20S1 CZ2007 Project: Lab 1 Submission

Team Members:

1. U1922940J - Le Quang Anh (Leader)
2. U1922557L - Chew Harris Rezal
3. U1922222E - Max Pang Liang Hui
4. U1922050A - Neo Yong Tai
5. U1922981C - Ong Jing Hong Elliott
6. U1923230B - Suhana Gupta

Structure of the submission:

We started by going through each requirement and sketching what our potential solution could be. However, each new bullet point added issues which meant that we had to repeatedly revise our sketch as well as request help from the lab instructor. Despite the issues, we were able to discuss and come up with solutions to the requirements.

The assumptions, discussions and choices that we have made in the ER diagram are presented on the next page while the completed ER diagram is on page 3. Note that for the explanations, the requirements are numbered according to the order that the bullet points in the lab manual.

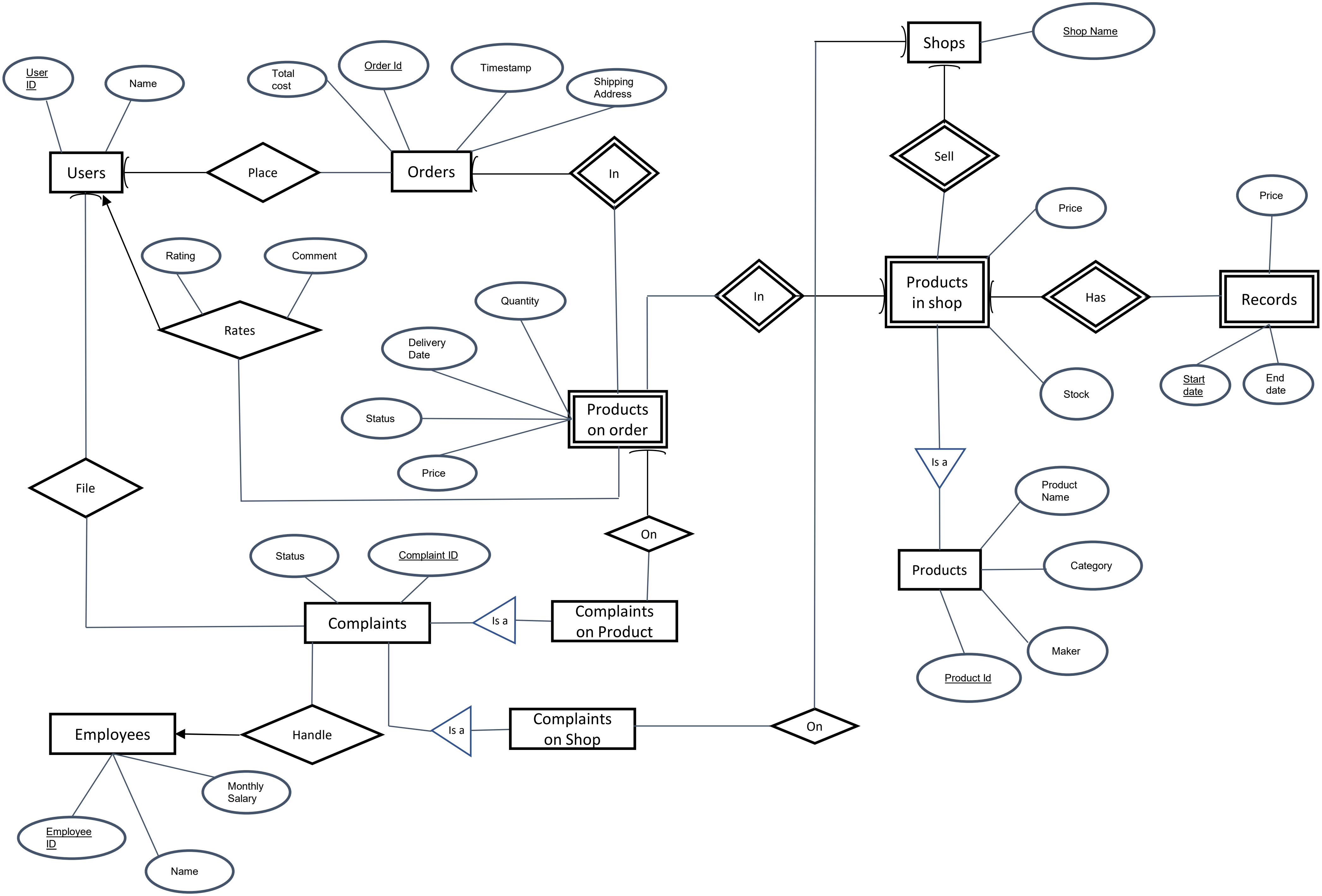
Explanation

From requirement 1, a preliminary idea is to have two entity sets “Shops” and “Products” linked together by the relationship “Sell”. Requirement 2 indicates that “One product could be sold at multiple shops at different prices”, which means that “Price” can be a property of the “Sell” relationship. However, we also need to record “the history of price changes”, which also vary for the same product across different shops. Due to this complexity, we chose to make a weak entity “Products In Shop” which is both a subclass of “Products” and has a supporting “Sell” relationship with “Shops”. The “Products” entity set can then have the general attributes such as “category” and “maker” which are shared while the “Products in Shops” have “Stock” and “Price” that vary from shop to shop. The price “Records” can then be a weak entity set dependent on “Products in shop”. Since prices fluctuate for periods of time, the “Start date” and “End date” are also captured as properties of “Records”. By the same token, since “Each order involves one or more products, which could be from different shops”, and the products involved in an order has their own properties like “quantity ordered” and “status”, we included another weak entity “Products on Order” since each product in an order is of “Products in Shop” but is tracked separately by the Order ID. Note that the entity set “Products on Order” also has a “Price” quantity which is separate from “Products in Shop”, since the price of a product at the time of ordering is assumed to be different from the price of that product in the same shop at the current moment.

The types of relationships and referential integrity constraints are thoroughly discussed. For example, the “Place” relationship between “Orders” and “Users” is many-to-one, with referential integrity since each user can place multiple orders, but each order with a unique “Order ID” can only be placed by exactly one user. Likewise for the “File” relationship between “Complaints” and “Users”. Requirement 5 states “After a user purchases a product, he/she is allowed to rate and comment on any product once”. Since it is not mandatory for the user to give a rating or a comment on their product, we did not include a referential integrity constraint on “Rates” relationship between “Users” and “Products on order”. Also, while the wording seems to suggest that any product can be rated or commented, which should not be the case. Users should only be allowed to rate and comment on the product they have purchased. This led to the link between “Rates” and “Products on Order”. Note that the same product may be rated differently across different shops, since they may sell different qualities of the same product.

“The average rating for a product, as well as the number of users that have rated the product, are shown on the web page.” Since this can be done through the client/server, it does not belong in the database and thus we have excluded it from the ER diagram.

Due to the wording of “After the complaint is picked up by a Sharkee employee” we assume that a complaint is not assigned to an employee by default at the time that the complaint is lodged, and there is a possibility that it will not be handled at all and may be left pending indefinitely. Therefore, we did not include a referential integrity constraint between “Employees” and “Complaints”. The “Status” (“pending”, “being handled”, “addressed”) is hence a property of the entity set “Complaints” and not a property of the relationship “Handle”. “Complaints on Products” and “Complaints on Shops” are two different subclasses created because they have different relationships with other entity sets, namely “Products on order” and “Shops” respectively.


Users
User ID
Name
Place
Orders
Total cost
Order Id
Timestamp
Shipping Address
In
Shops
Shop Name
Sell
Rating
Comment
Rates
Quantity
Delivery Date
Status
Price
Products on order
In
Products in shop
Price
Stock
Has
Records
Price
Start date
End date
Is a
Products
Product Name
Category
Maker
Product Id
File
On
Status
Complaint ID
Complaints
Is a
Complaints on Product
Is a
Complaints on Shop
On
Handle
Employees
Monthly Salary
Employee ID
Name