SPRING BOOT JPA

JPA는 표준 ORM이다

Object-Relational Mapping

이러한 객체를 저장하고 객체끼리 관계도 맺을 수 있다.

그래서 객체 지향 언어인 JAVA와 잘 맞음

굳이 query를 몰라도 쓸 수 있다는 장점!

알아야하는 것은 db 테이블 명이 되는 @Entity, query가 되는 JpaRepository 두가지!

[예제] 게시판 댓글 만들기

1. db 설계

팁! column에 뭘 넣어야할지 모르겠으면 육하 원칙을 생각하자!

	PK	FK	누가	언제	어디서	무엇을		어떻게	왜
게시판	UID		글쓴사람	글쓴 날짜	IP 주소	제목	내용	모바일/웹	NA
Board	uid		user	createdDate	ip	title	contents	1/2	
댓글	UID	게시판_UID	글쓴사람	글쓴 날짜	IP 주소		내용	모바일/웹	NA
Reply	uid	boardFk	user	createdDate	ip		contents	1/2	

2. Property 설정(application.properties 혹은 application.yml)

- log 남기는 설정!

```
spring.datasource.url=jdbc:h2:mem:testdb

spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=trace

```

3. Entity 작성

- 꿀팁 : pk, auto increment, INT UNSIGNED
- relation(다중성)

  @OneToOne,@OneToMany,@ManyToOne,@ManyToMany
- 방향성
  - 양방향은 지양하자

    만약, **@one to Many&@Many to One**양방향인 경우, fk의 주인을 설정하자
    (데이터를 삭제하면 어떤 테이블을 기준으로 관련 데이터를 다 삭제할 것인가?)

    1:다 중 “다” 가 주인임

JpaRepository

crud	method 이름
read	find~
delete	delete~
create	save
update	객체 조회 후 값 변경 & save

GET

- 받은 entity 그대로 보내면 X, 새로운 객체를 만들어 요청한 내용만 return할 것

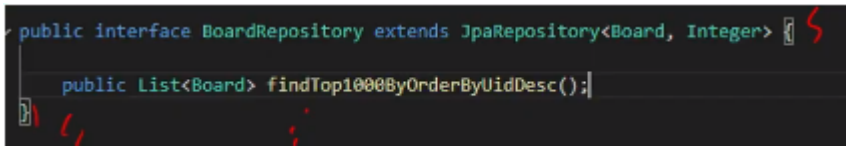
CREATE

- 객체를 만들 때 **@Builder**를 이용하면 좋다
  매개 변수의 타입을 지정함!

UPDATE

- 삭제 후 생성

사용자 정의 method



method 명이 query가 됨!

findByTitle 처럼~

이것을 위 interface 안에 선언하며됨

스프링부트 JPA docs 참고할 것