

# Conceptual Data Modelling using Entity-Relationship Model

## Contents

- 예제 데이터베이스 응용(COMPANY)
- ER 모델의 개념
  - 엔티티와 애트리뷰트
  - 엔티티 타입, 값 집합, 키 애트리뷰트
  - 관계와 관계 타입
  - 약한 엔티티 타입
  - 역할과 관계 타입에서의 애트리뷰트
- ER 다이어그램 - 표기법
- COMPANY 스키마를 위한 ER 다이어그램
- 다른 표기법 - UML 클래스 다이어그램, 기타

### 개체(Entity)

개체란 단독으로 존재하는 객체를 의미하며, 동일한 개체는 존재하지 않습니다.  
예) 학생 정보가 학번, 이름, 학년이 있을 때, 3개의 정보가 같은 학생이 오직 한 명이면 이를 개체라고 함.  
학생 한 명이 개체가 되는 것.

이러한 개체들의 집합 Entity type ⇒ 네로로 표현

### 애트리뷰트, 속성(attribute)

• 개체가 갖는 속성을 의미함. 원으로 표시  
Student 테이블 (학번), (이름), (학년) 같은 정보를 속성이라 한다.

### 관계 (relation)

Entity type 간의 관계를 의미합니다.  
수강을 뜻하는 Takes는 학생과 강의의 "수강"이라는 관계를 갖는다.

### 5. DEFAULT

해당 필드의 기본값을 설정할 수 있게 해준  
레코드 입력 할 때 해당 필드 값을 전달하지 않으면  
자동으로 설정된 기본 값을 저장한다.

### 1. CASCADE

참조되는 테이블에서 데이터를 삽입하거나  
수정하면 참조하는 테이블에서도 삽입과  
수정이 같이 이루어짐

### 2. SET NULL

- 참조되는 테이블에서 데이터를 삭제하거나  
수정하면 참조하는 테이블의 데이터는  
NULL로 변경됨

### 3. NO ACTION

- " 참조하는 테이블의  
데이터는 변경이 없음."

### 4. SET DEFAULT

- " 참조하는 테이블의 데이터는  
필드의 기본값으로 설정"

### 5. RESTRICT

- 참조하는 테이블의 데이터가 남아있으면 참조되는 테이블에 대한 수정X

## Phases of DB Design

### 제약 조건

1. NOT NULL 해당 필드는 NULL 값을 저장할 수 없다.  
이 제약 조건에 설정된 필드는 무조건 데이터를  
가지고 있어야 함.

2. UNIQUE 해당 필드는 서로 다른 값을 가짐해야.  
이 제약 조건에 설정된 필드는 중복된 값을 저장할 수  
없다.

3. PRIMARY KEY 해당 필드는 Not Null 과 Unique 제약 조건의  
특성을 모두 가진다.  
= 기본키  
이 제약 조건이 설정된 필드는 Null값 X, 중복값 X

4. FOREIGN KEY 외래키, 한 테이블을 다른 테이블과 연결해주는 역할  
외래키가 설정된 테이블에 레코드를 입력하면, 그에 따른 테이블의 내용을  
참조해서 레코드가 입력됨.  
즉, Foreign Key 제약 조건은 해외의 테이블을 다른 테이블에 의존하게  
만든다.

FOREIGN KEY를 설정할 때, 참조되는 테이블의 필드는 반드시  
UNIQUE OR PRIMARY KEY가 설정되어 있어야 한다.

CREATE TABLE Test2

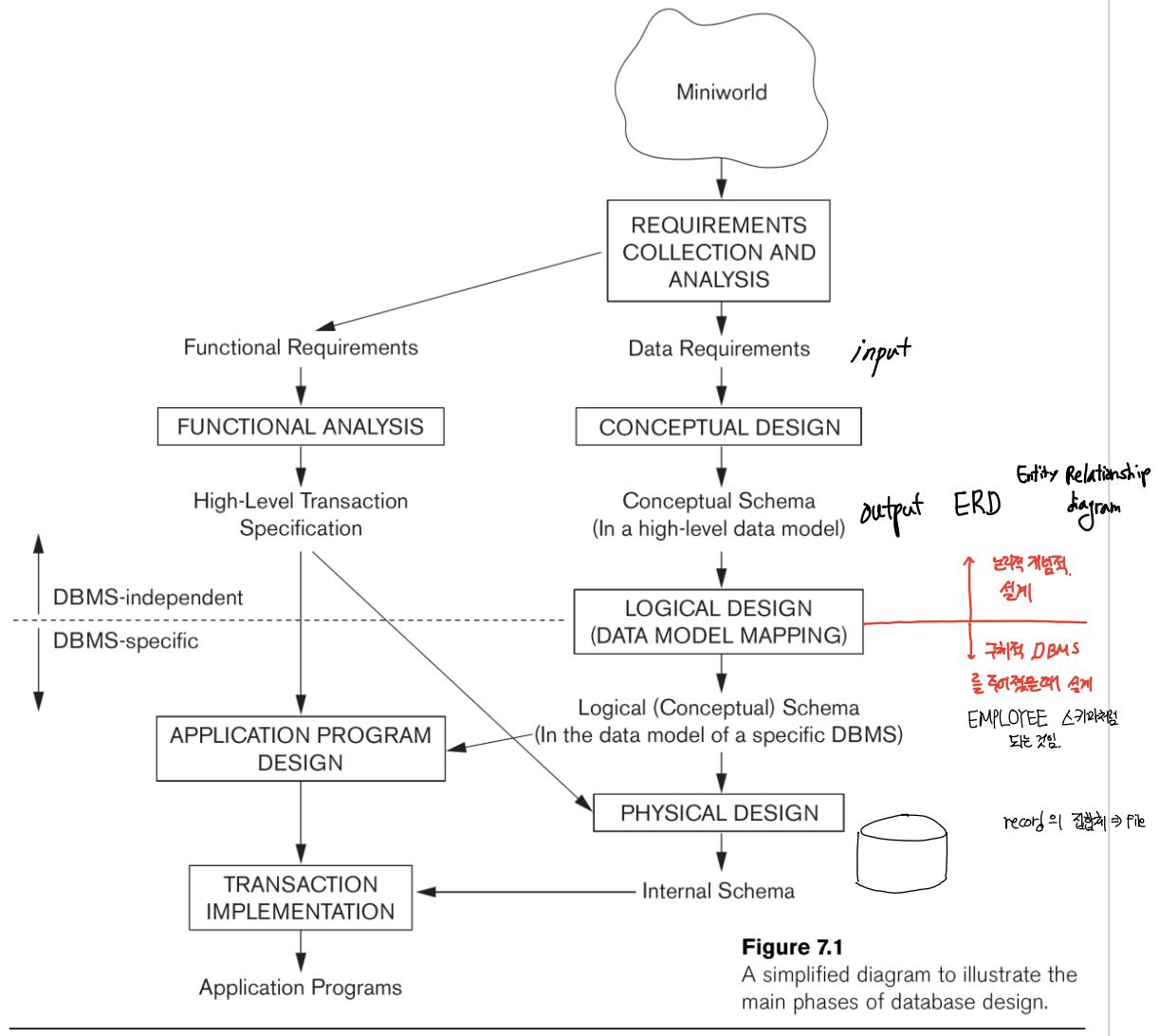
```
(  ID INT,  
  parent_ID INT,  
  FOREIGN KEY (parent_ID)  
  REFERENCES Test1 (ID) ON UPDATE CASCADE );
```

### 1. ON DELETE

- 참조되는 테이블의 값이 삭제될 경우의 동작을 ON DELETE 구문 설정

### 2. ON UPDATE

- 참조되는 테이블의 값이 수정될 경우의 동작을 ON UPDATE 구문 설정



## Data Requirements: COMPANY DB

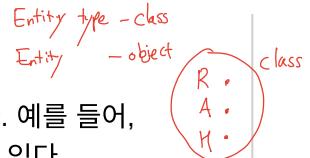
- 회사는 여러 부서(DEPARTMENT)들로 구성. 각 부서는 부서명(name), 번호(number), 부서장을 가진다. 부서장의 부임 날짜(start date)도 유지한다. **부서를 class로 본다면 부서명, 번호, 부서장이 attribute가 된다.**
- 한 부서는 여러 개의 프로젝트(PROJECT)들을 관리한다. 각 프로젝트는 이름(name)과 번호(number)를 가지며 한 곳(location)에 위치한다. **ER 모델에서는 entity type이라고 함.** 설명하는 변수들은 attribute라고 함.
- 각 사원(EMPLOYEE)의 주민등록번호(social security number), 주소(address), 월급(salary), 성별(sex), 생년월일(birthdate)을 저장한다.
- 각 사원은 한 부서에서 근무하며( works for) 여러 프로젝트에 관여한다( work on) .
- 각 사원이 각 프로젝트를 위해 주당 근무 시간을 저장한다.
- 또한, 각 직원의 직속 상사(direct supervisor)도 유지한다. 각 사원은 여러 명의 부양가족(DEPENDENT)들을 가진다. 단속, 부서를 찾아내는 것도 중요. 한 사원은 여러명의 부서들을 가질 수 있다.
- 각 부양가족에 대해 이름(name), 성별(sex), 생년월일(birthdate), 직원과의 관계(relationship)를 저장한다.

## The Concept of Entity Relationship Model

## Entity and Attribute

모델링하려는 대상

- 엔티티는 데이터베이스 내에 표현된 작은 세계에 존재하는 객체 또는 실체이다. 예를 들어, EMPLOYEE John Smith, 연구 DEPARTMENT, ProductX PROJECT 등이 있다.
- 애트리뷰트들은 엔티티를 기술하기 위한 속성들이다. 예를 들어, EMPLOYEE 엔티티는 Name, SSN, Address, Sex, BirthDate를 가질 수 있다.
- 특정한 엔티티는 자신의 각 애트리뷰트에 대해 값을 가진다. 예를 들어, 한 특정한 사원 엔티티는 Name='John Smith', SSN='123456789', Address='731, Fondren, Houston, TX', Sex='M', BirthDate='09-JAN-55'를 값으로 가진다.
- 각 애트리뷰트는 정수, 스트링과 같은 자신에 연계된 값 집합(또는 데이터 타입)을 가진다.



## Attribute Type

- 단순(원자) 애트리뷰트 (Simple, Atomic, Scalar)
  - 각 엔티티는 각 애트리뷰트에 대해 더 이상 나눌 수 없는 값을 가진다. 예를 들면, SSN 또는 Sex.
- 복합(Composite) 애트리뷰트
  - 애트리뷰트는 몇 개의 구성요소들로 이루어 질 수 있다. 예를 들면, Address(Apt#, House#, Street, City, State, ZipCode, Country) 또는 Name(FirstName, MiddleName, LastName). 구성은 그 구성요소가 다시 복합 애트리뷰트인 계층 구조를 형성할 수 있다.
- 다치(Multi-valued) 애트리뷰트
  - 각 엔티티는 애트리뷰트의 값으로 여러 값을 가질 수 있다. 예를 들면, {Color}로 표기되는 CAR의 색 또는 {PreviousDegrees}로 표기되는 STUDENT의 이전 학위가 다치 애트리뷰트이다.
- 일반적으로, 복합 및 다치 애트리뷰트는 몇 단계로 내포될 수 있지만 그런 경우는 흔하지 않다.
  - 예를 들면, STUDENT의 PreviousDegrees는 {PreviousDegrees (College, Year, Degree, Field)}로 표기되는 다치 애트리뷰트일 수 있다.

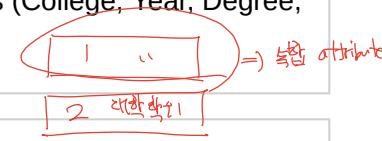


성과 이름을 냄는 것

나는 성을 고으한 값으로  
보았다는 것

주제는 없는데 명의 연관으로 놓았어

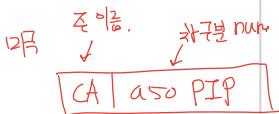
↳ 다치 attribute 예제 ? ?



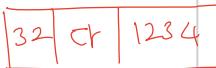
[ 3 대학 학위 ]

## Entity Type and Key Attribute

- 엔티티 타입은 동일한 애트리뷰트들을 갖는 엔티티들의 집합으로 정의함
  - 예를 들면, EMPLOYEE 엔티티 타입 또는 PROJECT 엔티티 타입.
- 키 애트리뷰트
  - 한 엔티티 타입에서 각 엔티티가 유일한 값을 가지는 애트리뷰트를 그 엔티티 타입의 키 애트리뷰트라 한다. 예를 들면, EMPLOYEE의 SSN.
  - 키 애트리뷰트는 복합형일 수 있다. 예를 들어, VehicleTagNumber는 구성요소로 (Number, State)를 가지는 CAR 엔티티 타입의 키이다.
  - 엔티티 타입은 한 개 이상의 키를 가질 수 있다. 예를 들어, CAR 엔티티 타입은 다음의 두 키를 가진다.
    - VehicleIdentificationNumber (popularly called VIN)
    - VehicleTagNumber (Number, State), 또는 license\_plate number.



복합 attribute

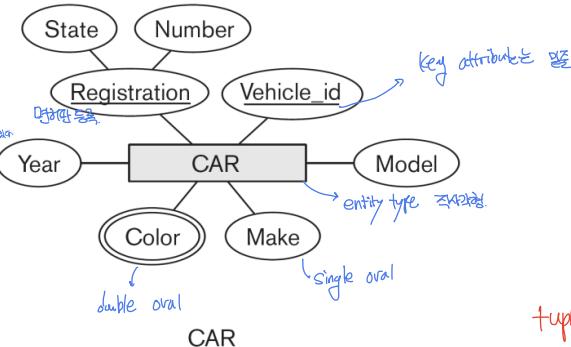
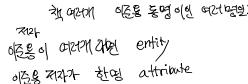


key  
키  
ck  
회원키

## Entity Type : CAR

**Figure 7.7**

The CAR entity type with two key attributes, Registration and Vehicle\_id. (a) ER diagram notation. (b) Entity set with three entities.



(b) Key 는 책의 ISBN

책에서 출판사와 저자는

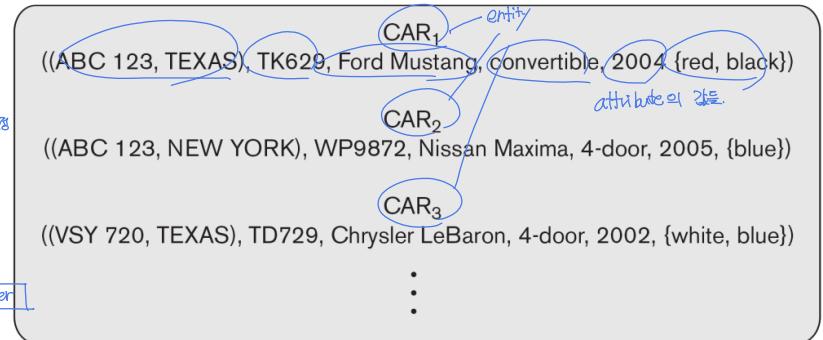
## 각속은 attribute

예 도서관이라면  
값으로 설정할지 entity로 설정  
할지 주도.

(— — — — 이준용)

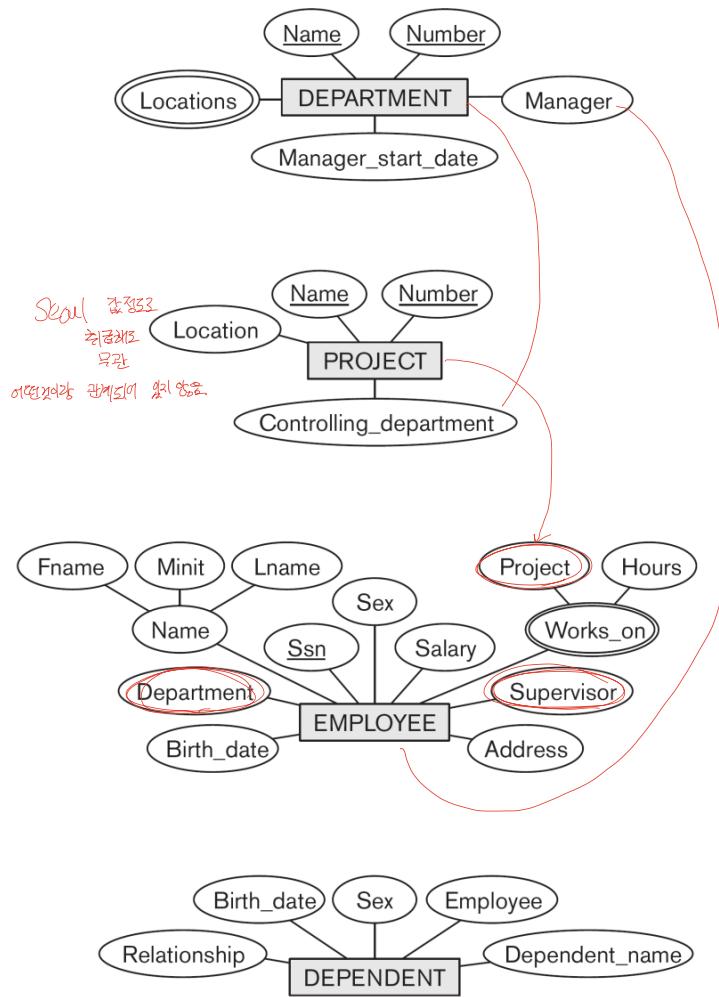


Registration (Number, State), Vehicle\_id, Make, Model, Year, {Color}



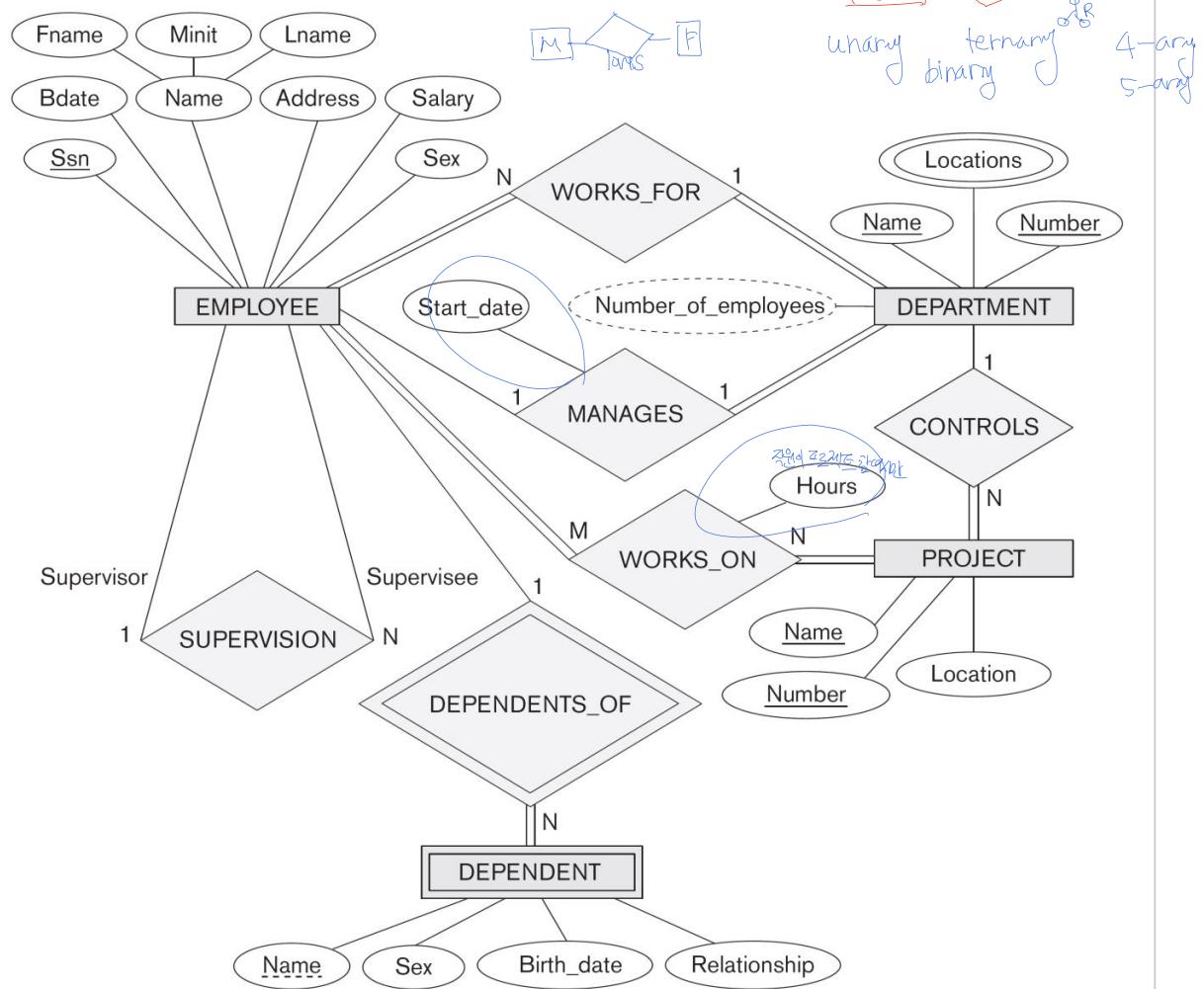
<sup>5</sup>We use a notation for ER diagrams that is close to the original proposed notation (Chen 1976). Many other notations are in use; we illustrate some of them later in this chapter when we present UML class diagrams and in Appendix A.

# Preliminary Design for Company

**Figure 7.8**

Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

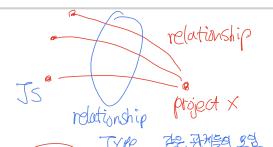
## ER Diagram Entity Type: EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT

**Figure 7.2**

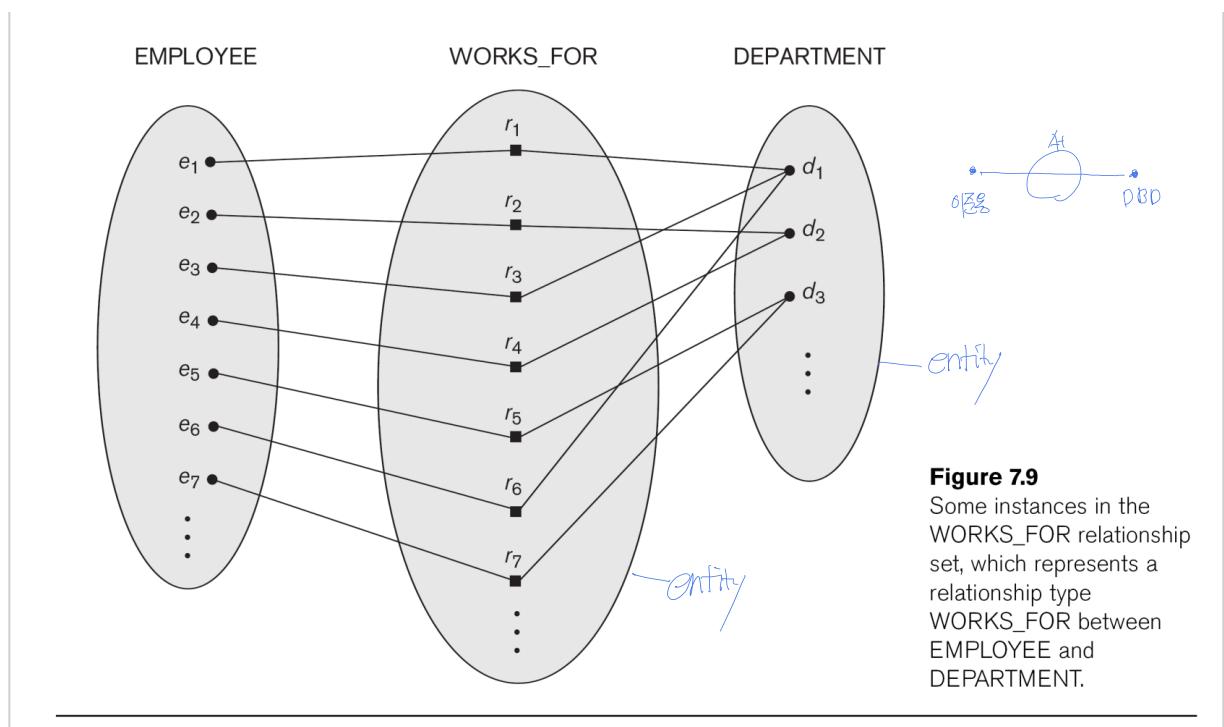
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

## Relationship and Relationship Type

- 관계는 두 개 또는 그 이상의 엔티티들을 특정한 의미로 연관 짓는 것임
  - 예를 들어, EMPLOYEE John Smith는 ProductX PROJECT를 위해 일하며, EMPLOYEE Franklin Wong은 Research DEPARTMENT를 관리한다.
- 같은 형의 관계들은 관계 타입으로 그룹화되어 형이 주어짐
  - 예를 들면, EMPLOYEE들과 PROJECT들이 참여하는 WORKS\_ON 관계 타입, EMPLOYEE들과 DEPARTMENT들이 참여하는 MANAGES 관계 타입.
- 관계 타입의 차수는 참여하는 엔티티 타입의 개수임
  - MANAGES와 WORKS\_ON은 모두 이진 관계이다.
- 두 엔티티 타입들에 대해 한 개 이상의 관계 타입이 있을 수 있다.
  - 예를 들면, EMPLOYEE와 DEPARTMENT 간의 MANAGES와 WORKS\_FOR는 서로 다른 의미와 관계 인스턴스들을 가지는 두 관계들이다.

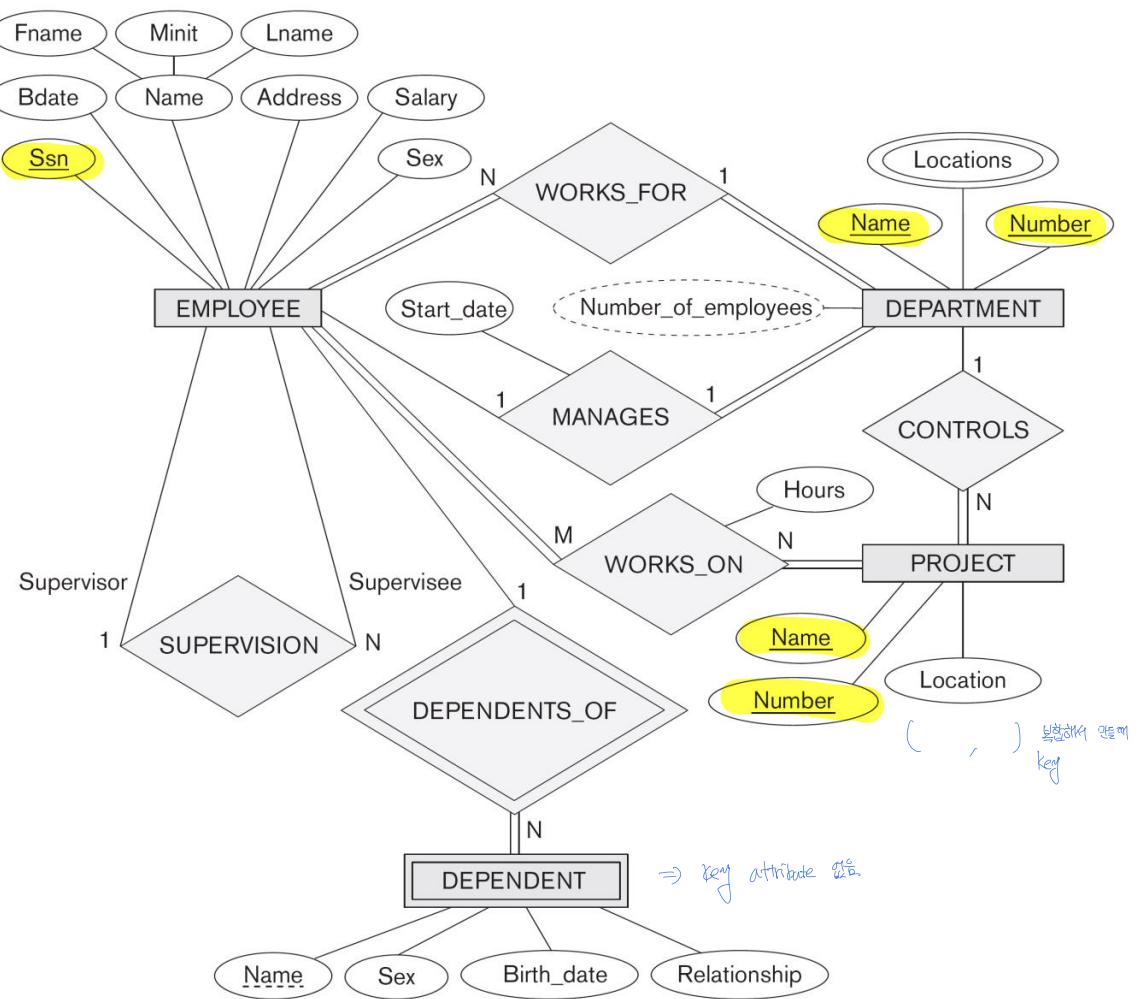


## WORKS\_FOR Relationship between EMPLOYEE and DEPARTMENT



**Figure 7.9**  
Some instances in the WORKS\_FOR relationship set, which represents a relationship type WORKS\_FOR between EMPLOYEE and DEPARTMENT.

## ER Diagram Relationship Type: WORKS\_FOR, MANAGES, WORKS\_ON, CONTROLS, SUPERVISION, DEPENDENTS\_OF

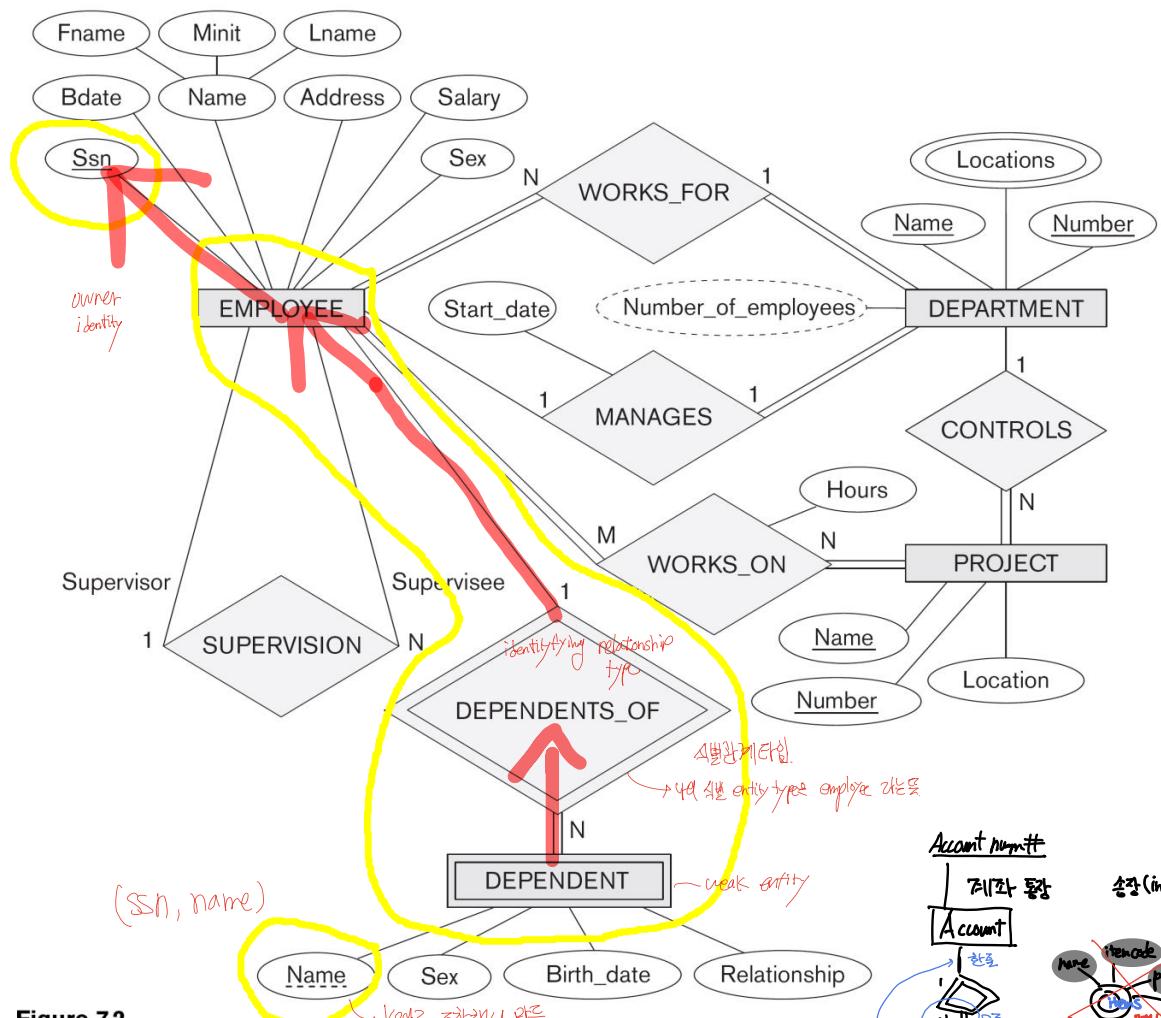
**Figure 7.2**

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

## Weak Entity Type

- 키 애트리뷰트가 없는 엔티티
- 약한 엔티티는 소유 또는 식별 엔티티 타입과의 식별 관계 타입에 참여해야 함
- 엔티티들은 다음의 조합에 의해 식별됨
  - 약한 엔티티 타입의 부분 키
  - 식별 엔티티 타입과 연관된 특정 엔티티
  - 예: DEPENDENT 엔티티는 부양가족의 이름과 생년월일 및 그 부양가족과 연관되는 사원에 의해 식별된다고 가정하자. DEPENDENT는 EMPLOYEE를 식별 엔티티 타입으로 하고 DEPENDENTS\_OF의 식별 관계 타입으로 연관되는 약한 엔티티 타입이다.

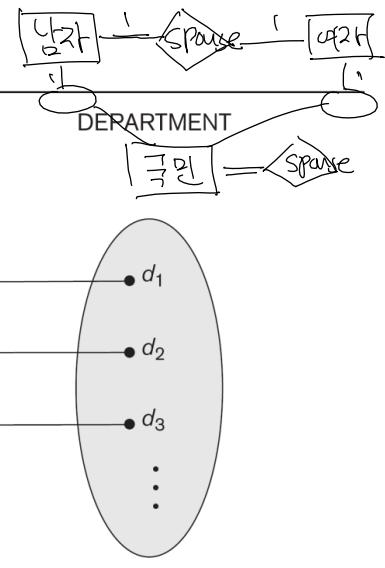
## Weak Entity Type DEPENDENT and IdentifyingRelationship DEPENDENTS\_OF



## Constraints on Relationship

- 관계 탑입에 대한 제약조건:** 카디널리티 비율 제약조건과 참여 제약조건 두 가지가 있음
  - 카디널리티 비율 제약조건**  $\text{Cardinality} = \text{length.size}$ 
    - 최대 카디널리티**
      - 1:1, 1:N 또는 N:1, M:N
    - 최소 카디널리티(참여 제약조건 또는 존재 종속 제약조건)**
      - 0 (선택적 참여, 존재 종속이 아님)  $\Rightarrow$  어떤 부속은 참여하고 있지 않고 있는 뜻
      - 1 또는 그 이상 (의무적, 존재 종속) mandatory
  - 참여제약 조건**
    - 관계에 참여하는 엔티티 탑입의 일부 엔티티만 관계에 참여함
    - 예를 들어, Employee 중에서 일부만 Department와의 manager 관계에 참여함
    - 전체 (또는 존재 종속성(existential dependence), total, mandatory) 또는 부분(partial, optional) 참여
    - 전체 참여의 경우 연결된 선을 이중선으로 표기함. EM

## One-to-one (1:1) Relationship



${}^9N$  stands for any number of related entities (zero or more).

## One-to-many (1:N) Relationship

1:1      M:N (o)  
1: Many    M:M (x)

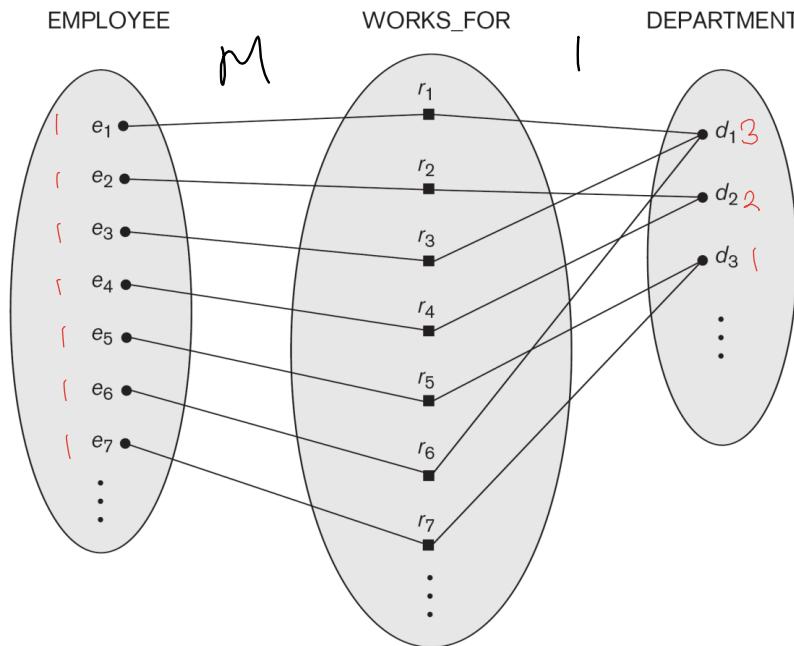
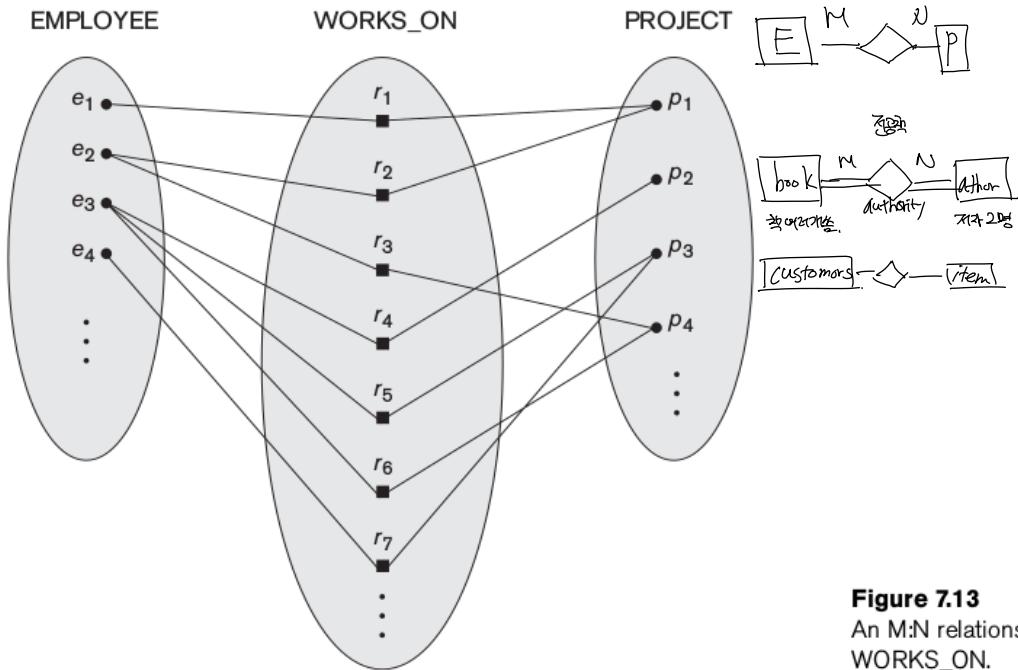


Figure 7.9

Some instances in the WORKS\_FOR relationship set, which represents a relationship type WORKS\_FOR between EMPLOYEE and DEPARTMENT.

## Many-to-many (M:N) Relationship

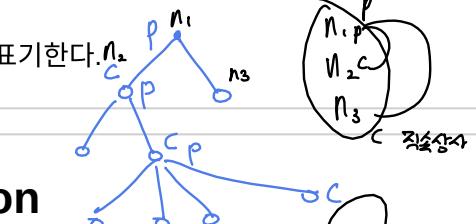
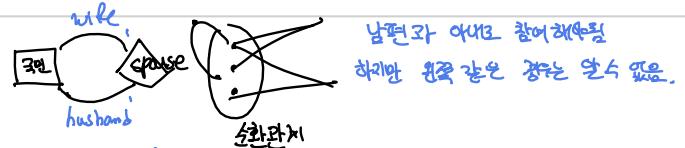


**Figure 7.13**  
An M:N relationship,  
WORKS\_ON.

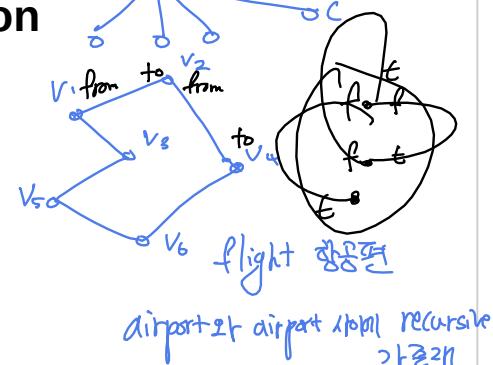
모든 사람이 결혼해야 되는게 아니고 한 줄

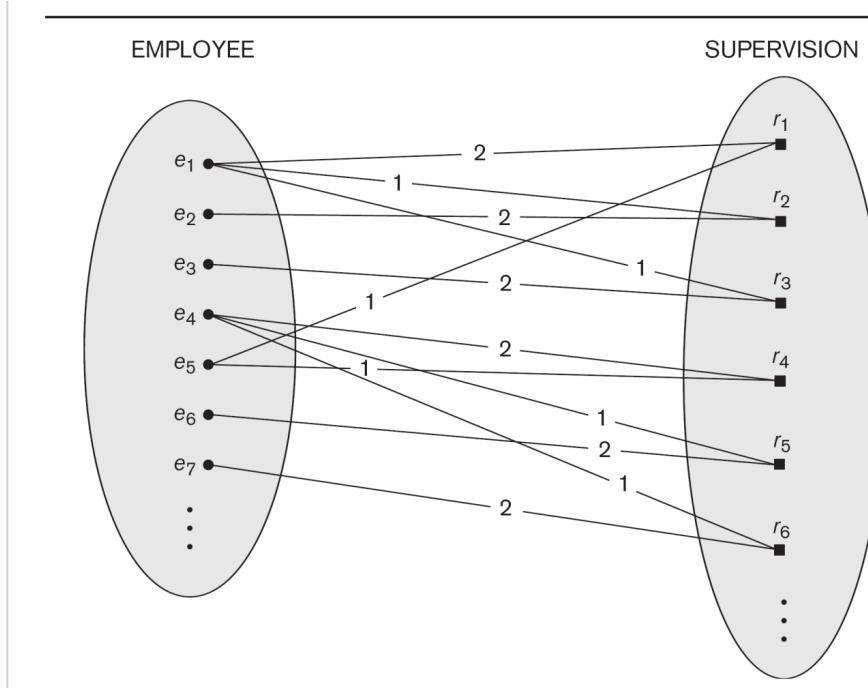
## Recursive Relationship

- 순환 관계
- 한 개의 엔티티 타입이 어떤 관계 타입에 서로 다른 역할로 중복 참여하는 경우를 의미함
- 예를 들어, EMPLOYEE (감독자 또는 상사의 역할로)와 EMPLOYEE (부하 또는 근로자의 역할로) 간의 SUPERVISION 관계
- 다음 page 그림에서 1이라고 표기된 첫 번째 역할(감독자 또는 상사의 역할) 참여와 2라고 표기된 두 번째 역할 (부하 또는 근로자의 역할) 참여.
- ER 다이어그램에서는 참여를 구분하기 위해 역할 이름을 표기한다.



## Recursive Relationship: Supervision



**Figure 7.11**

A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

## Attributes of Relationship Type

- 관계 타입도 앤트리뷰트들을 가질 수 있다.
- 예를 들면, 관계 타입 WORKS\_ON의 Hours 속성의 값은 각 관계 인스턴스에 대해 EMPLOYEE 가 PROJECT를 위해 주당 일한 시간을 나타낸다.

## Min-Max Notation on Constraints

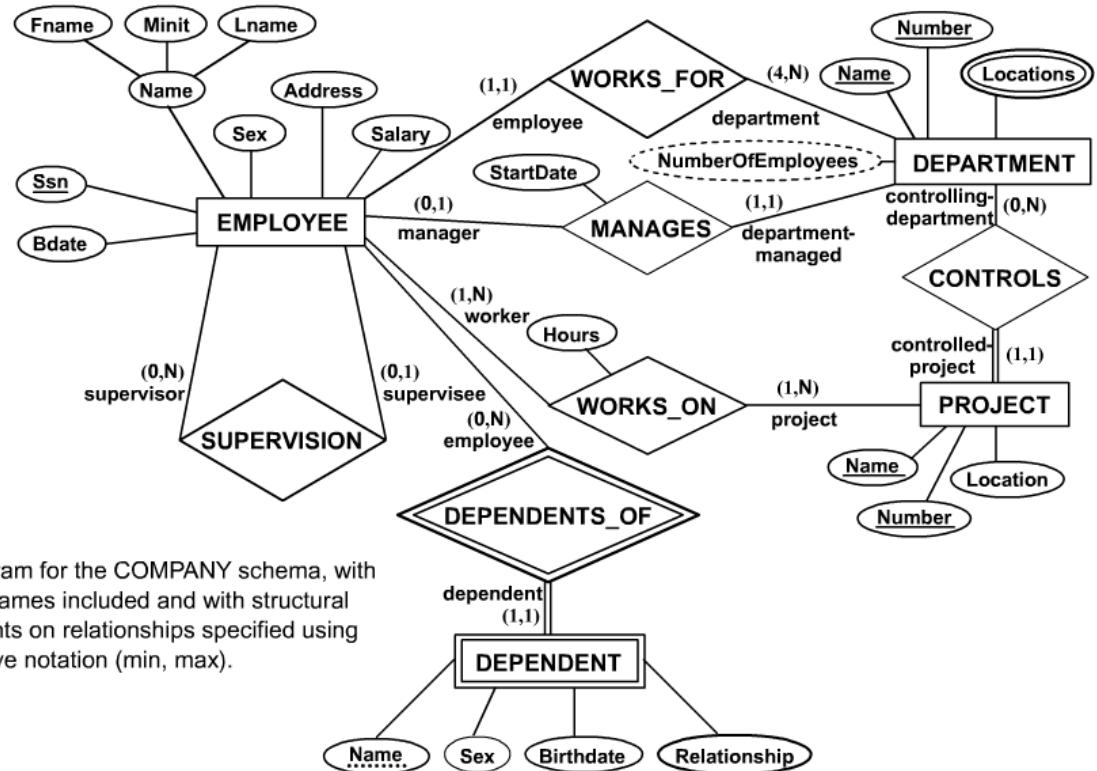
- 관계 타입 R에서 엔티티 타입 E의 각 참여를 명시
- E 내의 각 엔티티 e가 적어도 최소값 만큼 그리고 최대 최대값만큼 R 내의 관계 인스턴스들과 참여 함을 나타냄
- 디폴트(제약조건 무): 최소값=0, 최대값=n
- 반드시, 최소값 ≤ 최대값, 최소값 ≥ 0, 최대값 ≥ 1
- 작은-세계 제약조건으로 부터 유도됨

예:

- 부서는 정확히 한 명의 부서장을 가지며 사원은 최대 한 개의 부서를 관리할 수 있다.
  - MANAGES의 EMPLOYEE의 참여에 (0,1)로 표기
  - MANAGES의 DEPARTMENT의 참여에 (1,1)로 표기
- 사원은 정확히 한 부서에서 일 할 수 있으며 부서는 몇 명의 사원이라도 둘 수 있다.
  - WORKS\_FOR의 EMPLOYEE의 참여에 (1,1)로 표기
  - WORKS\_FOR의 DEPARTMENT의 참여에 (0,n)로 표기

## Min-Max Notation of ER Diagram

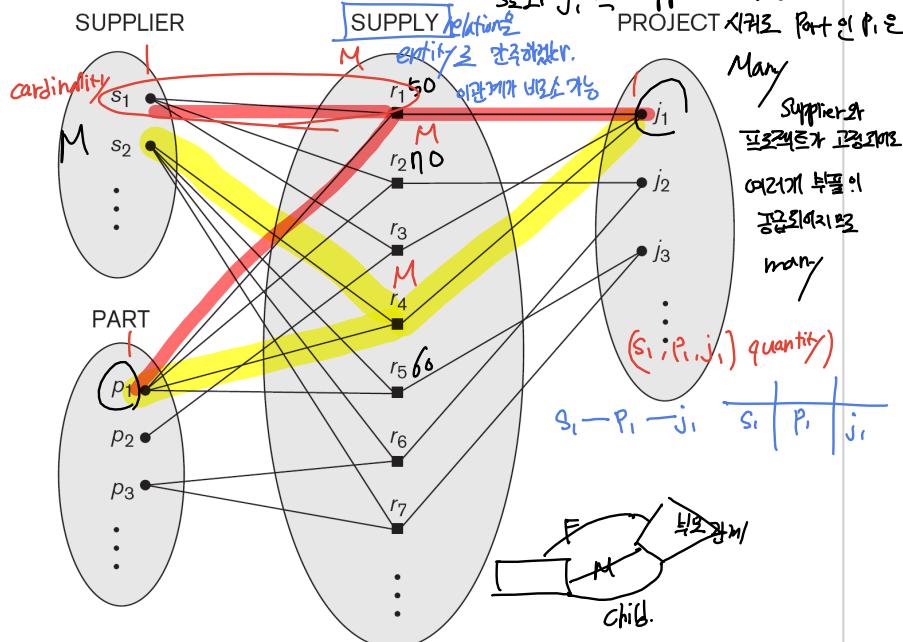
## Alternative ER Notations



## N-ary Relationship

- 2진관계 (Binary)
  - 차수가 2인 관계 타입을 2진 관계라 한다.
- 3진(Ternary) 관계, n-ary 관계
  - 차수가 3인 관계 타입을 3진 관계라 하고, 차수가 n인 관계를 n-ary 관계라 한다.
- “n-ary 관계”와 “n개의 이진 관계들” 사이의 차이
  - 일반적으로, n-ary 관계는 n개의 이진 관계들과 같지 않다.
- **cardinality는 하나의 entity type과 다른 나머지 entity type과의 cardinality를 고려하면, 이진 관계의 cardinality와 같아진다.**

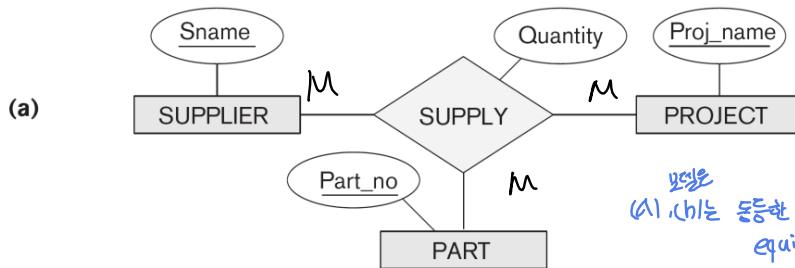
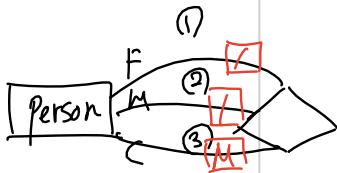
## Ternary Relationship

ex)  $P_1, j_1$  은 고정시킬 때 $S_1, j_1$  즉 Supplier가 Project를 향해PROJECT 시점으로 Part인  $P_1$  을

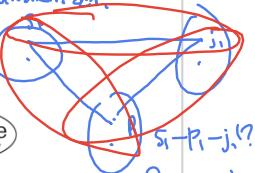
**Figure 7.10**  
Some relationship instances in the SUPPLY ternary relationship set.



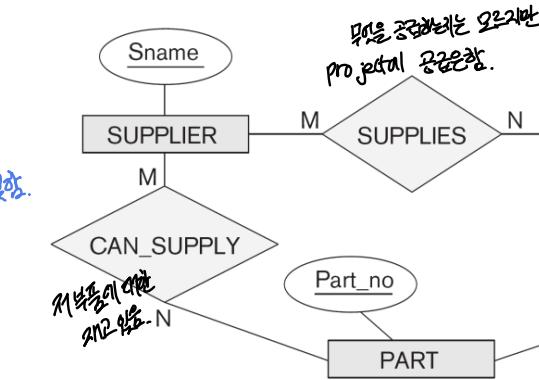
## Ternary Relationship and Three Binary Relationships



모임은  
(a), (b)는 동등한 능력을 가진다?  
equivalent인가?

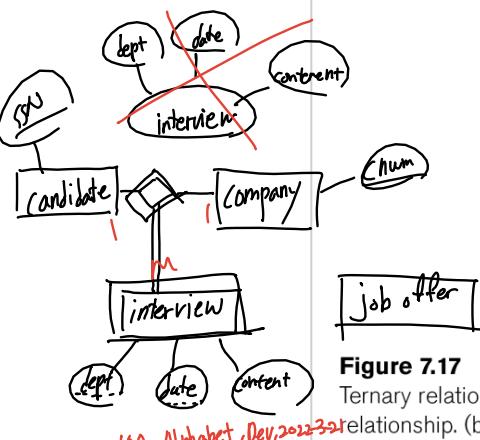


(b)  
모델링 잘못함.



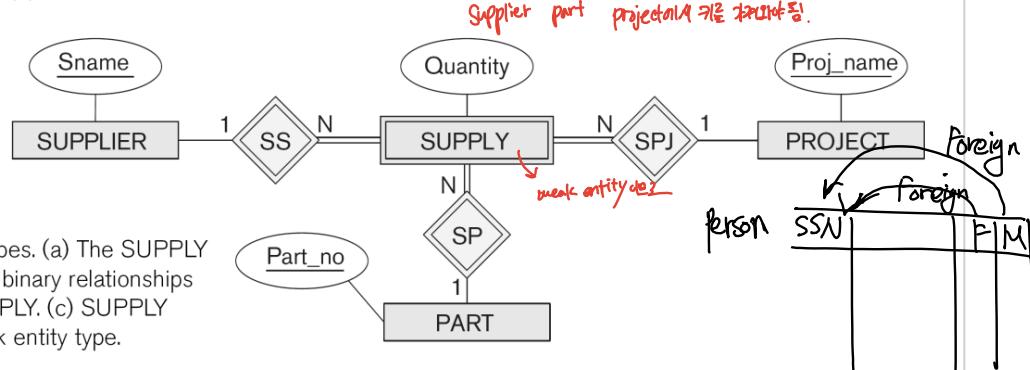
모임을 공급하는 모집과는  
project 공급과는.

모집과는  
Project를 공급하면  
이 브로드 사용해야함



**Figure 7.17**  
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

(c) 결론



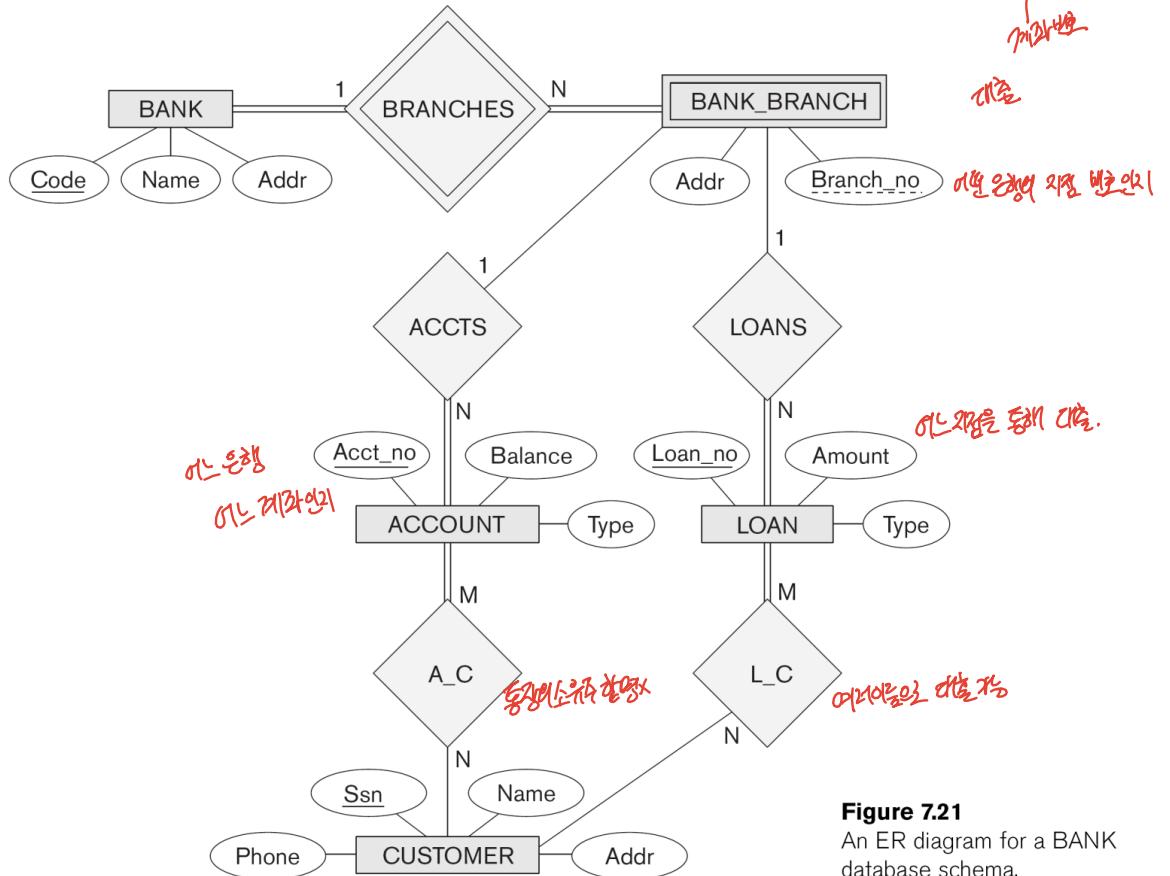
Supplier part project에서 키를 가져와야됨.

weak entity type

## Limitations of ER Model

- 원래 ER 모델은 객체 모델링 기법에서 제공하는 추상화(특수화/일반화) 기능을 지원하지 못함
- 확장 엔티티-관계(EER, Enhanced ER) 모델
  - 집합-부분집합 관계를 포함.
  - 특수화/일반화 계층구조를 포함 .

## ER Diagram for Bank Database



**Figure 7.21**  
An ER diagram for a BANK database schema.

## LAB: Requirements for University DB

- Draw ER Diagram

1221 03/14

- DBD 2022 | (1)
- DBD 2022 | (2)
- (a) The university keeps track of each student's name, student number, social security number, current address and phone, permanent address and phone, birthdate, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address, and to the student's last name. Both social security number and student number have unique values for each student.
  - (b) Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department.
  - (c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.
  - (d) Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ...; up to the number of sections taught during each semester.
  - (e) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, 4 for F, D, C, B, A, respectively).

## Example ER Diagram

**Figure 7.20**

An ER diagram for an AIRLINE database schema.

