

RDB(관계형 데이터베이스) 테이블 설계는 데이터를 효율적으로 저장하고 관리할 수 있도록 구조를 설계하는 과정입니다. 다음은 RDB 테이블을 설계하는 기본적인 단계입니다.

## 1. 요구사항 분석

- 먼저 시스템에서 관리해야 할 데이터와 그 데이터를 사용하는 방식에 대해 분석합니다. 어떤 데이터를 저장하고, 어떻게 사용할지, 어떤 관계가 있는지 파악합니다.
- 예: 고객 관리 시스템에서는 고객 정보, 주문 정보, 제품 정보 등이 필요할 수 있습니다.

## 2. 엔터티(Entity) 도출

- 요구사항을 분석하여 저장할 주요 객체들을 엔터티로 정의합니다. 엔터티는 실제로 존재하는 객체나 개념을 의미하며, 이는 테이블로 변환됩니다.
- 예: 고객(Customer), 주문(Order), 제품(Product) 등의 엔터티를 정의할 수 있습니다.

## 3. 속성(Attribute) 정의

- 각 엔터티가 가져야 할 속성을 정의합니다. 속성은 테이블의 컬럼이 됩니다.
- 예: 고객 엔터티는 이름, 이메일, 전화번호 등의 속성을 가질 수 있습니다.

## 4. 관계 정의

- 엔터티 간의 관계를 정의합니다. 관계는 외래 키(Foreign Key)를 통해 나타낼 수 있습니다.
  - **1:1 관계**: 한 엔터티의 레코드가 다른 엔터티의 레코드 하나와만 연관될 때.
  - **1:N 관계**: 한 엔터티의 레코드가 여러 개의 다른 엔터티 레코드와 연관될 때.
  - **N:M 관계**: 두 엔터티의 레코드들이 서로 다수의 레코드와 연관될 때. 이를 해결하기 위해 중간 테이블을 생성합니다.
- 예: 고객과 주문은 1:N 관계, 주문과 제품은 N:M 관계일 수 있습니다.

## 5. 정규화(Normalization)

- 데이터를 중복 없이 효율적으로 저장하기 위해 정규화를 진행합니다. 일반적으로 3차 정규형까지 적용하는 경우가 많습니다.
  - **1차 정규형(1NF)**: 테이블의 각 컬럼이 원자값(Atomic value)을 가져야 하며, 중복된 데이터가 없어야 합니다.
  - **2차 정규형(2NF)**: 1차 정규형을 만족하고, 부분 종속이 없어야 합니다.
  - **3차 정규형(3NF)**: 2차 정규형을 만족하고, 이행적 종속이 없어야 합니다.

- 예: 고객 테이블에 주소 정보가 여러 번 중복된다면 주소를 별도 테이블로 분리할 수 있습니다.

## 6. 인덱스 및 성능 최적화

- 데이터 검색 속도를 높이기 위해 인덱스를 고려합니다. 자주 조회되는 컬럼이나 외래 키가 있는 컬럼에 인덱스를 설정할 수 있습니다.

## 7. 데이터 무결성 및 제약조건 설정

- 각 테이블의 컬럼에 적절한 제약조건을 설정하여 데이터 무결성을 보장합니다.
  - Primary Key:** 각 테이블에서 중복되지 않는 고유한 값.
  - Foreign Key:** 다른 테이블의 Primary Key를 참조하여 엔터티 간의 관계를 표현.
  - Not Null:** 값이 반드시 존재해야 하는 속성.
  - Unique:** 값이 중복되지 않아야 하는 속성.

## 8. ERD(Entity Relationship Diagram) 작성

- 엔터티 간의 관계를 시각적으로 표현한 ERD를 작성합니다. 이를 통해 전체적인 데이터 구조를 한눈에 파악할 수 있습니다.

## 예시

### 고객-주문 관계

#### 1. Customer 테이블

- `customer_id` (Primary Key)
- `name`
- `email`

#### 2. Order 테이블

- `order_id` (Primary Key)
- `order_date`
- `customer_id` (Foreign Key)
- `total_price`

이러한 과정을 통해 설계된 테이블은 데이터의 중복을 최소화하고, 유지 보수성을 높이며, 성능을 최적화할 수 있습니다.