

## 5.2 ER 모델

### □ ER 모델

- ✓ 데이터베이스 설계를 용이하게 하기 위해서 P.P. Chen이 1976년에 제안하였음
- ✓ 그 후에 많은 학자들이 이 모델을 강화시켰음
- ✓ 현재는 EER(Enhanced Entity Relationship) 모델이 데이터베이스 설계 과정에 널리 사용되고 있음
- ✓ 개념적 설계를 위한 인기 있는 모델로서, 많은 CASE 도구들에서 지원됨
- ✓ 실세계를 엔티티, 애트리뷰트, 엔티티들 간의 관계로 표현함
- ✓ 쉽게 관계 데이터 모델로 사상됨

## 5.2 ER 모델(계속)

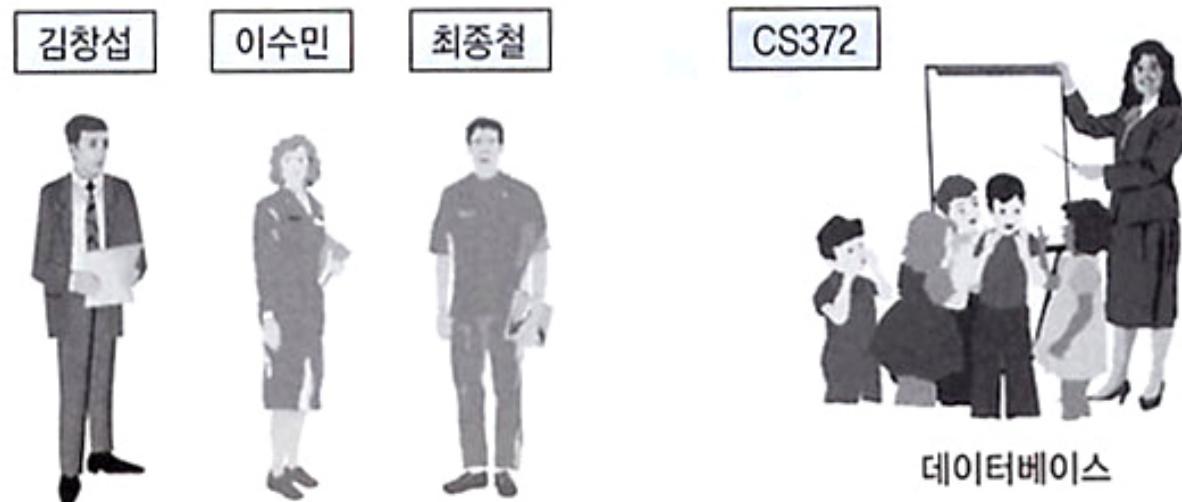
### □ ER 모델(계속)

- ✓ 기본적인 구문으로는 엔티티, 관계, 애트리뷰트가 있고, 기타 구문으로는 카디널리티 비율, 참여 제약조건 등이 있음
- ✓ 적은 노력으로 쉽게 배울 수 있고, 전문가가 아니어도 이해하기 쉬우며, 자연어보다는 좀더 정형적이고, 구현에 독립적이어서 데이터베이스 설계자들이 최종 사용자들과 의사 소통을 하는데 적합함
- ✓ ER 모델을 기반으로 만들어진 다수의 CASE 도구(예, **ERWin** 등)들이 존재함
- ✓ 이런 도구들은 ER 설계를 자동적으로 오라클, SQL Server, 사이베이스 등의 데이터 정의어로 변환하고, 어떤 도구는 XML로 변환함
- ✓ 현재는 데이터베이스 설계를 위한 다소 구형 그래픽 표기법

## 5.2 ER 모델(계속)

### □ 엔티티

- ✓ 하나의 엔티티는 사람, 장소, 사물, 사건 등과 같이 독립적으로 존재하면서 고유하게 식별이 가능한 실세계의 객체
- ✓ 사원처럼 실체가 있는 것도 있지만 생각이나 개념과 같이 추상적인 것도 있음



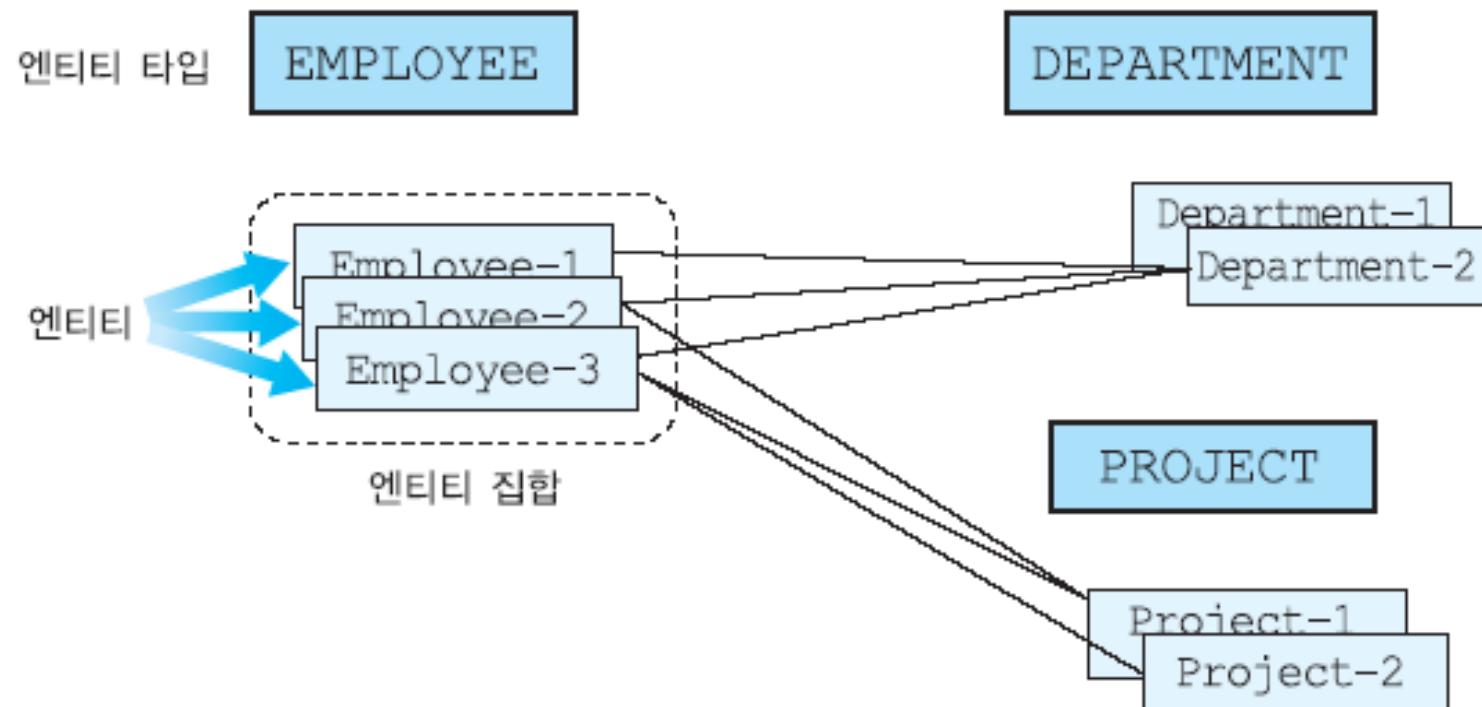
[그림 5.2] 엔티티의 예

## 5.2 ER 모델(계속)

### □ 엔티티 타입

- ✓ 엔티티들은 엔티티 타입(또는 엔티티 집합)들로 분류됨
- ✓ 엔티티 타입은 동일한 애트리뷰트들을 가진 엔티티들의 틀
- ✓ 엔티티 집합은 동일한 애트리뷰트들을 가진 엔티티들의 모임
- ✓ 하나의 엔티티는 한 개 이상의 엔티티 집합에 속할 수 있음
- ✓ 엔티티 타입은 관계 모델의 릴레이션의 내포에 해당하고, 엔티티 집합은 관계 모델의 릴레이션의 외연에 해당함
- ✓ 엔티티 집합과 엔티티 타입을 엄격하게 구분할 필요는 없음
- ✓ ER 다이어그램에서 엔티티 타입은 직사각형으로 나타냄

## 5.2 ER 모델(계속)



[그림 5.3] 엔티티, 엔티티 타입, 엔티티 집합

## 5.2 ER 모델(계속)

### □ 강한 엔티티 타입

- ✓ 강한 엔티티 타입(정규 엔티티 타입)은 독자적으로 존재하며 엔티티 타입 내에서 자신의 키 애트리뷰트를 사용하여 고유하게 엔티티들을 식별할 수 있는 엔티티 타입

### □ 약한 엔티티 타입

- ✓ 약한 엔티티 타입은 키를 형성하기에 충분한 애트리뷰트들을 갖지 못한 엔티티 타입
- ✓ 이 엔티티 타입이 존재하려면 소유 엔티티 타입이 있어야 함
- ✓ 소유 엔티티 타입의 키 애트리뷰트를 결합해야만 고유하게 약한 엔티티 타입의 엔티티들을 식별할 수 있음

## 5.2 ER 모델(계속)

### □ 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 탑 내에서 각 엔티티를 고유하게 식별함
- ✓ ER 다이어그램에서 기본 키에 속하는 애트리뷰트는 밑줄을 그어 표시함

## 5.2 ER 모델(계속)

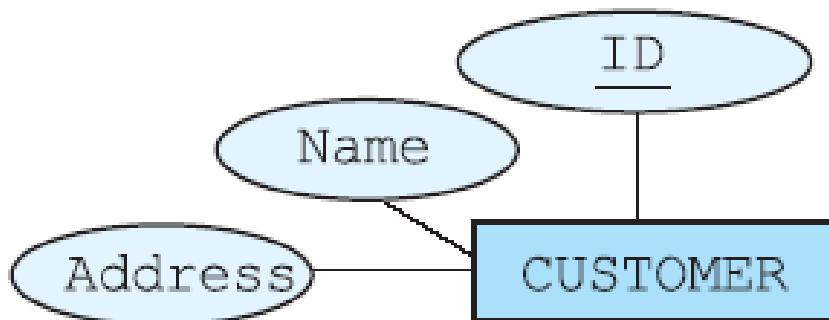
### □ 애트리뷰트(계속)

- ✓ 요구사항 명세에서 명사나 형용사로 표현됨
- ✓ 엔티티는 독립적인 의미를 갖는데 반해서 애트리뷰트는 독립적인 의미를 갖지 않음
- ✓ ER 다이어그램에서 타원형으로 나타냄
- ✓ 애트리뷰트와 엔티티 탑입은 실선으로 연결

## 5.2 ER 모델(계속)

### □ 단순 애트리뷰트(simple attribute)

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

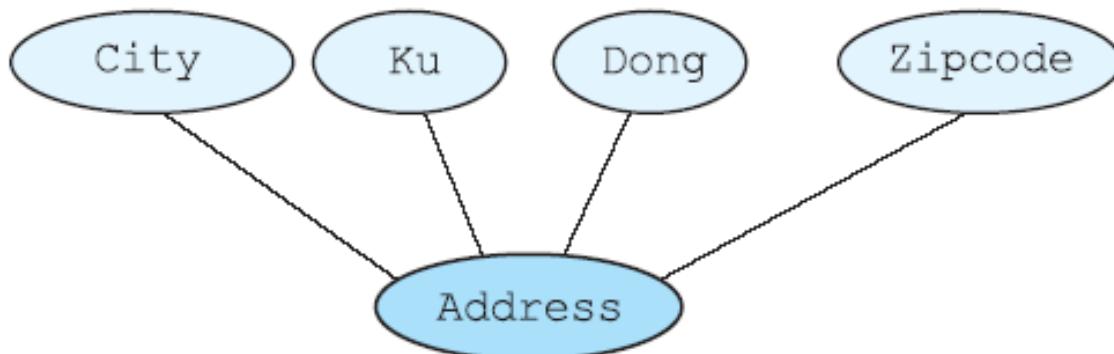


[그림 5.4] 단순 애트리뷰트

## 5.2 ER 모델(계속)

### 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

## 5.2 ER 모델(계속)

### □ 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

## 5.2 ER 모델(계속)

### □ 다치 애트리뷰트(multi-valued attribute)

- ✓ 각 엔티티마다 여러 개의 값을 가질 수 있는 애트리뷰트
- ✓ ER 다이어그램에서 이중선 타원으로 표현함



[그림 5.6] 다치 애트리뷰트

## 5.2 ER 모델(계속)

### □ 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

## 5.2 ER 모델(계속)

### □ 유도된 애트리뷰트(derived attribute)

- ✓ 다른 애트리뷰트의 값으로부터 얻어진 애트리뷰트
- ✓ 관계 데이터베이스에서 릴레이션의 애트리뷰트로 포함시키지 않는 것이 좋음
- ✓ ER 다이어그램에서 점선 타원으로 표현함

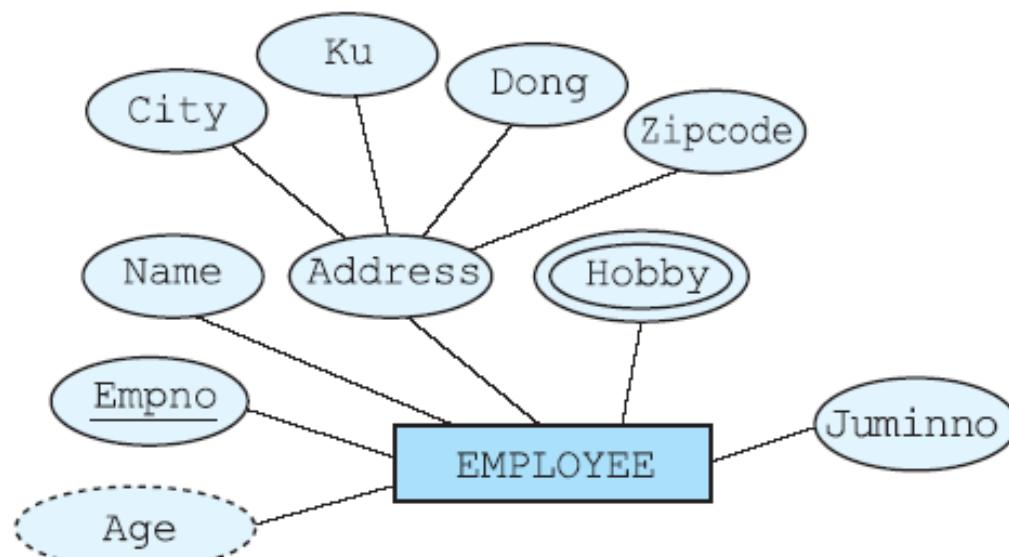


[그림 5.7] 유도된 애트리뷰트

## 5.2 ER 모델(계속)

### 예 : 애트리뷰트들의 유형

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



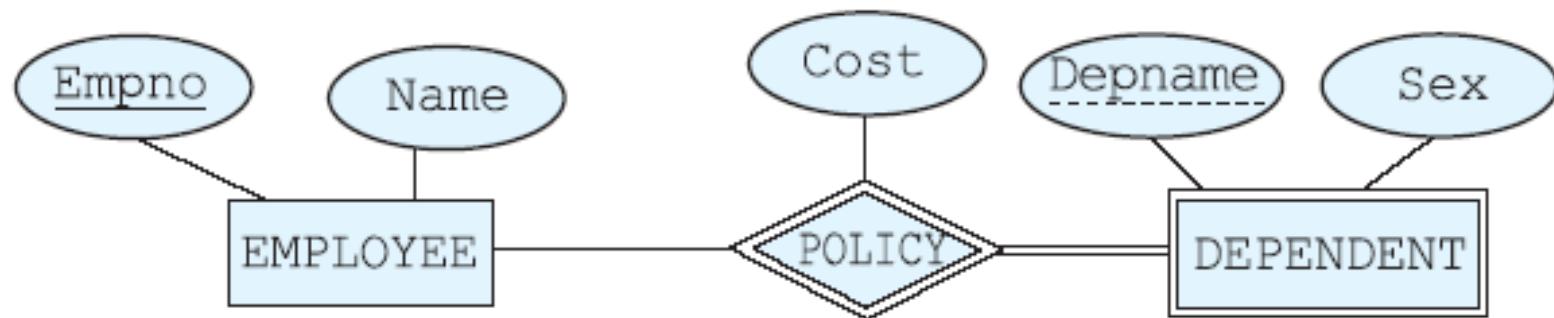
[그림 5.8] 여러 가지 애트리뷰트의 예

## 5.2 ER 모델(계속)

### □ 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

## 5.2 ER 모델(계속)



[그림 5.9] 약한 엔티티 타입

## 5.2 ER 모델(계속)

### □ 관계와 관계 타입

- ✓ 관계는 엔티티들 사이에 존재하는 연관이나 연결로서 두 개 이상의 엔티티 타입들 사이의 사상으로 생각할 수 있음
- ✓ 관계 집합은 동질의 관계들의 집합
- ✓ 관계 타입은 동질의 관계들의 틀
- ✓ 관계 집합과 관계 타입을 엄격하게 구분할 필요는 없음
- ✓ 요구사항 명세에서 흔히 동사는 ER 다이어그램에서 관계로 표현됨
- ✓ ER 다이어그램에서 다이어몬드로 표기
- ✓ 관계 타입이 서로 연관시키는 엔티티 타입들을 관계 타입에 실선으로 연결함

## 5.2 ER 모델(계속)



[그림 5.10] 관계 타입 WORKS\_FOR

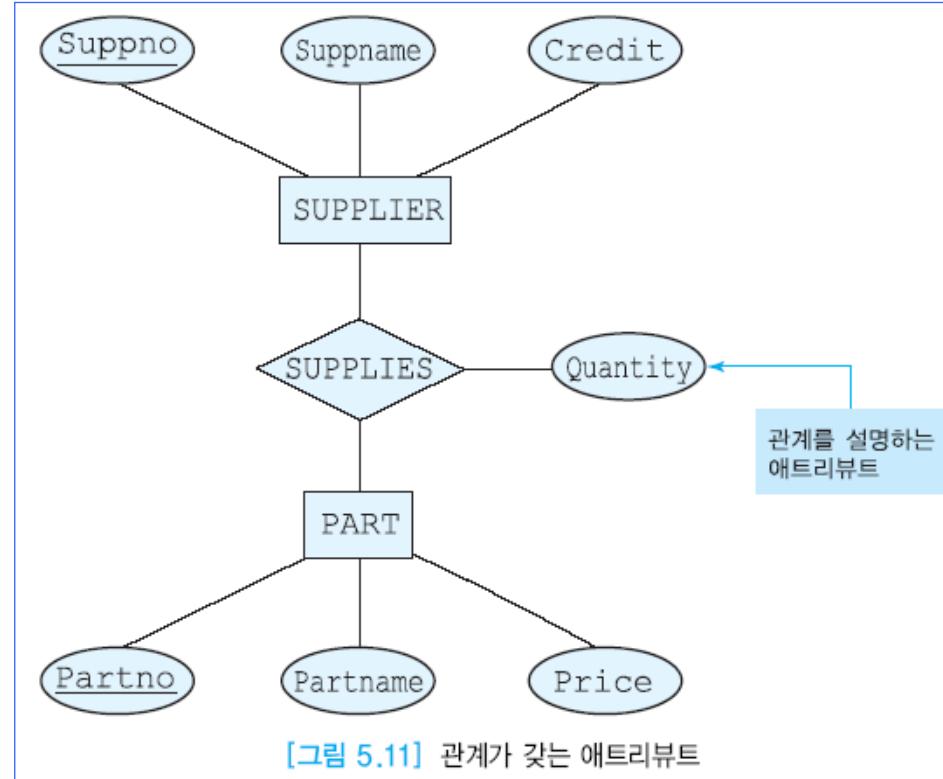
〈표 5.2〉 엔티티와 엔티티 간의 관계의 예

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

## 5.2 ER 모델(계속)

### □ 관계의 애트리뷰트

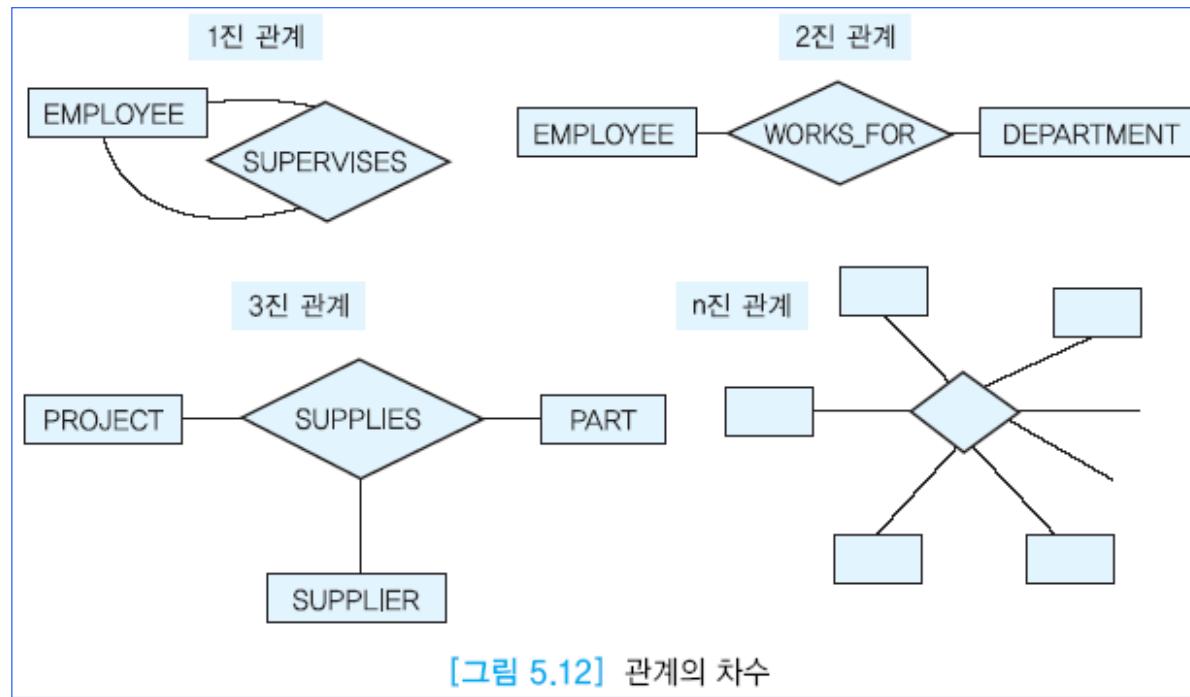
- ✓ 관계 타입은 관계의 특징을 기술하는 애트리뷰트들을 가질 수 있음
- ✓ 관계 타입은 키 애트리뷰트를 갖지 않음



## 5.2 ER 모델(계속)

### □ 차수(degree)

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

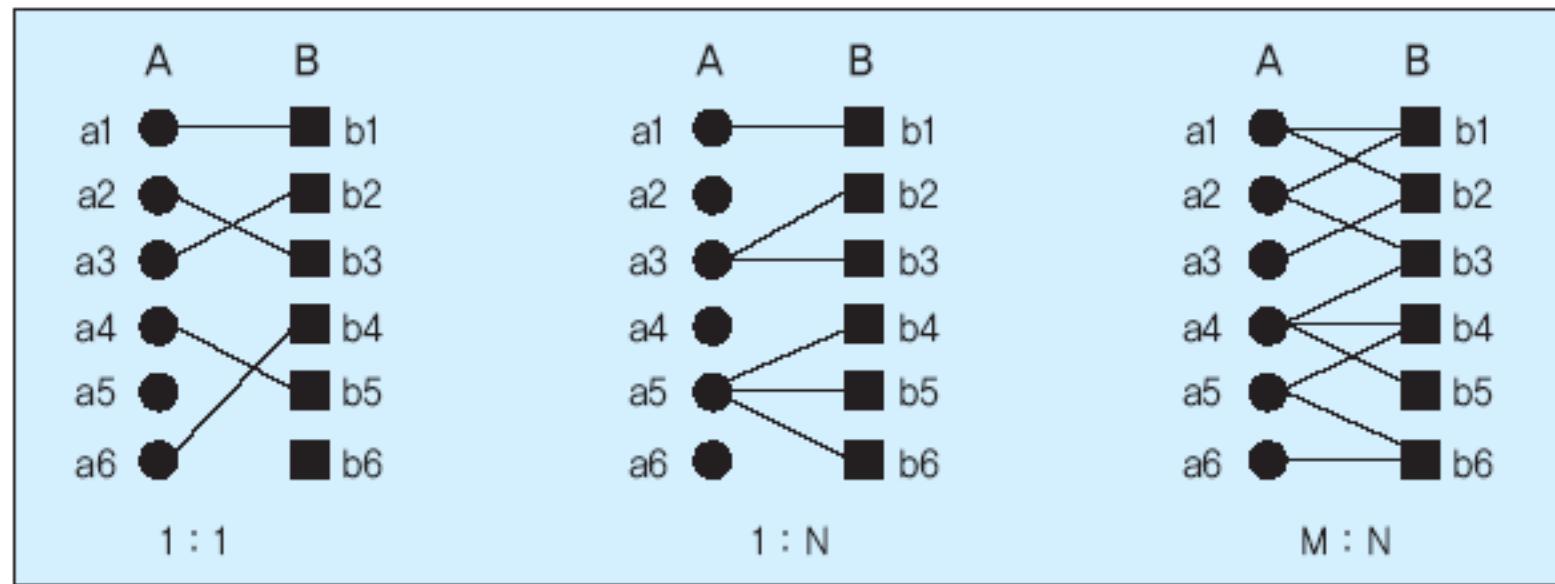


## 5.2 ER 모델(계속)

### □ 카디널리티 비율

- ✓ 카디널리티 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
- ✓ 카디널리티 비율에 관한 정보는 간선 위에 나타냄

## 5.2 ER 모델(계속)



[그림 5.14] 카디널리티 비율

## 5.2 ER 모델(계속)

### □ 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

## 5.2 ER 모델(계속)

### □ M:N 관계

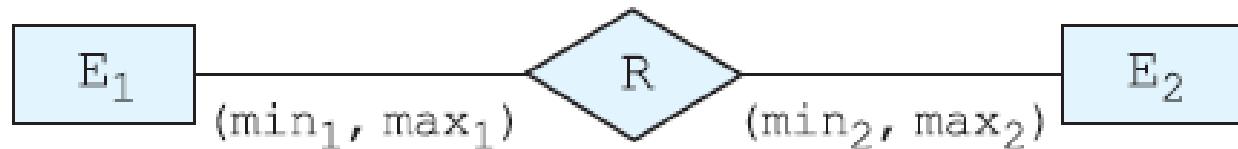
- ✓ 한 엔티티 타입에 속하는 임의의 개수의 엔티티가 다른 엔티티 타입에 속하는 임의의 개수의 엔티티와 연관됨
- ✓ 예: 각 사원에 대해 여러 대의 PC가 있고, 각 PC에 대해 여러 명의 사원들이 있으면 사원과 PC 간의 관계는 M:N 관계

## 5.2 ER 모델(계속)

### □ 카디널리티 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

## 5.2 ER 모델(계속)

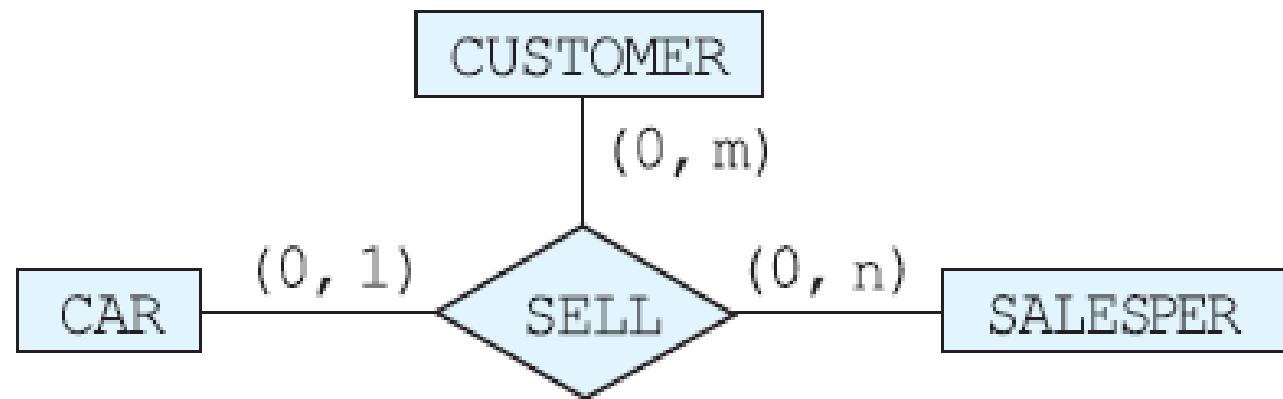


[그림 5.16] 카디날리티의 최소값과 최대값

〈표 5.3〉 카디날리티들의 몇 가지 유형

관계	$(\text{min}1, \text{max}1)$	$(\text{min}2, \text{max}2)$	그래픽 표기	화살표 표기
1 : 1	(0, 1)	(0, 1)		
1 : N	(0, *)	(0, 1)		
M : N	(0, *)	(0, *)		

## 5.2 ER 모델(계속)

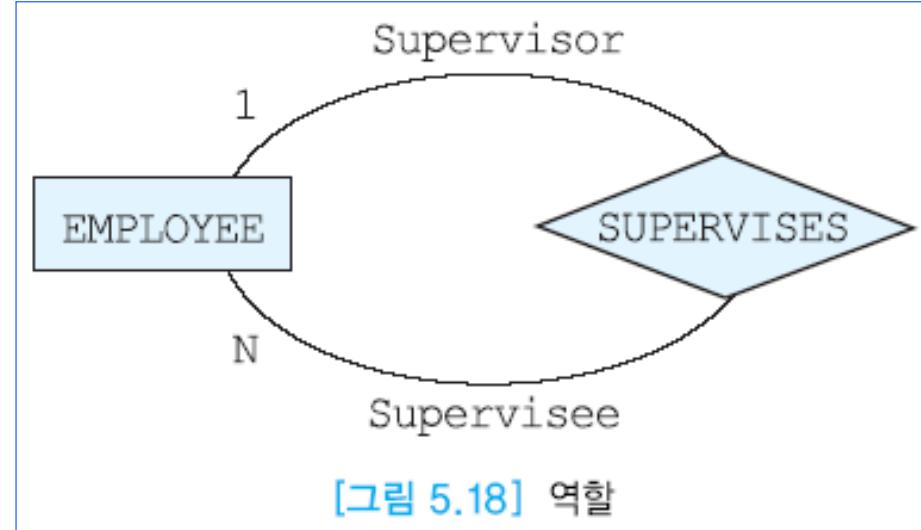


[그림 5.17] 카디널리티가 명시된 3진 관계 타입

## 5.2 ER 모델(계속)

### □ 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



## 5.2 ER 모델(계속)

### □ 전체 참여와 부분 참여

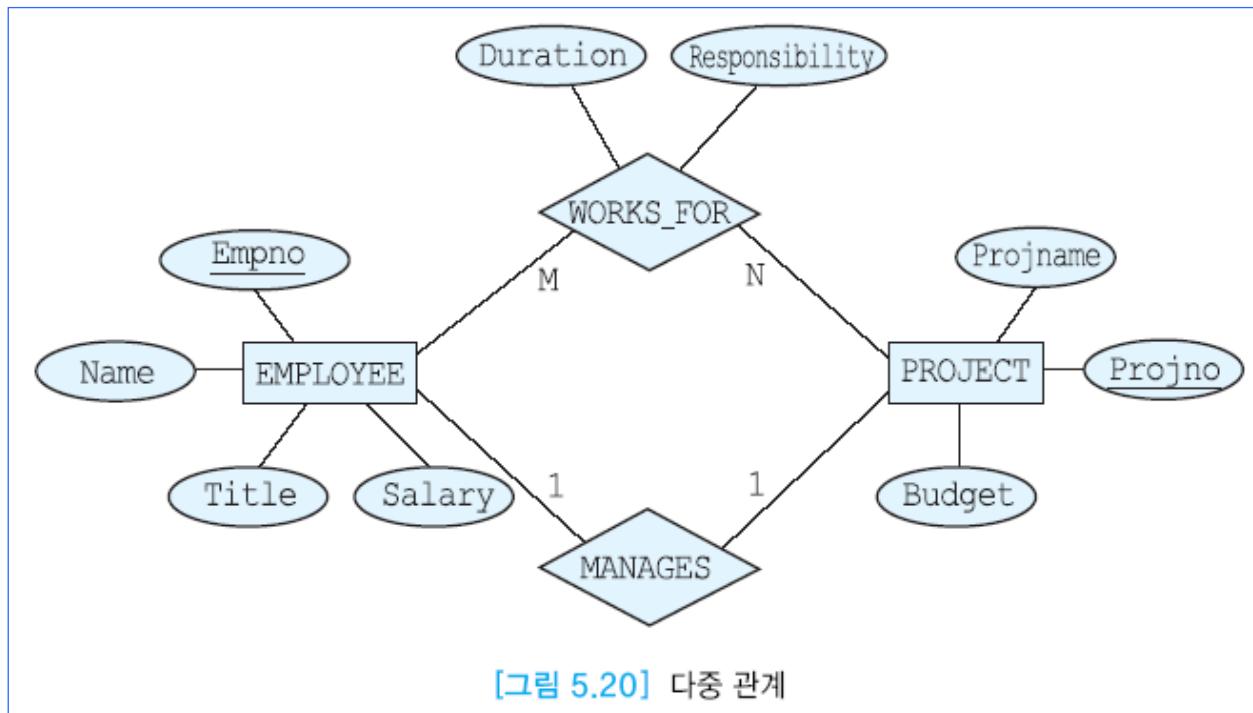
- ✓ 전체 참여는 어떤 관계에 엔티티 타입 E1의 모든 엔티티들이 관계 타입 R에 의해서 어떤 엔티티 타입 E2의 어떤 엔티티와 연관되는 것을 의미
- ✓ 부분 참여는 어떤 관계에 엔티티 타입 E1의 일부 엔티티만 참여하는 것을 의미
- ✓ 약한 엔티티 타입은 항상 관계에 전체 참여
- ✓ 전체 참여는 ER 다이어그램에서 이중 실선으로 표시
- ✓ 카디널리티 비율과 함께 참여 제약조건은 관계에 대한 중요한 제약조건



## 5.2 ER 모델(계속)

### □ 다중 관계

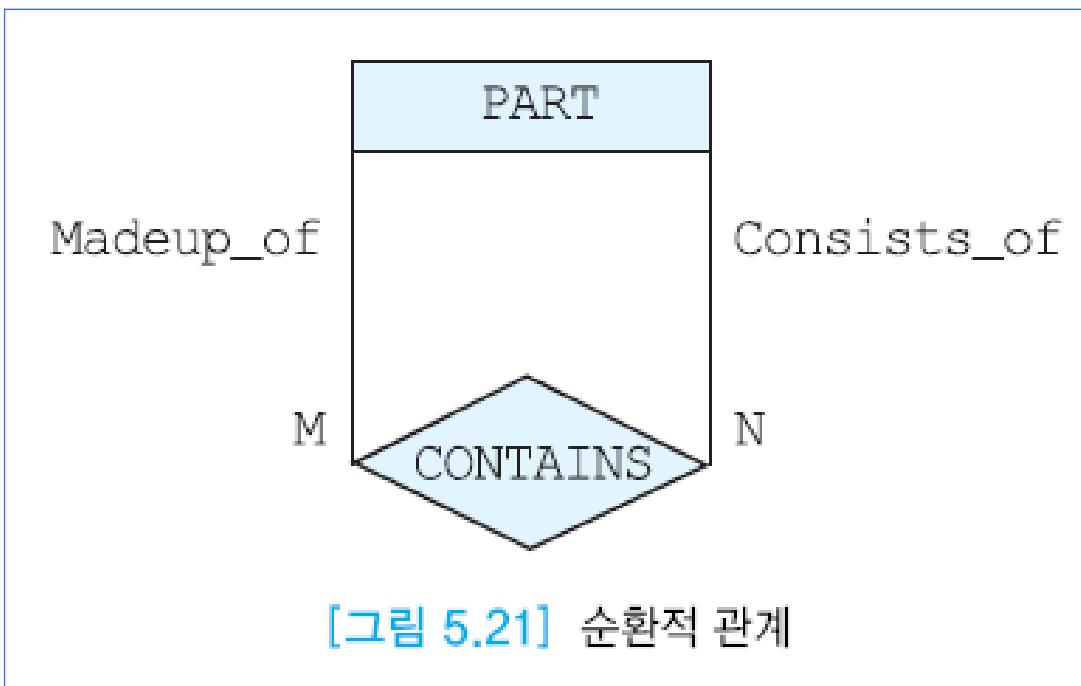
- ✓ 두 엔티티 타입 사이에 두 개 이상의 관계 타입이 존재할 수 있음



## 5.2 ER 모델(계속)

### □ 순환적 관계

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



## 5.2 ER 모델(계속)

### □ ER 스키마를 작성하기 위한 지침

- ✓ 엔티티는 키 애트리뷰트 이외에 설명 정보를 추가로 가짐
- ✓ 다치 애트리뷰트는 엔티티로 분류해야 함
- ✓ 가능한 한 복합 식별자를 피함
- ✓ 어떤 개념을 엔티티 타입과 관계 타입 중 어느 것으로 모델링할 것인지를 절대적으로 구분하는 것은 어려움

## 5.2 ER 모델(계속)

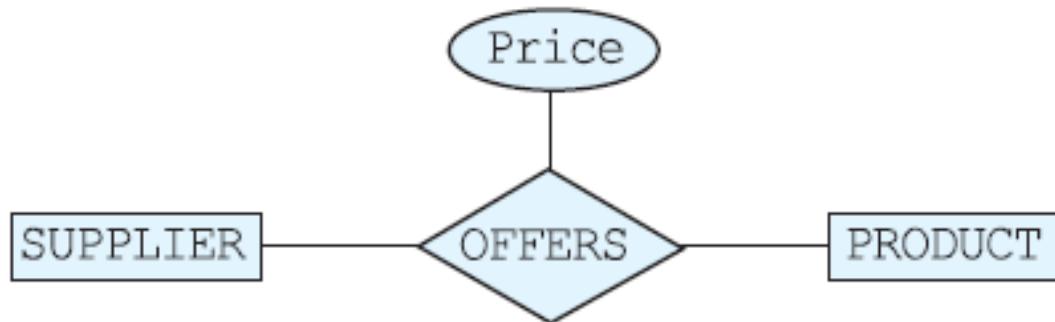
### □ 애트리뷰트 vs. 엔티티

- ✓ 엔티티 타입과 애트리뷰트를 구분하는 절대적인 기준 없음
- ✓ 예: 공급자에 대한 정보가 아래와 같다

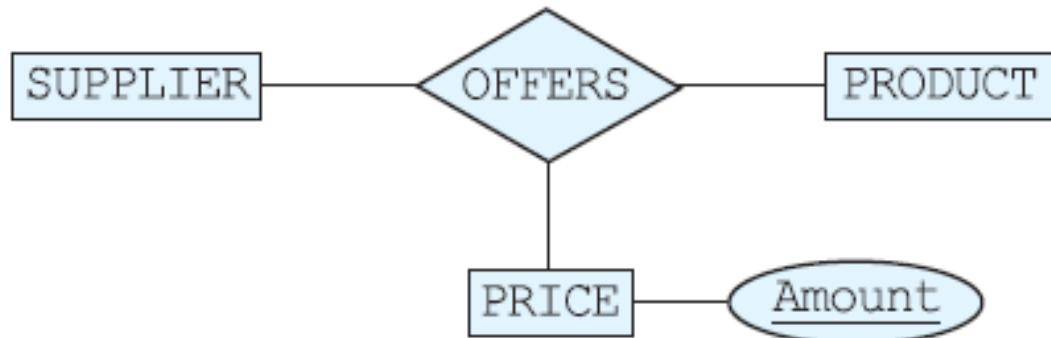
공급자 번호, 공급자 이름, 신용, 공급자 도시

- 공급자가 엔티티인 것은 명확
- 공급자 도시가 엔티티인가 또는 애트리뷰트인가?
- 고려사항:
  - 도시가 조직체에 관심이 있는 객체인가?
  - 도시에 관한 애트리뷰트들을 유지할 필요가 있는가?
  - 도시를 여러 엔티티 타입들이 공유하는가?
- 설계 방안
  - 위 고려사항 중에 하나라도 대답이 ‘예’라면 도시에 대한 추가 정보를 모아서 엔트리로 나타냄. 그렇지 않으면 애트리뷰트로 표현.

## 5.2 ER 모델(계속)



[그림 5.22] Price가 관계에 애트리뷰트로 사용됨



[그림 5.23] Price가 엔티티 타입으로 모델링됨

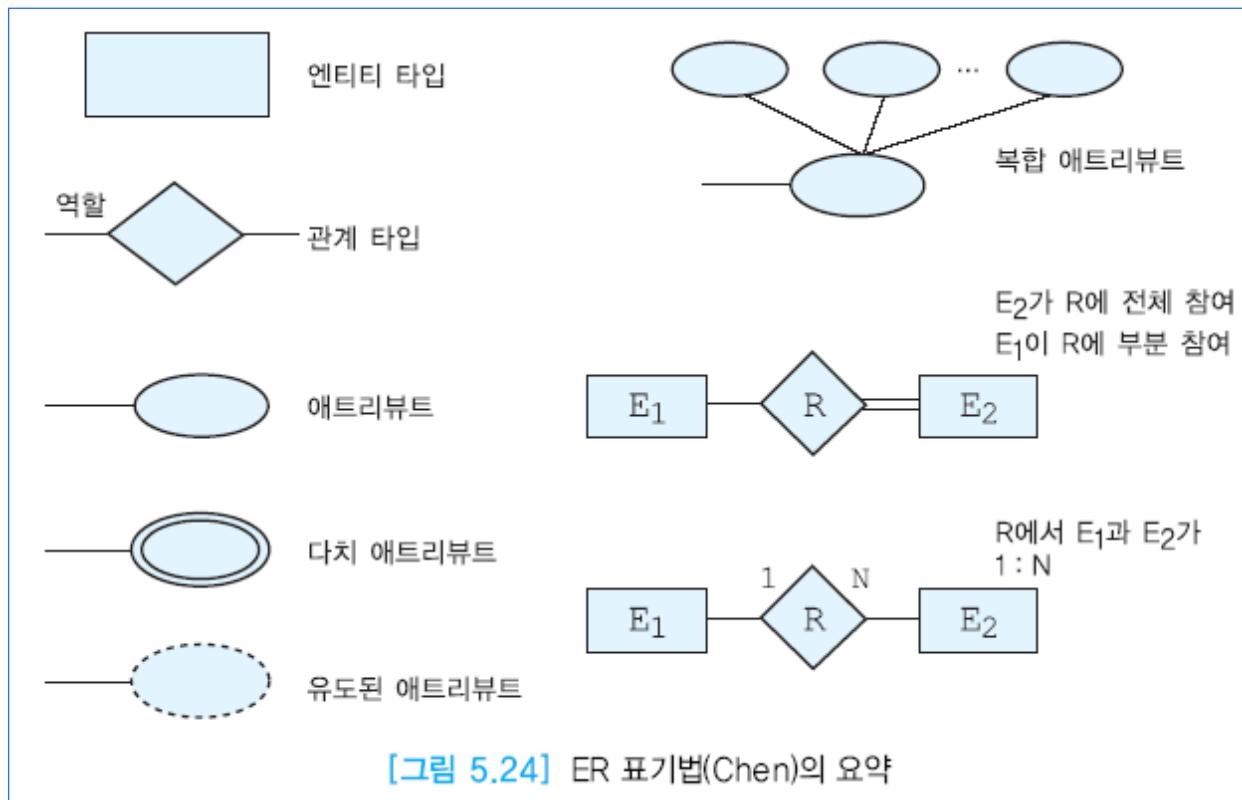
## 5.2 ER 모델(계속)

### □ 데이터베이스 설계 과정

- ✓ 응용의 요구사항을 수집하여 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 식별하고, 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

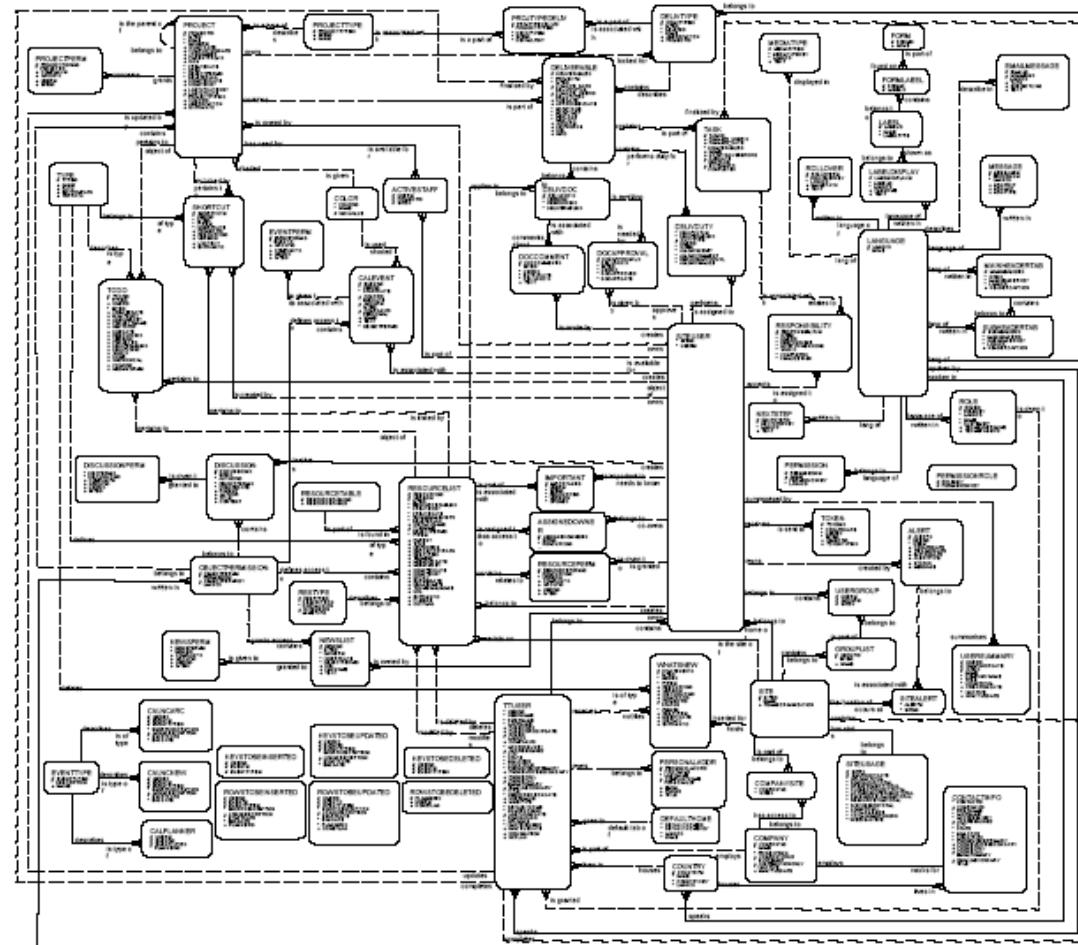
## 5.2 ER 모델(계속)

### □ 본 책의 ER 표기법의 요약



## 5.2 복잡한 ER 다이터그램의 예

### A Large ER Diagram



## 5.2 ER 모델(계속)

### □ ER 모델의 또 다른 표기법

- ✓ 본 장에서 사용한 표기법으로 수십 개 이상의 애트리뷰트가 엔티티 타입에 연결된 다이어그램을 나타내려면 매우 불편하고 공간을 많이 차지
- ✓ ERWin 등의 CASE 도구들에서는 **새발**(crow-feet) 표기법이 흔히 사용됨
- ✓ 새발 표기법에도 여러 가지 변형들이 존재함

## 5.2 ER 모델(계속)

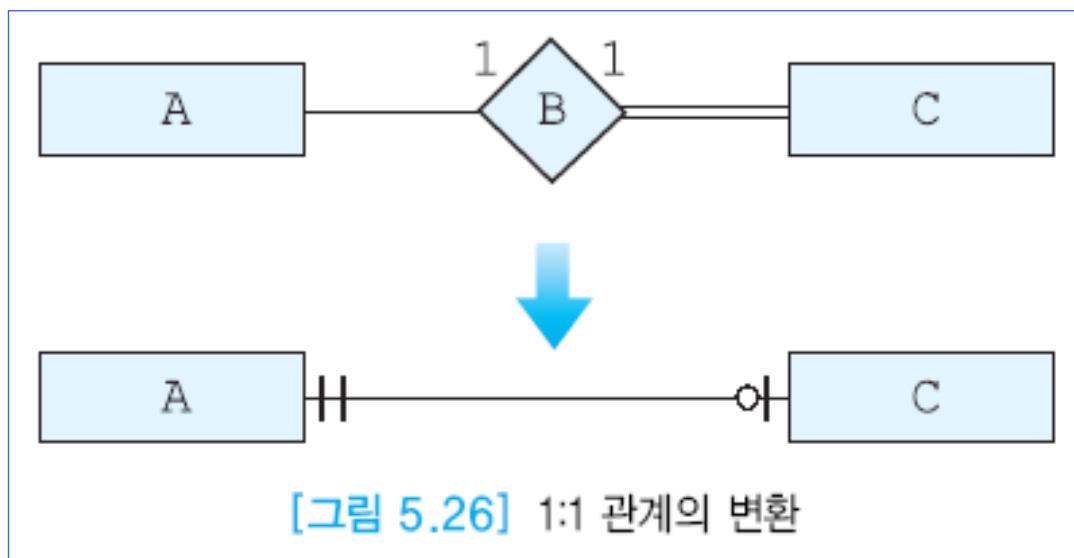
	1:1 관계. 엔티티 A의 각 인스턴스는 엔티티 B의 0 또는 1개의 인스턴스와 연관됨. 엔티티 B의 각 인스턴스는 엔티티 A의 0 또는 1개의 인스턴스와 연관됨.
	1:N 관계. 엔티티 A의 각 인스턴스는 엔티티 B의 0개 이상의 인스턴스와 연관됨. 엔티티 B의 각 인스턴스는 엔티티 A의 0 또는 1개의 인스턴스와 연관됨.
	M:N 관계. 엔티티 A의 각 인스턴스는 엔티티 B의 0개 이상의 인스턴스와 연관됨. 엔티티 B의 각 인스턴스는 엔티티 A의 0 개 이상의 인스턴스와 연관됨.
	1:1 관계. 엔티티 A의 각 인스턴스는 엔티티 B의 1개의 인스턴스와 연관됨. 엔티티 B의 각 인스턴스는 엔티티 A의 0 또는 1 개의 인스턴스와 연관됨.
	1:N 관계. 엔티티 A의 각 인스턴스는 엔티티 B의 1개 이상의 인스턴스와 연관됨. 엔티티 B의 각 인스턴스는 엔티티 A의 0 또는 1개의 인스턴스와 연관됨.
	M:N 관계. 엔티티 A의 각 인스턴스는 엔티티 B의 1개 이상의 인스턴스와 연관됨. 엔티티 B의 각 인스턴스는 엔티티 A의 0 개 이상의 인스턴스와 연관됨.

- ○ : 0을 의미
- | : 1을 의미
- ← : 이상을 의미

[그림 5.25] 새발 표기법의 예

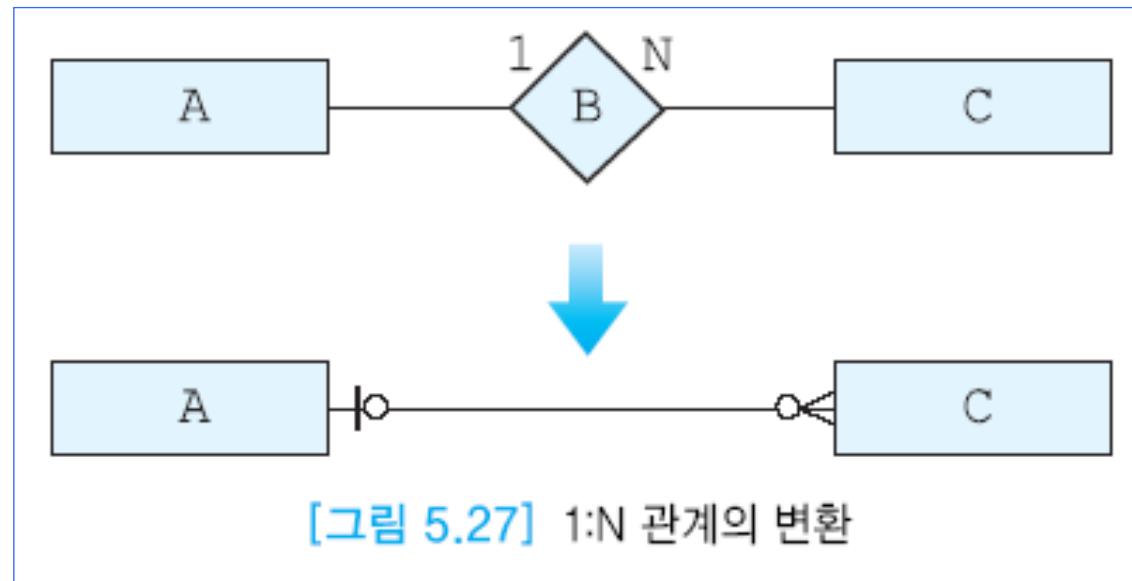
## 5.2 ER 모델(계속)

- 본 책의 표기법을 새발 표기법으로 표현하는 방법
  - ✓ 1:1 관계



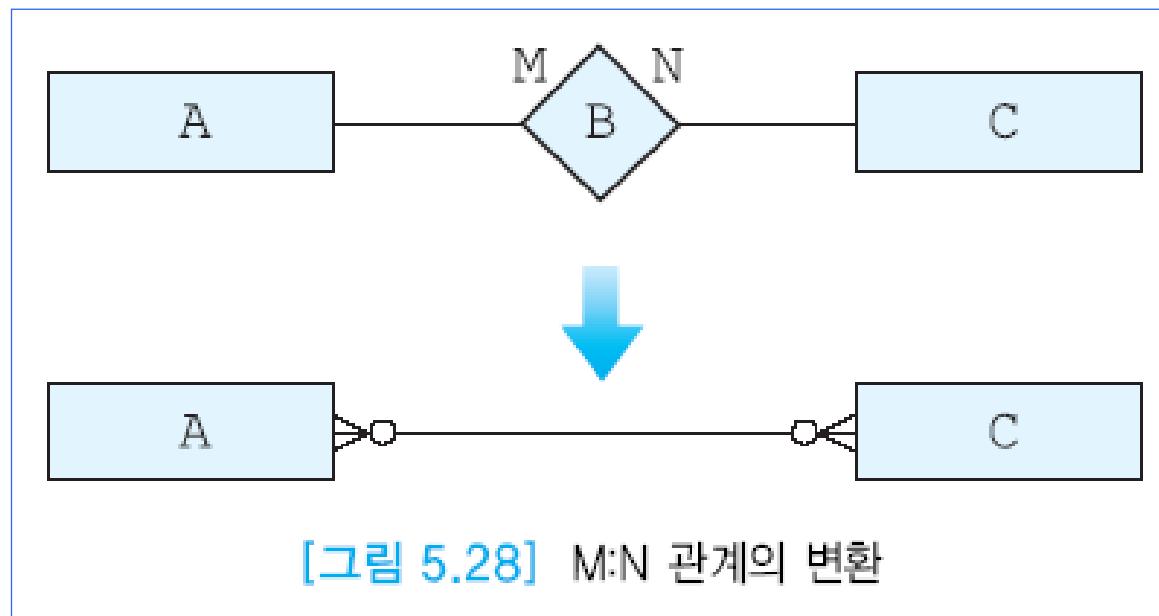
## 5.2 ER 모델(계속)

- 본 책의 표기법을 새발 표기법으로 표현하는 방법(계속)
  - ✓ 1:N 관계



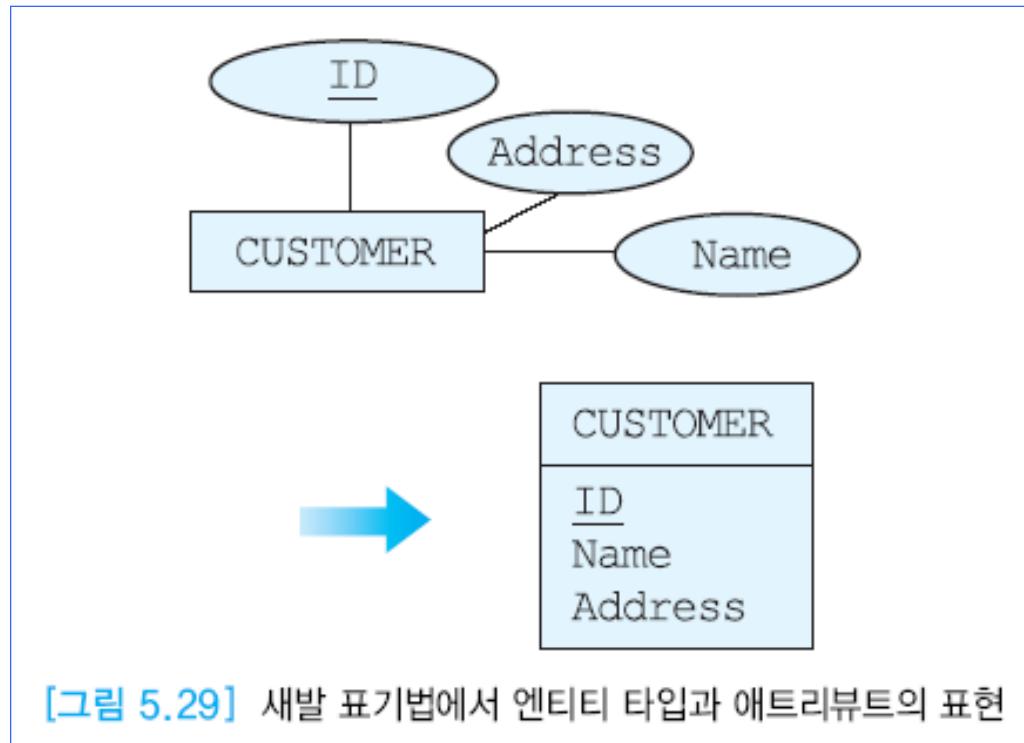
## 5.2 ER 모델(계속)

- 본 책의 표기법을 새발 표기법으로 표현하는 방법(계속)
  - ✓ M:N 관계

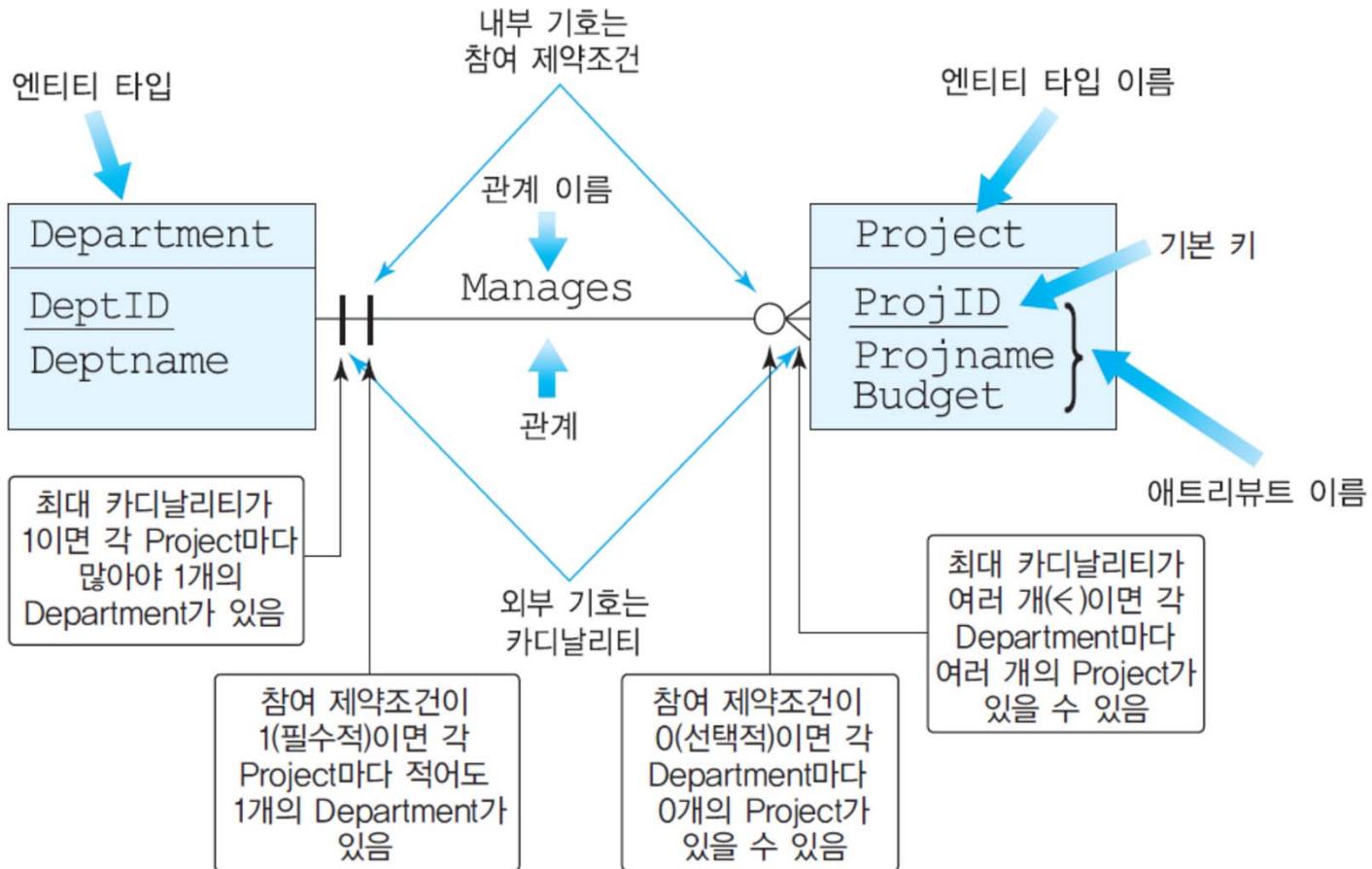


## 5.2 ER 모델(계속)

- 본 책의 표기법을 새발 표기법으로 표현하는 방법(계속)
  - ✓ 엔티티 타입과 애트리뷰트



## 5.2 ER 모델(계속)



[그림 5.30] 새발 표기법의 요약