2팀\_이소정\_정리본

섹션 10

1. 다양한 jpa 방법 중 JPQL

- EntityManager.find()

- 테이블이 아닌 엔티티 객체를 대상으로 검색

- //검색

String jpql = "select m From Member m where m.name like ‘%hello%'"; List<Member> result = em.createQuery(jpql, Member.class) .getResultList();

- //검색

String jpql = "select m from Member m where m.age > 18";

List<Member> result = em.createQuery(jpql, Member.class) .getResultList();

실행된 SQL

select

m.id as id,

m.age as age,

m.USERNAME as USERNAME,

m.TEAM\_ID as TEAM\_ID

from

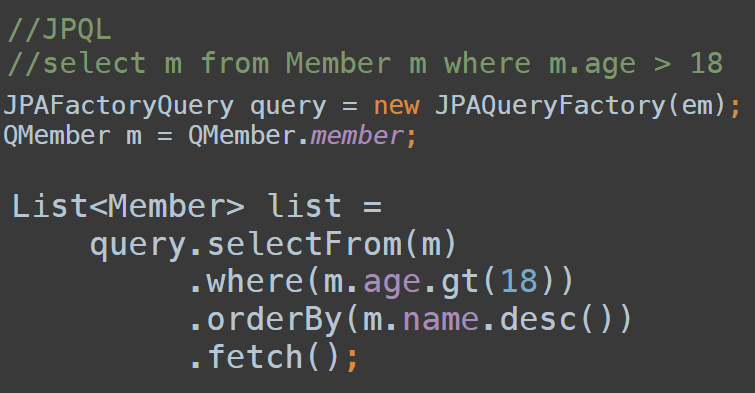
Member m

where

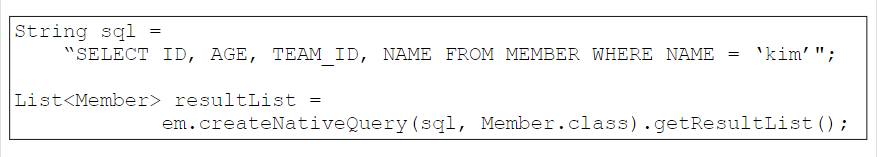
m.age>18

- criteria 소개 : jpql 빌더 역할, jpa 공식 기능, 너무 복잡하고 실용성이 없다 -> queryDSL 사용 권장

- QueryDSL : 실무 사용 권장

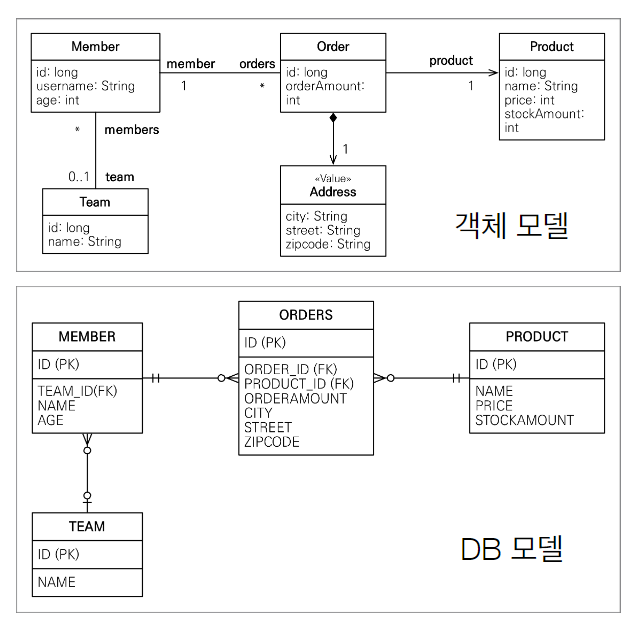


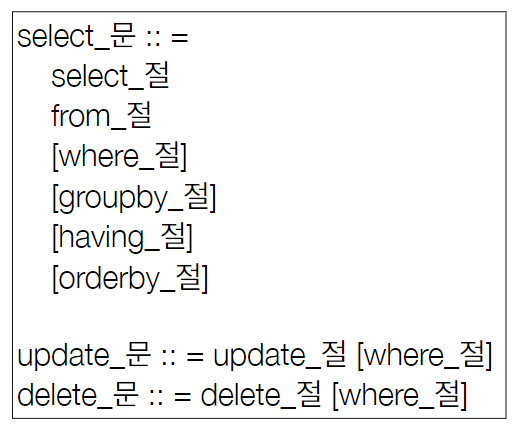
- 네이티브 SQL : JPA가 제공하는 SQL을 사용하는 기능

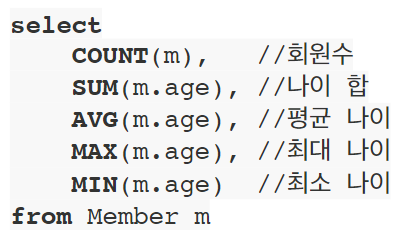


2. JPQL 소개

: 테이블 대상으로 쿼리하는 것이 아니라 엔티티 객체를 대상으로 쿼리한다.







- 결과 조회 API

: query.getResultList() : 결과가 하나 이상일때

query.getSingleResult() : 결과가 정확히 하나

- 프로젝션 :

SELECT m FROM Member m -> 엔티티프로젝션

SELECT m.team FROM Member m -> 엔티티프로젝션

SELECT m.address FROM Member m -> 임베디드타입프로젝션

SELECT m.username, m.age FROM Member m -> 스칼라타입프로젝션

- 페이징 API :

setFirstResult(int startPosition) : 조회시작위치(0부터시작)

setMaxResults(int maxResult) : 조회할데이터수

- 조인 :

내부조인:SELECT m FROM Member m [INNER] JOIN m.team t

외부조인:SELECT m FROM Member m LEFT [OUTER] JOIN m.team t

세타조인: select count(m) from Member m, Team t where m.username = t.name

-서브 쿼리 지원 함수

[NOT] EXISTS (subquery): 서브쿼리에 결과가 존재하면 참

{ALL | ANY | SOME} (subquery)

ALL 모두 만족하면 참

ANY, SOME: 같은의미, 조건을하나라도만족하면참

[NOT] IN (subquery): 서브쿼리의 결과 중 하나라도 같은 것이 있으면 참

- JPQL 표현

: 문자: ‘HELLO’, ‘She’’s’

•숫자: 10L(Long), 10D(Double), 10F(Float)

•Boolean: TRUE, FALSE

•ENUM: jpabook.MemberType.Admin (패키지명포함)

•엔티티타입: TYPE(m) = Member (상속관계에서사용)

- 조건식 (CASE식)

: COALESCE: 하나씩 조회해서 null이 아니면 반환

•NULLIF: 두값이 같으면 null 반환, 다르면 첫번째 값 반환