

데이터베이스: 강의노트 02

A. Silberschatz, H. Korth, S. Sudarshan
Database System Concepts,
Fourth Edition, McGraw-Hill, 2002.

Part I. Data Models

2 개체 관계 모델

2.1 기본 개념

2.1.1 개체 집합

- **개체(entity)**: 다른 모든 것과 구별되는 유, 무형의 사물
- 개체는 속성들의 집합을 가지며, 이 집합의 부분 집합은 개체를 고유하게 식별할 수 있다.
- **개체 집합(entity set)**: 같은 속성을 가지는 같은 유형의 개체들의 모임
- 개체 집합의 외연(extension): 개체 집합을 이루는 각 개별 개체
- 개체 집합은 서로 중첩될 수 있다. 예) 은행의 모든 직원과 은행의 모든 고객
- **속성(attribute)**: 각 개체가 지니는 각 개체를 설명하는 특성
 - 개체 집합에 있는 모든 개체는 같은 속성을 가지나 각 개체는 각 속성에 대해 다른 값을 가질 수 있다.
 - 속성의 **도메인(domain)** 혹은 값 집합(value set): 각 속성이 가질 수 있는 값의 집합
- 각 개체는 (속성, 데이터 값)의 쌍의 집합으로 표현될 수 있다.
- 속성의 유형
 - **단순 속성(simple attribute)** vs. **복합 속성(composite attribute)**
 - 단순 속성: 더 이상 나눌 수 없는 속성
 - 복합 속성: 여러 부분으로 나눌 수 있는 속성, 예) 사람 이름, 주소
 - 복합 속성은 관련있는 속성을 그룹으로 묶을 때 유용하다.
 - **단일값 속성(single-valued attribute)** vs. **다중값 속성(multivalued attribute)**
 - 단일값 속성: 오직 하나의 값만을 가지는 속성

- **다중값 속성**: 고객의 전화번호와 같이 여러 개의 값을 가질 수 있는 속성

다중값 속성의 경우에는 가질 수 있는 최대 수나 최저수를 지정할 수 있다.

- **유도된 속성(derived attribute)**: 관련있는 다른 속성이나 개체로부터 유도될 수 있는 속성

- 만약 속성이 값을 가지지 않으면 **널 값(null value)**을 가진다.
- 널 값을 가지게 되는 이유
 - 적용할 값이 없음(not applicable)
 - 값의 존재 여부를 모를 경우(not known)
 - 값이 존재하지만 값을 모를 경우(missing)

2.1.2 관계 집합

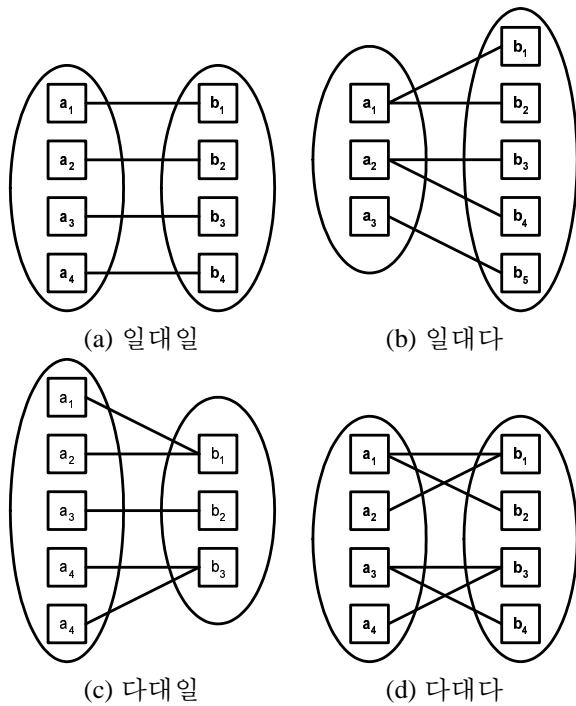
- **관계(relationship)**: 여러 개체 사이의 연관성
- **관계 집합(relationship set)**: 같은 종류의 관계들의 집합
- 관계 집합의 형식화된 정의: 관계 집합은 $n(\geq 2)$ 개의 개체 집합(중복 가능) 사이의 다음과 같은 수학적 관계를 말한다. E_1, \dots, E_n 이 개체 집합이면 관계 집합 R 은 다음과 같은 집합의 부분 집합이다.

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

여기서 (e_1, \dots, e_n) 은 관계이다.

- 이 때 개체 집합 E_1, \dots, E_n 은 관계 집합 R 에 **참여(participate)**한다고 한다.

- **관계 인스턴스**: 관계 집합의 특정한 원소
- **역할(role)**: 관계에서 개체가 행하는 기능
 - 보통 역할은 관계 집합을 통해 쉽게 알 수 있으므로 표시하지 않지만 한 개체가 어떤 관계 집합에서 여러 역할을 할 경우에는 역할을 표시한다. 이렇게 한 개체가 여러 역할을 하면 재귀적 관계 집합(recursive relationship set)이라 한다.
- 관계도 관계를 설명하는 속성(descriptive attribute)을 가질 수 있다.
- 두 개체 집합이 관련된 관계 집합을 이진 관계 집합(binary relationship set)이라 한다. 보통 관계 집합은 이진 관계 집합이지만 두 개체 집합 이상이 참여하는 관계도 있다.
- 관계 집합에 참여하는 개체 집합의 수를 관계 집합의 **차수(degree)**라 한다.



<그림 2.1> 대응수

2.2 제약 조건

2.2.1 대응수

- **대응수(mapping cardinality)**: 관계 집합을 통해 다른 개체와 관련될 수 있는 개체의 수
- 개체 집합 A 와 B 간에 이진 관계 집합 R 의 대응수의 종류
 - 일대일: A 의 한 개체는 최대한 B 의 한 개체와 연관되며, B 의 한 개체도 최대한 A 의 한 개체와 연관된다.
 - 일대다: A 의 한 개체는 임의의 수(0 또는 그 이상)의 B 개체와 연관되지만 B 의 한 개체는 최대한 A 의 한 개체와 연관된다.
 - 다대일: A 의 한 개체는 최대한 B 의 한 개체와 연관되지만, B 의 한 개체는 임의의 수(0 또는 그 이상)의 A 개체와 연관된다.
 - 다대다: A 의 한 개체는 임의의 수(0 또는 그 이상)의 B 개체와 연관되며, B 의 한 개체도 임의의 수(0 또는 그 이상)의 A 개체와 연관된다.

2.2.2 참여 제약 조건

- **전체 참가(total participation)**: E 의 모든 개체가 관계 집합 R 에 적어도 한 관계에 참여하는 경우
- **부분 참가(partial participation)**: E 의 일부 개체만 관계 집합 R 에 참여하는 경우

2.3 키

- 개체 집합 내에 있는 각 개체를 독특하게 식별할 수 있어야 한다. 따라서 개체 집합 내에 있는 어떤 두 개체도 모든 속성의 값이 같을 수 없다.
- **키(key)**: 개체들을 서로 구별할 수 있는 속성들의 집합

2.3.1 개체 집합

- **수퍼키(super key)**: 개체를 개체 집합에서 독특하게 식별해줄 수 있는 하나 또는 여러 개의 속성의 집합
- K 가 수퍼키이면 이것을 포함하는 속성의 집합은 항상 수퍼키가 된다.
- **후보키(candidate key)**: 수퍼키이면서 그것의 어떤 진부분집합도 수퍼키가 될 수 없는 키
 - *minimal key not minimum key*
- **주키(primary key)**: 후보키 중에 데이터베이스 설계자가 개체를 식별하기 위한 방법으로 선택한 키
- 키는 개별 개체에 의해 결정되는 것은 아니고 개체 집합에 의해 결정된다.
- 저장과 질의 처리의 효율성을 위해 주키는 값이 잘 변하지 않는 것으로 선택해야 한다.
 - 삽입시 같은 주키를 가진 개체가 있는지 검사해야 한다.
 - 주키는 널 값을 가질 수 없다.

2.3.2 관계 집합

- 개체 집합 E_1, \dots, E_n 이 참여하는 관계 집합 R 이 있고, $PK(E_i)$ 는 E_i 의 주키를 나타낸다고 하자. 또한 모든 개체 집합의 주키를 구성하는 속성 이름이 중복되지 않고, 각 개체가 관계에 한번만 참여한다고 하자.

- 관계 집합 R 이 속성을 가지지 않으면 R 의 각 관계는 다음에 의해 식별된다.

$$PK(E_1) \cup \dots \cup PK(E_n)$$

- 관계 집합 R 이 a_1, \dots, a_m 을 속성으로 가지면 다음과 같은 속성의 집합이 각 관계를 식별한다.

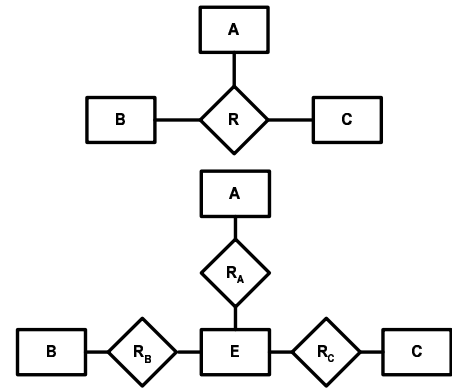
$$PK(E_1) \cup \dots \cup PK(E_n) \cup \{a_1, \dots, a_m\}$$

- 위 두 경우 모두 관계 집합 R 의 수퍼키는 다음과 같다.

$$PK(E_1) \cup \dots \cup PK(E_n)$$

- 속성의 이름이 중복된다면 속성의 이름을 재정의해야 한다.

- 개체가 여러 관계에 참여하면 개체 집합 이름 대신에 역할 이름을 사용하여 속성의 이름을 재정의한다.
- 관계 집합의 주키의 구조는 관계 집합의 대응수에 의해 결정된다.
- 이진 관계 집합에서 대응수에 따른 관계 집합의 주키
 - 일대일: 참여 개체 집합 중 어느 개체 집합의 주키도 관계 집합의 주키가 된다.
 - 일대다, 다대일: 그에 해당하는 개체 집합의 주키가 관계 집합의 주키가 된다.
 - 다대다: 참여 개체 집합의 주키의 합집합이 관계 집합의 주키가 된다.
- 이진 관계가 아니고 대응수 제약 조건이 없으면 참여하는 모든 개체 집합의 주키의 합집합이 유일한 관계 집합의 주키가 된다.



<그림 2.2> 3항 관계를 여러 개의 이진 관계로 바꾸는 방법

2.4 설계 논점

2.4.1 개체 집합 vs. 속성

- 다중값 속성 vs. 개체 집합
 - 속성에 대해 여러 정보를 유지하고 싶거나 여러 개체가 같은 속성 값을 공유하면 속성으로 사용하는 것보다 개체 집합으로 사용하는 것이 바람직하다.
- 개체 집합의 주키를 다른 개체 집합의 속성으로 사용하는 것은 바람직하지 않다. 이 경우에는 관계 집합을 이용해야 한다.
- 개체 집합의 주키를 관계 집합의 속성으로 사용하는 것은 잘못된 것이다. 이것은 관계 집합은 참여 개체 집합의 주키를 함축적으로 자신의 속성으로 가지고 있기 때문이다.

2.4.2 개체 집합 vs. 관계 집합

- 개체 집합 대신에 이 집합을 어떤 관계 집합의 속성으로 사용할 수 있다. 그러나 만들어진 관계 집합이 일대일 관계가 아니면 이것은 적절한 방법이 아니다. 그 이유는 다음과 같다.
 - 데이터의 중복으로 인한 저장 공간 낭비
 - 데이터의 중복으로 인한 일관성 위배 가능성
- 개체 간에 상호작용을 나타내는 정보는 관계 집합으로 모델링하는 것이 바람직하다.

2.4.3 이진 관계 집합 vs. n-항 관계 집합

- n-항 관계 집합은 여러 개의 이진 관계 집합으로 나타낼 수 있다.
- $n(> 2)$ -항 관계 집합을 여러 개의 이진 관계 집합으로 바꾸는 방법: 예) 개체 집합 A, B, C가 참여하는 $n = 3$ 인 관계 집합 R

- 단계 1. 관계 집합 R을 개체 집합 E로 바꾼다. 이때 R이 속성을 가지면 이 속성은 E의 속성이 된다. 또한 필요하다면 E의 각 개체를 구별할 수 있는 특수한 속성을 만든다. R의 각 관계 (a_i, b_i, c_i) 에 대해 E에 새로운 개체 e_i 를 추가한다.

- 단계 2. 다음과 같은 세 개의 관계 집합을 만든다.

- R_A : E와 A 간에 관계를 나타내는 관계 집합
 - R의 각 관계마다 (e_i, a_i) 를 R_A 에 추가한다.
- R_B : E와 B 간에 관계를 나타내는 관계 집합
- R_C : E와 C 간에 관계를 나타내는 관계 집합
- 모든 n-항 관계 집합은 여러 개의 이진 관계 집합으로 대체할 수 있지만 이렇게 하는 것이 항상 바람직한 것은 아니다.

- 관계 집합을 나타내기 위해 새롭게 만든 개체 집합의 각 개체를 식별하기 위한 새로운 속성을 만들어야 한다. 이 속성과 추가로 필요한 관계 집합 때문에 설계가 복잡해지고 저장 요구사항이 증가한다.

- n-항 관계 집합은 여러 개의 이진 관계 집합을 사용하였을 때보다 어떤 관계에 여러 개체가 참여한다는 것이 명확하다.

- n-항 관계에 있는 제약 조건을 이진 관계에 나타내는 것이 어려울 수 있다.

2.4.4 관계 속성의 위치

- 관계의 대응수는 관계 속성의 위치에 영향을 준다.
- 일대일 관계 집합 또는 일대다 관계 집합의 경우에 관계의 속성은 참여하는 개체 집합의 속성으로 설정할 수 있다.

- 일대일일 경우에는 참여하는 어느 개체 집합의 속성으로 사용할 수 있지만 일대다일 경우에는 다에 해당하는 개체 집합의 속성으로만 사용할 수 있다.
- 다대다 관계 집합에서 관계 집합의 속성 값이 관계에 참여하는 두 개체에 의해 결정되면 이 속성은 개체 집합의 속성으로 사용할 수 없다.

2.5 E-R 다이어그램

- E-R 다이어그램의 구성요소
 - 사각형: 개체 집합
 - 타원: 속성
 - 이중타원: 다중 값 속성
 - 점선 타원: 유도된 속성
 - 다이아몬드: 관계 집합
 - 이중 다이아몬드: 식별 관계 집합
 - 선: 속성을 개체 집합과, 개체 집합을 관계 집합과 연결할 때 사용
 - 이중선: 관계 집합에서 개체의 전체 참여를 나타낼 때 사용
 - 이중 사각형: 약한 개체 집합을 나타낼 때 사용
- 속성 중 주키는 밑줄로 나타낸다. 부분키는 점선 밑줄로 나타낸다.
- 대응수를 나타내기 위해 화살표 또는 직선을 사용한다. 화살표 방향은 일을 나타낸다.
- 역할은 개체 집합과 관계 집합을 연결하는 선에 이름을 붙여 나타낸다.
- 비이진 관계 집합에 연결된 선 중에 화살표선은 오직 하나만 존재해야 한다. 만약 여러 개 존재하면 두 가지 이상으로 해석할 수 있기 때문이다.
- 개체 집합 A_1, \dots, A_n 간에 관계 집합 R 에서 $A_{i+1}, A_{i+2}, \dots, A_n$ 의 끝에만 화살표가 있을 때 가능한 두 가지 해석
 - A_1, \dots, A_i 에서 특정 개체의 조합은 A_{i+1}, \dots, A_n 에서의 개체 조합 중에 최대 한 개와 연관될 수 있다. 이 경우 관계 집합 R 의 주키는 A_1, \dots, A_i 의 주키의 합집합이다.
 - 각 개체 집합 $A_k, i < k \leq n$ 에 대하여 A_k 와 다른 개체 조합은 최대 A_k 의 한 개체와 연관될 수 있다. 이 경우 $i < k \leq n$ 에 대해 $\{A_1, \dots, A_{k-1}, A_{k+1}, \dots, A_n\}$ 는 관계 집합 R 의 후보키가 된다.
- 개체 집합과 이진 관계 집합 간에 연결선에는 최소와 최대 대응수를 표시할 수 있다. 예) 0..*: 0이 최소이고 최대는 제한이 없다.

2.6 약한 개체 집합

- 약한 개체 집합(weak entity set): 주키를 형성할 수 있는 충분한 속성을 갖고 있지 않는 개체 집합
- 강한 개체 집합(strong entity set): 주키를 형성할 수 있는 충분한 속성을 갖고 있는 개체 집합
- 약한 개체 집합이 의미가 있기 위해서는 다른 개체 집합과 연관되어야 하며, 이 때 연관되는 개체 집합을 식별 개체 집합(identifying entity set) 또는 지배 개체 집합(owner entity set)이라 한다.
- 약한 개체 집합은 그것의 식별 개체 집합과 존재 종속 관계(existence dependent)가 있다고 하며, 이 관계를 식별 관계(identifying relationship)라 한다. 반대로 식별 개체 집합은 그것의 약한 개체 집합을 소유한다고 말한다.
- 식별 관계의 대응수는 약한 개체 집합에서 식별 개체 집합으로 다대일이며, 이 때 약한 개체 집합은 이 관계에 전체적으로 참여한다.
- 약한 개체 집합은 그것의 식별 개체 집합의 특정 개체와의 연관성에 따라 분할될 수 있다.
- 약한 개체 집합은 주키를 갖지 않지만 관련 식별 개체 집합의 한 개체에 종속하는 약한 개체 집합의 모든 개체를 서로 구별할 수 있는 속성의 집합이 있어야 한다. 이 속성의 집합을 구별자(discriminator) 또는 부분키(partial key)라 한다.
- 약한 개체 집합의 주키는 식별 개체 집합의 주키와 약한 개체 집합의 부분키를 합쳐 만든다.
- 식별 관계 집합은 기술적 속성을 가지지 않는다. 만약 필요하다면 이 속성은 약한 관계 집합의 속성으로 사용한다.
- 약한 개체 집합은 그것의 식별 개체 집합의 주키를 자신의 속성으로 사용하여 강한 개체 집합으로 바꿀 수 있다. 이렇게 하지 않는 이유는 다음과 같다.
 - 데이터의 중복성
 - 개체 간에 논리적 의존 관계를 잘 나타냄
 - 약한 개체 집합의 개체는 그것과 연관된 식별 개체 집합의 개체가 삭제되면 자동으로 삭제할 수 있다.
- 약한 개체 집합은 식별 관계 이외에 다른 관계에 참여할 수 있다.
 - 식별 관계 집합에 참여할 뿐만 아니라 다른 일반 관계 집합에 참여한다.
 - 또 다른 약한 개체 집합을 식별하는 개체 집합으로 참여한다.
 - 여러 개의 식별 개체 집합과 연관될 수 있다.

- 약한 관계 집합 대신에 이것을 식별 개체 집합의 다중 속성으로 사용할 수 있다. 약한 관계 집합의 속성 수가 적고, 이 집합이 식별 관계에만 참여하면 식별 개체 집합의 다중 속성으로 사용하는 것이 바람직하다.

2.7 E-R 모델의 확장

2.7.1 세분화

- 하나의 개체 집합은 집합 내의 다른 개체들과 어떤 방법으로 구분되는 개체들의 하위 그룹을 가질 수 있다.
- 개체 집합 내에 하위 그룹을 나누는 과정을 **세분화(specialization)**라 한다.
- 예) person 개체 집합은 하위 그룹으로 customer, employee를 가질 수 있다. 이 경우 customer와 employee 하위 그룹은 person의 모든 속성과 각자의 추가 속성을 가진다.
- 하위 그룹은 다시 또 하위 그룹으로 나눌 수 있다.
- E-R 다이어그램에서 세분화는 ISA라는 라벨을 가진 역삼각형 구성요소로 나타낸다.
- ISA 관계는 다른 말로 상위클래스(superclass)/하위클래스(subclass) 관계라고도 한다.

2.7.2 일반화

- 세분화는 하향식 설계(top-down desing) 방법이다. 반대로 상향식 설계(bottom-up design) 방법을 사용할 수 있다.
- 예) customer 개체 집합과 employee 개체 집합의 공통 분모를 뽑아 상위 클래스 person을 만들 수 있다.
- 이와 같이 두 개의 개체 집합의 공통 분모를 이용하여 상위 클래스를 만드는 과정을 **일반화(generalization)**라 한다.
- 일반화는 세분화의 역이다. 일반화는 유사성을 강조하는 것이고 세분화는 차이점을 강조하는 것이다.
- E-R 다이어그램에서는 일반화와 세분화를 구분하지 않는다.

2.7.3 속성의 상속

- 상위 클래스의 속성은 하위 클래스가 상속(inherit)한다고 한다.
- 하위 클래스는 또한 상위 클래스의 관계를 상속한다.
- 단일 상속(single inheritance): 하나의 상위 클래스만을 가질 경우
- 다중 상속(multiple inheritance): 여러 개의 상위 클래스를 가질 경우

2.7.4 일반화의 제약 조건

- 제약 조건 1. 어떤 개체가 하위 개체 집합의 구성원이 되기 위한 조건
 - 조건에 의한 정의: 개체가 어떤 명확한 조건을 만족할 경우
 - 예) account 개체 집합의 account-type 속성에 따른 분류
 - 사용자에 의한 정의: 사용자가 직접 개체를 분류하는 경우
- 제약 조건 2. 어떤 개체가 하나 이상의 하위 개체 집합의 구성원이 되기 위한 조건
 - 분리(disjoint): 한 개체는 오직 하나의 하위 개체 집합에만 속해야 하는 경우
 - 중첩(overlapping): 한 개체가 하나 이상의 하위 개체 집합에 속할 수 있는 경우
- 중첩 제약조건이 기본이며, 분리 제약조건은 E-R 다이어그램에서 역삼각형 기호 옆에 disjoint을 표시하여 나타낸다.
- 제약 조건 3. 상위 개체 집합의 개체가 반드시 하위 개체 집합 중 하나에 속해야 하는지 여부를 나타내는 조건, 완전성 제약조건(completeness constraint)
 - 전체 일반화(total generalization): 상위 개체 집합의 모든 개체가 반드시 하위 개체 집합 중 하나에 속해야 하는 경우
 - 부분 일반화(partial generalization): 상위 개체 집합의 개체 중 어떤 하위 개체 집합에도 속하지 않는 개체가 있을 수 있는 경우
- 완전성 제약조건에서는 부분 일반화가 기본이며, 전체 일반화는 E-R 다이어그램에서 상위 개체와 역삼각형 기호를 연결하는 선을 이중선으로 사용하여 나타낸다.
- 이런 제약 조건은 개체를 삽입하고 삭제할 때 영향을 준다.

2.7.5 통합화

- E-R 모델은 관계 간에 관계를 나타내지 못한다.
- 통합화(agggregation)는 관계를 상위 개체로 취급할 수 있도록 해주는 추상화 기법이다.
- E-R 다이어그램에서 통합화는 전체 관계를 큰 직사각형에 넣어 나타낸다.

2.8 E-R 데이터베이스 스키마의 설계

- 스키마 설계할 때 몇 가지 쟁점
 - 속성 vs. 개체 집합
 - 개체 집합 vs. 관계 집합
 - 3항 관계 vs. 여러 개의 이진 관계

- 강한 개체 집합 vs. 약한 관계 집합
- 일반화 사용 여부
- 통합화 사용 여부

2.8.1 스키마 설계 단계

- 단계 1. 사용자 요구사항 분석, 결과물: 사용자 요구사항 명세서
- 단계 2. 데이터 모델을 선택하고, 선택한 데이터 모델의 개념을 적용하여 사용자 요구사항을 데이터베이스의 개념적 스키마로 바꾼다.
- 단계 3. 기능적 요구사항 분석, 결과물: 기능적 요구사항 명세서
- 단계 4. 구현
 - 단계 4-1. 논리적 설계 단계
 - 단계 4-2. 물리적 설계 단계: 파일 구성 형식, 내부 저장 구조

- 방법 2. 일반화가 분리 제약조건과 완전성 제약조건을 만족하는 경우에는 상위 개체 집합을 위한 테이블은 만들지 않고, 각 하위 개체 집합마다 그 개체 집합의 고유 속성과 상위 개체 집합의 모든 속성으로 구성된 테이블을 만든다.

- 통합화: 통합화 관계 집합을 위한 테이블을 만든다. 이 테이블은 통합화 관계와 연관된 관계의 주키, 참여하는 개체 집합의 주키, 관계 집합의 속성으로 구성된다.

2.9 E-R 스키마를 테이블로 변환

- E-R 스키마를 테이블로 변환하는 일반적인 원칙은 각 개체 집합과 관계 집합을 하나의 테이블로 만드는 것이다.
- 강한 개체 집합: 개체의 각 속성마다 그 속성을 나타내는 열을 만든다.
- 약한 개체 집합: 약한 개체 집합의 각 속성과 그것의 종속 관계에 있는 강한 개체 집합의 주키를 구성하는 속성마다 그 속성을 나타내는 열을 만든다.
- 관계 집합: 관계 집합의 속성과 관계 집합에 참여하는 개체 집합의 주키를 구성하는 속성마다 그 속성을 나타내는 열을 만든다.
 - 식별 관계 집합은 데이터가 중복되므로 일반적으로 테이블로 만들 필요가 없다.
 - 개체 집합 A에서 개체 집합 B로의 다대일 관계 집합 R이 있으면 이것을 테이블로 나타내기 위해서는 3개의 테이블이 필요하다. 이 때 A가 R에 전체적으로 참여하면 테이블 A와 테이블 R을 결합할 수 있다.
- 복합 속성: 복합 속성을 구성하는 하위 속성마다 새로운 열을 만들어 나타낸다. 이 경우 복합 속성 자체는 열로 표현되지 않는다.
- 다중값 속성: 다중값 속성과 그 속성에 속한 개체 집합의 주키로 구성된 테이블을 만든다.
- 일반화를 테이블로 변환하는 방법
 - 방법 1. 상위 개체 집합을 나타내는 테이블과 각 하위 개체 집합마다 그 개체 집합의 고유 속성과 상위 개체 집합의 주키로 구성된 테이블을 만든다.