5장. 데이터베이스 설계와 ER모델

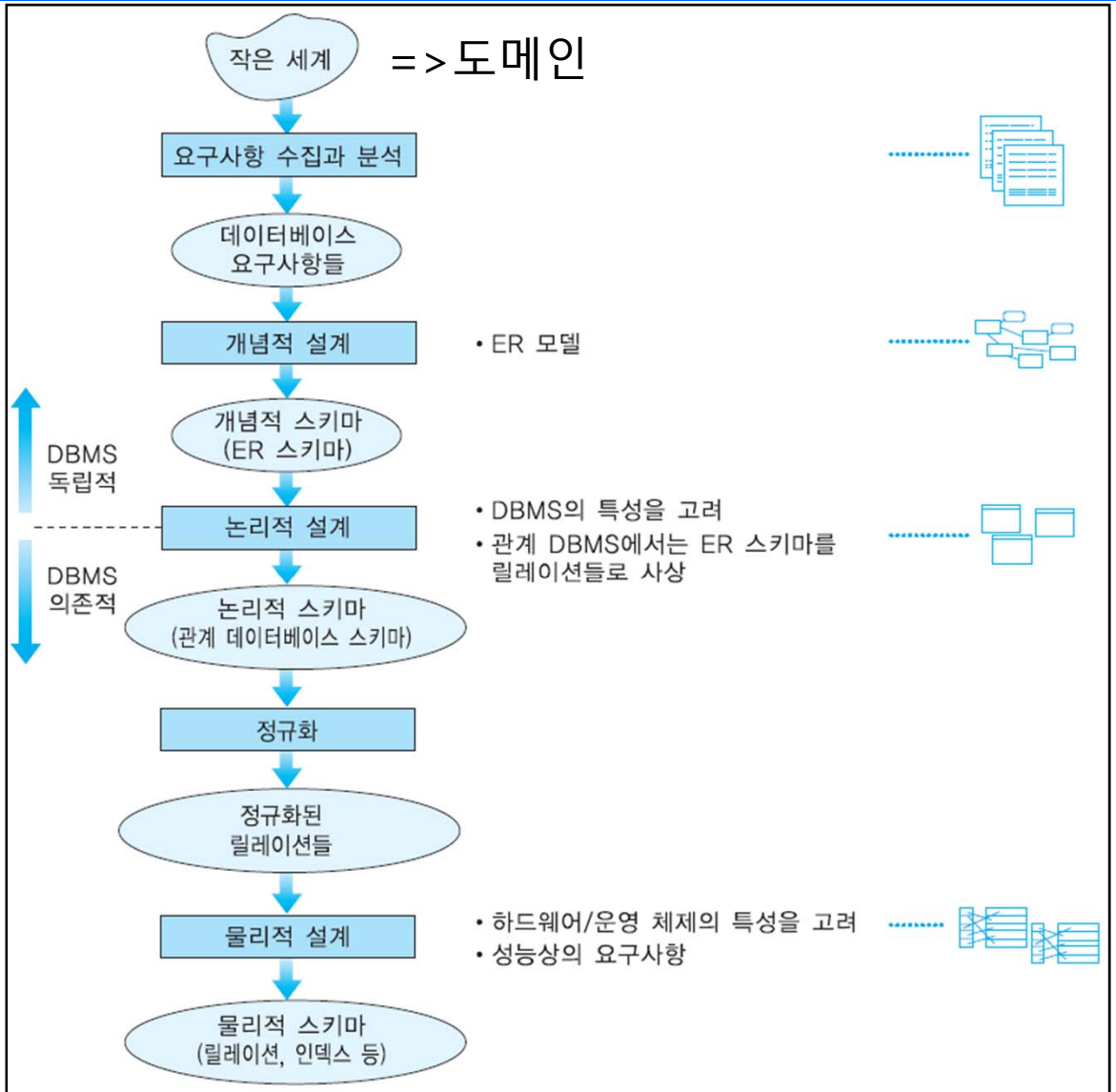
데이터베이스 설계와 ER모델

❑ 데이터베이스 설계

- ✓ 개념적 데이터베이스 설계와 물리적 데이터베이스 설계로 구분
  - ✓개념적 데이터베이스 설계 : 정보 사용의 모델을 개발하는 과정
  - ✓물리적 데이터베이스 설계 : 물리적인 저장 장치와 접근 방식

- ✓ 개념적 데이터베이스 설계 과정에서 조직체의 엔티티, 관계, 프로세스, 무결성 제약조건 등을 나타내는 추상화 모델을 구축
  - ✓엔티티: 서로 구분이 되면서 조직체에서 데이터베이스에 나타내려는 객체(사람, 장소, 사물 등)를 의미
  - ✓관계: 두 개 이상의 엔티티들 간의 연관을 나타냄
  - ✓프로세스: 관련된 활동
  - ✓무결성 제약조건: 데이터의 정확성과 비즈니스 규칙을 의미
  - ✓엔티티-관계(ER: Entity-Relationship) 모델

데이터베이스 설계의 개요



[그림 5.1] 데이터베이스 설계의 주요 단계

데이터베이스 설계의 개요

❑ 요구사항 수집과 분석

- ✓ 기존의 문서 조사, 인터뷰나 설문 조사 등이 시행
- ✓ 요구사항에 관한 지식을 기반으로
  - ✓**관련 있는 엔티티들**과 이들의 **애트리뷰트들**이 무엇인가,
    엔티티들 간의 관계가 무엇인가 등을 파악
- ✓ 데이터 처리에 관한 요구사항에 대하여
  - ✓전형적인 연산들은 무엇인가
  - ✓연산들의 의미,
  - ✓접근하는 데이터의 양 등을 분석

데이터베이스 설계의 개요

- **개념적 설계**
  - ✓ 한 조직체에서 사용되는 **정보의 모델**을 **구축**하는 과정
  - ✓ 사용자들의 **요구사항 명세로부터 개념적 스키마가 만들어짐**
    - ✓높은 추상화 수준의 데이터 모델을 기반으로 정형적인 언어로 데이터 구조를 명시함 (대표적인 데이터 모델=> ER 모델)

- **개념적 설계의 단계에서는**
  - ✓ 엔티티 타입, 관계 타입, 애트리뷰트들을 식별
  - ✓ 애트리뷰트들의 도메인을 결정
  - ✓ 후보 키와 기본 키 애트리뷰트들을 결정함
  - ✓ 완성된 개념적 스키마(ER 스키마)는 ER 다이어그램으로 표현됨

데이터베이스 설계의 개요

❑ 논리적 설계

- ✓ 데이터베이스 관리를 위해 선택한 DBMS의 데이터 모델을 사용하여 논리적 스키마를 생성함
  - ✓**ER 모델**로 표현된 **개념적 스키마**를 **관계 데이터베이스 스키마로 사상함**
- ✓ 관계 데이터베이스 스키마를 더 좋은 관계 데이터베이스 스키마로 변환하기 위해서 **정규화 과정을 적용함**

데이터베이스 설계의 개요

❑ 물리적 설계

- ✓ 처리 요구사항들을 만족시키기 위해 저장 구조와 접근 경로 등을 결정함
- ✓ 성능상의 주요 기준은 몇 가지로 구분할 수 있음
  - 응답 시간: 질의와 갱신이 평균적으로 또는 피크 시간 때 얼마나 오래 걸릴 것인가?
  - 트랜잭션 처리율: 1초당 얼마나 많은 트랜잭션들이 평균적으로 또는 피크 시간 때 처리될 수 있는가?
  - 전체 데이터베이스에 대한 보고서를 생성하는데 얼마나 오래 걸릴 것인가?

데이터베이스 설계의 개요

❑ 트랜잭션 설계

- ✓ 트랜잭션이란..
  - ✓데이터베이스의 상태를 변화 시키기 위해 수행하는 작업의 단위
- ✓ 요구사항 수집과 분석 후에 데이터베이스 설계 과정과 별도로 트랜잭션 설계를 진행할 수 있음
- ✓ 검색, 갱신, 혼합 등 세 가지 유형으로 구분하여 입력과 출력, 동작 등을 식별함

ER 모델

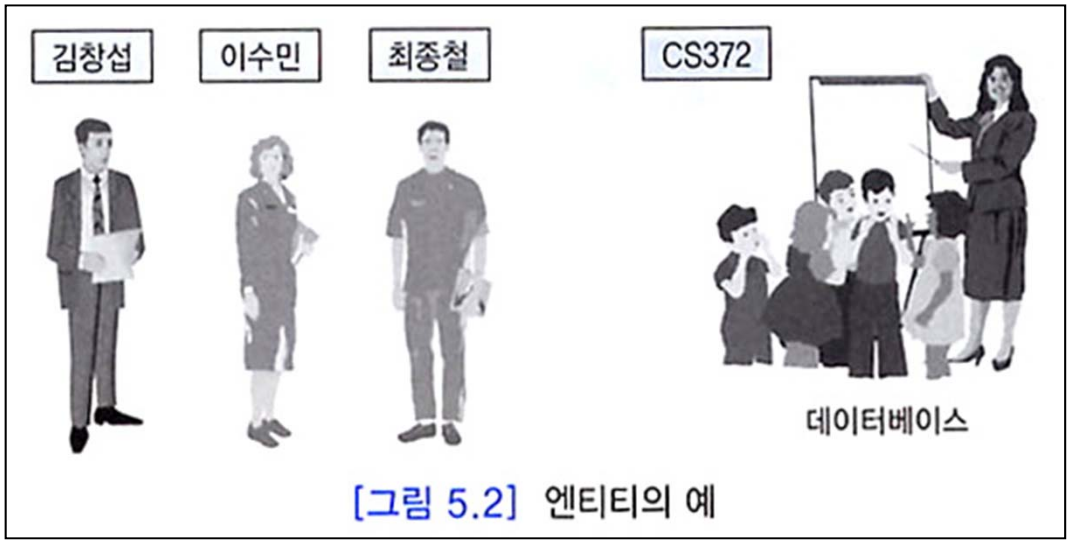
❑ ER 모델

- ✓ 개념적 설계를 위한 모델
- ✓ 실세계를 엔티티, 애트리뷰트, 엔티티들 간의 관계로 표현함
- ✓ 관계 데이터 모델로 사상됨
- ✓ 기본적인 구문 : 엔티티, 관계, 애트리뷰트
- ✓ 기타 구문 : 카디날리티 비율, 참여 제약조건
- ✓ 데이터베이스 설계를 위한 다소 구형 그래픽 표기법

ER 모델

❑ 엔티티

- ✓ 사람, 장소, 사물, 사건 등과 같이 독립적으로 존재하면서 고유하게 식별이 가능한 실세계의 객체

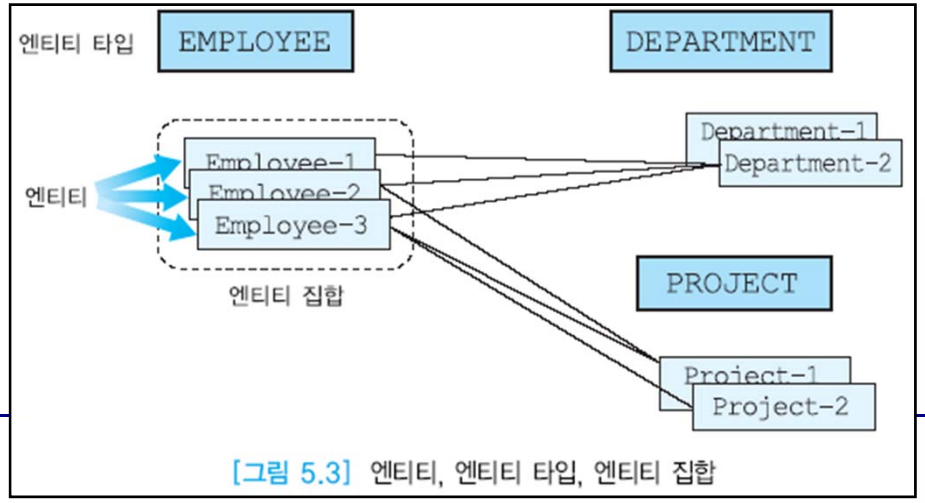


[그림 5.2] 엔티티의 예

ER 모델

❑ 엔티티 타입

- ✓ 엔티티들은 엔티티 타입(또는 엔티티 집합)들로 분류됨
- ✓ 엔티티 타입:
  - ✓동일한 애트리뷰트들을 가진 엔티티들의 틀
  - ✓관계 모델의 릴레이션의 내포(스키마, 구조)에 해당
- ✓ 엔티티 집합:
  - ✓동일한 애트리뷰트들을 가진 엔티티들의 모임
  - ✓관계 모델의 릴레이션의 외연(특정 시점의 내용)에 해당
- ✓ ER 다이어그램에서 **엔티티 타입은 직사각형으로 나타냄**



[그림 5.3] 엔티티, 엔티티 타입, 엔티티 집합

ER 모델

❑ 애트리뷰트

- ✓ 하나의 엔티티는 연관된 애트리뷰트들의 집합으로 설명됨
  - 예: 사원 엔티티는 사원번호, 이름, 직책, 급여 등의 애트리뷰트를 가짐
- ✓ 한 애트리뷰트의 도메인은 그 애트리뷰트가 가질 수 있는 모든 가능한 값들의 집합을 의미
  - 예: 사원번호는 1000부터 9999까지의 값을 가짐
- ✓ 여러 애트리뷰트가 동일한 도메인을 공유할 수 있음
  - 예: 사원번호와 부서번호가 네 자리 정수를 가질 수 있음
- ✓ 키 애트리뷰트는 한 애트리뷰트 또는 애트리뷰트들의 모임으로서 한 엔티티 타입 내에서 각 엔티티를 고유하게 식별함
- ✓ ER 다이어그램에서 **기본 키에 속하는 애트리뷰트는 밑줄을 그어 표시함**

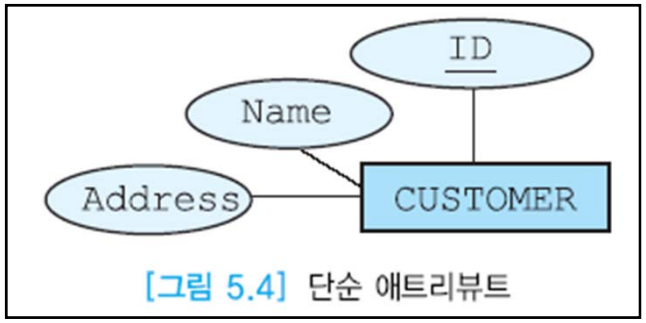
ER 모델

❑ 애트리뷰트(계속)

- ✓ ER 다이어그램에서 **타원형으로 나타냄**
- ✓ 애트리뷰트와 엔티티 타입은 실선으로 연결

❑ 단순 애트리뷰트 (simple attribute)

- ✓ 더 이상 다른 애트리뷰트로 나눌 수 없는 애트리뷰트
- ✓ ER 다이어그램에서 실선 타원으로 표현함
- ✓ ER 다이어그램에서 대부분의 애트리뷰트는 단순 애트리뷰트 연결

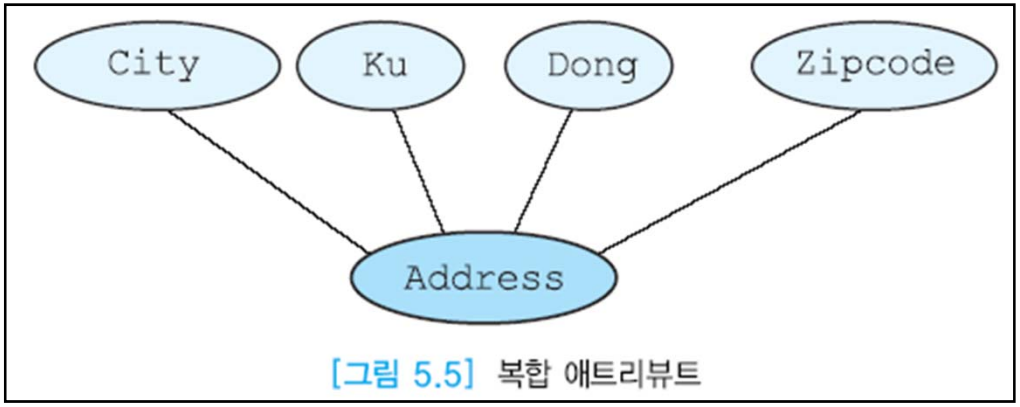


[그림 5.4] 단순 애트리뷰트

ER 모델

❑ 복합 애트리뷰트(composite attribute)

- ✓ 두 개 이상의 애트리뷰트로 이루어진 애트리뷰트
- ✓ 동일한 엔티티 타입이나 관계 타입에 속하는 애트리뷰트들 중에서 밀접하게 연관된 것을 모아놓은 것

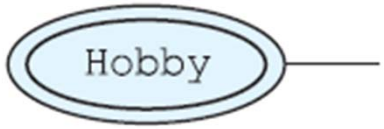


[그림 5.5] 복합 애트리뷰트

ER 모델

- **단일 값 애트리뷰트(single-valued attribute)**
  - ✓ 각 엔티티마다 정확하게 하나의 값을 갖는 애트리뷰트
  - ✓ ER 다이어그램에서 단순 애트리뷰트와 동일하게 표현됨
    - ✓예: 사원의 사원번호 애트리뷰트는 어떤 사원도 두 개 이상의 사원번호를 갖지 않으므로 단일 값 애트리뷰트
  - ✓ ER 다이어그램에서 대부분의 애트리뷰트는 단일 값 애트리뷰트

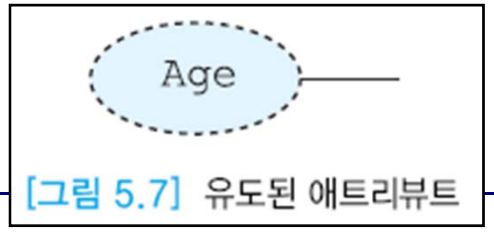
- **다치 애트리뷰트(multi-valued attribute)**
  - ✓ 각 엔티티마다 여러 개의 값을 가질 수 있는 애트리뷰트
  - ✓ ER 다이어그램에서 이중선 타원으로 표현함



[그림 5.6] 다치 애트리뷰트

ER 모델

- **저장된 애트리뷰트(stored attribute)**
  - ✓ 다른 애트리뷰트와 독립적으로 존재하는 애트리뷰트
  - ✓ ER 다이어그램에서 단순 애트리뷰트와 동일하게 표현됨
  - ✓ ER 다이어그램에서 대부분의 애트리뷰트는 저장된 애트리뷰트
    - ✓예: 사원 엔티티 타입에서 사원이름, 급여는 다른 애트리뷰트와 독립적으로 존재함
- **유도된 애트리뷰트(derived attribute)**
  - ✓ 다른 애트리뷰트의 값으로부터 얻어진 애트리뷰트
  - ✓ 관계 데이터베이스에서 릴레이션의 애트리뷰트로 포함시키지 않는 것이 좋음
  - ✓ ER 다이어그램에서 점선 타원으로 표현함

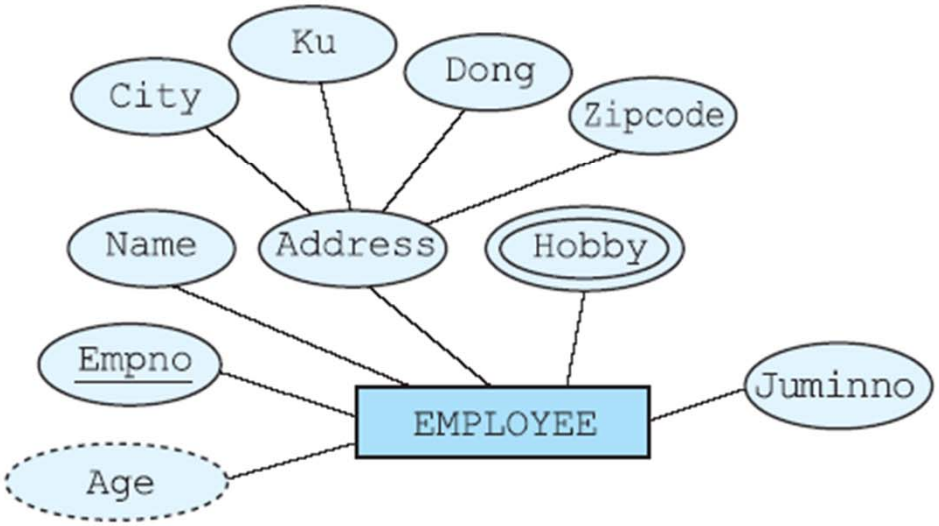


[그림 5.7] 유도된 애트리뷰트

ER 모델

예 : 애트리뷰트들의 유형

아래 그림 5.8에서 단순 애트리뷰트, 복합 애트리뷰트, 단일 값 애트리뷰트, 다치 애트리뷰트, 키 애트리뷰트, 저장된 애트리뷰트, 유도된 애트리뷰트들을 구분하라.

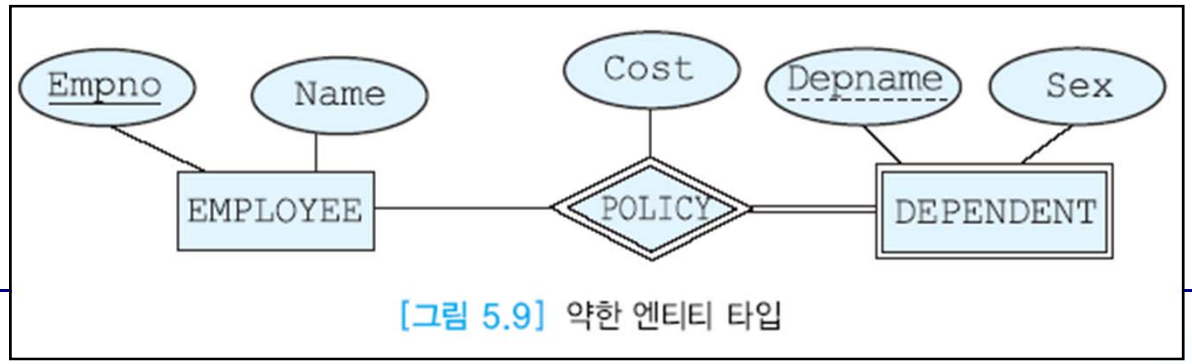


[그림 5.8] 여러 가지 애트리뷰트의 예

ER 모델

❑ 약한 엔티티 타입

- ✓ 키를 형성하기에 충분한 애트리뷰트들을 갖지 못한 엔티티 타입
- ✓ 약한 엔티티 타입에게 키 애트리뷰트를 제공하는 엔티티 타입을 소유 엔티티 타입(owner entity type) 또는 식별 엔티티 타입(identifying entity type)라고 부름
- ✓ ER 다이어그램에서 이중선 직사각형으로 표기
- ✓ 약한 엔티티 타입의 부분 키는 점선 밑줄을 그어 표시
- ✓ 부분 키(partial key): 부양가족의 이름처럼 한 사원에 속한 부양가족 내에서는 서로 다르지만 회사 전체 사원들의 부양가족들 전체에서는 같은 경우가 생길 수 있는 애트리뷰트



[그림 5.9] 약한 엔티티 타입

ER 모델

❑ 관계와 관계 타입

- ✓ 관계는 엔티티들 사이에 존재하는 연관이나 연결로서 **두 개 이상의 엔티티 타입들 사이의 사상(mapping)으로 생각할 수 있음**
- ✓ 요구사항 명세에서 흔히 동사는 ER 다이어그램에서 관계로 표현됨
- ✓ ER 다이어그램에서 다이어몬드로 표기
- ✓ 관계 타입이 서로 연관시키는 엔티티 타입들을 관계 타입에 실선으로 연결함

[그림 5.10] 관계 타입 WORKS_FOR

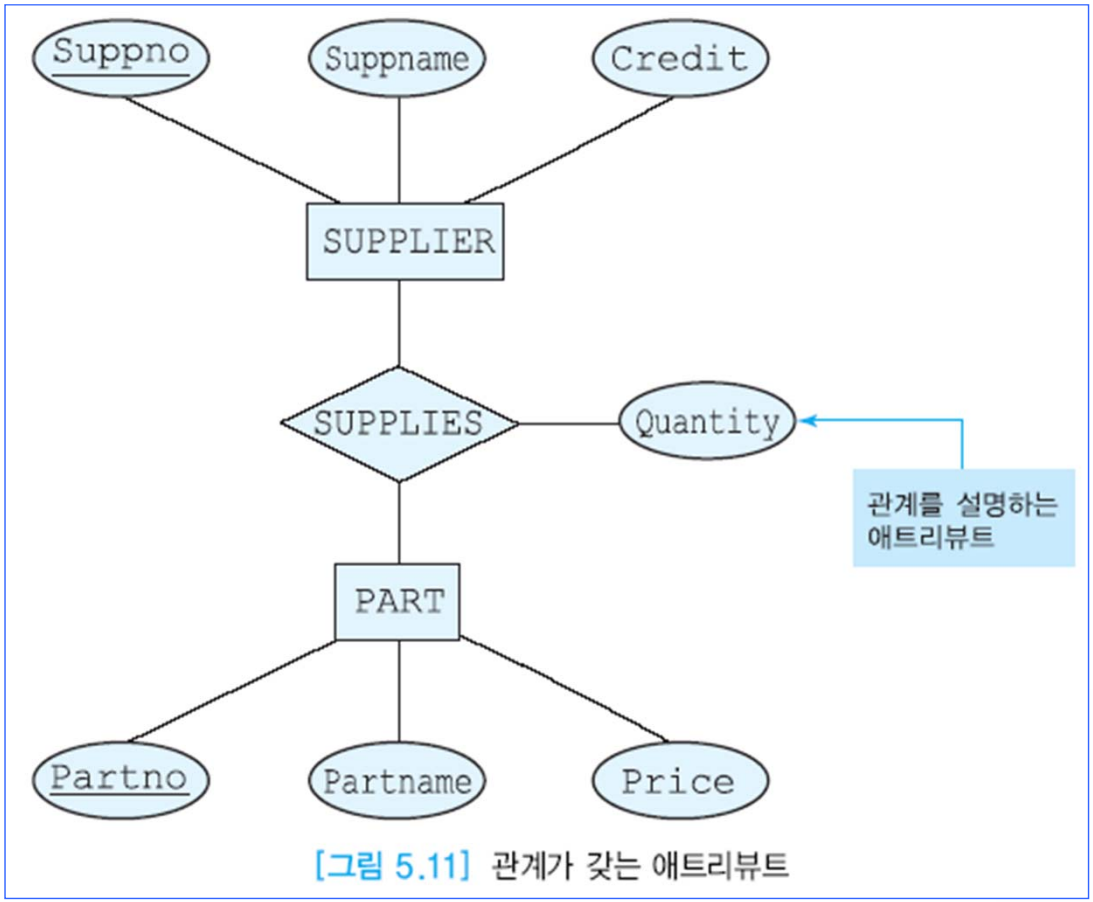
〈표 5.2〉 엔티티와 엔티티 간의 관계의 예

엔티티	관계	엔티티
사원(employee)	근무한다(works for)	부서(department)
공급자(supplier)	공급한다(supplies)	부품(part)
학생(student)	수강한다(enrolls)	과목(course)

ER 모델

❑ 관계의 애트리뷰트

- ✓ 관계 타입은 관계의 특징을 기술하는 애트리뷰트들을 가질 수 있음
- ✓ 관계 타입은 키 애트리뷰트를 갖지 않음

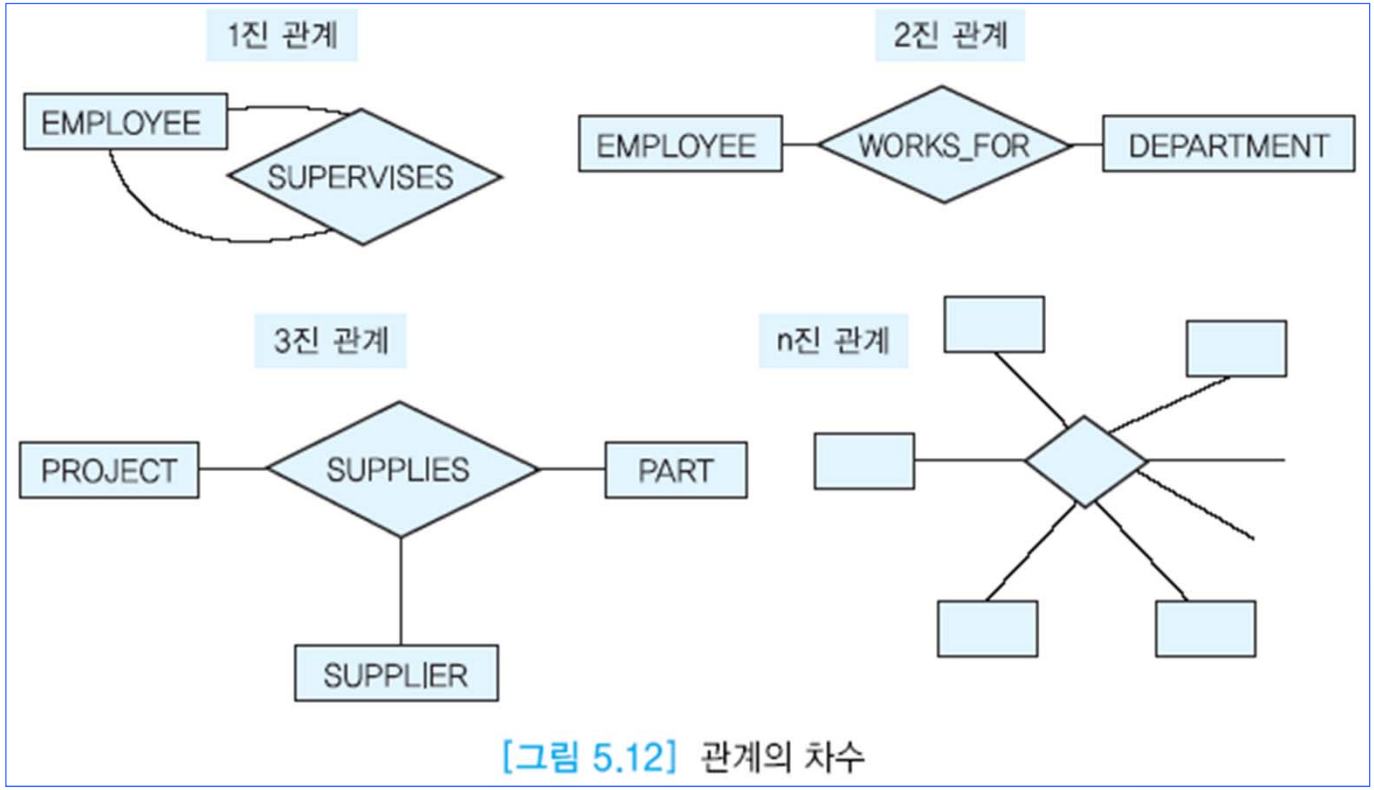


[그림 5.11] 관계가 갖는 애트리뷰트

ER 모델

❑ 차수(degree)

- ✓ 관계로 연결된 엔티티 타입들의 개수를 의미
- ✓ 실세계에서 가장 흔한 관계는 두 개의 엔티티 타입을 연결하는 2진 관계

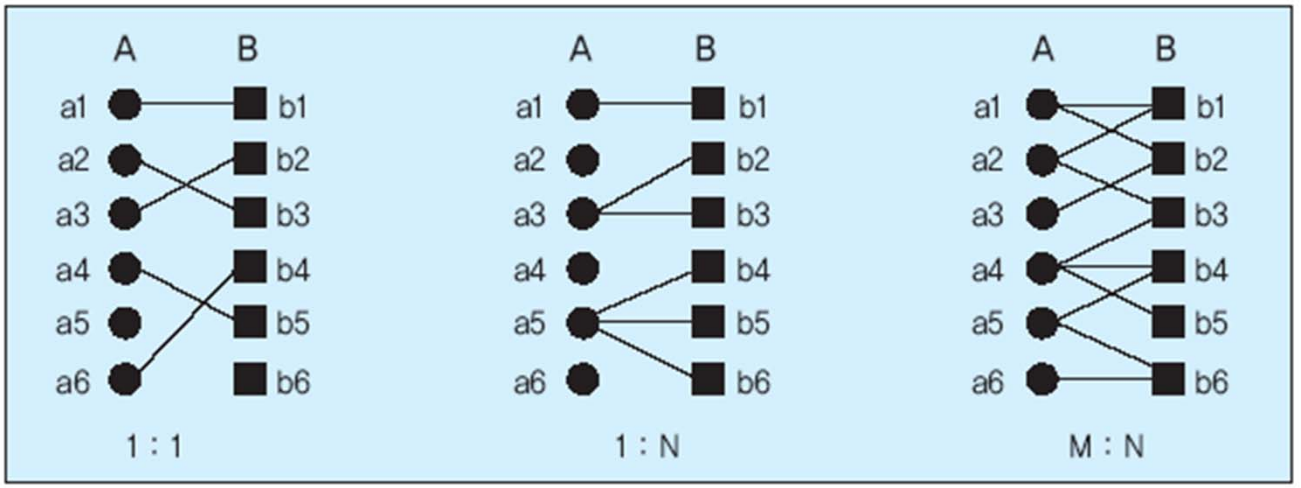


[그림 5.12] 관계의 차수

ER 모델

❑ 카디날리티

- ✓ 카디날리티 비율은 한 엔티티가 참여할 수 있는 관계의 수를 나타냄
- ✓ 관계 타입에 참여하는 엔티티들의 가능한 조합을 제한함
- ✓ 관계를 흔히 1:1, 1:N, M:N으로 구분
- ✓ 카디날리티에 관한 정보는 간선 위에 나타냄



[그림 5.14] 카디날리티 비율

ER 모델

❑ 1:1 관계

- ✓ E1의 각 엔티티가 정확하게 E2의 한 엔티티와 연관되고, E2의 각 엔티티가 정확하게 E1의 한 엔티티와 연관되면 이 관계를 1:1 관계라고 함
  - ✓ 예: 각 사원에 대해 최대한 한 개의 PC가 있고, 각 PC에 대해 최대한 한 명의 사원이 있으면 사원과 PC 간의 관계는 1:1 관계

❑ 1:N 관계

- ✓ E1의 각 엔티티가 E2의 임의의 개수의 엔티티와 연관되고, E2의 각 엔티티는 정확하게 E1의 한 엔티티와 연관되면 이 관계를 1:N 관계라고 함
  - ✓ 예: 각 사원에 대해 최대한 한 대의 PC가 있고, 각 PC에 대해 여러 명의 사원들이 있으면 PC와 사원 간의 관계는 1:N 관계
- ✓ 실세계에서 가장 흔히 나타나는 관계

ER 모델

❑ M:N 관계

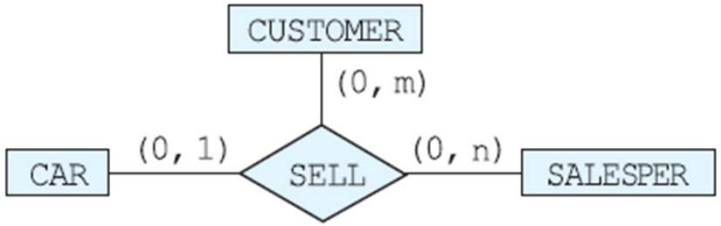
- ✓ 한 엔티티 타입에 속하는 임의의 개수의 엔티티가 다른 엔티티 타입에 속하는 임의의 개수의 엔티티와 연관됨
  - ✓예: 각 사원에 대해 여러 대의 PC가 있고, 각 PC에 대해 여러 명의 사원들이 있으면 사원과 PC 간의 관계는 M:N 관계

ER 모델

❑ 카디날리티 비율의 최소값과 최대값

- ✓ ER 다이어그램에서 관계 타입과 엔티티 타입을 연결하는 실선 위에 (min, max) 형태로 표기
  - ✓어떤 관계 타입에 참여하는 각 엔티티 타입에 대하여 min은 이 엔티티 타입 내의 각 엔티티는 적어도 min 번 관계에 참여함을 의미
  - ✓max는 이 엔티티 타입 내의 각 엔티티는 최대한 max 번 관계에 참여함을 의미
  - ✓min=0은 어떤 엔티티가 반드시 관계에 참여해야 할 필요는 없음을 의미
  - ✓max=*는 어떤 엔티티가 관계에 임의의 수만큼 참여할 수 있음을 의미

[그림 5.16] 카디날리티의 최소값과 최대값



[그림 5.17] 카디날리티가 명시된 3진 관계 타입

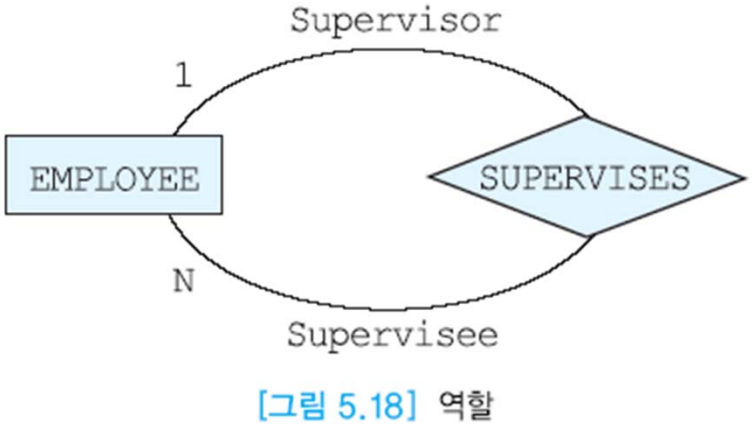
〈표 5.3〉 카디날리티들의 몇 가지 유형

관계	(min1, max1)	(min2, max2)	그래픽 표기
1 : 1	(0, 1)	(0, 1)	1 — 1
1 : N	(0, *)	(0, 1)	1 — N
M : N	(0, *)	(0, *)	M — N

ER 모델

❑ 역할(role)

- ✓ 관계 타입의 의미를 명확하게 하기 위해 사용됨
- ✓ 특히 하나의 관계 타입에 하나의 엔티티 타입이 여러 번 나타나는 경우에는 반드시 역할을 표기해야 함
- ✓ 관계 타입의 간선 위에 표시



[그림 5.18] 역할

ER 모델

❑ 전체 참여와 부분 참여

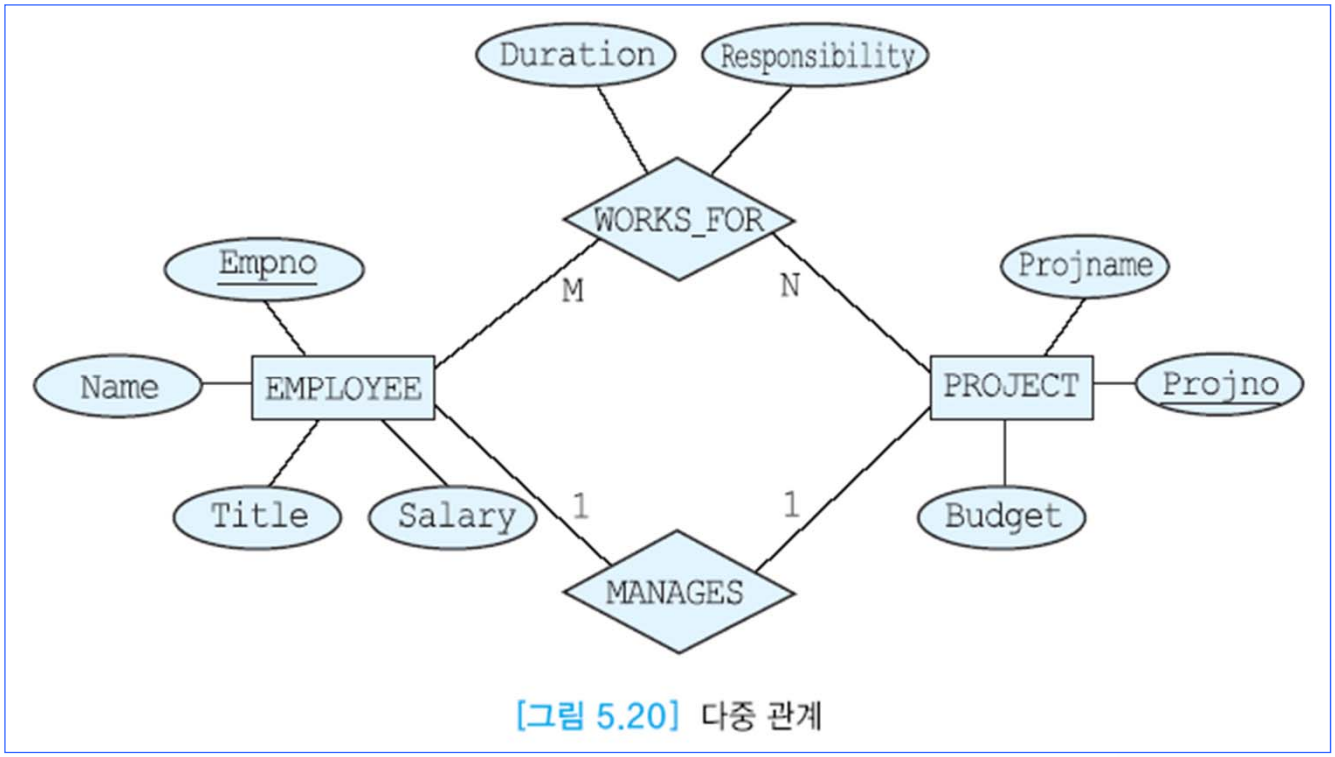
- ✓ 전체 참여는 어떤 관계에 엔티티 타입 E1의 모든 엔티티들이 관계 타입 R에 의해서 어떤 엔티티 타입 E2의 어떤 엔티티와 연관되는 것을 의미
- ✓ 부분 참여는 어떤 관계에 엔티티 타입 E1의 일부 엔티티만 참여하는 것을 의미
- ✓ 약한 엔티티 타입은 항상 관계에 전체 참여
- ✓ 전체 참여는 ER 다이어그램에서 이중 실선으로 표시
- ✓ 카디날리티 비율과 함께 참여 제약조건은 관계에 대한 중요한 제약조건

[그림 5.19] 전체 참여와 부분 참여

ER 모델

❑ 다중 관계

✓ 두 엔티티 타입 사이에 두 개 이상의 관계 타입이 존재할 수 있음

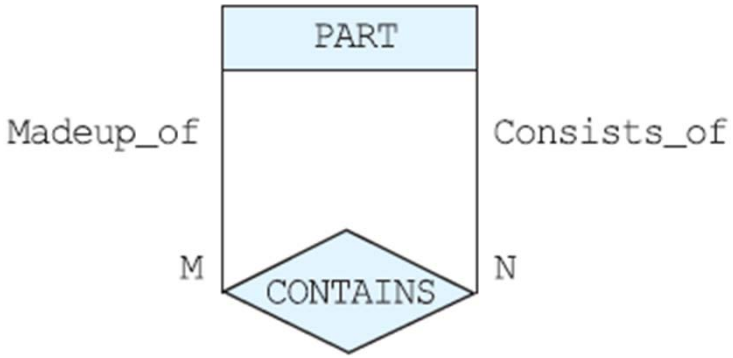


[그림 5.20] 다중 관계

ER 모델

❑ 순환적 관계

✓ 하나의 엔티티 타입이 동일한 관계 타입에 두 번 이상 참여하는 것



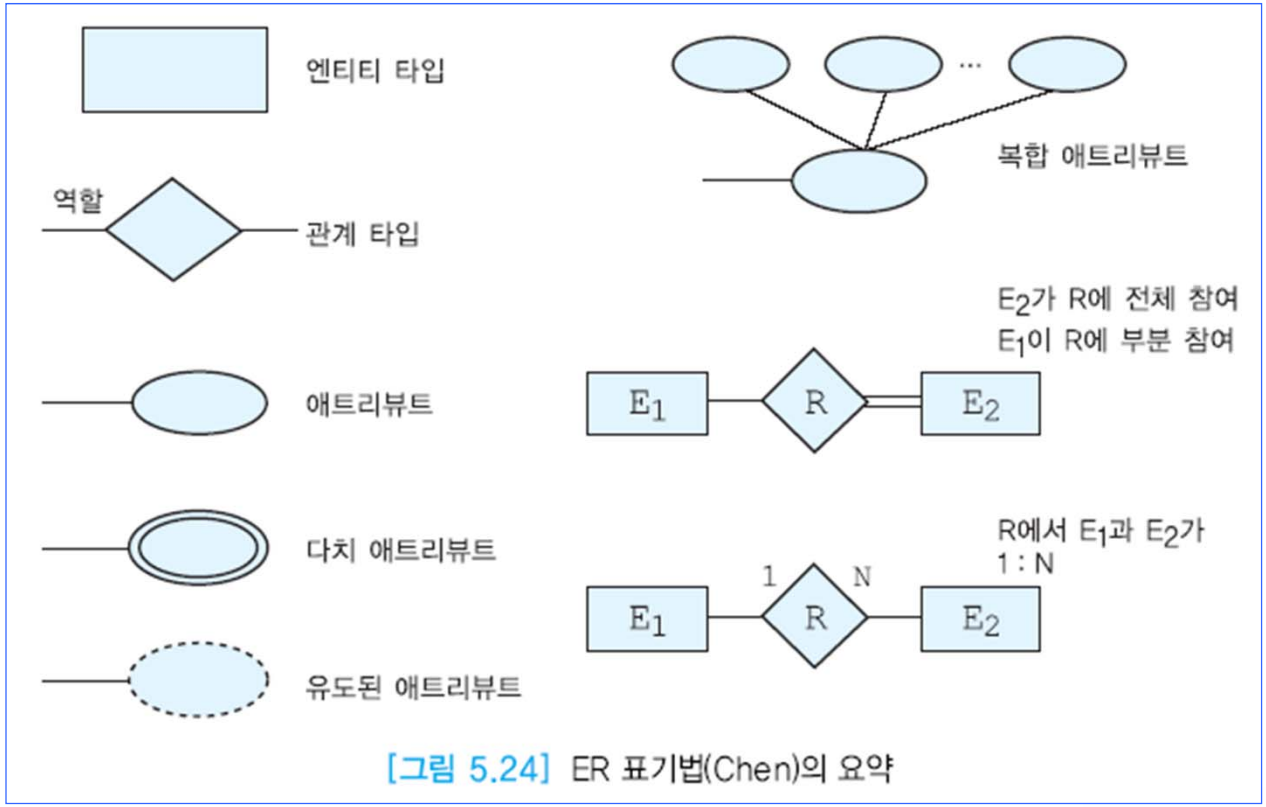
[그림 5.21] 순환적 관계

ER 모델

- 데이터베이스 설계 과정
  - ✓ 응용의 **요구사항을 수집**하여 기술
  - ✓ 응용과 **연관이 있는 엔티티 타입들을 식별**
  - ✓ 응용과 연관이 있는 **관계 타입들을 식별**
  - ✓ 관계가 1:1, 1:N, M:N 중에서 어느 것에 해당하는지 결정
  - ✓ 엔티티 타입과 관계 타입들에 **필요한 애트리뷰트들을 식별**하고, 각 애트리뷰트가 가질 수 있는 **값들의 집합을 식별**
  - ✓ 엔티티 타입들을 위한 **기본 키를 식별**
  - ✓ 응용을 위한 **ER 스키마 다이어그램을 그림**
  - ✓ ER 스키마 다이어그램이 응용에 대한 **요구사항과 부합되는지 검사**
  - ✓ ER 스키마 다이어그램을 DBMS에서 사용되는 **데이터베이스 모델로 변환**

ER 모델

❑ 본 책의 **ER** 표기법의 요약

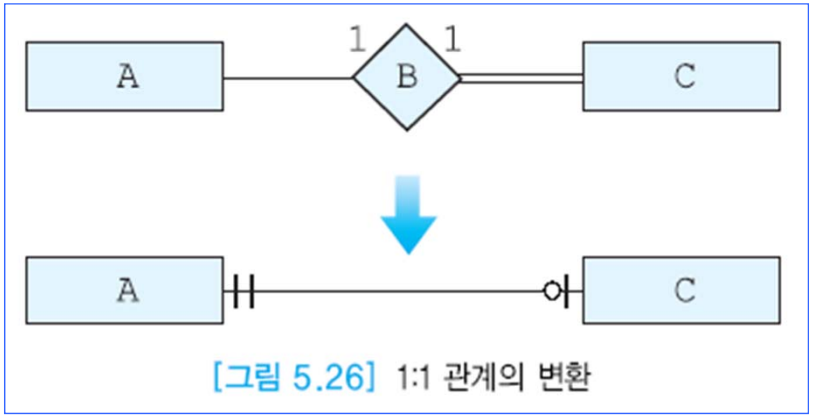


[그림 5.24] ER 표기법(Chen)의 요약

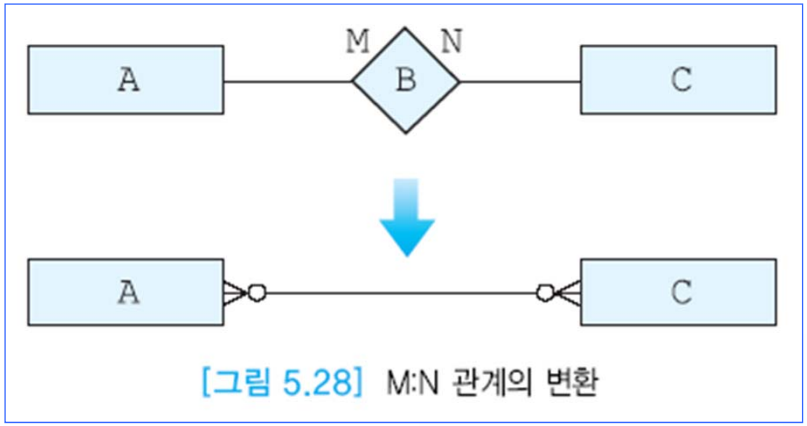
ER 모델

- **ER** 모델의 또 다른 표기법
  - ✓ 새발**(crow-feet)** 표기법이 흔히 사용됨
  - ✓ 새발 표기법에도 여러 가지 변형들이 존재함

- 본 책의 표기법을 새발 표기법으로 표현하는 방법
  - ✓ **1:1** 관계, **M:N**관계



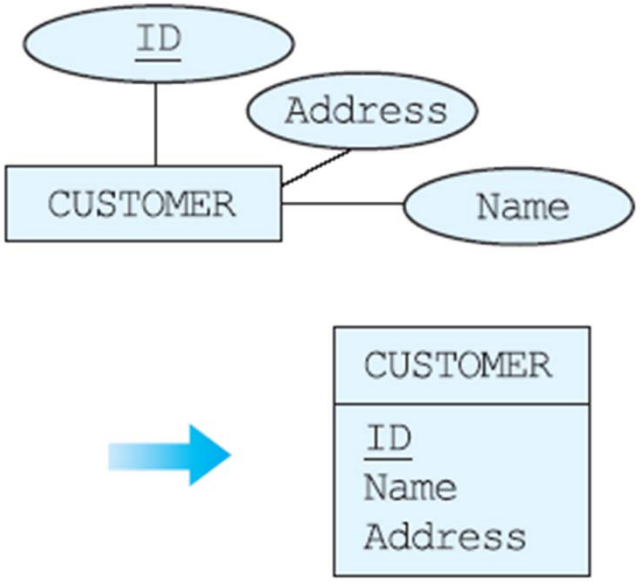
[그림 5.26] 1:1 관계의 변환



[그림 5.28] M:N 관계의 변환

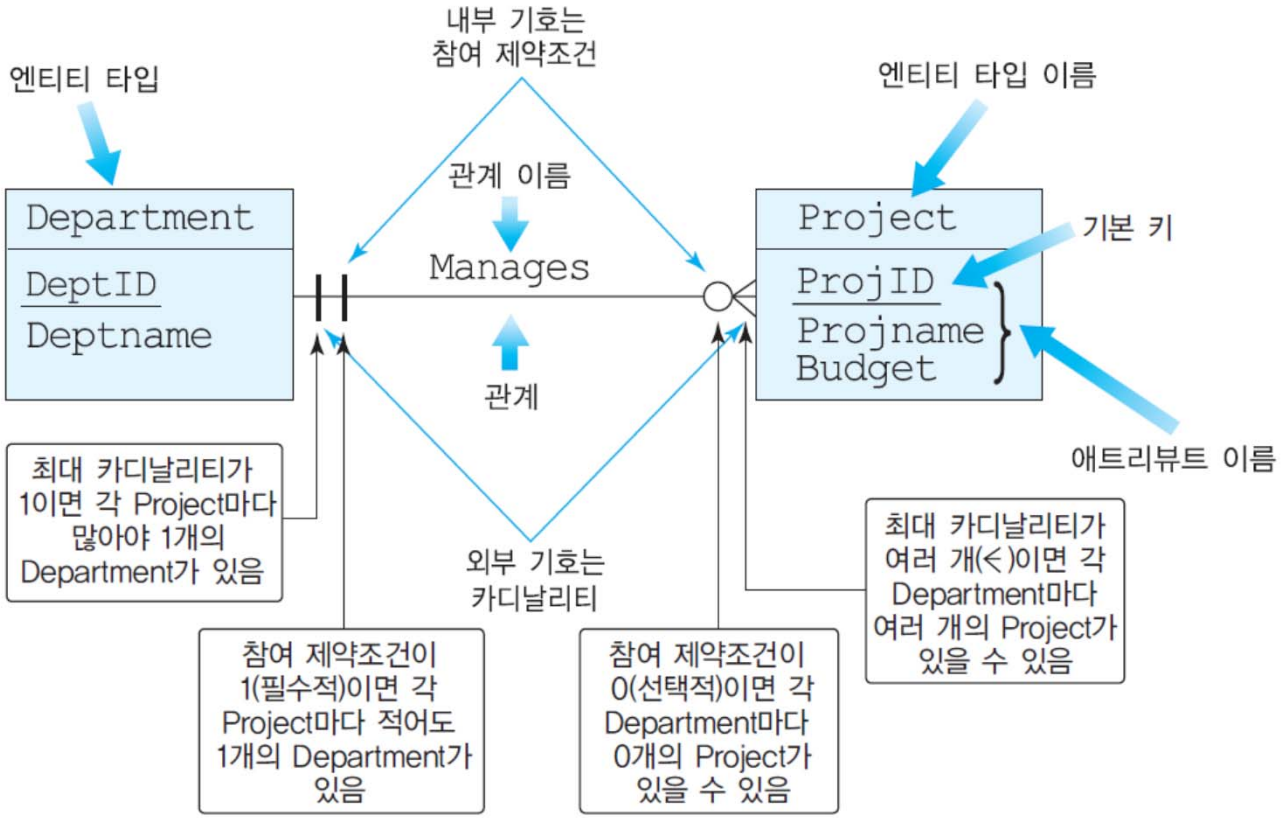
ER 모델

✓ 엔티티 타입과 애트리뷰트



[그림 5.29] 새발 표기법에서 엔티티 타입과 애트리뷰트의 표현

ER 모델



[그림 5.30] 새발 표기법의 요약

데이터베이스 설계 사례

- 기업에서 흔히 볼 수 있는 작은 세계에 관한 요구사항
  - ✓ 회사에는 다수의 **사원들이 재직**
  - ✓ 각 사원에 대해서 **사원번호(고유함), 이름, 직책, 급여, 주소**를 저장.
    - ✓ **주소는 시, 구, 동으로 세분**하여 나타냄
  - ✓ 각 사원은 **0명 이상의 부양가족**을 가질 수 있음.
    - ✓ 한 부양가족은 두 명 이상의 사원에게 속하지 않음.
    - ✓ 각 부양가족에 대해서 **부양가족의 이름과 성별을 저장**
  - ✓ 회사의 프로젝트에 대해서 **프로젝트번호(고유함), 이름, 예산, 프로젝트가 진행되는 위치**를 나타냄.
    - ✓ 한 프로젝트는 여러 위치에서 진행될 수 있음.
    - ✓ 각 프로젝트마다 여러 명의 사원들이 일함.
    - ✓ 각 사원은 여러 프로젝트에서 근무할 수 있음.
    - ✓ 각 사원이 해당 프로젝트에서 어떤 역할을 수행하고, 얼마 동안 근무해 왔는가를 나타냄.
    - ✓ 각 프로젝트마다 한 명의 프로젝트 관리자가 있음.
    - ✓ 한 사원은 두 개 이상의 프로젝트의 관리자가 될 수는 없음.
    - ✓ 프로젝트 관리자 임무를 시작한 날짜를 기록

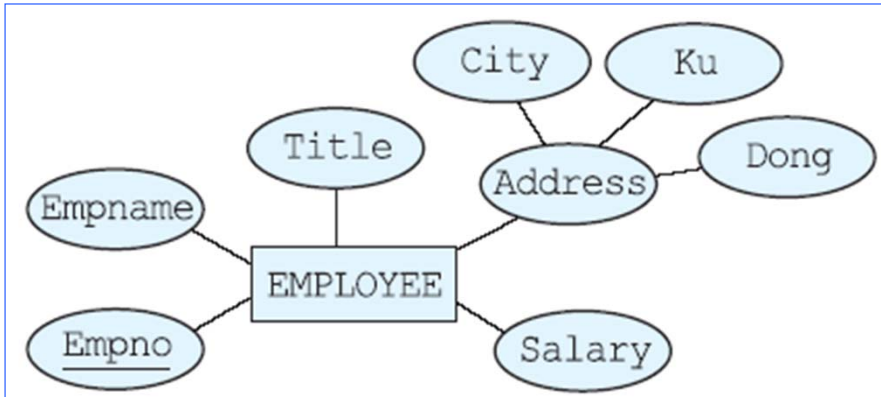
데이터베이스 설계 사례

❑ 기업에서 흔히 볼 수 있는 작은 세계에 관한 요구사항

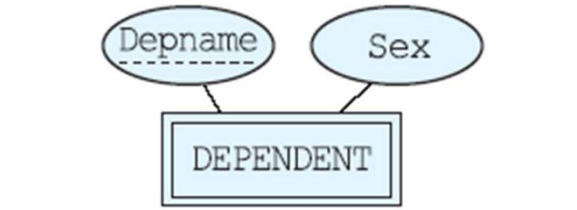
- ✓ 각 사원은 한 부서에만 속함.
  - ✓ 각 **부서**에 대해서 **부서번호(고유함), 이름, 부서가 위치한 층**을 나타냄
- ✓ 각 프로젝트에는 부품들이 필요.
  - ✓ 한 부품이 두 개 이상의 프로젝트에서 사용될 수 있음.
  - ✓ 하나의 부품은 다른 여러 개의 부품들로 이루어질 수 있음.
  - ✓ 각 부품에 대해서 **부품번호(고유함), 이름, 가격**, 그 부품이 다른 부품들을 포함하는 경우에는 그 부품들에 관한 정보도 나타냄
- ✓ 각 부품을 공급하는 공급자들이 있음.
  - ✓ 한 명의 공급자는 여러 가지 부품들을 공급할 수 있고, 각 부품은 여러 공급자들로부터 공급될 수 있음.
  - ✓ 각 공급자에 대해서 공급자번호(고유함), 이름, 신용도를 나타냄.
  - ✓ 각 공급자에 대해서 그 공급자가 어떤 부품을 어떤 프로젝트에 얼마나 공급하는가를 나타냄

데이터베이스 설계 사례

- 복잡한 요구사항을 바탕으로부터 **ER**모델 작성
- 기업에서 흔히 볼 수 있는 작은 세계에 관한 요구사항
  - ✓ 회사에는 다수의 **사원들이 재직**
  - ✓ 각 사원에 대해서 **사원번호(고유함), 이름, 직책, 급여, 주소**를 저장.
    - ✓**주소는 시, 구, 동으로 세분**하여 나타냄
  - ✓ 각 사원은 **0명 이상의 부양가족**을 가질 수 있음.
    - ✓한 부양가족은 두 명 이상의 사원에게 속하지 않음.
    - ✓각 부양가족에 대해서 **부양가족의 이름과 성별을 저장**



[그림 5.31] EMPLOYEE 엔티티 타입



[그림 5.35] DEPENDENT 엔티티 타입

데이터베이스 설계 사례

❑ 기업에서 흔히 볼 수 있는 작은 세계에 관한 요구사항

✓ 회사의 프로젝트에 대해서 **프로젝트번호(고유함), 이름, 예산, 프로젝트가 진행되는 위치**를 나타냄.

✓한 프로젝트는 여러 위치에서 진행될 수 있음.

✓각 프로젝트마다 여러 명의 사원들이 일함.

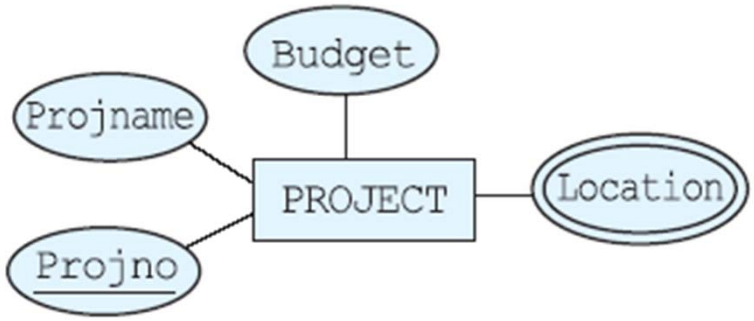
✓각 사원은 여러 프로젝트에서 근무할 수 있음.

✓각 사원이 해당 프로젝트에서 어떤 역할을 수행하고, 얼마 동안 근무해 왔는가를 나타냄.

✓각 프로젝트마다 한 명의 프로젝트 관리자가 있음.

✓한 사원은 두 개 이상의 프로젝트의 관리자가 될 수는 없음.

✓프로젝트 관리자 임무를 시작한 날짜를 기록

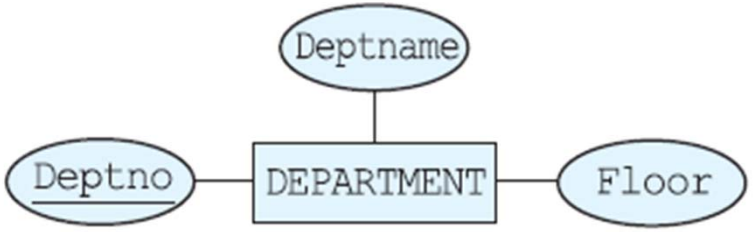


[그림 5.32] PROJECT 엔티티 타입

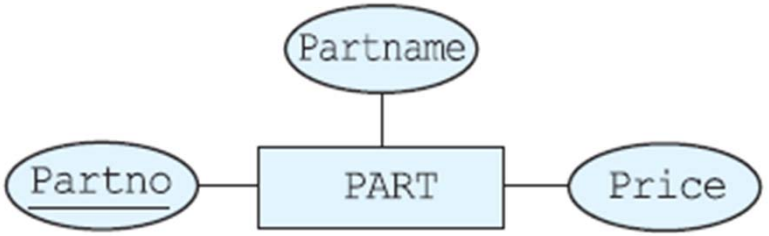
데이터베이스 설계 사례

❑ 기업에서 흔히 볼 수 있는 작은 세계에 관한 요구사항(계속)

- ✓ 각 사원은 한 부서에만 속함.
  - ✓각 **부서**에 대해서 **부서번호(고유함), 이름, 부서가 위치한 층**을 나타냄
- ✓ 각 프로젝트에는 부품들이 필요.
  - ✓한 부품이 두 개 이상의 프로젝트에서 사용될 수 있음.
  - ✓하나의 부품은 다른 여러 개의 부품들로 이루어질 수 있음.
  - ✓각 부품에 대해서 **부품번호(고유함), 이름, 가격**, 그 부품이 다른 부품들을 포함하는 경우에는 그 부품들에 관한 정보도 나타냄



[그림 5.33] DEPARTMENT 엔티티 타입

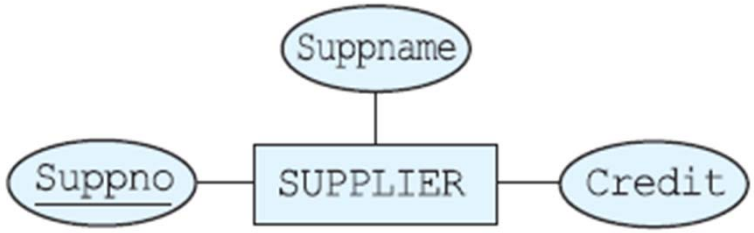


[그림 5.36] PART 엔티티 타입

데이터베이스 설계 사례

❑ 기업에서 흔히 볼 수 있는 작은 세계에 관한 요구사항(계속)

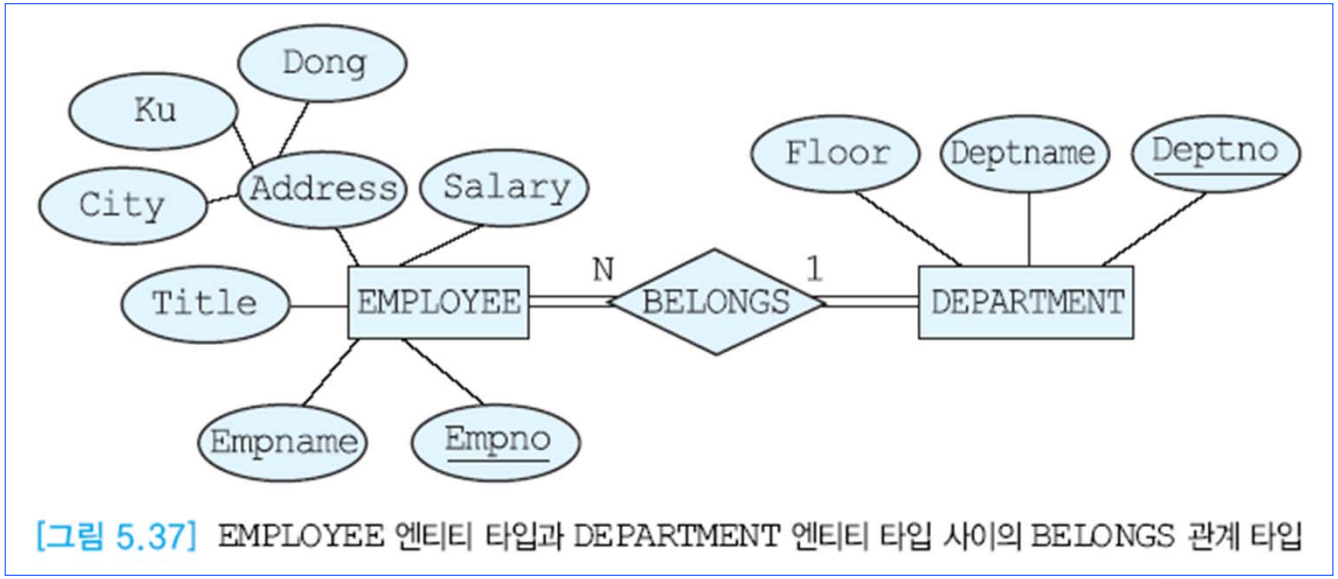
- ✓ 각 부품을 공급하는 공급자들이 있음.
  - ✓한 명의 공급자는 여러 가지 부품들을 공급할 수 있고, 각 부품은 여러 공급자들로부터 공급될 수 있음.
  - ✓각 공급자에 대해서 공급자번호(고유함), 이름, 신용도를 나타냄.
  - ✓각 공급자에 대해서 그 공급자가 어떤 부품을 어떤 프로젝트에 얼마나 공급하는가를 나타냄



[그림 5.34] SUPPLIER 엔티티 타입

데이터베이스 설계 사례

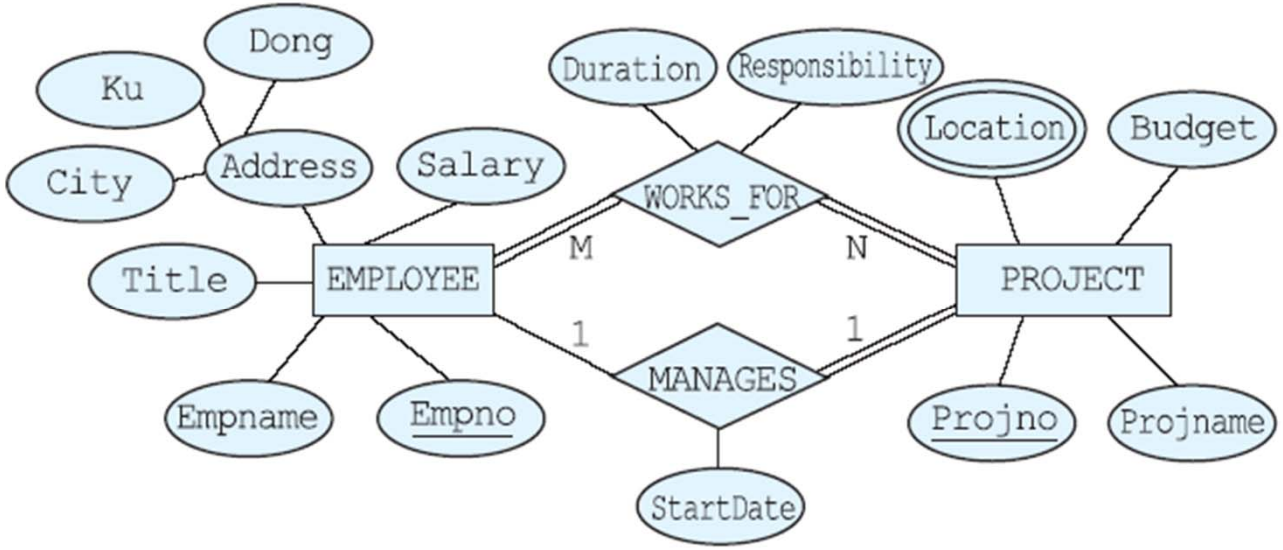
- 관계와 애트리뷰트들을 식별
  - 사원에 대한 요구 사항에서
    - 각 사원은 한 부서에만 속함



[그림 5.37] EMPLOYEE 엔티티 타입과 DEPARTMENT 엔티티 타입 사이의 BELONGS 관계 타입

데이터베이스 설계 사례

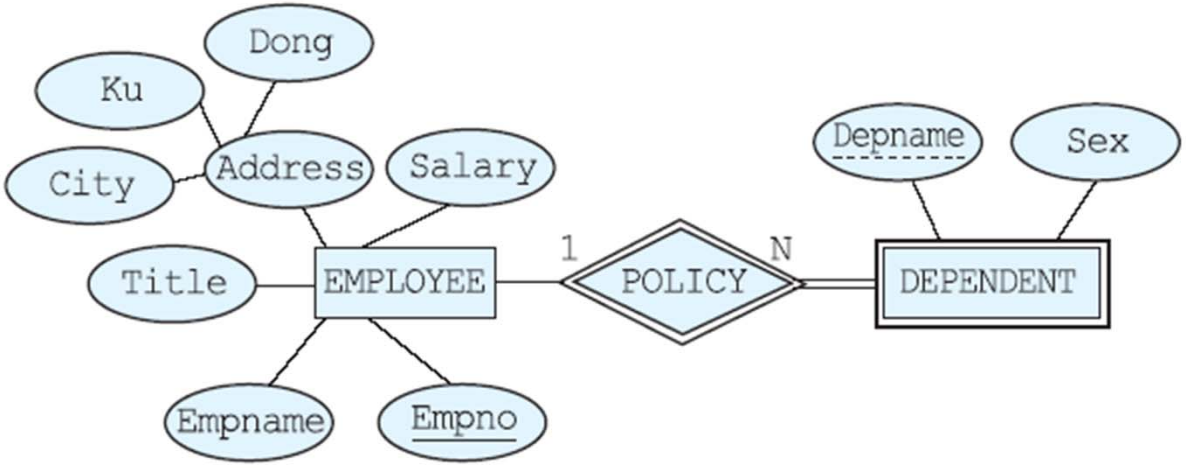
- 관계와 애트리뷰트들을 식별
  - 프로젝트와 사원에 대한 요구에서
    - ✓각 프로젝트마다 여러 명의 사원들이 일함.
    - ✓각 사원은 여러 프로젝트에서 근무할 수 있음.
      - ✓각 사원이 해당 프로젝트에서 어떤 역할을 수행하고, 얼마 동안 근무해 왔는가를 나타냄.
    - ✓각 프로젝트마다 한 명의 프로젝트 관리자가 있음.
    - ✓한 사원은 두 개 이상의 프로젝트의 관리자가 될 수는 없음.
      - ✓프로젝트 관리자 임무를 시작한 날짜를 기록



[그림 5.38] EMPLOYEE 엔티티 타입과 PROJECT 엔티티 타입 사이의 두 개의 관계 타입

데이터베이스 설계 사례

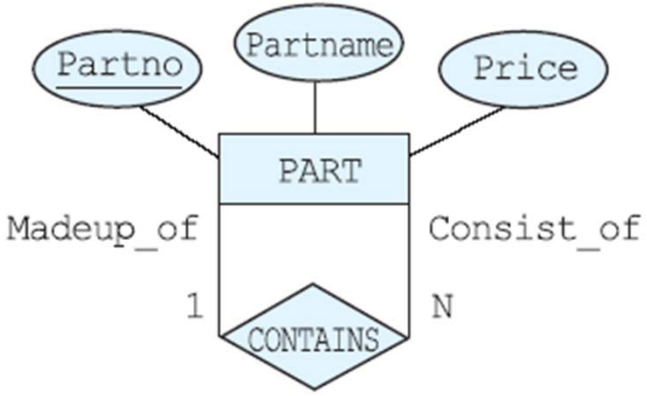
- 관계와 애트리뷰트들을 식별
  - 사원과 부양가족에 대한 요구에서
    - ✓ 사원의 부양가족은 회사의 의료보험 혜택을 받음
    - ✓ 각 사원은 **0명 이상의 부양가족**을 가질 수 있음.
      - ✓한 부양가족은 두 명 이상의 사원에게 속하지 않음.
      - ✓각 부양가족에 대해서 **부양가족의 이름과 성별을 저장**



[그림 5.39] EMPLOYEE 엔티티 타입과 DEPENDENT 엔티티 타입 사이의 약한 관계 타입

데이터베이스 설계 사례

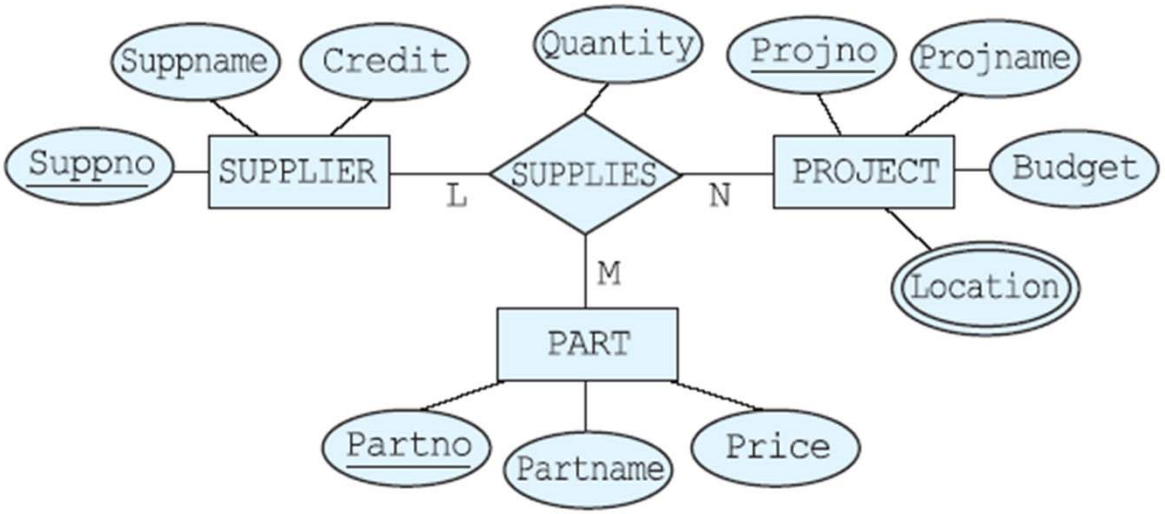
- 관계와 애트리뷰트들을 식별
  - 부품에 대한 요구에서
    - ✓하나의 부품은 다른 여러 개의 부품들로 이루어질 수 있음.



[그림 5.40] PART 엔티티 타입이 두 번 참여하는 순환적 관계 타입

데이터베이스 설계 사례

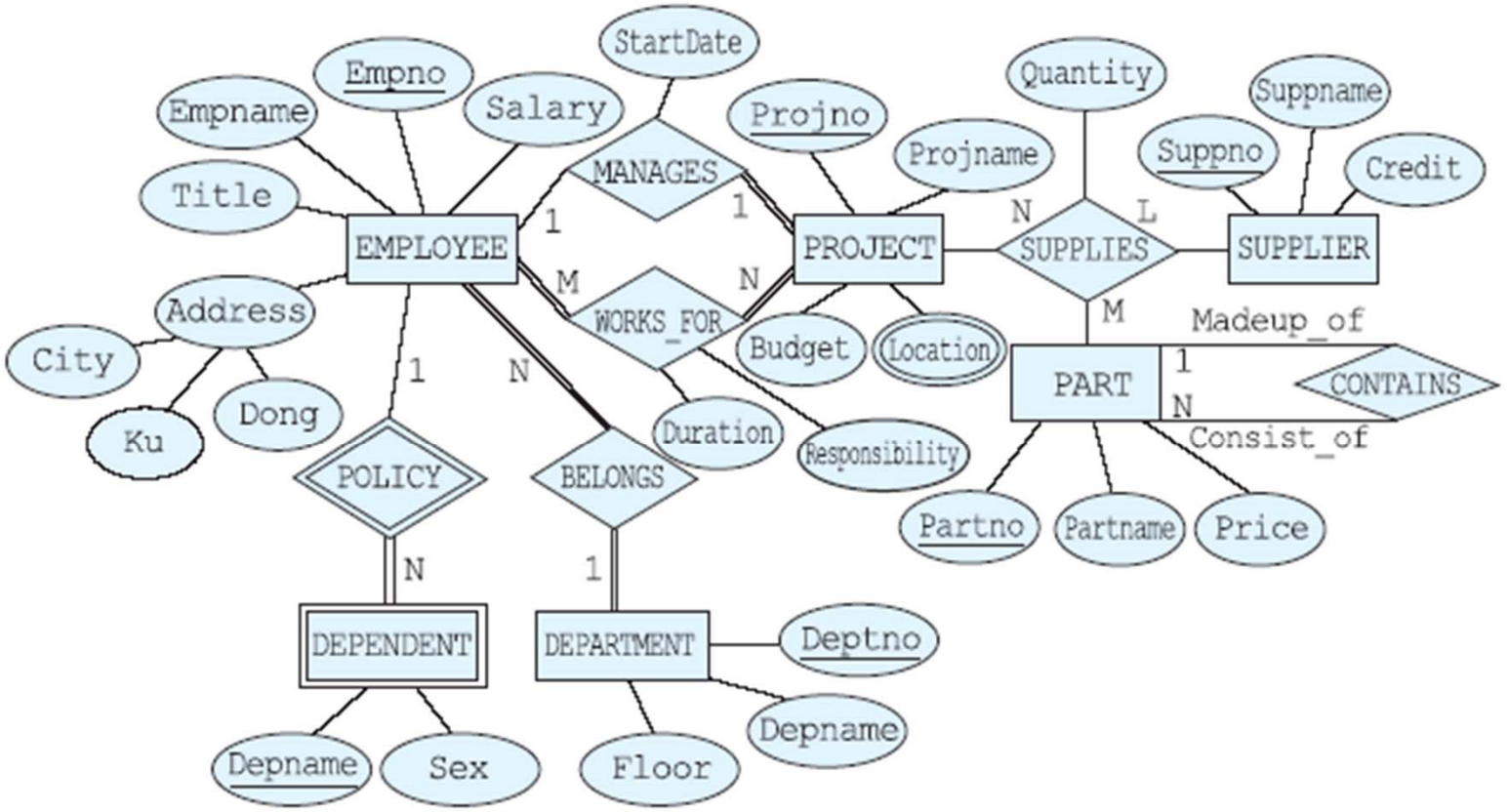
- 관계와 애트리뷰트들을 식별
  - 공급자, 부품, 프로젝트에 대한 요구에서
    - ✓한 명의 공급자는 여러 가지 부품들을 공급할 수 있고, 각 부품은 여러 공급자들로부터 공급될 수 있음.
    - ✓한 부품이 두 개 이상의 프로젝트에서 사용될 수 있음.



[그림 5.41] 세 개의 엔티티 타입이 참여하는 3진 관계 타입

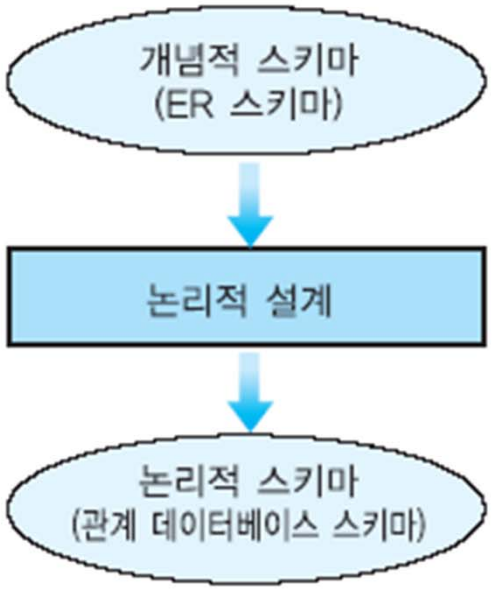
데이터베이스 설계 사례

❑ 소개된 모든 **ER**모델의 합



[그림 5.42] 회사의 ER 스키마 다이어그램

ER 스키마를 관계 모델의 릴레이션으로 사상



- 관계 DBMS에서는 ER 스키마를 릴레이션들로 사상

[그림 5.43] 논리적 설계

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 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

ER 스키마를 관계 모델의 릴레이션으로 사상

〈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

ER 스키마를 관계 모델의 릴레이션으로 사상

❑ 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 애트리뷰트들**을 릴레이션 **R에 모두 포함시킴**

✓E에서 복합 애트리뷰트는 그 복합 애트리뷰트를 구성하는 단순 애트리뷰트들만 릴레이션 R에 포함시킴

✓E의 **기본 키가 릴레이션 R의 기본 키가 됨**

[그림 5.44] 정규 엔티티 타입을 릴레이션으로 사상

ER 스키마를 관계 모델의 릴레이션으로 사상

- 데이터베이스 설계 사례에 알고리즘 적용

단계 1: 정규 엔티티 타입과 단일 값 애트리뷰트

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

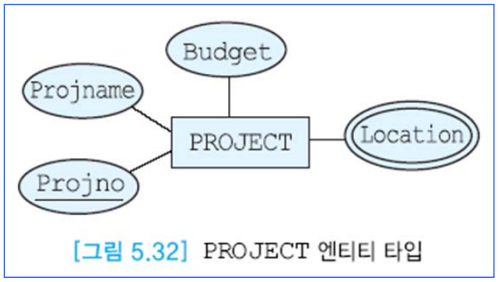
PROJECT(Projno, Projname, Budget)

DEPARTMENT(Deptno, Deptname, Floor)

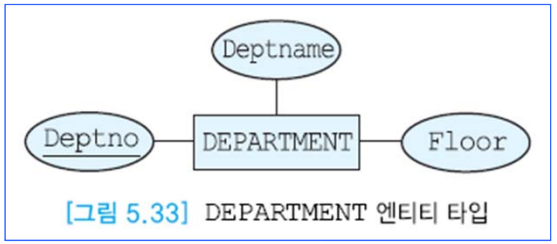
SUPPLIER(Suppno, Suppname, Credit)

PART(Partno, Partname, Price)

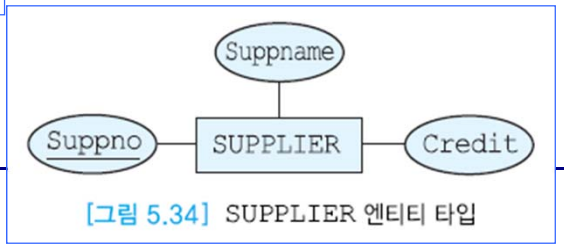
[그림 5.31] EMPLOYEE 엔티티 타입



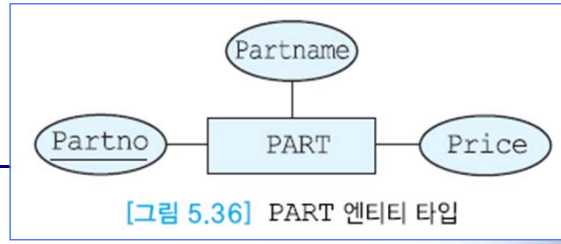
[그림 5.32] PROJECT 엔티티 타입



[그림 5.33] DEPARTMENT 엔티티 타입



[그림 5.34] SUPPLIER 엔티티 타입



[그림 5.36] PART 엔티티 타입

ER 스키마를 관계 모델의 릴레이션으로 사상

❑ **ER-관계 사상 알고리즘(계속)**

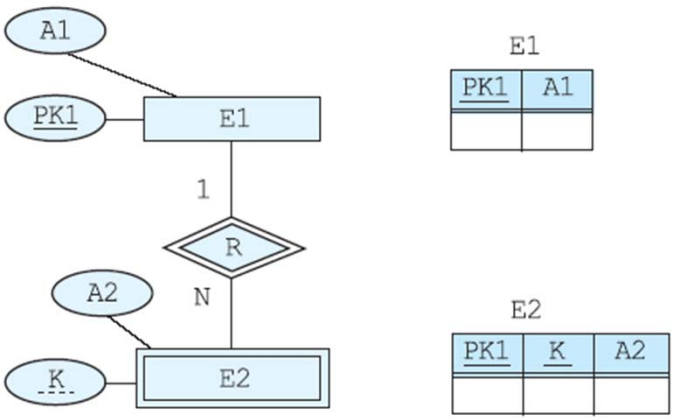
단계 **2:** 약한 엔티티 타입과 단일 값 애트리뷰트

✓ ER 스키마에서 소유 엔티티 타입 **E**를 갖는 **각 약한 엔티티 타입 W**에 대하여 릴레이션 **R**을 생성함

✓ W에 있던 모든 **단순 애트리뷰트들을 릴레이션 R**에 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 키**와 소유 엔티티 타입에 해당하는 릴레이션을 참조하는 **외래 키의 조합**으로 이루어짐



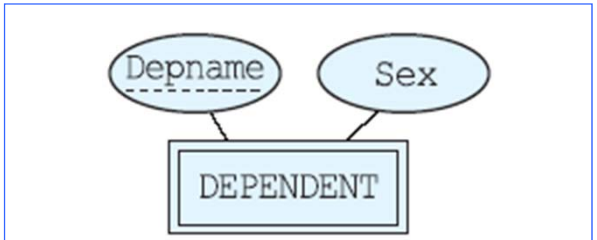
[그림 5.45] 약한 엔티티 타입을 릴레이션으로 사상

ER 스키마를 관계 모델의 릴레이션으로 사상

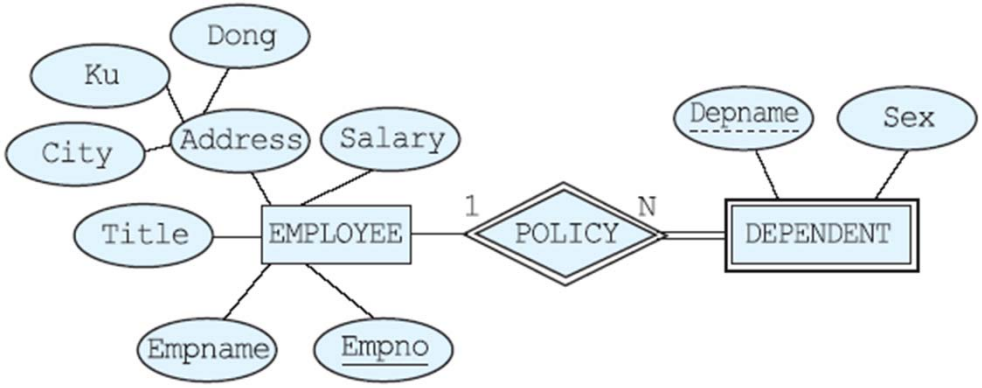
- 데이터베이스 설계 사례에 알고리즘 적용(계속)

  단계 **2:** 약한 엔티티 타입과 단일 값 애트리뷰트

  DEPENDENT(Empno, Depname, Sex)



[그림 5.35] DEPENDENT 엔티티 타입



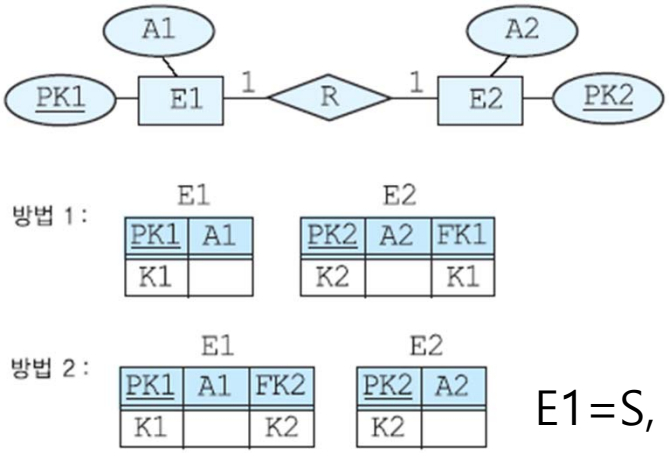
[그림 5.39] EMPLOYEE 엔티티 타입과 DEPENDENT 엔티티 타입 사이의 약한 관계 타입

ER 스키마를 관계 모델의 릴레이션으로 사상

❑ ER-관계 사상 알고리즘(계속)

단계 3: 2진 1:1 관계 타입

- ✓ ER 스키마의 각 2진 1:1 **관계 타입 R**에 대하여, **R에 참여하는 엔티티 타입에 대응되는 릴레이션 S와 T를 찾음**
- ✓ S와 T 중에서 한 릴레이션을 선택하여, **만일 S를 선택했다면 T의 기본 키를 S에 외래키로 포함시킴**
- ✓ **관계 타입 R이 가지고 있는 모든 단순 애트리뷰트**(복합 애트리뷰트를 갖고 있는 경우에는 복합 애트리뷰트를 구성하는 단순 애트리뷰트)들을 **S에 대응되는 릴레이션에 포함시킴**



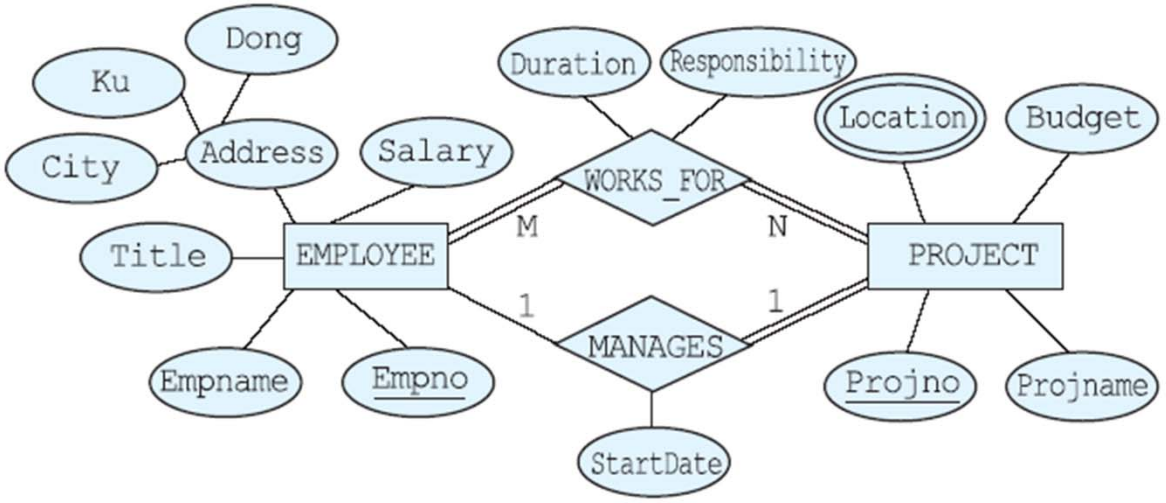
E1=S, E2=T

ER 스키마를 관계 모델의 릴레이션으로 사상

- 데이터베이스 설계 사례에 알고리즘 적용(계속)

  단계 **3: 2**진 **1:1** 관계 타입

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



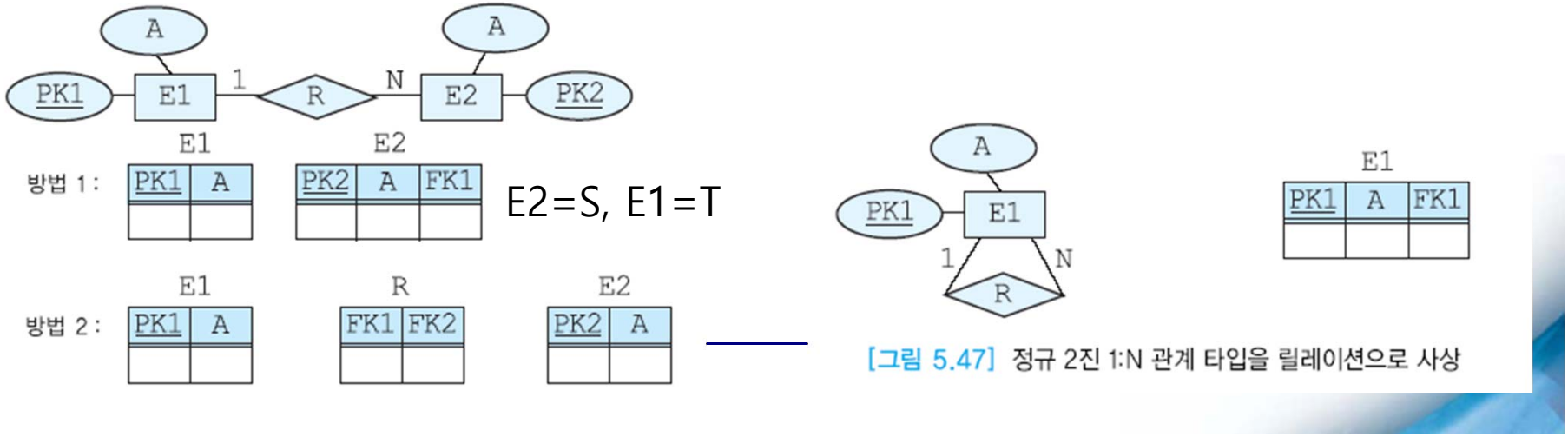
[그림 5.38] EMPLOYEE 엔티티 타입과 PROJECT 엔티티 타입 사이의 두 개의 관계 타입

ER 스키마를 관계 모델의 릴레이션으로 사상

❑ ER-관계 사상 알고리즘(계속)

단계 4: 정규 2진 1:N 관계 타입

- ✓ 정규 2진 1:N 관계 타입 **R**에 대하여 **N**측의 참여 엔티티 타입에 대응되는 릴레이션 **S**를 찾음
  - ✓관계 타입 R에 참여하는 **1**측의 엔티티 타입에 대응되는 릴레이션 **T**의 기본 키를 릴레이션 **S**에 외래 키로 포함시킴
  - ✓관계 타입 **R**이 가지고 있는 모든 단순 애트리뷰트(복합 애트리뷰트를 갖고 있는 경우에는 복합 애트리뷰트를 구성하는 단순 애트리뷰트)들을 **S**에 해당하는 릴레이션에 포함시킴



[그림 5.47] 정규 2진 1:N 관계 타입을 릴레이션으로 사상

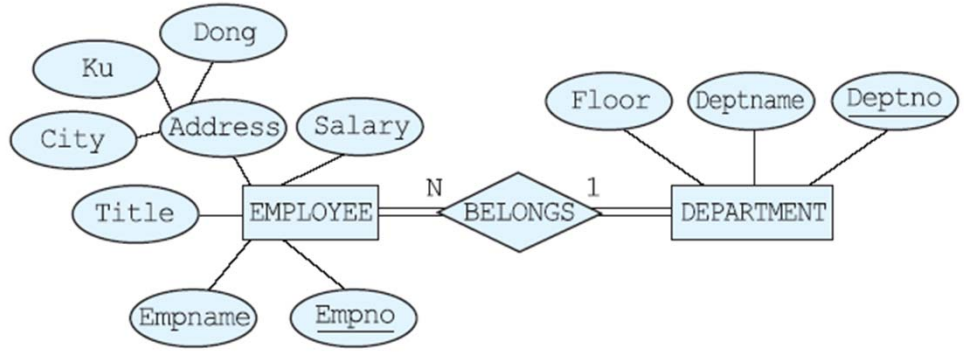
ER 스키마를 관계 모델의 릴레이션으로 사상

- 데이터베이스 설계 사례에 알고리즘 적용(계속)

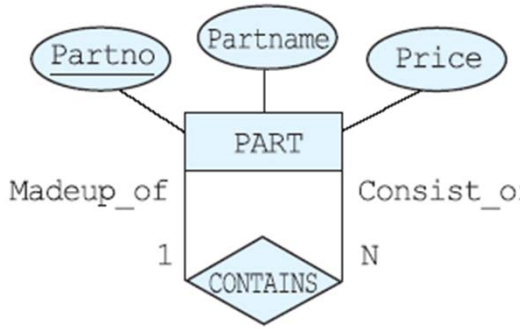
  단계 4: 정규 2진 1:N 관계 타입

  EMPLOYEE(<u>Empno</u>, Empname, Title, City, Ku, Dong,
  Salary, Dno)

  PART(<u>Partno</u>, Partname, Price, Subpartno)



[그림 5.37] EMPLOYEE 엔티티 타입과 DEPARTMENT 엔티티 타입 사이의 BELONGS 관계 타입



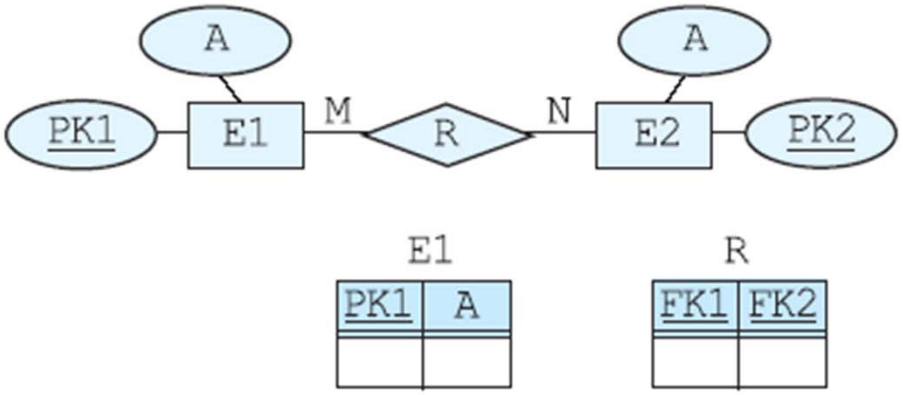
[그림 5.40] PART 엔티티 타입이 두 번 참여하는 순환적 관계 타입

ER 스키마를 관계 모델의 릴레이션으로 사상

❑ **ER-관계 사상 알고리즘(계속)**

단계 **5: 2진 M:N** 관계 타입

- ✓ 2진 M:N 관계 타입 **R**에 대해서는 릴레이션 **R**을 생성함
- ✓ 참여 엔티티 타입에 해당하는 **릴레이션들의 기본 키를 릴레이션 R**에 **외래 키로 포함**시키고, 이들의 조합이 릴레이션 **R**의 기본 키가 됨
- ✓ **관계 타입 R이 가지고 있는 모든 단순 애트리뷰트**(복합 애트리뷰트를 갖고 있는 경우에는 복합 애트리뷰트를 구성하는 단순 애트리뷰트)들을 **릴레이션 R**에 포함시킴



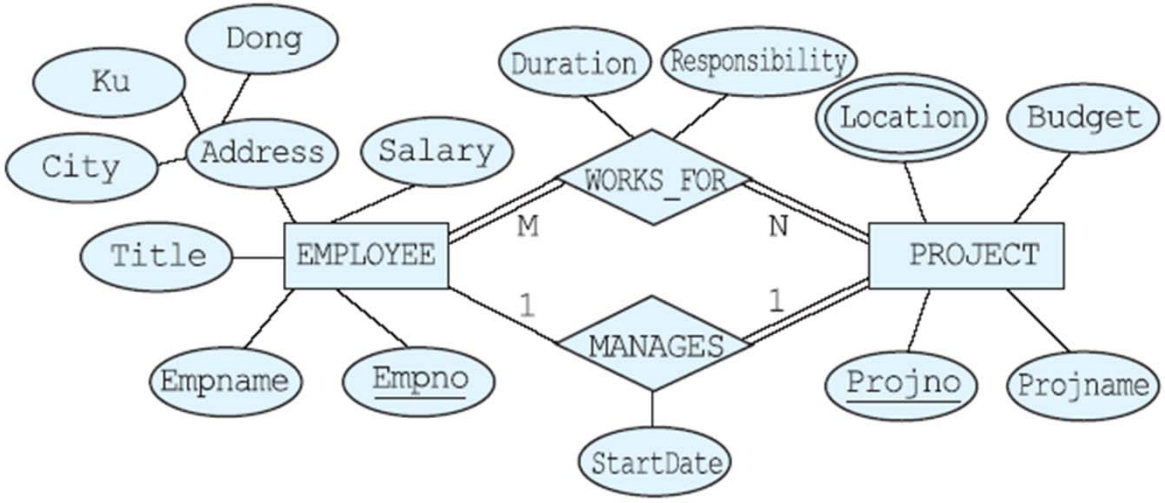
[그림 5.48] 2진 M:N 관계 타입을 릴레이션으로 사상

ER 스키마를 관계 모델의 릴레이션으로 사상

- 데이터베이스 설계 사례에 알고리즘 적용(계속)

  단계 **5: 2**진 **M:N** 관계 타입

  WORKS_FOR(Empno, Projno, Duration, Responsibility)



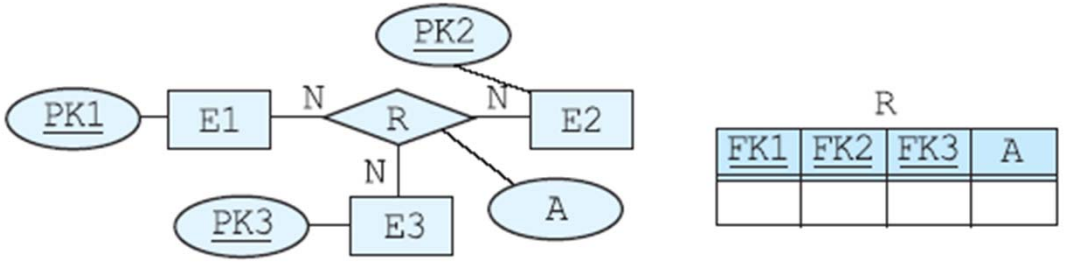
[그림 5.38] EMPLOYEE 엔티티 타입과 PROJECT 엔티티 타입 사이의 두 개의 관계 타입

ER 스키마를 관계 모델의 릴레이션으로 사상

❑ ER-관계 사상 알고리즘(계속)

단계 6: 3진 이상의 관계 타입

- ✓ 3진 이상의 각 관계 타입 R에 대하여 릴레이션 R을 생성함
- ✓ 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

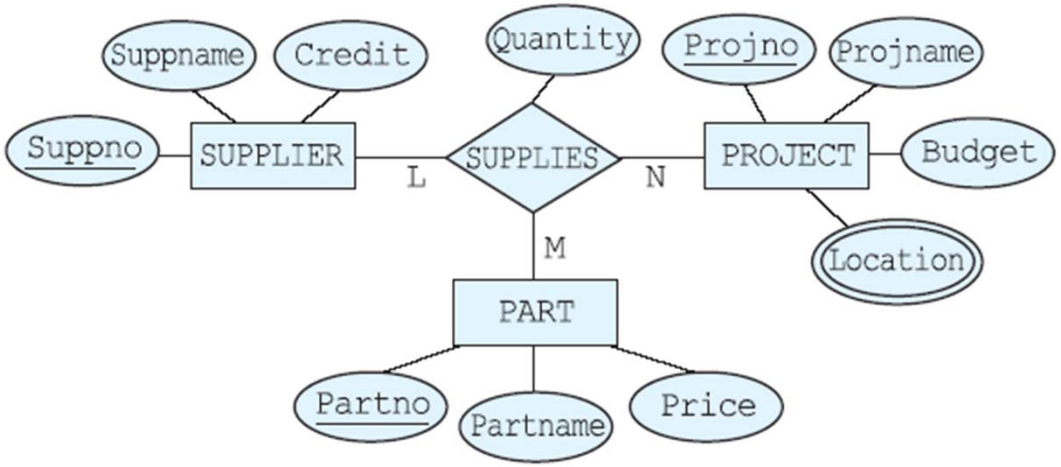


ER 스키마를 관계 모델의 릴레이션으로 사상

- 데이터베이스 설계 사례에 알고리즘 적용(계속)

  단계 **6: 3**진 이상의 관계 타입

SUPPLY(Suppno, Projno, Partno, Quantity)



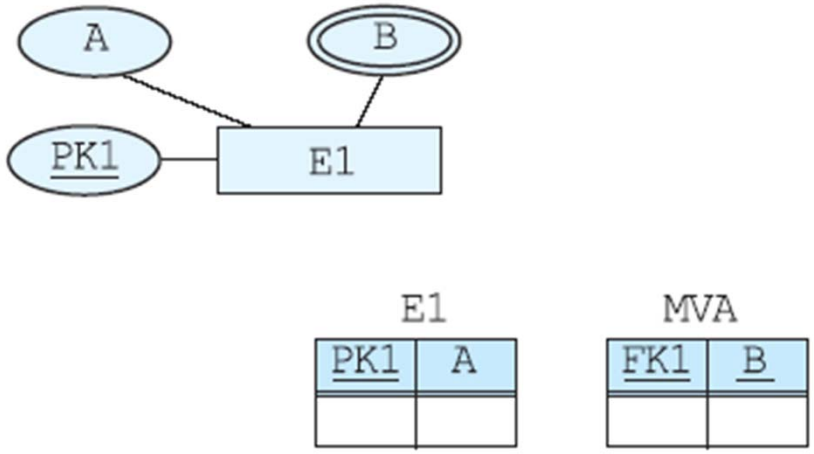
[그림 5.41] 세 개의 엔티티 타입이 참여하는 3진 관계 타입

ER 스키마를 관계 모델의 릴레이션으로 사상

❑ ER-관계 사상 알고리즘(계속)

단계 7: 다치 애트리뷰트

- ✓ 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



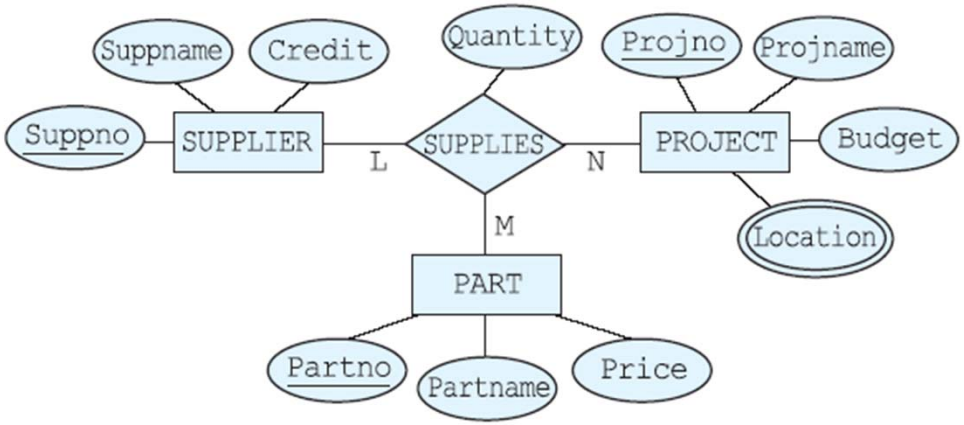
[그림 5.50] 다치 애트리뷰트를 릴레이션으로 사상

ER 스키마를 관계 모델의 릴레이션으로 사상

- 데이터베이스 설계 사례에 알고리즘 적용(계속)

단계 7: 다치 애트리뷰트

PROJ_LOC(Projno, Location)



[그림 5.41] 세 개의 엔티티 타입이 참여하는 3진 관계 타입

ER 스키마를 관계 모델의 릴레이션으로 사상

- 회사 **ER** 스키마는 관계 데이터베이스에서 총 **9**개의 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)

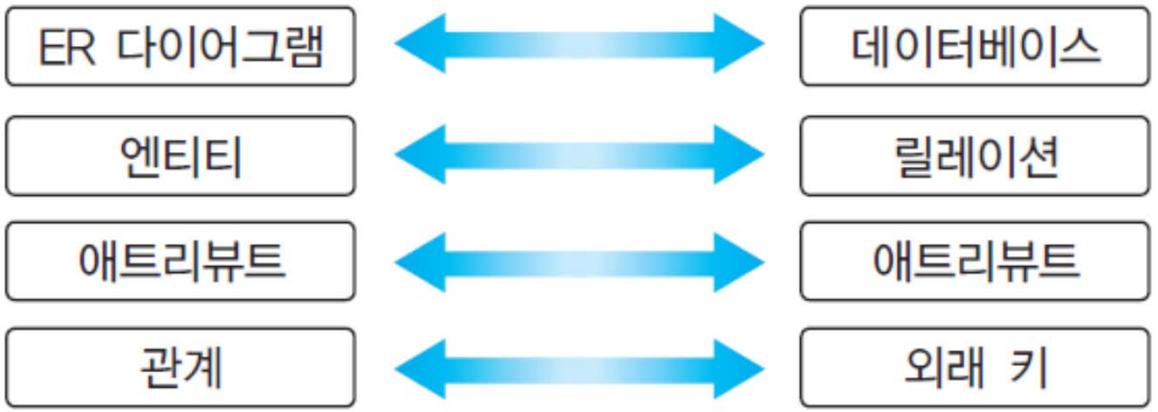
DEPENDENT(Empno, Depname, Sex)

WORKS_FOR(Empno, Projno, Duration, Responsibility)

SUPPLY(Suppno, Projno, Partno, Quantity)

PROJ_LOC(Projno, Location)

ER 스키마를 관계 모델의 릴레이션으로 사상



[그림 5.51] ER 개념과 데이터베이스 개념들의 대응 관계