

DB 모델링

목차

- ✓ Chap01. DB 모델링 개요
- ✓ Chap02. 개념적 모델링
- ✓ Chap03. 논리적 모델링
- ✓ Chap04. 물리적 모델링

Chap01.

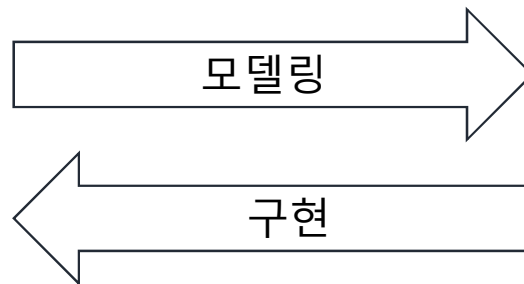
DB 모델링 개요

▶ 모델링

말 그대로 모델을 만드는 작업을 뜻함. 즉, 현실 세계를 단순화 시켜 표현하는 기법



실체
(Reality)



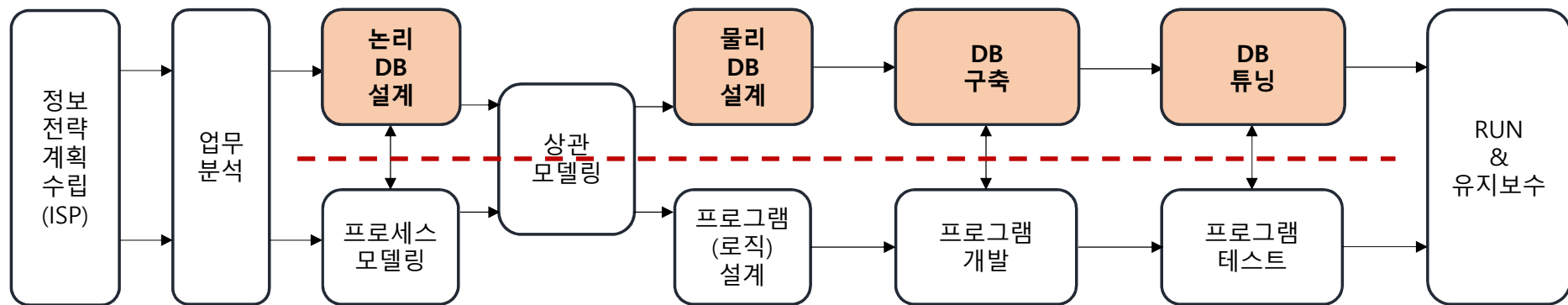
모델
(Model)

▶ 소프트웨어 개발과정

✓ 소프트웨어(프로그램) 개발 프로세스



데이터 관점



프로세스 관점

▶ DB 모델링의 주요 개념

✓ 엔티티(Entity)

업무의 관심 대상이 되는 정보를 갖고 있거나 그에 대한 정보를 관리할 필요가 있는 유형, 무형의 사물(개체) (유형, 무형, 문서, 이력, 코드)

학생

✓ 속성(Attribute)

엔티티에서 관리해야 할 최소 단위 정보 항목(관심이 있는 항목)을 말하며 엔티티는 하나 이상의 속성을 포함 (기본, 유도, 설계)

학생
학번 이름 주소 전공

✓ 인스턴스(Instance)

엔티티의 속성으로 실제로 구현된 하나의 값

학번	이름	주소	전공
201801	김하루	마산	경영
201802	박지윤	울산	환경

▶ DB 모델링의 주요 개념

✓ 엔티티(Entity) 조건

1. 업무의 관심 대상이 되는 사물이어야 된다.
2. 두 개 이상의 인스턴스를 소유해야된다.
3. 마땅한 속성을 소유해야 된다.

✓ 속성(Attribute) 명명규칙

1. 속성의 의미가 분명히 드러나게 작성할 것 (명확)
2. 해당 업무에서 사용하는 이름 부여할 것
3. 서술식(수식어, 소유격) X, 약어 X
4. 엔티티에서 유일하게 식별 가능하도록 지정할 것 (중복 X)

▶ DB 모델링의 주요 개념

✓ 서브 타입

- 인스턴스들의 집합인 엔티티라는 전체 집합에서 일부의 인스턴스들만 모아 놓은 부분집합

✓ 슈퍼 타입

- 서브타입의 전체 집합
- 하나의 엔티티가 서브타입을 정의하면 엔티티가 슈퍼타입이 됨
- 하나의 서브타입에 하위 서브타입을 정의하면 상위 서브타입은 슈퍼타입이 된다.

▶ DB 모델링의 주요 개념

서브타입세트명

년도	학기	과목명	이수구분	학점	점수	등급
2015	1	사고와표현	교양	2	100	A+
2015	1	회로이론1	기본전공	3	84	A+
2015	1	공업수학	기본전공	3	89	B+
2015	2	마이크로프로세서1	기본전공	3	89	B+
2015	2	고체역학	기본전공	3	74	C
2016	1	회로이론2	기본전공	3	96	A+
2016	1	통합영어(듣기)2	교양	1	92	A

슈퍼타입
< 전체성적 엔티티 >

이수구분 기준
부분집합 생성

서브타입세트

서브타입1
< 기본전공 >

서브타입2
< 교양 >

2015	1	회로이론1	기본전공	3	84	A+
2015	1	공업수학	기본전공	3	89	B+
2015	2	마이크로프로세서1	기본전공	3	89	B+
2015	2	고체역학	기본전공	3	74	C
2016	1	회로이론2	기본전공	3	96	A+
2015	1	사고와표현	교양	2	100	A+
2016	1	통합영어(듣기)2	교양	1	92	A

▶ DB 모델링의 주요 개념

✓ 관계(Relationship)

두 엔티티 사이의 관련성을 나타냄

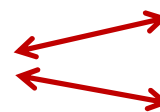
학생	수강한다	수강과목
학번 이름	수강된다	학번 과목

✓ 카디널리티(Cardinality)

- 각 엔티티에 속해 있는 인스턴스들 간에 수적으로 어떤 관계에 있는지를 나타냄
- 종류로는 1:1, 1:N, M:N의 관계가 있다.

학번	이름	주소	전공
201801	김하루	마산	경영
201802	박지윤	울산	환경
201803	권하인	광주	영문
201804	최봉구	서울	불어

학생



학번	이름
201802	바둑
201803	수영
201802	여행
201804	게임

학생의 취미

< 1 : N 관계 >

▶ DB 모델링의 주요 개념

✓ 주식별자(Primary Identifier)

엔티티 내 각 인스턴스를 구별하는 기준이 되는 속성

사원정보
◆ 사번
이름 주민등록번호 생년월일 나이

부서정보
부서코드 (PK)
부서명 연간 매출액 매출순위 총인원

✓ 외래식별자(Foreign Identifier)

관계가 있는 엔티티 간의 연결고리 역할을 하는 속성

사원정보
◆ 사번
이름 주민등록번호 생년월일 나이
부서코드(FK)

부서정보
◆ 부서코드
부서명 연간 매출액 매출순위 총인원

▶ DB 모델링의 주요 개념

✓ 개념적 설계

요구분석 단계에서 정의된 핵심 개체와 그들 간의 관계를 바탕으로 ERD를 생성하는 단계

✓ 논리적 설계

개념 설계에서 추상화된 데이터를 구체화하여 개체, 속성을 테이블화하고 상세화 하는 과정

✓ 물리적 설계

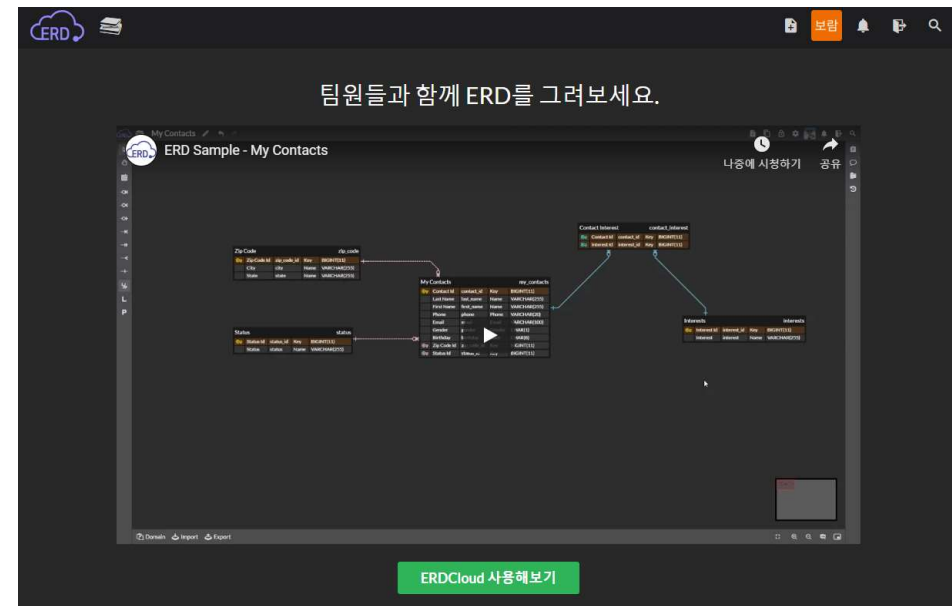
논리적 설계 단계에서 표현된 데이터(ERD)를 실제 컴퓨터의 저장장치에 어떻게 표현할 것인가
(관계형 데이터베이스로 전환)

▶ DB 모델링 툴 활용

✓ www.erdcloud.com

1. <https://www.erdcloud.com/> 접속
2. 회원가입 후 로그인 (구글)
3. 나의 ERD 에서 + 버튼 클릭 후 ERD 작업

* 팀 공유 기능이 있어 프로젝트 진행 시 유용하다.



Chap02.

개념적 모델링

Part 1.

Entity 도출하기

▶ 업무 분석 자료 예

[illegible]

< 거래 명세서 양식 >

서문 도서관

회원 가입 신청서

이름 : _____
집전화 : _____
집주소 : _____
이메일 : _____
가입일자 : _____

회원 번호 부여 :

서문 도서관

도서대출 전표

대출일 : _____
회원번호 : _____ 이름 _____

no	도서 관리번호	제목	반납예정일

대출번호 :

< 가입 신청서 및 전표 >

■ 수행 업무	
작무내용	작 무 질 차 및 방 법
교육프로그램 기획	<ul style="list-style-type: none"> ▶ 설문조사 및 직접 Interview를 통한 사람들의 교육 Needs를 정확히 조사한다 ▶ 교육 Needs를 최대한 충족할수 있는 교육 프로그램 기획을 위하여 관련 자료를 수집하여 조사하고 분석한다, ▶ 사내,외 교육관련 담당자 및 전문가와의 직접 Interview를 통하여 최근 교육 프로그램 Concept를 설정한다, ▶ 효율적인 교육성과를 위한 적절한 교재 및 강사를 선정하여 Contact한다 ▶ 교육프로그램의 실현의 과정을 최종 확정하여 해당 결정권자의 결재를 득한후 시행한다
교육프로그램 진행	<ul style="list-style-type: none"> ▶ 교육대상자 및 강사에게 일시, 장소, 교육내용등의 자세한 안내문을 작성하여 발송한다 ▶ 효과적인 교육을 위한 신뢰성있는 강사소개 및 탐색있는 교육진행률 한다
교육결과 평가	<ul style="list-style-type: none"> ▶ 교육평가의 기초자료로 활용하기 위하여 교육내용 및 강의진행관련 적절한 설문서를 작성한다 ▶ 설문대상자별로 설문서를 배포, 회합하여 교육결과 기초자료로 활용을 위해 분석한다 ▶ 관련자료의 최종 평가 및 진행자의 교육내용 및 효과에 대한 분석내용 및 개선안을 마련하여 보고한다

< 업무 기술서 >

▶ Entity 정의

✓ 엔티티 도출

업무 분석 단계 이후, 분석 자료(업무 기술서, 인터뷰 자료, 장부와 전표 등)로부터 엔티티 도출

✓ 엔티티 도출 과정

정해진 공식은 없으나 경험이 없으면 다음의 과정들을 거쳐서 엔티티 기술서 작성

1. 엔티티 후보 풀과 엔티티 리스트를 그린다.
2. 분석 대상 문서를 보고 명사를 찾아 표시한다.
3. 명사 하나하나에 대해 속성인지 엔티티인지 구분한다.
4. 중복된 명사나 유사한 의미의 명사는 하나로 정리한다.
5. 엔티티 후보 풀에 있는 명사들을 검토한다.
6. 도출된 엔티티에 대하여 구축될 시스템에서 데이터를 관리할 필요가 있는지 판단한다.

▶ Entity 정의 - 업무 기술서

✓ 업무 기술서에서 엔티티 도출하기

- ① 도서를 대출하기 위해서는 회원가입을 하여야 한다.
- ② 회원가입 시에는 이용자이름, 전화번호, 집주소, 이메일 주소를 입력한다.
- ③ 회원가입을 하면 회원증이 발급되고 회원카드에 부착된 바코드를 가지고 대출 시 신분 확인을 할 수 있다.
- ④ 회원에게는 자체 관리를 위해 회원번호가 부여된다.
- ⑤ 회원의 대출 가능 도서 수는 5권으로 제한한다.

< 도서대출 업무기술서(회원등록) >

엔티티 후보 풀

--

엔티티명	포함 속성	유사어

▶ Entity 정의 - 업무 기술서

✓ 업무 기술서에서 엔티티 도출하기

- ① **도서**를 **대출**하기 위해서는 **회원가입**을 하여야 한다.
- ② 회원가입 시에는 **이용자이름**, **전화번호**, **집주소**, **이메일 주소**를 입력한다.
- ③ 회원가입을 하면 **회원증**이 발급되고 **회원카드**에 부착된 **바코드**를 가지고 대출 시 **신분** 확인을 할 수 있다.
- ④ **회원**에게는 자체 관리를 위해 **회원번호**가 부여된다.
- ⑤ 회원의 **대출 가능 도서 수**는 5권으로 제한한다.

< 도서대출 업무기술서(회원등록) >

엔티티 후보 풀



엔티티명	포함 속성	유사어
도서		
대출		
회원	이름, 전화번호, 집 주소, 이메일, 회원번호, 대출 가능 도서 수	
회원증		회원카드

▶ Entity 정의 - 업무 기술서

✓ 업무 기술서에서 엔티티 도출하기

엔티티명	엔티티 설명	관련 속성	유사어
도서	대출하는 도서에 대한 상세 정보		
회원	도서관 회원에 대한 상세 정보	이름, 전화번호, 집 주소, 이메일 주소, <u>회원번호</u> , 대출 가능 도서 수	
대출	회원이 도서를 대출한 내역 정보		

< 엔티티 기술서 >

▶ Entity 정의 - 전표

✓ 전표에서 엔티티 도출하기

서문 도서관

도서대출 전표

대출일 : _____

회원번호 : _____ 이름 _____

no	도서 관리번호	제목	반납예정일

대출번호 :

< 도서대출 전표 >

엔티티 후보 풀

엔티티명	포함 속성	유사어

< 엔티티 후보 풀과 엔티티 리스트 >

▶ Entity 정의 - 전표

✓ 전표에서 엔티티 도출하기

서문 도서관

도서대출 전표

대출일 : _____

회원번호 : _____ 이름 : _____

no	도서 관리번호	제목	반납예정일

대출번호 : _____

< 도서대출 전표 >

엔티티 후보 풀

반납예정일

도서, 대출엔티티 중
어디 있어야 될까?

엔티티명	포함 속성	유사어
대출	대출일, 회원번호, 이름, 대출번호, 반납예정일	
도서	도서관리번호, 제목	

< 엔티티 후보 풀과 엔티티 리스트 >

▶ Entity 정의 - 전표

✓ 전표에서 엔티티 도출하기

엔티티명	엔티티 설명	관련 속성	유사어
대출	회원이 도서를 대출한 내역 정보	대출일, 회원번호, 이름, <u>대출번호</u> , 반납예정일	
도서	대출하는 도서에 대한 상세 정보	<u>도서관리번호</u> , 제목	

< 엔티티 기술서 >

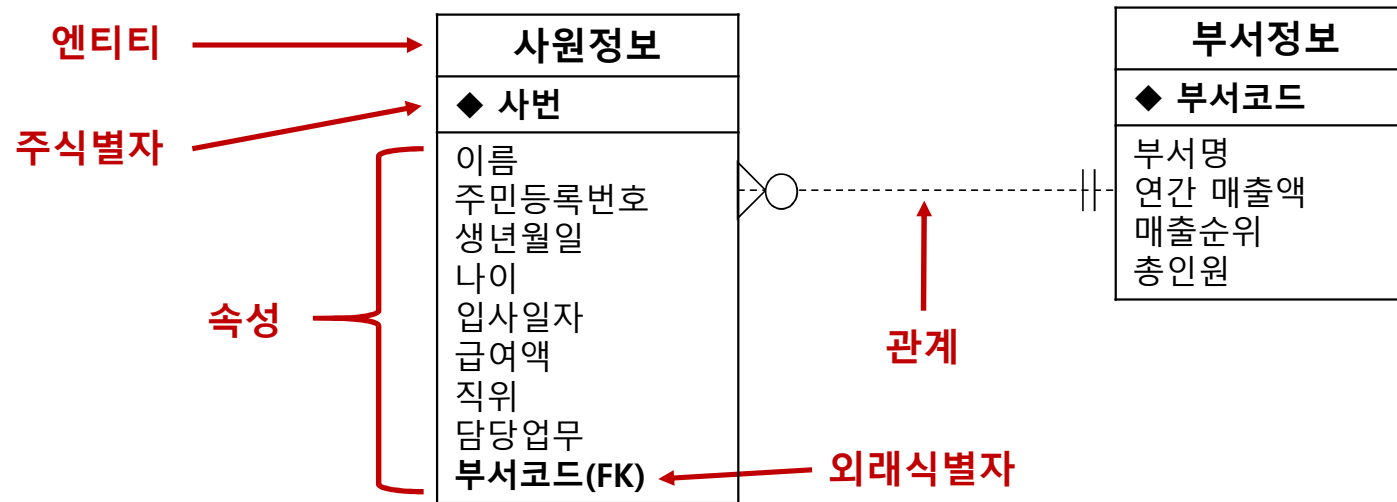


Part 2.

ER-Diagram

▶ ERD(Entity-Relationship Diagram)

'개체 관계도'라고도 불리며 요구분석사항에서 얻은 엔티티와 속성들을 그림으로 그려 관계를 도출한 것



▶ ERD 표기법 (식별자)

✓ 주식별자(Primary Identifier)

- 엔티티에 소속된 인스턴스들을 구별하는 기준 역할을 하는 속성
- 주식별자는 **유일성, 최소성, 불변성, 존재성**의 특징을 가짐
- 주식별자는 하나가 아닌 여러 속성일 수 있음 (복합 키)
- 엔티티의 속성 중 주식별자 속성이 없다면 새로운 속성을 만들어 줌 (인위적 주식별자)

✓ 외래 식별자(Foreign Identifier)

- 관계가 있는 두 엔티티를 부모, 자식 엔티티로 구분한 후 부모의 주식별자와 공통 속성이 자식에게도 존재하면 해당 속성을 외래 식별자로 지정
- 자식 엔티티에 부모 엔티티 주식별자 공통 속성이 없을 경우 자식에게 속성을 추가한 후 외래 식별자로 지정

▶ ERD 표기법 (관계)

✓ 엔티티 간의 부모-자식 관계

- 상호 관계가 있는 두 엔티티 중에서 어느 쪽의 정보가 먼저 생성이 되는가에 따라 결정
- 부모 엔티티의 정보가 있어야지만 존재할 수 있는 것이 자식 엔티티

✓ 참여도

- 참여도에는 **필수(mandatory)**, **선택(optional)** 두 가지로 존재
- 어떤 기준이 되는 엔티티가 있을 때 반드시 대응되는 엔티티가 존재해야 한다면 필수, 존재 할 수도, 하지 않을 수도 있다면 선택

✓ 카디널리티

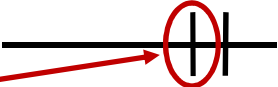

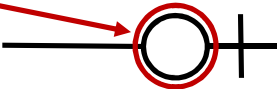
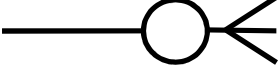
- 두 개의 엔티티 간 관계에서 엔티티에 속해 있는 인스턴스들을 수적으로 표현한 것
- 인스턴스가 1개와 대응된다면 '1' 로 표시 다수와 대응된다면 'N' 로 표시

▶ ERD 표기법 (관계)

✓ 카디널리티와 참여도에 따른 관계의 종류

관계의
참여도

{ 필수
선택

Symbol	Meaning
	One-필수
	Many-필수
	One-선택
	Many-선택

▶ ERD 표기법 (식별-비식별 관계)

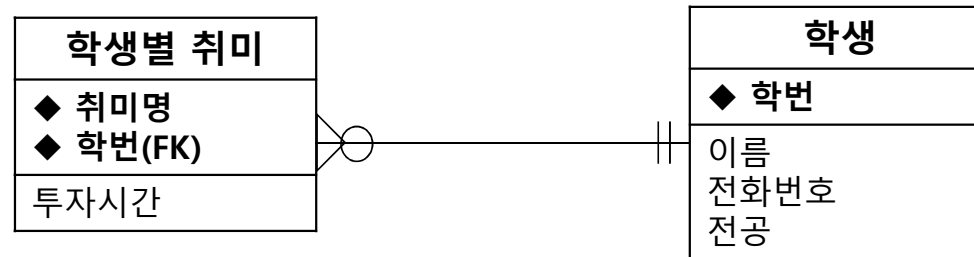
✓ 식별 관계(Identifying Relationship)

- 1:N 관계에서 외래 식별자가 자식 엔티티의 주식별자의 일부가 되는 관계
- PFK로 표시 (**외래 식별자가 주식별자의 역할도 함**)
- 실선으로 관계 표시

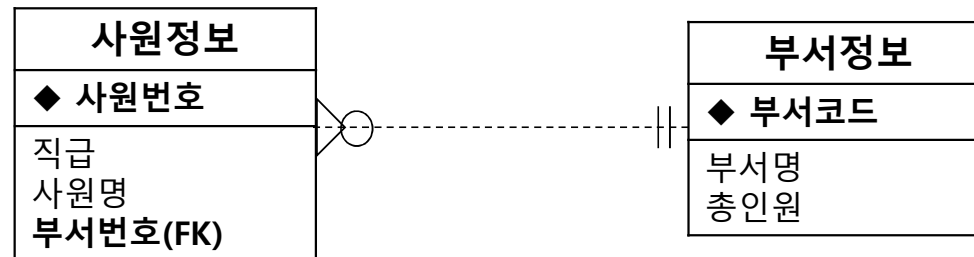
✓ 비식별 관계(Non-Identifying Relationship)

- 1:N 관계에서 외래 식별자가 자식 엔티티의 주식별자 역할을 하지 못하고 단순히 새로운 속성으로 추가되는 관계
- FK로 표시 (**단지 외래식별자의 역할만 함**)
- 점선으로 관계 표시

▶ ERD 표기법 (식별-비식별 관계)



< 식별관계 >



< 비식별 관계 >

▶ ERD 표기법 (관계)

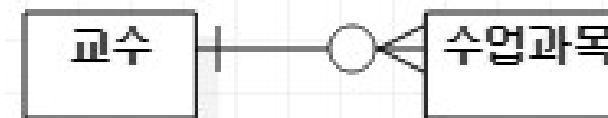
✓ 1 : 1 관계

X에 속하는 한 개체는 Y에 속하는 한 개체에만 연결되며,
Y에 속하는 한 개체도 X에 속하는 한 개체에만 연결될 때



✓ 1 : N 관계

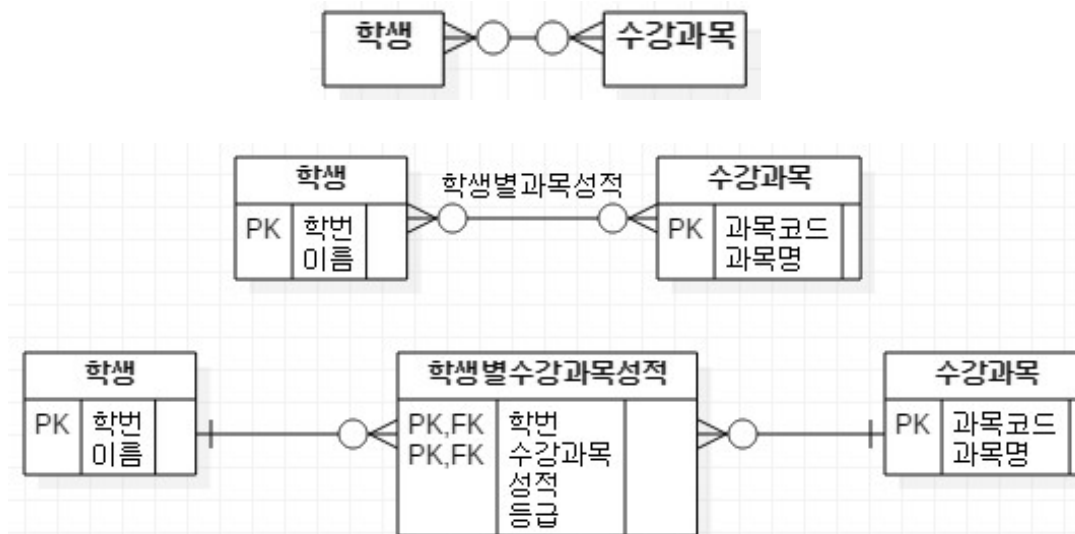
X에 속하는 한 개체는 Y에 속하는 여러 개체에만 연결되며,
Y에 속하는 한 개체는 X에 속하는 한 개체와 연결될 때



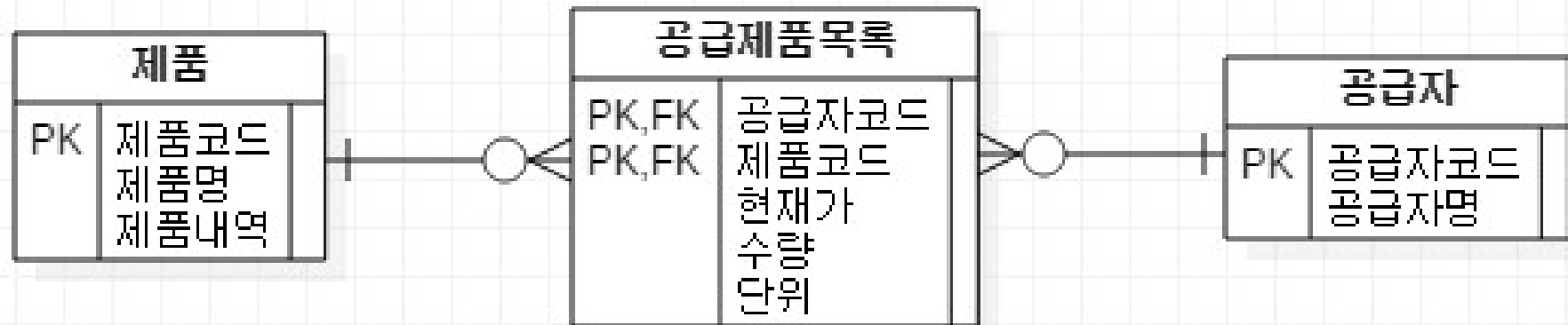
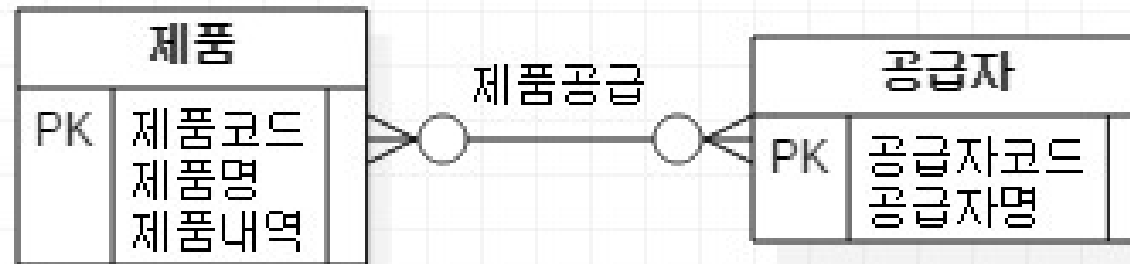
▶ ERD 표기법 (관계)

✓ M : N 관계

- X에 속하는 한 개체는 Y에 속하는 여러 개체와 연결될 수 있으며, Y에 속하는 한 개체도 X에 속하는 여러 개체와 연결될 수 있을 때
- M:N 관계는 덜 완성된 모습으로 데이터 구조에 있어서 어떠한 실제적 방법으로도 구현이 불가능하다. (따라서 M:N관계는 해소해 주어야 한다.)

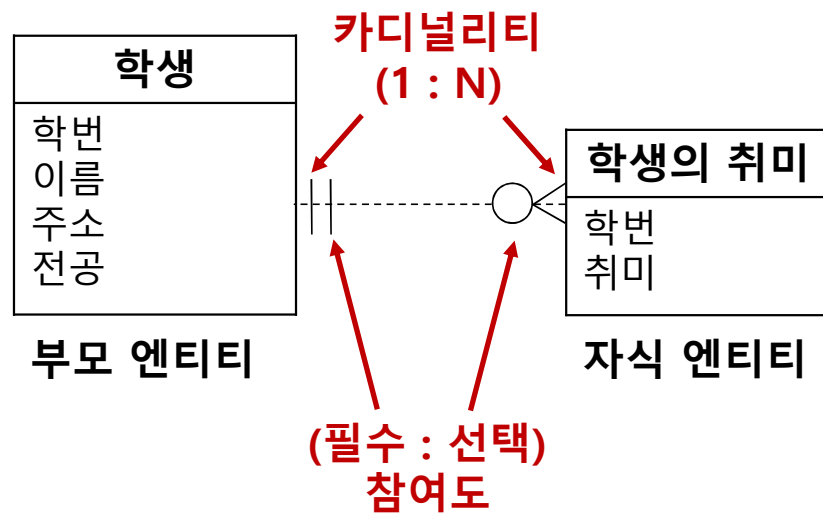


< M:N 관계 해소 >

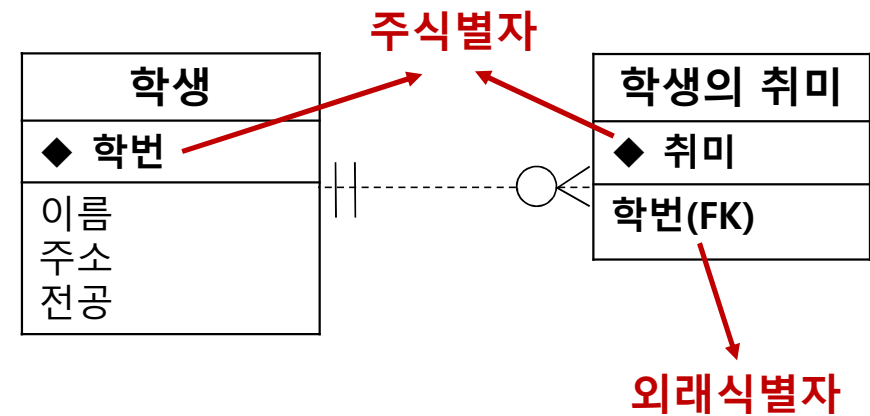


▶ ERD 표기법 (관계 + 식별자)

< 일반적인 카디널리티와 참여도 >



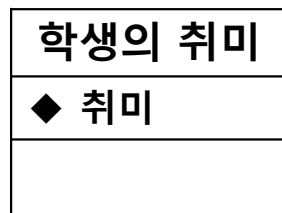
식별자 표현



▶ ERD 표기법 (관계 + 식별자)

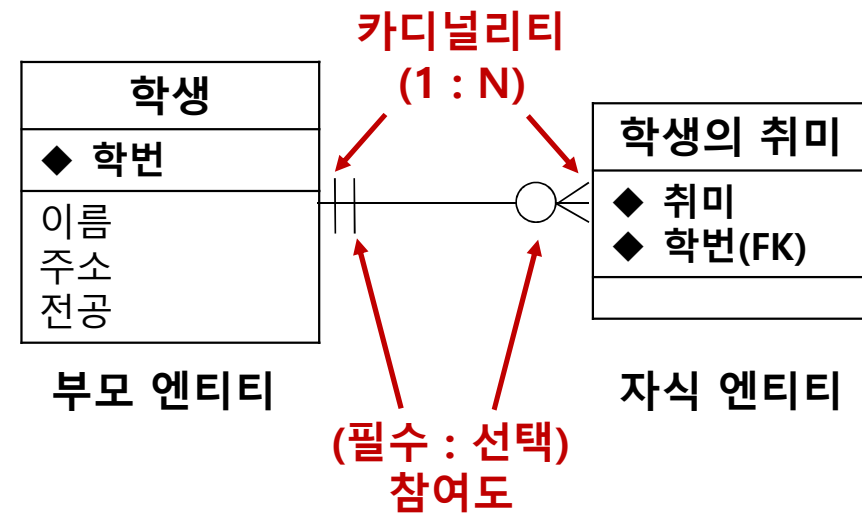


부모 엔티티



자식 엔티티

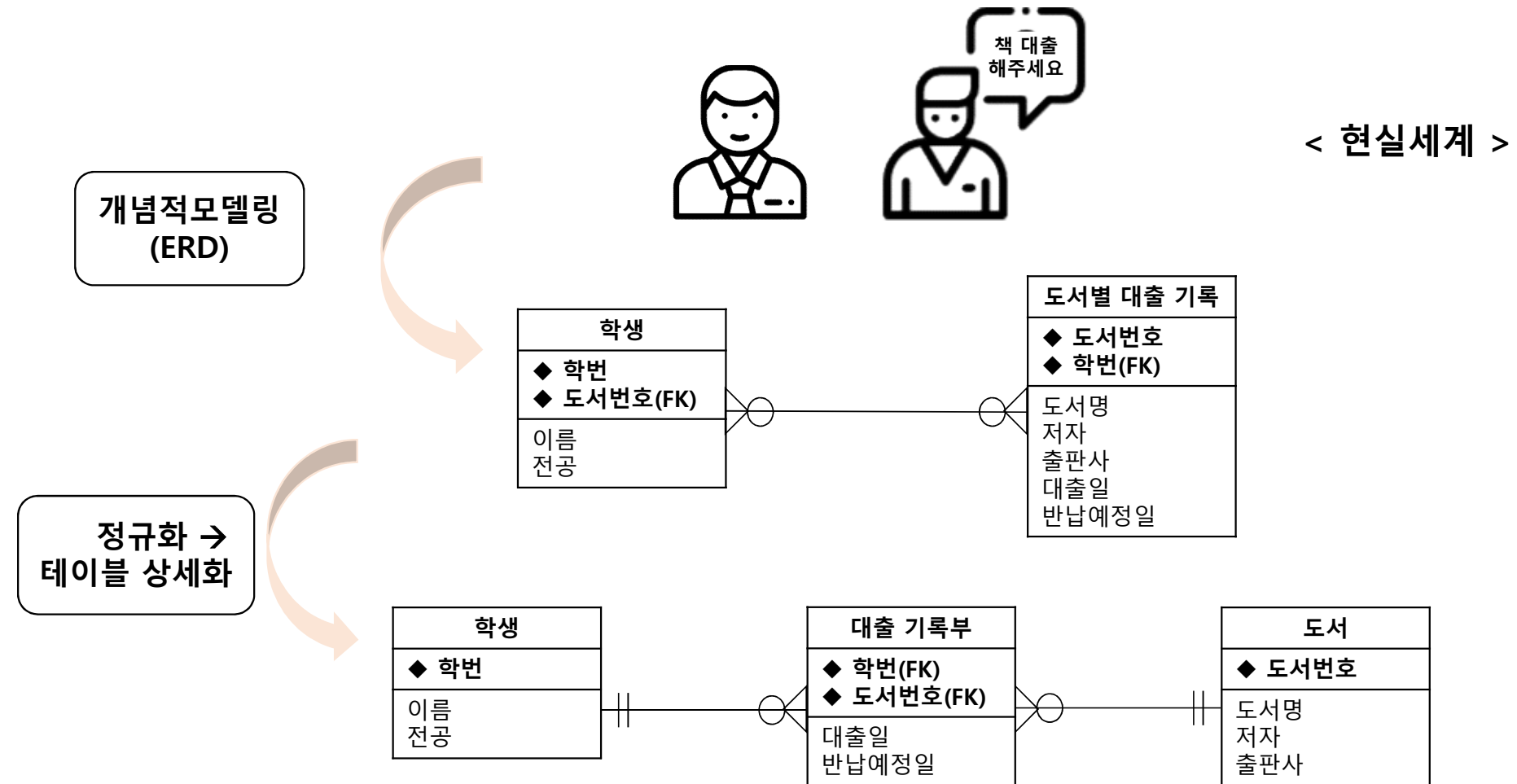
관계 표현



Chap03.

논리적 모델링

▶ 논리적 모델링 단계



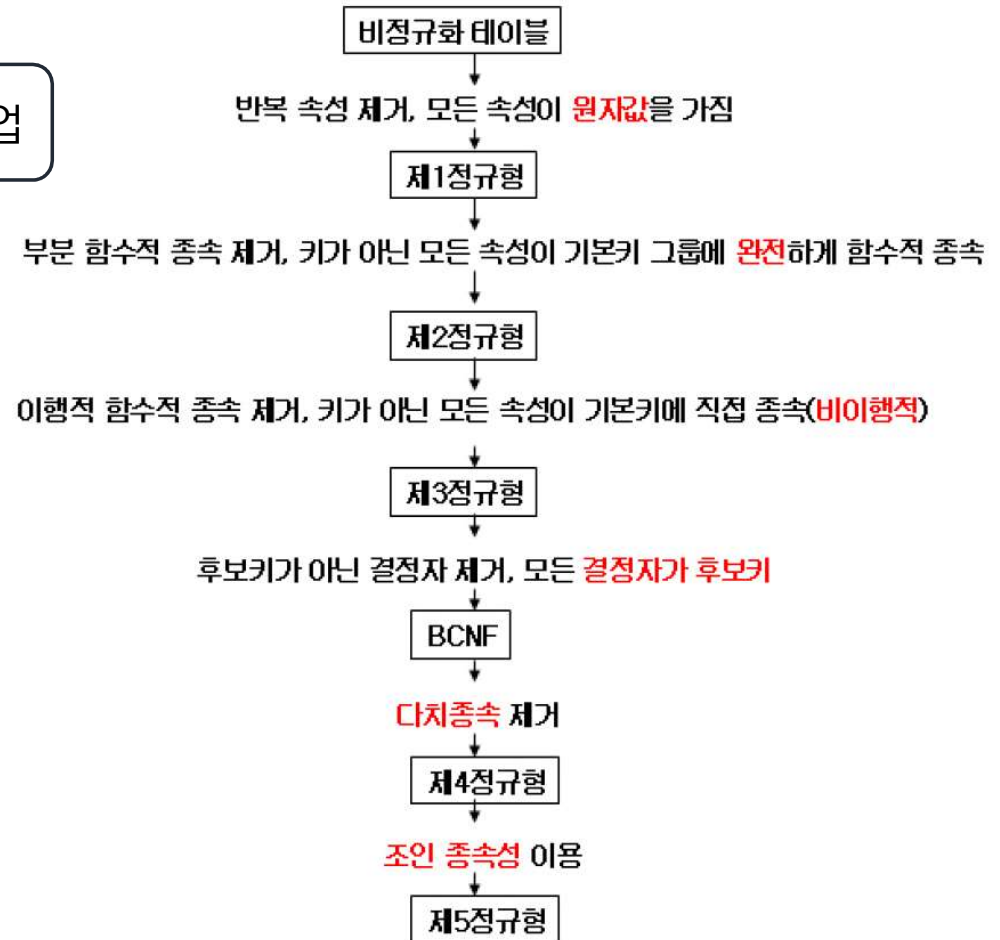
▶ 정규화 (DB normalization)

✓ 정규화

관계형 데이터베이스에서 데이터를 구조화 하는 작업

✓ 정규화의 목적

- 데이터의 중복을 방지하고 보다 효율적으로 데이터를 저장하기 위함
- 삽입, 삭제, 갱신 이상의 발생 가능성을 줄이기 위함



▶ 이상 (Anomaly)

✓ 삽입 이상

불필요한 정보를 함께 저장하지 않고는 어떤 정보를 저장하는 것이 불가능

✓ 갱신 이상

반복된 데이터 중에 일부만 수정하면 데이터의 불일치가 발생

✓ 삭제 이상

유용한 정보를 함께 삭제하지 않고는 어떤 정보를 삭제하는 것이 불가능

▶ 이상 (Anomaly)

✓ 삭제 이상

주문번호	제품번호	제품명	단가	주문수량
452790	1111	팔걸이 의자	20,000	1
452791	1111	팔걸이 의자	20,000	3
452792	1111	팔걸이 의자	20,000	2
452793	2222	침대	600,000	1
452794	3333	옷장	1,220,000	1

ex) 주문에 대한 정보(주문번호, 주문수량)를 삭제할 때 제품번호, 제품명, 단가도 같이 삭제됨

▶ 이상 (Anomaly)

✓ 삽입 이상

주문번호	제품번호	제품명	단가	주문수량
452790	1111	팔걸이 의자	20,000	1
452791	1111	팔걸이 의자	20,000	3
452792	1111	팔걸이 의자	20,000	2
452793	1111	팔걸이 의자	20,000	10
452794	1111	팔걸이 의자	20,000	5
NULL	3333	침대	1,220,000	NULL

ex) 주문에 대한 정보만 입력해도 제품에 대한 정보도 같이 입력해야 함
제품에 대한 정보만 입력하고 싶을때 주문에 대한 정보는 NULL값을 가짐

▶ 이상 (Anomaly)

✓ 갱신 이상

주문번호	제품번호	제품명	단가	주문수량
452790	1111	팔걸이 의자	50,000	1
452791	1111	팔걸이 의자	50,000	3
452792	1111	팔걸이 의자	50,000	2
452793	1111	팔걸이 의자	50,000	10
452794	1111	팔걸이 의자	50,000	5

ex) 동일한 제품명을 중복되어 있는 수만큼 수정 해야 함, 한번만 수정하거나 일부만 수정할 경우 일관성이 맞지 않는 문제 발생

▶ 정규화 과정

✓ 제 1 정규화

엔티티에서 하나의 속성이 복수의 값을 갖도록 설계되었을 때
하나의 속성이 단일 값을 갖도록 하는 것

사원번호(PK)	취미
1001	등산 낚시
1002	테니스 등산
1003	볼링

사원번호(PK)	취미1	취미2	취미3
1001	등산	낚시	
1002	테니스	등산	
1003	볼링		



사원별취미코드(PK)	사원번호	취미
h1	1001	등산
h2	1001	낚시
h3	1002	테니스
h4	1002	등산
h5	1003	볼링

<그림 1>

▶ 정규화 과정

✓ 제 1 정규화

엔티티에서 하나의 속성이 복수의 값을 갖도록 설계되었을 때
하나의 속성이 단일 값을 갖도록 하는 것

사원의 취미		
사번(PK)		
사원명 취미		
사번(PK)	사원명	취미
1001	김개똥	등산
1001	김개똥	낚시
1001	김개똥	테니스

< 그림 1 >

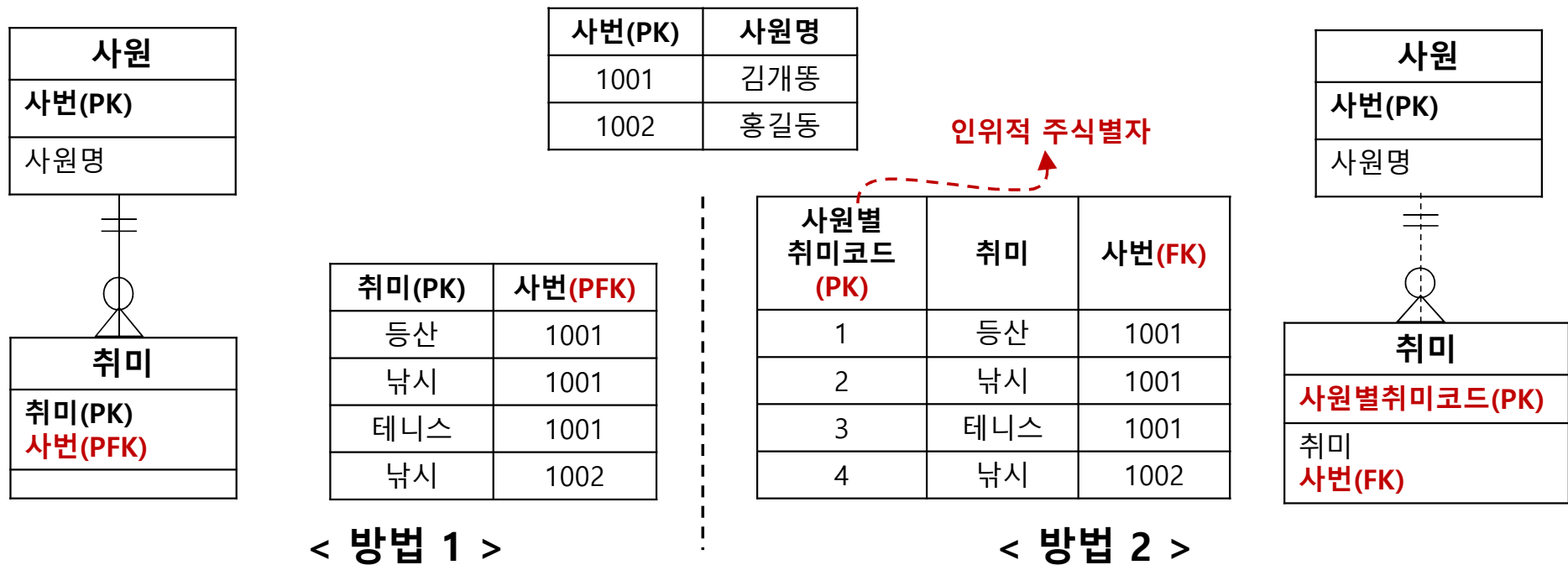
사원의 취미				
사번(PK)				
사원명 취미1 취미2 취미3				
사번(PK)	사원명	취미1	취미2	취미3
1001	김개똥	등산	낚시	테니스
1002	홍길동	낚시	null	null

< 그림 2 >

▶ 정규화 과정

✓ 제 1 정규화

엔티티에서 하나의 속성이 복수의 값을 갖도록 설계되었을 때
하나의 속성이 단일 값을 갖도록 하는 것

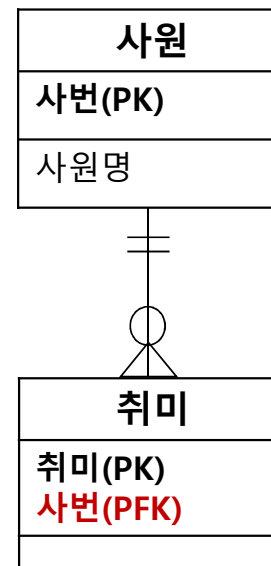
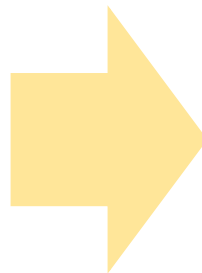


▶ 정규화 과정

✓ 제 1 정규화

엔티티에서 하나의 속성이 복수의 값을 갖도록 설계되었을 때
하나의 속성이 단일 값을 갖도록 하는 것

사원의 취미		
사번(PK)		
사원명 취미		
사번(PK)	사원명	취미
1001	김개똥	등산
1001	김개똥	낚시
1001	김개똥	테니스



사번(PK)	사원명
1001	김개똥
1002	홍길동

취미(PK)	사번(PFK)
등산	1001
낚시	1001
테니스	1001
낚시	1002

▶ 정규화 과정

✓ 제 1 정규화

엔티티에서 하나의 속성이 복수의 값을 갖도록 설계되었을 때
하나의 속성이 단일 값을 갖도록 하는 것

사원의 취미				
사번(PK)				
사원명				
취미1				
취미2				
취미3				

사번(PK)	사원명	취미1	취미2	취미3
1001	김개똥	등산	낚시	테니스
1002	홍길동	낚시	null	null



사원	
사번(PK)	
사원명	

취미	
사원별취미코드(PK)	
취미	
사번(FK)	

사번(PK)	사원명
1001	김개똥
1002	홍길동

인위적 주식별자

사원별 취미코드 (PK)	취미	사번(FK)
1	등산	1001
2	낚시	1001
3	테니스	1001
4	낚시	1002

▶ 정규화 과정

✓ 제 1 정규화

엔티티에서 하나의 속성이 복수의 값을 갖도록 설계되었을 때
하나의 속성이 단일 값을 갖도록 하는 것

사번(PK)	사원명	취미
1001	김개똥	등산
1001	김개똥	낚시
1001	김개똥	테니스

사번(PK)	사원명	취미1	취미2	취미3
1001	김개똥	등산	낚시	테니스
1002	홍길동	낚시	null	null

< 그림 1 >

사번(PK)	사원명
1001	김개똥
1002	홍길동

취미(PK)	사번(PFK)
등산	1001
낚시	1001
테니스	1001
낚시	1002

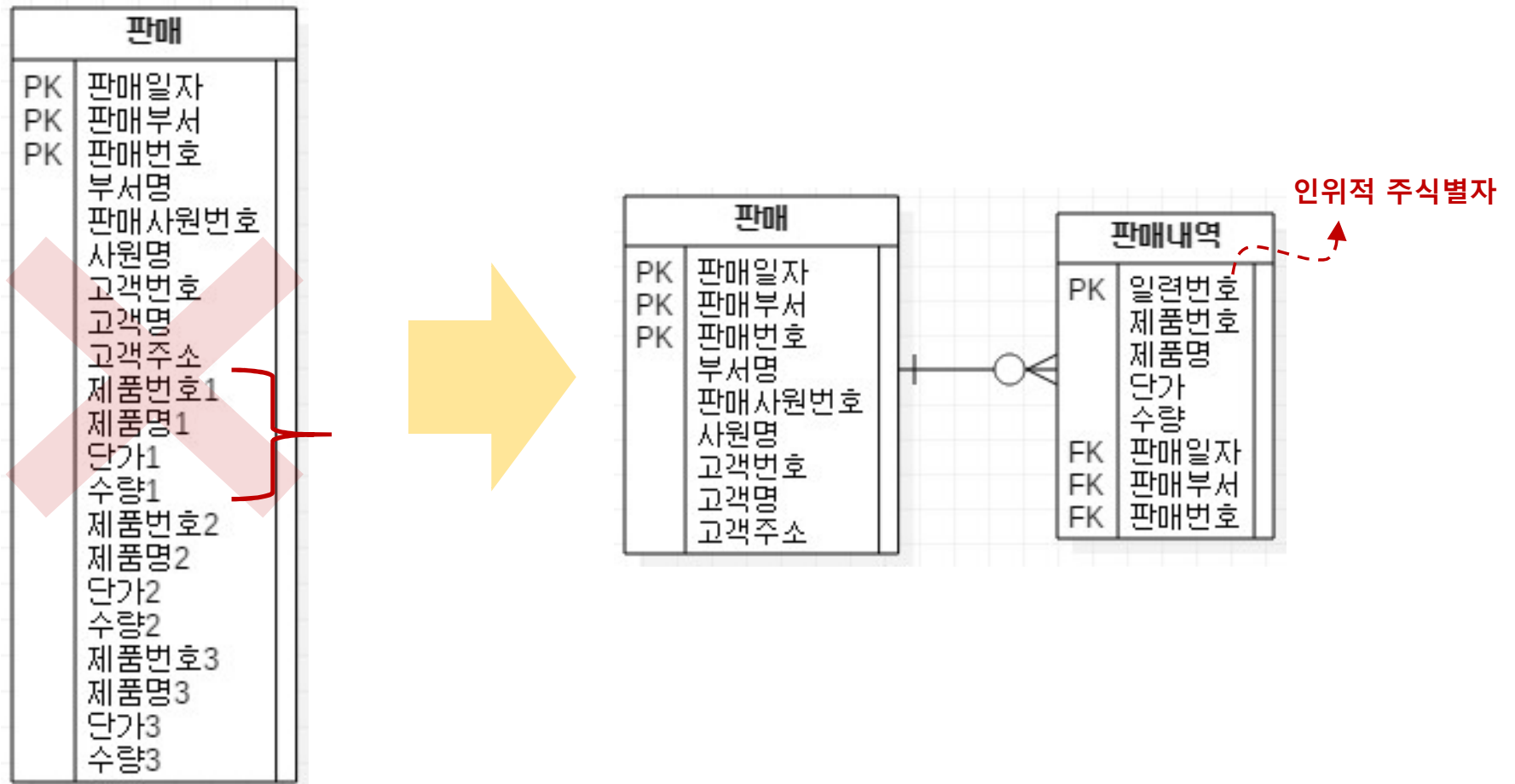
< 방법 1 >

인위적 주식별자

사원별 취미코드 (PK)	취미	사번(FK)
1	등산	1001
2	낚시	1001
3	테니스	1001
4	낚시	1002

< 방법 2 >

▶ 정규화 과정



<그림 2>

▶ 정규화 과정

✓ 제 2 정규화

주식별자가 아닌 속성 중에서 주식별자 전체가 아닌 일부 속성에 종속된 속성을 찾아 제거하는 것

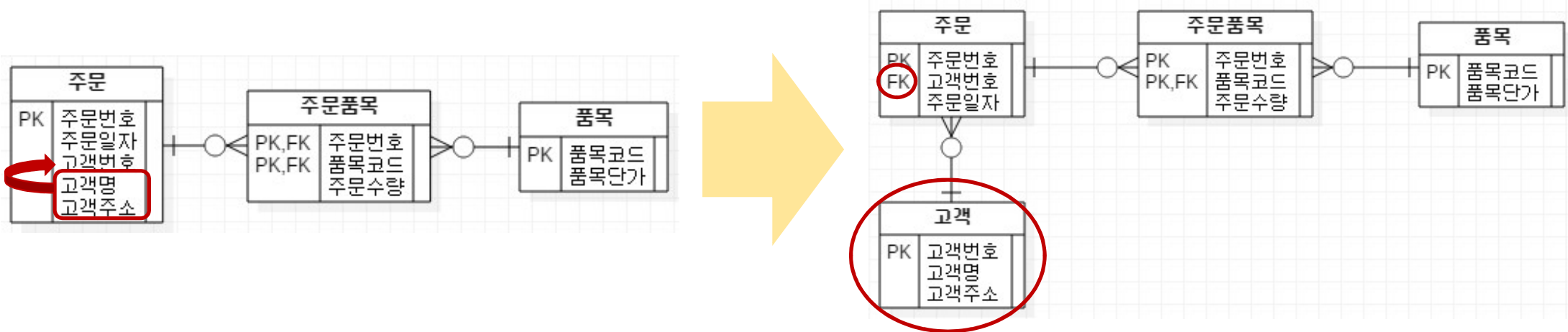


<그림 3>

▶ 정규화 과정

✓ 제 3 정규화

주식별자가 아닌 속성들 중에서 종속 관계에 있는 속성을 찾아 제거하는 것

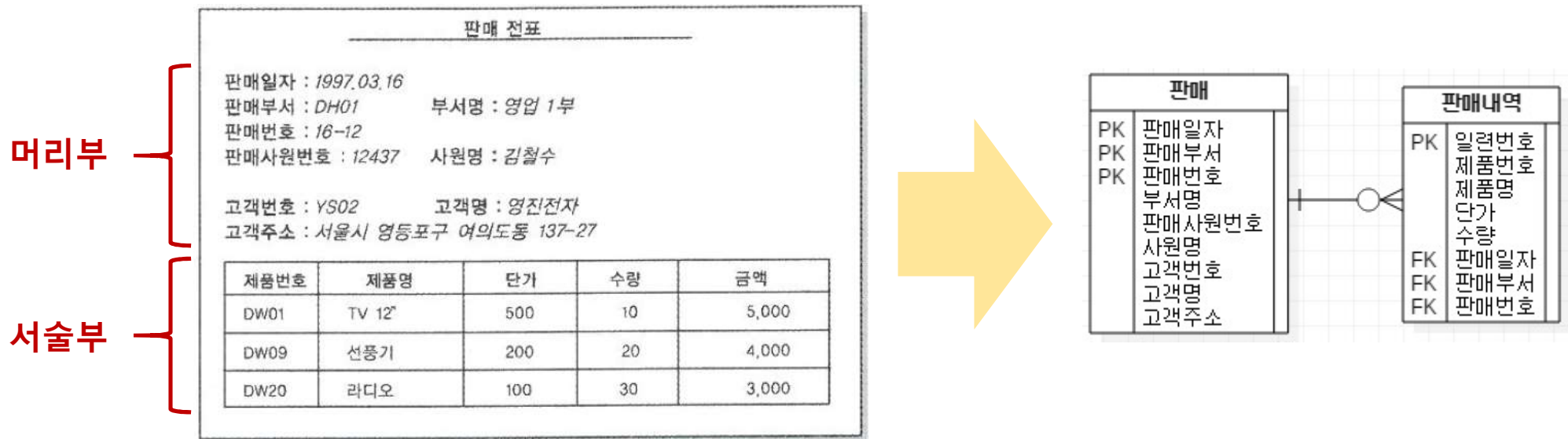


<그림 4>

▶ 정규화가 필요 없는 설계 기법

✓ 원칙 1

장부나 전표에서 머리부와 서술부는 별도의 엔티티로 저장한다.

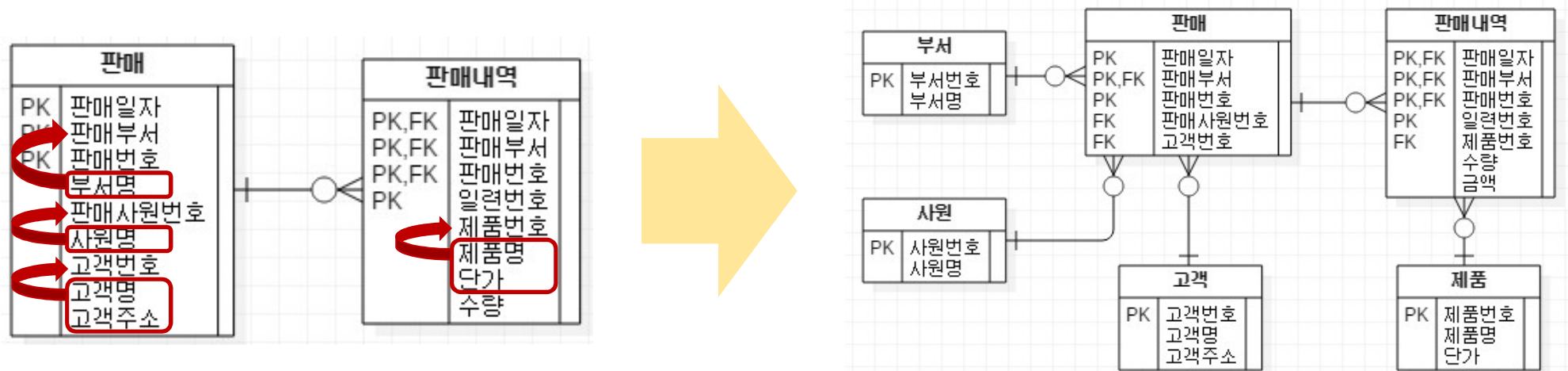


<그림 5>

▶ 정규화가 필요 없는 설계 기법

✓ 원칙 2

코드-코드 값의 관계에 있는 속성들은 별도의 엔티티로 구성한다.



<그림 6>

Chap04.

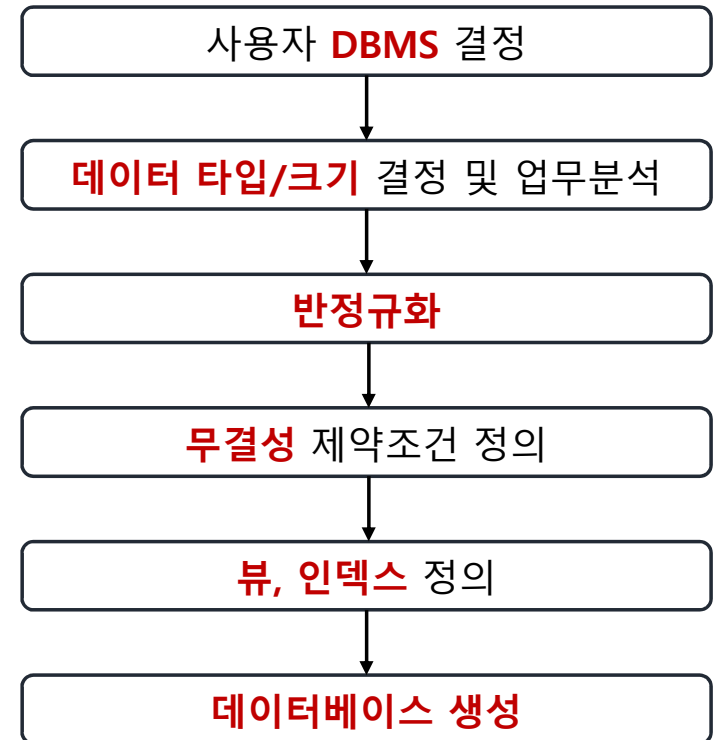
물리적 모델링

▶ 물리적 모델링 개요

✓ 물리적 모델링

논리적 설계의 산출물인 ERD의 요소들을
관계형 데이터베이스의 요소들로 전환하는 것

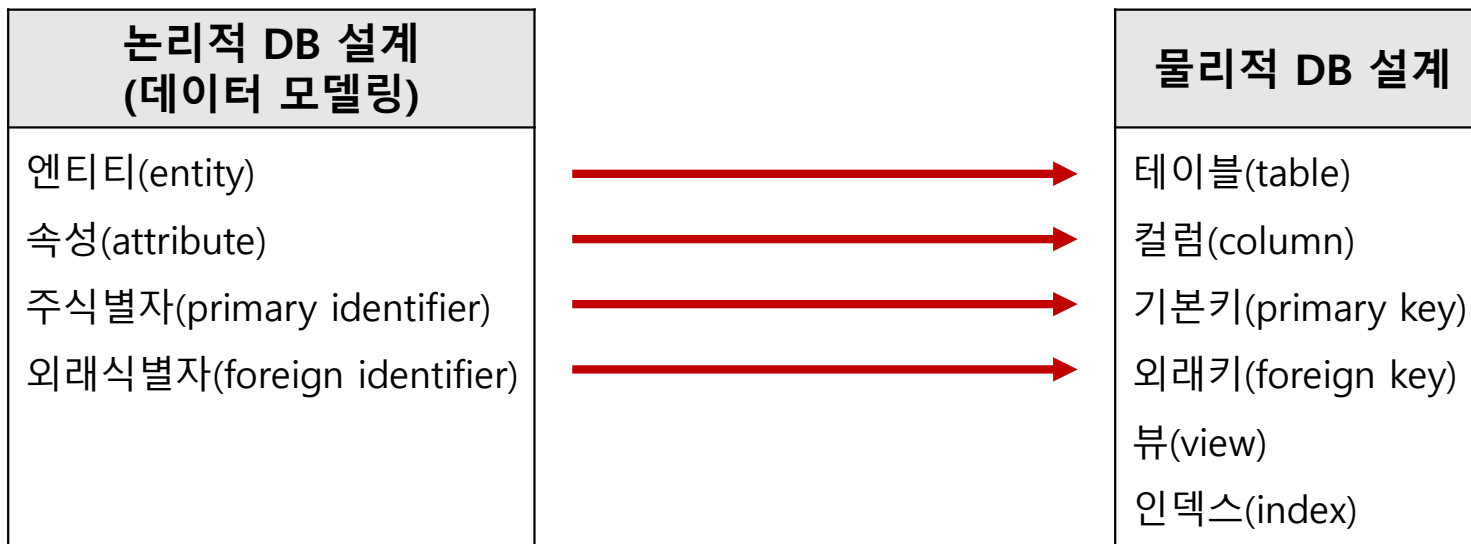
논리적 DB 설계 (데이터 모델링)	물리적 DB 설계
DBMS의 종류/제품에 상관 없이 진행 (ERD는 어떤 데이터베이스를 사용해도 적용가능)	특정 DBMS를 전제로 진행 (적용 DBMS의 특성을 고려)



<물리 모델링 과정>

▶ 논리적 모델링 VS 물리적 모델링

✓ ERD 요소 전환



▶ Oracle 데이터 타입

✓ 문자형 데이터 타입

데이터 유형	정의
CHAR(n)	고정 길이 데이터 타입(최대 2000byte) 지정된 길이보다 짧은 데이터 입력될 시 나머지 공간 공백으로 채워짐
VARCHAR2(n)	가변 길이 데이터 타입(최대 4000byte) 지정된 길이보다 짧은 데이터 입력될 시 나머지 공간은 채우지 않음
NCHAR(n)	고정 길이 유니코드 데이터 타입(최대 2000byte)
NVARCHAR2(n)	가변 길이 유니코드 데이터 타입(최대 4000byte)
LONG	가변 길이 데이터 타입(최대 2Gbyte)
CLOB	대용량 텍스트 데이터 타입(최대 4Gbyte)
NCLOB	대용량 텍스트 유니코드 데이터 타입(최대 4Gbyte)

▶ Oracle 데이터 타입

✓ 숫자형 데이터 타입

데이터 유형	정의
BINARY_FLOAT	부동 소수형 데이터 타입(4byte), 32bit 부동 소수
BINARY_DOUBLE	부동 소수형 데이터 타입(8byte), 64bit 부동 소수
NUMBER(P,S)	p, s로 표현 숫자 데이터 타입 p: 1 ~ 38, s: -84 ~ 127 p(precision) : 유효자리 수, s(scale) : 소수점 유효자리

✓ 이진 데이터 타입

데이터 유형	정의
RAW(n)	가변 길이 이진 데이터 타입(최대 2Gbyte)
LONGRAW	가변 길이 이진 데이터 타입(최대 4Gbyte)
BLOB	대용량의 바이너리 데이터를 저장하기 위한 데이터 타입(최대 4Gbyte)
BFILE	대용량의 바이너리 데이터를 파일형태로 저장하기 위한 데이터 타입(최대 4Gbyte)

▶ Oracle 데이터 타입

✓ 날짜형 데이터 타입

데이터 유형	정의
DATE	고정 길이 날짜
INTERVAL_YEAR	날짜(년도, 월)형태의 기간 표현 데이터 타입
INTERVAL_DAY	날짜 및 시간(요일, 시, 분, 초)형태의 기간 표현 데이터 타입
TIMESTAMP	밀리초(ms)까지 표현 데이터 타입
TIMESTAMP_WITH TIME_ZONE	날짜 및 시간대 형태의 데이터 타입
TIMESTAMP_WITH LOCAL TIME_ZONE	저장 시 데이터베이스 시간대를 준수, 조회 시 조회하는 클라이언트 시간 표현 데이터 타입

▶ 반정규화(De-Normalization)

- 정규화 작업이 완료된 후 데이터 물리 모델링 과정 중 시스템의 성능 향상, 개발 과정의 편의성, 운영의 단순화를 추구
- 중복은 감수하고 데이터베이스의 성능을 향상시키는 것 (특히 검색 속도)
- 정규화를 통한 데이터 무결성 유지도 중요하지만, 다수 사용자가 동시 이용하는 환경에서 일정 성능을 유지하는 것도 매우 중요

▶ 반정규화 사례 - 엔티티 통합 / 분할에 의한 반정규화

✓ CASE 1 - 엔티티의 통합

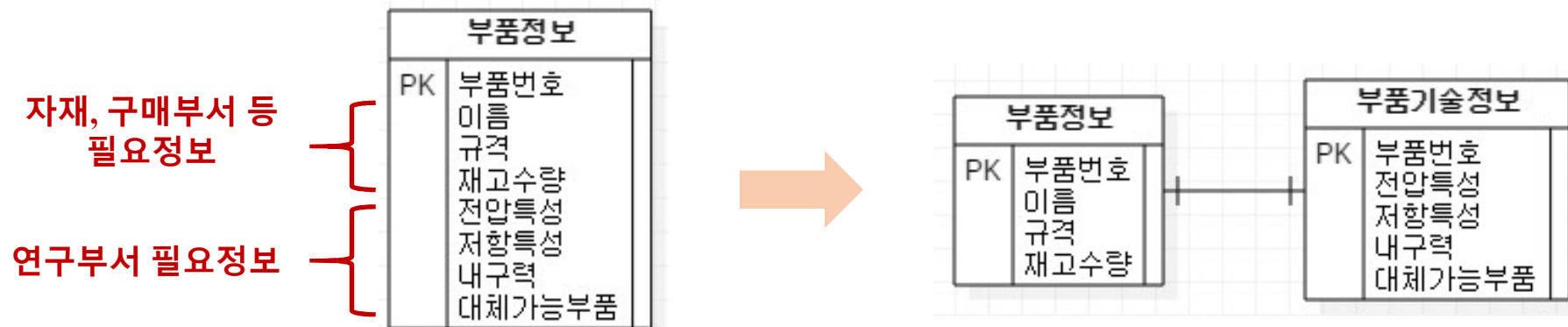
항상 혹은 대부분 조인에 의한 검색을 하고, 검색이 빈번히 이뤄지는 두 엔티티를 대상으로 함



▶ 반정규화 사례 - 엔티티 통합 / 분할에 의한 반정규화

✓ CASE 2 - 수직분할에 의한 반정규화

엔티티의 튜플 수 및 속성의 수가 매우 많고, 엔티티의 속성들이 그룹화되어 각 그룹이 특정 부서 혹은 응용 프로그램에 의해서만 사용될 때



▶ 반정규화 사례 - 엔티티 통합 / 분할에 의한 반정규화

✓ CASE 3 - 수평분할에 의한 반정규화

한 테이블 내에서 튜플의 조회 빈도에 따라 엔티티를 분할

도서정보	
도서번호	
제목	
저자	
장르	
가격	



도서정보 A	
도서번호	
제목	
저자	
장르	
가격	

자주 검색되는
상위 20%의 책

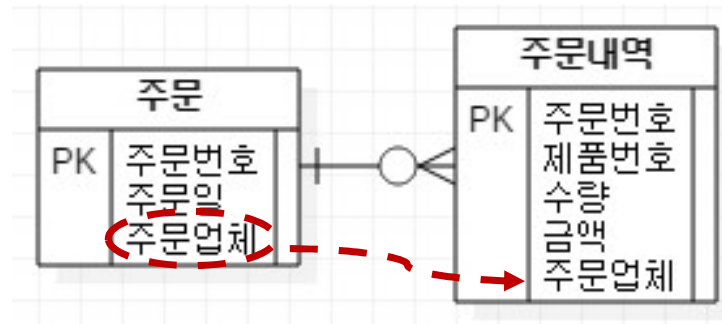
도서정보 B	
도서번호	
제목	
저자	
장르	
가격	

나머지
80%의 책

▶ 반정규화 사례 - 속성의 중복에 의한 반정규화

✓ CASE 4 - 속성의 중복에 의한 반정규화

조인을 이용하여 가져다 쓰는 속성의 수가 적을 때 엔티티의 통합은 비효율적이므로 속성을 중복하여 저장

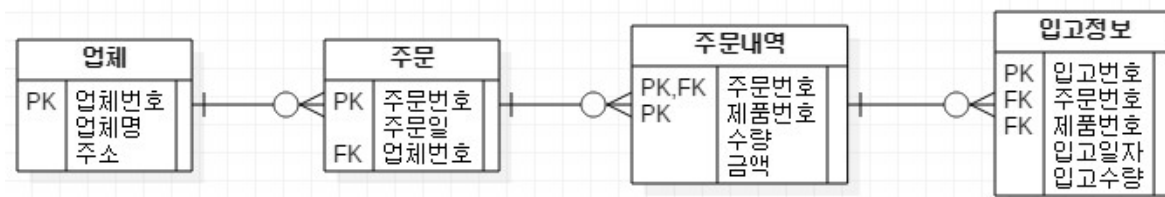


< 속성의 중복에 의한 반정규화 >

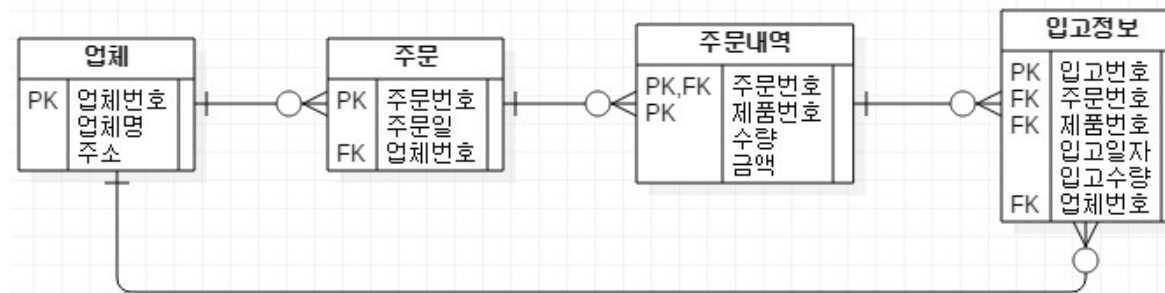
▶ 반정규화 사례 – 관계에 대한 반정규화

✓ CASE 5 - 관계에 대한 반정규화

속성의 중복에 의한 반정규화와 비슷하나 차이가 있다면 중복을 하는 속성이 다른 엔티티와의 관계를 맺어주기 위한 외래키로 사용 (조인해야 할 엔티티를 줄여줌)



< 조인 경로가 복잡한 정규화 >



< 관계에 대한 반정규화 >