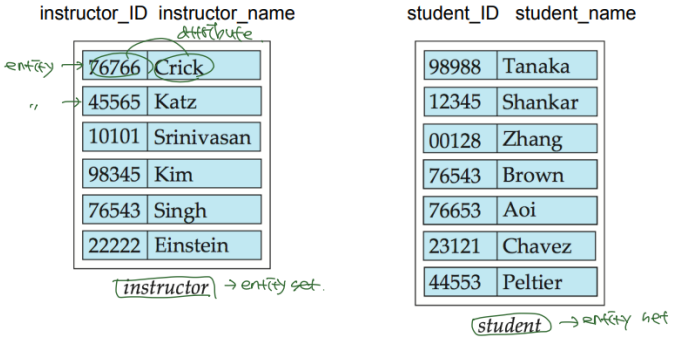
10. E-R Diagram

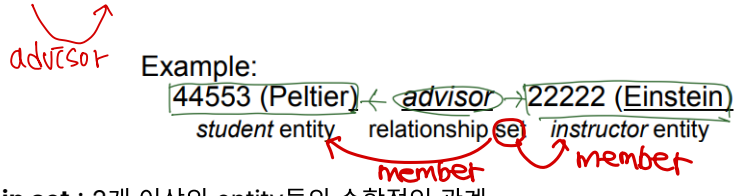
▼ Modeling(Database model)

- **database**
  - a collection of entities
  - entity 간의 관계

- **entity set**
  - **entity** : 다른 객체와 구별할 수 있는 객체 → 하나의 tuple
    - ex. williams 라는 학생
  - **attribute** : entity의 성질
    - ex. 학생은 name, phone number들의 고유한 성질을 가짐
  - **entity set** : 속성을 공유하는 같은 type의 entity set
    - ex. 모든 학생 집합, 학과 등



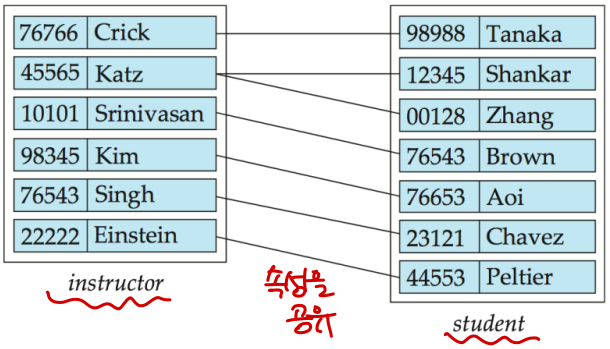
- **Relationship Sets**
  - **relationship** : 여러 entity들의 상관관계
    - ex. 지도학생, 지도교수와 같은 맥락



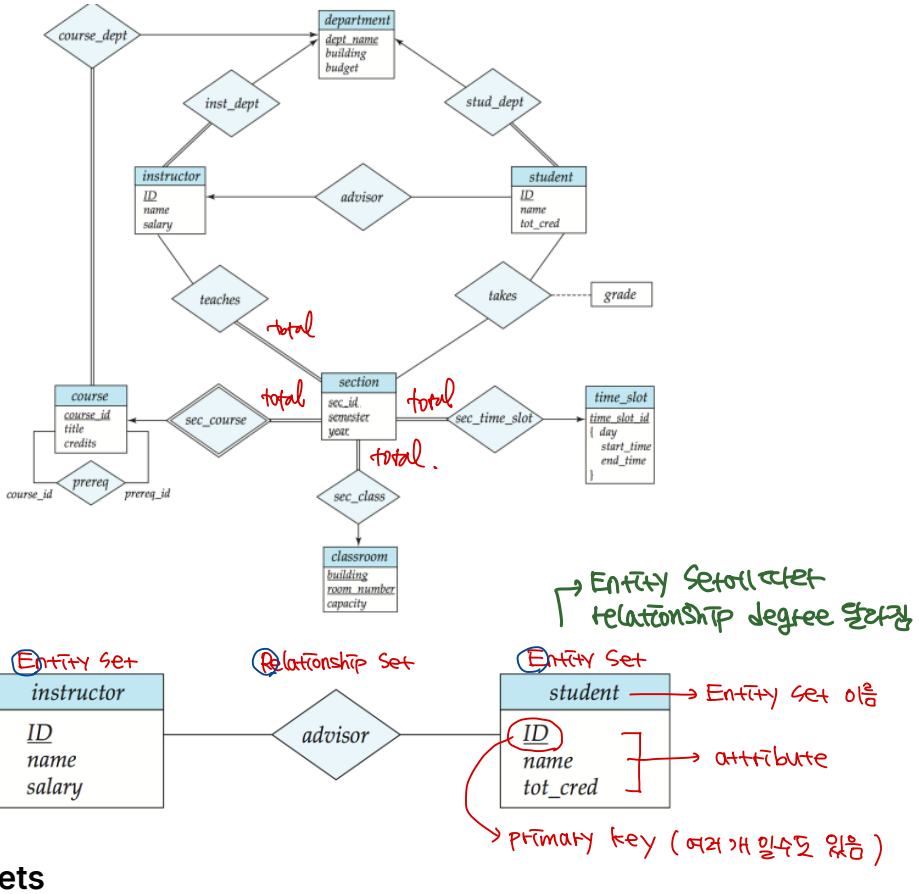
  - **relationship set** : 2개 이상의 entity들의 수학적인 관계

$$\{(e_1, e_2, ...e_n)|e_1 \in E_1, e_2 \in E_2, ..., e_n \in E_n\}$$

- (44553,22222) in advisor

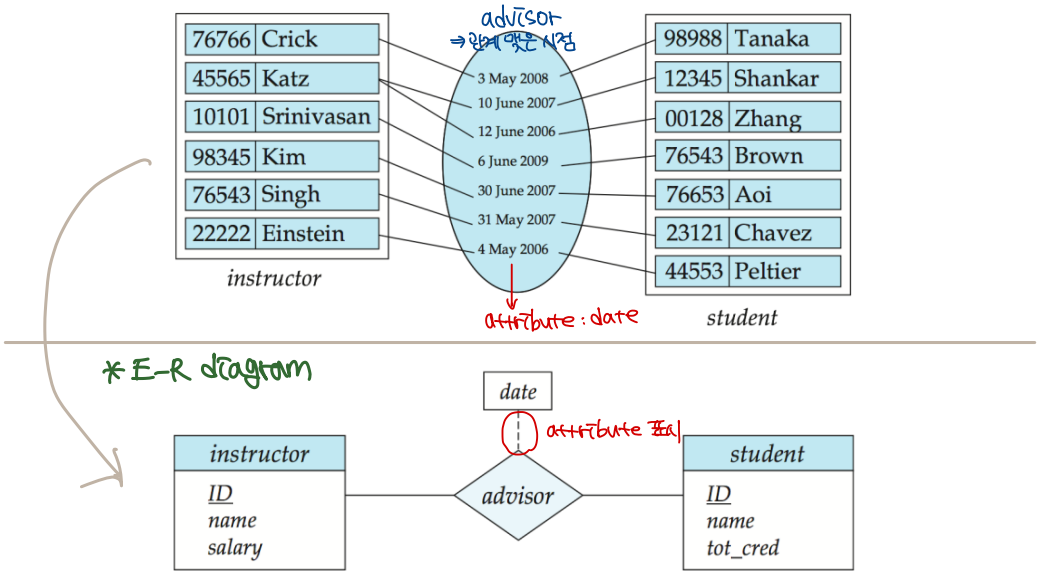


▼ E-R diagrams



▼ Relationship sets

- attribute : relationship set의 property 될 수 있음



- binary relationship
  - degree = 2 → 2개의 entity set 포함 (ternary : 3개)
  - 대부분의 relationship이 binary
    - 두 개보다 많아지는 건 좀 모호해지는 경향이 있음 → 일관적이게 표현
    - example
      - *student* : *instructor*의 가이드에 따라 *project* 진행 ⇒ entity set이 3개
      - **proj_guide** : ternary relationship(degree = 3)
        - *instructor, student, project*

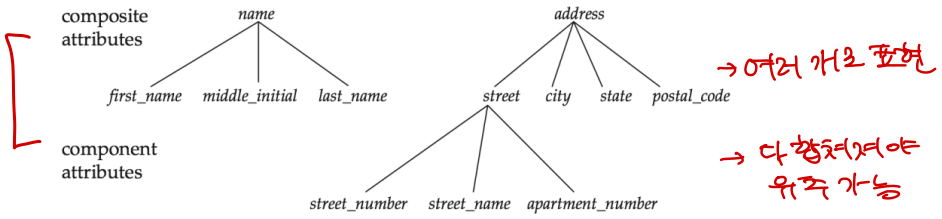
        ⇒ 몇 개의 project, 몇 명의 student와 상관관계가 있는지 모호해짐

▼ Attributes

- entity : entity set의 모든 멤버가 entity set attribute set 가짐
  - example

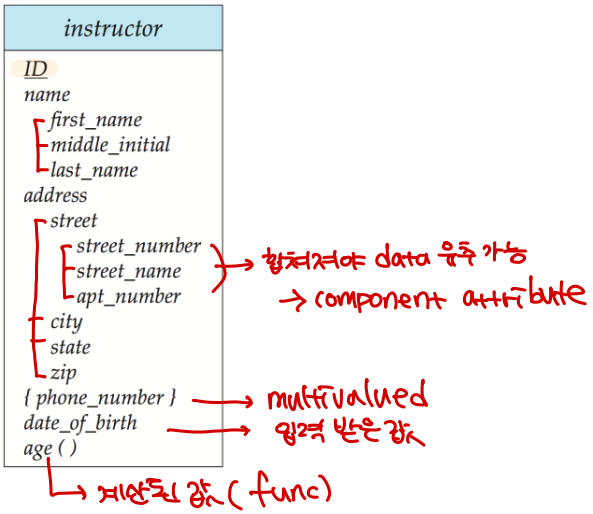
    instructor = (ID, name, street, city, salary) ⇝ domain 과학

    course= (course_id, title, credits)
- Domain : 각 attribute의 permitted value set
- attribute types
  1. **simple** or **composite** : 하나 혹은 여러 개
     a. ex : 학번 → simple / 집 주소 → compositie
     b. composite attributes



2. **single-valued** or **multivalued**
    a. ex : 학번 → single, 집 주소 → multivalued
3. **derived** : 계산이 가능한 attribute
    a. 값을 직접 입력 받지 않아도 계산 가능해야 함
    b. ex. 생일 → 나이 계산 가능

- example

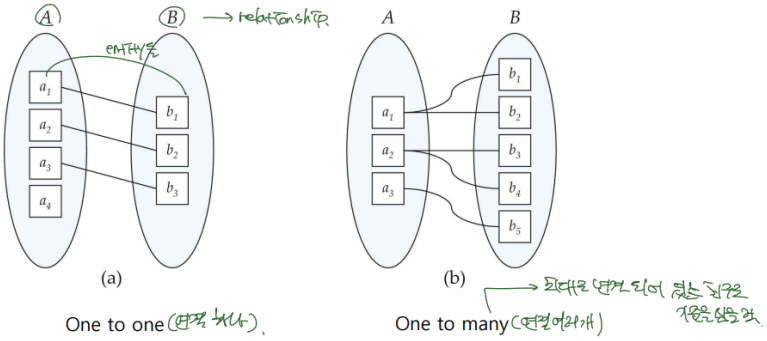


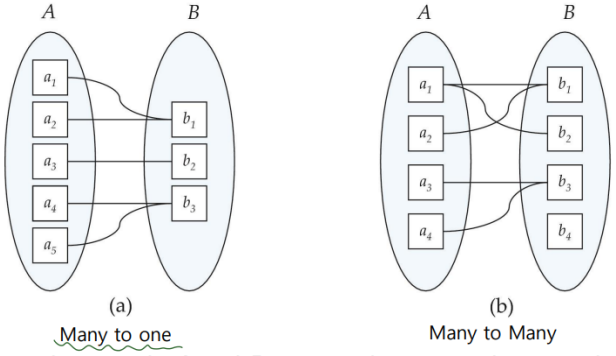
- composite attribute :
    - 각 component attribte에 대해 seperate attribute를 만들어서 동등하게 만듦
- ignoring multivalued attribute → 확장된 instructor schema

```
instructor(ID,
    first_name, middle_initial, last_name,
    street_number, street_name,
    apt_number, city, state, zip_code,
    date_of_birth)
```

▼ Mapping Cardinality Constraints

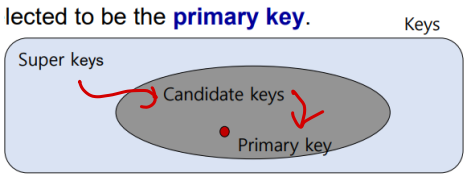
- relationship set을 통해 다른 entity와 연관될 수 있는 entity들의 수를 표현할 수 있음
    - A and B에 있는 element : 다른 set에 어떤 element와도 mapping 되지 않을 수도





- **binary relationship set types**
  1. one to one
  2. one to many
  3. many to one
  4. many to many

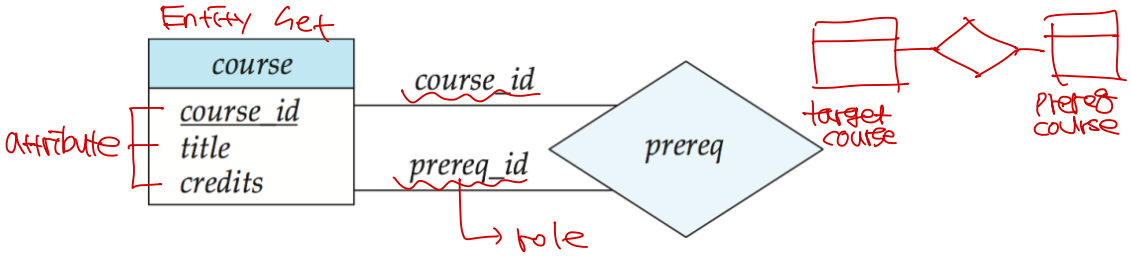
- **Keys**
  - record의 unique identifier



1. **Super Key** : 각 entity를 구분할 수 있는 attribute 집합
   - ex) (ID, name) is a super key of *instructor*
2. **Candidate Key** : is a minimal super key ⇒ 하나만으로 구분이 가능
   a. 여러 개 존재 가능(null 포함 가능)
   b. 여러 개를 합쳤을 때만 하나의 candidate key로 존재할 수 있음
   c. candidate key 중 하나는 **Primary Key**로 selected
   - ex) ID is a candidate key of *instructor*
   - ex) course_id is a candidate key of *course*

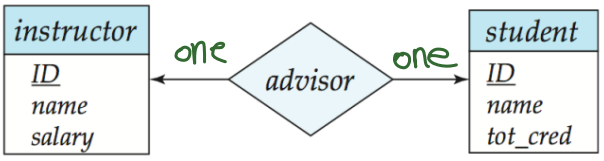
3. **Primary Key** : candidate key 중 하나
    a. entity set 내부에 있는 Primary Key의 집합 → relationship set의 super key
    - ex)
        - s_id : primary key of *student*
        - i_id : primary key of *instructor*
        - ⇒ (s_id, i_id) is the super key of *advisor*

- **Roles**
    - entity가 relationship 내에서 어떠한 역할을 수행하는지 나타냄
    - entity set 이름 자체가 역할로 사용 → 특별히 명시하지 않아도 됨
    - 명시가 필요한 경우 → relationship 내에서 같은 table이 2번 이상 참여할 때 명시



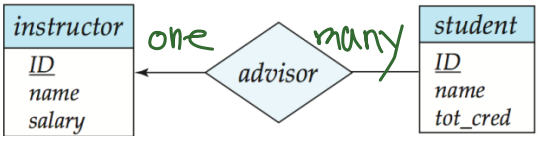
- **Cardinality Constraints**
    - line type에 따라 다르게 표시됨
        - → : one
        - — : many

1. **One-to-One Relationship**



- one-to-one relationship (instructor and student)
    - *instructor* : 최대 하나의 *student*와 advisor relationship
    - *student* : 최대 하나의 *instructor*와 advisor relationship

2. **One-to-Many Relationship**

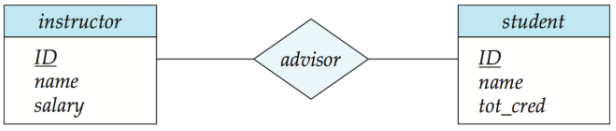


- One-to-Many Relationship
  - *instructor* : 여러 *students*(0개도 가능)와 advisor relationship
  - *student* : 최대 하나의 *instructor*와 advisor relationship

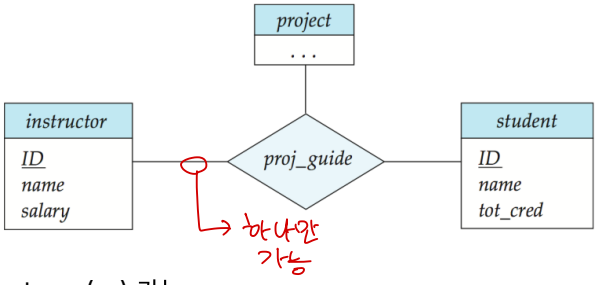
3. **Many-to-One Relationship**

- Many-to-One Relationship
  - *instructor* : 최대 하나의 *student*와 advisor relationship
  - *student* : 여러 *instructor*(0개도 가능)와 advisor relationship

4. **Many-to-Many Relationship**



- Many-to-Many Relationship
  - *instructor* : 여러 *students*(0개도 가능)와 advisor relationship
  - *student* : 여러 *instructor*(0개도 가능)와 advisor relationship
- ternary relationship relationship한 entity set 사이에 arrow 하나만!



  - arrow : at most one(—) 가능
    - **proj_guide**에서 *instructor*로 가는 arrow : 각 학생이 project마다 최대 한 명의 가이드를 가질 수 있음

Cardinality constraints → more than one arrow → coufusing

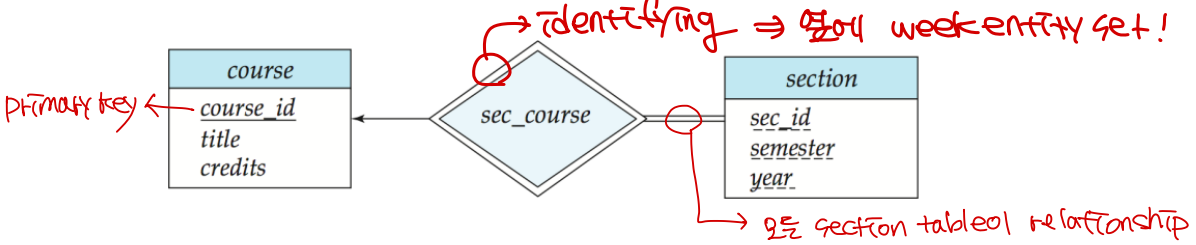
- ① 각 entity A가 unique한 entity B, C와 연관 있음 A,B / A,C
- ② (A, B) entity pair가 unique한 entity C와 연관 + (A,C) entity pair가 unique한 entity B와 연관 있음

(A,B) → C
(C,D) → B

⇒ 두 가지 중 어떤 의미인지 헷갈림 → one arrow 이상 불가

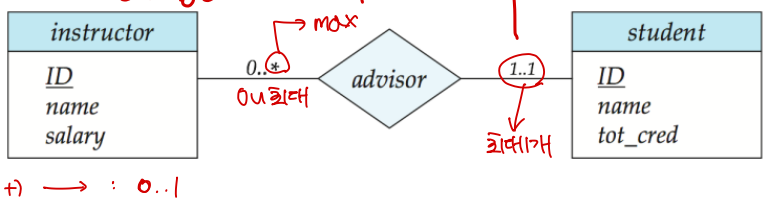
- Participation of an entity set in a relationship set

- total participation (=)
  - entity set에 있는 모든 entity가 적어도 하나의 relationship에 참여



  - ex) **sec_course**에 속하는 *section* ⇒ 모든 *section*이 참여함
- paritial participation : 부분적으로 참여

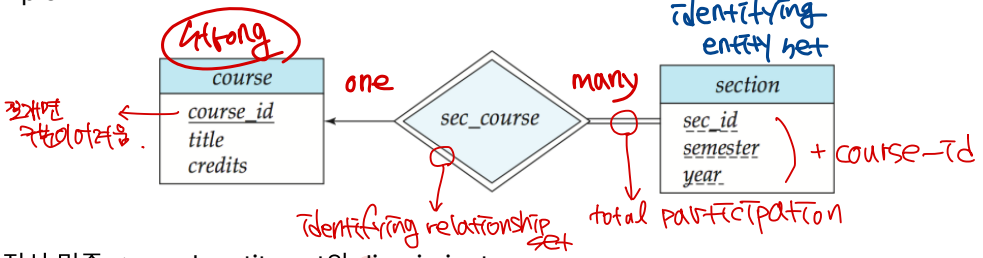
⇒cardinality limits : participation constraints 표현



▼ Weak Entity sets

primary key를 가지지 않는 entity set

- depends on **identifying entity set**
  - weak entity sets : total, one-to-many relationship set
  - **identifying relationship** : double diamond 사용
- **discriminator** (partial key) : weak entity set의 entity를 구별할 수 있는 attribute 집합
  - primary key를 만드는 다른 relation의 attribute
- weak set의 primary key
  - dentifying entity set의 primary key + weak entity set의 discriminator
- example



  - 점선 밑줄 → weak entity set의 discriminator
  - 두 줄 다이아몬드 → identifying relationship of a weak entity
  - *section* (course_id, sec_id, semester, year) ⇒ primary key

⇒ 하지만 section, course 간의 관계가 duplicated