

# Spring boot 5일차

## Keyword

도메인 주도 개발 - DDD (Domain - Driven Development)

착각

- DB를 잘 몰라도 JPA 를 충분히 사용할 수 있다? NONO
- 모델링도 잘 몰라도 된다? freeing No

단일 테이블 => 엔티티

- 테이블 하나당 엔티티 여러개가 될 수 있음.
- 즉 테이블의 수와 엔티티의 수는 일치하지 않음 -> 테이블 != 엔티티

회원 테이블 - 회원 권한 테이블 => 2개의 테이블이지만 이건 하나의 인테테

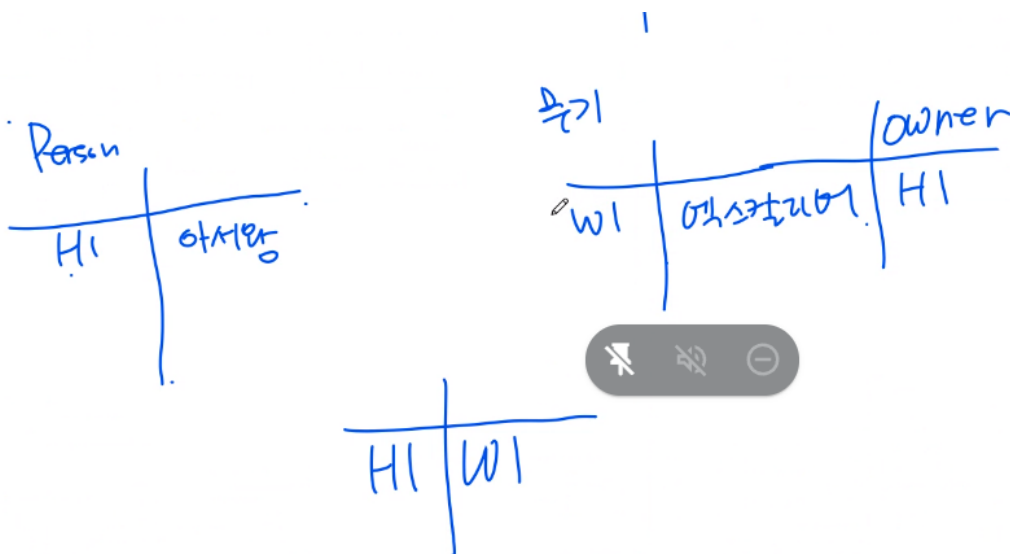
JPA

- 엔티티
- 값객체

## 요약

1. 테이블 수 != 엔티티 수
2. 테이블/클래스 => 의미가 생김
3. 데이터 베이스에서는 1:1, 다대일. 다다다 => JPA 들어가면 존나 복잡해진다고 한다.

관계형 데이터 베이스



관계형 데이터 베이스

## 회원 데이터

아이디	이름	패스워드
user1	사용자1	111
user2	사용자2	111
user3	사용자3	111
user4	사용자4	111
user5	사용자5	111

## 게시글 데이터

번호	제목	작성자	내용
1	오늘 날씨가..	user1	
2	집에 오늘 길은..	user2	
3	차가 막혀서..	user3	
4	어이 없는 일이..	user4	
5	감기 조심하세요..	user1	
6	이번 개봉 영화..	user2	
7	오늘 점심에..	user1	

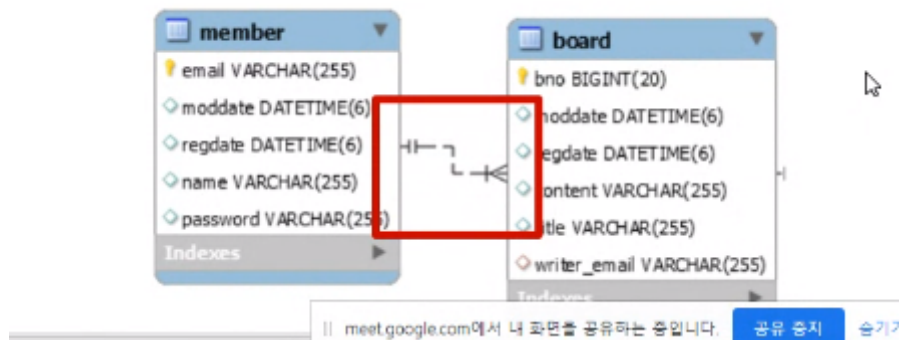
- OneToOne

- 명제 1 : 한명의 회원은 여러개의 게시글

- OneToMany

- 명제 2 : 하나의 게시글은 한명의 회원에 의해서 작성

★★★★★★ 문장으로 판단하지 말고 데이터로 판단 ★★★★★★



=> 하지만 DB의 관계(ERD)로 봤을때 명제 1일 맞다는 것을 확인 할 수 있다.

관계의 주어는 fk 에서부터 읽는다. 관계라는것은 fk 가 있어야 형성될 수 있는 것이기 때문에 fk 부터 읽는게 맞다.

그래서 한명의 회원이 여러개의 게시글을 작성할 수 있다는 것이다.

## 요약

해석할때 ERD를 그려서 해석해라

그리고 fk 에서부터 읽어 들어가라.

## ★★ 원칙!!!! ★★

1. 무조건 ERD 기준
2. FK 기준으로 볼것 - @ManyToOne - fk 가 관계의 원칙 - 댓글이 ManyToOne 의 관계이다.
3. 단방향 과 양방향 중 단방향 우선으로 생각할 것.

JPA 관련 검색할때는 1년 미만까 보기

## 오늘 댓글 작업

entity

Reply 클래스 생성

```
@Entity
@Builder
@AllArgsConstructor
@NoArgsConstructor
@Getter
@ToString
public class Reply {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long rno;

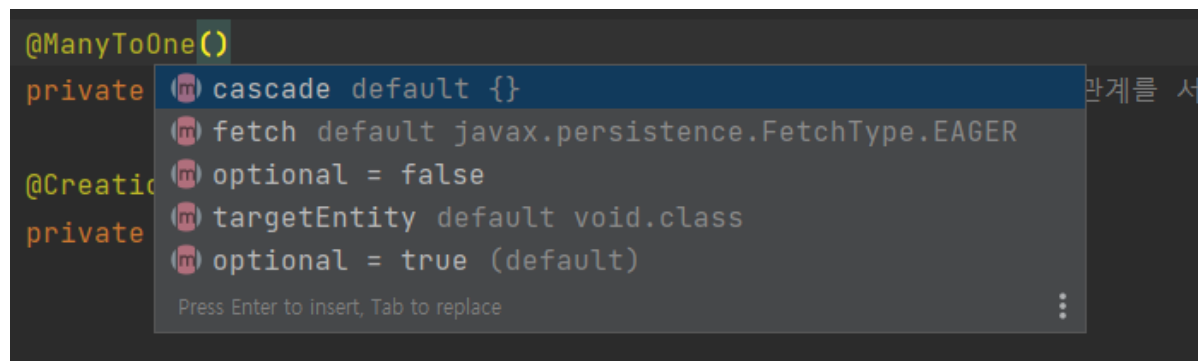
    private String replyText;

    private String replyer;

    // 이부분이 중요!!!
    @ManyToOne(fetch = FetchType.LAZY) // 일단 LAZY를 걸어주고 시작하는게 정석 - 왜냐
    // 쓸데없이 보드를 가져오면 x
    private Board board; // 애는 JPA 이기 때문에 entity 와 entity 의 관계를 서술화지
    // 않으면 오류남

    @CreationTimestamp
    private LocalDateTime replyDate;

}
```



fetch 란? - reply 를 가져올때 Board table 도 같이 로딩해서 가져와야해! 라고 해주는 것. 아니면 Board 는 나중에 필요할때까지 안가져와도 되 -> 쓸데없이 보드를 가져오면 안됨!!!! 그래서 lazy를 걸어줘야함.

관계형 DB에서 다 단방향 참조임.

**Reply는 entity 이기 때문에!! 독자적인 repository 가 필요하다!!!**

ReplyRepository 인터페이스 생성

```
public interface ReplyRepository extends JpaRepository<Reply, Long> {

}
```

테스트 ㄱㄱ

ReplyRepositoryTests 생성

```
@SpringBootTest
@Log4j2
public class ReplyRepositoryTests {

    @Autowired
    private ReplyRepository replyRepository;

    @Test
    public void insert200() {
        IntStream.rangeClosed(1,200).forEach(i -> {
            Long bno= (Long)(200 - (1 % 5)); // 200, 199, 198...

            int replyCount = (i % 5); // 0, 1, 2, 3, 4

            IntStream.rangeClosed(0, replyCount).forEach(j -> {

                Board board = Board.builder().bno(bno).build();

                // 이게 단방향 참조이다. reply 에서만 board 를 물고있는 구조이다. fk 위
                Reply reply = Reply.builder()
                    .replyText("Reply....")
                    .replyer("replyer...")
                    .board(board) // board 객체를 만들어서 reply 에 board 에 주
                    .build();

                replyRepository.save(reply);
            }); //inner loop

        }); // outer loop
    }
}
```

테스트 결과

```

2021-10-08 11:18:27.996 INFO 13976 --- [    Test worker] o.s.b.a.w.s
2021-10-08 11:18:28.615 INFO 13976 --- [    Test worker] o.z.sb.repos
Hibernate:
    insert
    into
        reply
        (board_bno, reply_date, reply_text, replyer)
    values
        (?, ?, ?, ?)
Hibernate:
    insert
    into
        reply
        (board_bno, reply_date, reply_text, replyer)
    values
        (?, ?, ?, ?)
Hibernate:
    insert
    into
        reply
        (board_bno, reply_date, reply_text, replyer)
    values
        (?, ?, ?, ?)
Hibernate:
    insert
    into
        reply
        (board_bno, reply_date, reply_text, replyer)
    values
        (?, ?, ?, ?)

```

200개가 insert 됨

그다음 db 확인하면 데이터가 들어가있음

그다음

에러나는 경우를 예시로 듦

```

@Test
public void testRead() {
    Long rn = 1L;
    Reply reply = replyRepository.findById(rn).get();

    log.info(reply);
}

```

이러면 에러남, 조회에서 에러남

```

2021-10-08 11:27:05.006 INFO 20612 --- [    Test worker] o.z.sb.repository.ReplyRepositoryTests : Started ReplyRepos
Hibernate:
select
    reply0_.rno as rno1_1_0_,
    reply0_.board_bno as board_bn5_1_0_,
    reply0_.reply_date as reply_da2_1_0_,
    reply0_.reply_text as reply_te3_1_0_,
    reply0_.replyer as replyer4_1_0_
from
    reply reply0_
where
    reply0_.rno=?

could not initialize proxy [org.zerock.sb.entity.Board#199] - no Session
org.hibernate.LazyInitializationException: could not initialize proxy [org.zerock.sb.entity.Board#199] - no Session
at app//org.hibernate.proxy.AbstractLazyInitializer.initialize(AbstractLazyInitializer.java:170)

```

위에 select 가 있고 no session 에러가 남.

reply 테이블을 select 를 했는데 no session 에러 - 데이터베이스 컨넥션 에러

board 에도 toString이 찍혀야 하는데 board 는 지연로딩에 의해 lazy 상태라서 안불러와짐

lazy 를 고쳐줘야함

entity - Reply ㄱㄱ

```

@Entity
@Builder
@AllArgsConstructor
@NoArgsConstructor
@Getter
@ToString(exclude = "board") // 수정 // LAZY 와 exclude 는 항상 같이 붙어다녀야 한다.
public class Reply {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long rno;

    private String replyText;

    private String replyer;

    @ManyToOne(fetch = FetchType.LAZY) // LAZY 유지
    private Board board; // 애는 JPA 이기 때문에 entity 와 entity 의 관계를 서술화지
    않으면 오류남

    @CreationTimestamp
    private LocalDateTime replyDate;

}

```

ReplyRepository ㄱㄱ

```
public interface ReplyRepository extends JpaRepository<Reply, Long> {

    List<Reply> findReplyByBoard_BnoOrderByRno(Long bno);

}
```

Test ㄱㄱ

ReplyRepositoryTests ㄱㄱ

```
@Test
public void testByBno() {
    Long bno = 200L;
    List<Reply> replyList = replyRepository.findReplyByBoard_BnoOrderByRno(bno);

    replyList.forEach(reply -> log.info(reply));
}
```

테스트 결과

```
2021-10-08 11:54:10.452 INFO 9600 --- [ Test worker] o.z.sb.repository.ReplyRepositoryTests : Started ReplyRepositoryTests in 6.402 seconds (JVM
Hibernate:
select
  reply0_.rno as rno1_1_,
  reply0_.board_bno as board_bn5_1_,
  reply0_.reply_date as reply_da2_1_,
  reply0_.reply_text as reply_te3_1_,
  reply0_.replier as replier4_1_
from
  reply reply0_
left outer join
  board board1_
    on reply0_.board_bno=board1_.bno
where
  board1_.bno=?
order by
  reply0_.rno asc
2021-10-08 11:54:10.808 INFO 9600 --- [ Test worker] o.z.sb.repository.ReplyRepositoryTests : Reply(rno=15, replyText=Reply..., replier=replier
2021-10-08 11:54:10.809 INFO 9600 --- [ Test worker] o.z.sb.repository.ReplyRepositoryTests : Reply(rno=30, replyText=Reply..., replier=replier
2021-10-08 11:54:10.809 INFO 9600 --- [ Test worker] o.z.sb.repository.ReplyRepositoryTests : Reply(rno=45, replyText=Reply..., replier=replier
2021-10-08 11:54:10.809 INFO 9600 --- [ Test worker] o.z.sb.repository.ReplyRepositoryTests : Reply(rno=60, replyText=Reply..., replier=replier
2021-10-08 11:54:10.809 INFO 9600 --- [ Test worker] o.z.sb.repository.ReplyRepositoryTests : Reply(rno=75, replyText=Reply..., replier=replier
```

## 댓글 개수 찍기

BoardRepository ㄱㄱ

영관관계?

```
public interface BoardRepository extends JpaRepository<Board, Long>, BoardSearch
{ // extends 는 여러개 가능

    @Query("select b, count(r) from Board b left join Reply r on r.board = b
group by b ")
    Page<Object[]> ex1(Pageable pageable);

}
```

test ㄱㄱ

```
@Test
public void testEx1() {

    Pageable pageable = PageRequest.of(0, 10, Sort.by("bno").descending());

    Page<Object[]> result = boardRepository.ex1(pageable);

    log.info(result);
}
```

실행

```
2021-10-08 12:04:51.144 INFO 23300 --- [    Test worker] o.z.sb.repository.
Hibernate:
    select
        board0_.bno as col_0_0_,
        count(reply1_.rno) as col_1_0_,
        board0_.bno as bno1_0_,
        board0_.content as content2_0_,
        board0_.mod_date as mod_date3_0_,
        board0_.reg_date as reg_date4_0_,
        board0_.title as title5_0_,
        board0_.writer as writer6_0_
    from
        board board0_
    left outer join
        reply reply1_
        on (
            reply1_.board_bno=board0_.bno
        )
    group by
        board0_.bno
    order by
        board0_.bno desc limit ?
Hibernate:
```



```

board0_.bno desc limit ?
Hibernate:
    select
        count(board0_.bno) as col_0_0_
    from
        board board0_
    left outer join
        reply reply1_
        on (
            reply1_.board_bno=board0_.bno
        )
    group by
        board0_.bno

```

```

2021-10-08 12:04:51.400 INFO 23300 --- [    Test work
2021-10-08 12:04:51.424 INFO 23300 --- [ionShutdownHo

```

sql 을 모르면 JPA 를 하기가 어렵다.

이력서에 또는 자소서에서 JPA를 하던 도중 sql 에 대한 어려움을 느꼈고 sql 문을 더 공부했다! 이런식으로.

12시 15분쯤 뭔가를 놓쳤다.

테스트 수정

```

@Test
public void testEx1() {

    Pageable pageable = PageRequest.of(0, 10, Sort.by("bno").descending());

    Page<Object[]> result = boardRepository.ex1(pageable);

    log.info(result);

    result.get().forEach(element -> { // 이게 배열 안에 배열이 있는 것이다.

        Object[] arr = (Object[])element;

        log.info(Arrays.toString(arr));
        // 배열안에 배열은 이렇게 Arrays.toString() 으로 inner 배열을 뽑아 올수 있나보
다.

    });
}

```

그후 실행

```
ts : Page 1 of 21 containing [Ljava.lang.Object; instances
ts : [Board(bno=202, title=한글한글, content=한글 한글 내용, writer=후하하, regDate=2021-10-07T11:58:04.861031, modDate=2021-10-07T12:30:41.098853), 0]
ts : [Board(bno=201, title=안녕, content=후후후후r, writer=하하, regDate=2021-10-07T11:57:11.783742, modDate=2021-10-07T11:57:11.783742), 0]
ts : [Board(bno=200, title=그림지 그림지, content=작동 되어라, writer=user0, regDate=2021-10-07T09:50:09.100365, modDate=2021-10-07T12:34:38.764983), 40]
ts : [Board(bno=199, title=Title199, content=Content199, writer=user9, regDate=2021-10-07T09:50:09.095379, modDate=2021-10-07T09:50:09.095379), 80]
ts : [Board(bno=198, title=Title198, content=Content198, writer=user8, regDate=2021-10-07T09:50:09.088397, modDate=2021-10-07T09:50:09.088397), 120]
ts : [Board(bno=197, title=Title197, content=Content197, writer=user7, regDate=2021-10-07T09:50:09.083416, modDate=2021-10-07T09:50:09.083416), 160]
ts : [Board(bno=196, title=Title196, content=Content196, writer=user6, regDate=2021-10-07T09:50:09.077428, modDate=2021-10-07T09:50:09.077428), 200]
ts : [Board(bno=195, title=Title195, content=Content195, writer=user5, regDate=2021-10-07T09:50:09.072440, modDate=2021-10-07T09:50:09.072440), 0]
ts : [Board(bno=194, title=Title194, content=Content194, writer=user4, regDate=2021-10-07T09:50:09.067455, modDate=2021-10-07T09:50:09.067455), 0]
ts : [Board(bno=193, title=Title193, content=Content193, writer=user3, regDate=2021-10-07T09:50:09.062467, modDate=2021-10-07T09:50:09.062467), 0]
Bean : Closing JPA EntityManagerFactory for persistence unit 'default'
      : HikariPool-1 - Shutdown initiated...
      : HikariPool-1 - Shutdown completed.
```

---

## 오늘 요약

entity는 하나 하지만 DTO 는 여러개

- 예를들어 PageRequestDTO PageResponseDTO 같은 SQL 에 집어 넣는용도랑 데이터를 뺄때 사용하는 용도.