

2024.07.11

오전\

새로운 maven project 생성

groupid : com.springboot

artifactid : advanced_jpa

dependency tools : Lombok, Spring configuration Processor , Spring Web, Spring Data JPA, Mariadb driver

hadoop@worker2:~/workspace/dbtest\$ docker start mydb2

properties의 db정보 옮기기

jpql이란?

- Jpa Query Language
- JPA에서 사용할 수 있는 쿼리를 의미
- JPQL의 문법은 SQL과 매우 비슷해서 데이터베이스 쿼리에 익숙한 분들은 쉽게 적용 가능
- 차이점은 SQL에서는 테이블이나 컬럼의 이름을 사용하는 것과 달리 JPQL은 아래 그림처럼 엔티티 객체를 대상으로 수행
- 그렇기 때문에 매핑된 엔티티의 이름과 필드의 이름을 사용한다는 것

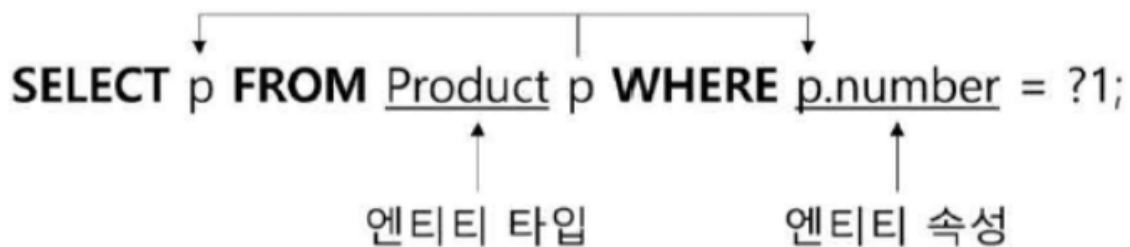


그림 8.2 JPQL 쿼리의 기본 구조

*** 쿼리 메서드

- 리포지토리는 JpaRepository를 상속받는 것만으로도 다양한 CRUD 메서드를 제공
- 이러한 기본 메서드들은 식별자 기반으로 생성되기 때문에 결국 별도의 메서드를 정의해서 사용하는 경우가 많음

- 이 때 간단한 쿼리문을 작성하기 위해 사용되는 것이 쿼리 메서드

생성

- 동작을 결정하는 주체(Subject)와 서술어(Predicate)로 구분
- find .. By, exists .. By와 같은 키워드로 쿼리의 주제를 정하며 By는 서술어의 시작을 나타내는 구분자 역할을 함
- 서술어 부분은 검색 및 정렬 조건을 지정하는 영역
- 기본적으로 엔티티의 속성으로 정의할 수 있고, AND나 OR를 사용해 조건을 확장하는 것도 가능

리포지토리의 쿼리 메서드 생성 예

```
// (리턴타입) + {주제 + 서술어(속성)} 구조의 메서드
List<Person> findByLastnameAndEmail(String lastName, String email);
```



메서드의 주제 키워드

- 쿼리 메서드의 주제 부분에 사용할 수 있는 주요 키워드는 다음과 같음
- find ... By
- read ... By
- get ... By
- query ... By
- search ... By
- stream ... By

// find ... by

Optional findByNumber(Long number);

List findAllByName(String name);

Product queryByNumber(Long number);

****## 조건자 키워드**

Is

Not

Null, NotNull

List findByUpdatedAtNull();

True, False

And, Or**

**

// exists..By (예제 8.2)

boolean existsByNumber(Long number);

// count..By (예제 8.3)

long countByName(String name);

// delete..By, remove..By (예제 8.4)

void deleteByNumber(Long number);

long removeByName(String name);

// ...First..., ...Top... (예제 8.5)

List findFirst5ByName(String name);

List findTop10ByName(String name);

/ 쿼리 메소드의 조건자 키워드 /

// Is, Equals (생략 가능) (예제 8.6)

// Logical Keyword : IS , Keyword Expressions : Is, Equals, (or no keyword)

// findByNumber 메소드와 동일하게 동작

Product findByNumberIs(Long number);

Product findByNumberEquals(Long number);

// (Is)Not (예제 8.7)

Product findByNumberIsNot(Long number);

Product findByNumberNot(Long number);

// (Is)Null, (Is)NotNull (예제 8.8)

List findByUpdatedAtNull();

List findByUpdatedAtIsNull();

List findByUpdatedAtNotNull();

List findByUpdatedAtIsNotNull();

// And, Or (예제 8.10)

Product findByNumberAndName(Long number, String name);

Product findByNumberOrName(Long number, String name);

// (Is)GreaterThan, (Is)LessThan, (Is)Between (예제 8.11)

List findByPriceIsGreaterThan(Long price);

List findByPriceGreaterThan(Long price);

List findByPriceGreaterThanEqual(Long price);

List findByPriceIsLessThan(Long price);

List findByPriceLessThan(Long price);

List findByPriceLessThanEqual(Long price);

List findByPriceIsBetween(Long lowPrice, Long highPrice);

List findByPriceBetween(Long lowPrice, Long highPrice);

// (Is)Like, (Is)Containing, (Is)StartingWith, (Is)EndingWith (예제 8.12)

List findByNameLike(String name);

List findByNamesLike(String name);

List findByNameContains(String name);

List findByNameContaining(String name);

List findByNamesContaining(String name);

List findByNameStartsWith(String name);

List findByNameStartingWith(String name);

List findByNamesStartingWith(String name);

```
List findByNameEndsWith(String name);
```

```
List findByNameEndingWith(String name);
```

```
List findByNameelsEndingWith(String name);
```

/ 정렬 처리하기 /

```
// Asc : 오름차순, Desc : 내림차순 (예제 8.13)
```

```
List findByNameOrderByNumberAsc(String name);
```

```
List findByNameOrderByNumberDesc(String name);
```

```
// 여러 정렬 기준 사용하기, And를 붙이지 않음 (예제 8.14)
```

```
List findByNameOrderByPriceAscStockDesc(String name);
```

debug console을 보자

꼭 entity에 to_String을 넣자..

오후

django -> compose ->

settings.py

```
DATABASES = {
```

```
    'default': {
```

```
        'ENGINE': 'django.db.backends.mysql',
```

```
        'NAME': 'encore_web',
```

```
        'USER': 'class5',
```

```
        'PASSWORD' : 'pw',
```

```
        'HOST' : 'ip',
```

```
        'PORT': 'port' }
```

```
}
```

config 안에 setting.py가있다..

ngins파일에도 dockerfile이있다.

아파비 nginx는 웹 - 정적

alpine은 용량이 작은 linux 이미지만들때 사용한다.

리눅스 apt

리눅스 um

dnp

alpine apk

requirt칼면 구니콘이 생기므로 wsgi는 was랑 바인디해줘..

DOKCERFILE 있는곳에서 docker-compose up

```
sudo netstat -tuln | grep :80
```

누가 80번을 쓰고있으면

```
sudo systemctl stop nginx
```

bind 0.0.0.0:8000을지워도

그냥 ip로 들어가는ㄷ데 진입은 80번 포트로 가는데 nginx는 web의 80으로 가는데

하지만 직접 ip:8000은 사라졌으므로