

(1)

1-1. 엔티티 매니저 팩토리는 jpa 동작시키기 위해 기반 객체를 만들고 jpa 구현체에 따라서는 DB 커넥션 풀까지도 생성하므로, 엔티티매니저팩토리 생성 비용이 매우 크기 때문이다.

1-2. 엔티티 매니저는 기본적으로 **엔티티를 DB에 등록,수정,삭제,조회(CRUD)**할 수 있도록 한다.

예를 들어 em.persist(), em.find(), em.remove(), member.setAge() 가 있다. – 수정 메서드는 따로 없다. 엔티티를 변경해주면 jpa가 변경사항을 추적하기 때문에 자동으로 update 쿼리가 날라가 db를 수정한다.

2-1. dialect는 방언이라는 뜻으로, db마다 다른 sql 문법과 함수를 이야기한다. jpa는 특정 db에 종속되어있지 않으므로, 다양한 db를 사용할 수 있는데 이때 각 db마다 다른 문법과 함수를 사용하게 된다. 따라서 jpa의 구현체들은 다양한 데이터베이스 방언을 제공한다. 우리는 그저 jpa를 사용하기만 하면 데이터베이스 방언이 해당 db에 맞추어 sql을 처리해주는 것이다.

즉, 데이터베이스가 바뀌더라도 어플리케이션 코드를 변경하지 않고, 데이터베이스 방언만 교체해주면 된다는 장점이 있다.

3-1.

JPQL

엔티티 객체를 대상으로 쿼리한다. 그렇기 때문에 JPQL은 데이터베이스 테이블을 전혀 알지 못한다는 점을 기억하자.

더 조사하기 – JPQL에 우리가 어떤 명령을 어떻게 주는지 정확히 모르겠다

SQL

데이터베이스 테이블을 대상으로 쿼리한다.

3-2

우리가 코드를 작성하는 것은 엔티티 객체에 대한 것이다.

그렇기 때문에, db의 데이터를 조회하려면, db의 데이터를 엔티티 객체로 변경하고 검색해야한다.

이건 사실상 불가능하므로, 엔티티 객체에 대한 쿼리문을 작성하면 jpa가 JPQL에 대한 SQL을 만들어 DB에서 데이터를 조회하는 방식을 사용하는 것이다.

4.

JPA 를 사용하기 위해선, 클래스를 테이블에(클래스->테이블) 매핑해야한다. 이때, 매핑 정보를 표시하기 위해 사용되는 것이 아래의 어노테이션이다.

각 어노테이션(전부 다는 아님)에 속성으로 (name = "name")을 줄 수 있다. 이 말은 어노테이션이 붙은 클래스의 요소를 name 이라는 이름을 가진 테이블의 요소에 매핑한다는 뜻이다.

@Entity

해당 클래스를 테이블과 매핑한다고 JPA 에 알려준다.

즉, 엔티티 클래스임을 말해준다

@Table

해당 클래스에 매핑할 **테이블 정보**를 알려준다. **name 속성 사용 가능하다.**

@Id

해당 필드를 테이블의 기본키(pk)에 매핑한다

즉, 해당 필드가 pk 임을 말해준다.

(2)

<https://github.com/king0104/JpaProgramming>

완료

(3)

String, Date

출제 빈도 낮음

평균 점수 낮음

루시와 엘라 찾기

Level 2 • 17911명 완료



이름에 ei이 들어가는 동물 찾기

Level 2 • 17392명 완료



중성화 여부 파악하기

Level 2 • 15519명 완료



오랜 기간 보호한 동물(2)

Level 3 • 14150명 완료



DATETIME에서 DATE로 형 변환

Level 2 • 14384명 완료



(4)

(4-1)

```
select title from course
where dept_name = 'Comp. Sci.' and credits = '3'
```

	title
1	Robotics
2	Image Processing
3	Database System Concepts


(4-2)

```
# 중복 제거는 느낌이 안온다.
# 세션 id가 같은 것끼리 뭉치고 싶은데...
select student.ID from instructor
inner join teaches t on instructor.ID = t.ID
inner join section s on t.course_id = s.course_id and t.sec_id = s.sec_id and t.semester = s.semester and t.year = s.year
inner join takes on s.course_id = takes.course_id and s.sec_id = takes.sec_id and s.semester = takes.semester and s.year = takes.year
inner join student on takes.ID = student.ID
where instructor.name = 'Einstein'
```

	ID
1	44553

- 중복 제거는 어떻게 하는 것인지??
- join 문을 통해,
- 논리적으로는 section 을 기준으로 테이블들을 합쳐서 같은 section 을 가리키고 있는 row 를 찾으려고 했다
- 물리적으로는 join 문을 하나의 테이블부터 끝 테이블까지 진행한 후에, 조건을 걸어주었다.

(4-3)



```
select name from instructor
order by salary desc
limit 1;
```

	name
1	Einstein