Group 11: Congo
CS 6750 Database Systems
Final Project Document.

Project Idea / User Requirements

A. What kind of business are we modeling?

We are trying to model an online retailing business, like Amazon, with multiple customers and types of items to purchase.

B. Possible Users?

Possible users can include people who are trying to sell items that they may own, and people who are trying to buy those corresponding items.

C. Why did you choose this application / why is it important?

This application is important because it's a common and necessary application of databases. This marketplace implementation of a database was most interesting to us, and we wanted to learn more about how this type of database would be structured.

D. Main questions and transactions (insert, delete, update, find) are we trying to do through database?

We are trying to add merchandise for others to purchase (insert), allow users to purchase new merchandise (delete, update), allow users to search for items (find), as well as add new customers and allow customers to modify their information on file (insert, update). Furthermore, we want to keep a record of all transactions made.

Possible transactions that might be performed:

- Add customers/sellers accounts to the site
- Change customer info/information
- Remove items that are out of stock
- Add shopping cart/wishlist
- Update visible stock numbers
- Add order history
- Keep a profile of each customer for a customized experience/recommendations

(continued on next page)

Database Tables

```
-- CREATE TABLE CUSTOMER(
-- c_id int primary key,
-- c_name varchar(100),
-- c_dob varchar(100),
-- c_addr varchar(100),
-- cc_number int,
-- foreign key (cc_number) references CREDITCARD (cc_number)
-- );

-- CREATE TABLE MERCHANDISE(
-- merchandise_id int primary key,
-- m_name varchar(100),
-- m_price int,
-- m_desc varchar(100),
-- m_rating varchar(200)
-- );

-- CREATE TABLE ORDERS(
-- order_id int primary key,
-- order_status varchar(30),
```

We also have:

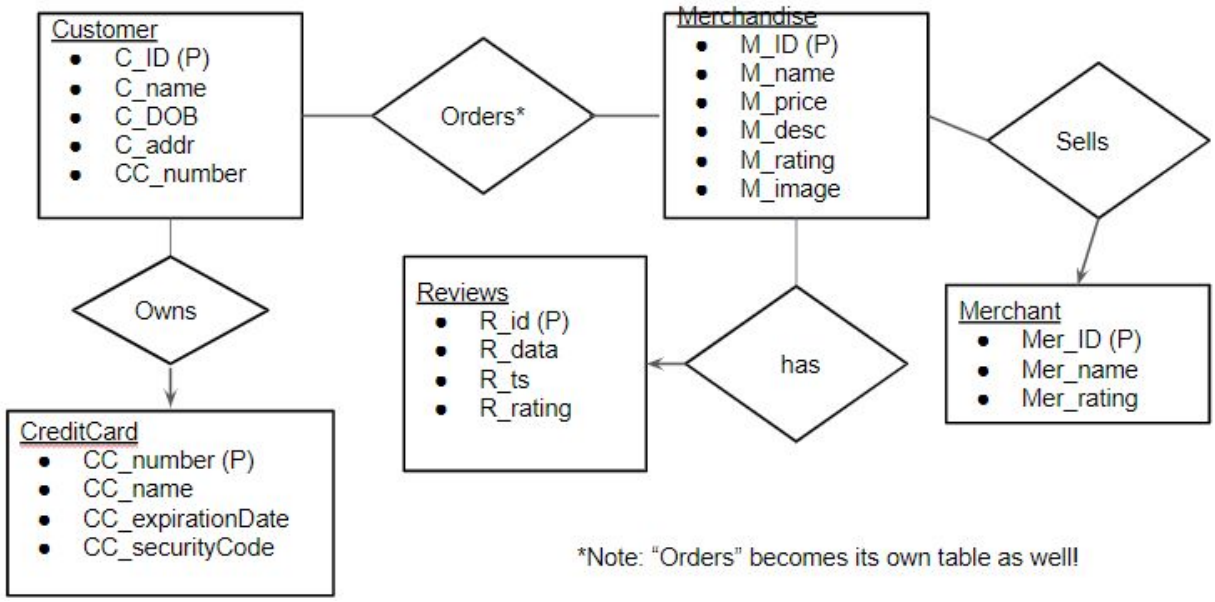
CREATE TABLE MERCHANTS(
M_id int,
M_name varchar(100),
M_rating int
);

CREATE TABLE CREDITCARD(
Cc_number int,
Cc_name varchar(100),
cc_expirationDate date,
cc_securityCode int
);

CREATE TABLE REVIEWS(
R_id int,
R_data varchar(200),
R_ts int,
R_rating int
);

Data Sources
We generated our data, based on items actually sold on Amazon. Obviously, to protect PII, we created entirely fake buyers and sellers random generators with a mix of information based on our group members.

ERD



SQL Queries / Sample Results (results are implied by description of query execution)
1. Query gets non-duplicate names of items which have been ordered in the past. Uses group by and natural join.
select m_name from MERCHANDISE natural join ORDERS group by merchandise_id;

2. Dummy query: selecting all values from the merchandise database
SELECT * FROM MERCHANDISE;

3. Query to retrieve the average merchandise rating posted by a merchant, organized by merchant name
SELECT merchant_name, avg(m_rating) from MERCHANDISE natural join MERCHANT group by merchant_name;

4. Query to fetch all customers who are 21 or older in this year
SELECT * FROM CUSTOMER WHERE 2019 - CONVERT(SUBSTRING(c_dob,7,4),UNSIGNED INTEGER) >= 21;

5. Query to fetch all published reviews with a score higher than 2
SELECT R_data from REVIEWS where R_rating > 2;

6. Query to obtain the name and average merchandise rating from reviews for all merchandise with reviews with an average > 4 stars.
select avg(R_Rating), m_name from REVIEWS natural join MERCHANDISE group by merchandise_id HAVING avg(R_Rating) > 4;

7. Selects the customer name and product name of all orders that have the order status of delivered
SELECT c_name, m_name from CUSTOMER natural join ORDERS natural join MERCHANDISE where order_status = "delivered";

8. Selects all customers and checks to see which ones have placed an order and which ones have not
SELECT c_name, order_status from CUSTOMER left join ORDERS on CUSTOMER.c_id = ORDERS.c_id;

9. Selects all merchandise names which are currently sold by a vendor.
select m_name from MERCHANDISE where merchandise_id in (select merchandise_id from MERCHANT natural join SELLS);

10. Update customer's name to Jake Smith for any user with customer ID = 1
UPDATE CUSTOMER SET c_name = "Jake Smith" where c_id = 1;

Tools for interface implementation / Application Interface

- Django (for interface only, mySQL was used for backend entirely and accessed through a MySQLDB connector in python)
- MySQL DB for all databases / querying
- Amazon AWS for hosting database connection

- Github (for version control) located: https://github.com/sk8wt/Congo/tree/master/congo
- Bootstrap for CSS help
- Ajax/jquery for form submission