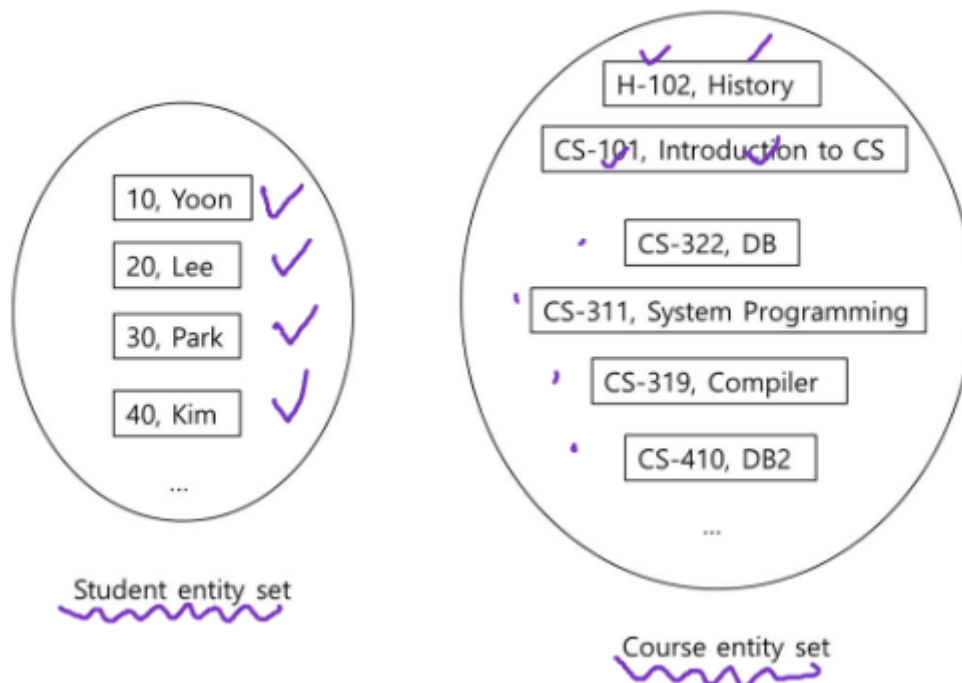# Entity

ER(Entity-Relation) DB 설계할때 쓰는 모델링

- 구분할 수 있는 엔티티
- 다 된다 (person, company, student.. 등등)
- Entity have attribute people have names and address
- entity set/type is a set of entities with the same properties

Entity Sets

동일한 속성을 가진 Entity의 집합

Relationship
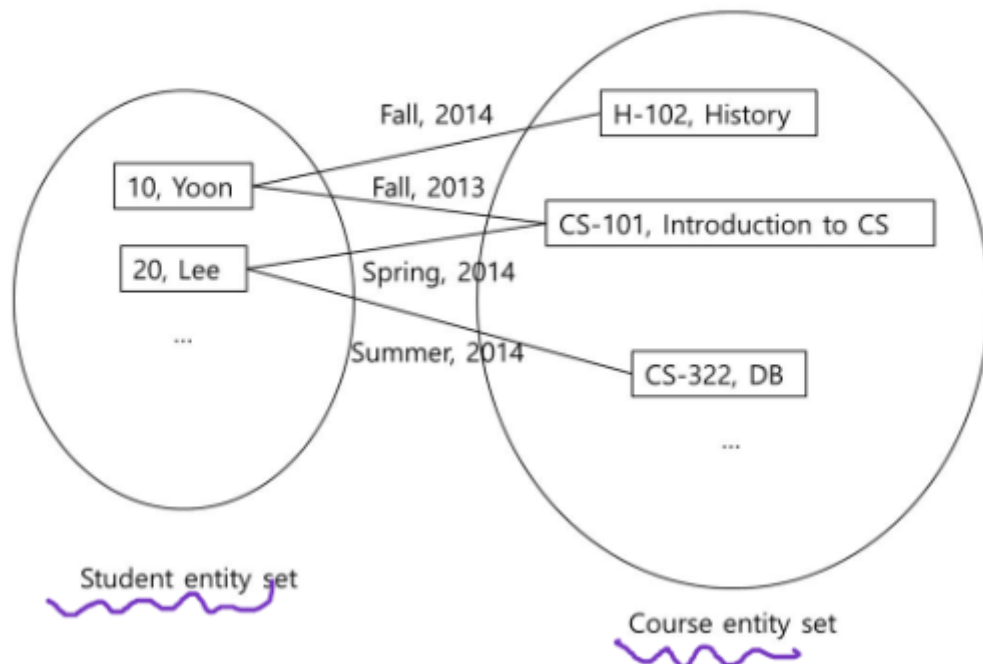
entity가 관련있으면.. ex) Student s1 related to course c2

- 속성을 가질 수 있음 어느 학기의 어느 년도?

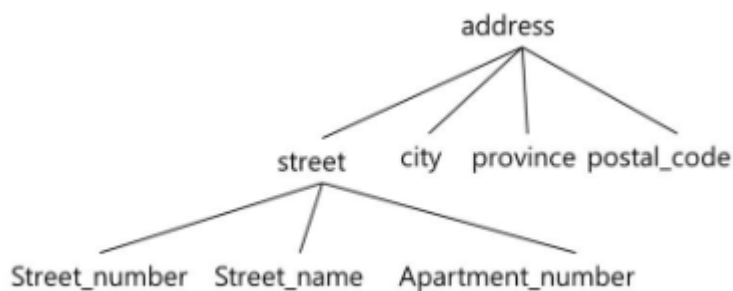- degree of a relationship set 관련되는 entity몇개? ternary/quteray/...

Attribute

- 어떤 entity는 속성의 집합으로 나타남 professor = {professorID, name, deptname, salary}

- derivea attribute 생년월일 입력받으면 나이까지 알 수 있음

Composite Attribute Example

Cardinality Constraints

- 1:1
- 1:n
- n:1
- 1:n, n:1

keys

super key.. 등등 예전에 배운 것과 동일함

- relationship type에 대한..

E-R Diagram

ER data 모델링의 결과치... 표기를 어떻게 할 것인지? -> ER Diagram

- Rectangles represent entity sets ✓
- Diamonds represent relationship sets ✓
- Attributes are listed inside entity rectangle
- Underline indicates primary key attributes

- 네모: entity set
- 다이아몬드: relationship set (두 네모 사이 relation)
- 네모 밑: 속성들
- 밑줄: primary key
- (속성): 하나 이상이어도 된다는 뜻 -> composite attribute
- 속성(): derived attribute

relationship(다이아몬드)도 attribute를 가질 수 있음

Cardinality Constraints

화살표 존재 각 하나씩이면 1:1 화살표 없으면 n

- Many-to-one
  - A professor is associated with at most one student
  - A student is associated with several (including 0) professors
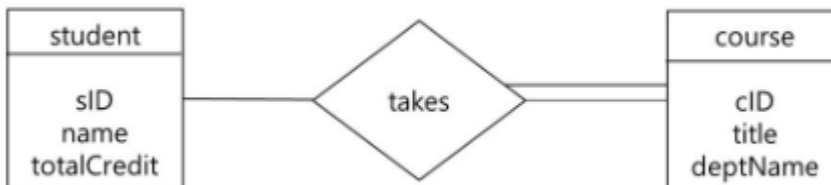
- Many-to-many
  - A professor is associated with several (possibly 0) students
  - A student is associated with several (possibly 0) professors
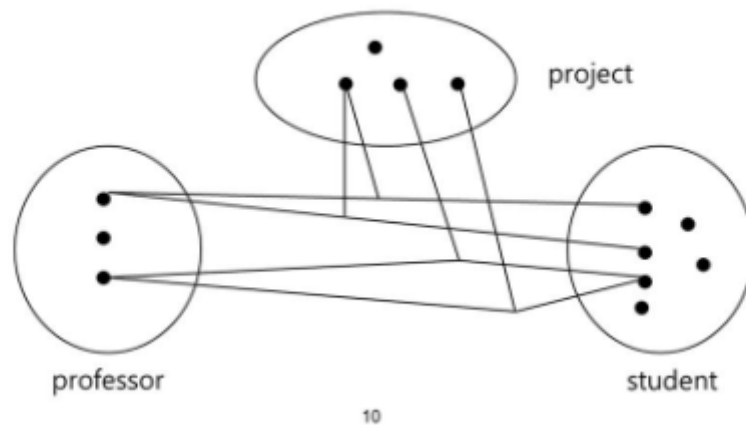
Particiption Constraints

- Total participation 더블라인으로 표현 -> 해당 테이블의 모든 entity가 포함되어야함

- Partial participation 모든 entuty 표현 안되어도 됨

- m:n
- course는 모두 student를 가져야함
- student에서는 course안들어도 됨.

Ternary Relationship

Cardinality 할 때

one은 하나만(??)

Roles

course entity간의 relationship? recursive relationship 선후수 과목... role이 필요함 (역할)

Weak Entity Set

primary key를 갖지 않는 Entity set -> 왜 존재? identifying entity set에 의존적임 더블 다이아몬드 / 더블 네모로 표현

Weak의 key는?

weak entity type의 primary key... 없으므로 partial key

Exercise

1. Draw an ER diagram for the company below. Make assumptions on constraints if necessary. The company keeps track of a company's employees, departments, projects, and so on.

2. The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department, We keep track of the start data when that employee began managing the department. A department may have several locations.

3. A department controls a number of projects, each of which has a unique name, a unique number, and a single locat

4. We store each employee's name, social security number, address, salary, gender, and birthdate. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee

5. We store each employee's name, social security number, address, salary, gender, and birthdate. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee
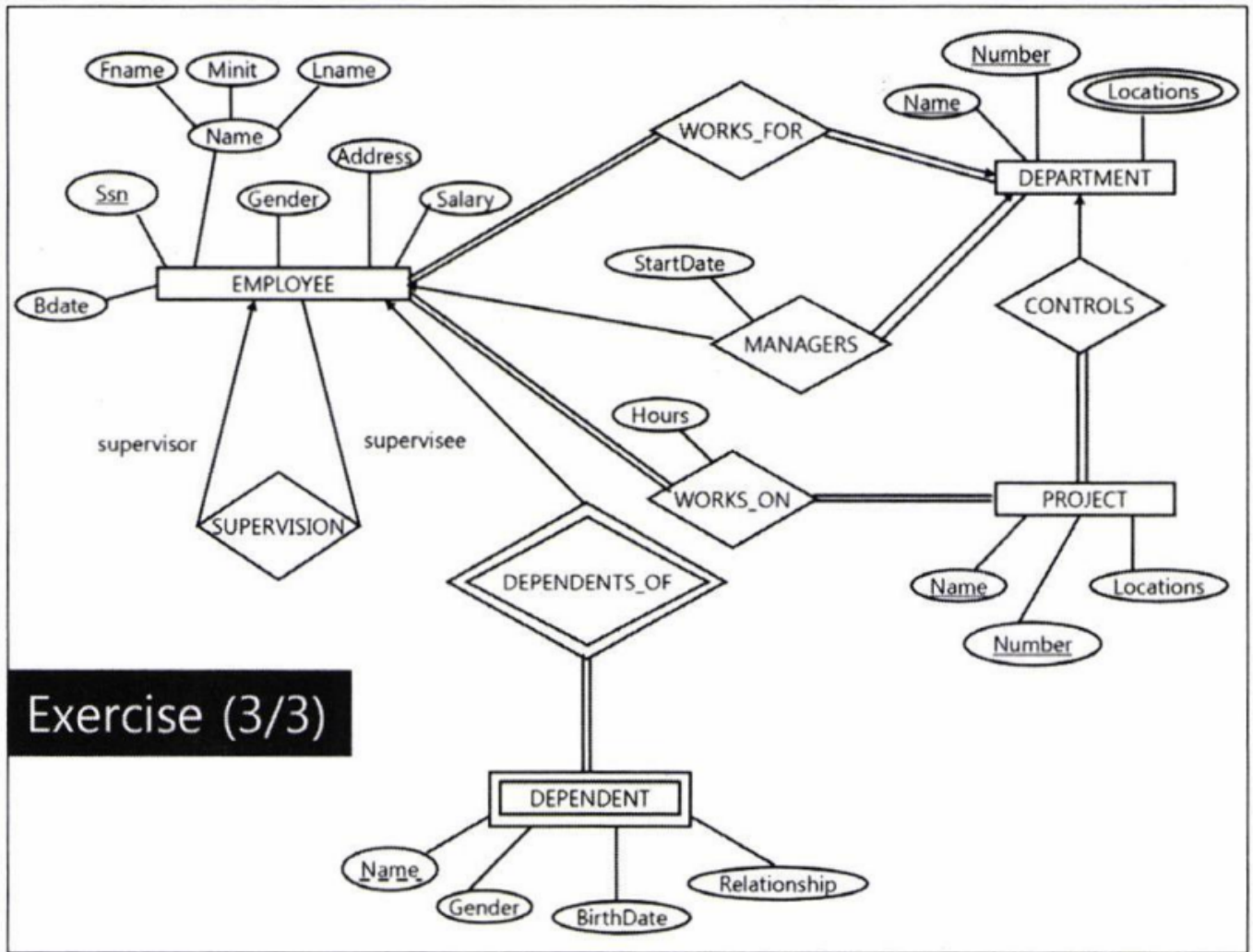
Reduction to Relation Schemas
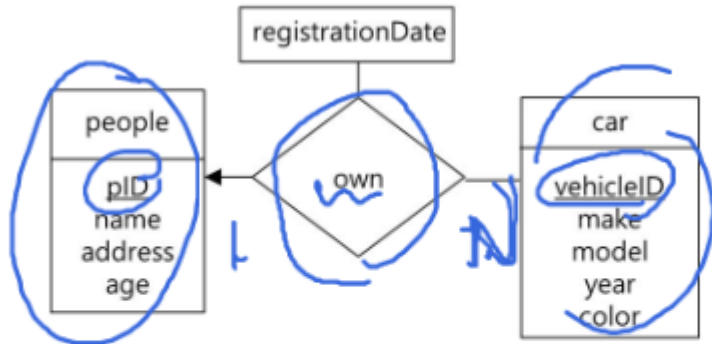
ER 다이어그램으로부터 스키마 생성하기

Strong -> Table weak -> 변화한 후 Table

n:m인 경우 그냥 그대로 가능 n:1인 경우 many side역할 1:1안 아무 쪽이나 entity type으로 가능?

1:n) people <- own - car

people, car은 하나의 테이블로 가면 됨.

- own은 nside의 key를 가져오기 (car-vehicleID)

- nside 밀기 own의 속성인 registrationDate를 car에 넣고 people의 주키인 pID도 넣기

- people(pID, name, address, age) ✓
  own(pID, vehicleID, registrationDate)
  car(vehicleID, make, model, year, color) ✓

- people(pID, name, address, age) ✓
  car(vehicleID, make, model, year, color, registrationDate, pID)

1:1

own - 속성 + 양쪽 주키 왼쪽을 밀어도 되고 오른쪽을 밀어도 되고..

multivalue attribute

table로 간다. 그런 경우 키 - ID+Phonenum

Database Design Issues

ER diagram	decision
Attribute	entity
entity set	relationship set
ternary relationship	a pair of binary relationship
strong	weak
specialization	generalization

entity set and attribute has no right/no wrong

Redundant Attributes

각 테이블에 중복되는 속성 존재시 하나는 삭제해야함 -> 나중에 일치성 문제...

대부분은 Binary 사용