

# 7. ENTITY-RELATIONSHIP MODEL

Entity, Relationship, Attribute

# MODELING

- 데이터베이스 모델링
  - 개체(entity)들의 집합
  - 개체들 간의 관계(relationship)
- **entity** (개체) : 다른 객체와 구별되어 존재하는 하나의 객체
  - 예) 사람, 책 (구체적) / 수업, 비행기 예약 (추상적)
  - 개체는 속성(attribute)을 가짐
    - 예) 사람들은 이름과 주소를 가짐
- **entity set** (개체 집합) : 같은 특성, 같은 타입인 entity들의 집합
  - 예) 사람은 교수님이 될 수도, 학생이 될 수도, 둘 다 될 수도, 둘 다 아닐 수도 있음



# ENTITY SET – 예시

- 2개의 entity set ( *instructor*, *student* )

instructor_ID instructor_name	
76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

*instructor*

student-ID student_name	
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

*student*

- 두 개체 집합은 각각 ID, name 두 개의 속성을 가짐

# RELATIONSHIP SET

- Relationship (관계) : 여러 개체 사이의 의미있는 연관성
- Relationship set : 둘 이상의 개체들 간의 수학적 relation

$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

// 위의 엔티티들은 엔티티 집합에서 왔고,  $E_1, E_2, E_n$  각각의 원소를  $e_1, e_2, e_n$ 이라 하면  
// 이들의 순서쌍인  $(e_1, e_2, \dots, e_n)$ 이 relationship임.

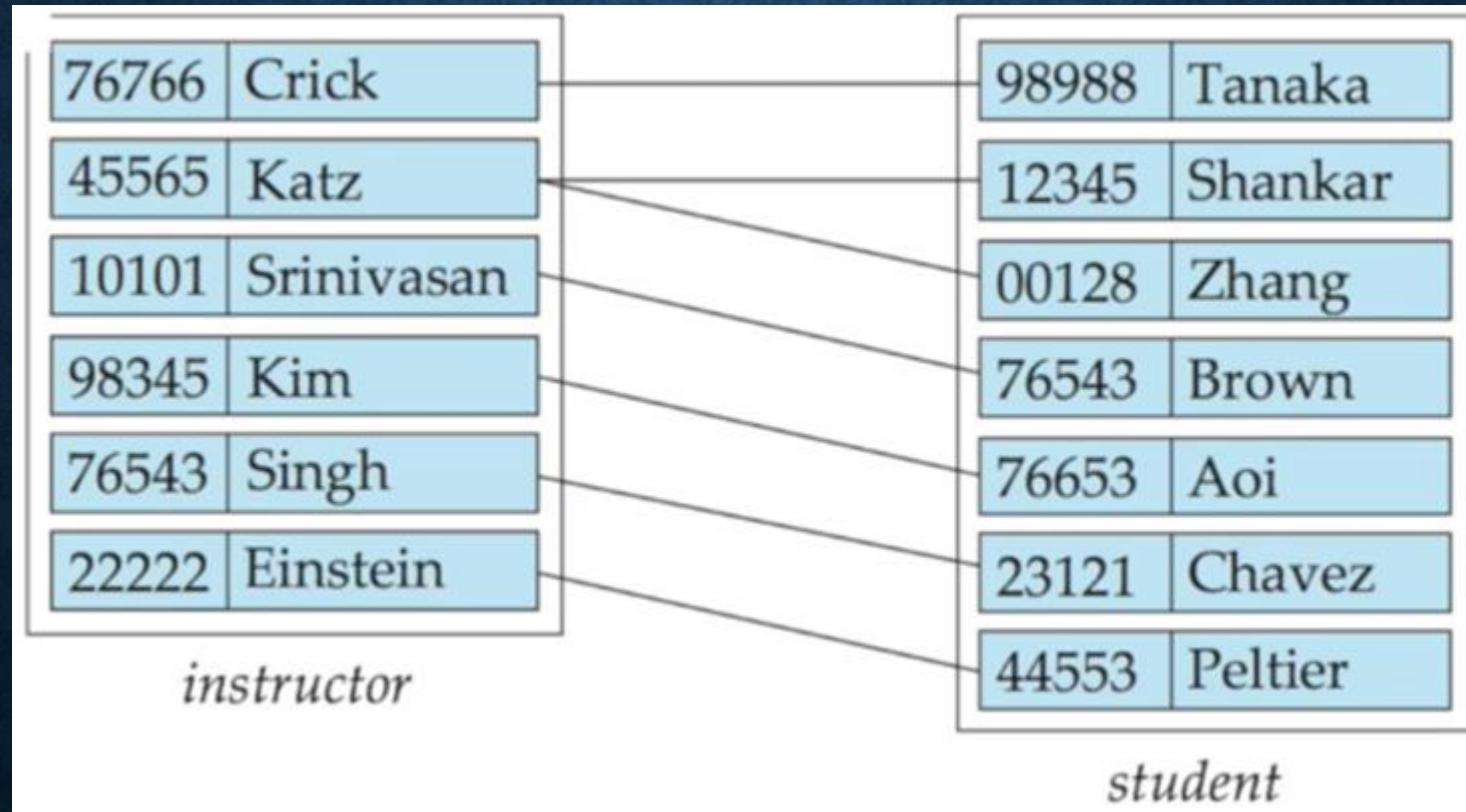
예) Peltier라는 학생 개체와 Einstein이라는 교수 개체가 서로 *advisor* 관계를 맺음

44553 (Peltier)	<u>advisor</u>	22222 (Einstein)
student entity	relationship set	instructor entity

$(44553, 22222) \in \text{advisor}$



## RELATIONSHIP SET - 예시

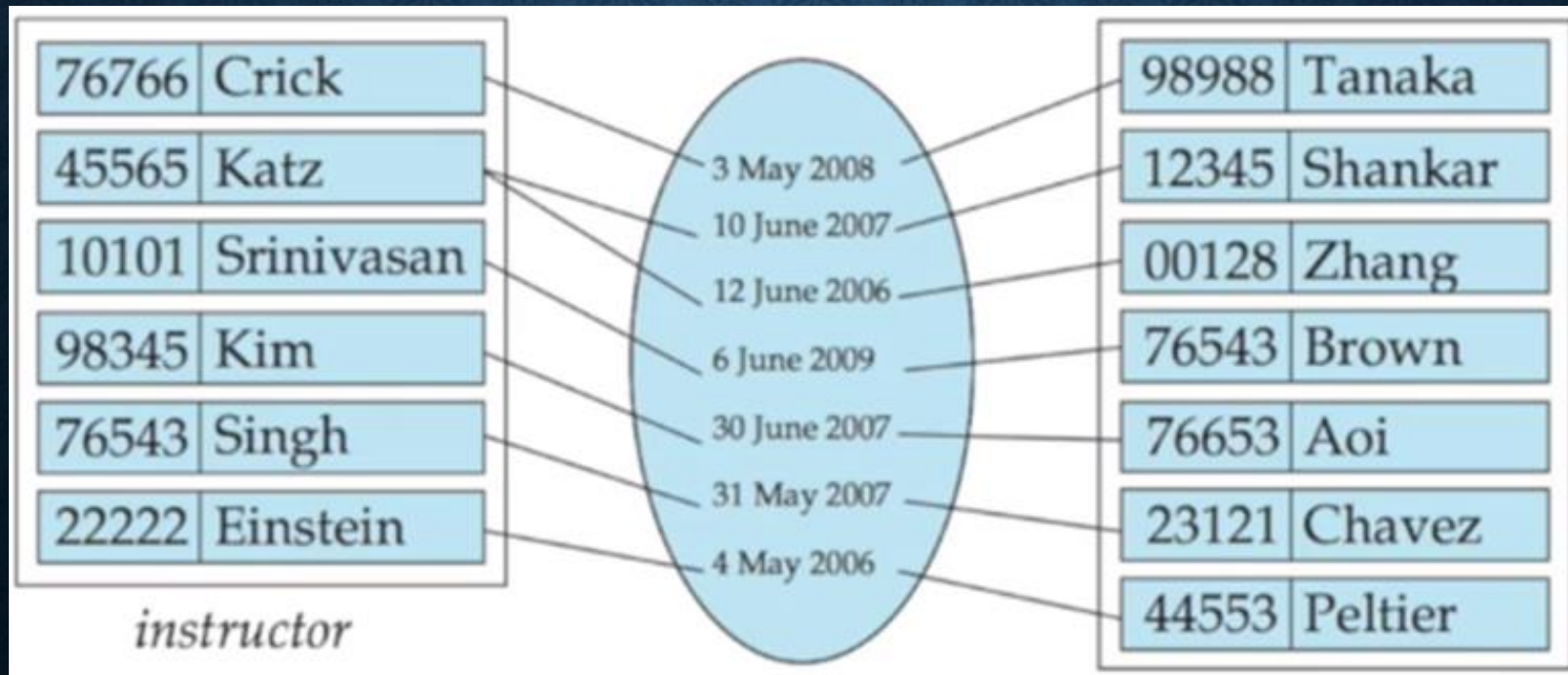


- 선 하나하나가 **entity relationship**을 나타냄

# RELATIONSHIP SET - 예시

- 관계의 특성을 나타내는 **attribute**(속성)을 가질 수 있음

예) 교수가 학생의 지도교수가 된 날짜를 명시하기 위해 'date'라는 속성을 관계에 연관시킴





# RELATIONSHIP SET - DEGREE

- **Relationship Degree** (관계 차수) : 관계 집합의 관계에 참여하는 엔티티 집합의 수
- **Binary Relationship** : 2개의 개체 집합을 포함하는 관계 집합  
( DB 시스템 내 대부분의 관계 집합은 binary임)
- 3개 이상의 개체 집합 포함 : 드문 경우  
예) 교수님의 지도 하에 학생들이 연구 프로젝트를 진행할 때  
*proj\_guide*는 *instructor, student, project* 3개의 개체 집합을 포함  
( ternary relationship = 삼항 관계를 가짐 )

# ATTRIBUTE

- 엔티티는 속성의 집합으로 표현할 수 있음
- **Attribute** (속성) : 개체 집합의 모든 멤버가 가지고 있는 특성을 설명

예) *instructor* = ( ID, name, street, city, salary )

*course* = ( course\_id, title, credits )

- **Domain** : 각 attribute에 허용되는 값들의 집합



# ATTRIBUTE

- Attribute types

1. **Simple** attribute (단순 속성) / **Composite** attribute (복합 속성)

더 이상 나눌 수 x

나눌 수 o

2. **Single-valued** (단일 값 속성) / **multi-valued** attribute (다중 값 속성)

하나의 값 가짐

여러 값 가질 수 o

예) single-valued : 학번(1학생 1학번) / multi-valued : 전화번호(1사람 여러 개o)

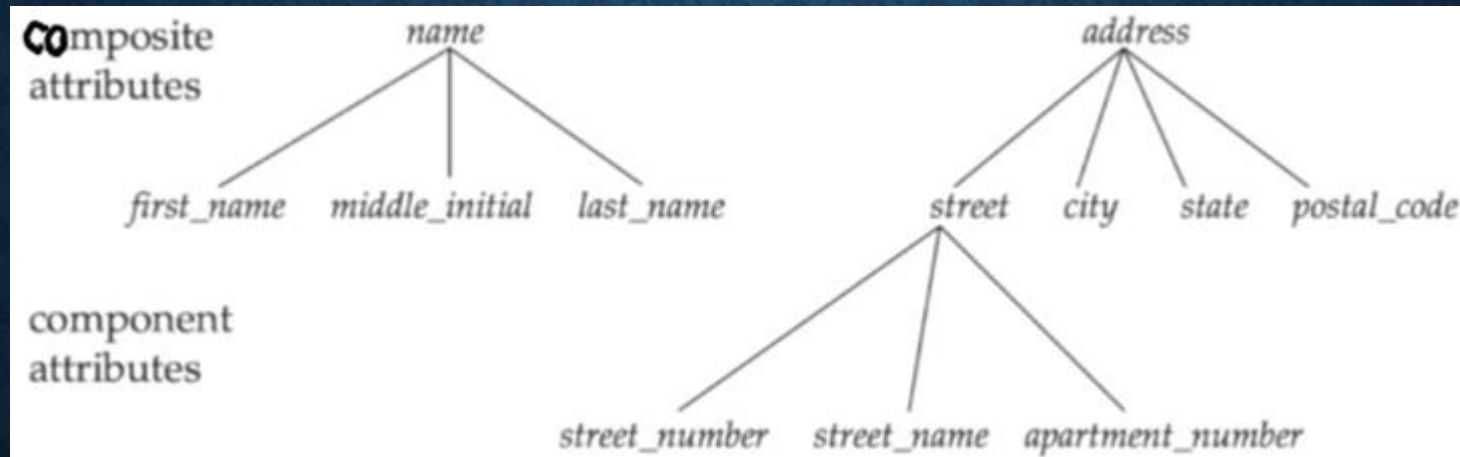
3. **Derived** attribute : 저장 x, 필요할 때마다 다른 속성을 이용하여 계산되는 속성

예) 나이 = 생일을 이용해서 계산 가능

- 개체가 속성값을 갖지 않는다면, 속성값은 null 값 가능

# COMPOSITE ATTRIBUTE

- Composite Attribute (복합 속성) : 속성의 일부만을 참조할 때 good



- 계층적인 관계를 가질 수 있음
- 장점
  1. 보다 명확한 모델링
  2. 연관된 속성 – 하나의 그룹으로 묶기 가능



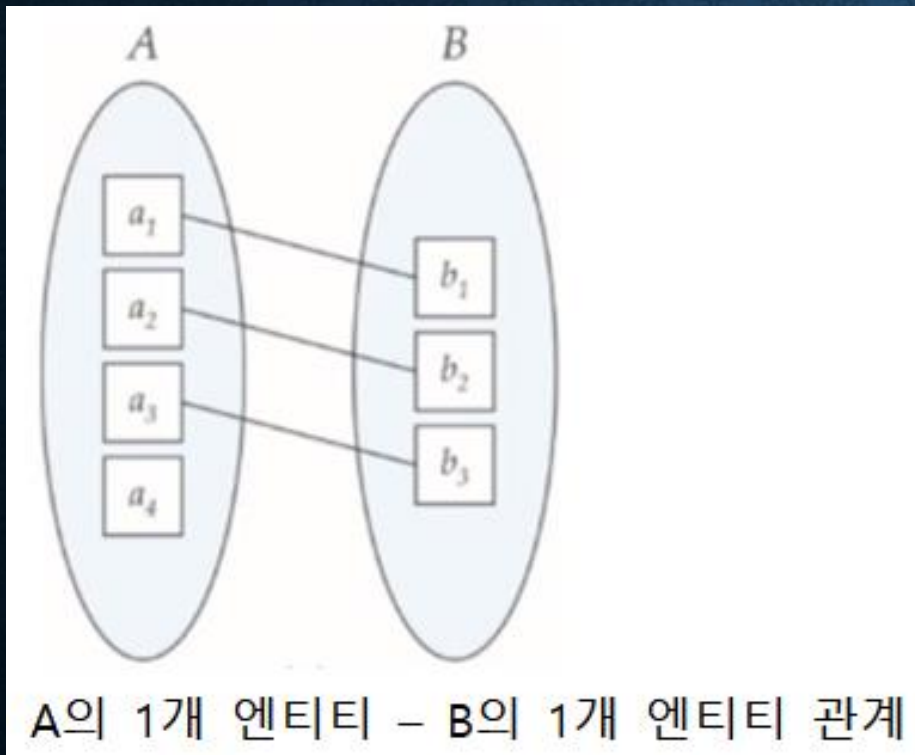
# CONSTRAINTS – E-R 모델

## 1. Mapping Cardinality Constraints

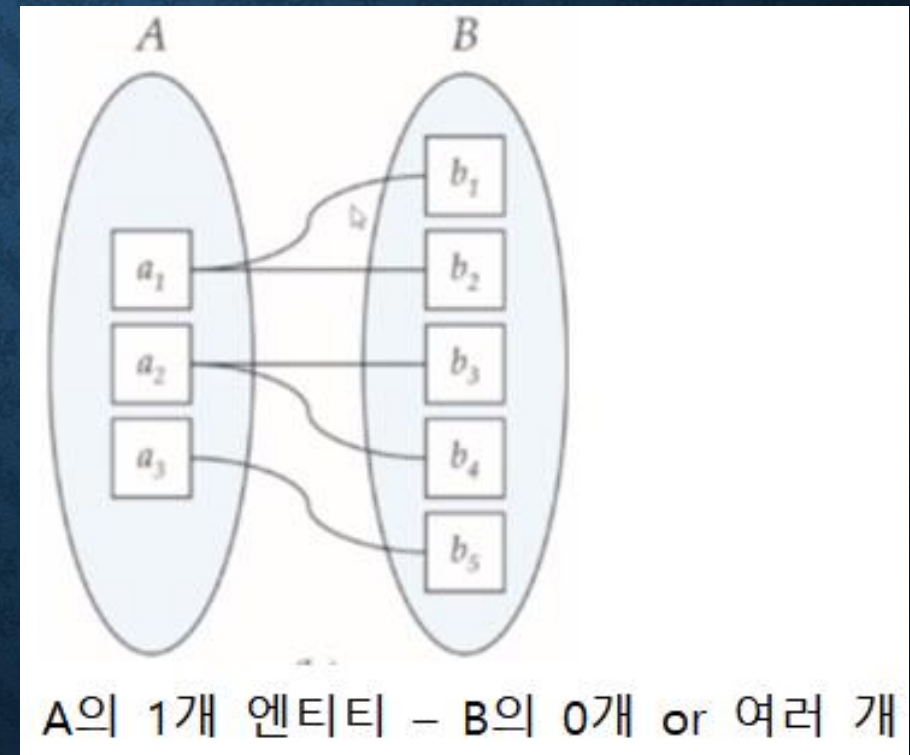
- 관계 집합을 통해, 다른 개체와 연결될 수 있는 개체의 수 나타냄  
(=릴레이션에 참여 중인 엔티티 집합의 수)
- Binary Relation 집합을 기술하는 데 유용
- Binary Relationship Set에 대한 유형 4가지
  - 1) One to One
  - 2) One to Many
  - 3) Many to One
  - 4) Many to Many

# MAPPING CARDINALITY

## 1. One to One



## 2. One to Many

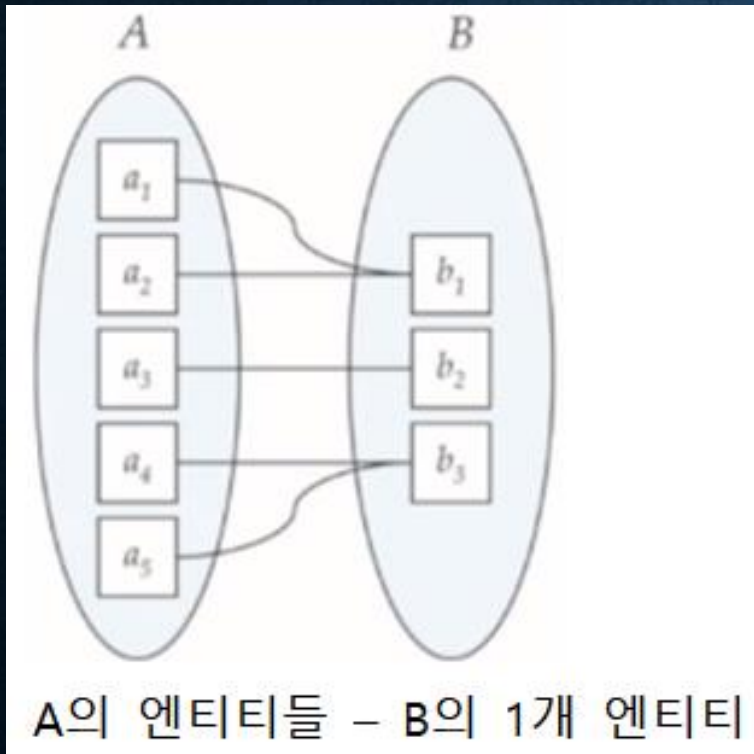


- 다른 집합의 개체와 매핑되지 않는 개체도 있을 수 있음 (ex. 왼쪽 예시 A의  $a_4$ )

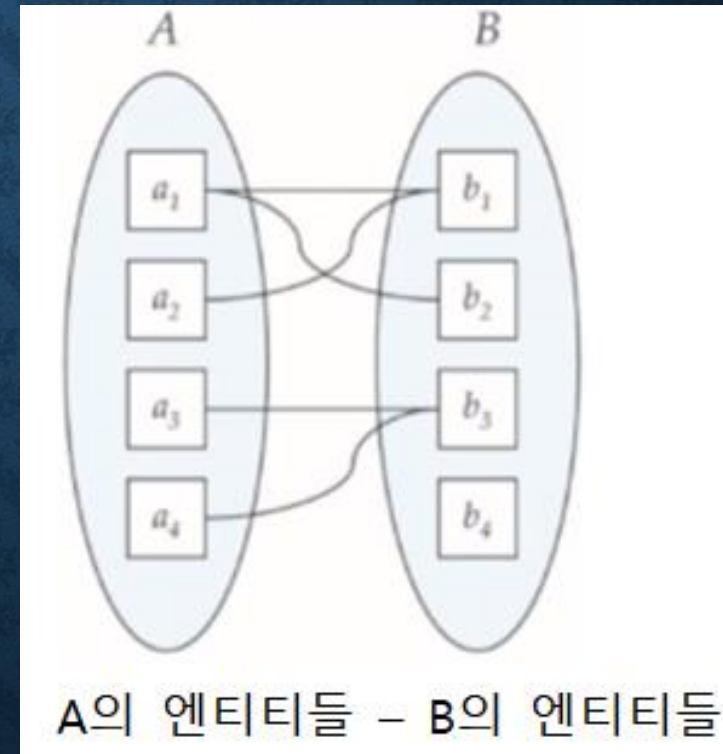


# MAPPING CARDINALITY

## 3. Many to One



## 4. Many to Many



- 다른 집합의 개체와 매핑되지 않는 개체도 있을 수 있음 (ex. 오른쪽 예시 B의  $b_4$ )

# KEY

- **Super key** : 수퍼키의 값은 각 개체를 유일하게 결정하는 1개 이상의 속성 집합
- **Candidate key** : minimal super key
- 여러 **Candidate key**(후보키)가 존재할 경우, 그 중 하나를 **Primary key**(기본키)로 선택



# KEY – RELATIONSHIP SET

- 관계 집합의 **Super key** : 관계집합에 참여하는 엔티티들의 기본키 조합으로 구성  
예) 관계 집합에 A, B 엔티티가 참여하고 각각의 기본키를  $A_p, B_p$ 라 할 때 관계 집합의 Super key =  $(A_p, B_p)$
- 관계집합에 참여하는 엔티티집합의 한 쌍은 그 관계 집합에서 최대 1개의 관계를 가질 수 있음
- Candidate key를 선택할 때 고려해야 하는 관계 집합의 Mapping Cardinality
  1. Many to Many : 관계 **Primary key**는 양쪽 개체 집합의 기본키의 합집합
  2. Many to One / One to Many : 관계 **Primary key**는 many 쪽의 기본키  
( many의 개체들은 어떤 개체가 관계에 참여하는지 명확하게 구분 필요 )
  3. One to One : 어느 쪽 기본키든 상관 없음 (모두 relationship 구별 O)  
( 한 쪽이 어떤 엔티티인지 알면 1-1으로 연결되어 있어서 반대쪽도 앎 )
- 하나 이상 Candidate key가 존재하면, 기본키 선택 시 그 관계 집합의 의미 고려

# REDUNDANT ATTRIBUTE

- DB 설계 순서 (E-R 모델)
  1. 포함해야 할 엔티티 집합 결정
  2. 그 엔티티 집합에 적절한 속성들 선택
  3. 다양한 엔티티 사이에 관계 집합들을 형성
- 관계 집합은 엔티티 집합의 속성과 똑같은 정보 제공함
  - 중복 제공되는 정보 제거 → 속성 제거 (관계집합 vs 엔티티집합 중 엔티티집합에서)
- E-R 다이어그램 → relational 모델 매핑(테이블로 변환) : 사라졌던 attribute 재도입하기도 함