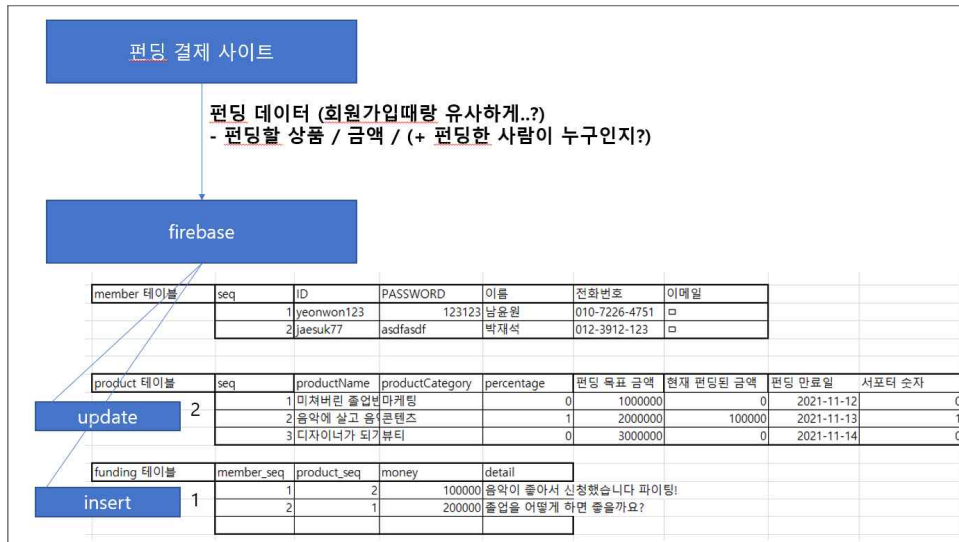
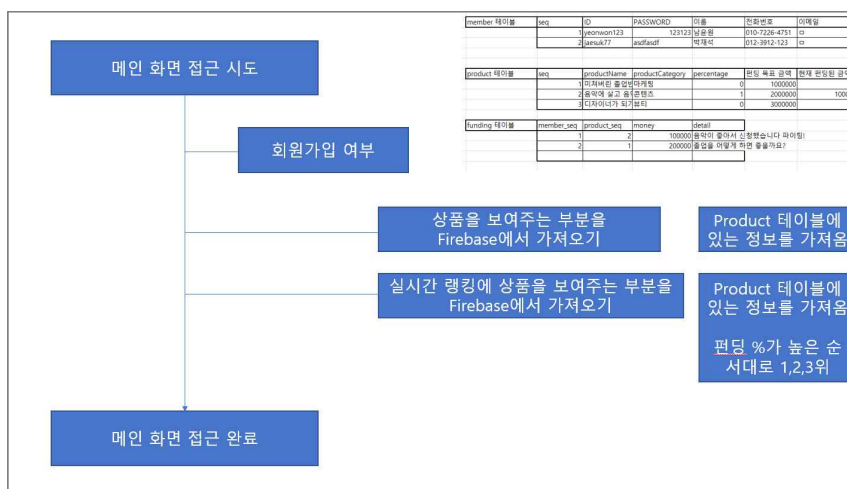


[백엔드를 다루기 위해 설계한 내용]

1. 데이터 저장을 위한 테이블 설계와 플로우 예상



기본적으로 세 개의 테이블이 필요할 것이라고 생각합니다. 각각에 대해 설명하자면, 가입한 사용자의 정보를 저장할 용도인 member테이블, 사이트에 올라온 펀딩글에 대한 정보를 저장할 product테이블, 어떤 유저가 어떤 상품을 펀딩했는지 가격과 함께 정보를 저장하는 funding테이블까지 총 세 개를 설계했습니다. member테이블과 update테이블의 경우 어떤 유저인지와 어떤 펀딩글인지 구분가능해야 하기 때문에 seq number로 구별하려고 했습니다. member는 개인정보를 저장하고 product는 실시간 업데이트와 새로고침시 반영될 데이터들을 담기 위함입니다. funding테이블에 money 칼럼에, 유저가 펀딩한 금액을 insert시키면 그 정보를 바탕으로 특정 펀딩글의 money 합계를 product테이블의 현재 펀딩된 금액에다 update 시키는 방식으로 플로우를 구성했습니다. 이렇게 한다면, 나중에 목표금액 중 몇 퍼센트가 펀딩되었는지도 알 수 있을 것입니다.



위 사진은 메인화면에서 펀딩을 보여주는 방법과 실시간랭킹을 뽑아와서 보여주는것에 대한 기준입니다.

2. 데이터베이스의 생성과 더미데이터 삽입

```
MariaDB [(none)]> CREATE DATABASE LifeFunding default CHARACTER SET UTF8;
Query OK, 1 row affected (0.007 sec)

MariaDB [(none)]> use LifeFunding
Database changed
```

우선 LifeFundging이라는 데이터베이스를 생성한 후, 그 데이터베이스로 이동했습니다.

```
MariaDB [LifeFunding]> show tables;
Empty set (0.003 sec)

MariaDB [LifeFunding]> CREATE TABLE member (
  -> seq INT NOT NULL AUTO_INCREMENT,
  -> ID VARCHAR(20) NOT NULL,
  -> PASSWORD VARCHAR(20) NOT NULL,
  -> name VARCHAR(5) NOT NULL,
  -> tel VARCHAR(15) NOT NULL,
  -> email VARCHAR(30) NOT NULL,
  -> PRIMARY KEY (seq)
  -> );
Query OK, 0 rows affected (0.066 sec)
```

테이블 목록이 비어있는 것을 확인하고, member테이블 먼저 만들어주었습니다.

```
MariaDB [LifeFunding]> desc member;
```

Field	Type	Null	Key	Default	Extra
seq	int(11)	NO	PRI	NULL	auto_increment
ID	varchar(20)	NO		NULL	
PASSWORD	varchar(20)	NO		NULL	
name	varchar(5)	NO		NULL	
tel	varchar(15)	NO		NULL	
email	varchar(30)	NO		NULL	

6 rows in set (0.024 sec)

테이블 자세히 보기로 이상이 없는 것을 확인한 후 나머지 테이블도 생성합니다.

```
MariaDB [LifeFunding]> CREATE TABLE product (
  -> seq INT NOT NULL AUTO_INCREMENT,
  -> productName VARCHAR(25) NOT NULL,
  -> productCategory VARCHAR(10) NOT NULL,
  -> percentage INT,
  -> goalMoney INT NOT NULL,
  -> nowMoney INT,
  -> endDate DATETIME DEFAULT NOW(),
  -> supporter INT,
  -> PRIMARY KEY (seq)
  -> );
Query OK, 0 rows affected (0.027 sec)
```

```
MariaDB [LifeFunding]> CREATE TABLE funding (
  -> member_seq INT NOT NULL,
  -> product_seq INT NOT NULL,
  -> money INT,
  -> detail VARCHAR(30)
  -> );
Query OK, 0 rows affected (0.026 sec)
```

```

MariaDB [LifeFunding]> INSERT into member (ID, PASSWORD, name, tel, email) values ("
aaa1111", "aaapw!", "user1", "010-1111-1111", "aa@naver.com");
Query OK, 1 row affected (0.064 sec)

MariaDB [LifeFunding]> INSERT into member (ID, PASSWORD, name, tel, email) values ("
bbb1111", "bbbpw!", "user2", "010-2222-2222", "bb@naver.com");
Query OK, 1 row affected (0.002 sec)

MariaDB [LifeFunding]> SELECT * FROM member;
+----+-----+-----+-----+-----+-----+
| seq | ID      | PASSWORD | name  | tel      | email      |
+----+-----+-----+-----+-----+-----+
| 1   | aaa1111 | aaapw!   | user1 | 010-1111-1111 | aa@naver.com |
| 2   | bbb1111 | bbbpw!   | user2 | 010-2222-2222 | bb@naver.com |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.002 sec)

```

우선, member테이블에 더미데이터를 넣어주었습니다. 이후 product테이블에도 더미데이터를 추가하여 이 두 가지 테이블을 가지고 funding테이블에 데이터가 제대로 들어가는지 확인해 볼 예정입니다.

```

MariaDB [LifeFunding]> INSERT into product (productName, productCategory, percentag
e, goalMoney, nowMoney, endDate, supporter) values ("음악에 살고 음악에 죽는다", "
콘텐츠", 0, 2000000, 0, "2021-12-12", 0);
Query OK, 1 row affected (0.004 sec)

MariaDB [LifeFunding]> INSERT into product (productName, productCategory, percentag
e, goalMoney, nowMoney, endDate, supporter) values ("디자이너가 되기까지", "뷰티",
0, 1500000, 0, "2021-12-25", 0);
Query OK, 1 row affected (0.002 sec)

```

```

MariaDB [LifeFunding]> SELECT * FROM product;
+----+-----+-----+-----+-----+-----+-----+-----+
| seq | productName          | productCategory | percentage | goalMoney | nowMoney | endDate          | supporter |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1   | 미쳐버린 졸업반      | 마케팅          | 0          | 1000000   | 0        | 2021-12-14 00:00:00 | 0        |
| 2   | 음악에 살고 음악에 죽는다 | 콘텐츠          | 0          | 2000000   | 0        | 2021-12-12 00:00:00 | 0        |
| 3   | 디자이너가 되기까지   | 뷰티            | 0          | 1500000   | 0        | 2021-12-25 00:00:00 | 0        |
+----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.002 sec)

```

product테이블에도 더미데이터를 성공적으로 넣었음을 확인할 수 있습니다.

3. 한계점 파악

위 상황까지 진행하는데는 별다른 어려움이 없었습니다. 하지만 시간부족과 백엔드에 대한 경험 부족으로 인해 프론트와 백을 제대로 연결하지 못했으며 이 부분은 추후 보완할 예정입니다. 서버를 이용하여 편딩글을 작성할 때 데이터를 가져와 product 테이블에 저장하고 데이터베이스의 정보를 뽑아서 홈 화면에 보여주고 사용자가 편딩을 진행하면 실시간으로 정보를 업데이트 하는 등의 과정은 백엔드에 대한 지식을 쌓고 바로 작업에 들어갈 예정입니다.

