Querydsl

- description : Querydsl
- author : 주레피
- email : dhan@repia.com
- lastupdate : 2022-05-18

Intro

Case Study

Spring Boot QueryDSL 사용

Term

Tip

문법

```
@ManyToOne를 사용할 경우 (fetch = FetchType.LAZY)를 항상 사용할 것

Entity는 기본 생성자가 있어야 됨

toString에서는 연관관계를 피해야 됨

Querydsl 장점
오타를 미리 방지할 수 있음
쿼리를 자바 코드로 작성

Querydsl은 결국에는 JQPL 빌더 역할임
use_sql_comments: true <- jqpl 보여줌

count 조건이 간단하면 count, contents 쿼리를 각각 실행하는게 효율적임

다양한 타입이 오면 Tuple 사용 <- 실무에서는 잘 사용하지 않음

쎄타 조인은 leftJoin이 되지 않음.

페치 조인: 실무에서 많이 사용

서브 쿼리: 인라인 뷰는 지원하지 않음(from절 내에서 서브쿼리)
원래는 select에서 서브쿼리가 안되지만 하이버네이트에서 지원
```

```
 from 절의 서브쿼리 해결방안
1. 서브쿼리를 join으로 변경한다.(높은 확률로 가능할 경우가 많음)
2. 애플리케이션에서 쿼리를 2번 분리해서 실행한다.
 3. nativeSQL을 사용한다.

 SQL AntiPatterns (책)
원 쿼리가 복잡하면 조금씩 나눠서 가져오는 것도 방법이다.

프로젝션: select 대상 지정

 Tuple 타입은 querydsl에 종속적인 타입이다. 따라서 repository내에서만 사용하자!!

 @QueryProjection
 Dto도 Q File로 생성이 됨(compile 진행 필요)

동적 쿼리
 실무에서 사용은 Predicate 사용

벌크 연산, 대용량 처리, 변경 감지(Dirty Checking)
벌크 연산은 바로 DB에 처리함, 하지만 영속성 컨텍스트는 예전정보가 있음, 영속성 컨텍스트가 항상 우선권을 가짐
 incompatible read를 방지하기 위해서
   em.flush();
   em.clear(); 실행

테스트 코드에서는 기본적으로 Transactional이 끝나고 롤백함

 sql function 사용을 하려면 방언(dialect에 등록이 되어야 함, 아니면 구현해야 함)
```

실무 활용

```
동시성 문제는 전혀 없음. 트랜잭션 단위로 움직임
조건절이 있으며 조건이 없으면 데이터를 모두 가져옴
(조건이 없으면 limit라도 있어야 됨)
 Predicate 조건 보다는 BooleanExpression를 사용하면 composition이 가능함
 DB의 테스트 데이터가 중요한 이유는 데이터 변경이 있을 경우
테스트가 모두 깨지기 때문이다.
```

스프링 데이터 JPA 활용

```
페이징은 제로 인덱스(0부터 시작)
```

- [QueryDsl] Can not find symbol 에러 해결

Troubleshooting

Ref

- Querydsl Official Site
- Spring Boot에 QueryDSL을 사용해보자

사용자, 주제, 2013년, 2014년

From:
http://rwiki.repia.com/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://rwiki.repia.com/doku.php?id=wiki:spring:jpa:querydsl&rev=1664368684

Last update: **2022/09/28 21:38**