

# E-Commerce Information System

## 1. Project Description

Our team planned to implement an E-Commerce database application that allows users to browse products and place orders originally. However, after experimenting with Streamlit, we found that Streamlit has so many restrictions that prevent us from achieving our initial goal. Therefore, we changed our initial project proposal, and decided to implement an E-Commerce Information System that allows the system managers or analysts to interact with our database, query different tables, and obtain meaningful insights based on the information in our database.

Our database is designed to record basic product information, rating, review text and more for each product in the market. With all that product information given by customers, we can easily analyze the most successful products in different categories and discover insights into consumer reviews. If applying with machine learning models, we can also use these data to generate detailed analysis of any product from both micro and macro perspectives. E.g.: What are the mostly reviewed products? We can also map the keywords in the review text against the review rating to gain an overall sentiment analysis.

## 2. Business Rules

The entity set includes: [Products, Customers, Reviews, Orders, Manufacturers, Categories, Shipments]. The relationship set includes: [Write, Ship, Contain, Review, Make, Belong]. And our business rules are listed as follow:

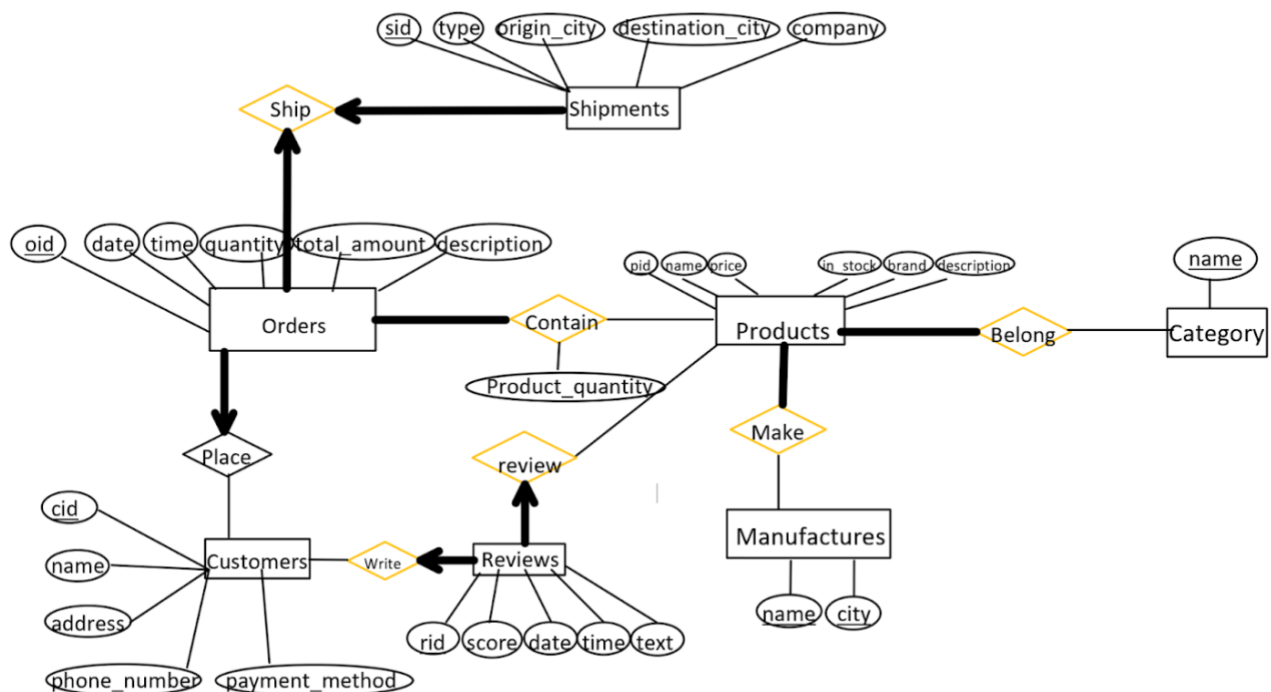
- Each review has an id(rid) that uniquely identifies them, and has a score, a date, a time and a text. Each review describes exactly one product and is written by exactly one customer; A customer can write any number of reviews.
- Each customer has an id(cid) that uniquely identifies them, and has a name, an address, a phone number and a payment method. Each customer can place any number of orders.
- Each order has an id(oid) that uniquely identifies them, and has a date, a time, a quantity number, a total amount spent and a description. Each order is placed by exactly one customer and has exactly one shipment information. Each order contains at least one product.

Our Project is deployed at port number 8619

By Yibei Huang(yh3655), Xixuan Wang(xw2579)

- Each product is uniquely identified by an id(pid) and has a name, a price, an in-stock number, a brand and a description while not all products have a description. Each product is made by at least one manufacturer and belongs to at least one category.
- Each shipment has an id(sid) that uniquely identifies them, and has a type, an origin, a destination and a company (service provider). Each shipment corresponds to exactly one order.
- Each manufacturer has a tuple of name and city that uniquely identifies them.
- Each category has a name that uniquely identifies them.

### 3. ER Model



Our Project is deployed at port number 8619

By Yibei Huang(yh3655), Xixuan Wang(xw2579)

## 4. Relational Schema

Note: The participation constraints of relationship sets “Contain”, “Make” and “Belong” are not able to be represented in the following sql statements.

```
drop table if exists Reviews_Reviewed_Written cascade;
drop table if exists O_Contain_P cascade;
drop table if exists Orders_Placed_Shipped cascade;
drop table if exists P_BelongTo_C cascade;
drop table if exists M_Make_P cascade;
drop table if exists Customers;
drop table if exists Products;
drop table if exists Shipments;
drop table if exists Manufactures;
drop table if exists Categories;
```

```
create table Customers(
    cid integer primary key,
    name varchar(128),
    address varchar(256),
    phone_number varchar(128),
    payment_method varchar(128)
);
```

```
create table Products(
    pid integer primary key,
    name varchar(128),
    price decimal,
    in_stock integer,
    brand varchar(128),
    description varchar(512)
);
```

```
create table Reviews_Reviewed_Written (
    rid integer primary key,
    score integer,
    review_date date,
    review_time time,
    review_text varchar(512),
    pid integer not null,
    cid integer not null,
    foreign key (pid) references Products(pid),
    foreign key (cid) references Customers(cid)
);
```

```
create table Shipments(
    sid integer primary key,
    type varchar(128),
    origin_city varchar(128),
    destination_city varchar(128),
    company varchar(128)
);
```

Our Project is deployed at port number 8619

By Yibei Huang(yh3655), Xixuan Wang(xw2579)

```
create table Orders_Placed_Shipped(  
    order_id integer primary key,  
    order_date date,  
    order_time time,  
    quantity integer,  
    total_amount decimal,  
    description varchar(512),  
    cid integer not null,  
    sid integer not null unique,  
    foreign key (cid) references Customers(cid),  
    foreign key (sid) references Shipments(sid)  
);
```

```
create table Manufactures(  
  
    name varchar(128),  
    city varchar(128),  
    primary key (name, city)  
);
```

```
create table M_Make_P(  
    pid integer,  
    manufacture_name varchar(128),  
    manufacture_city varchar(128),  
    primary key (pid, manufacture_name, manufacture_city),  
    foreign key (pid) references Products(pid),  
    foreign key (manufacture_name, manufacture_city) references Manufactures(name, city)  
);
```

```
create table Categories(  
    name varchar(128) primary key  
);
```

```
create table O_Contain_P(  
    order_id integer,  
    pid integer,  
    product_quantity integer,  
    primary key (order_id, pid),  
    foreign key (order_id) references Orders_Placed_Shipped(order_id),  
    foreign key (pid) references Products(pid)  
);
```

```
create table P_BelongTo_C(  
    pid integer,  
    category varchar(128),  
    primary key (pid, category),  
    foreign key (pid) references Products(pid),  
    foreign key (category) references Categories(name)  
);
```

Our Project is deployed at port number 8619

By Yibei Huang(yh3655), Xixuan Wang(xw2579)

```
create table Reviews_Reviewed_Written (  
    rid integer primary key,  
    score integer,  
    review_date date,  
    review_time time,  
    review_text varchar(512),  
    pid integer not null,  
    cid integer not null,  
    foreign key (pid) references Products(pid),  
    foreign key (cid) references Customers(cid)  
);  
  
create table P_BelongTo_C(  
    pid integer,  
    category varchar(128),  
    primary key (pid, category),  
    foreign key (pid) references Products(pid),  
    foreign key (category) references Categories(name)  
);  
  
create table Products(  
    pid integer primary key,  
    name varchar(128),  
    price decimal,  
    in_stock integer,  
    brand varchar(128),  
    description varchar(512)  
);
```

## 5. Data Loading Procedure

There are two parts of our project's data loading procedure.

First, after entering the PostgreSQL database system, run the file "schema.sql" by typing "\i schema.sql". This is used to create all the tables in the database of our project. However, these tables are empty tables with no data inside, so we need to load our actual data into them.

Second, quit the PostgreSQL database system and you should be at the command line window. Now, copy and paste the following commands:

```
cat project/data/Customers.csv | psql -U yh3655 -d yh3655_db -c "COPY Customers from STDIN  
CSV HEADER"
```

```
cat project/data/Shipment.csv | psql -U yh3655 -d yh3655_db -c "COPY Shipments from STDIN  
CSV HEADER"
```

Our Project is deployed at port number 8619

By Yibei Huang(yh3655), Xixuan Wang(xw2579)

```
cat project/data/Orders_Placed_Shipped.csv | psql -U yh3655 -d yh3655_db -c "COPY  
Orders_Placed_Shipped from STDIN CSV HEADER"
```

```
cat project/data/Products.csv | psql -U yh3655 -d yh3655_db -c "COPY Products from STDIN CSV  
HEADER"
```

```
cat project/data/Reviews_Reviewed_Written.csv | psql -U yh3655 -d yh3655_db -c "COPY  
Reviews_Reviewed_Written from STDIN CSV HEADER"
```

```
cat project/data/Categories.csv | psql -U yh3655 -d yh3655_db -c "COPY Categories from STDIN  
CSV HEADER"
```

```
cat project/data/Manufactures.csv | psql -U yh3655 -d yh3655_db -c "COPY Manufactures from  
STDIN CSV HEADER"
```

```
cat project/data/M_Make_P.csv | psql -U yh3655 -d yh3655_db -c "COPY M_Make_P from  
STDIN CSV HEADER"
```

```
cat project/data/O_Contain_P.csv | psql -U yh3655 -d yh3655_db -c "COPY O_Contain_P from  
STDIN CSV HEADER"
```

```
cat project/data/P_BelongTo_C.csv | psql -U yh3655 -d yh3655_db -c "COPY P_BelongTo_C  
from STDIN CSV HEADER"
```

```
cd project
```

```
streamlit run project.py --server.address=localhost --server.port=8619
```

These commands should load all the data into those pre-created tables in our database.

Note that you can also run our project by changing the net id to be “xw2579” and changing the port number to be “8682” because both of the port numbers work for our application.