

# 제 5 장



## 데이터베이스 설계와 ER 모델

5.4 논리적 설계: ER 스키마를 관계 모델의 릴레이션으로 사상

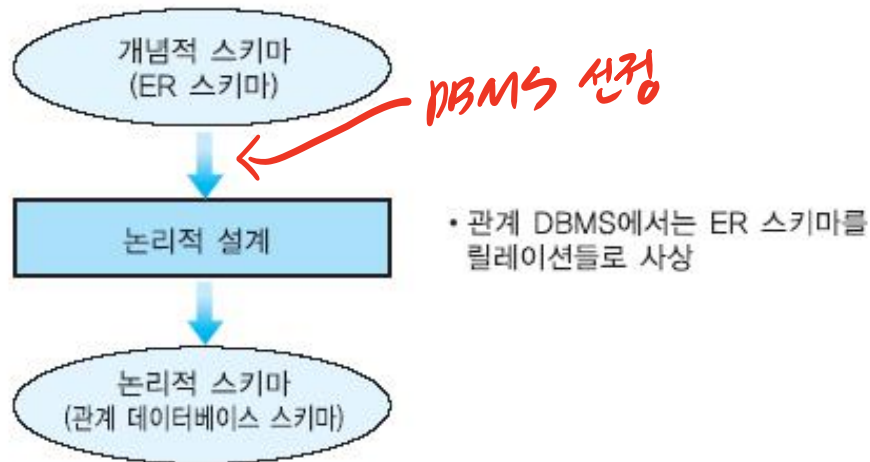
- 연습문제

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상

### □ ER 스키마를 관계 모델의 릴레이션으로 사상

- ✓ 논리적 설계 단계에서는 ER 스키마를 관계 데이터 모델의 릴레이션들로 사상함
  - ✓ ER 스키마에는 엔티티 타입과 관계 타입이 존재하지만 / 관계 데이터베이스에는 엔티티 타입과 관계 타입을 구분하지 않고 릴레이션들만 있음
  - ✓ 릴레이션으로 사상할 대상이 ER 스키마에서 엔티티 타입인지 또는 관계 타입인지, 엔티티 타입이라면 정규 엔티티 타입인지 또는 약한 엔티티 타입인지, 관계 타입이라면 2진 관계 타입인지 3진 이상의 관계 타입인지, 애트리뷰트가 단일 값 애트리뷰트인지 또는 다치 애트리뷰트인지 등에 따라 사상하는 방법이 달라짐
- ER 모델을 릴레이션들로 사상하는 7개의 단계로 이루어진 알고리즘

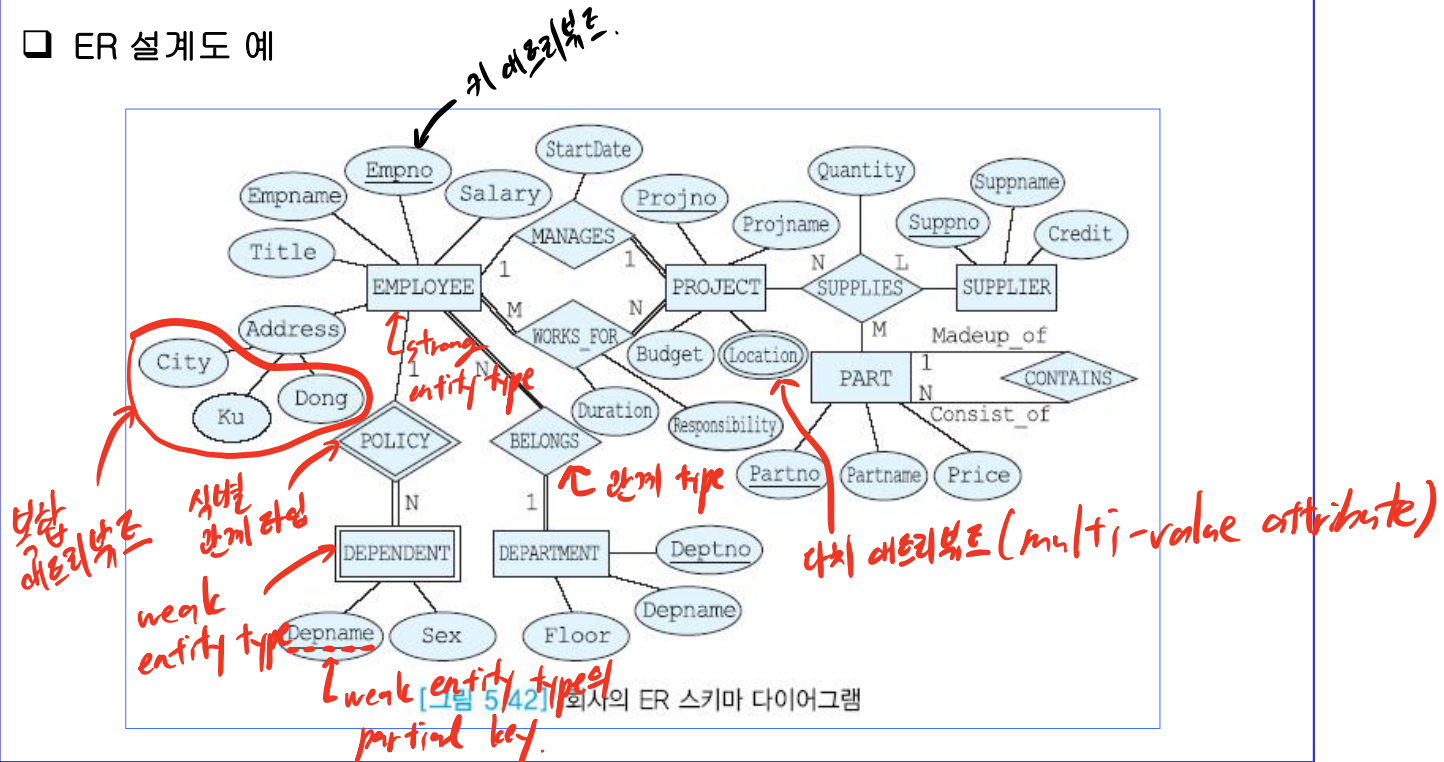
## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



[그림 5.43] 논리적 설계

## 개념 설계 결과: ER Diagram 예

### □ ER 설계도 예



## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

〈표 5.4〉 알고리즘의 각 단계에서 릴레이션으로 사상되는 ER 스키마의 대상

사상할 대상	알고리즘의 단계
엔티티 타입과 단일 값 애트리뷰트	단계 1: 정규 엔티티 타입
	단계 2: 약한 엔티티 타입
2진 관계 타입	단계 3: 2진 1:1 관계 타입
	단계 4: 정규 2진 1:N 관계 타입
	단계 5: 2진 M:N 관계 타입
3진 이상의 관계 타입	단계 6: 3진 관계 타입
다치 애트리뷰트	단계 7: 다치 애트리뷰트

← 키 조합 (파티터 ⊕ 외래키)  
 ← 키 조합 (외래키 ⊕ 외래키)  
 ← 키 조합 (외래키 ⊕ 외래키)  
 ← 키 조합 (다치 애트리뷰트 ⊕ 외래키)

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

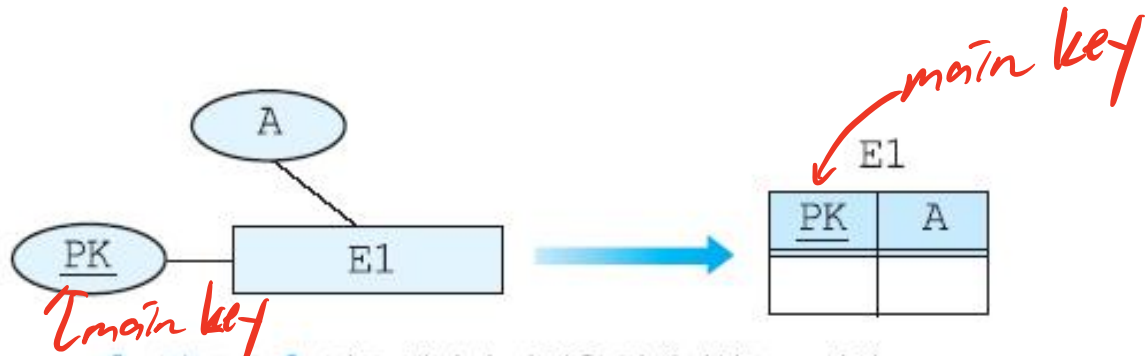
### □ ER-관계 사상 알고리즘

#### 단계 1: 정규 엔티티 타입과 단일 값 애트리뷰트

- ✓ ER 스키마의 각 정규 엔티티 타입 E에 대해 하나의 릴레이션 R을 생성함
- ✓ E에 있던 단순 애트리뷰트들을 릴레이션 R에 모두 포함시킴 (여기서 포함시킴)
- ✓ E에서 복합 애트리뷰트는 그 복합 애트리뷰트를 구성하는 단순 애트리뷰트들만 릴레이션 R에 포함시킴
- ✓ E의 기본 키가 릴레이션 R의 기본 키가 됨

해상!  
⇒ 복합 애트리뷰트들은  
릴레이션에서 제외됨.

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



[그림 5.44] 정규 엔티티 타입을 릴레이션으로 사상

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

### □ ER-관계 사상 알고리즘(계속)

#### 단계 2: 약한 엔티티 타입과 단일 값 애트리뷰트

✓ ER 스키마에서 소유 엔티티 타입 E를 갖는 각 약한 엔티티 타입 W에 대하여 릴레이션 R을 생성함

✓ W에 있던 모든 단순 애트리뷰트들을 릴레이션 R에 포함시킴

✓ 소유 엔티티 타입에 해당하는 릴레이션의 기본 키를 약한 엔티티 타입에 해당하는 릴레이션에 외래 키로 포함시킴

✓ 약한 엔티티 타입에 해당하는 릴레이션 R의 기본 키는 약한 엔티티 타입의 부분 키와 소유 엔티티 타입에 해당하는 릴레이션을 참조하는 외래 키의 조합으로 이루어짐

★ 약한 엔티티 타입도 테이블로 만든다.

약한 엔티티 타입에 대한 테이블

해상!!!

해상!!!

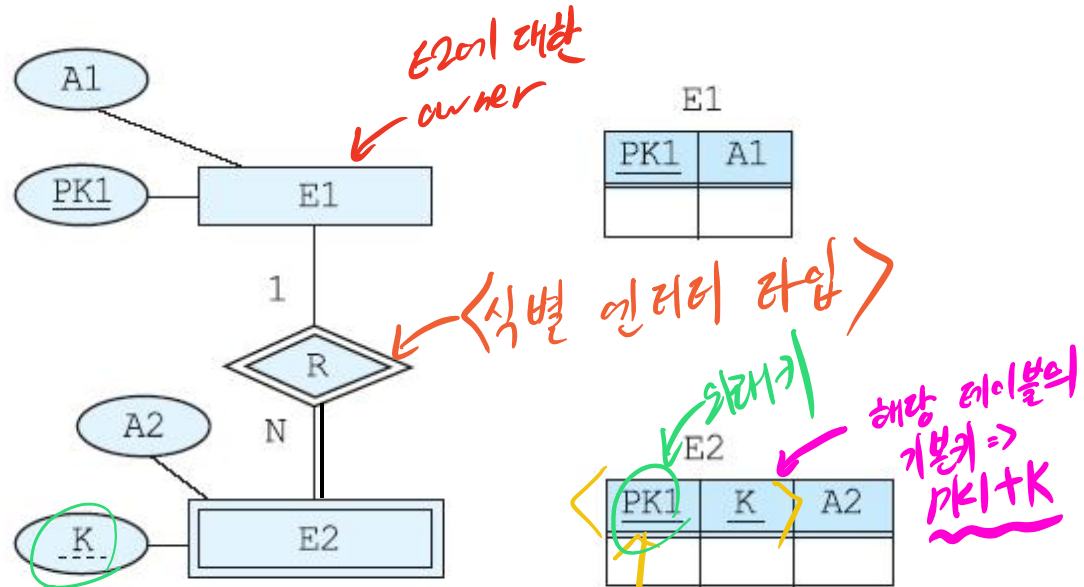
ER 모델에서  
파생으로  
생성됨

★ 부분키 + 외래키 (소유 엔티티 타입의 주키)

= 약한 엔티티 타입 테이블의 주키



## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



[그림 5.45] 약한 엔티티 타입을 릴레이션으로 사상

# 포기상 무엇인가가 많은 엔터티 타입의 레이블이 영역 더 많아진다!!!

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

대응수 (카드 날리터)

### □ ER-관계 사상 알고리즘(계속)

#### 단계 3: 2진 1:1 관계 타입

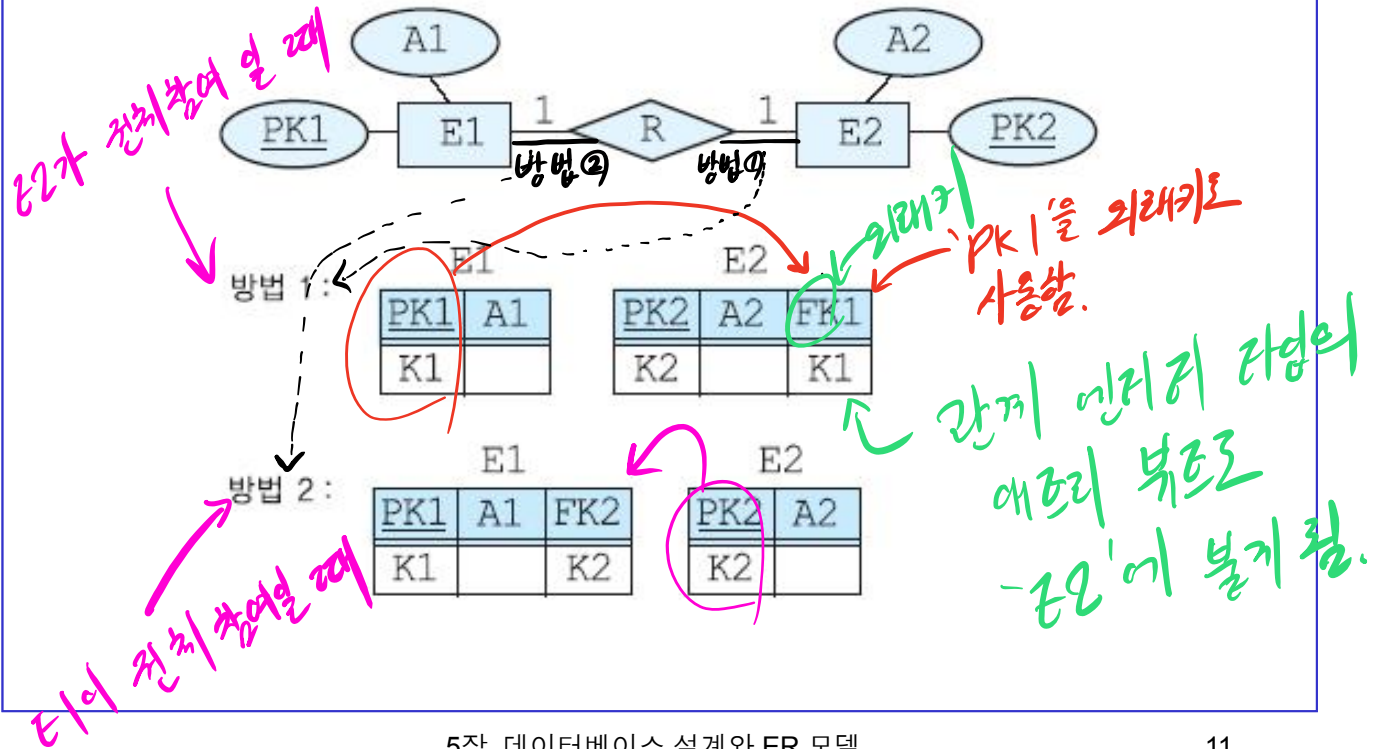
- ✓ ER 스키마의 각 2진 1:1 관계 타입 R에 대하여, R에 참여하는 엔티티 타입에 대응되는 릴레이션(S와 T)을 찾음

① S와 T 중에서 관계 타입에 완전하게 참여하는 릴레이션은 S의 역할을 하는 릴레이션으로 선택함

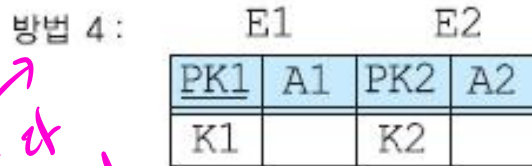
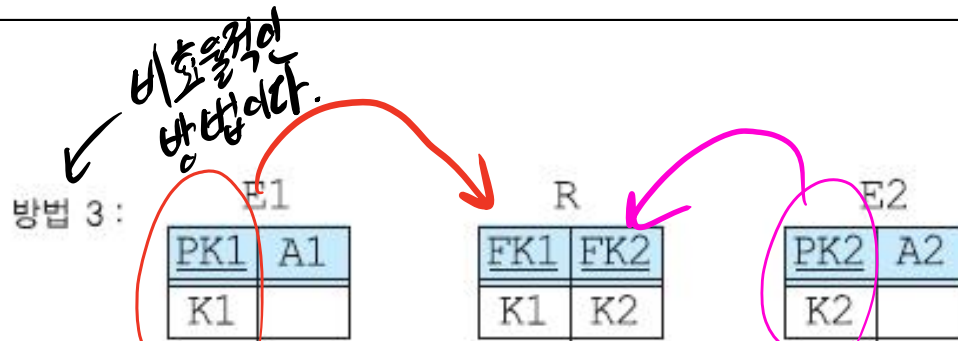
② T의 기본 키를 S 릴레이션에 외래 키로 포함시킴

- ✓ 관계 타입 R이 가지고 있는 모든 단순 애트리뷰트(복합 애트리뷰트를 갖고 있는 경우에는 복합 애트리뷰트를 구성하는 단순 애트리뷰트)들을 S에 대응되는 릴레이션에 포함시킴
- ✓ 두 엔티티 타입이 관계 타입 R에 완전하게 참여할 때는 두 엔티티 타입과 관계 타입을 하나의 릴레이션으로 합치는 방법도 가능함

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



← <주키 설정은 둘 중 하나로 한다.>

← E1, E2 둘 다  
전체 참여 일대일

[그림 5.46] 2진 1:1 관계 타입을 릴레이션으로 사상

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

### □ ER-관계 사상 알고리즘(계속)

#### 단계 4: 정규 2진 1:N 관계 타입

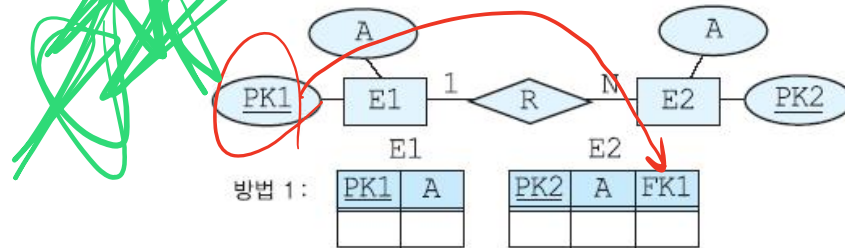
① 정규 2진 1:N 관계 타입 R에 대하여 N측의 참여 엔티티 타입에 대응되는 릴레이션 S를 찾음

② 관계 타입 R에 참여하는 1측의 엔티티 타입에 대응되는 릴레이션 T의 기본 키를 릴레이션 S에 외래 키로 포함시킴

- ✓ N측의 릴레이션 S의 기본 키를 1측의 릴레이션 T에 외래 키로 포함시키면  
    애트리뷰트에 값들의 집합이 들어가거나 정보의 중복이 많이 발생함
- ✓ 관계 타입 R이 가지고 있는 모든 단순 애트리뷰트(복합 애트리뷰트를 갖고  
    있는 경우에는 복합 애트리뷰트를 구성하는 단순 애트리뷰트)들을 S에  
    해당하는 릴레이션에 포함시킴

해상!!

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



비효율적이라  
선택안함.



recursive relation type.

[그림 5.47] 정규 2진 1:N 관계 타입을 릴레이션으로 사상

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

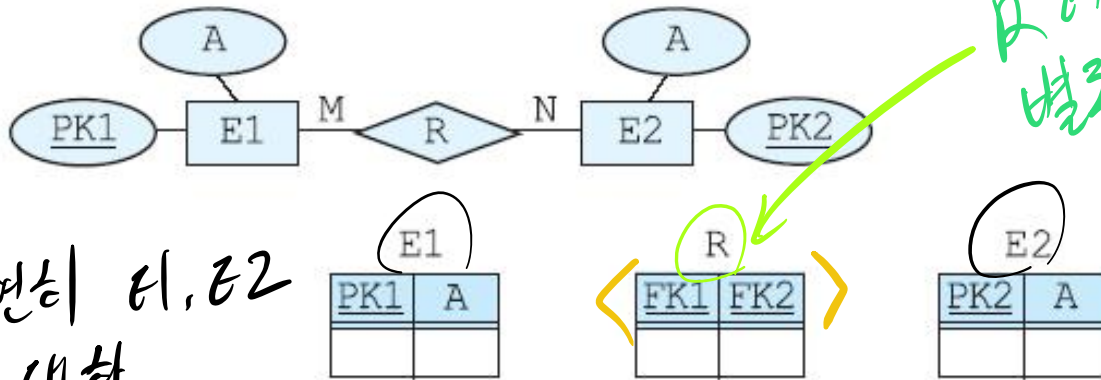
### □ ER-관계 사상 알고리즘(계속)

#### 단계 5: 2진 M:N 관계 타입

- ✓ 2진 M:N 관계 타입 R에 대해서는 릴레이션 R을 생성함
- ✓ 참여 엔티티 타입에 해당하는 릴레이션들의 기본 키를 릴레이션 R에 외래 키로 포함시키고, 이들의 조합이 릴레이션 R의 기본 키가 됨
- ✓ 관계 타입 R이 가지고 있는 모든 단순 애트리뷰트(복합 애트리뷰트를 갖고 있는 경우에는 복합 애트리뷰트를 구성하는 단순 애트리뷰트)들을 릴레이션 R에 포함시킴

관계 타입 엔티티'에 대한 테이블을 별도로 만든다.

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



※ 당연히 E1, E2  
에 대한  
테이블도 각각  
만들어야 함.

그림 5.48] 2진 M:N 관계 타입을 릴레이션으로 사상



## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

### □ ER-관계 사상 알고리즘(계속)

#### 단계 6: 3진 이상의 관계 타입

① 3진 이상의 각 관계 타입 R에 대하여 릴레이션 R을 생성함

② 관계 타입 R에 참여하는 모든 엔티티 타입에 대응되는 릴레이션들의 기본 키를 릴레이션 R에 외래 키로 포함시킴

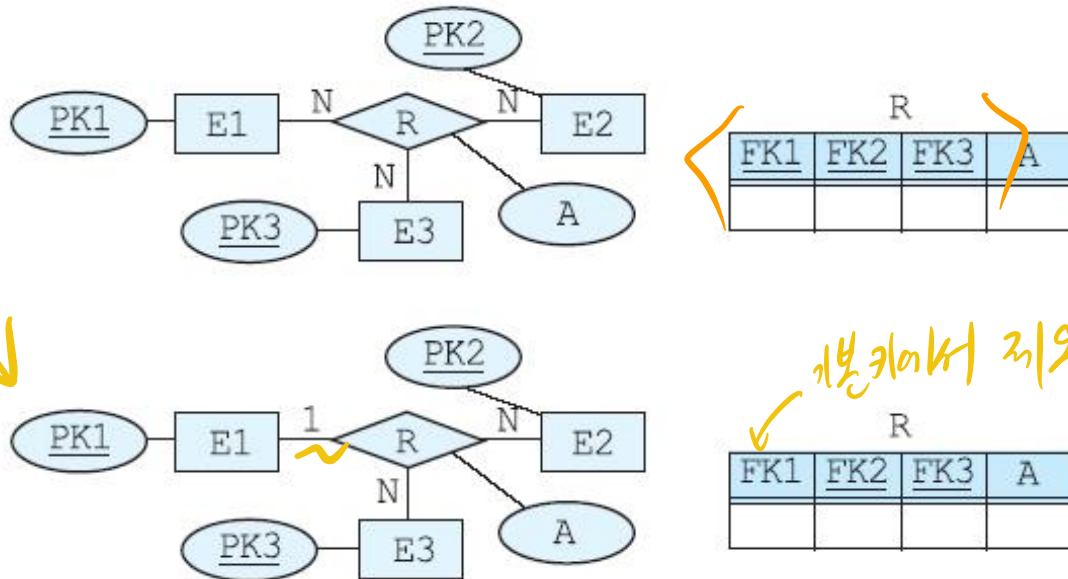
✓ 관계 타입 R이 가지고 있는 모든 단순 애트리뷰트(복합 애트리뷰트를 갖고 있는 경우에는 복합 애트리뷰트를 구성하는 단순 애트리뷰트)들을 릴레이션 R에 포함시킴

③ 일반적으로 외래 키들의 조합이 릴레이션 R의 기본 키가 됨

✓ 관계 타입 R에 참여하는 엔티티 타입들의 카디널리티가 1:N:N이면 카디널리티가 1인 릴레이션의 기본 키를 참조하는 외래 키를 제외한 나머지 외래 키들의 조합이 릴레이션 R의 기본 키가 됨

해상!!

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



[그림 5.49] 3진 이상의 관계 타입을 릴레이션으로 사상

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

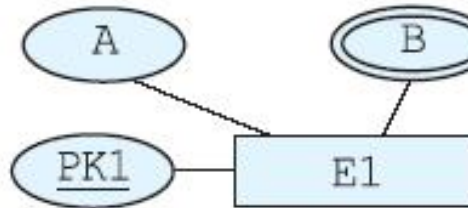
### □ ER-관계 사상 알고리즘(계속)

단계 7: 다치 애트리뷰트 (다중 값을 가지는 대외키부르)

- ✗ 각 다치 애트리뷰트에 대하여 릴레이션 R을 생성함
  - ✓ 다치 애트리뷰트에 해당하는 애트리뷰트를 릴레이션 R에 포함시키고, 다치 애트리뷰트를 애트리뷰트로 갖는 엔티티 타입이나 관계 타입에 해당하는 릴레이션의 기본 키를 릴레이션 R에 외래 키로 포함시킴
  - ✓ 릴레이션의 R의 기본 키는 다치 애트리뷰트와 외래 키의 조합

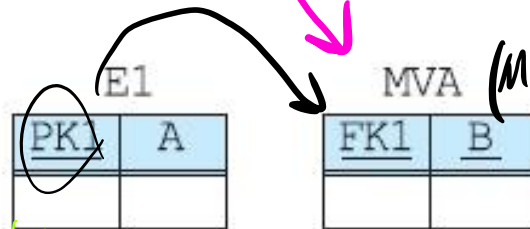
< 예는 기존 엔티티 타입에 대한 레이블에서  
변화하게 된다!! >

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



<여를 위한  
데이터를 별도로  
만들고, 기존 인터리 타입에  
대한 데이터에서  
빠지게 된다.>

<이 경우를 통해,  
에트리뷰트에 대한  
레이블로 존재할 수 있다는  
것을 알 수 있다.>

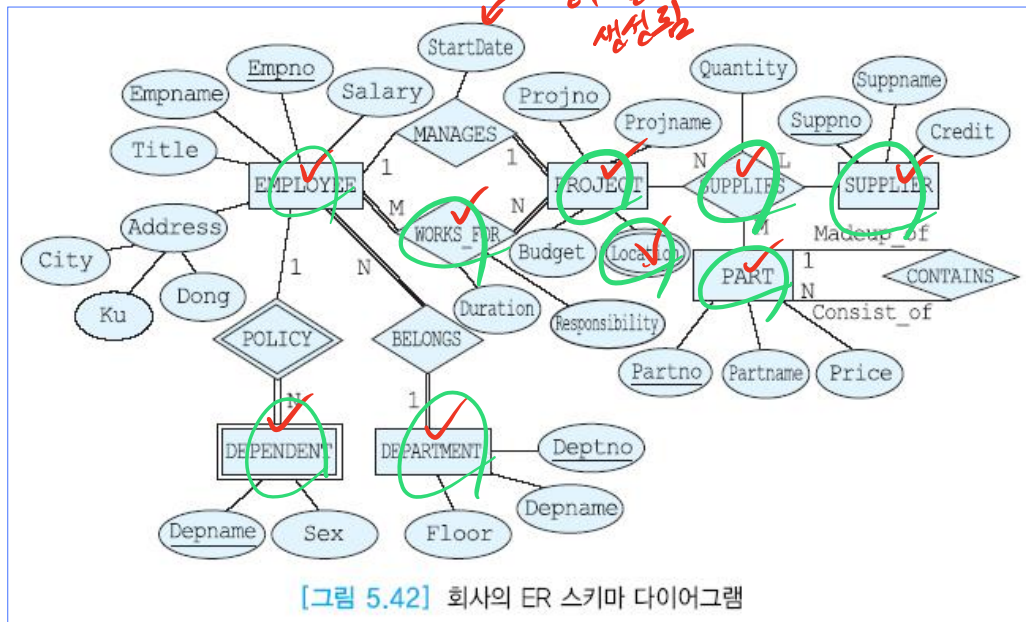


(Multi-Value Attribute)

[그림 5.50] 다치 애트리뷰트를 릴레이션으로 사상

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

### □ ER 설계도 예



## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

### ❑ 데이터베이스 설계 사례에 알고리즘 적용

#### 단계 1: 정규 엔티티 타입과 단일 값 애트리뷰트

**EMPLOYEE**(Empno, Empname, Title, City, Ku, Dong,  
Salary)

**PROJECT**(Projno, Projname, Budget)

**DEPARTMENT**(Deptno, Deptname, Floor)

**SUPPLIER**(Suppno, Suppname, Credit)

**PART**(Partno, Partname, Price)

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

### ❑ 데이터베이스 설계 사례에 알고리즘 적용(계속)

단계 2: 약한 엔티티 타입과 단일 값 애트리뷰트

DEPENDENT(Empno, Depname, Sex)

단계 3: 2진 1:1 관계 타입

PROJECT(Projno, Projname, Budget, StartDate, Manager)

단계 4: 정규 2진 1:N 관계 타입

EMPLOYEE(Empno, Empname, Title, City, Ku, Dong,  
Salary, Dno)

PART(Partno, Partname, Price, Subpartno)

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

### □ 데이터베이스 설계 사례에 알고리즘 적용(계속)

단계 5: 2진 M:N 관계 타입

WORKS\_FOR(Empno, Projno, Duration, Responsibility)

단계 6: 3진 이상의 관계 타입

SUPPLY(Suppno, Projno, Partno, Quantity)

단계 7: 다치 애트리뷰트

PROJ\_LOC(Projno, Location)



## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)

- 회사 ER 스키마는 관계 데이터베이스에서 총 9개의 릴레이션으로 사상되었음

EMPLOYEE(Empno, Empname, Title, City, Ku, Dong,  
Salary, Dno)

PROJECT(Projno, Projname, Budget, StartDate, Manager)

DEPARTMENT(Deptno, Deptname, Floor)

SUPPLIER(Suppno, Suppname, Credit)

PART(Partno, Partname, Price, Subpartno)

DEPENDENT(Empno, Depname, Sex)

WORKS\_FOR(Empno, Projno, Duration, Responsibility)

SUPPLY(Suppno, Projno, Partno, Quantity)

PROJ\_LOC(Projno, Location)

## 5.4 ER 스키마를 관계 모델의 릴레이션으로 사상(계속)



[그림 5.51] ER 개념과 데이터베이스 개념들의 대응 관계

관계 엔티티 타입

=> 테이블의 '외래키'로 나타낸다!!