

9-2. 즉시 로딩과 지연 로딩

지연 로딩

`@ManyToOne(fetch = FetchType.LAZY)` 를 사용하여 지연 로딩을 함.

member는 많이 쓰고 team은 잘 안쓰는데 member만 조회하려면?

지연 로딩!

=> **team**을 직접 쓸 때만 쿼리가 날아간다. : `findMember.getTeam().getName()` 같은 형식..

team은 프록시를 가져오고, 만약 진짜로 team을 쓸 때만 쿼리가 날아가 프록시가 초기화된다.

→ 실무에서는 항상 지연 로딩. 만약 즉시 로딩이 필요하다면 fetch join같은 방법을 씬.

즉시 로딩

`@ManyToOne(fetch = FetchType.EAGER)` 를 사용하여 즉시 로딩을 함.

em.find로 member를 찾을 때 바로 팀을 로딩한다.

=> member, team에 대한 쿼리가 둘 다 날아간다.

→ JPQL에서 N+1문제가 일어남. (처음 쿼리를 날렸는데 걸려있는 것 때문에 하나가 더 나감)

```
List<Member> m = em.createQuery("select m from Member m", Member.class)
    .getResultList()
```

이 때 즉시 로딩이기 때문에 멤버도 쿼리가 나가고 멤버에 걸려있는 팀들까지 모두 쿼리가 나가야한다.

#JPA