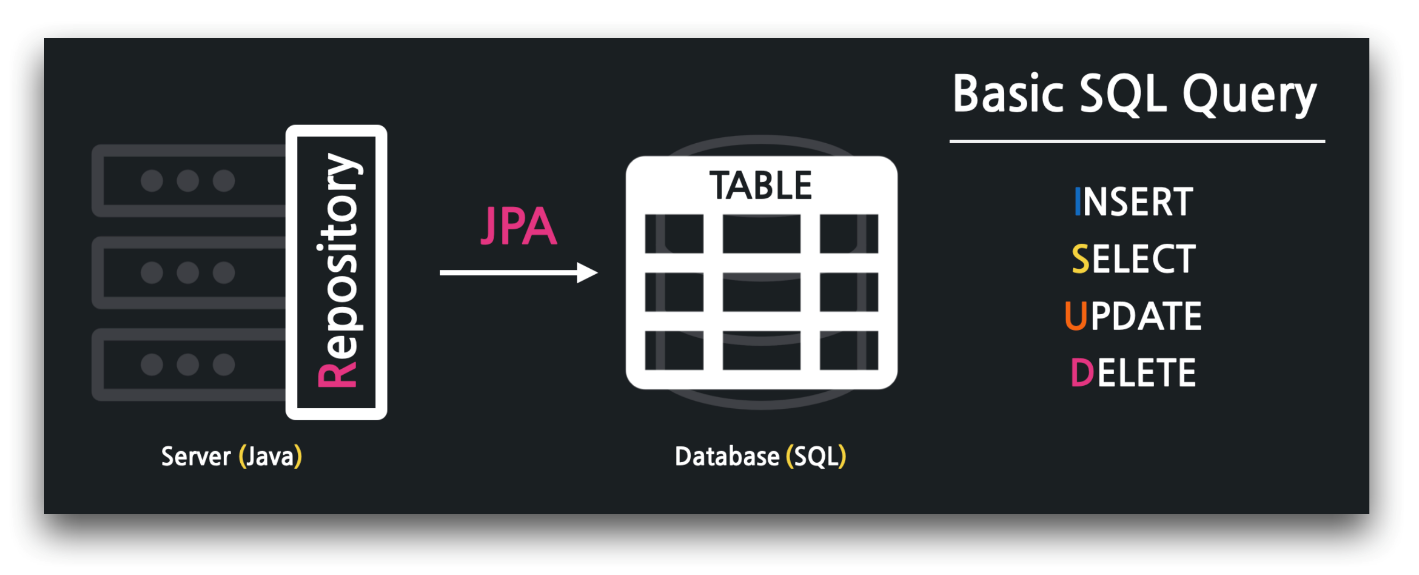
17 데이터 CRUD와 SQL 쿼리

**데이터 CRUD와 SQL 쿼리**

**미션**

데이터 CRUD 과정 중 발생하는 SQL 쿼리를 분석하시오.  


**01:36 JPA 로깅 설정 - application.properties**

../resources/application.properties

1. # 09강: h2 DB, 웹 콘솔 설정
2. spring.h2.console.enabled=true
3. # 15강: data.sql 적용을 위한 설정(스프링부트 2.5 이상)
4. spring.jpa.defer-datasource-initialization=true
5. # 17강: JPA 로깅 설정
6. ## 디버그 레벨로 쿼리 출력
7. logging.level.org.hibernate.SQL=DEBUG
8. ## 이쁘게 보여주기
9. spring.jpa.properties.hibernate.format\_sql=true
10. ## 파라미터 보여주기
11. logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
12. ## 고정 url 설정
13. spring.datasource.url=jdbc:h2:mem:testdb

**05:49 ID 자동 생성 전략**

../entity/Article

1. package com.example.firstproject.entity;
2. import lombok.AllArgsConstructor;
3. import lombok.Getter;
4. import lombok.NoArgsConstructor;
5. import lombok.ToString;
6. import javax.persistence.\*;
7. @Entity
8. @AllArgsConstructor
9. @NoArgsConstructor
10. @ToString
11. @Getter
12. public class Article {
13. @Id
14. @GeneratedValue(strategy = GenerationType.IDENTITY) // 자동 생성 전략
15. private Long id;
16. @Column
17. private String title;
18. @Column
19. private String content;
20. }

**10:35 기본 SQL 쿼리 연습**

커피 테이블 생성

1. CREATE TABLE coffee (
2. id BIGINT GENERATED BY DEFAULT AS IDENTITY,
3. name VARCHAR(255),
4. price INTEGER,
5. PRIMARY KEY (id)
6. );

커피 레코드 추가

1. -- 단건 추가
2. INSERT INTO
3. coffee(id, name, price)
4. VALUES
5. (1, '아메리카노', 4100)
6. ;
7. -- 여러건 추가
8. INSERT INTO
9. coffee(id, name, price)
10. VALUES
11. (2, '라떼', 4600),
12. (3, '모카', 5100),
13. (4, '오늘의 커피', 3800)
14. ;

커피 레코드 조회

1. SELECT
2. id,
3. name,
4. price
5. FROM
6. coffee
7. ;

커피 레코드 수정

1. UPDATE
2. coffee
3. SET
4. price = 9900
5. WHERE
6. id = 4
7. ;

커피 레코드 삭제

1. DELETE FROM
2. coffee
3. WHERE
4. id = 4
5. ;

**🔥 구글링 훈련하기**

* JPA 쿼리 로그
* JPA GeneratedValue IDENTITY
* SQL CREATE TABLE
* SQL UPDATE
* SQL DELETE
* SQL WHERE 사용법