이보경 2주차 과제

1. DB 설계

카테고리

PK	카테고리 아이디	varchar(10)
	카테고리명	varchar(20)

추천

FK	고객아이디	varchar(10)
FK	상품아이디	varchar(10)
	상품추천유형코드	varchar(10)

고객

PK	이름	varchar(30)
	고객아이디	varchar(10)
	성별	varchar(5)
	출생연도	year
	가입경로구분코드	varchar(20)
	휴대폰번호	varchar(23)

상품

PK	상품아이디	varchar(10)
	상품명	varchar(30)
	가격	int
	상품상세내용	longtext
	상품이미지(blob)	blob
FK	카테고리 아이디	varchar(10)

장바구니

PK	장바구니 아이디	varchar(20)
FK	고객아이디	varchar(10)
FK	상품아이디	varchar(10)
	수량	int

구매

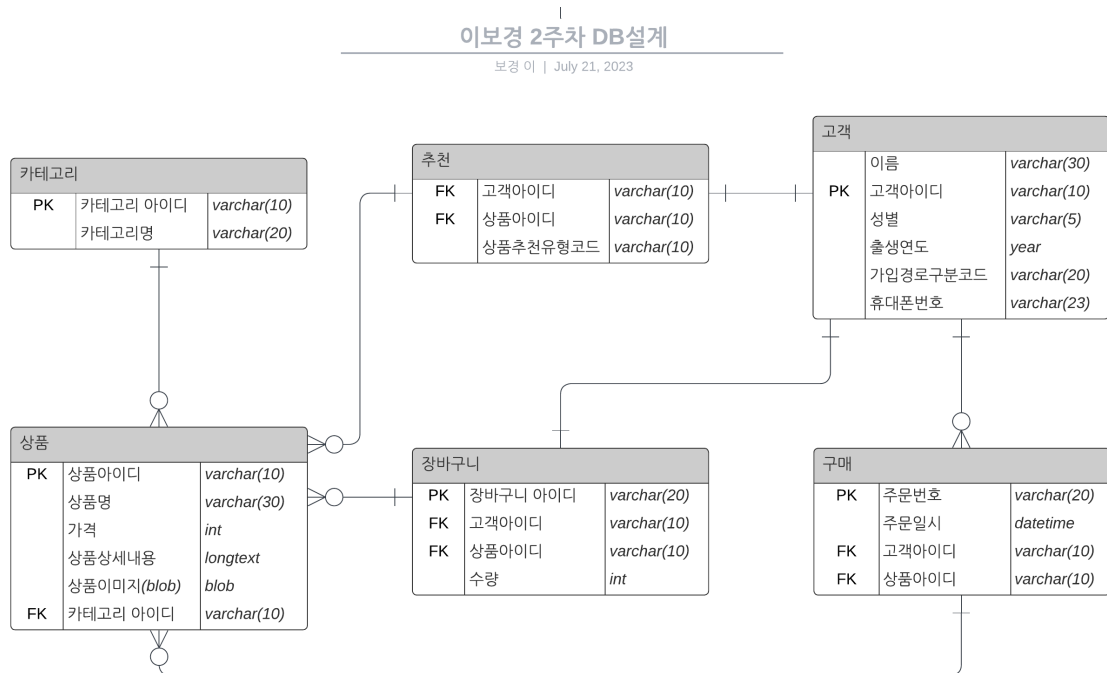| PK	주문번호	varchar(20)
	주문일시	datetime
FK	고객아이디	varchar(10)
FK	상품아이디	varchar(10)

2. 데이터 업로드

1) category

```
mysql> create table category (
    -> cate_id   varchar(10) PRIMARY KEY,
    -> cate_name varchar(30) NOT NULL
    -> );
Query OK, 0 rows affected (0.19 sec)

mysql> select * from category;
+---------+-----------+
| cate_id | cate_name |
+---------+-----------+
| ct01    | top       |
| ct02    | neat      |
| ct03    | pants     |
| ct04    | skirt     |
| ct05    | dress     |
| ct06    | sports    |
| ct07    | outer     |
+---------+-----------+
7 rows in set (0.01 sec)
```

2) product

```
mysql> create table product (
    -> pd_id    varchar(10) PRIMARY KEY,
    -> pd_name varchar(30) NOT NULL,
    -> pd_price int NOT NULL,
    -> pd_detail longtext,
    -> pd_image  blob NOT NULL,
    -> cate_id   varchar(10),
    -> FOREIGN KEY(cate_id) REFERENCES category(cate_id) ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.11 sec)
mysql> select * from product;
+-------+---------+----------+---------------+-------------------+---------+
| pd_id | pd_name | pd_price | pd_detail     | pd_image          | cate_id |
+-------+---------+----------+---------------+-------------------+---------+
| pd01  | sh1     |    10000 | pretty t-shirt | 0x               | ct01    |
| pd02  | sk1     |    30000 | cargo skirt   | 0x                | ct04    |
| pd03  | dr1     |    59000 | floral dress  | 0x                | ct05    |
| pd04  | tr1     |    30000 | training set  | 0x                | ct06    |
+-------+---------+----------+---------------+-------------------+---------+
4 rows in set (0.00 sec)
```

3) customer

```
mysql> create table customer (
    -> cs_id     varchar(10) PRIMARY KEY,
    -> cs_name  varchar(30) NOT NULL,
    -> gender   varchar(5),
    -> birth    year,
    -> cs_path  varchar(20) NOT NULL,
    -> cs_phone varchar(23) NOT NULL
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> select * from customer;
+-------+---------+--------+-------+---------+---------------+
| cs_id | cs_name | gender | birth | cs_path | cs_phone      |
+-------+---------+--------+-------+---------+---------------+
| cs01  | anna    | F      | 2001  | p01     | 010-0028-3928 |
| cs02  | jane    | M      | 1992  | p02     | 010-3492-3119 |
| cs03  | bell    | F      | 1998  | p01     | 010-0038-9899 |
+-------+---------+--------+-------+---------+---------------+
3 rows in set (0.00 sec)
```

4) recommend

```
mysql> create table recommend (
    -> cs_id  varchar(10) NOT NULL,
    -> pd_id  varchar(10) NOT NULL,
    -> rc_code varchar(10) NOT NULL,
    -> FOREIGN KEY(cs_id) REFERENCES customer(cs_id) ON UPDATE CASCADE,
    -> FOREIGN KEY(pd_id) REFERENCES product(pd_id) ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> select * from recommend;
+-------+-------+---------+
| cs_id | pd_id | rc_code |
+-------+-------+---------+
| cs01  | pd01  | rc01    |
| cs02  | pd03  | rc02    |
| cs03  | pd02  | rc03    |
+-------+-------+---------+
3 rows in set (0.00 sec)
```

5) purchase

```
mysql>  create table purchase (
    -> pc_id   varchar(20) PRIMARY KEY,
    -> pc_date datetime NOT NULL,
    -> cs_id   varchar(10) NOT NULL,
    -> pd_id   varchar(10) NOT NULL,
    -> FOREIGN KEY(cs_id) REFERENCES customer(cs_id) ON UPDATE CASCADE,
    -> FOREIGN KEY(pd_id) REFERENCES product(pd_id) ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> select * from purchase;
+-------+---------------------+-------+-------+
| pc_id | pc_date             | cs_id | pd_id |
+-------+---------------------+-------+-------+
| pc01  | 2023-07-21 06:39:24 | cs01  | pd02  |
| pc02  | 2023-07-21 06:39:24 | cs02  | pd01  |
| pc03  | 2023-07-21 06:39:24 | cs03  | pd03  |
+-------+---------------------+-------+-------+
3 rows in set (0.00 sec)
```

6) cart

```
mysql> create table cart (
    -> cart_id varchar(20) PRIMARY KEY,
    -> cs_id   varchar(10) NOT NULL,
    -> pd_id   varchar(10) NOT NULL,
    -> amount  int,
    -> FOREIGN KEY(cs_id) REFERENCES customer(cs_id) ON UPDATE CASCADE,
    -> FOREIGN KEY(pd_id) REFERENCES product(pd_id) ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> select * from cart;
+---------+-------+-------+--------+
| cart_id | cs_id | pd_id | amount |
+---------+-------+-------+--------+
| ct01    | cs01  | pd01  |      1 |
| ct02    | cs02  | pd02  |      3 |
| ct03    | cs03  | pd04  |      2 |
+---------+-------+-------+--------+
3 rows in set (0.01 sec)
```

질문) 한 사람이 한 번에 여러 상품을 주문한 경우에도 상품 한 개씩 따로따로 입력해야 하나요?
예를 들어, 장바구니의 경우 장바구니 번호를 pk로 설치해서 따로 입력하면 한 고객에게
장바구니가 여러개 생기게 되는데 이걸 해결할 수 있는 방법이 있는지 궁금합니다!

3. MongoDB 설치

```
bokyung@bokyungui-MacBookAir ~ brew install mongodb-community
==> Fetching dependencies for mongodb/brew/mongodb-community: mongodb/brew/mongodb-database-tools,
 c-ares, libnghttp2, libuv, node@16 and mongosh
==> Fetching mongodb/brew/mongodb-database-tools
==> Downloading https://fastdl.mongodb.org/tools/db/mongodb-database-tools-macos-arm64-100.7.3.zip
######################################################################################## 100.0%
==> Fetching c-ares
bokyung@bokyungui-MacBookAir ~ brew services start mongodb-community
==> Successfully started `mongodb-community` (label: homebrew.mxcl.mongodb-community)
bokyung@bokyungui-MacBookAir ~ mongosh
Current Mongosh Log ID: 64ba2953c32fef192bc34936
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=
2000&appName=mongosh+1.10.1
Using MongoDB:          6.0.6
Using Mongosh:          1.10.1
```

```
mongodb> use mongodb
already on db mongodb
mongodb> show dbs;
admin   40.00 KiB
config  12.00 KiB
local   40.00 KiB
mongodb> db.stats()
{
  db: 'mongodb',
  collections: 0,
  views: 0,
  objects: 0,
  avgObjSize: 0,
  dataSize: 0,
  storageSize: 0,
  indexes: 0,
  indexSize: 0,
  totalSize: 0,
  scaleFactor: 1,
  fsUsedSize: 0,
  fsTotalSize: 0,
  ok: 1
}
mongodb> show collections;
```

1) collection 생성, 삭제

```
mongodb> db.createCollection('coct2');
{ ok: 1 }
mongodb> db.coct2.isCapped();
false
mongodb> db.coct2.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount
.
0
mongodb> db.coct2.drop();
true
mongodb> db.createCollection('people');
{ ok: 1 }
mongodb> db.people.updateMany({ }, { $set: { join_date: new Date() } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

2) insert

```
mongodb> db.people.insertMany(
... [
... {"SN":"sn01", "no":1},
... {"SN":"sn02", "no":2},
... {"SN":"sn03", "no":3, "reg_dttm":new Date()} ]
... );
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64ba2a87c32fef192bc34937"),
    '1': ObjectId("64ba2a87c32fef192bc34938"),
    '2': ObjectId("64ba2a87c32fef192bc34939")
  }
}
```

3) find

```
mongodb> db.people.find();
[
  { _id: ObjectId("64ba2a87c32fef192bc34937"), SN: 'sn01', no: 1 },
  { _id: ObjectId("64ba2a87c32fef192bc34938"), SN: 'sn02', no: 2 },
  {
    _id: ObjectId("64ba2a87c32fef192bc34939"),
    SN: 'sn03',
    no: 3,
    reg_dttm: ISODate("2023-07-21T06:49:43.449Z")
  }
]
```