

데이터베이스 실습 12주차 강의

데이터베이스 설계 : 사용자의 다양한 요구 사항을 고려하여 데이터베이스를 생성하는 과정 (테이블을 1차적으로 생성 뒤 -> DB를 커스텀마이징[정규화 과정])



그림 8-2 데이터베이스 설계 과정의 각 단계별 주요 작업과 결과물

! 기말고사 ! 각 단계별 설명

1단계 : 요구 사항 분석 -> 사용자의 요구 사항을 수집, 분석 -> 요구 사항 명세서

2단계 : 개념적 설계 -> E – R 다이어그램 설계 (E : 명사 [개체 속성] / R : 동사 [관계])

STEP 1) 개체 추출, 각 개체의 주요 속성과 키 속성 선별 (명사를 추출)

STEP 2) 개체 간의 관계 결정 (동사를 추출)

→ 매핑 카디널리티를 찾기[필수적 참여 -> 회원 가입 / 선택적 참여 - 글쓰기]

STEP 3) E-R 다이어그램으로 표현

3단계 : 논리적 설계 -> E - R 다이어그램을 통해 테이블을 생성 (릴레이션 스키마 -> 테이블 구조)

규칙 1 : 모든 개체는 릴레이션으로 변환 -> 테이블

규칙 2 : 다대다(n:m) 관계는 릴레이션으로 변환 -> 테이블 (거의 대부분 테이블로 변환)

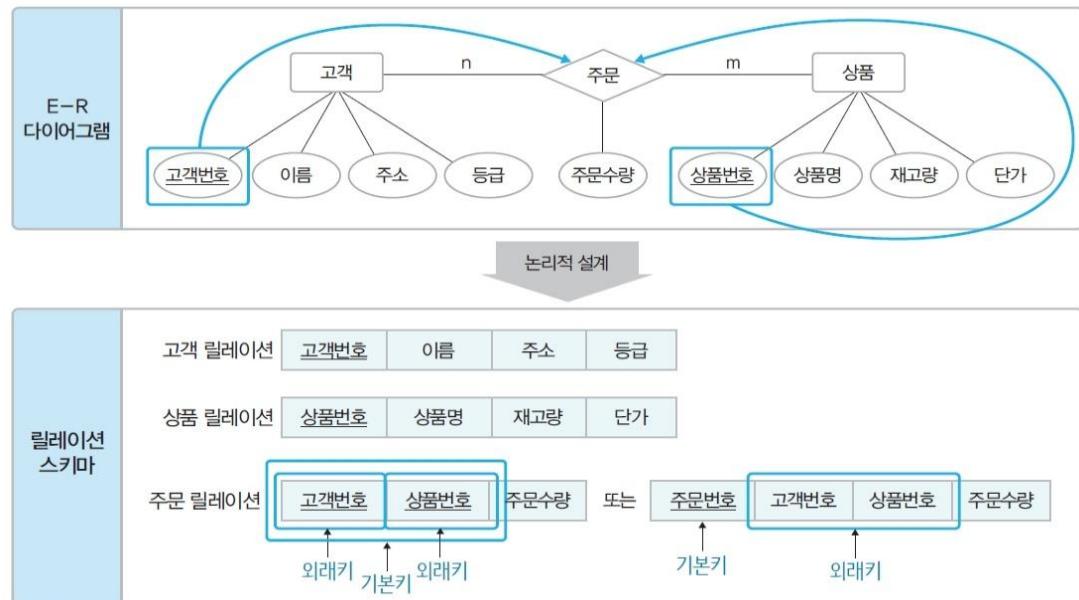


그림 8-30 다대다 관계를 릴레이션으로 변환하는 규칙을 적용한 예

규칙 3 : 일대다(1:n) 관계는 외래키로 표현 -> 필드

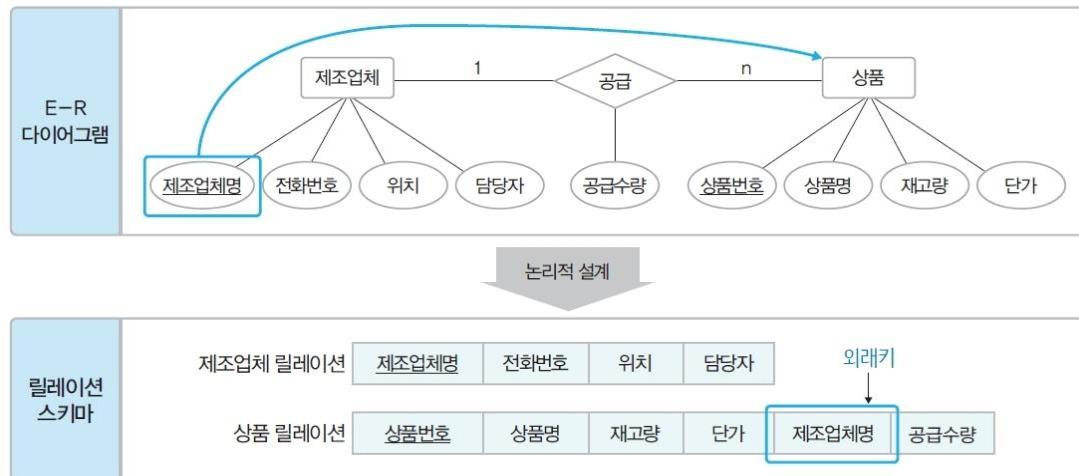


그림 8-31 일반적인 개체가 참여하는 일대다 관계를 외래키로 표현하는 규칙을 적용한 예

규칙 4 : 일대일(1:1) 관계는 외래키로 표현 -> 필드

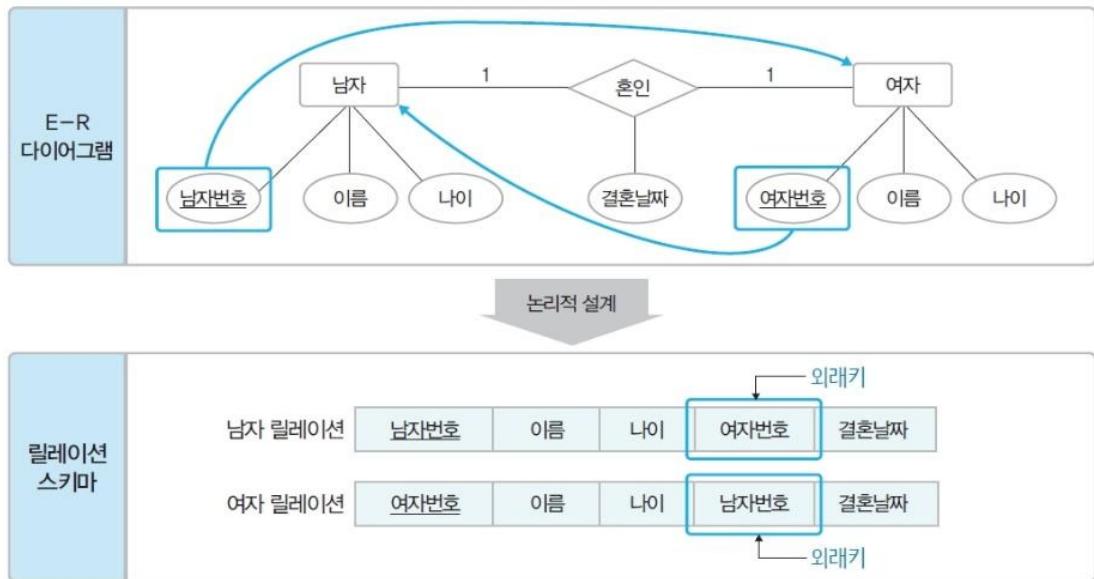


그림 8-33 일반적인 일대일 관계를 외래키로 표현하는 규칙을 적용한 예

규칙 5 : 다중 값 속성은 릴레이션으로 변환 -> 테이블

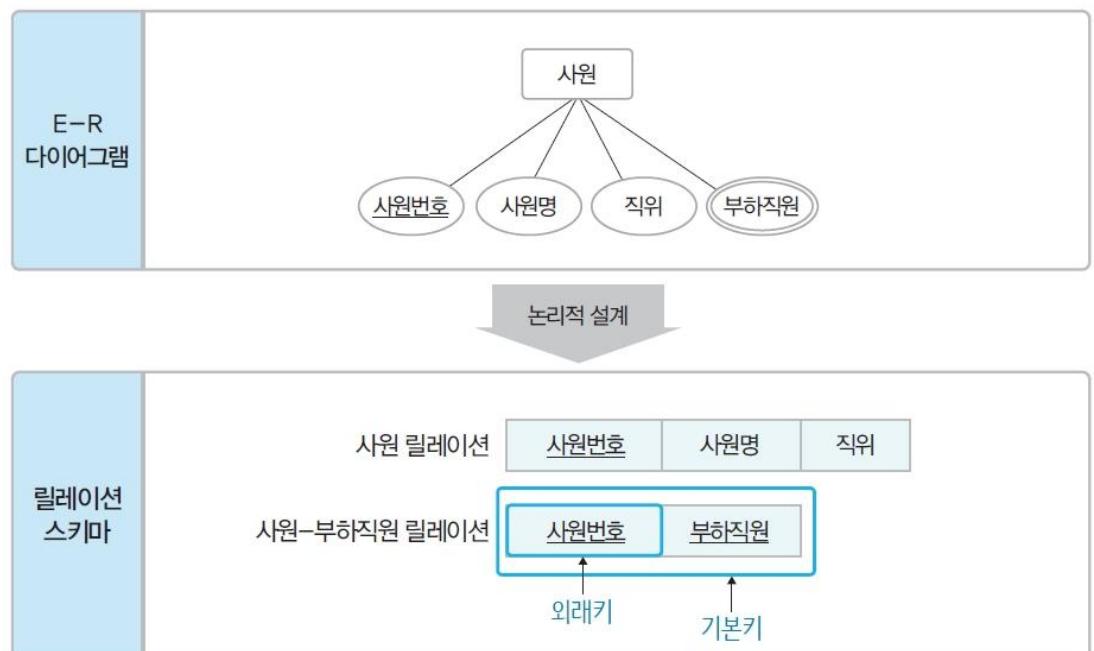


그림 8-36 다중 값 속성을 릴레이션으로 변환하는 규칙을 적용한 예

기타 고려 사항 -> 모든 관계를 독립적인 테이블로하는 것이 좋다. (독립적으로 운영)

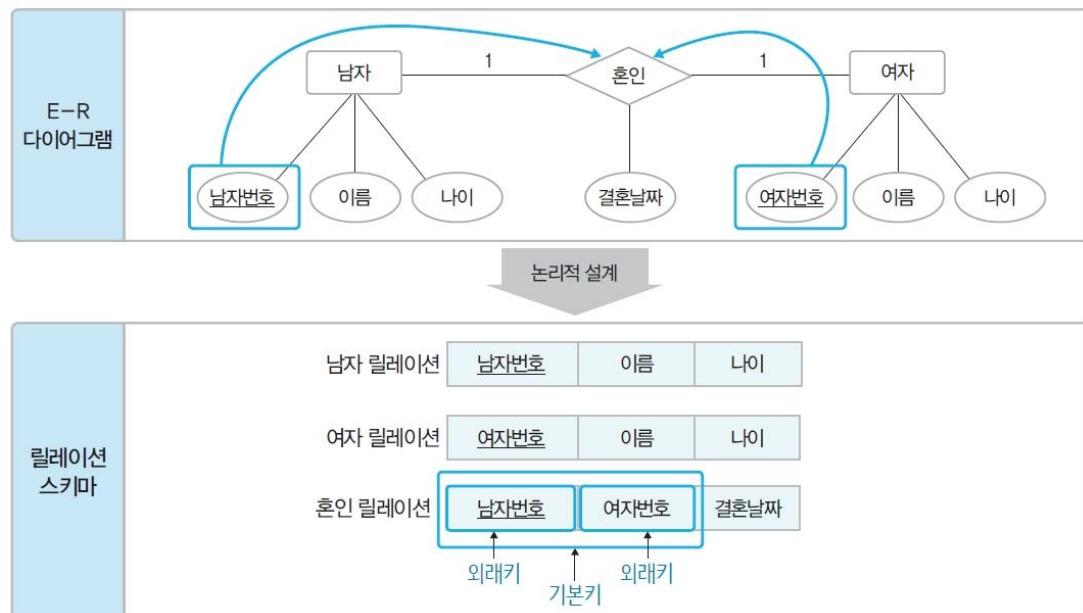


그림 8-40 일대일 관계를 릴레이션으로 변환하는 예

! 릴레이션 스키마 변환 규칙을 이용한 논리적 설계 예 = PPT 참고 (p.58) !

4단계 : 물리적 설계 -> 인덱스 구조나 내부 저장 구조 등에 대한 물리적인 구조를 설계

5단계 : 구현 -> DBMS에서 SQL로 작성한 명령문을 실행하여 데이터베이스를 생성

! 기말고사 ! 요구 사항 명세서를 보고 릴레이션 스키마(구조)를 생성하시오.

- 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.

그림 8-4 한빛 마트의 데이터베이스를 위한 요구 사항 명세서

1번, 2번 -> 고유명사 : 한빛 마트, 대표 명사 : 회원, 명사 : 회원아이디, 비밀번호, 이름, 나이, 직업, 등급, 적립금 / 동사 : 가입

3번 -> PRIMARY KEY : 회원아이디

4번 -> 고유명사 : 상품, 명사 : 상품번호, 상품명, 재고량, 단가 정보 / 동사 : 유지

5번 -> PRIMARY KEY : 상품번호

6번 -> 명사 : 회원, 상품 / 동사 : 주문 (테이블로 구현)

7번 (동사형 테이블) -> 고유명사 : 주문, 명사 : 주문번호, 주문수량, 배송지, 주문일자 / 관계 맺는 테이블(회원, 상품 테이블)의 PRIMARY KEY를 가져와서 외래키로 만들기

8번 -> 명사 : 상품, 제조업체 / 동사 : 공급 (테이블로 구현)

9번 (동사형 테이블) -> 고유명사 : 공급, 명사 : 공급일자, 공급량 / 관계 맺는 테이블의 PRIMARY KEY를 가져와서 외래키로 만들기

10번, 11번 -> 고유명사 : 제조업체, 명사 : 제조업체명, 전화번호, 위치, 담당자 / PRIMARY KEY : 제조업체명

12번 -> 고유명사 : 게시글, 명사 : 글번호, 글제목, 글내용, 작성일자 / 동사 : 작성 (작성이란 관계가 존재, 기능으로 구현)

동사는 테이블 또는 기능으로 구현

정규화 (정규화는 필수)

이상(anomaly) 현상 : 불필요한 데이터 중복으로 인해 릴레이션에 대한 데이터 삽입·수정·삭제 연산을 수행할 때 발생할 수 있는 부작용

삽입 이상 새 데이터를 삽입하기 위해 불필요한 데이터도 함께 삽입해야 하는 문제

갱신 이상 중복 투플 중 일부만 변경하여 데이터가 불일치하게 되는 모순의 문제

삭제 이상 투플을 삭제하면 꼭 필요한 데이터까지 함께 삭제되는 데이터 손실의 문제

3가지의 공통적인 특징 : 불필요한 데이터가 데이터의 필드로 만들어져 있으면 이상 현상이 생긴다.

! 정규화 : 함수적 종속성을 이용해, 릴레이션을 연관성이 있는 속성들로만 구성되도록 분해하여 이상 현상이 발생하지 않는 올바른 릴레이션으로 만들어 나가는 과정 !

정규화 주의 사항 : 정규화를 통해 릴레이션은 무손실 분해되어야 한다.

함수 종속 : X가 Y를 함수적으로 결정한다 (X [결정자] -> Y [종속자]로 표현)

고객아이디	고객이름	등급
apple	정소화	gold
banana	김선우	vip
carrot	고명석	gold
orange	김용욱	silver

고객아이디 → 고객이름
고객아이디 → 등급

또는

고객아이디 → (고객이름, 등급)

완전 함수 종속(FFD) : 릴레이션에서 속성 집합 Y가 속성 집합 X에 함수적으로 종속되어 있지만, 속성 집합 X의 전체가 아닌 일부분에는 종속되지 않음을 의미

부분 함수 종속(PFD) : 릴레이션에서 속성 집합 Y가 속성 집합 X의 전체가 아닌 일부분에도 함수적으로 종속됨을 의미

정규형(NF) : 릴레이션이 정규화된 정도, 각 정규형마다 제약조건이 존재

제1정규형(1NF) : 릴레이션의 모든 속성이 더는 분해되지 않는 원자 값만 가짐

제2정규형(2NF) : 릴레이션이 제1정규형에 속하고, 기본키가 아닌 모든 속성이 기본키에 완전 함수 종속