

# Spring Boot CRUD

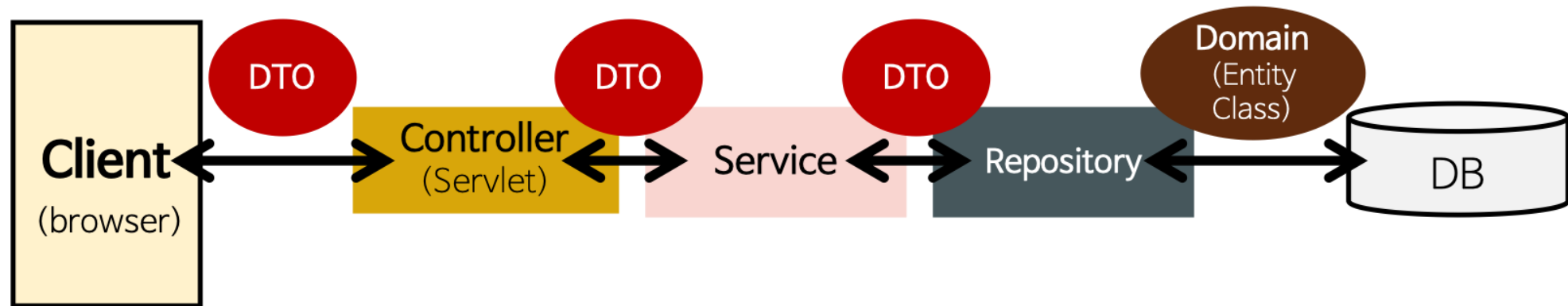
---

Jae Hyeon Kim

# — Contents

1. Database & CRUD
2. Spring Boot Database
3. Spring Boot CRUD

# Review : Spring Boot Architecture



Client **<-dto->** controller(web) - service - repository(dao) **<-domain(entity)->** DB

# Database & CRUD

# DBMS

- DBMS(Database Management System)은 데이터베이스라는 데이터의 집합을 만들고, 저장 및 관리할 수 있는 기능들을 제공하는 응용 프로그램
- 데이터의 관리에 특화된 프로그램



# SQL

- SQL(Structured Query Language)는 데이터베이스 시스템에서 자료를 처리하는 용도로 사용되는 프로그래밍 언어

```
INSERT INTO Reservation(ID, Name, ReserveDate, RoomNum)  
VALUES(5, '이순신', '2016-02-16', 1108);
```

```
SELECT *  
FROM Reservation;
```

ID	Name	ReserveDate	RoomNum
1	홍길동	2016-01-05	2014
2	임꺽정	2016-02-12	918
3	장길산	2016-01-16	1208
4	홍길동	2016-03-17	504
5	이순신	2016-02-16	1108

# — CRUD

- CRUD는 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 Create(생성), Read(읽기), Update(갱신), Delete(삭제)를 묶어서 일컫는 말
- 사용자 인터페이스가 갖추어야 할 기능(정보의 참조/검색/갱신)을 가리키는 용어

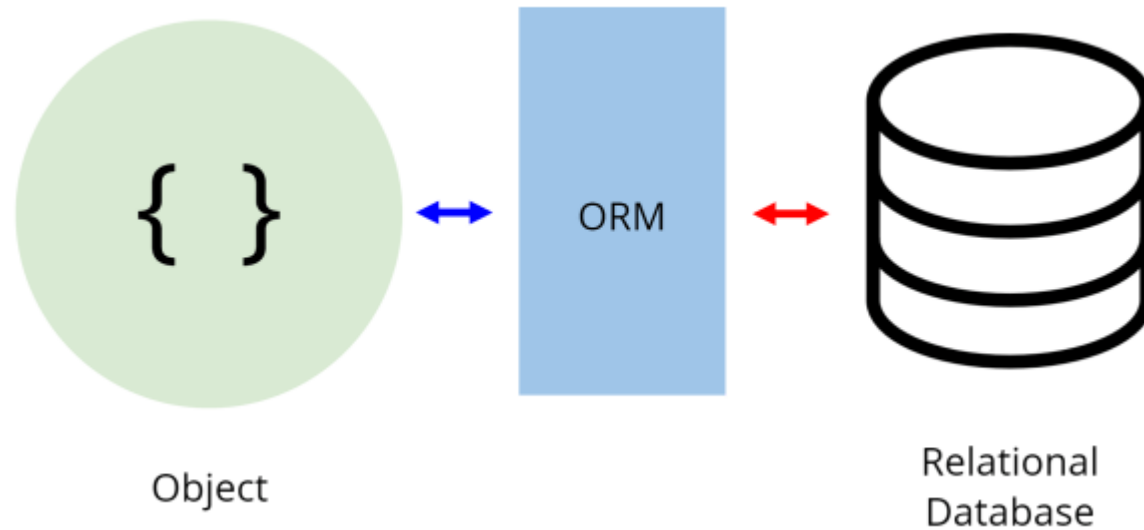
이름	조작	SQL
Create	생성	INSERT
Read(또는 Retrieve)	읽기(또는 인출)	SELECT
Update	갱신	UPDATE
Delete(또는 Destroy)	삭제(또는 파괴)	DELETE

# **Spring Boot Database**



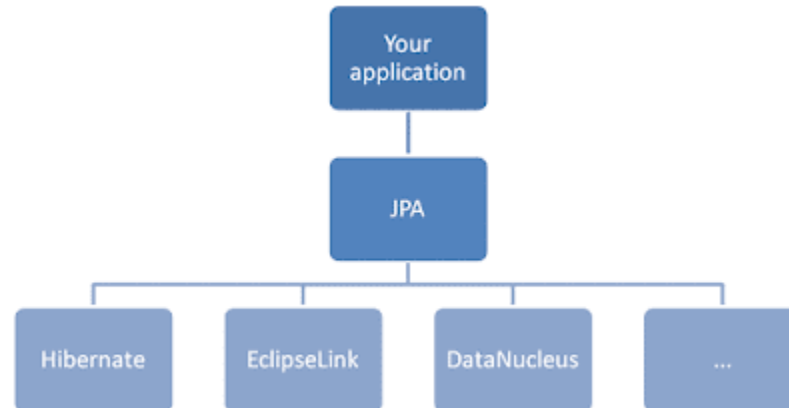
# ORM

- ORM(Object-Relational Mapping)은 우리가 일반적으로 알고 있는 어플리케이션의 객체와 RDB의 테이블을 매핑(연결)해주는 기술

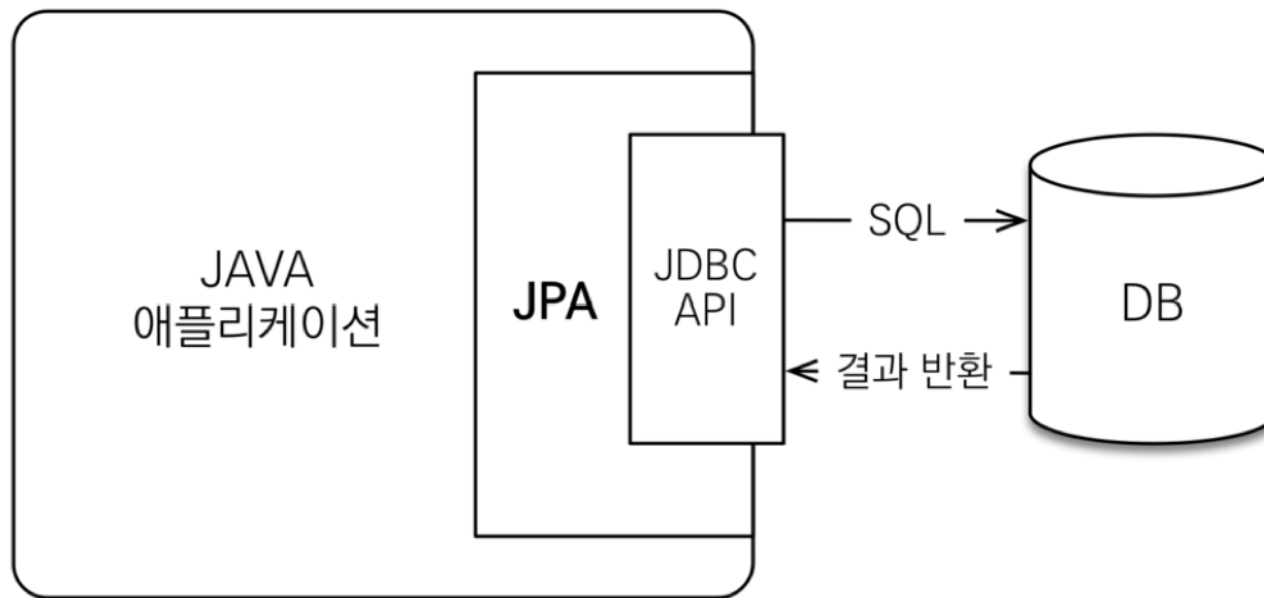


# JPA

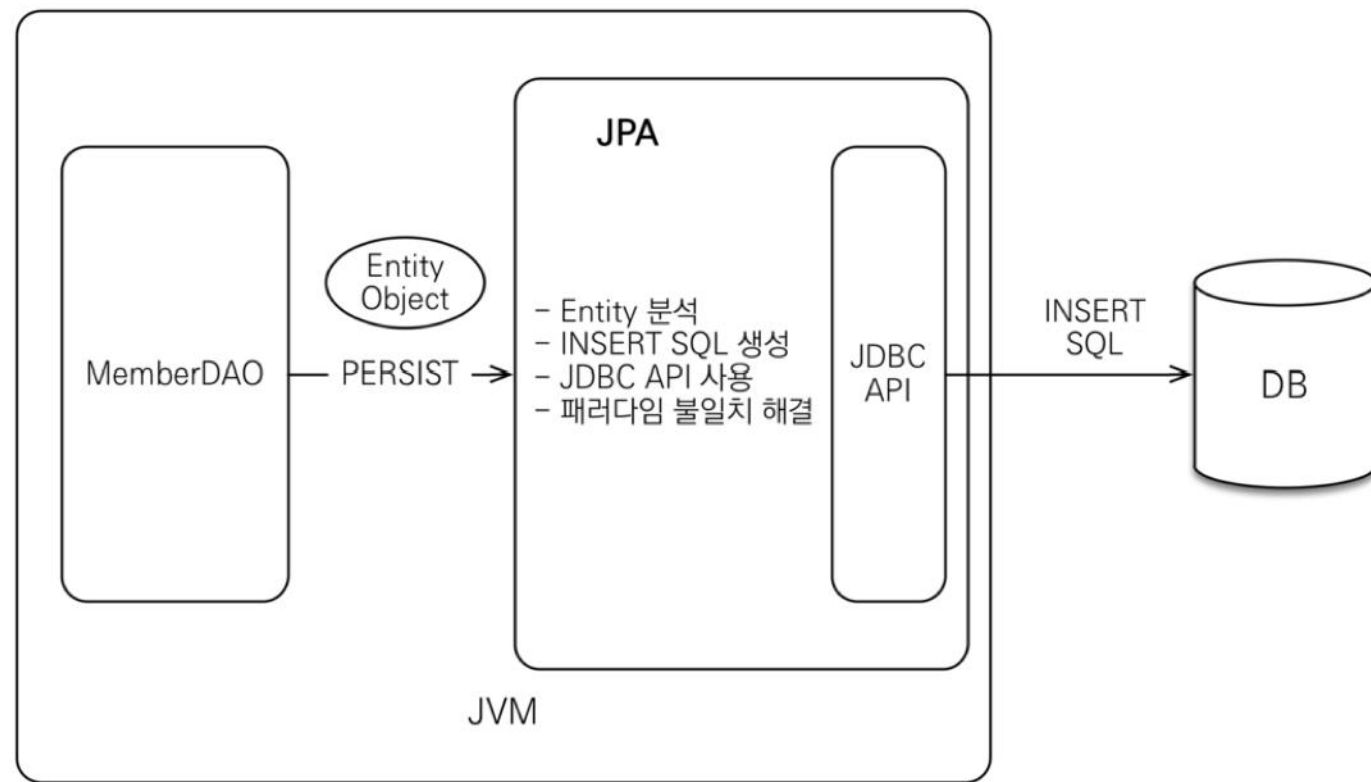
- JPA(Java Persistence API)는 Java 진영에서 ORM 기술 표준으로 사용하는 인터페이스 모음
- Java 어플리케이션에서 관계형 데이터베이스를 사용하는 방식을 정의한 인터페이스
- 인터페이스이기 때문에 Hibernate, OpenJPA등이 JPA를 구현함



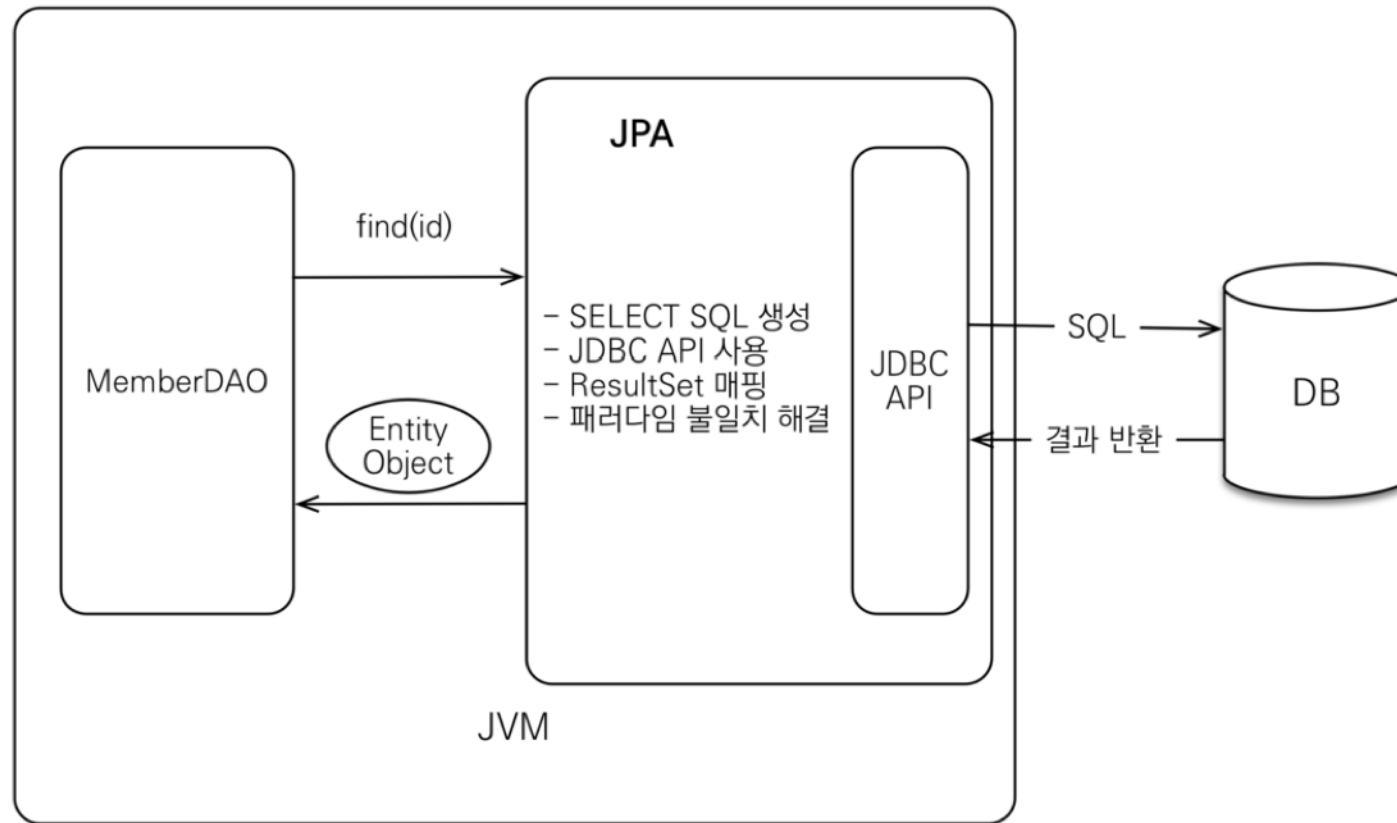
## — JPA : Process



# JPA : Save

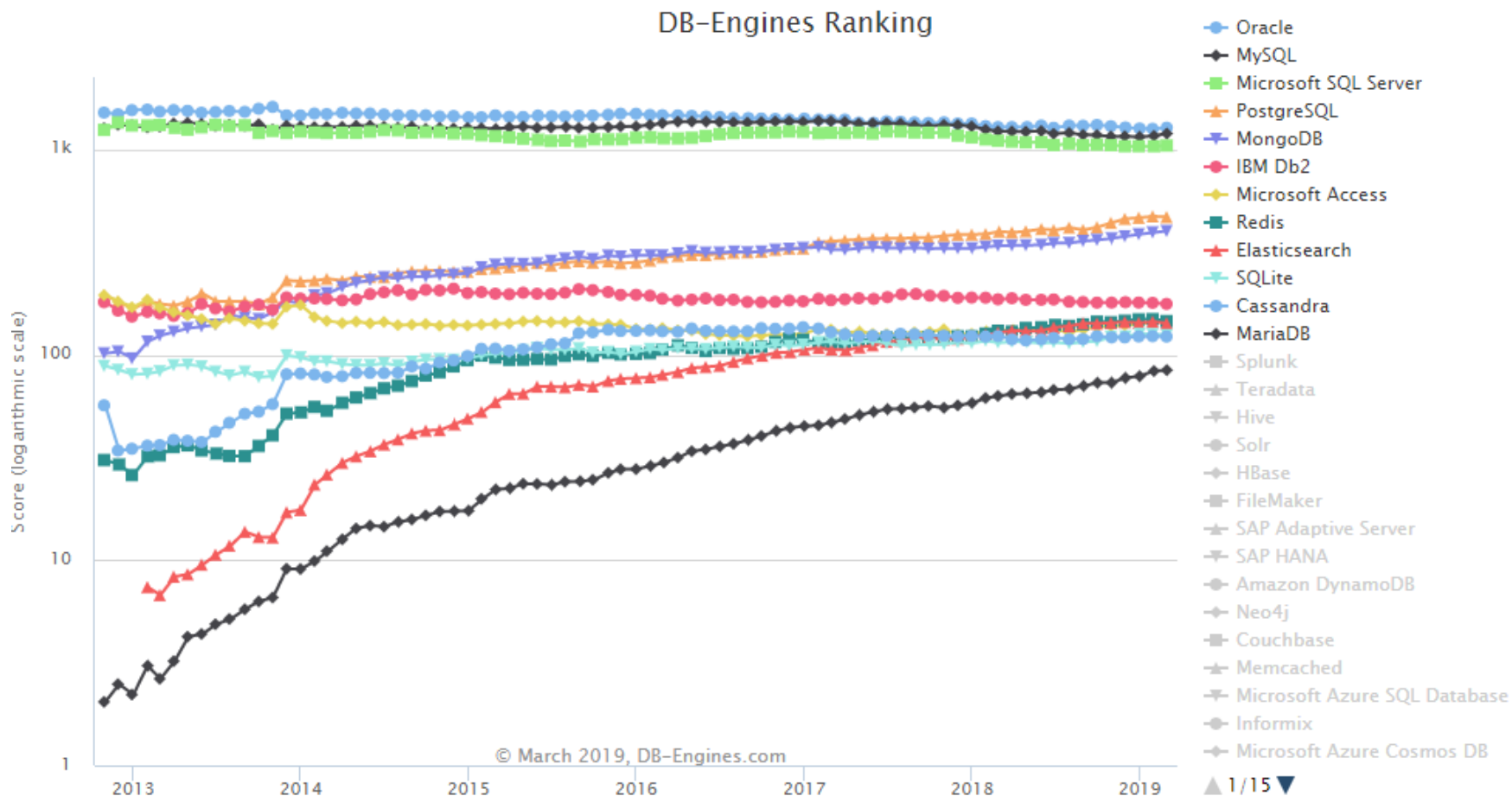


# — JPA : Select



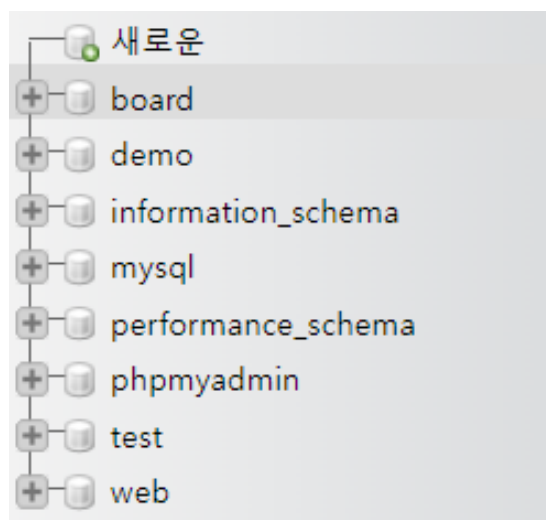
# RDBMS : MySQL

	
개발자	오라클
발표일	1995년 5월 23일
라이선스	GPL v2 (커뮤니티 에디션) 상용 라이선스 (표준, 엔터프라이즈, 클러스터)
링크	<a href="#">홈</a> <a href="#">문서</a>



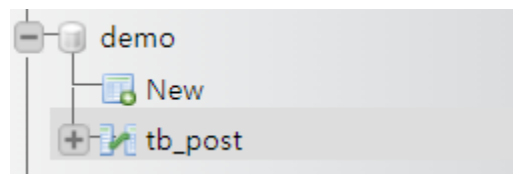
# Create Database

```
create database demo;
```



```
CREATE TABLE `tb_post` (  
  `id`          bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'PK',  
  `title`       varchar(100) NOT NULL COMMENT '제목',  
  `content`     varchar(3000) NOT NULL COMMENT '내용',  
  `writer`      varchar(20) NOT NULL COMMENT '작성자',  
  `view_cnt`    int(11) NOT NULL COMMENT '조회 수',  
  `notice_yn`   tinyint(1) NOT NULL COMMENT '공지글 여부',  
  `delete_yn`   tinyint(1) NOT NULL COMMENT '삭제 여부',  
  `created_date` datetime NOT NULL DEFAULT current_timestamp() COMMENT '생성일시',  
  `modified_date` datetime DEFAULT NULL COMMENT '최종 수정일시',  
  PRIMARY KEY (`id`)  
) COMMENT '게시글';
```

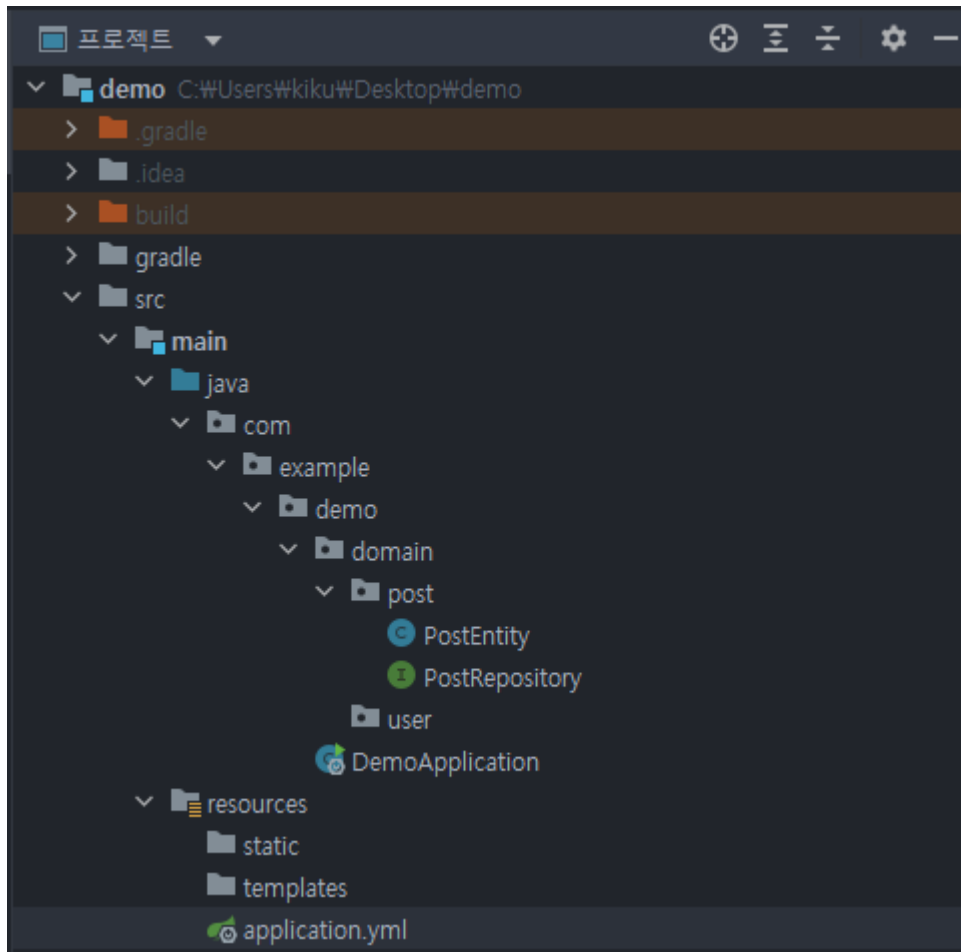
# Create Database



	#	이름	종류	데이터정렬방식	보기	Null	기본값	설명	추가
<input type="checkbox"/>	1	<u>id</u>	bigint(20)			아니오	없음	PK	AUTO_INCREMENT
<input type="checkbox"/>	2	<u>title</u>	varchar(100)	utf8mb4_general_ci		아니오	없음	제목	
<input type="checkbox"/>	3	<u>content</u>	varchar(3000)	utf8mb4_general_ci		아니오	없음	내용	
<input type="checkbox"/>	4	<u>writer</u>	varchar(20)	utf8mb4_general_ci		아니오	없음	작성자	
<input type="checkbox"/>	5	<u>view_cnt</u>	int(11)			아니오	없음	조회 수	
<input type="checkbox"/>	6	<u>notice_yn</u>	tinyint(1)			아니오	없음	공지글 여부	
<input type="checkbox"/>	7	<u>delete_yn</u>	tinyint(1)			아니오	없음	삭제 여부	
<input type="checkbox"/>	8	<u>created_date</u>	datetime			아니오	current_timestamp()	생성일시	
<input type="checkbox"/>	9	<u>modified_date</u>	datetime			예	NULL	최종 수정일시	



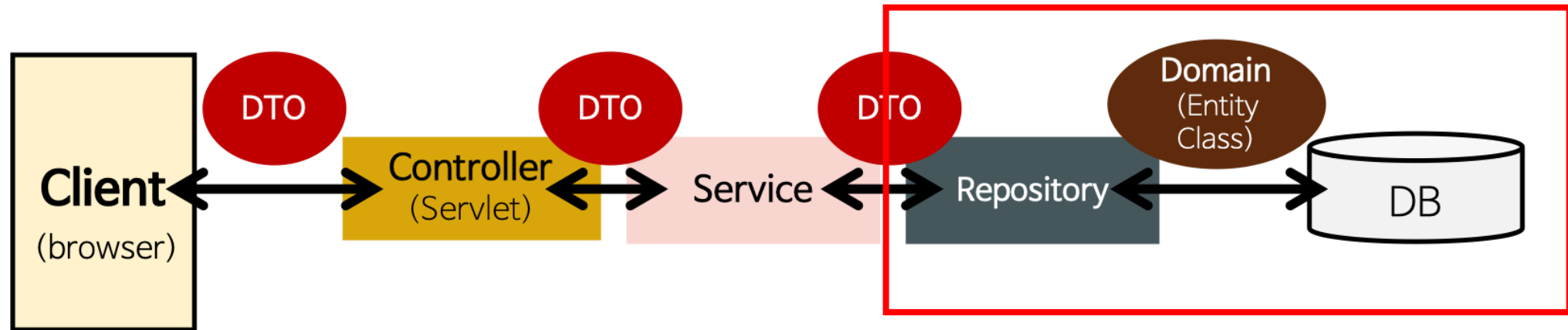
# Spring Boot Database Connection



```
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/demo
    username: root
    password:
  jpa:
    show-sql: true
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        format_sql: true
```

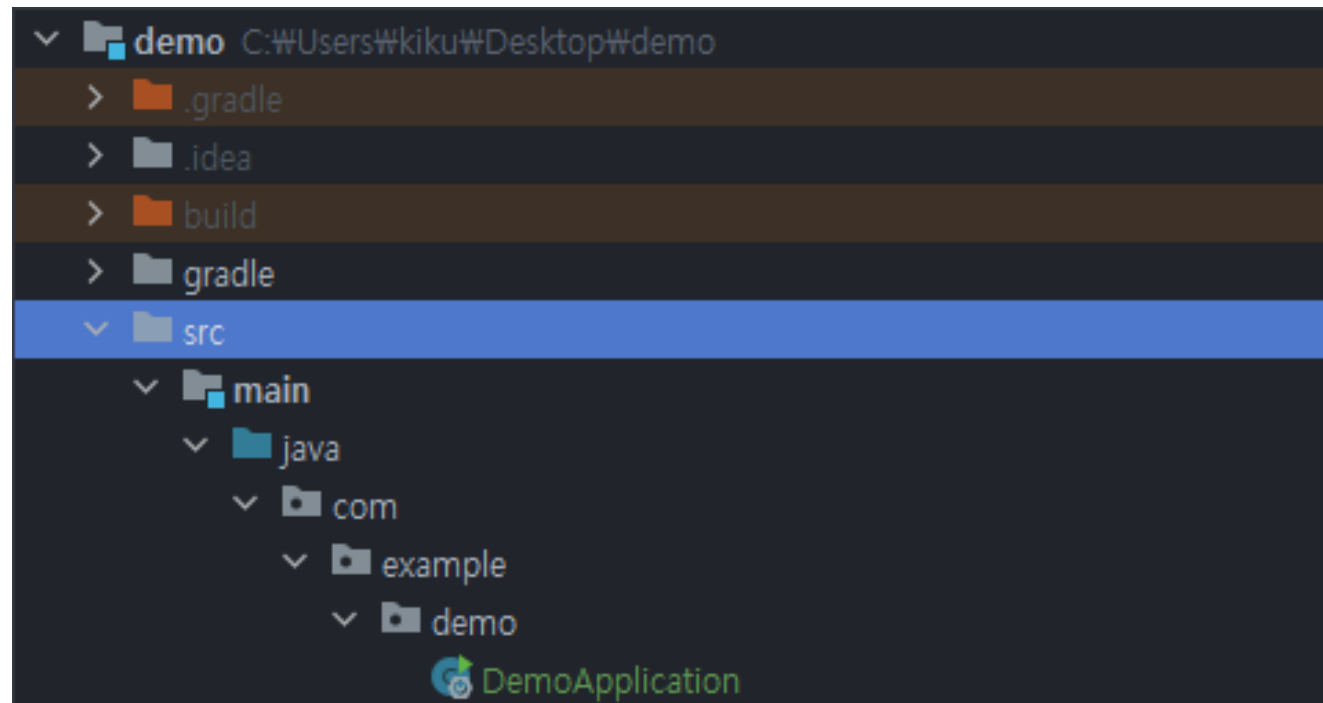
# **Spring Boot CRUD**

# Spring Boot Architecture

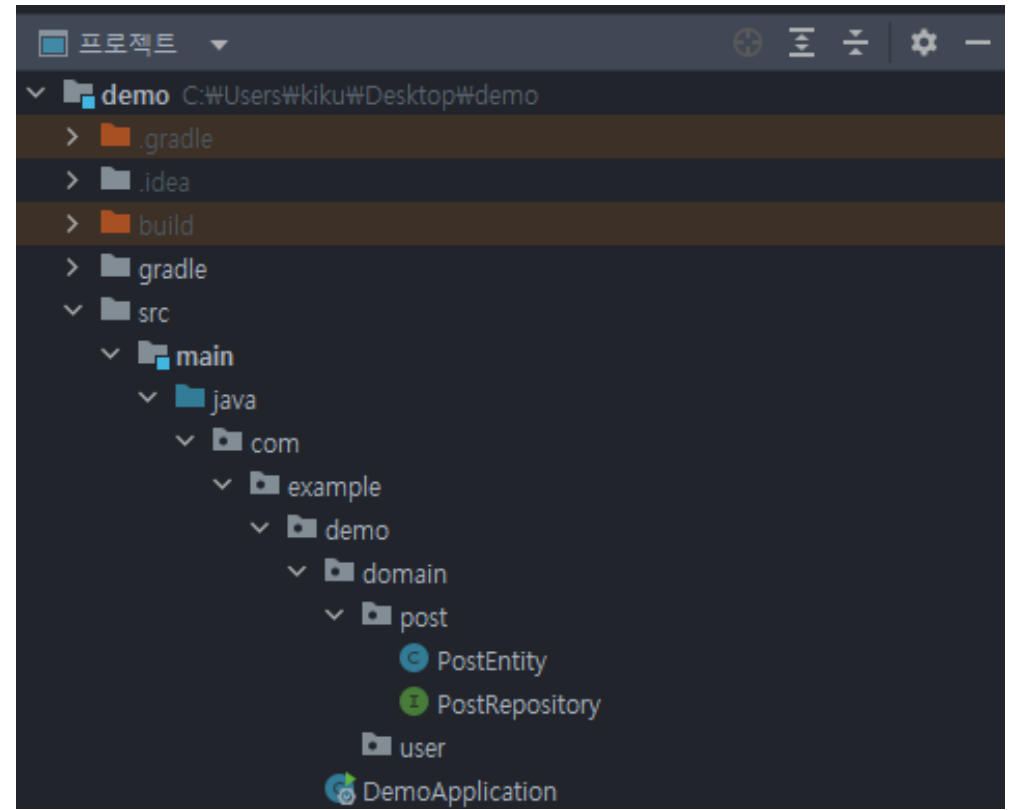
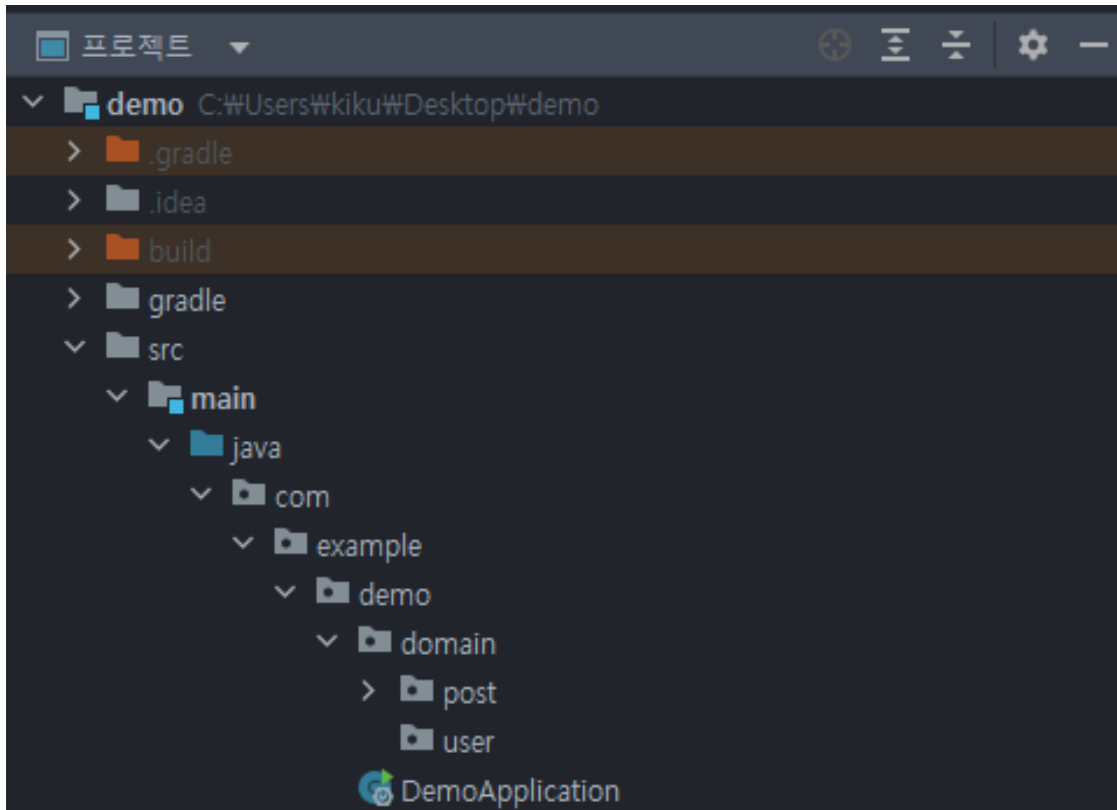


Client **<-dto->** controller(web) – service – repository(dao) **<-domain(entity)->** DB

# — Spring Boot CRUD



# Spring Boot CRUD



# Entity

```
import lombok.*;
import javax.persistence.*;
import java.util.Date;
```

7개 사용 위치 kiku99

```
@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
@Table(name = "tb_post")
@Entity
public class PostEntity {
```

```
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int id;

@Column
private String title;

@Column
private String content;

@Column
private String writer;

@Column
private int view_cnt;

@Column
private int notice_yn;


@Column
private int delete_yn;

@Column
private Date created_date;

@Column
private Date modified_date;
}
```

## — Repository(DAO)

```
import org.springframework.data.jpa.repository.JpaRepository;
```

1개 사용 위치  kiku99

```
public interface PostRepository extends JpaRepository<PostEntity, Integer> {  
}
```

# Repository(DAO)

```
@NoRepositoryBean
public interface JpaRepository<T, ID> extends PagingAndSortingRepository<T, ID>, QueryByExampleExecutor<T> {

    /*
     * (non-Javadoc)
     * @see org.springframework.data.repository.CrudRepository#findAll()
     */
    /*
     1개 구현
     */
    @Override
    List<T> findAll();

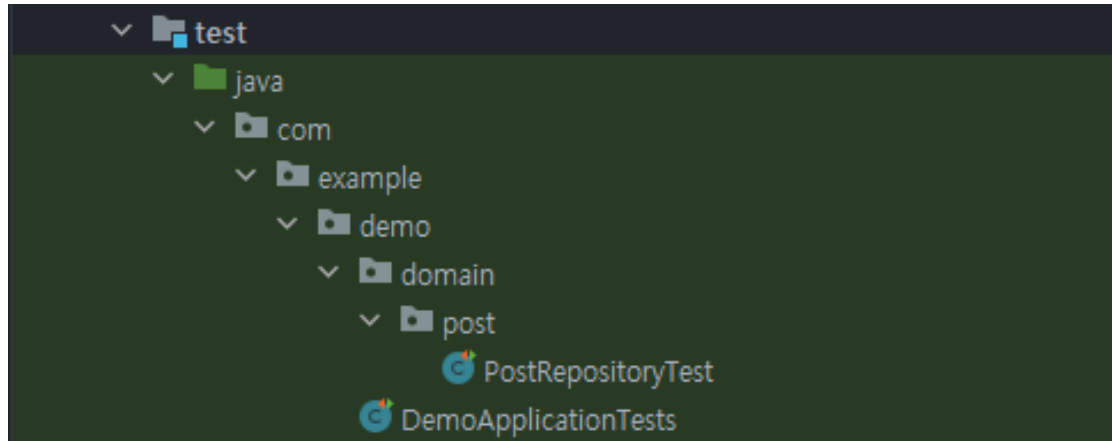
    /*
     * (non-Javadoc)
     * @see org.springframework.data.repository.PagingAndSortingRepository#findAll(org.springframework.data.domain.Sort)
     */
    /*
     1개 구현
     */
    @Override
    List<T> findAll(Sort sort);

    /*
     * (non-Javadoc)
     * @see org.springframework.data.repository.CrudRepository#findAll(java.lang.Iterable)
     */
    /*
     1개 사용 위치 1개 구현
     */
    @Override
    List<T> findAllById(Iterable<ID> ids);

    /*
```



# — CRUD Test



kiku99

```
@SpringBootTest
```

```
public class PostRepositoryTest {
```

4개 사용 위치

```
@Autowired
```

```
PostRepository postRepository;
```

## — CRUD Test : Create

```
@Test
public void InsertDummies(){
    PostEntity postEntity = PostEntity.builder()
        .id(1)
        .title("insert test")
        .content("insert test content")
        .writer("kiku")
        .view_cnt(0)
        .notice_yn(0)
        .created_date(new Timestamp(System.currentTimeMillis()))
        .build();

    postRepository.save(postEntity);
}
```

# — CRUD Test : Create

WHERE			ORDER BY						
	id	title	content	writer	view_cnt	notice_yn	delete_yn	created_date	modified_date
1	2	insert test	insert test content	kiku	0	0	0	2022-11-25 15:45:43	<null>

## — CRUD Test : Read

```
@Test
public void SelectDummies(){
    int id = 1;

    Optional<PostEntity> result = postRepository.findById(id);

    System.out.println("=====");

    if (result.isPresent()){
        PostEntity postEntity = result.get();
        System.out.println(postEntity);
    }
}
```

# CRUD Test : Read

```
select
  postentity0_.id as id1_0_0_,
  postentity0_.content as content2_0_0_,
  postentity0_.created_date as created_3_0_0_,
  postentity0_.delete_yn as delete_y4_0_0_,
  postentity0_.modified_date as modified5_0_0_,
  postentity0_.notice_yn as notice_y6_0_0_,
  postentity0_.title as title7_0_0_,
  postentity0_.view_cnt as view_cnt8_0_0_,
  postentity0_.writer as writer9_0_0_
from
  tb_post postentity0_
where
  postentity0_.id=?
=====
PostEntity(id=1, title=insert test, content=insert test content, writer=kiku, view_cnt=0, notice_yn=0, delete_yn=0, created_date=2022-11-25 13:42:26.0, modified_date=null)
2022-11-25 13:59:18.637 INFO 23056 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2022-11-25 13:59:18.639 INFO 23056 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2022-11-25 13:59:18.646 INFO 23056 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
BUILD SUCCESSFUL in 6s
4 actionable tasks: 2 executed, 2 up-to-date
오후 1:59:18: 실행이 완료되었습니다 ':test --tests "com.example.demo.domain.post.PostRepositoryTest.SelectDummies"'.

```

## — CRUD Test : Update

```
@Test
public void UpdateDummies(){
    PostEntity postEntity = PostEntity.builder()
        .id(1)
        .title("insert test")
        .content("Update content")
        .writer("kiku")
        .view_cnt(0)
        .notice_yn(0)
        .created_date(new Timestamp(System.currentTimeMillis()))
        .modified_date(new Timestamp(System.currentTimeMillis()))
        .build();

    postRepository.save(postEntity);
}
```

# — CRUD Test : Update

WHERE				ORDER BY					
	id	title	content	writer	view_cnt	notice_yn	delete_yn	created_date	modified_date
1	1	insert test	Update content	kiku	0	0	0	2022-11-25 14:05:57	2022-11-25 14:05:57

## — CRUD Test : Delete

```
@Test
public void DeleteDummies(){
    int id = 1;

    postRepository.deleteById(id);
}
```



# CRUD Test : Delete

```
Hibernate:
  delete
  from
    tb_post
  where
    id=?
2022-11-25 14:07:39.586 INFO 23744 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2022-11-25 14:07:39.588 INFO 23744 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2022-11-25 14:07:39.597 INFO 23744 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
BUILD SUCCESSFUL in 6s
4 actionable tasks: 2 executed, 2 up-to-date
오후 2:07:39: 실행이 완료되었습니다 ':test --tests "com.example.demo.domain.post.PostRepositoryTest.DeleteDummies"'.
```

WHERE

ORDER BY

id title content writer view\_cnt notice\_yn delete\_yn created\_date modified\_date