
WAFFLE
STUDIO

와플스튜디오 Backend Seminar

Instructor: 강지혁

2022.09.20.(화) 19:30

Session 1

Table of Contents

WA#LE
STUDIO

Review

우리가 한 일

- 서버, HTTP, 스프링 이해하기
- 필요한 컴포넌트 (Bean) 만들고, 사용하는 법 이해하기
- 요청 받고, 적절히 처리하기

WA#LE
STUDIO

Review

더 잘 만들 수 없을까?

값들을 추가하거나, 수정하기

서버가 껐다 켜져도 데이터가 유실되지 않기

허가된 유저에게만 API 허락하기

등등 ..

WA LE
STUDIO

Persistence

"영속성" : 데이터를 생성한 프로그램이 꺼져도, 데이터는 유지되는 성질

어떻게 유지할까요?

- 파일 시스템 (txt, csv, ..)
- 별도의 데이터베이스
- 스프레드시트, 클라우드 드라이브 (S3, Google Drive ..)

WA LE
STUDIO

Database

데이터 저장소

DBMS 통해 접근 & 관리

- RDB : “관계형” 데이터베이스
- SQL : RDBMS에서 사용하는 데이터 질의 언어
- NoSQL : 관계형 DB일 필요가 있을까?
  - Document DB, Key-Value DB 등등..

WA LE
STUDIO

사전 준비

DataGrip 설치 (우리는 MySQL로 실습할 거에요)

Docker, Docker-Compose 설치

MySQL 로컬 Docker 서버 실행

```
➜  seminar1 git:(main) ✗ docker-compose up -d
[+] Running 1/1
 ⠿ Container seminar1-db-1  Started
```

WA#LE STUDIO

RDB

데이터는 하나의 테이블로 표현할 수 있다.

데이터 사이에는 “관계”가 존재할 수 있다.

- 데이터 사이에 관계가 왜 필요할까요?

기본키(Primary Key)　　외부키(Foreign Key)　　기본키(Primary Key)

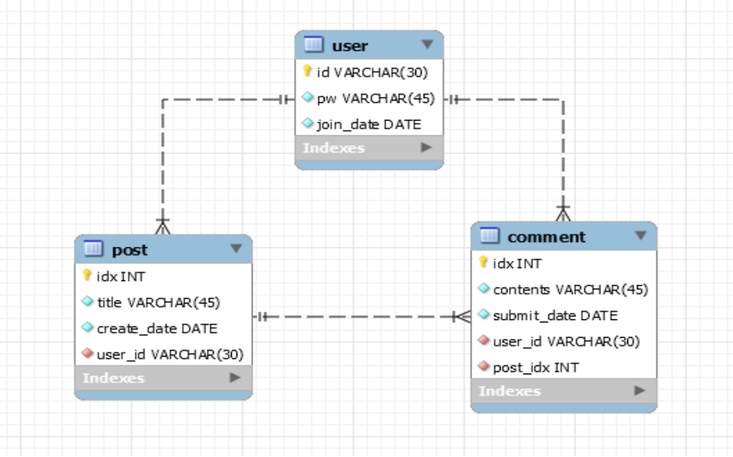
선수번호	이름	팀 코드	포지션	등번호	키
1	김남일	K03	DF	33	177
2	박지성	K07	MF	7	178
3	이영표	K02	MF	22	176

〈선수 테이블〉

팀 코드	팀 명	연고지
K03	스틸러스	포항
K07	드래곤즈	전남
K02	블루윙즈	수원

〈구단 테이블〉

[그림 Ⅱ-1-5] 테이블의 정규화



WA LE
STUDIO

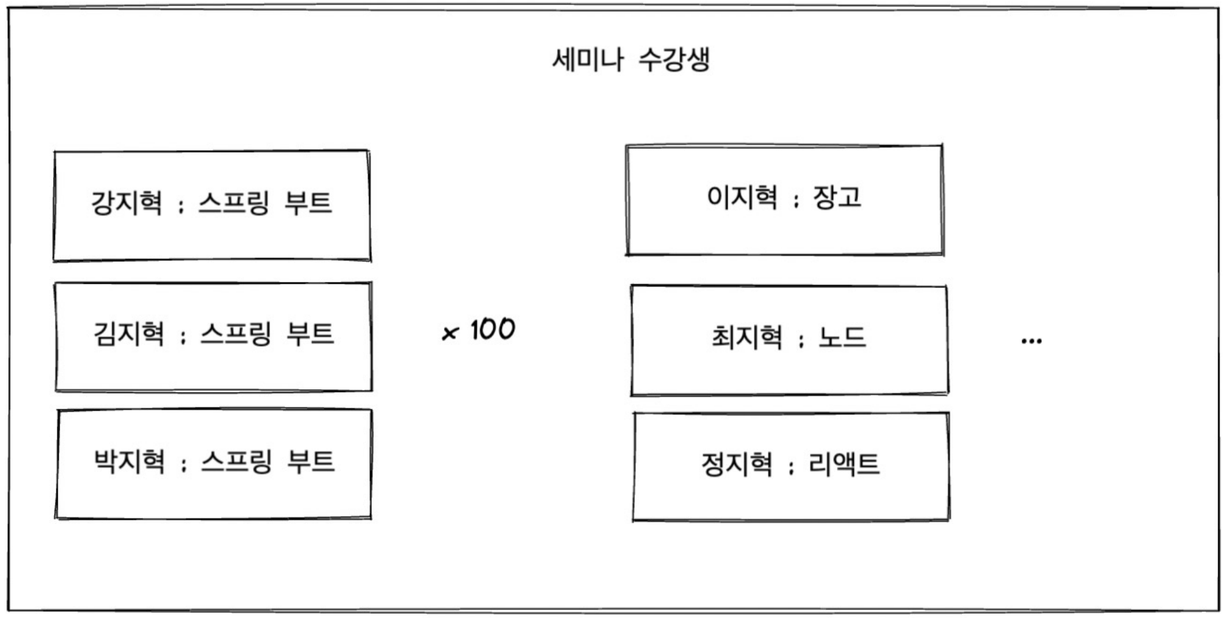
Why RDB ?

1. 정형화된 데이터

   - 엄격한 스키마 : 테이블 (Row & Column) & 데이터 타입과 제약 존재

2. 데이터 사이의 관계

   - 수강생 - 세미나 관계
   - 세미나 이름을 바꾸고 싶을 때..
   - 잘 저장하지 않았다면? 😂



WA LE
STUDIO

SQL

RDBMS 에서 사용하는 '프로그래밍 언어'

하시다보면 감이 올거에요

DDL

종류	역할
CREAT	데이터베이스, 테이블등을 생성하는 역할을 합니다.
ALTE	테이블을 수정하는 역할을 합니다.
DRO	데이터베이스, 테이블을 삭제하는 역할을 합니다.
TRUNCATE	테이블을 초기화 하는 역할을 합니다.

DML

종류	역할
SELECT	데이터를 조회하는 역할을 합니다.
INSERT	데이터를 삽입하는 역할을 합니다.
UPDATE	데이터를 수정하는 역할을 합니다.
DELETE	데이터를 삭제하는 역할을 합니다.

DCL

종류	역할
GRANT	특정 데이터베이스 사용자에게 특정 작업에 대한 수행권한 부여 합니다.
REVOKE	특정 데이터베이스 사용자에게 특정 작업에 대한 수행 권한을 박탈, 회수 합니다.
COMMIT	트랜잭션의 작업을 취소 및 원래래 복구하는 역할을 합니다.
ROLLBACK	트랜잭션의 작업을 취소 및 원래대로 복구하는 역할을 합니다.

실제 동작 확인하기 DataGrip
실제 확인하기 csv, tsv 파일

WA LE
STUDIO

RDBMS : DIY

- DDL : 운영체제, 설문결과 테이블 만들기
  - Create Table …
  - Primary Key, Column Type, …

```
data class OperatingSystem(
    val id: Long,
    val osName: String,
    val price: Long,
    val desc: String,
)
```

```
data class SurveyResponse(
    val id: Long,
    val operatingSystem: OperatingSystem,
    val springExp: Int,
    val rdbExp: Int,
    val programmingExp: Int,
    val major: String,
    val grade: String,
    val timestamp: LocalDateTime,
    val backendReason: String? = null,
    val waffleReason: String? = null,
    val somethingToSay: String? = null
)
```

- DML : 데이터 만지기
  - Insert
  - Update
  - Delete
- DataGrip GUI 통해서 데이터 조작하기

WA LE
STUDIO

DB transaction

1. 운영체제를 하나 만들고

2. 운영체제를 참고하는 SurveyResponse를 만들었어요.

만약에, 1번 작업은 실패하고, 2번 작업만 성공하면, 어떤 일이 일어날까요?

나중에 2번 데이터를 조회했는데, 운영체제 정보가 어디에도 없을거에요.

데이터베이스에 접근할 때, 이런 종류의 작업을 안전하게 처리할 장치가 필요합니다.

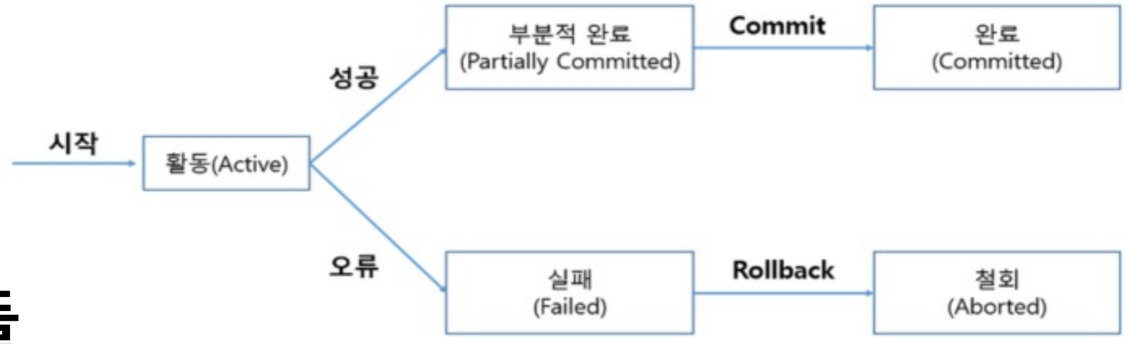
=> Transaction!

WAFFLE STUDIO

DB transaction

데이터베이스 작업의 단위

- **작업이란 : DB 상태를 변화시키는 일들**
- **안전하게 수행됨을 보장할 수 있어야 함 !**



Atomicity : 원자성. 트랜잭션은 작업의 최소 단위로서, 일부만 반영될 수 없음

Consistency: 일관성. 같은 상태에서 시작된 동일한 트랜잭션은 언제나 같은 결과를 도출

Isolation : 독립성. 처리 중인 트랜잭션끼리는 서로 간섭할 수 없다

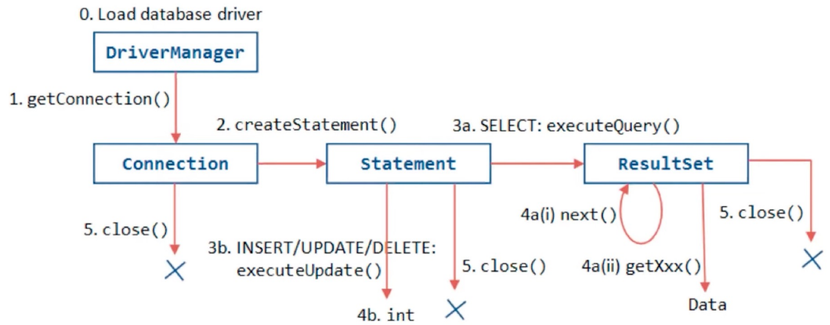
Durability : 영구성. 성공한 트랜잭션의 결과는 영구적으로 반영되어야 한다

WA LE
STUDIO

Spring Boot Data

스프링에서 어떻게 DB에 접근할 수 있을까요?

1. JDBC 이용하기



2. Persistence Framework 이용하기 ⭐

=> 또 다시, 번거로운 작업은 건너뛰고, 비즈니스 요구사항에 집중하기

[우아한 형제들 테코톡 : JDBC, SQL Mapper, ORM]

- 영상이 너무 좋아요! 꼭 한번 보시면 좋을 것 같아요

WA#LE STUDIO

Persistence Framework

~~**1. SQL Mapper :** JdbcTemplate, MyBatis, …~~

- SQL 실행 결과를 불러와서, 적절히 객체에 Mapping
- 중복을 많이 제거함
- 여전히 코드에 SQL 존재 (MyBatis의 경우 별도 .xml 파일을 관리해야 함)
- 또한 RDB <=> OOP 사이의 패러다임 불일치
  - RDB는 데이터 중심, OOP는 객체 중심의 패러다임

2. ORM : Object – Relational Mapping

- Entity 선언을 통해, 별도의 매핑 없이 빠르게 사용하고, 객체 간 관계 설정이 용이함
- JPA 인터페이스와 그 구현체 Hibernate가 메이저
- 트랜잭션, dirty checking, lazy fetching 등의 기능 제공
- Spring-data-jpa 라이브러리를 통해 쉽게 이용 가능!

WA#LE STUDIO

JPA & Hibernate

테이블 설계하고, UseCase 익혀보기

1. 엔티티 선언 및 저장은 어떻게 할 수 있을까?
2. 삭제, 조회는 어떻게 할 수 있을까?
3. 수정은 어떻게 할 수 있을까?
    - Hibernate Dirty Checking

```
@Entity
class LectureEntity(
    val title: String,
    @ManyToOne(fetch = FetchType.LAZY)
    val instructor: UserEntity,
) : BaseTimeEntity()
```

Deep Dive

1. 트랜잭션 관리는 어떻게 할 수 있을까요?
2. 동등성 비교는 어떤 식으로 이루어질까요?
3. 이미 한 번 조회한 데이터는 가지고 있을 수도 있을까요?

https://ultrakain.gitbooks.io/jpa/content/chapter3/chapter3.4.html

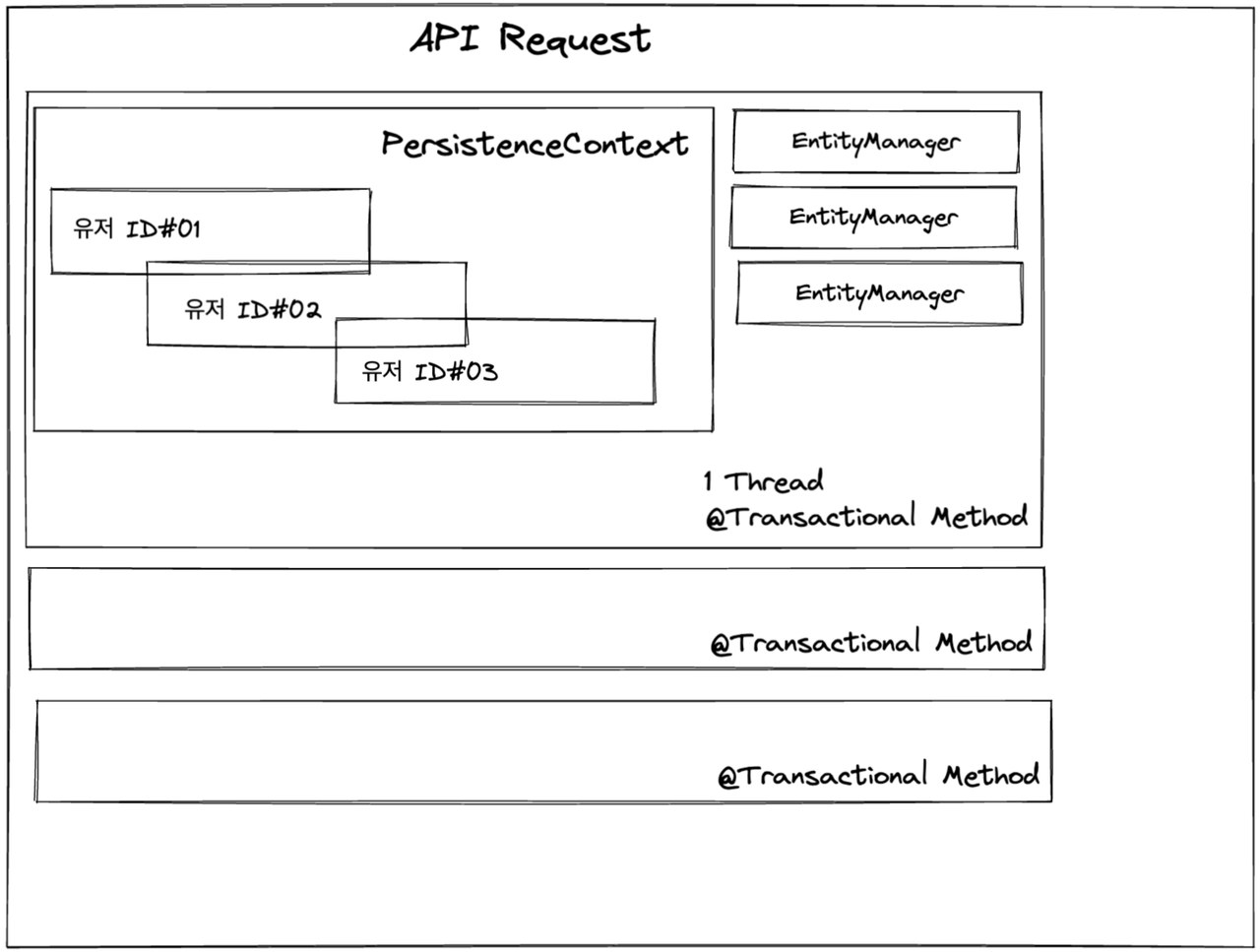
https://data-make.tistory.com/629

WA#LE STUDIO

영속성 컨텍스트

API 요청이 와서, DB 접근할 일이 생기면,,

- JPA 프레임워크에서 EntityManager 생성
- 같은 트랜잭션끼리는 영속성 컨텍스트 공유
- 영속성 컨텍스트에서는 ..
  - 1차 캐시를 통해 데이터 조회 성능 향상
  - 변경 감지
    - Update 메소드는 따로 없음
  - 쓰기 지연
    - 성능 최적화



WA LE
STUDIO

QueryDSL

DSL : Domain Specific Language

복잡한 쿼리를 SQL이 아닌 DSL로 세련되게 표현

이를 통해 컴파일 에러 등 프로그래밍 언어의 혜택을 사용할 수 있음

- Inner join, left join 등 복잡한 쿼리 컨트롤
- Projection을 통해 쉽게 데이터 매핑 가능

WA LE
STUDIO

How to Study Development?
당부의 말씀

세미나의 내용이나 과제에만 의존하지 않고, 뭐라도 공부하고 프로그래밍 하기

에러를 안내는 것이 잘하는 게 아니라, 잘 파악하고 고치는 것이 잘하는 것

- (Nomad Coder) 자꾸만 에러가 나는데 왜 그런 걸까요?

질문하기 전에 구글링, 구글링 전에 생각하기, 로그 읽기

WA LE
STUDIO

Q&A

WAFFLE STUDIO

WAFFLE STUDIO

다음 시간 키워드

AOP, DI, PSA

N+1, Transaction 관리

Testing

WA#LE
STUDIO