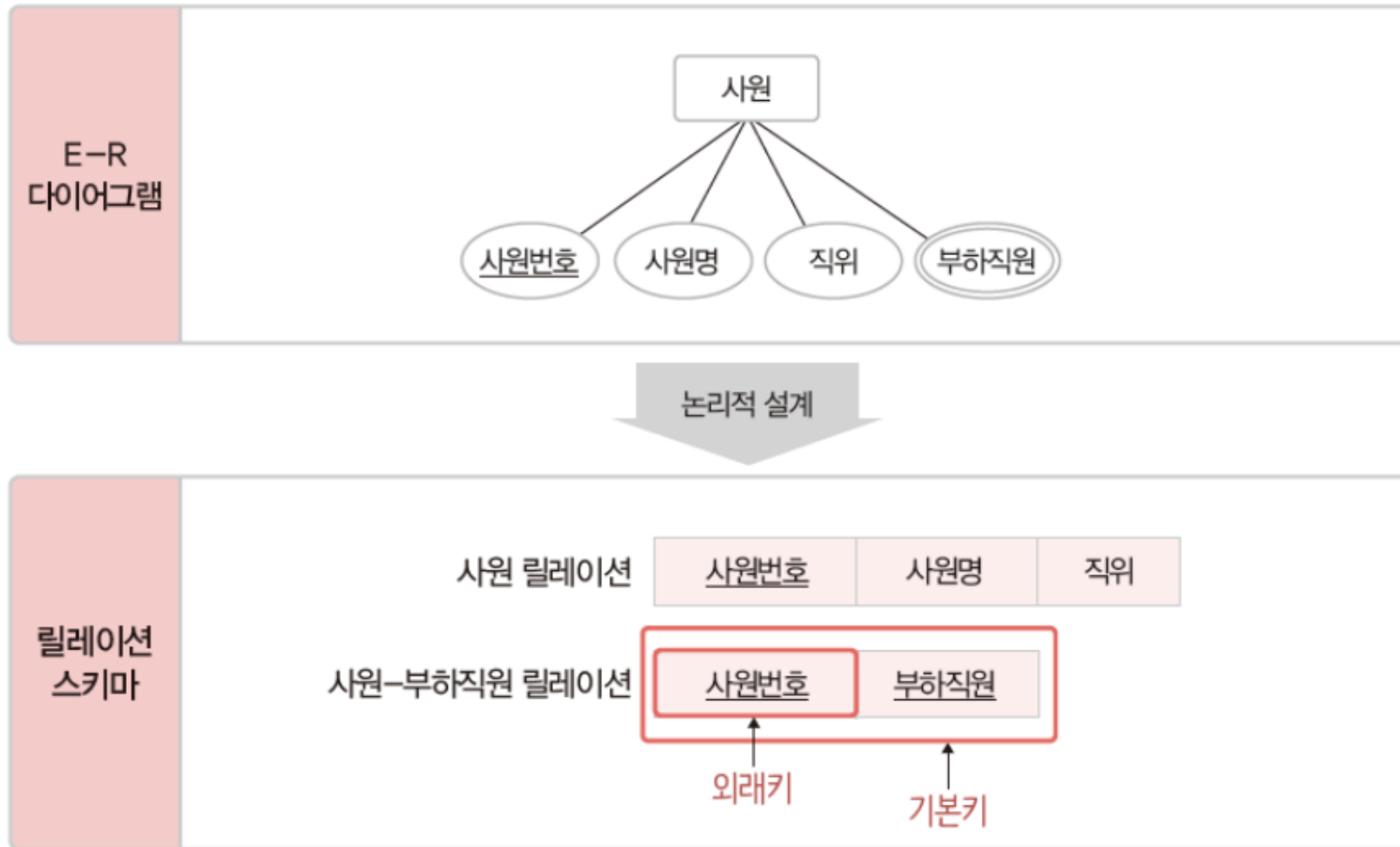


## ◆ ERD의 다중 값 속성은 독립적인 릴레이션으로 변환

- 다중 값 속성과 함께 그 속성을 가지고 있던 개체 릴레이션의 기본키를 외래키로 가져와 새로운 릴레이션에 포함시킴
- 새로운 릴레이션의 기본 키는 다중 값 속성과 외래키를 조합하여 지정

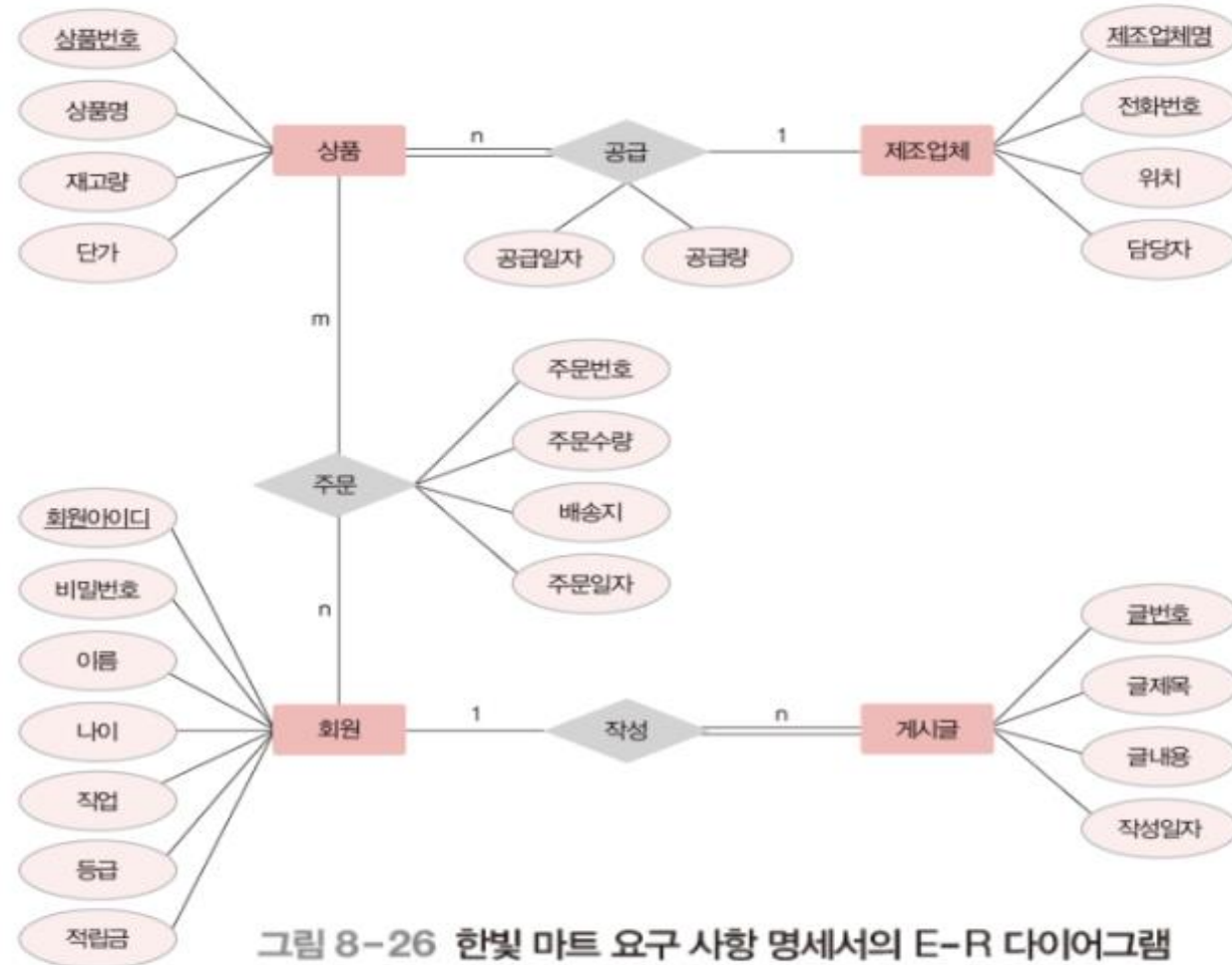
# ☐ 논리적 설계 – 규칙5(다중 값 속성 → 릴레이션)

## ◆ ERD의 다중 값 속성은 독립적인 릴레이션으로 변환



# 논리적 설계 - 최종 변환

## ◆ 개념적 설계의 결과물



# 논리적 설계 – 최종 변환

## ◆ 규칙 1 적용 – 모든 개체는 릴레이션으로 변환

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
----------	------------	-----	-----	------

# 논리적 설계 - 최종 변환

## ◆ 규칙 2 적용 - 다대다 관계는 릴레이션으로 변환



# 논리적 설계 - 최종 변환

## ◆ 규칙 3 적용 - 다대다 관계는 릴레이션으로 변환

- 규칙 4 적용 - 일대일 관계 X → 규칙 4 적용 X
- 규칙 5 적용 - 다중 값 속성 X → 규칙 5 적용 X

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

# 논리적 설계 - 최종 변환

## ◆ 규칙 3 적용 - 다대다 관계는 릴레이션으로 변환

- 규칙 4 적용 - 일대일 관계 X → 규칙 4 적용 X
- 규칙 5 적용 - 다중 값 속성 X → 규칙 5 적용 X

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

## ◆ 테이블 명세서 : 릴레이션 스키마에 대한 설계 정보를 기술한 문서

- 릴레이션 스키마 변환 후 속성의 데이터 타입과 길이, 널 값 허용 여부 등을 세부적으로 결정하고 문서화

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			



## ◆ 물리적 설계

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

## ◆ 구현

- SQL로 작성한 명령문을 DBMS에서 실행하여 데이터베이스를 실제로 생성

```
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'))  
);
```

## ◆ 요구사항에서 개체 및 속성을 도출해보세요

- 500만여명이 넘는 관중을 확보하고 있는 인기 스포츠 프로야구는 여러 팀에 소속된 선수에 대한 상세한 정보를 관리하고 이를 관중에게 다양한 분석 정보로 제공하기 위해 데이터베이스 시스템을 구축하고자 한다.
- 프로리그에는 여러 팀이 포함되며 각 팀은 한 프로리그에만 출전 가능하다. 각 프로리그에 대하여 리그 이름, 스폰서이름, 슬로건을 관리한다. 팀에 대하여 팀 이름, 소속지역소속 지역, 구단주이름, 예산을 관리한다.
- 각 팀에는 최대 50여 명의 선수가 소속되며 한 선수에는 한 팀에만 소속될 수 있다.
- 선수에 대하여 선수 이름, 포지션, 연봉, 입단 연도를 관리한다.
- 선수에 대한 자세한 정보를 제공하기 위해 신체정보를 관리한다.
- 신체정보는 키, 몸무게, 혈액형, 점프 높이, 100m 달리기 기록, 흡연 여부를 관리한다.
- 한 신체정보는 한 선수에만 관리되며, 한 선수는 한 신체정보만을 관리한다.
- 선수의 은퇴 시 선수의 정보뿐만 아니라 신체정보 또한 같이 삭제된다.

## ◆ 도출한 개체 및 속성을 기반으로 데이터베이스 및 테이블을 제작합니다

- 데이터베이스 이름 : sport
- 테이블 이름
  - 프로리그 → league
  - 팀 → team
  - 선수 → player
  - 선수 신체 정보 → health

## ◆ league 테이블

- league\_name
  - primary key
  - varchar(30)
- sponsor
  - varchar(30)
  - not null
- slogan
  - varchar(100)
  - default "NONE"

## ◆ team 테이블

- team\_name 속성
  - varchar(30)
  - primary key
- league\_name 속성
  - varchar(30)
  - not null
- area 속성
  - varchar(30)
  - not null
- owner 속성
  - varchar(30)
  - not null
- budget
  - int
  - default 0

## ◆ player 테이블

- player\_name
  - primary key
  - varchar(30)
- team\_name
  - varchar(30)
  - not null
  - foreign key → team(team\_name)
- position
  - varchar(30)
  - not null
- annual
  - int
  - default 0
- year
  - date
  - not null

## ◆ health 테이블

- player\_name
  - primary key
  - foreign key → player(player\_name)
  - varchar(30)
- height
  - int
  - not null
- weight
  - int
  - not null
- type
  - varchar(1)
  - not null
- jump
  - int
  - default 0
- run
  - int
  - default 0
- smoke
  - varchar(1)
  - default F



## ◆ 요구사항 정리

1. 교수는 아이디, 이름, 나이, 직위, 연구 분야를 가진다
2. 학과에는 학과번호, 학과이름, 학과사무실이 있다
3. 대학원생은 아이디, 이름, 나이, 학위과정을 가진다
4. 과제는 과제번호, 지원기관, 개시일, 종료일, 예산액이 있다
5. 학과마다 그 학과를 운영하는 교수(학과장)가 한 명씩 있다
6. 한 교수가 여러 학과에서 근무할 수 있는데 이때 각 학과별로 참여 백분율이 기록된다
7. 대학원생에게는 학위 과정을 밟을 전공학과가 하나씩 있다
8. 대학원생에게는 어떤 과목을 들으면 좋을지 조언해주는 선임 대학원생이 있다
9. 과제는 교수에 의해 관리된다
10. 과제는 한 사람 이상의 교수에 의해 수행된다
11. 한 과제는 한 명 이상의 대학원생에 의해 수행된다