

# 9주차

## [9주차] 전산 기본

- JOIN에 대해서 설명해 주세요. 정진님

두개 이상의 테이블을 묶어서 하나의 결과를 만들어 내는 것

⇒ 여러 테이블에서 가져온 데이터들을 조합하여 하나의 테이블이나 결과 집합으로 표현하는 것

inner join 교집합

left, right join 부분집합

outer join 합집합

cross join

+데이터베이스 설계시 join을 고려해야한다

방법은?

기본키와 외래키의 관계로 맺어 추출

그이유는?

데이터의 중복을 피하고 데이터 정규화를 잘 수행하여 데이터의 일관성과 무결성을 유지하기 위해

- 내부 조인과 외부 조인의 차이는 무엇인가요?

-내부조인

흔히 조인이라할때 말하는 게 내부조인

두 테이블에서 공통으로 존재하는 열의 데이터만 뽑기

-외부조인

1개의 테이블에만 값이 있어도 결과를 추출

빈 내용은 null로 채우기

- 스키마란 무엇인가요?

데이터베이스의 구조와 제약조건에 관한 전반적인 명세

데이터 개체(Entity), 개체의 특성을 나타내는 속성(Attribute), 개체 사이에 존재하는 관계(Relationship) 및 데이터 조작 시 데이터 값들이 갖는 제약 조건 등에 관하여 기술

## -외부 스키마

개인의 입장에서 필요한 데이터베이스의 논리적 구조  
하나의 데이터베이스 시스템에는 여러 개의 외부 스키마가 존재

## -개념 스키마

데이터베이스의 전체적인 논리적 구조  
조직 전체의 데이터베이스로 하나만 존재  
일에 저장되는 데이터의 형태를 나타내는 것으로, 단순히 스키마라고 하면 개념 스키마를 의미  
DBA에 의해서 구성

## -내부 스키마

물리적인 저장장치 입장에서 데이터가 저장되는 방법을 기술  
실제 데이터베이스에 저장될 레코드의 물리적인 구조를 정의  
저장 데이터 항목의 표현방법, 내부 레코드의 물리적 순서, 인덱스 유/무 등

- 정규화에 대해서 설명해 주세요.

### 정규화 (1NF, 2NF, 3NF, BCNF, 4NF, 5NF)

1. 정의 Normalization 관계형 데이터 모델에서 데이터의 중복성을 제거하여 이상 현상(Anomaly)을 방지하고, 데이터의 일관성과 정확성을 유지하기 위해 무손실 분해하는 과정을 말한다. > ##### 💡 이상 (Anomaly)

 <https://velog.io/@wisdom-one/정규화Normalization>

### 정규화

Normalization

Database

정규화:

데이터를 일정한 기준 또는 범위로 조정하는 과정

테이블 간에 중복된 데이터를 허용하지 않는 것 → 무결성 보장

수정, 삭제 시 이상 현상을 방지함으로써 데이터 구조의 일관성을 최대화

정규화를 통해 데이터를 표준화하면 모델의 성능을 향상시키거나 데이터 분석의 결과를 더욱 정확하게

제1정규화:

-테이블의 컬럼이 하나의 값만 갖는 것, 원자성

-하나의 컬럼은 같은 종류나 타입(type)의 값

제2정규화: 제1정규화+ 부분적 종속 제거(=완전 함수 종속), 기본키의 부분 집합이 다른 키값의 결정자가 되어선 안됨

제3정규화: 제2정규화+ 이행정 종속 제거,  $a \rightarrow b \rightarrow c$ 일때  $a \rightarrow c$ 인 거 제거,  $a \rightarrow b$ ,  $b \rightarrow c$ 분리로 해결

BCNF: 제3정규화+ 모든 결정자가 후보키 집합에 속함

→ 후보키 집합에 없는 컬럼이 결정자가 되어선 안된다.

제4정규화: BCNF+다치 종속 해결

다치종속: 두개의 독립된 애트리뷰트가 1:N 관계로 대응하는 관계

제5정규화: 제4정규화+ 조인 종속 제거

하나의 릴레이션을 여러개의 릴레이션으로 분해하였다가, 다시 조인했을 때 데이터 손실이 없고 필요없는 데이터가 생기는 것

정규화의 장점

-데이터베이스 변경시 이상 현상을 제거할 수 있다

정규화의 단점

-조인 연산이 많아진다

디노멀라이제이션

- 파티셔닝과 샤딩에 대해서 설명해 주세요.

파티셔닝: 물리적으로 여러 테이블로 분리되지만 논리적으로 하나의 테이블 접근

파티셔닝 기준

-데이터값이 특정 목록에 포함되는 경우 분리 ex.지역별

-범위

-해시

샤딩: 동일한 스키마를 가지고 있는 여러대의 데이터베이스 서버들에 데이터를 샤드(작은단위)로 나누어 분산

물리적으로 분산된 컴퓨터 환경에서 사용

애플리케이션 차원에서 구현하는 것이 일반적

독립된 데이터베이스 서버에 분할하여 저장하므로 여러 샤드에 걸쳐 조인이 어렵다

한 데이터베이스 서버에 집중되지 않도록 조심

샤딩 기준

-해시, pk 값의 모듈러 연산 값을 기준으로 저장

-범위, pk 값의 범위를 지정하여 샤드 지정

- **ORM이란 무엇인가요?**

Object Relational Mapping(객체-관계-매핑)

객체와 데이터베이스의 관계를 매핑

프로그래밍 언어 객체와 관계형 데이터베이스의 데이터를 자동 매핑

객체간 관계를 바탕으로 sql문을 자동으로 생성해서 불일치 해결

ORM 장점

- 비즈니스 로직 집중 가능

- DBMS에 의한 종속성 저하

ORM 단점

- 완벽한 ORM 으로만 서비스를 구현하기가 어렵

- **NoSQL이란 무엇인가요?**

NoSQL은 "Not Only SQL"의 약자로, 관계형 데이터베이스 관리 시스템(RDBMS)이 아닌 다른 형태의 데이터 저장 및 처리 기술을 가리키는 용어

주로 대규모의 분산 데이터 처리와 관련하여 설계

확장성, 유연성 강조

- 비정형 데이터 지원

- 분산 데이터베이스

- 높은 읽기/쓰기 성능

- 수평적 확장성

## **NoSQL의 종류**

Key-Value Store: 간단한 키와 값으로 구성된 데이터베이스로, Redis, Riak 등

Document Store: 비정형 데이터를 JSON 또는 BSON 형태로 저장하는 데이터베이스로, MongoDB, Couchbase 등

Column Family Store: 열 지향 데이터 저장 방식으로, HBase, Cassandra 등

Graph Database: 데이터의 관계를 그래프로 표현하고 처리하는 데이터베이스로, Neo4j, ArangoDB 등

## **+RDBMS 관계형 데이터베이스**

- 구조화된 데이터

- ACID (Atomicity, Consistency, Isolation, Durability) 트랜잭션을 지원하여 데이터의 무결성과 일관성

