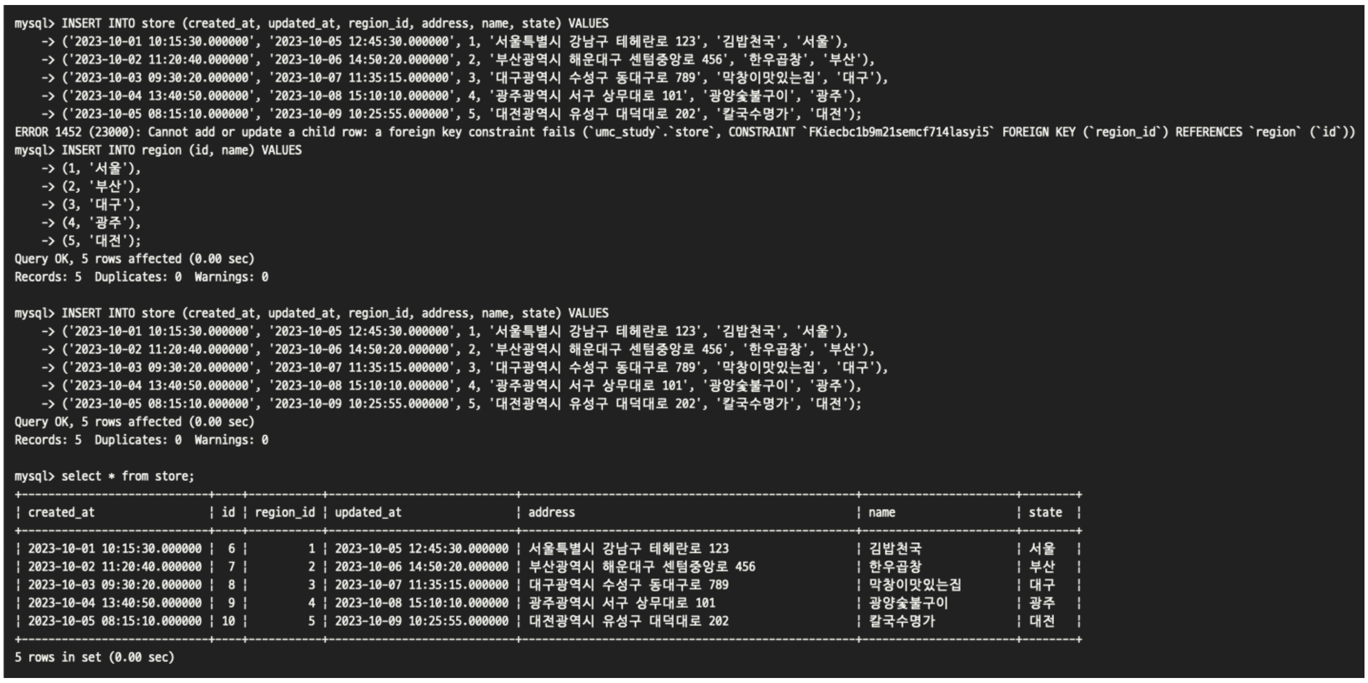
실습

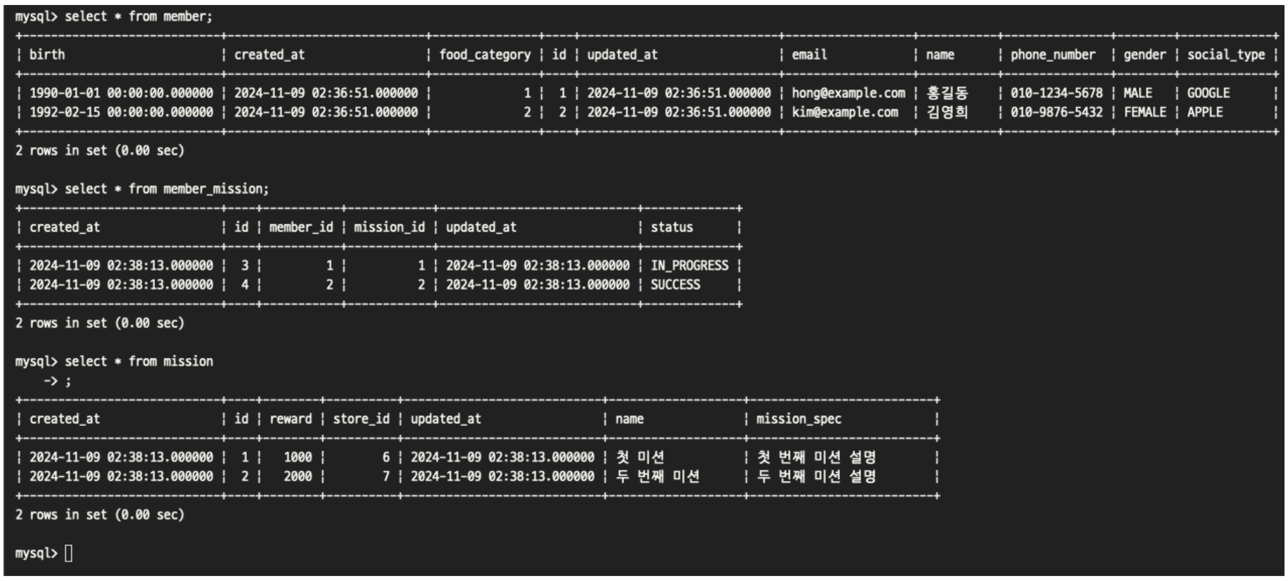
store 더미데이터



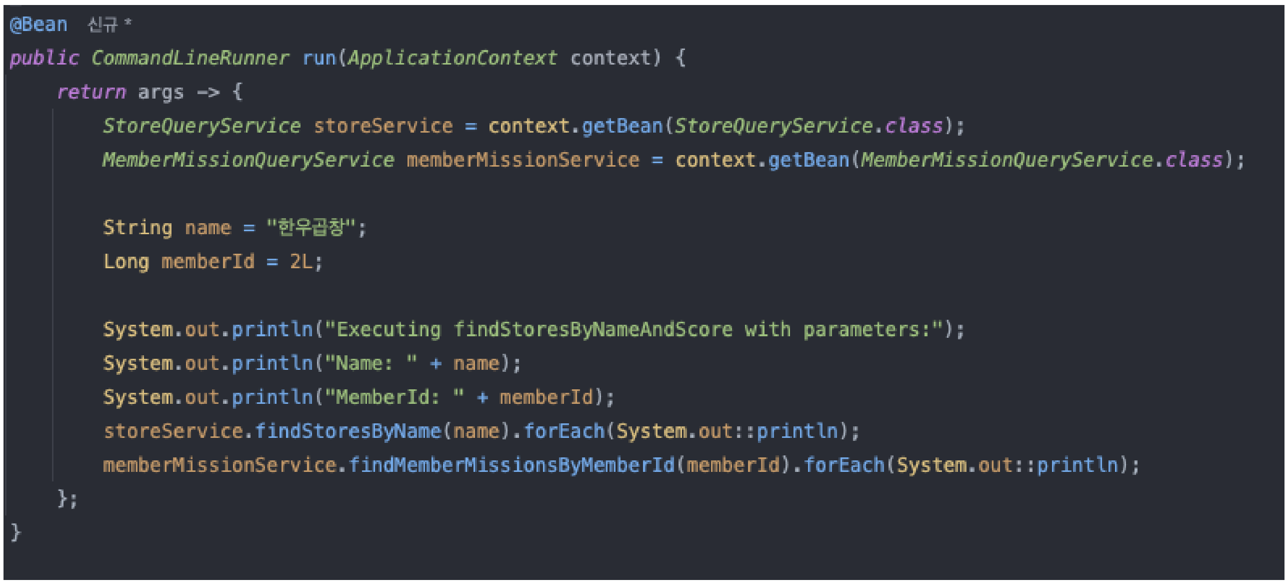
```
mysql> INSERT INTO store (created_at, updated_at, region_id, address, name, state) VALUES
    -> ('2023-10-01 10:15:30.000000', '2023-10-05 12:45:30.000000', 1, '서울특별시 강남구 테헤란로 123', '김밥천국', '서울'),
    -> ('2023-10-02 11:20:40.000000', '2023-10-06 14:50:20.000000', 2, '부산광역시 해운대구 센텀중앙로 456', '한우곱창', '부산'),
    -> ('2023-10-03 09:30:20.000000', '2023-10-07 11:35:15.000000', 3, '대구광역시 수성구 동대구로 789', '막창이맛있는집', '대구'),
    -> ('2023-10-04 13:40:50.000000', '2023-10-08 15:10:10.000000', 4, '광주광역시 서구 상무대로 101', '광양숯불구이', '광주'),
    -> ('2023-10-05 08:15:10.000000', '2023-10-09 10:25:55.000000', 5, '대전광역시 유성구 대덕대로 202', '칼국수명가', '대전');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`umc_study`.`store`, CONSTRAINT `FKiecbc1b9m21semcf714lasyi5` FOREIGN KEY (`region_id`) REFERENCES `region` (`id`))
mysql> INSERT INTO region (id, name) VALUES
    -> (1, '서울'),
    -> (2, '부산'),
    -> (3, '대구'),
    -> (4, '광주'),
    -> (5, '대전');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> INSERT INTO store (created_at, updated_at, region_id, address, name, state) VALUES
    -> ('2023-10-01 10:15:30.000000', '2023-10-05 12:45:30.000000', 1, '서울특별시 강남구 테헤란로 123', '김밥천국', '서울'),
    -> ('2023-10-02 11:20:40.000000', '2023-10-06 14:50:20.000000', 2, '부산광역시 해운대구 센텀중앙로 456', '한우곱창', '부산'),
    -> ('2023-10-03 09:30:20.000000', '2023-10-07 11:35:15.000000', 3, '대구광역시 수성구 동대구로 789', '막창이맛있는집', '대구'),
    -> ('2023-10-04 13:40:50.000000', '2023-10-08 15:10:10.000000', 4, '광주광역시 서구 상무대로 101', '광양숯불구이', '광주'),
    -> ('2023-10-05 08:15:10.000000', '2023-10-09 10:25:55.000000', 5, '대전광역시 유성구 대덕대로 202', '칼국수명가', '대전');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from store;
+----------------------------+----+-----------+----------------------------+------------------------------------------------+----------------------+--------+
| created_at                 | id | region_id | updated_at                 | address                                        | name                 | state  |
+----------------------------+----+-----------+----------------------------+------------------------------------------------+----------------------+--------+
| 2023-10-01 10:15:30.000000 |  6 |         1 | 2023-10-05 12:45:30.000000 | 서울특별시 강남구 테헤란로 123                 | 김밥천국             | 서울   |
| 2023-10-02 11:20:40.000000 |  7 |         2 | 2023-10-06 14:50:20.000000 | 부산광역시 해운대구 센텀중앙로 456             | 한우곱창             | 부산   |
| 2023-10-03 09:30:20.000000 |  8 |         3 | 2023-10-07 11:35:15.000000 | 대구광역시 수성구 동대구로 789                  | 막창이맛있는집       | 대구   |
| 2023-10-04 13:40:50.000000 |  9 |         4 | 2023-10-08 15:10:10.000000 | 광주광역시 서구 상무대로 101                   | 광양숯불구이         | 광주   |
| 2023-10-05 08:15:10.000000 | 10 |         5 | 2023-10-09 10:25:55.000000 | 대전광역시 유성구 대덕대로 202                 | 칼국수명가           | 대전   |
+----------------------------+----+-----------+----------------------------+------------------------------------------------+----------------------+--------+
5 rows in set (0.00 sec)
```

미션

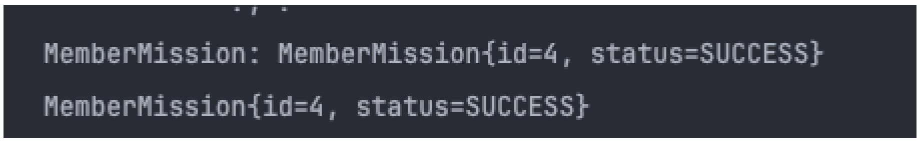
1. Member에게 해당되는 미션 데이터 조회



```
mysql> select * from member;
+----------------------------+----------------------------+---------------+----+----------------------------+------------------+----------+---------------+--------+-------------+
| birth                      | created_at                 | food_category | id | updated_at                 | email            | name     | phone_number  | gender | social_type |
+----------------------------+----------------------------+---------------+----+----------------------------+------------------+----------+---------------+--------+-------------+
| 1990-01-01 00:00:00.000000 | 2024-11-09 02:36:51.000000 |             1 |  1 | 2024-11-09 02:36:51.000000 | hong@example.com | 홍길동   | 010-1234-5678 | MALE   | GOOGLE      |
| 1992-02-15 00:00:00.000000 | 2024-11-09 02:36:51.000000 |             2 |  2 | 2024-11-09 02:36:51.000000 | kim@example.com  | 김영희   | 010-9876-5432 | FEMALE | APPLE       |
+----------------------------+----------------------------+---------------+----+----------------------------+------------------+----------+---------------+--------+-------------+
2 rows in set (0.00 sec)

mysql> select * from member_mission;
+----------------------------+----+-----------+------------+----------------------------+-------------+
| created_at                 | id | member_id | mission_id | updated_at                 | status      |
+----------------------------+----+-----------+------------+----------------------------+-------------+
| 2024-11-09 02:38:13.000000 |  3 |         1 |          1 | 2024-11-09 02:38:13.000000 | IN_PROGRESS |
| 2024-11-09 02:38:13.000000 |  4 |         2 |          2 | 2024-11-09 02:38:13.000000 | SUCCESS     |
+----------------------------+----+-----------+------------+----------------------------+-------------+
2 rows in set (0.00 sec)

mysql> select * from mission
    -> ;
+----------------------------+----+--------+----------+----------------------------+------------------+-------------------------+
| created_at                 | id | reward | store_id | updated_at                 | name             | mission_spec            |
+----------------------------+----+--------+----------+----------------------------+------------------+-------------------------+
| 2024-11-09 02:38:13.000000 |  1 |   1000 |        6 | 2024-11-09 02:38:13.000000 | 첫 미션          | 첫 번째 미션 설명       |
| 2024-11-09 02:38:13.000000 |  2 |   2000 |        7 | 2024-11-09 02:38:13.000000 | 두 번째 미션     | 두 번째 미션 설명       |
+----------------------------+----+--------+----------+----------------------------+------------------+-------------------------+
2 rows in set (0.00 sec)

mysql> 
```

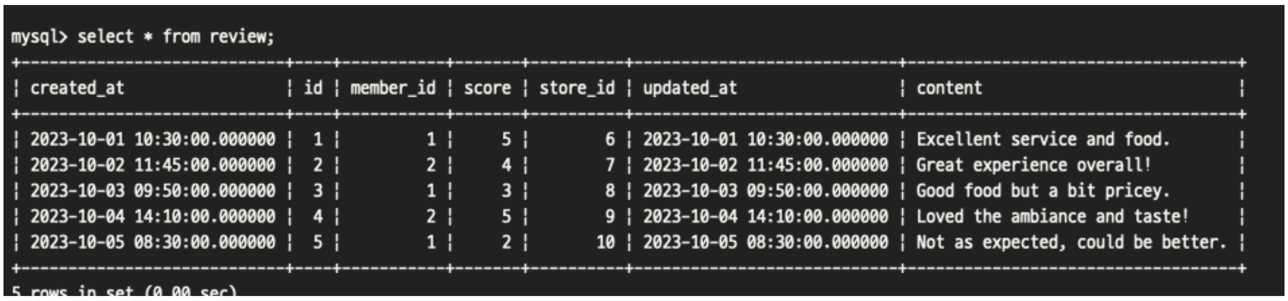


```java
@Bean  신규 *
public CommandLineRunner run(ApplicationContext context) {
    return args -> {
        StoreQueryService storeService = context.getBean(StoreQueryService.class);
        MemberMissionQueryService memberMissionService = context.getBean(MemberMissionQueryService.class);

        String name = "한우곱창";
        Long memberId = 2L;

        System.out.println("Executing findStoresByNameAndScore with parameters:");
        System.out.println("Name: " + name);
        System.out.println("MemberId: " + memberId);
        storeService.findStoresByName(name).forEach(System.out::println);
        memberMissionService.findMemberMissionsByMemberId(memberId).forEach(System.out::println);
    };
}
```

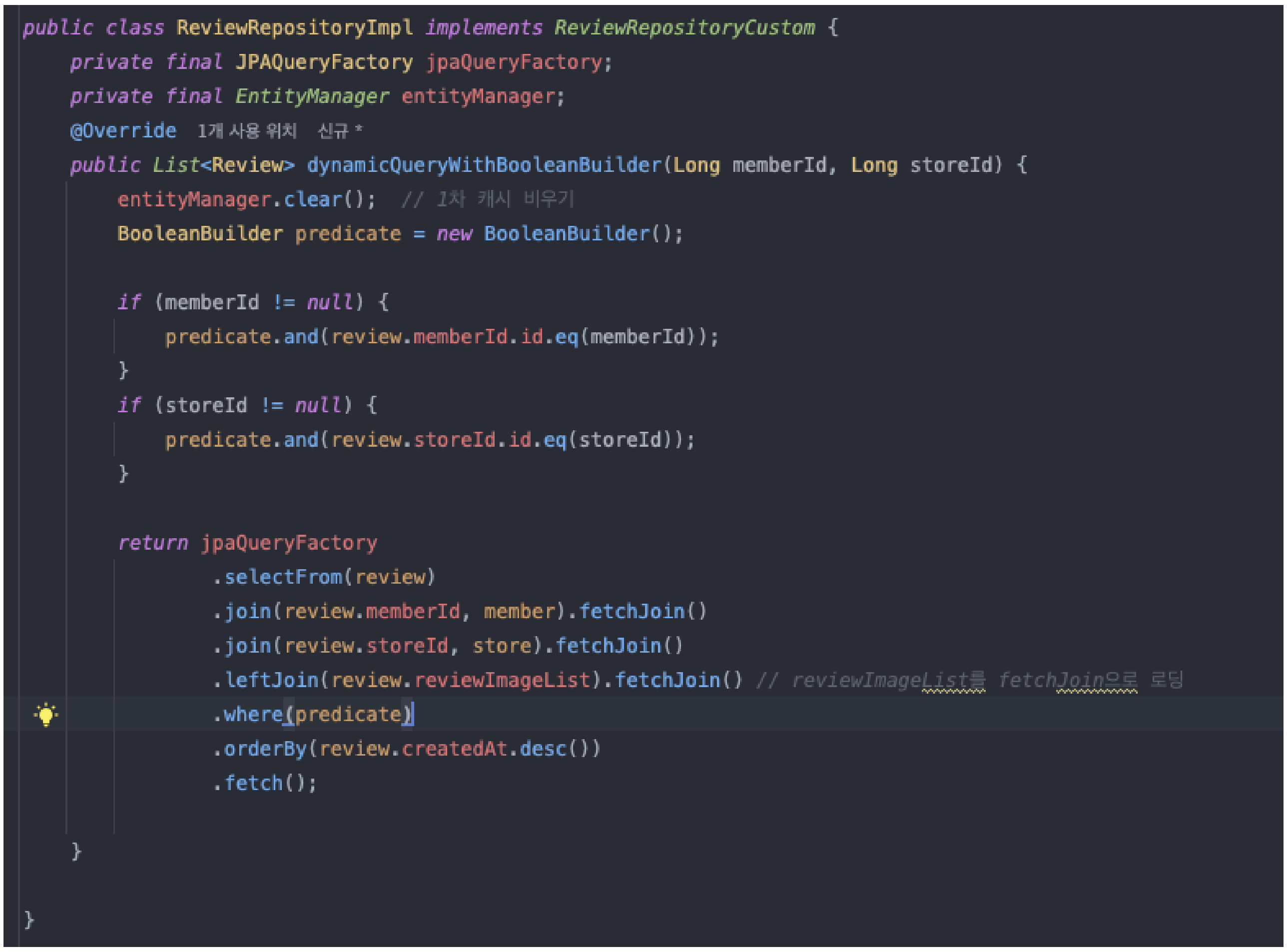


```
MemberMission: MemberMission{id=4, status=SUCCESS}
MemberMission{id=4, status=SUCCESS}
```

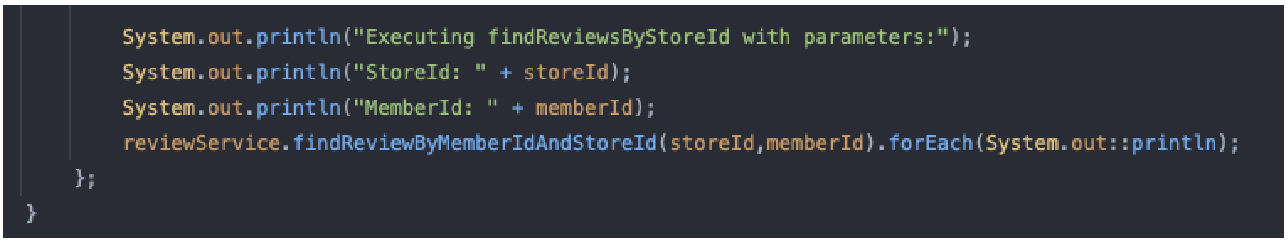
2. 해당 가게에 해당되는 리뷰 데이터 조회



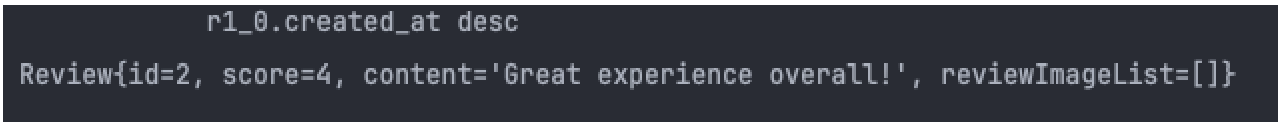
```
mysql> select * from review;
+----------------------------+----+-----------+-------+----------+----------------------------+-----------------------------------+
| created_at                 | id | member_id | score | store_id | updated_at                 | content                           |
+----------------------------+----+-----------+-------+----------+----------------------------+-----------------------------------+
| 2023-10-01 10:30:00.000000 |  1 |         1 |     5 |        6 | 2023-10-01 10:30:00.000000 | Excellent service and food.       |
| 2023-10-02 11:45:00.000000 |  2 |         2 |     4 |        7 | 2023-10-02 11:45:00.000000 | Great experience overall!         |
| 2023-10-03 09:50:00.000000 |  3 |         1 |     3 |        8 | 2023-10-03 09:50:00.000000 | Good food but a bit pricey.       |
| 2023-10-04 14:10:00.000000 |  4 |         2 |     5 |        9 | 2023-10-04 14:10:00.000000 | Loved the ambiance and taste!     |
| 2023-10-05 08:30:00.000000 |  5 |         1 |     2 |       10 | 2023-10-05 08:30:00.000000 | Not as expected, could be better. |
+----------------------------+----+-----------+-------+----------+----------------------------+-----------------------------------+
5 rows in set (0.00 sec)
```



```java
public class ReviewRepositoryImpl implements ReviewRepositoryCustom {
    private final JPAQueryFactory jpaQueryFactory;
    private final EntityManager entityManager;
    @Override  1개 사용 위치  신규 *
    public List<Review> dynamicQueryWithBooleanBuilder(Long memberId, Long storeId) {
        entityManager.clear();  // 1차 캐시 비우기
        BooleanBuilder predicate = new BooleanBuilder();

        if (memberId != null) {
            predicate.and(review.memberId.id.eq(memberId));
        }
        if (storeId != null) {
            predicate.and(review.storeId.id.eq(storeId));
        }

        return jpaQueryFactory
                .selectFrom(review)
                .join(review.memberId, member).fetchJoin()
                .join(review.storeId, store).fetchJoin()
                .leftJoin(review.reviewImageList).fetchJoin() // reviewImageList를 fetchJoin으로 로딩
                .where(predicate)
                .orderBy(review.createdAt.desc())
                .fetch();

    }

}
```

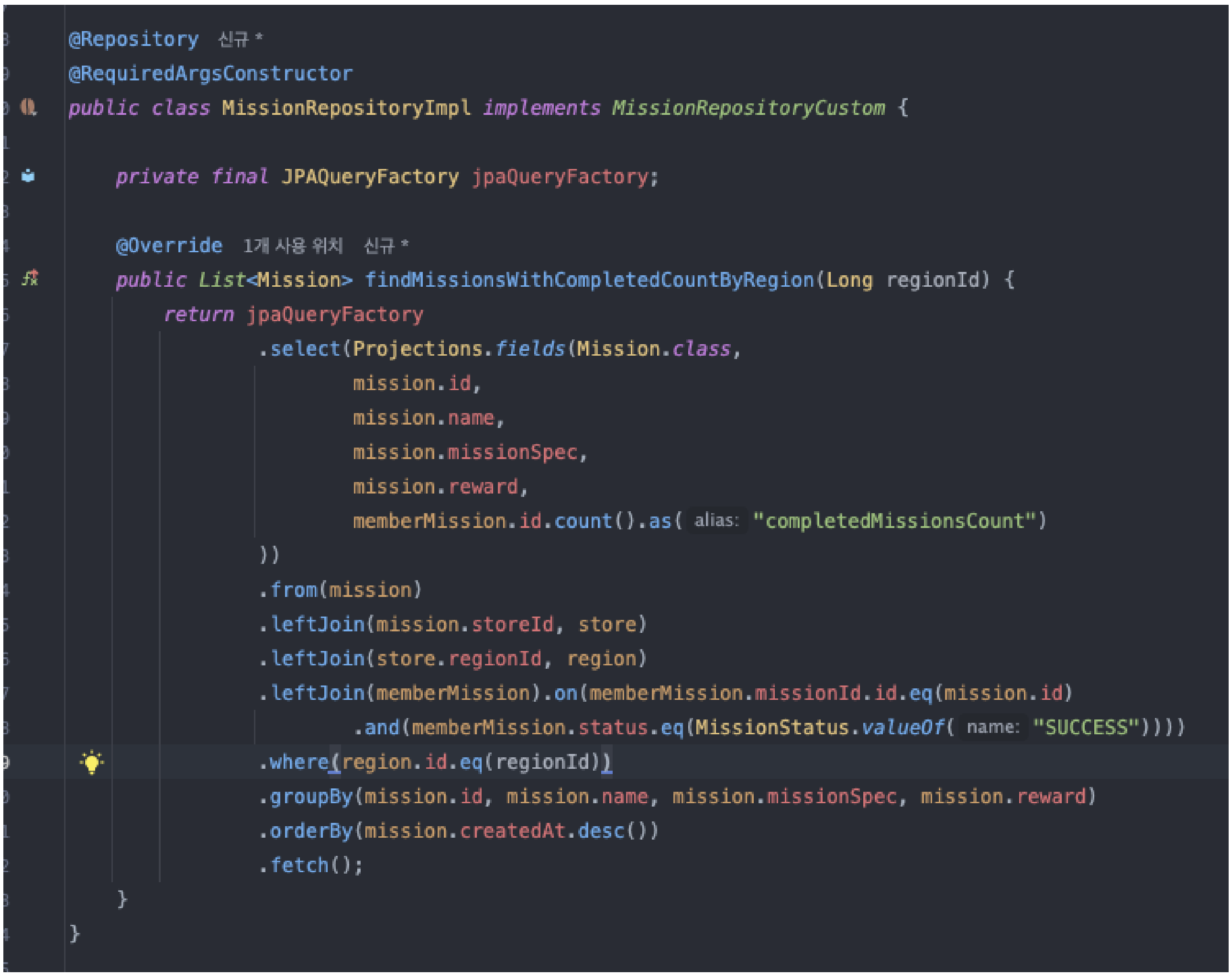


```java
        System.out.println("Executing findReviewsByStoreId with parameters:");
        System.out.println("StoreId: " + storeId);
        System.out.println("MemberId: " + memberId);
        reviewService.findReviewByMemberIdAndStoreId(storeId,memberId).forEach(System.out::println);
    };
}
```

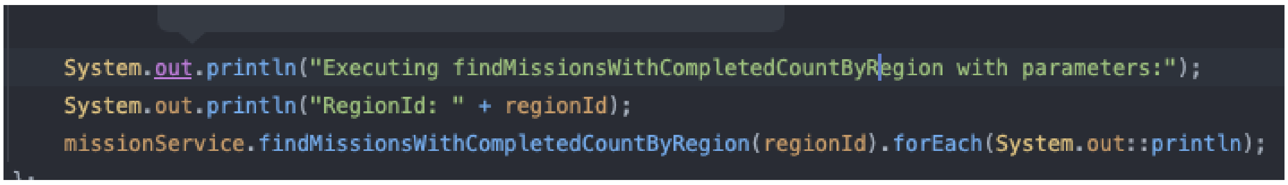


```
            r1_0.created_at desc
Review{id=2, score=4, content='Great experience overall!', reviewImageList=[]}
```

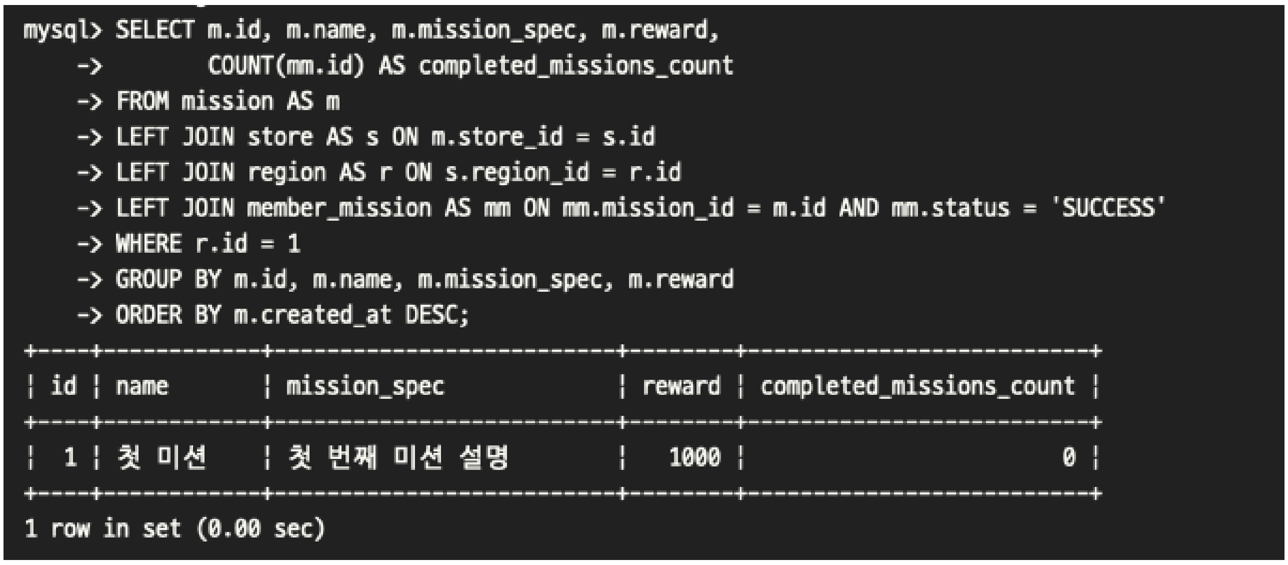
3. 홈 화면 쿼리 (현재 선택 된 지역에서 도전이 가능한 미션 목록, 페이징 포함)



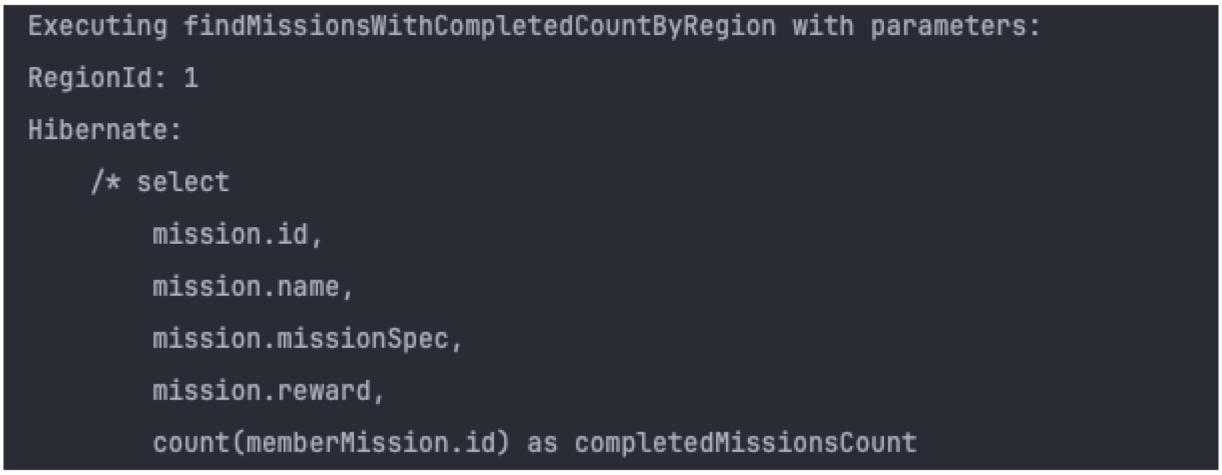
```
@Repository  신규 *
@RequiredArgsConstructor
public class MissionRepositoryImpl implements MissionRepositoryCustom {

    private final JPAQueryFactory jpaQueryFactory;

    @Override  1개 사용 위치  신규 *
    public List<Mission> findMissionsWithCompletedCountByRegion(Long regionId) {
        return jpaQueryFactory
                .select(Projections.fields(Mission.class,
                        mission.id,
                        mission.name,
                        mission.missionSpec,
                        mission.reward,
                        memberMission.id.count().as( alias: "completedMissionsCount")
                ))
                .from(mission)
                .leftJoin(mission.storeId, store)
                .leftJoin(store.regionId, region)
                .leftJoin(memberMission).on(memberMission.missionId.id.eq(mission.id)
                        .and(memberMission.status.eq(MissionStatus.valueOf( name: "SUCCESS"))))
                .where(region.id.eq(regionId))
                .groupBy(mission.id, mission.name, mission.missionSpec, mission.reward)
                .orderBy(mission.createdAt.desc())
                .fetch();
    }
}
```



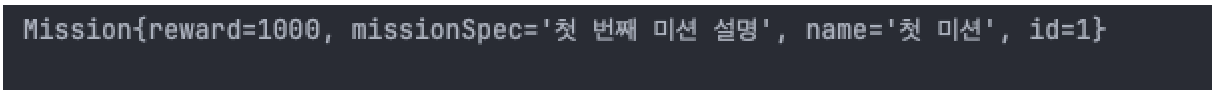
```
    System.out.println("Executing findMissionsWithCompletedCountByRegion with parameters:");
    System.out.println("RegionId: " + regionId);
    missionService.findMissionsWithCompletedCountByRegion(regionId).forEach(System.out::println);
```



```
mysql> SELECT m.id, m.name, m.mission_spec, m.reward,
    ->        COUNT(mm.id) AS completed_missions_count
    -> FROM mission AS m
    -> LEFT JOIN store AS s ON m.store_id = s.id
    -> LEFT JOIN region AS r ON s.region_id = r.id
    -> LEFT JOIN member_mission AS mm ON mm.mission_id = m.id AND mm.status = 'SUCCESS'
    -> WHERE r.id = 1
    -> GROUP BY m.id, m.name, m.mission_spec, m.reward
    -> ORDER BY m.created_at DESC;
+----+------------+---------------------------+--------+--------------------------+
| id | name       | mission_spec              | reward | completed_missions_count |
+----+------------+---------------------------+--------+--------------------------+
|  1 | 첫 미션    | 첫 번째 미션 설명         |   1000 |                        0 |
+----+------------+---------------------------+--------+--------------------------+
1 row in set (0.00 sec)
```

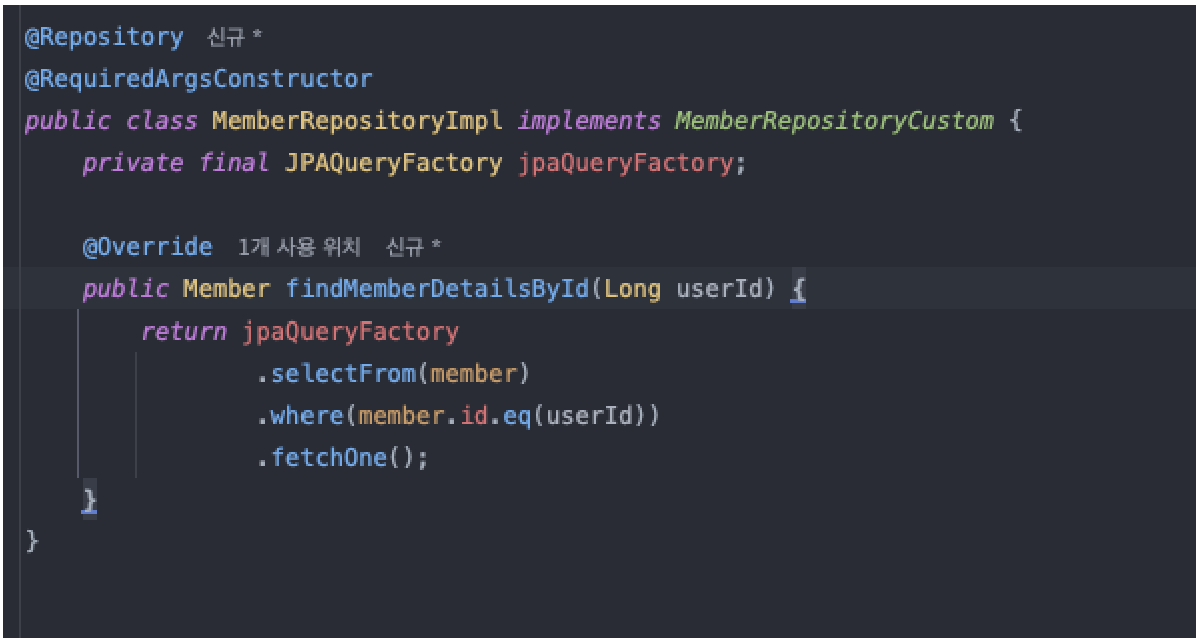


```
Executing findMissionsWithCompletedCountByRegion with parameters:
RegionId: 1
Hibernate:
    /* select
        mission.id,
        mission.name,
        mission.missionSpec,
        mission.reward,
        count(memberMission.id) as completedMissionsCount
```

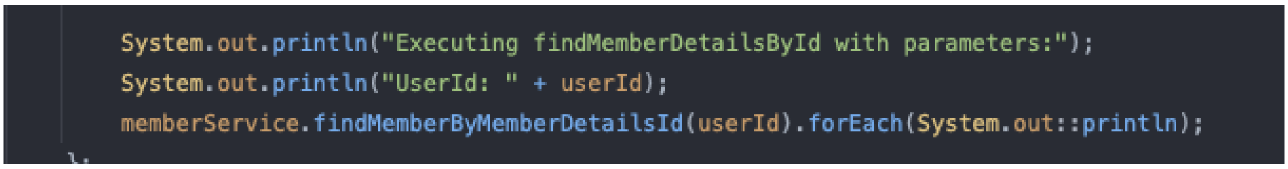


```
Mission{reward=1000, missionSpec='첫 번째 미션 설명', name='첫 미션', id=1}
```

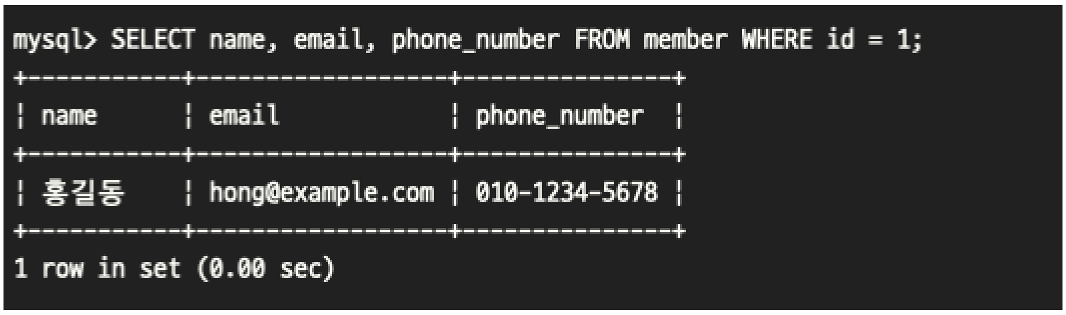
4. 마이 페이지 화면 쿼리



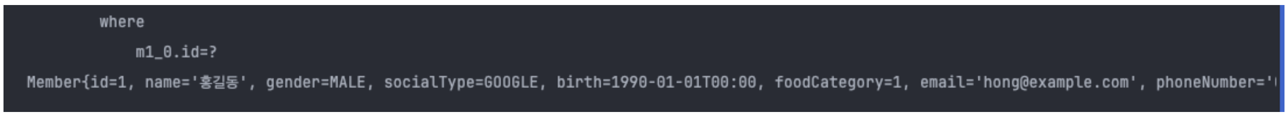
```
@Repository  신규 *
@RequiredArgsConstructor
public class MemberRepositoryImpl implements MemberRepositoryCustom {
    private final JPAQueryFactory jpaQueryFactory;

    @Override  1개 사용 위치  신규 *
    public Member findMemberDetailsById(Long userId) {
        return jpaQueryFactory
                .selectFrom(member)
                .where(member.id.eq(userId))
                .fetchOne();
    }
}
```



```
        System.out.println("Executing findMemberDetailsById with parameters:");
        System.out.println("UserId: " + userId);
        memberService.findMemberByMemberDetailsId(userId).forEach(System.out::println);
```



```
mysql> SELECT name, email, phone_number FROM member WHERE id = 1;
+-----------+------------------+--------------+
| name      | email            | phone_number |
+-----------+------------------+--------------+
| 홍길동    | hong@example.com | 010-1234-5678 |
+-----------+------------------+--------------+
1 row in set (0.00 sec)
```



```
        where
            m1_0.id=?
Member{id=1, name='홍길동', gender=MALE, socialType=GOOGLE, birth=1990-01-01T00:00, foodCategory=1, email='hong@example.com', phoneNumber='
```