제 7 장
개체–관계(ER) 모델을 사용한
데이타 모델링

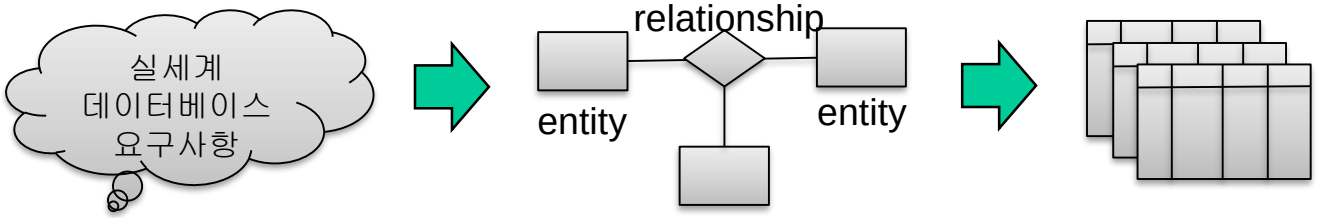
Fundamentals of Database Systems
R. A. Elmasri and S. B. Navathe

목 차

- 데이타베이스 설계를 위한 고수준 개념적 데이타 모델의 사용
- 데이타베이스 응용 예제
- 개체 타입, 개체 집합, 애트리뷰트, 키
- 관계, 관계 타입, 역할, 구조적 제약조건
- 약한 엔티티 타입(Weak Entity Type)
- COMPANY 데이타베이스에 대한 ER 설계의 개선
- ER 다이어그램, 명명에 관한 규칙, 설계에 관한 사항
- 요약

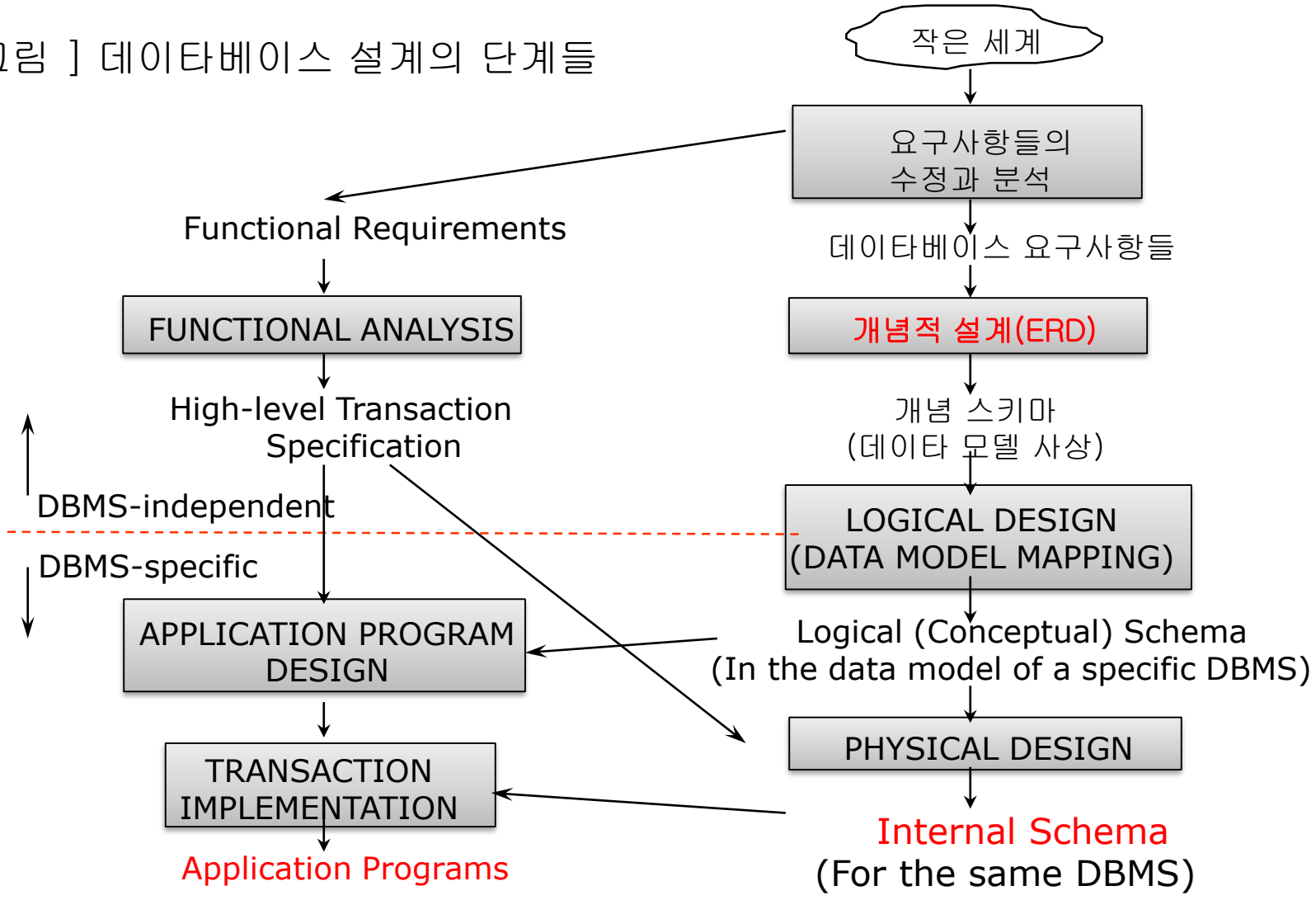
개 요

- 개념적 모델링은 개체-관계(Entity-Relationship: ER) 다이어그램을 사용하며, 데이타베이스 설계에서 중요한 단계임 : 건물의 설계도 역할
- 조직의 데이터베이스 요구사항을 ERD (ER Diagram)로 모델링한 후 테이블로 변환하여 DB 구축을 완성함



DB 설계를 위한 고수준의 개념적 데이타 모델 사용

[그림] 데이타베이스 설계의 단계들



데이타베이스 응용 예제

COMPANY 데이타베이스의 작은 세계

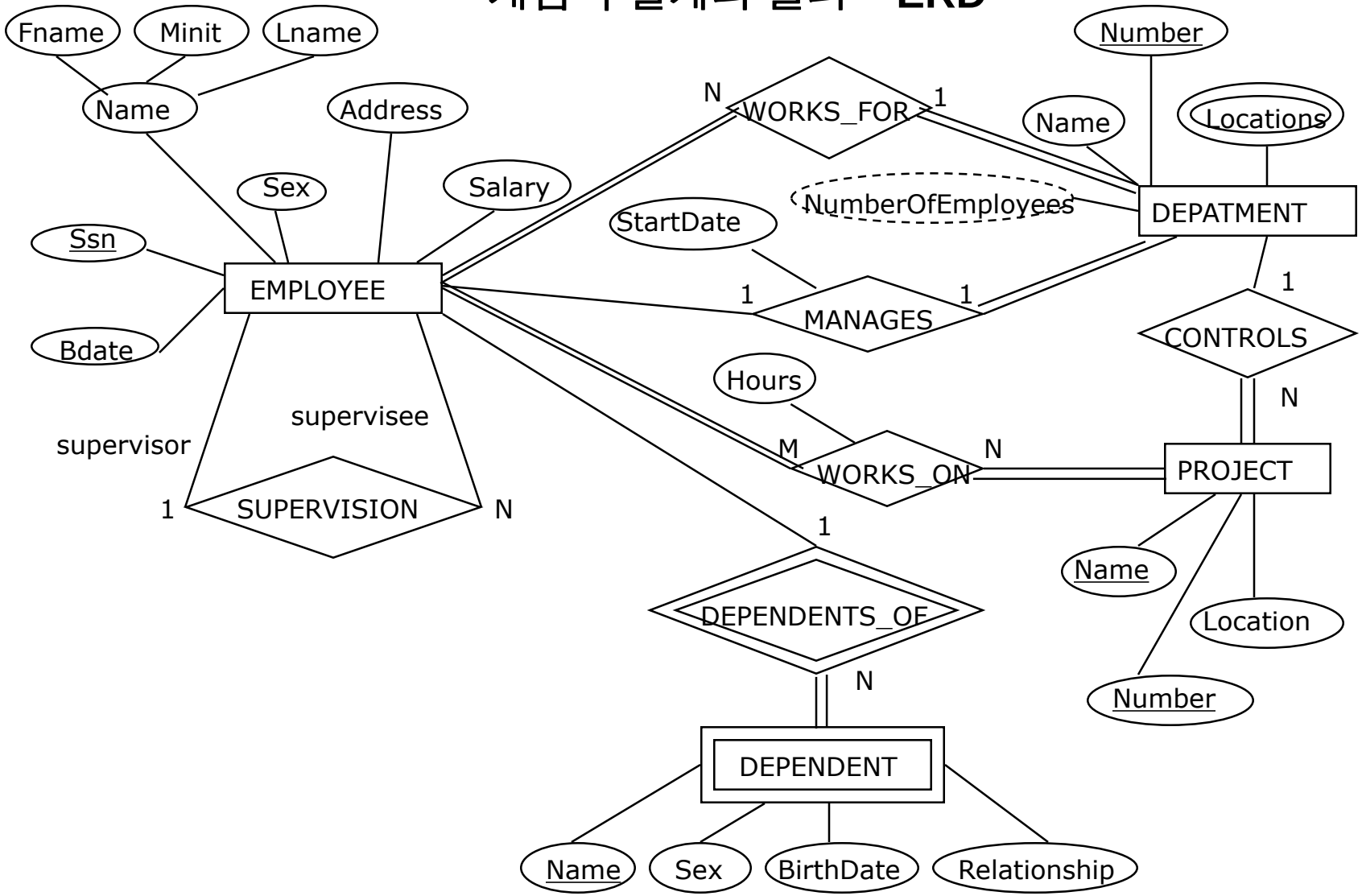
1. 회사는 여러 부서들로 구성된다. 각 부서마다 고유한 이름, 고유한 번호, 부서를 관리하는 특정 사원이 있다. 사원이 부서를 관리하기 시작한 날짜도 유지한다. 한 부서는 여러 위치에 있을 수 있다.

2. 한 부서는 여러 프로젝트들을 관리한다. 각 프로젝트는 고유한 이름, 고유한 번호, 한 개의 위치를 가진다.

3. 각 사원에 대해서 이름, 사회보장번호, 주소, 급여, 성별, 생년월일을 저장한다. 한 사원은 한 부서에 속하지만, 여러 프로젝트들에 관여할 수 있다. 한 사원이 관여하는 프로젝트들은 그 사원이 소속된 부서가 관리하는 프로젝트가 아니어도 무방하다. 반드시 한 부서의 각 사원이 각 프로젝트를 위해 일하는 주당 근무 시간을 기록한다. 또한 각 사원의 직속 상사도 유지한다.

4. 보험 목적을 위해서 각 사원의 부양 가족들을 기록한다. 각 부양 가족에 대해서 이름, 성별, 생년월일, 사원과의 관계를 기록한다.

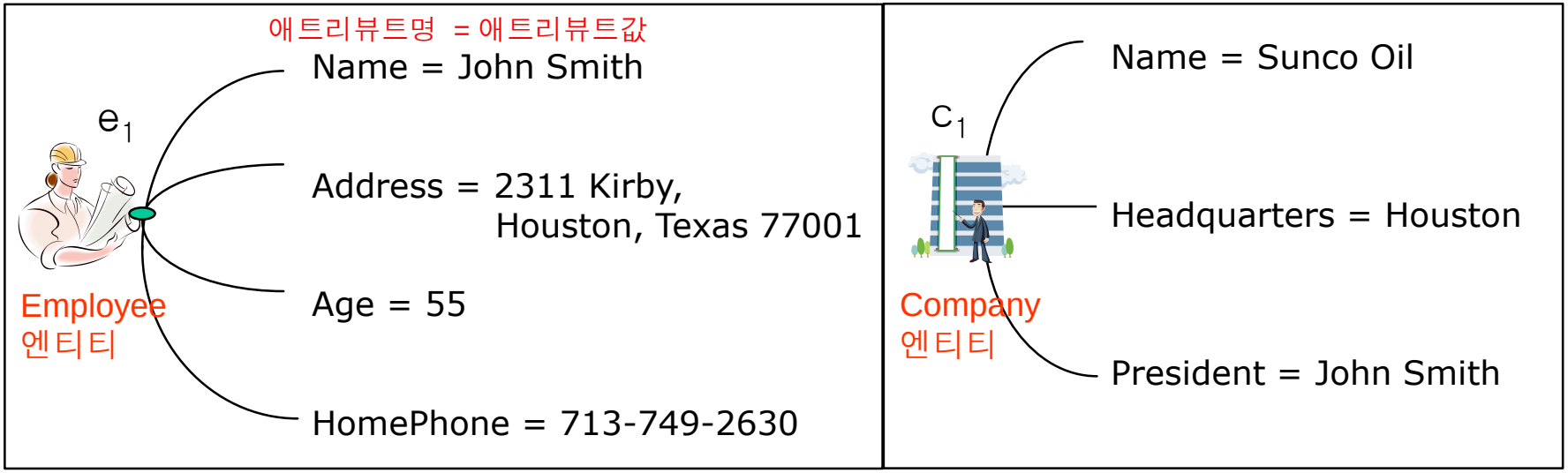
개념적 설계의 결과 – ERD



[그림] COMPANY 데이타베이스를 위한 ER 스키마 다이어그램

개체, 개체 타입, 애트리뷰트, 키

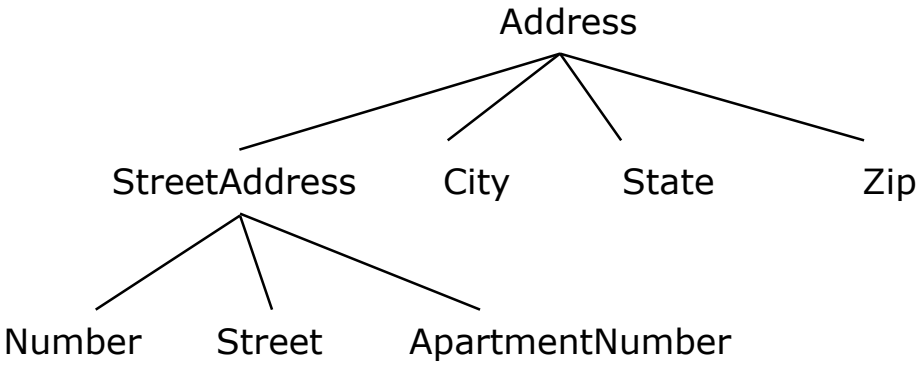
- ER 모델에서는 실세계의 데이터를 엔티티(개체)와 관계로 모델링함
- 엔티티와 애트리뷰트
  - 엔티티: 실세계에서 독립적으로 존재하는 실체로써 애트리뷰트 (속성)을 가짐
  - 애트리뷰트: 엔티티를 기술하는 속성



[그림] 두 개의 엔티티(직원 e_1과 회사 c_1)와 애트리뷰트 및 값

개체, 개체 집합, 애트리뷰트, 키

- 애트리뷰트 유형
  - 단순 애트리뷰트
  - 복합 애트리뷰트



[그림] 복합 애트리뷰트의 계층구조

  - 단일 값 애트리뷰트
  - 다치 애트리뷰트
  - 저장된 애트리뷰트
  - 유도된 애트리뷰트 : *생년월일 => 나이*

개체, 개체 집합, 애트리뷰트, 키

- 널 값
  - 두가지 의미로 사용됨; '값이 없음' / '값을 모름' (존재하는데 그 값을 모르는 경우와 존재여부를 모르는 경우)

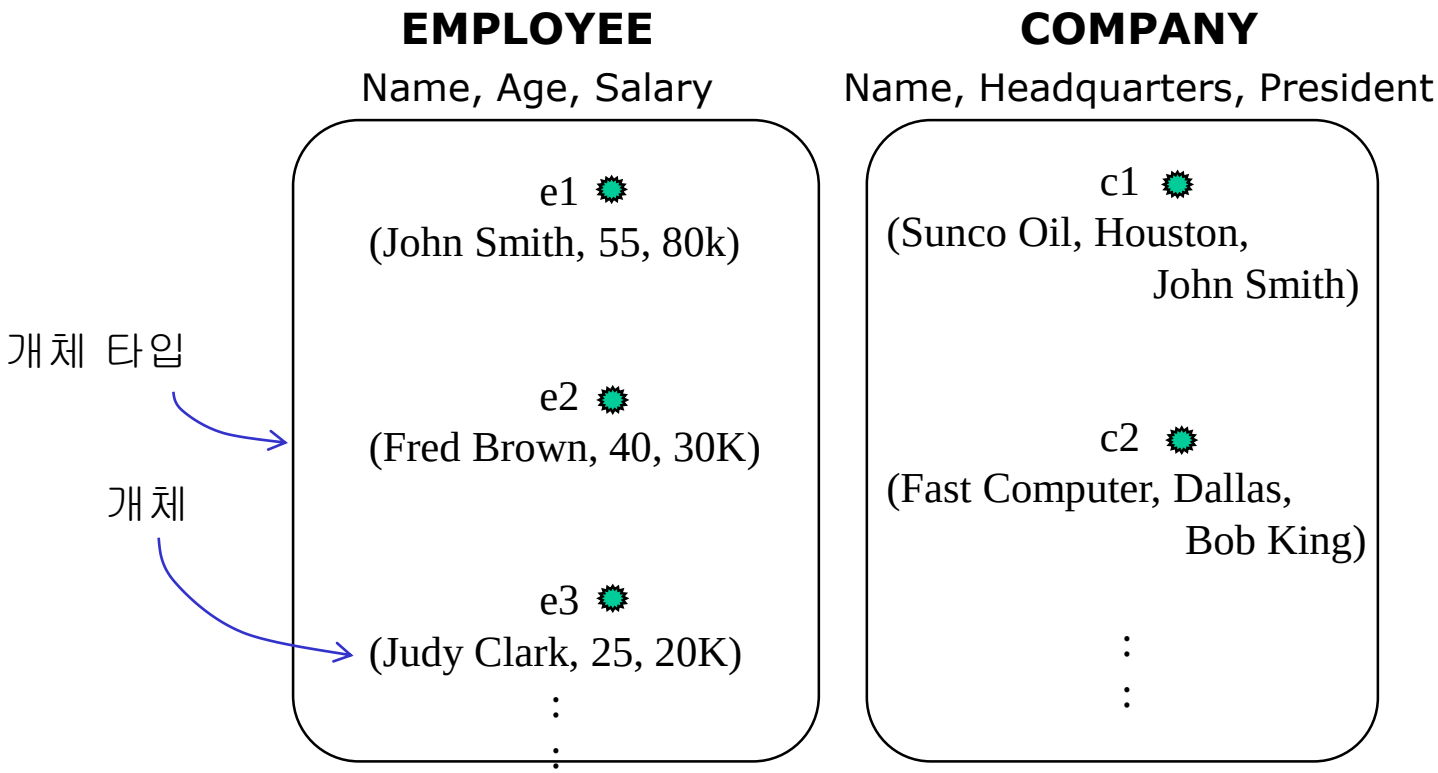
- 복합 (composite) 애트리뷰트의 표현

```
{AddressPhone({Phone  (AreaCode, PhoneNumber)},
            Address(StreetAddress(Number, Street,ApartmentNumber),
                    City, State, Zip
                   )
           )
 }
```

[그림] 다치와 복합 구성요소를 가지는 복합 애트리뷰트 *AddressPhone*

개체, 개체 집합, 애트리뷰트, 키

- 개체 타입 (엔티티 타입)
  - 동일한 유형의 개체 집합 혹은 개체들의 공통정보
  - 예: 개체 타입 Employee와 Company 및 개체들



개체, 개체 타입, 애트리뷰트, 키

[그림] CAR 개체 타입

CAR – Entity Type

Registration(RegistrationNumber, State), VehicleID, Make, Model, Year, {Color} – Attribute

((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 1989, {red,black})

((ABC 123, NEW YORK), WP9872, Nissan Sentra, 2-door, 1992, {blue})

((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 1993, {white, blue})

:

개체, 개체 타입, 애트리뷰트, 키

- **키 애트리뷰트**
  - 각 엔티티마다 서로 다른 값을 가지는 애트리뷰트

    예: PERSON 엔티티 타입의 키 애트리뷰트는 주민등록번호
  - 복합키
    - 두 개 이상의 애트리뷰트들이 모여서 하나의 키 애트리뷰트 역할을 하는 키
- **값 집합(도메인)**
  - 각 엔티티에서 애트리뷰트가 가질 수 있는 값들의 집합

    예: EMPLOYEE의 Age 애트리뷰트의 도메인은 (16~70)

개체, 개체 타입, 애트리뷰트, 키

COMPANY 데이타베이스에 대한 초기 개념적 설계

1. 엔티티 타입 DEPARTMENT는 Name, Number, Location, Manager, ManagerStartDate 애트리뷰트를 가진다. Location은 유일한 다치 애트리뷰트이다. Name과 Number는 각 부서마다 고유하기 때문에 각각 키 애트리뷰트로 지정할 수 있다.
2. 엔티티 타입 PROJECT는 Name, Number, Locations, ControllingDepartment 애트리뷰트들을 가진다. Name과 Number가 각각 키 애트리뷰트이다.
3. 엔티티 타입 EMPLOYEE는 Name, SSN, Sex, Address, Salary, BirthDate, Department, Supervisor 애트리뷰트들을 가진다. 사용자가 사원 Name의 각 구성요소(FirstName, MiddleInitial, LastName)와 Address의 각 구성요소를 참조할 것인지의 여부를 알기 위해서 사용자와 다시 협의해야 한다.
4. 엔티티 타입 DEPARTMENT는 Employee, DependentName, Sex, BirthDate, Relationship(사원과의 관계) 애트리뷰트들을 가진다.

DEPARTMENT

Name, Number, {Locations}, Manager, ManagerStartDate

PROJECT

Name, Number, Location, ControllingDepartment

EMPLOYEE

Name(FName, MInit, LName), SSN, Sex, Address, Salary,
BirthDate, Department, Supervisor, {WorksOn (Project, Hours)}

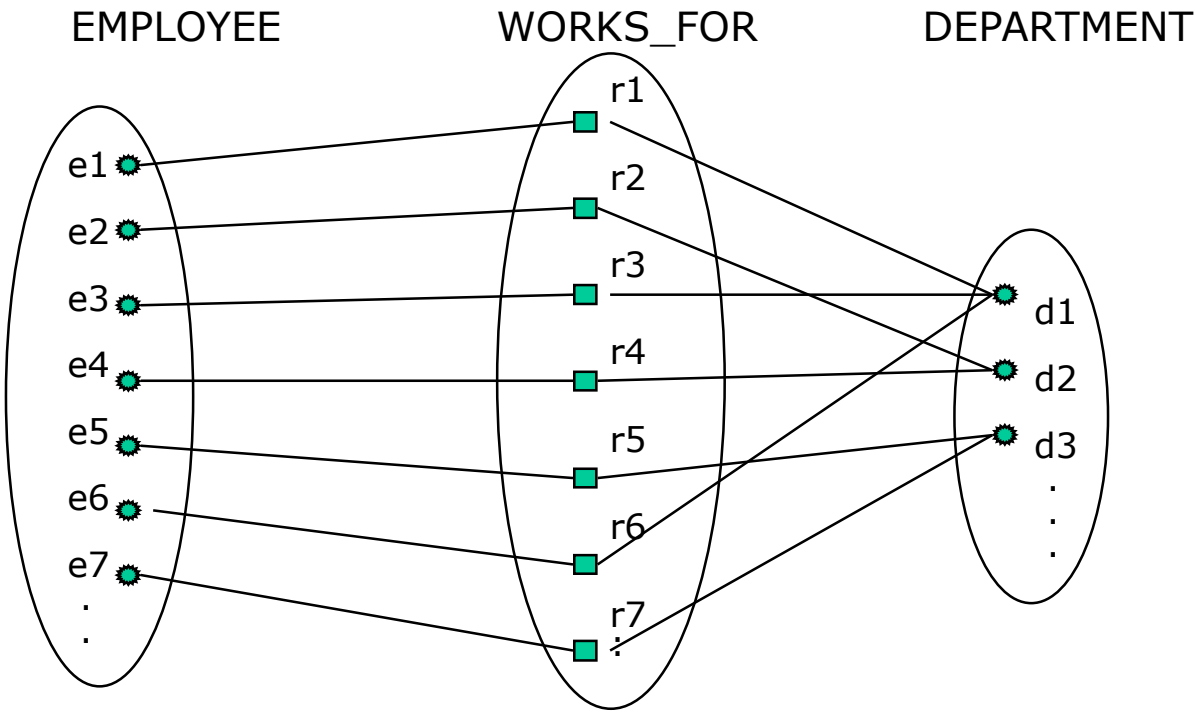
DEPENDENT

Employee, DependentName, Sex, BirthDate, Relationship

[그림] company 데이타베이스를 위한 엔티티 타입들의 초기 설계

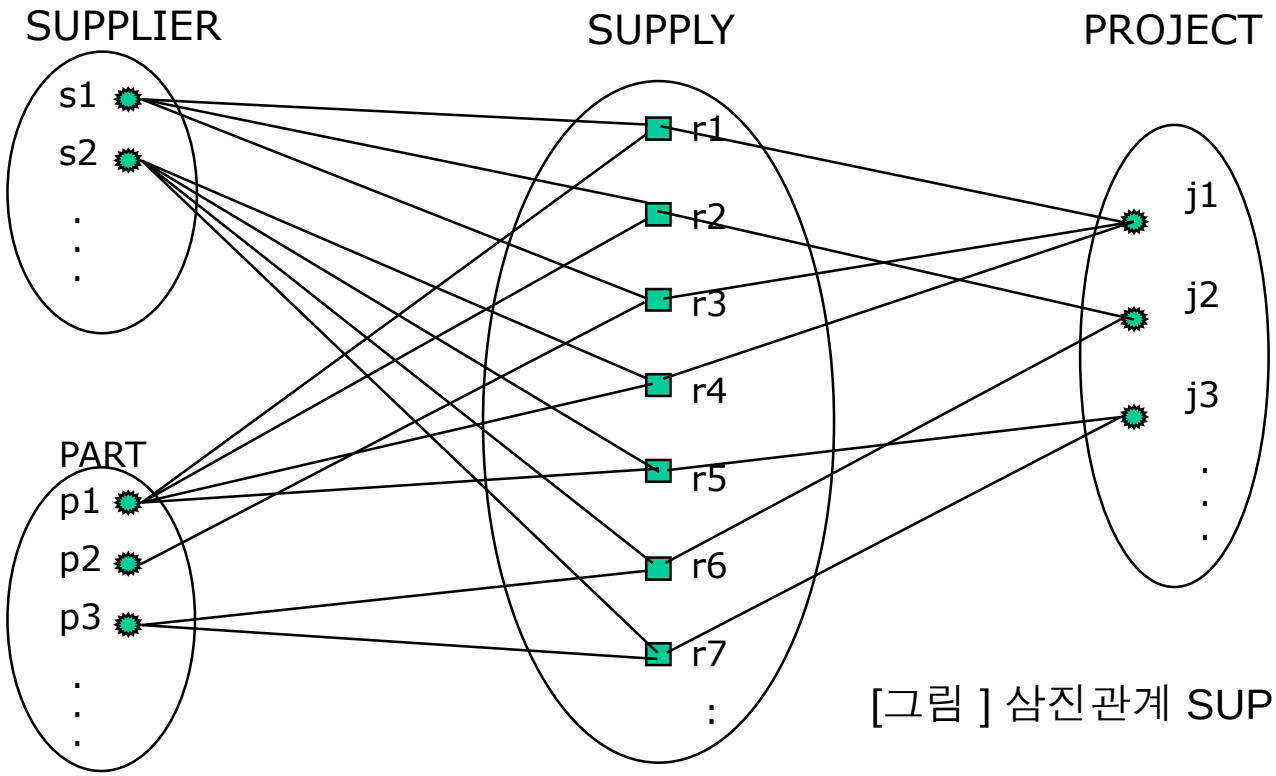
관계, 관계 타입, 역할, 구조적 제약조건

- 관계 타입과 관계 인스턴스
  - 관계 타입 (관계 집합) R은 엔티티 간의 연관 (관계 인스턴스)들의 집합임 : 그림에서 WORKS_FOR
  - [그림] 관계 WORKS_FOR에서 관계 인스턴스 : 7 개 (r1 ~ r7)



관계, 관계 타입, 역할, 구조적 제약조건

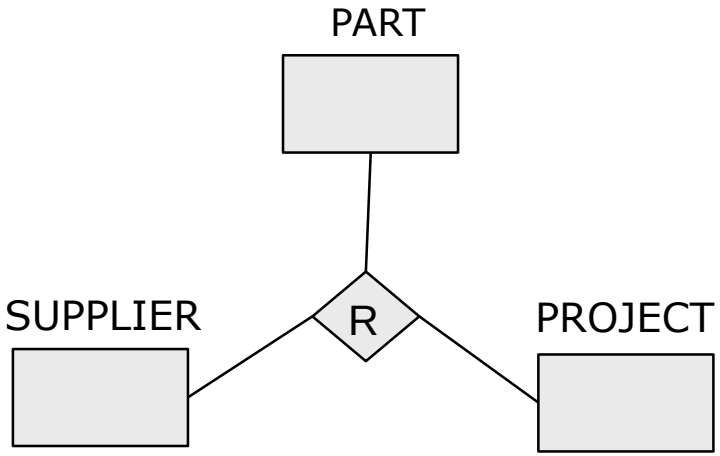
- 관계 타입의 차수(degree): 관계에 참여하고 있는 엔티티 타입의 수
  - 이진(binary) 차수: [그림]의 WORKS_FOR 관계
  - 3진(ternary) 차수 : [그림] 3진 관계 SUPPLY

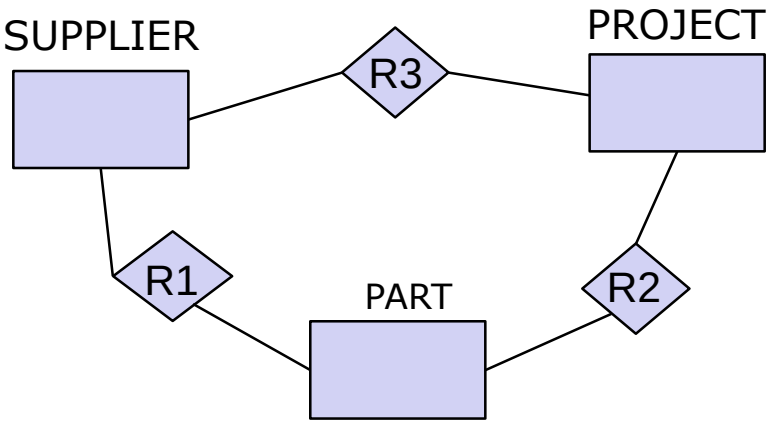


[그림] 삼진관계 SUPPLY

관계, 관계 타입, 역할, 구조적 제약조건

- 3진관계와 2진관계 – 차이





R

S1	P1	J1
S1	P1	J2
S2	P1	J1
S3	P2	J1
	…	

R1

S1	P1

R2

P1	J1

=> R3

S1	J1

??

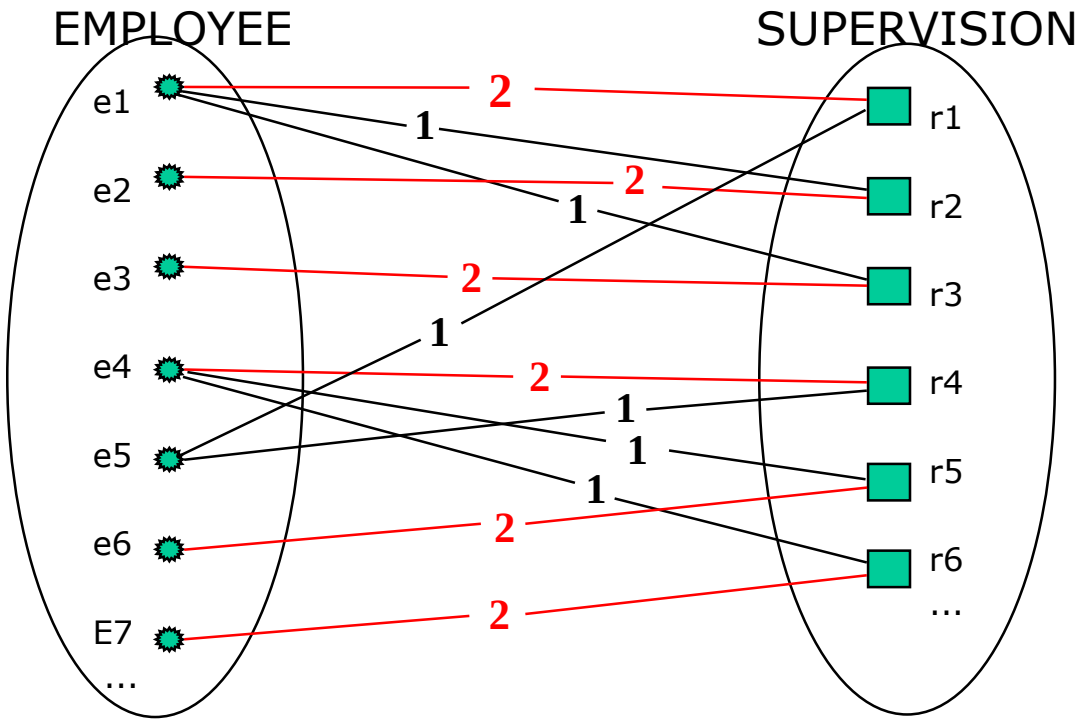
3진관계는 3자간의 관계를 모델링함
2진관계는 2자간의 관계를 모델링하므로
(S1, P1)과 (P1, J1)가 존재해도 (S1, J1)이 존재하지 않을 수 있음

관계, 관계 타입, 역할, 구조적 제약조건

- 애트리뷰트로서의 관계: 관계는 참여 엔티티 타입의 속성으로 볼 수도 있음
  예: (EMPLOYEE의 Department 또는 DEPARTMENT의 Employees)
- 역할과 순환적(recursive) 관계

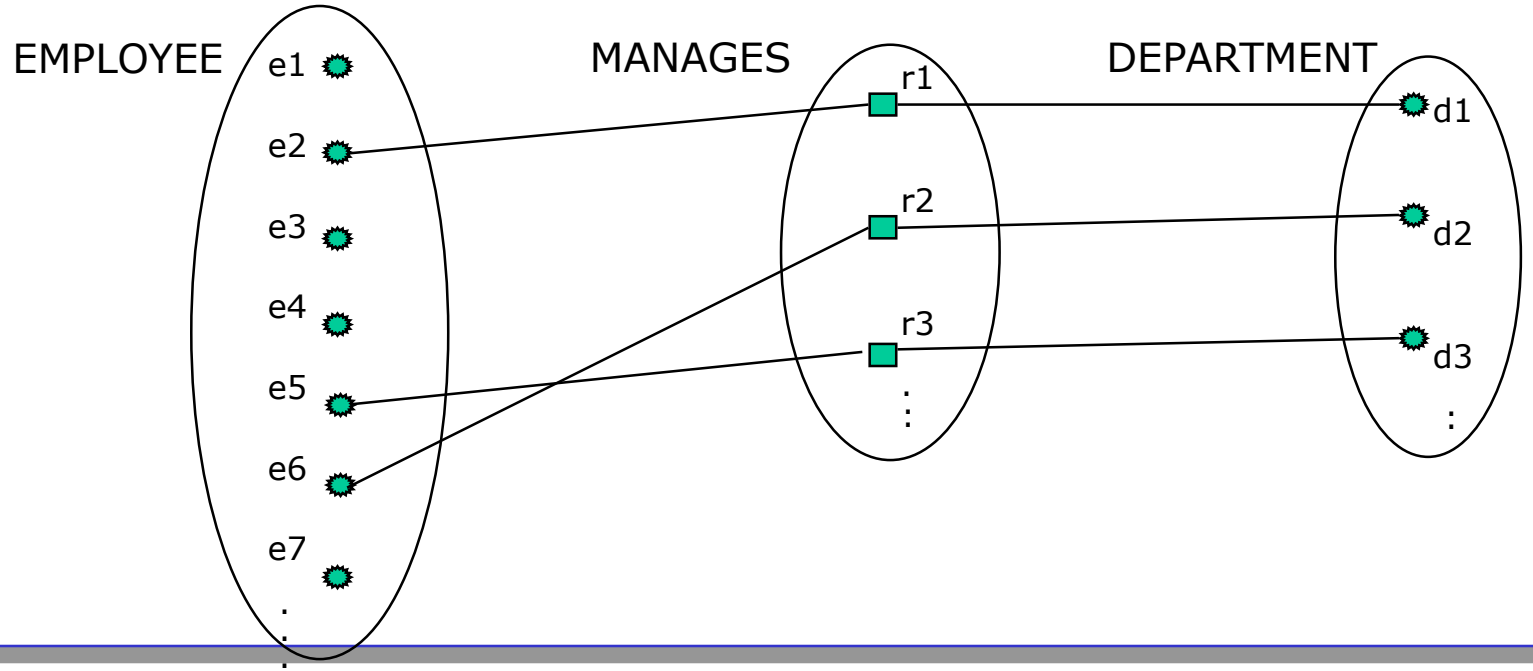
[그림]

Employee에서의 순환적 관계 SUPERVISION은 상사(1)와 부하(2)의 두 역할로 구분할 수 있음



관계, 관계 타입, 역할, 구조적 제약조건

- 관계 타입에서 제약조건
  - 카디날리티 비율 : 관계 인스턴스에 참여하는 엔티티의 개수의 비율
    (이진 관계 타입에 대한 카디날리티 비율은 1:1, 1:N, M:N)
  - 참여 제약조건 : 한 엔티티의 존재가 관계 타입을 통해 연관되어 있는 다른 엔티티에 의존하는지의 여부 (부분참여와 전부참여)
  - MANAGER 관계 : EMPLOYEE는 부분참여하고, DEPARTMENT는 전부참여함

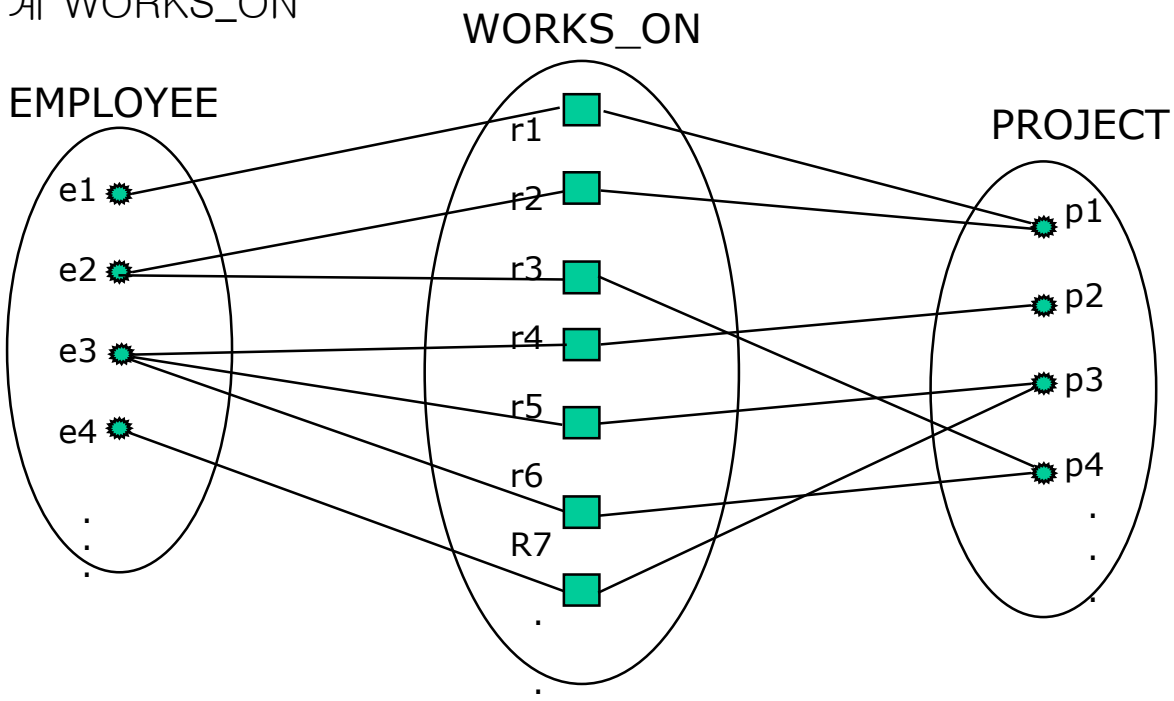


관계, 관계 타입, 역할, 구조적 제약조건

- 관계 타입의 애트리뷰트
  - 1:1, 1:N 관계 타입의 애트리뷰트

    MANAGES의 StartDate, WORKS_FOR의 StartDate
  - M:N 관계 타입

    WORKS_ON의 Hours
  - [그림] M:N 관계 WORKS_ON



약한 엔티티 타입(Weak Entity Type)

- 자신의 키 애트리뷰트가 없는 엔티티 타입
  - 약한 엔티티 타입의 개체들은 다른 타입의 엔티티와 연계됨으로써 식별될 수 있음
  - 예: DEPENDENT 엔티티 타입은 Employee 엔티티와 연계될 경우에 식별됨
- 식별(소유) 엔티티 타입과 식별 관계
  - 약한 엔티티 타입의 개체들을 식별하는 엔티티 타입을 식별 엔티티 타입이라고 함
  - 예: EMPLOYEE와 DEPENDENT에서 EMPLOYEE가 식별 엔티티 타입이며, 두 엔티티 타입 사이의 관계를 식별 관계라고 부름
- 부분 키
  - 특정 소유 엔티티와 연관된 약한 엔티티 집합 내에서 서로를 구분할 수 있는 속성들의 집합
  - 예를들어 employee e1의 dependents set 내에서 서로를 구분할 수 있는 속성들의 집합은 Dependent.name이고, 이를 부분 키라고 함
- 약한 엔티티의 표현
  - 소유 엔티티 타입의 복합 속성으로 표현될 수도 있음
  - 그러나, 다음의 경우에는 별도의 엔티티 타입으로 표현하는 것이 바람직함

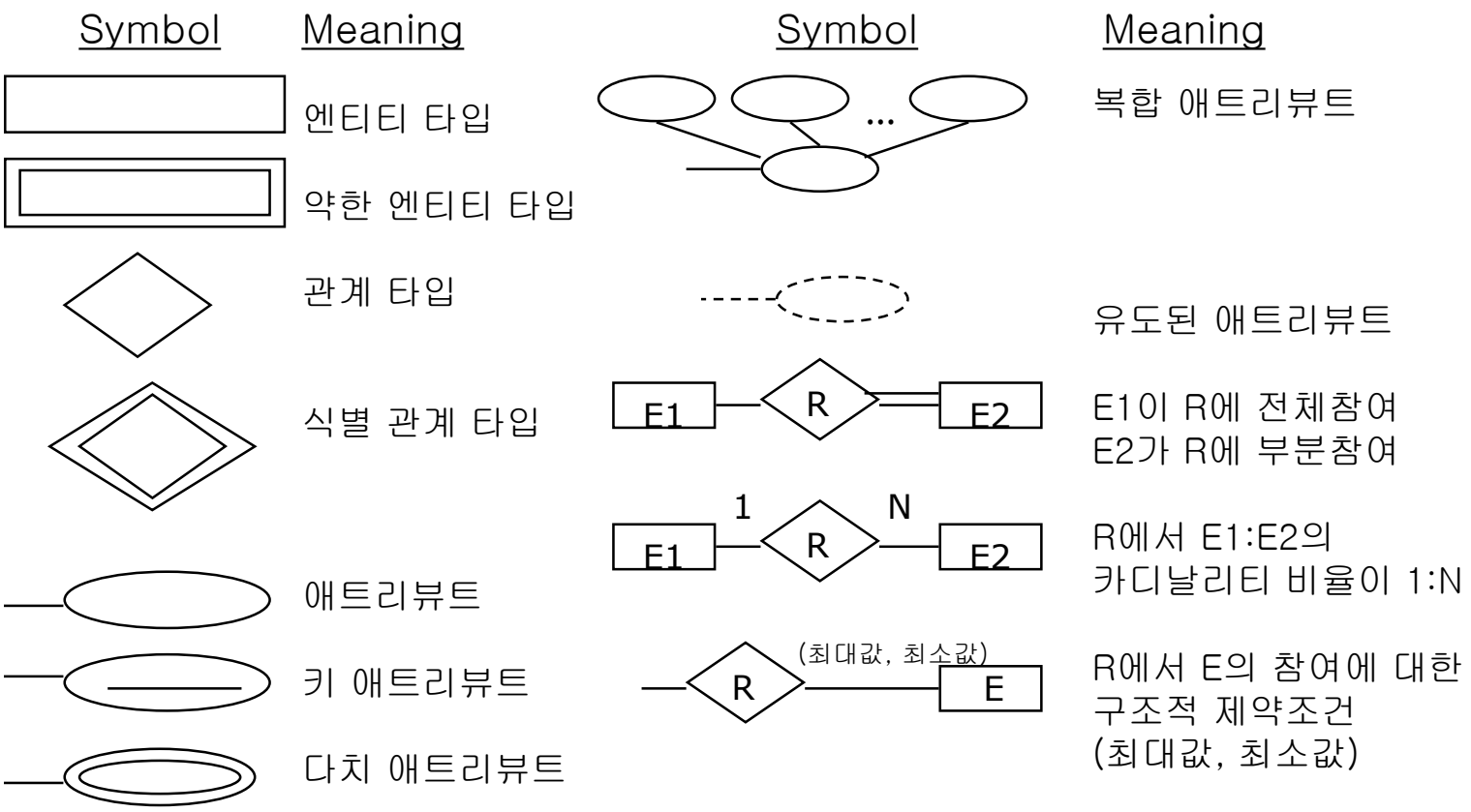
    (1) 엔티티가 많은 애트리뷰트들을 가지고,
    (2) 식별 관계 타입 외에 다른 관계 타입들에 독립적으로 참여하는 경우

COMPANY DB에 대한 ER 설계의 개선

- 그림에서 관계를 나타내는 애트리뷰트들을 관계 타입으로 변환하여 발전시킴
- 생성되는 관계 타입들
  1. MANAGES : EMPLOYEE와 DEPARTMENT 사이의 1:1 관계 타입
  2. WORKS_FOR : DEPARTMENT와 EMPLOYEE 사이의 1:N 관계 타입
  3. CONTROLS : DEPARTMENT와 PROJECT 사이의 1:N 관계 타입
  4. SUPERVISION : EMPLOYEE(상사역할)와 EMPLOYEE(사원역할) 사이의 1:N 관계 타입
  5. WORKS_ON : EMPLOYEE와 PROJECT 사이의 M:N 관계 타입
  6. DEPENDENTS_OF :EMPLOYEE와 DEPENDENT 사이의 1:N 관계 타입

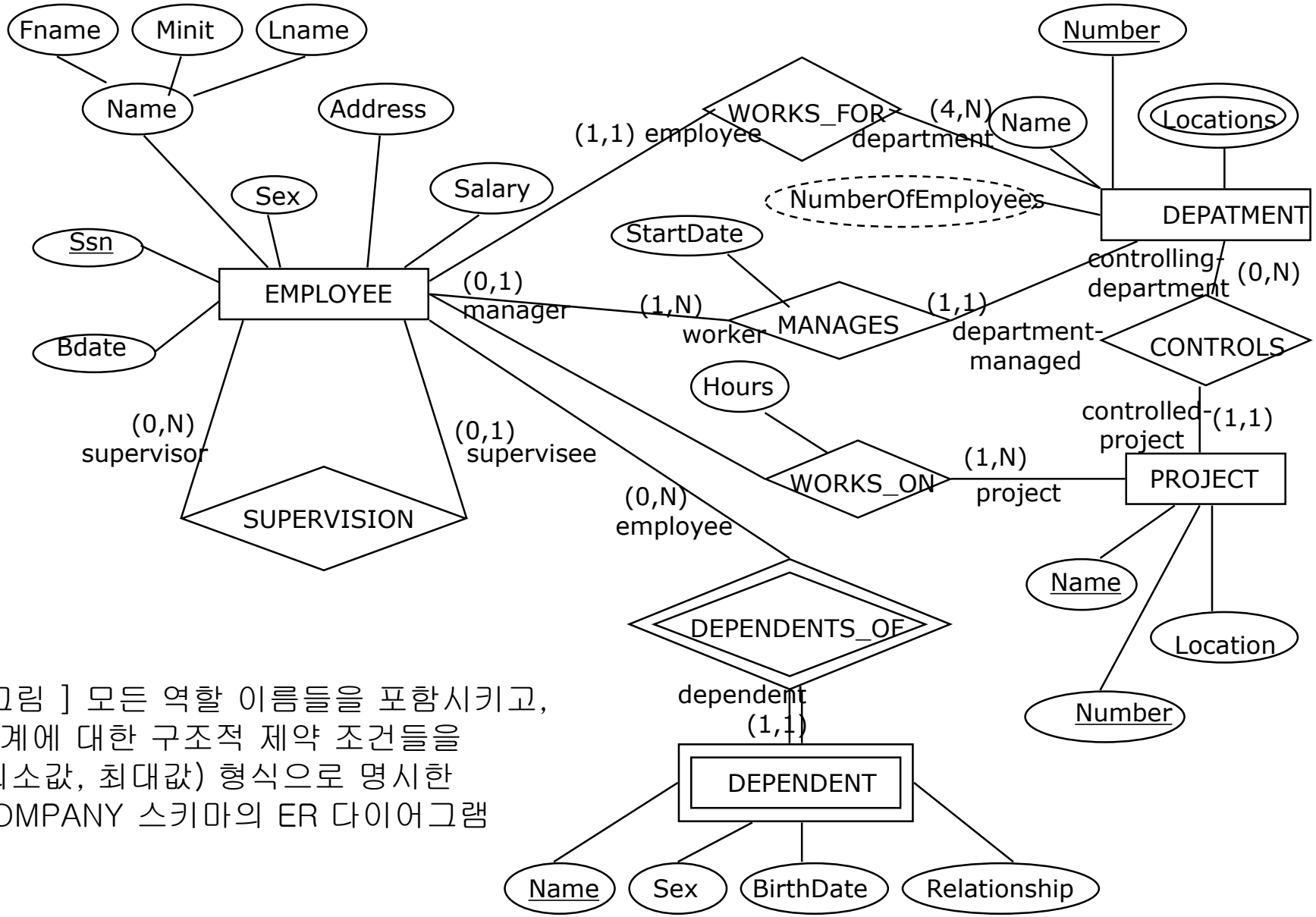
ER 다이어그램, 명명에 관한 규칙, 설계에 관한 사항

- [그림] ER 다이어그램의 표기법 요약



ER 다이어그램, 명명에 관한 규칙, 설계에 관한 사항

- 스키마에 사용된 각 구조물에 대해 가능한 한 그 의미를 전달할 수 있는 이름 선택
- 복수보다 단수 이름 선택
- 일반적으로 자연어로 기술된 요구 사항에서 명사는 엔티티 타입 이름, 동사는 관계 타입 이름으로 해석하는 것이 도움이 됨
- ER 다이어그램은 왼쪽에서 오른쪽, 위에서 아래로 읽기 쉽게 작성함
- 스키마 설계는 반복해서 개선하는 작업이 필요함 – 한번에 완성하기는 쉽지 않음
- 다음 TP에서는 ER 다이어그램의 또 다른 표기법 소개



[그림] 모든 역할 이름들을 포함시키고, 관계에 대한 구조적 제약 조건들을 (최소값, 최대값) 형식으로 명시한 COMPANY 스키마의 ER 다이어그램

요 약

- 엔티티-관계(ER) 모델을 사용한 모델링 개념
  - 엔티티 (개체)와 관계의 정의
- 스키마 레벨에서 ER 모델 개념
- 관계 타입의 구조적 제약조건
- ER 다이어그램