

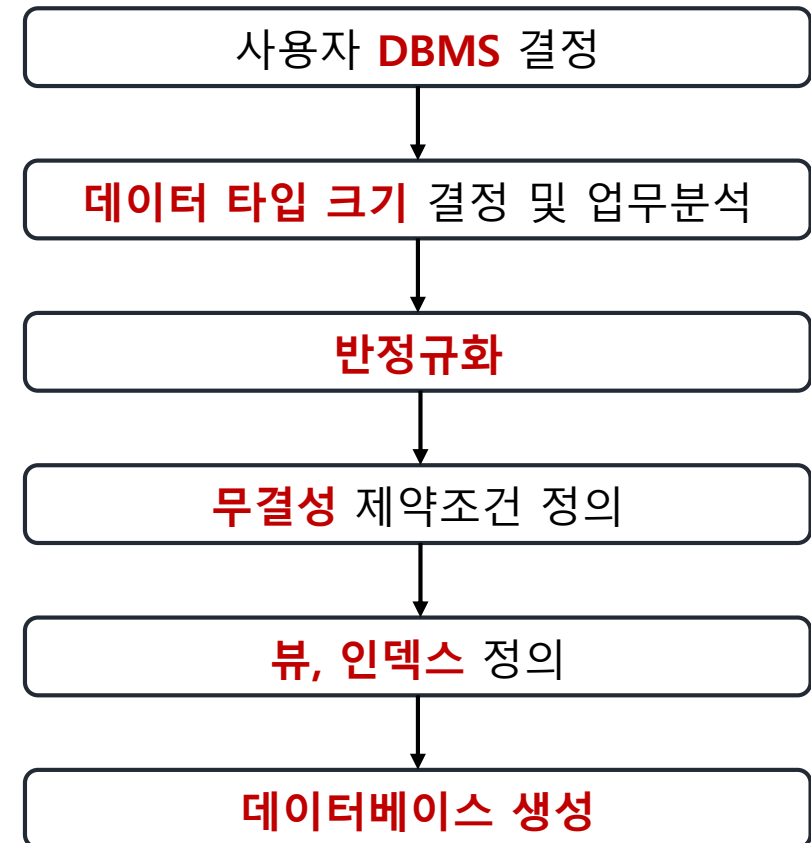
# Chap04. 물리적 모델링

# ▶ 물리적 모델링 개요

## ✓ 물리적 모델링이란 ?

논리적 설계의 산출물인 ERD의 요소들을 관계형 데이터베이스의 요소들로 전환하는 것

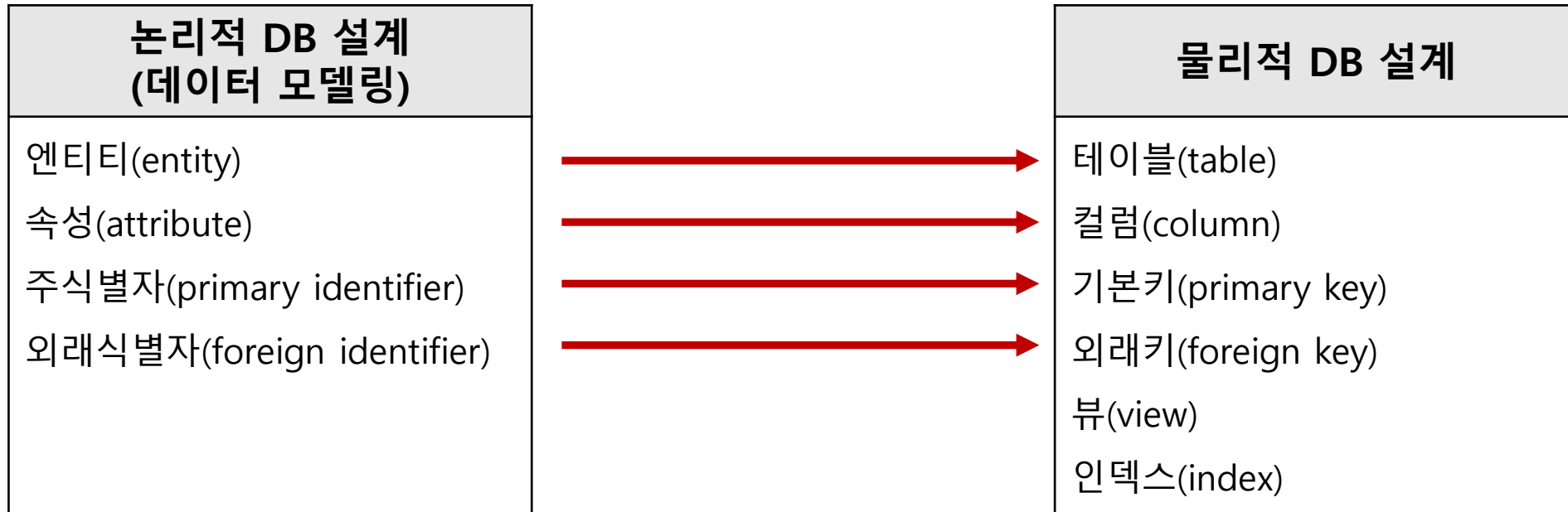
논리적 DB 설계 (데이터 모델링)	물리적 DB 설계
DBMS의 종류나 제품에 상관없이 진행 (ERD는 어떤 데이터베이스를 사용해도 적용가능)	특정 DBMS를 전제로 진행 (적용 DBMS의 특성을 고려)



<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 채워진다.
VARCHAR2(n)	가변 길이 데이터 타입(최대 4000byte)-지정된 길이보다 짧은 데이터 입력될 시 나머지 공간은 채우지 않는다.
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

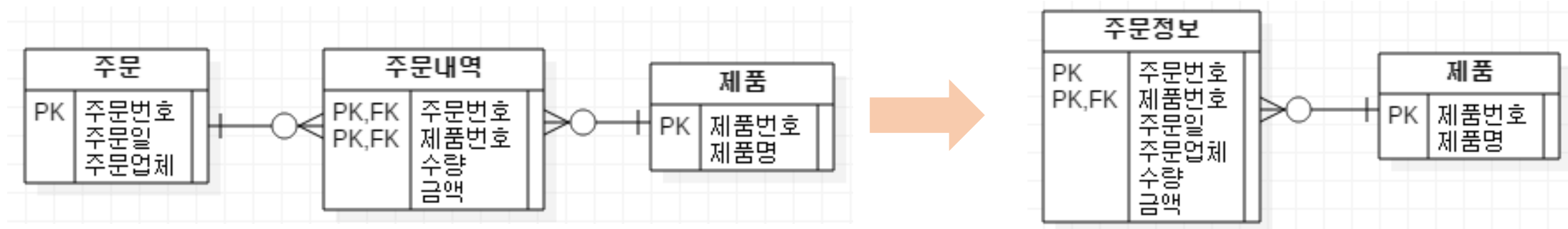
### ✓ 반정규화(De-Normalization)란 ?

- 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 이루어지는 두 개의 엔티티를 대상으로 한다.



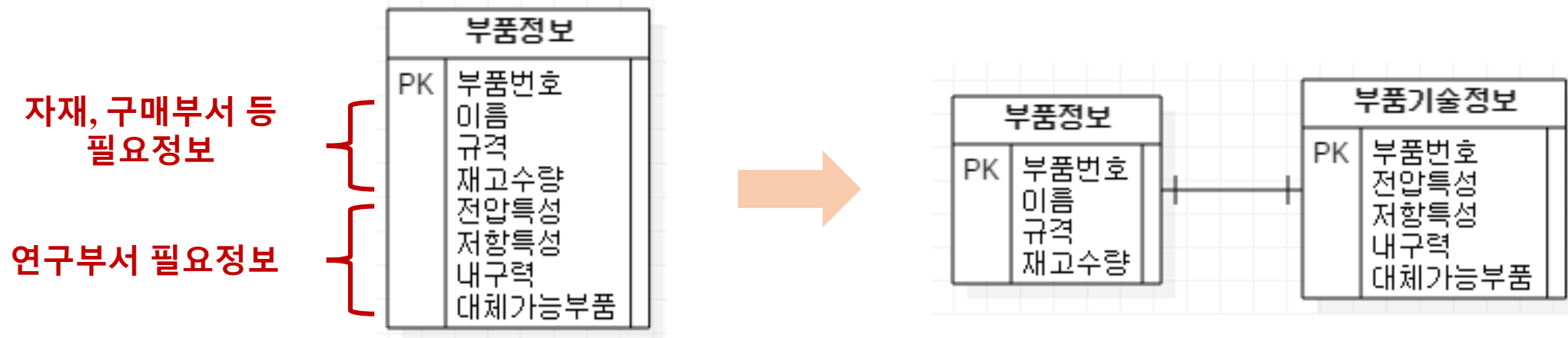
< 두 엔티티의 통합에 의한 반정규화 >



# ▶ 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

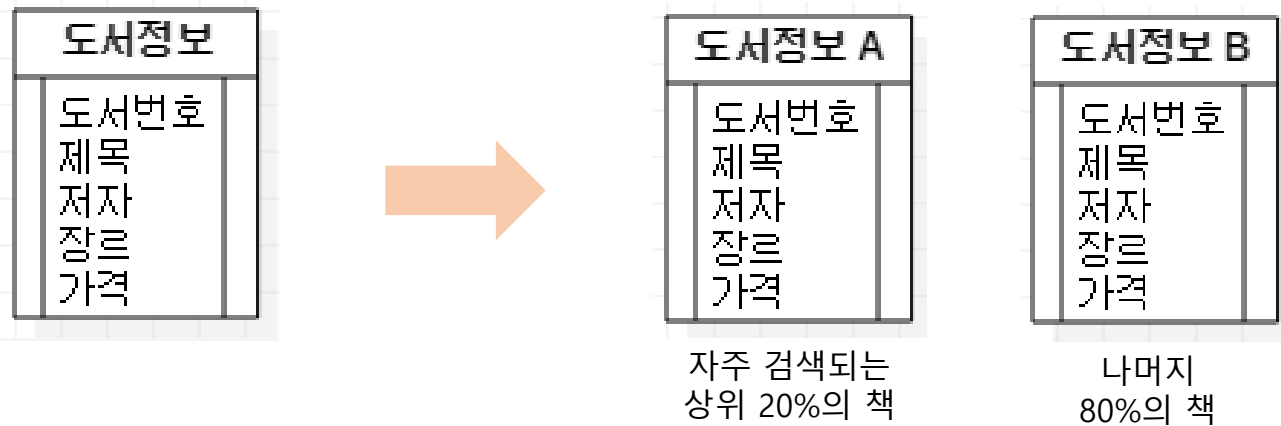


< 엔티티 수직분할에 의한 반정규화 >

# ▶ 반정규화 사례 - 엔티티 통합 / 분할에 의한 반정규화

## ✓ CASE 3 - 수평분할에 의한 반정규화

한 테이블 내에서 튜플의 조회 빈도에 따라 엔티티를 분할.

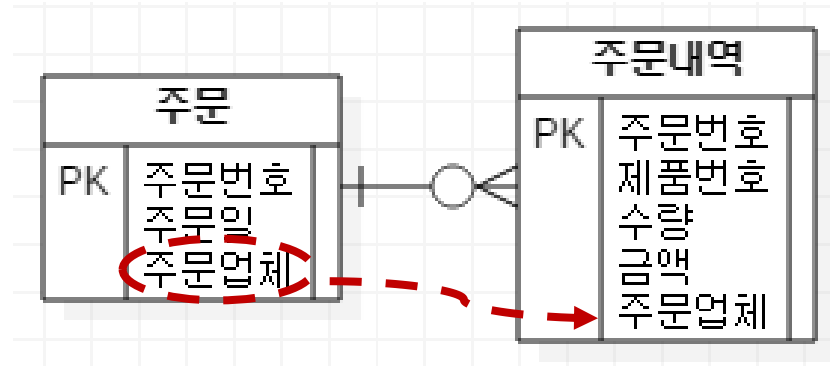


< 엔티티 수평분할에 의한 반정규화 >

## ▶ 반정규화 사례 - 속성의 중복에 의한 반정규화

### ✓ CASE 4 - 속성의 중복에 의한 반정규화

조인하여 가져다 쓰는 속성의 수가 적을 때는 엔티티의 통합은 비효율적이다.  
따라서 속성을 중복하여 저장한다.

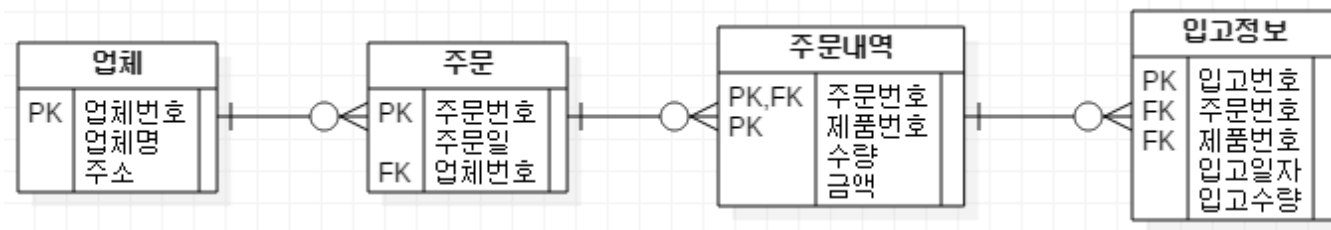


< 속성의 중복에 의한 반정규화 >

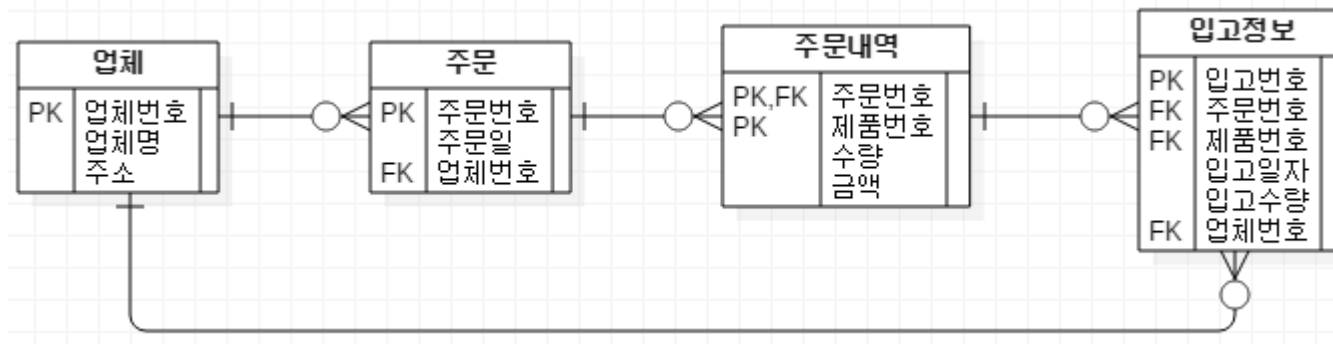
# ▶ 반정규화 사례 - 관계에 대한 반정규화

## ✓ CASE 5 - 관계에 대한 반정규화

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



< 조인 경로가 복잡한 정규화 >



< 관계에 대한 반정규화 >