

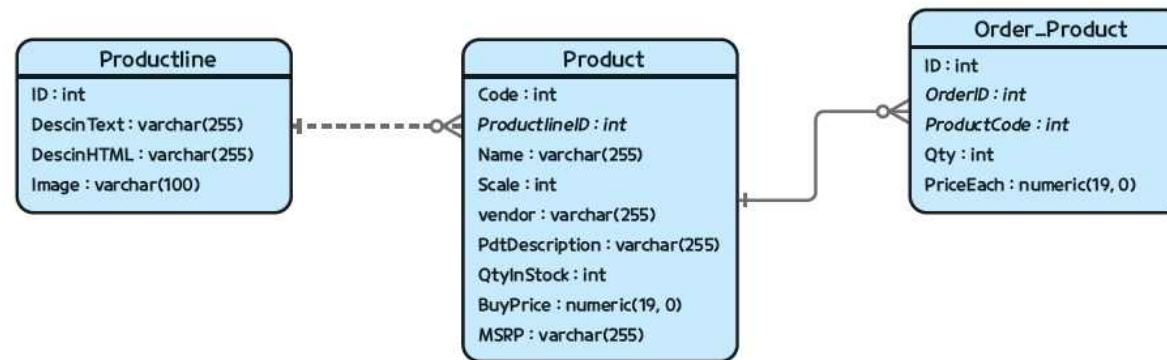
---

# 데이터 모델링 - 정규화

## 2. 데이터 모델의 종류

### ■ 관계형 데이터 모델(Relational Data Model)

- 계층 및 네트워크 데이터 모델에 대한 보다 유연한 대안으로 만들어짐
- 테이블 간의 관계를 나타내는 데 사용되며, 개체-관계형 데이터 모델에 비해 테이블 간의 관계를 도출하기가 좀 더 쉬움
- 관계형 데이터 모델에서 데이터는 테이블(릴레이션(Relation))에 저장됨
  - ✓ 테이블은 최소 0개 이상의 행(Row)과 1개 이상의 열(Column)로 구성됨
  - ✓ 열은 테이블의 항목들로 필드(Field), 속성(Attribute)이라고도 함
  - ✓ 행은 관련된 열의 집합으로 레코드(Record) 또는 튜플(Tuple)이라고도 함
  - ✓ 테이블의 각 레코드는 기본키(Primary Key) 속성에 의해 구별됨



# 데이터 모델링

# 1. 데이터 모델링의 개념

---

## ■ 데이터 모델링

- 현실 세계의 엔티티(개체), 속성, 관계를 분석하고 이를 데이터 모델로 변환하는 프로세스
- 데이터 모델링 과정에서는 비즈니스 요구사항을 이해하고 데이터 요구사항을 수집하며, 데이터의 구조와 관계를 정의하고 문서화함
- 데이터 모델링은 데이터 모델의 설계, 개발, 유지보수를 위한 체계적인 방법론을 사용하여 진행됨
- 데이터 모델링은 데이터베이스 설계와 밀접한 관련이 있으며 데이터베이스 시스템의 성능, 효율성 및 유지보수 용이성에 영향을 미침

# 1. 데이터 모델링의 개념

---

## ■ 데이터 모델링 프로세스

- ① 요구사항 분석
- ② 개념적 데이터 모델링
- ③ 논리적 데이터 모델링
- ④ 물리적 데이터 모델링
- ⑤ 구현 및 유지보수
- ⑥ 검증 및 최적화
- ⑦ 문서화

## 2. 요구사항 분석

---

### ■ 요구사항 분석

- 데이터 모델링의 시작 단계
- 시스템 또는 응용프로그램에서 필요한 사용자 및 시스템의 요구사항을 정의하고 문서화하는 단계
- 주요 산출물
  - ✓ 요구사항 분석서, 인터뷰 기록 문서 등

### 3. 개념적 데이터 모델링

---

- 개념적 데이터 모델링(Conceptual Data Modeling)
  - 요구사항 분석 결과를 기반으로 개념적 데이터 모델을 구축함
  - 특정 DBMS와 독립적인 개념적 스키마를 기술하는 단계로 업무 중심적이고 포괄적인 수준의 모델링을 진행함
  - 엔티티와 관계 위주의 모델링을 통해 전반적인 골격을 파악하는 데 중점을 둠
  - 이 단계에서 주제 영역 및 주요 엔티티, 주요 속성, 식별자와의 관계를 도출함
  - 개념적 데이터 모델은 ERD 등으로 표현할 수 있으며, 개념적 데이터 모델로 표현한 결과를 '개념적 스키마'라고 함

### 3. 개념적 데이터 모델링

---

#### ■ 주요 속성 도출

- 속성(Attribute): 엔티티 집합에 포함되는 최소의 데이터 단위
- 엔티티 집합의 각 엔티티는 각각의 속성에 대해 단일 속성값을 가져야 함

#### ■ 식별자 도출

- 식별자(Identifier): 하나의 엔티티 집합에서 각 엔티티를 구분할 수 있는 속성이나 속성의 집합
- 엔티티 집합은 반드시 식별자를 가져야 함



### 3. 개념적 데이터 모델링

---

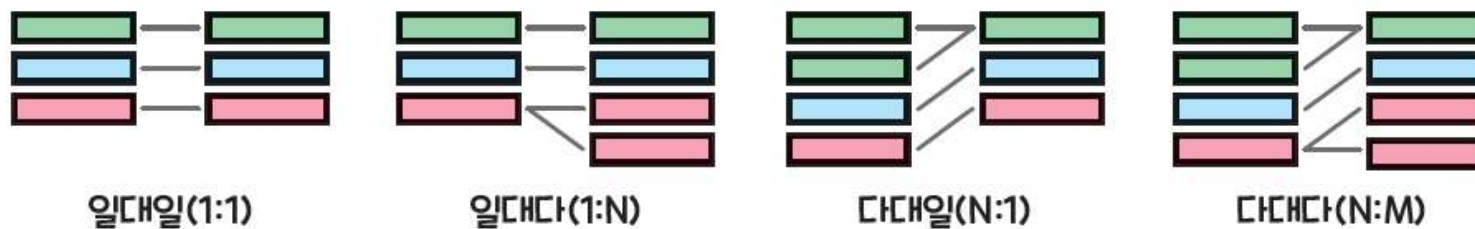
#### ■ 관계 도출

- 관계(Relationship): 엔티티 간의 관련성을 의미하는 것으로 존재 관계와 행위 관계로 구분할 수 있음
- 존재 관계(Existence Relationship)
  - ✓ 두 엔티티 간의 소속이나 소유의 관계
  - ✓ [예] 사원과 가족, 학과와 학생, 은행과 계좌 엔티티 간의 관계
- 행위 관계(Behavioral Relationship)
  - ✓ 두 엔티티 간의 상호작용이나 동작의 관계
  - ✓ [예] 교수와 학생, 고객과 주문 엔티티 간의 관계

### 3. 개념적 데이터 모델링

#### ■ 관계 도출

- 관계의 카디널리티(Cardinality, Degree)
  - ✓ 두 테이블 간의 관계에서 카디널리티는 대응수라고도 함
  - ✓ 한 테이블의 행이 다른 테이블에서 두 개 이상의 행과 연결될 수 있는지의 여부를 나타냄
  - ✓ 종류: 일대일(1:1), 일대다(1:N), 다대일(N:1), 다대다(N:M)



### 3. 개념적 데이터 모델링

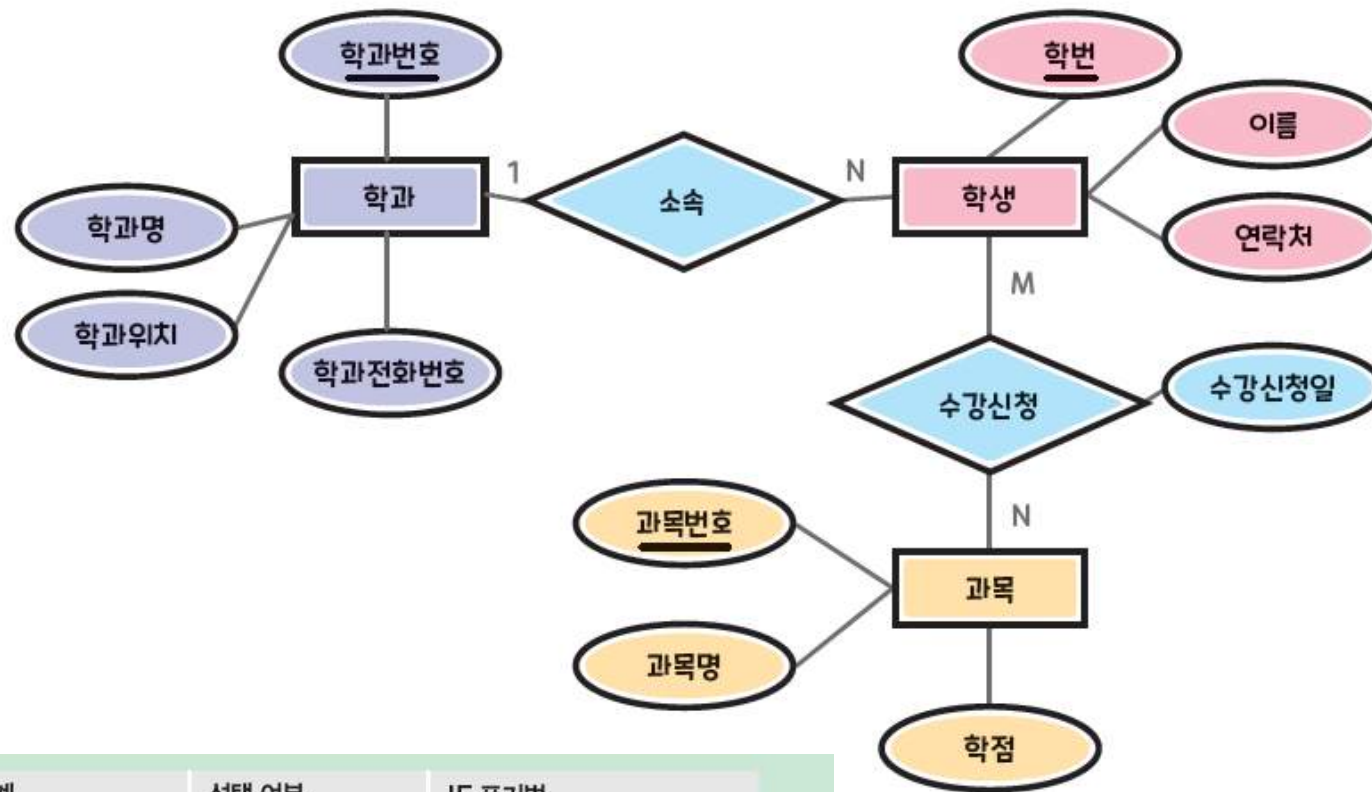
---

#### ■ 관계 도출

- 관계 이름(Membership)
  - ✓ 객체 간에 관계가 맺어지는 형태를 의미함
  - ✓ 관계 시작점(The Beginning): 관계가 시작되는 쪽의 엔티티
  - ✓ 관계 끝점(The End): 관계가 끝나는 쪽의 엔티티
- 관계 선택 사양(Optionality)
  - ✓ 선택적(Optional) 참여 관계: 두 엔티티의 인스턴스 간의 관계가 선택적으로 존재할 수 있음
  - ✓ 필수적(Mandatory) 참여 관계: 두 엔티티의 인스턴스 간에 관계가 반드시 존재해야 함

### 3. 개념적 데이터 모델링

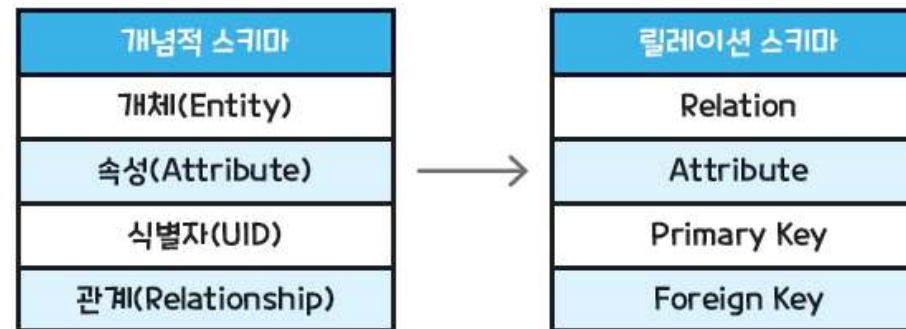
학과에 소속된 학생이 과목을 수강신청하는 시나리오에 대해 엔티티 간의 관계를 피터 첸 표기법으로 표현한 개념적 데이터 모델 ERD의 예는 다음과 같습니다.



관계	선택 여부	IE 표기법
1:1	필수	+ — +
1:1	선택	+ — ☉
1:N	필수	+ — ∞
1:N	선택	+ — ☉∞

## 4. 논리적 데이터 모델링

- 논리적 데이터 모델링(Logical Data Modeling)
  - 개념적 데이터 모델을 더 상세하게 정의하는 단계
  - 개념적 데이터 모델을 개발에 사용하는 특정 DBMS에 적합한 모델로 변환함
  - DBMS가 관계형 데이터 모델인 경우에는 개념적 스키마를 릴레이션 스키마 등으로 변환해야 함
  - 릴레이션 스키마와 같이 논리적 데이터 모델로 표현된 결과를 '논리적 스키마'라고 함



## 4. 논리적 데이터 모델링

---

- 논리적 데이터 모델링(Logical Data Modeling)
  - 개념적 스키마를 릴레이션 스키마로 변환하기 위한 규칙
    - ✓ 모든 엔티티는 릴레이션으로 변환함
    - ✓ 일대일 관계(1:1)는 외래키로 표현함
    - ✓ 일대다 관계(1:N)는 외래키로 표현함
    - ✓ 다대다 관계(N:M)는 릴레이션으로 변환함

## 4. 논리적 데이터 모델링

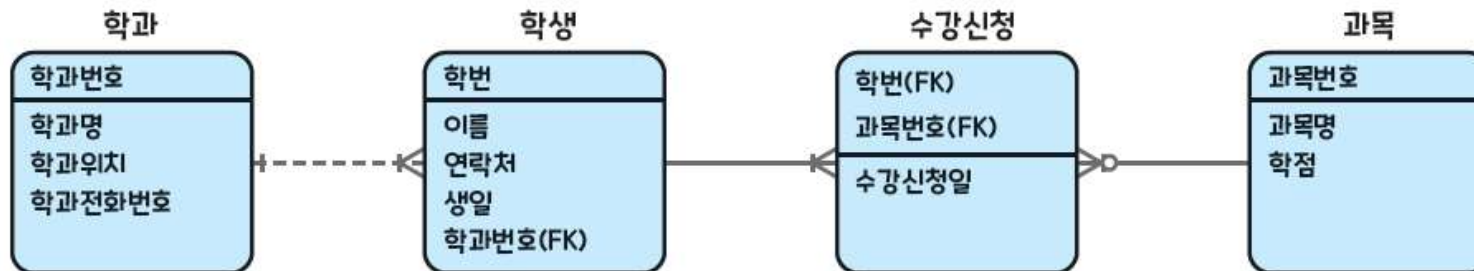
- 논리적 데이터 모델링(Logical Data Modeling)
  - 규칙을 적용하여 개념적 데이터 모델을 릴레이션 스키마로 변환한 결과

학과 {학과번호, 학과명, 학과위치, 학과전화번호}

학생 {학번, 이름, 생일, 학과번호}

과목 {과목번호, 과목명, 학점}

수강신청 {학번, 과목번호, 수강신청일}



## 4. 논리적 데이터 모델링

---

### ■ 논리적 데이터 모델링(Logical Data Modeling)

- 식별 관계와 비식별 관계

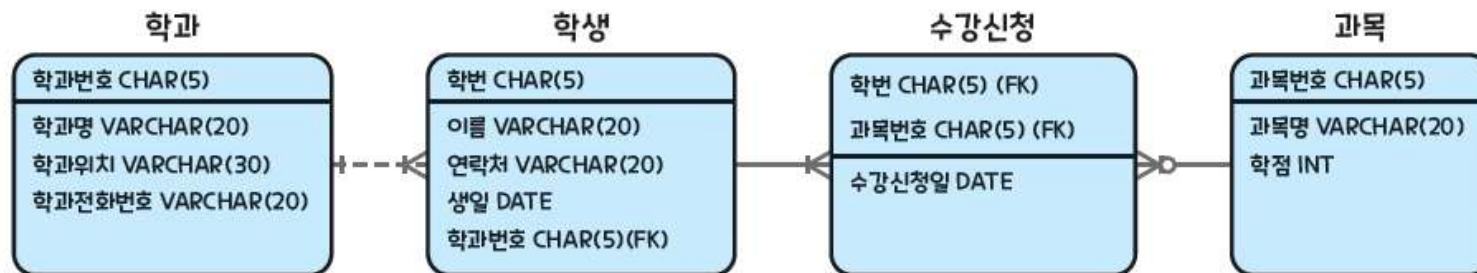
- ✓ 점선: 비식별 관계, 실선: 식별 관계
- ✓ 이 두 관계는 부모 릴레이션과 자식 릴레이션 간의 관계를 나타냄
- ✓ 식별 관계는 부모 릴레이션의 기본키가 자식 릴레이션의 기본키의 일부로 사용되는 관계로 두 릴레이션 간의 속성이 서로에 대해 필수적일 때 적합함
- ✓ 비식별 관계는 부모 릴레이션의 기본키가 자식 릴레이션의 기본키에 포함되지 않는 관계로, 두 릴레이션 간의 관계가 서로에 대해 필수 정보가 아닐 때 적합함



## 5. 물리적 데이터 모델링

### ■ 물리적 데이터 모델링(Physical Data Modeling)

- 논리적 데이터 모델을 구체적인 데이터베이스 시스템에 맞게 변환함
- 테이블에 대해 기본키 컬럼 순서, 영문 컬럼명 매핑, 데이터타입 및 크기 등을 결정함
- 또한 인덱스, 참조 무결성 규칙 등의 제약조건, 파티셔닝과 같은 세부 사항을 정의하고, 저장 구조와 접근 경로 등을 결정함



## 6. 구현 및 유지보수

---

### ■ 구현 단계 및 유지보수

- 물리적 데이터 모델을 기반으로 데이터베이스를 생성함
- 이후 데이터베이스 시스템을 운영하고 유지보수하며, 필요에 따라 모델을 조정하거나 확장함
- 주요 산출물: 데이터베이스 데이터베이스 객체 정의문, 데이터 마이그레이션 계획서 등

## 7. 검증 및 최적화, 문서화

---

### ■ 검증 및 최적화, 문서화

- 데이터 모델링 단계가 완료된 후에는 모델의 검증과 최적화를 수행함
- 데이터 모델의 정확성, 일관성, 성능 등을 평가하고, 필요하다면 수정 및 개선 작업을 진행함
- 문서화는 각 단계에서 산출된 문서를 정리하여 모델을 이해하고 유지보수할 수 있도록 만들어줌

# 정규화

# 1. 이상

---

## ■ 이상(Anomaly)

- 관계형 데이터베이스에서 데이터 삽입, 삭제 및 수정과 관련하여 발생할 수 있는 불일치 또는 오류
- 데이터베이스의 정상적인 동작을 방해하거나 데이터의 무결성과 일관성을 저해할 수 있음

## ■ 이상의 유형

- 삽입 이상(Insertion Anomaly)
  - ✓ 데이터를 삽입할 때 발생하는 이상으로 데이터를 삽입할 때 원하지 않는 불필요한 데이터를 반복해서 입력해야 하는 경우
- 삭제 이상(Deletion Anomaly)
  - ✓ 데이터를 삭제할 때 발생하는 이상으로 테이블에서 일부 레코드를 삭제하면 삭제할 필요가 없는 다른 정보도 함께 삭제되는 경우
- 수정 이상(Update Anomaly)
  - ✓ 데이터를 변경할 때 발생하는 이상으로 테이블의 일부 레코드의 값만 수정되어 데이터 일관성이 깨지는 경우

## 1. 이상

- 동아리, 학생, 학과 정보를 한번에 관리하도록 설계된 동아리가입학생학과 테이블

동아리번호	동아리명	동아리개설일	학번	이름	동아리가입일	학과번호	학과명
C1	지구한바퀴여행	2000-02-01	231001	문지영	2023-04-01	D1	화학공학과
C1	지구한바퀴여행	2000-02-01	231002	배경민	2023-04-03	D4	경영학과
C2	클래식연주동아리	2010-06-05	232001	김명희	2023-03-22	D2	통계학과
C3	위너비글퍼	2020-03-01	232002	전은정	2023-03-07	D2	통계학과
C3	위너비글퍼	2020-03-01	231002	배경민	2023-04-02	D4	경영학과
C4	댄스배틀	2021-07-01	231001	문지영	2023-04-30	D1	화학공학과
C4	댄스배틀	2021-07-01	233001	이현경	2023-03-27	D3	역사학과

## 2. 함수적 종속성

---

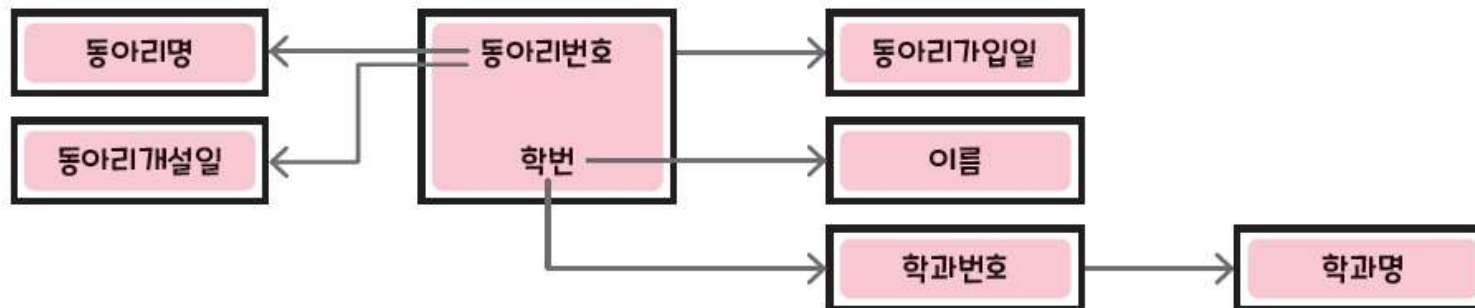
- 함수적 종속성(Functional Dependency, FD)
  - 관계형 데이터베이스의 릴레이션에서 속성 간의 관계
  - 특정 릴레이션에서 속성 X와 속성 Y 사이에 함수적 종속성이 존재한다
    - ✓ = X의 값이 주어지면 Y의 값이 항상 유일하게 결정된다
    - ✓  $X \rightarrow Y$
    - ✓ X: 결정자(Determinant), Y: 종속자(Dependent)

## 2. 함수적 종속성

### ■ 함수적 종속성(Functional Dependency, FD)

- 함수적 종속성 다이어그램(Functional Dependency Diagram, FDD)

- ✓ 함수적 종속성을 나타내는 표기법
- ✓ 이것을 통해 속성 간의 관계를 쉽게 파악할 수 있음



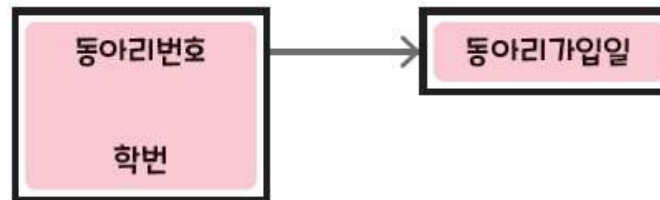
동아리번호 → 동아리명  
동아리번호 → 동아리개설일  
{동아리번호, 학번} → 동아리가입일  
학번 → 이름  
학번 → 학과번호  
학과번호 → 학과명



## 2. 함수적 종속성

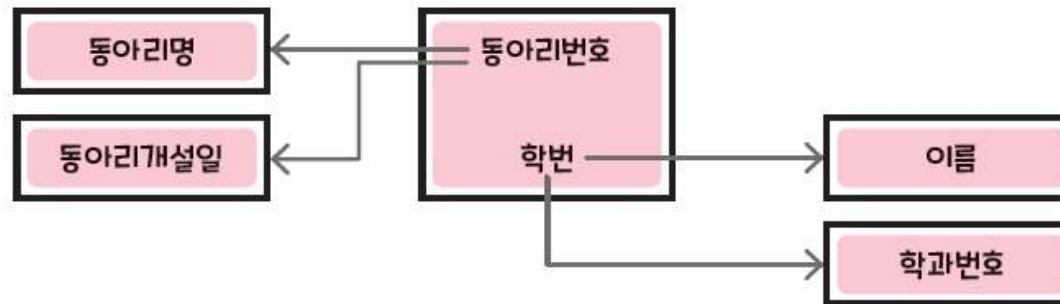
### ■ 완전 함수적 종속성(Full Functional Dependency)

- 기본키가 아닌 속성들이 기본키에 종속이 되는 종속 관계
- 만일 기본키가 복합키인 경우에는 기본키를 구성하는 속성 집합에 종속되어야 함



### ■ 부분 함수적 종속성(Partial Functional Dependency)

- 기본키가 아닌 속성들이 기본키의 일부에 종속되는 관계



## 2. 함수적 종속성

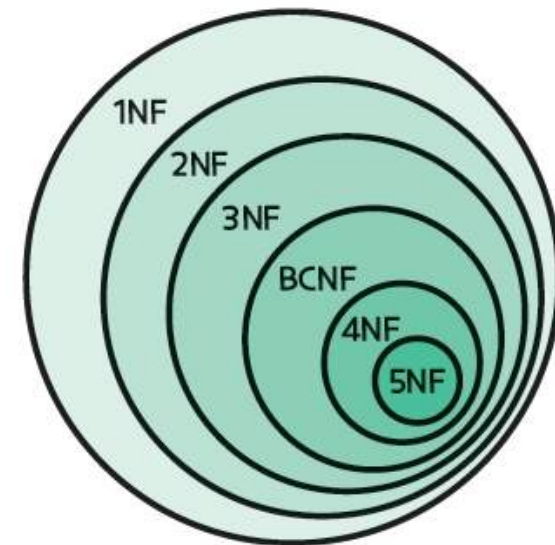
- 이행 함수적 종속성(Transitive Functional Dependency)
  - 세 개 이상의 속성이 연결되어 있을 때 발생하는 종속 관계
  - X, Y, Z라는 3개의 속성이 있을 때  $X \rightarrow Y$ ,  $Y \rightarrow Z$ 와 같이 속성 간에 종속성이 연쇄적으로 전파되는 경우
  - Y가 X에 함수적으로 종속되고, Z가 Y에 함수적으로 종속되면, Z는 X에 대해 이행 함수적으로 종속됨



### 3. 정규화의 개념과 형식

#### ■ 데이터베이스 정규화(Normalization)

- 데이터베이스 설계의 중요한 개념으로 이상을 제거하여 데이터베이스를 효율적으로 구조화하는 과정
- 정규화를 통해 데이터베이스를 구조화하면 데이터 삽입 이상, 수정 이상, 삭제 이상이 최소화됨
- 그 결과 데이터의 중복을 줄일 수 있고, 데이터 무결성을 강화하며 일관성을 유지할 수 있음
- 또한 데이터의 논리적 구조가 좀 더 간결하고 명확하게 표현되므로 데이터를 검색하는 데 효율적이며, 유지보수성이 향상됨



### 3. 정규화의 개념과 형식

---

#### ■ 제1정규형(1NF)

- 테이블의 각 속성 값은 반복 그룹이 없는 원자 값으로만 구성되어야 함
- 이를 위해서 중복되는 속성이나 속성 그룹은 별도의 테이블로 분리함

#### ■ 제2정규형(2NF)

- 제2정규형은 제1정규형을 만족하면서 부분 함수적 종속성을 제거해야 함

#### ■ 제3정규형(3NF)

- 제3정규형은 제2정규형을 만족하면서 이행 함수적 종속성을 제거해야 함

#### ■ 추가적인 정규화 형식

- 보이스-코드 정규형(Boyce-Codd NF, BCNF), 제4정규형(4NF), 제5정규형(5NF) 등
- 실무에서 데이터베이스 정규화는 주로 제3정규형까지 진행함

## 4. 반정규화

---

### ■ 반정규화(Denormalization)

- 중복 데이터를 하나 이상의 테이블에 추가하는 데이터베이스 최적화 기술
- 반정규화의 장점
  - ✓ 조인 개수를 줄일 수 있고, 이로 인해 검색 쿼리가 간단해짐
  - ✓ 빠른 읽기 작업의 수행으로 인해 검색 성능을 높일 수 있음
- 반정규화의 단점
  - ✓ 반정규화는 중복 저장으로 인해 더 많은 스토리지가 필요함
  - ✓ 데이터 불일치가 발생할 수 있음
  - ✓ 쓰기 작업에 추가적인 비용이 소요될 수 있음

## 4. 반정규화

---

- 동아리가입학생학과 테이블에 대해 제1정규형을 적용하시오.
  - 정규화 대상인 동아리가입학생학과 테이블
- 동아리가입학생학과 테이블

## 4. 반정규화

---

- 동아리가입학생학과 테이블에 대해 제1정규형을 적용하시오.
  - 제1정규형을 만족하려면 반복 속성 그룹을 별도의 테이블로 분할해야 함
- 분할된 동아리 테이블과 동아리가입학생학과 테이블 간의 카디널리티는 일대다

## 4. 반정규화

---

- 동아리가입학생학과 테이블에 대해 제1정규형을 적용하시오.
  - 동아리 테이블
- 동아리가입학생학과 테이블



## 4. 반정규화

---

- 동아리가입학생학과 테이블에 대해 제2정규형을 적용하시오
  - 제2정규형을 만족하기 위해서는 테이블 분할을 통해 부분 함수적 종속성을 제거해야 함
- 분할된 학생학과 테이블과 동아리가입 테이블 간의 카디널리티는 일대다

## 4. 반정규화

### ■ 동아리가입학생학과 테이블에 대해 제2정규형을 적용하시오

#### • 학생학과 테이블

학번	이름	학과번호	학과명
231001	문지영	D1	화학공학과
231002	배경민	D4	경영학과
232001	김명희	D2	통계학과
232002	전은정	D2	통계학과
233001	이현경	D3	역사학과

#### • 동아리가입 테이블

동아리번호	학번	동아리가입일
C1	231001	2023-04-01
C1	231002	2023-04-03
C2	232001	2023-03-22
C3	232002	2023-03-07
C3	231002	2023-04-02
C4	231001	2023-04-30
C4	233001	2023-03-27

## 4. 반정규화

### ■ 학생학과 테이블에 대해 제3정규형을 적용하시오

- 동아리가입 테이블은 학번 → 학과번호, 학과번호 → 학과명과 같이 속성의 종속성이 연쇄 발생하고 있어 테이블을 분할하여 종속성을 제거해야 함

- 분할된 학과 테이블과 학생 테이블 간의 카디널리티는 일대다
  - ✓ 이 경우에 학과 테이블의 기본키인 학과번호를 학생 테이블에 추가한다.

#### • 학과 테이블

학과번호	학과명
D1	화학공학과
D2	통계학과
D3	역사학과
D4	경영학과

#### • 학과 테이블

학번	이름	학과번호
231001	문지영	D1
231002	배경민	D4
232001	김명희	D2
232002	전은정	D2
233001	이현경	D3

## 4. 반정규화

---

- 학생학과 테이블에 대해 제3정규형을 적용하시오

## 4. 반정규화

---

- 학생학과 테이블에 대해 제3정규형을 적용하시오
  - 제3정규형을 만족하는 정규화 결과