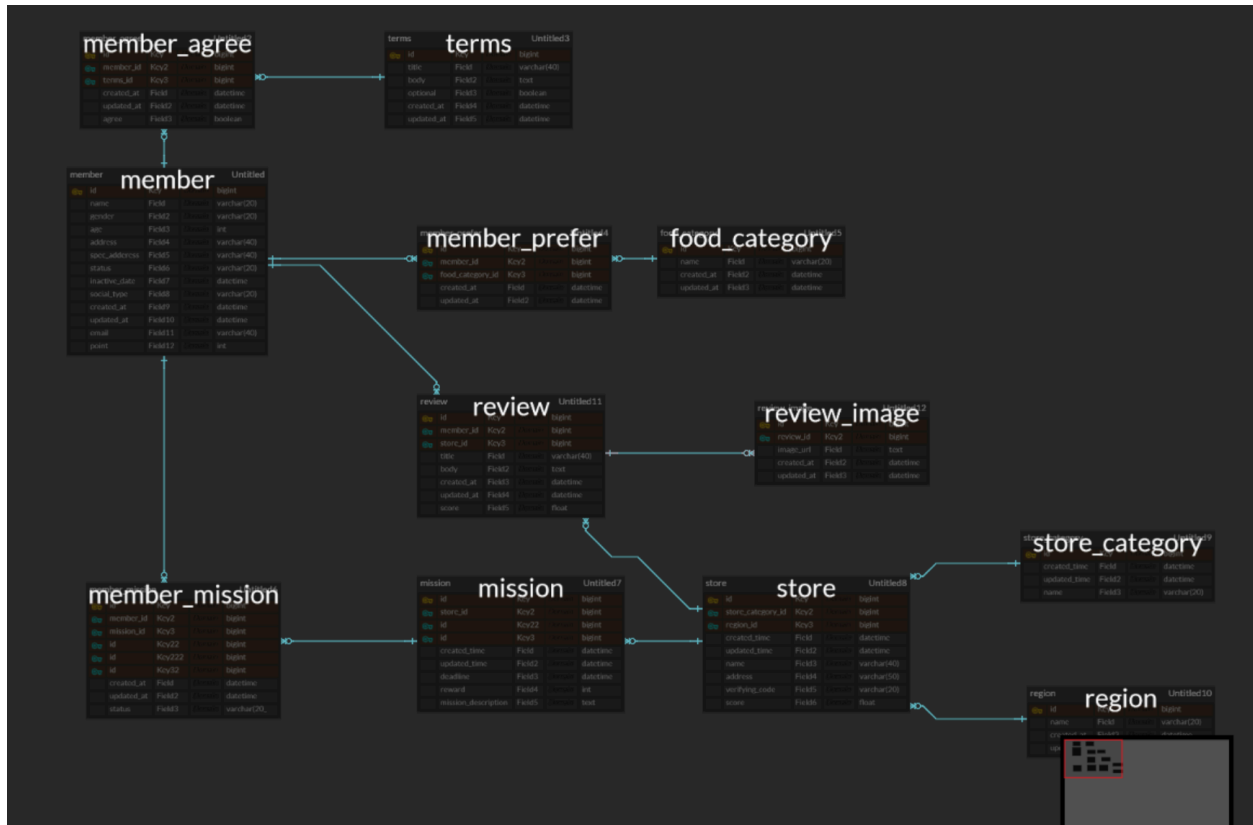


Chapter 2. 실전 SQL 미션



일단 cp1 미션 예시를 보고 ERD를 뜯어고쳤다

<https://www.erdcloud.com/d/uSekSAmPemDLoPBzc>

1.



내가 진행중, 진행 완료한 미션 모아서 보는 쿼리(페이징 포함)

여기 해당되는 db는 다음과 같다

member_mission				Untitled6
	id	Key	Domain	bigint
	member_id	Key2	Domain	bigint
	mission_id	Key3	Domain	bigint
	created_at	Field	Domain	datetime
	updated_at	Field2	Domain	datetime
	status	Field3	Domain	varchar(20_

별도로 엔티티를 구분하지 않고 status로 진행 현황을 구분했으므로 그냥 member_mission에서 '진행중' 미션과 '진행완료' 미션을 한꺼번에 가져오면 될 것 같다. (나의 member_id가 3인 경우를 가정함)

```
SELECT mm.*
FROM member_mission as mm
WHERE mm.member_id = 3
AND mm.status IN ('진행중', '진행완료');
```

여기에 페이징을 한다고 하자. created_at을 기준으로 최신순부터 정렬하여 페이징한다고 하면,

```
SELECT mm.*
FROM member_mission as mm
WHERE mm.member_id = 3
AND mm.status IN ('진행중', '진행완료');
ORDER BY mm.created_at DESC, mm.id DESC //중복값이 존재할까봐 넣어줌
LIMIT 10 OFFSET 10;
```

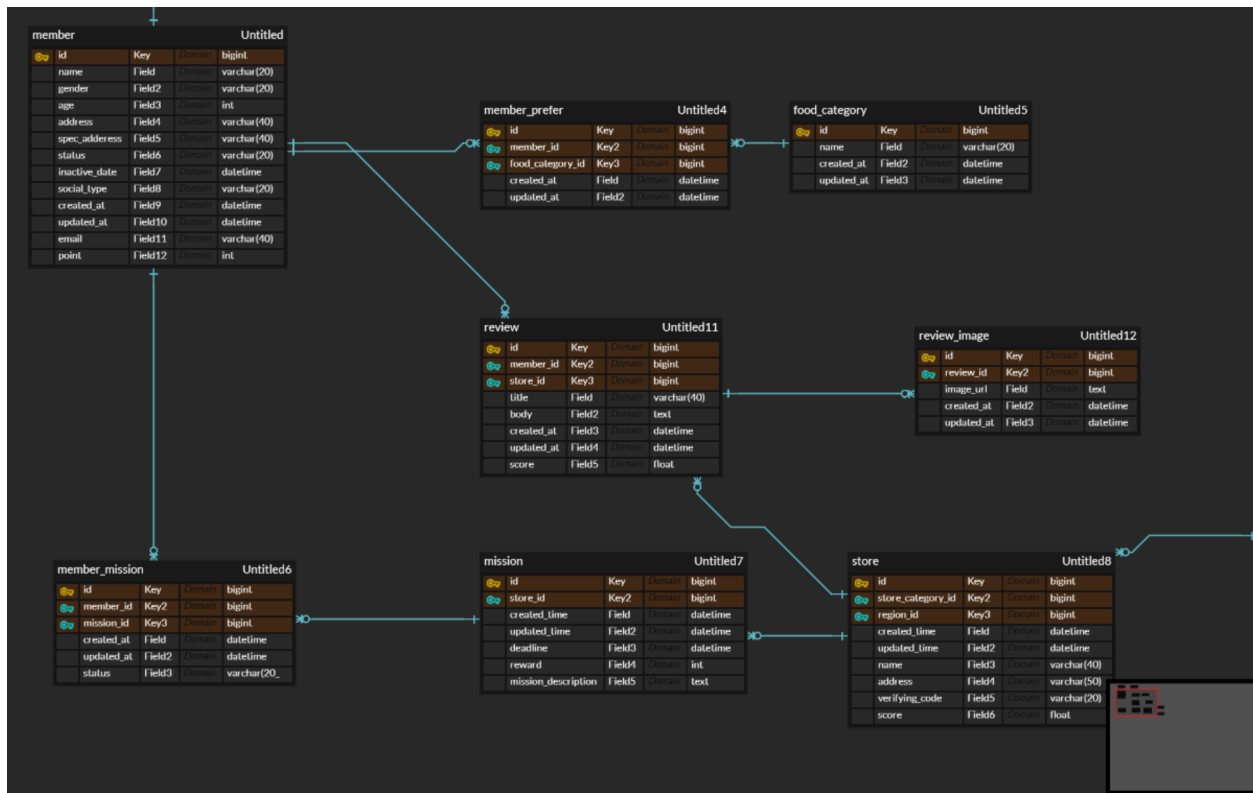
한 번에 10개씩 보여줄 수 있게 페이징하였다

2.



리뷰 작성하는 쿼리,
* 사진의 경우는 일단 배제

여기 해당하는 db는 다음과 같다

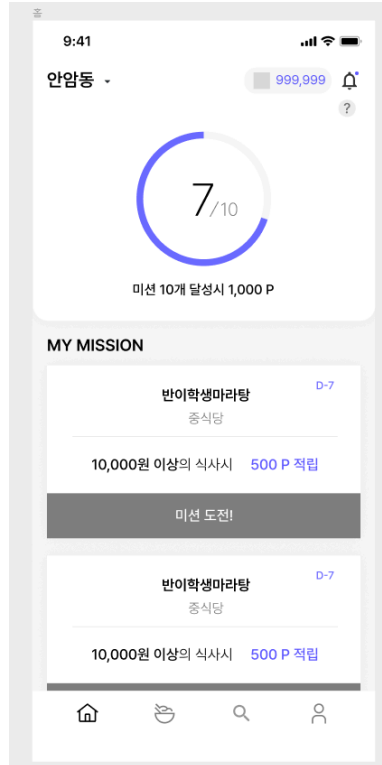


작성의 경우에는 SELECT가 아닐 것 같아서 찾아봤고, INSERT INTO를 쓰면 된다는 것을 알게 되었다

(나의 member_id가 3이고 store_id가 5일 경우)

```
INSERT INTO review (member_id, store_id, title, body, score, created_at, updated_at)
VALUES (3, 5, '제목', '음 너무 맛있어요!', 4.5, NOW(), NOW());
```

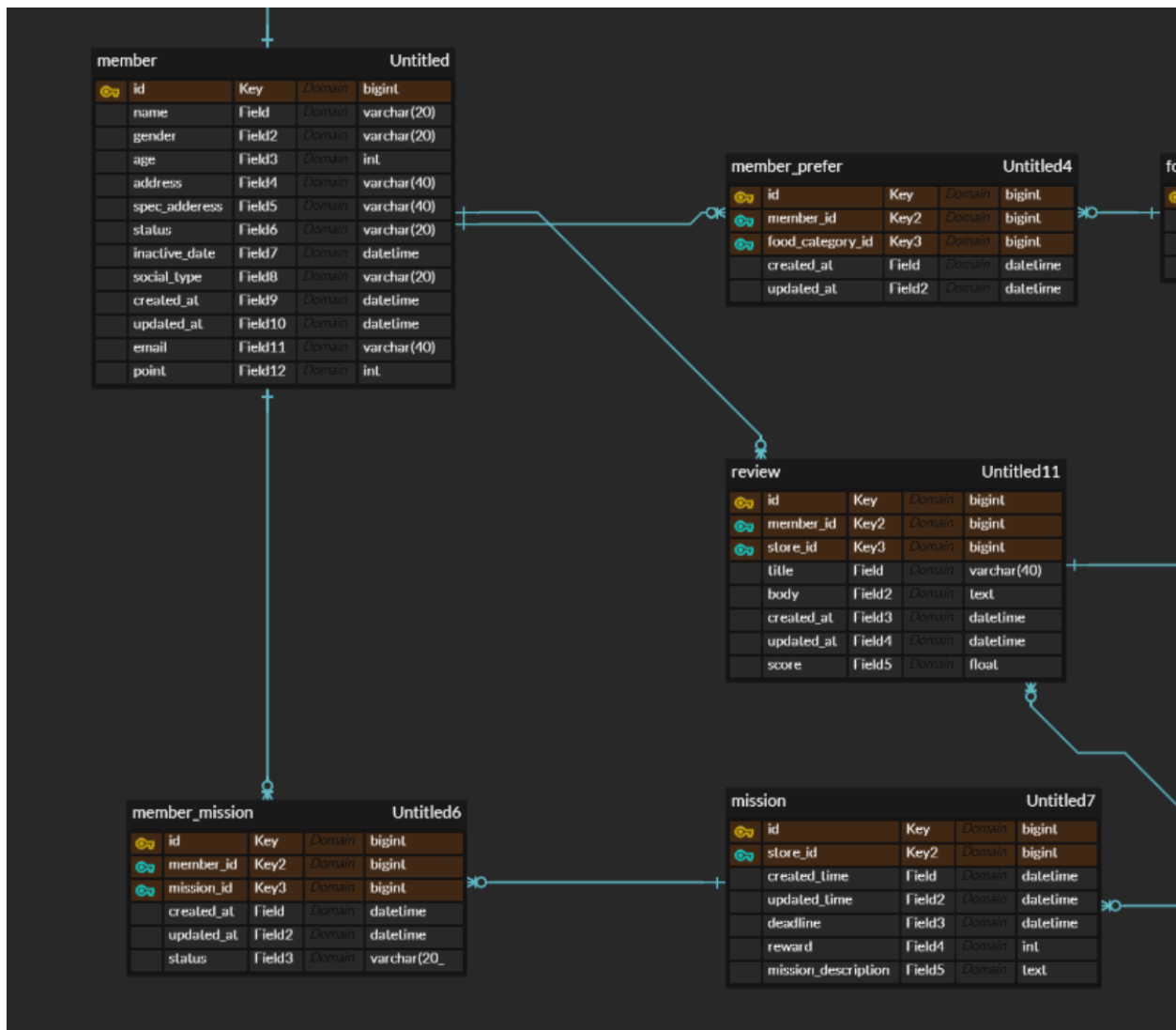
3.

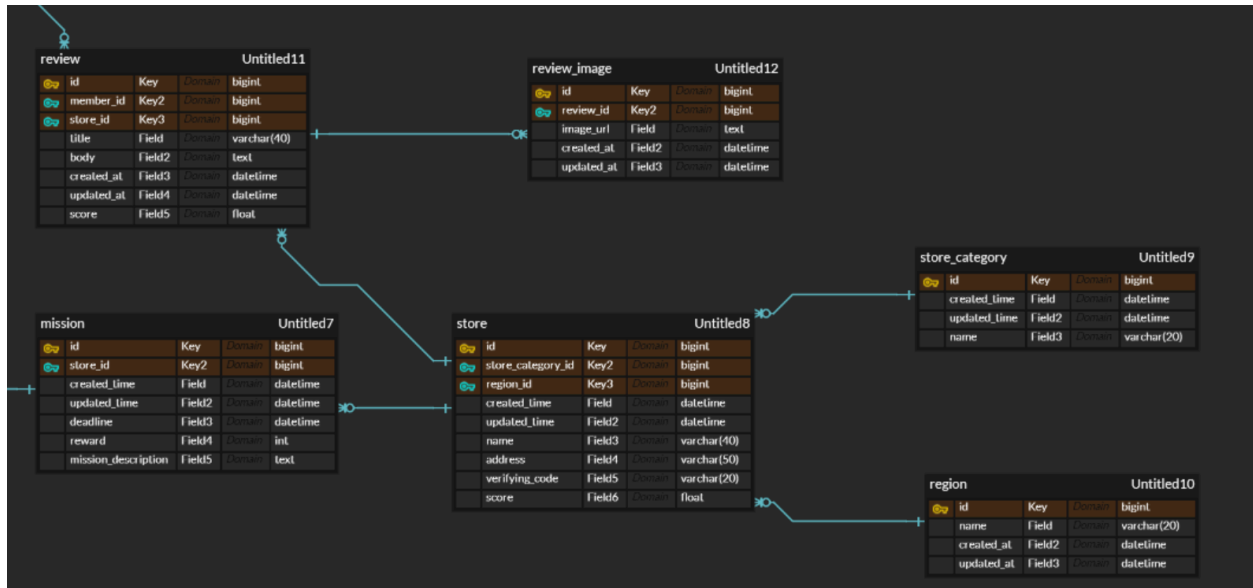


홈 화면 쿼리

(현재 선택 된 지역에서 도전이 가능한 미션 목록, 페이징 포함)

여기 해당되는 db를 가져왔다





내가 선택한 지역이 안암동이라고 했을 때,

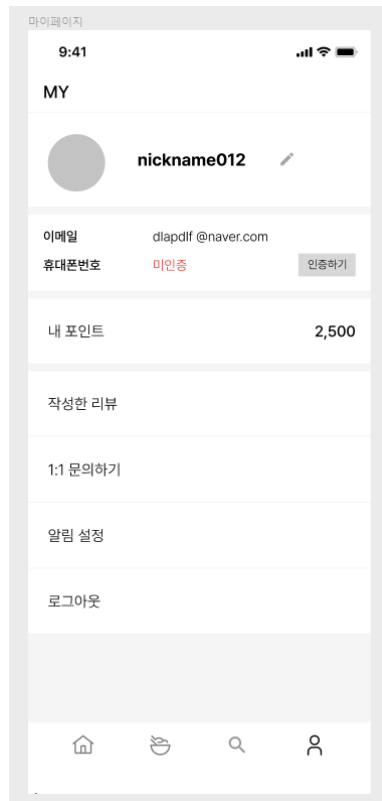
```
SELECT mm.*
FROM member_mission as mm
WHERE mm.mission_id in
  (SELECT id FROM mission WHERE store_id in
    (SELECT id FROM store WHERE region_id in
      (SELECT id FROM region WHERE name = '안암동'))))
```

이런 식으로 서브쿼리를 사용하여 나타낼 수 있다

여기에 페이징을 고려한다면

```
SELECT mm.*
FROM member_mission as mm
WHERE mm.mission_id in
  (SELECT id FROM mission WHERE store_id in
    (SELECT id FROM store WHERE region_id in
      (SELECT id FROM region WHERE name = '안암동'))))
ORDER BY mm.created_at DESC, mm.id DESC
LIMIT 10 OFFSET 0;
```

4.



마이 페이지 화면 쿼리

member					Untitled				
id	Key	Domain	bigint						
name	Field	Domain	varchar(20)						
gender	Field2	Domain	varchar(20)						
age	Field3	Domain	int						
address	Field4	Domain	varchar(40)						
spec_address	Field5	Domain	varchar(40)						
status	Field6	Domain	varchar(20)						
inactive_date	Field7	Domain	datetime						
social_type	Field8	Domain	varchar(20)						
created_at	Field9	Domain	datetime						
updated_at	Field10	Domain	datetime						
email	Field11	Domain	varchar(40)						
point	Field12	Domain	int						

해당하는 엔티티는 위와 같다

```
SELECT m.name, m.email, m.point  
FROM member as m  
WHERE m.id=3
```

이미 point도 엔티티 안에 있어서 쿼리가 복잡하진 않게 나왔다