COMP2411 Database System First-stage Project

Wang Ruijie, Zhu Jin Shun, Zeng Tianyi and Liu Yuyang

November 7, 2023

1 Assumptions of the Entity Relationship Model

We assume that the *Online Shopping System* should be with the following entities, attributes, relationships and relevant properties:

1. The entity sets *Customer* and *Administrator* are two subclasses of *User*, and both of them are weak with the identifying relationship *INHERITS*. E.g., for a person served by our proposed *Online Shopping System*, he or she belongs to two entity sets, i.e., *User* and *Customer*. Similarly, for a manager, he or she belongs to both *User* and *Administrator*. As a result, a customer or administrator has his or her attributes by inheriting from *User*.

2. The *INHERITS* relationship is set to be an one-to-one relationship, as it represents the relationship of inheritance between a superclass and a subclass, and *Customer* and *Administrator* can not exist without their superclass *User* (See the Stack Overflow question we referred to). Hence, the *INHERITS* relationship is fully participated by *Customer* and *Administrator* and partially participated by *User*.

3. *User* is uniquely identified by *User ID*, so do *Customer* and *Administrator*. *Status* is the attribute to display a user's account status (online, offline, canceled). *User* uniquely holds attributes *User Name* and *Password*, while *Personal Information* is multi-valued for there can be multiple contents (profile, age, gender, self-introduction, etc.).

4. *Registration* is provided as an attribute for users to register and create new accounts. Please note that we do not explicitly separate user account, customer account and administrator account from *User*, *Customer* and *Administrator*. Instead, the concepts of users and users' accounts are integrated by *User*, *Customer* and *Administrator*.

5. *Product* is uniquely identified by *Product ID*. We regard *Description* and *Specification* as the integrations (sets) of a product's descriptions and specifications, while *Customer Reviews* is multi-valued since customer reviews are in pieces.

6. A product individually holds an individual Inventory. Therefore, we assume that *Inventory* is a weak entity set with the identifying relationship *HOLDS*. *HOLDS* is fully participated by both.

7. A customer individually owns and adjusts a shopping cart. Therefore, we assume that *Shopping Cart* is a weak entity set with the identifying relationship *ADJUSTS*. *ADJUSTS* is fully participated by both.

8. *Product* can be selected by *Customer* to *Shopping Cart*. Based on that, *SELECTS* is established, which is an one (*Customer* and *Shopping Cart*)-to-many (*Product*) relationship.

9. *User* may filter *Product* with given criteria. Hence, we construct *FILTER* relationship with the multi-valued attribute *Criteria*.

10. *Order* is uniquely identified *Order ID* and it is with foreign keys *User ID* and *Product ID*. In reality, *Order ID* can uniquely identify an order, but the foreign key constraint should be shown, for *Order* is meaningless without the existence of *User* and *Product*. *Order* has *Fee* (derived from prices of the products and the quantities of products), *Shipping Address*, *Quantities* (multi-valued, one quantity for each kind of products), *Status* (recognized, shipped, received or reviewed), and *Payment* (the method of paying) as other attributes.

11. An order can contains multiple products, and every order must contain at least one product. Hence, *CONTAINS* is partially participated by *Product* and fully participated by *Order*. And an order must be initialized by a check-out, and before the check-out, it should be added to a shopping cart. Therefore *CHECKS OUT* is with full participation of *Order* and partial participation of *User* or *Shopping Cart*. When checking out, *Date & Time* are supposed to be recorded. In addition, a customer is able to check out one or multiple products for one order.

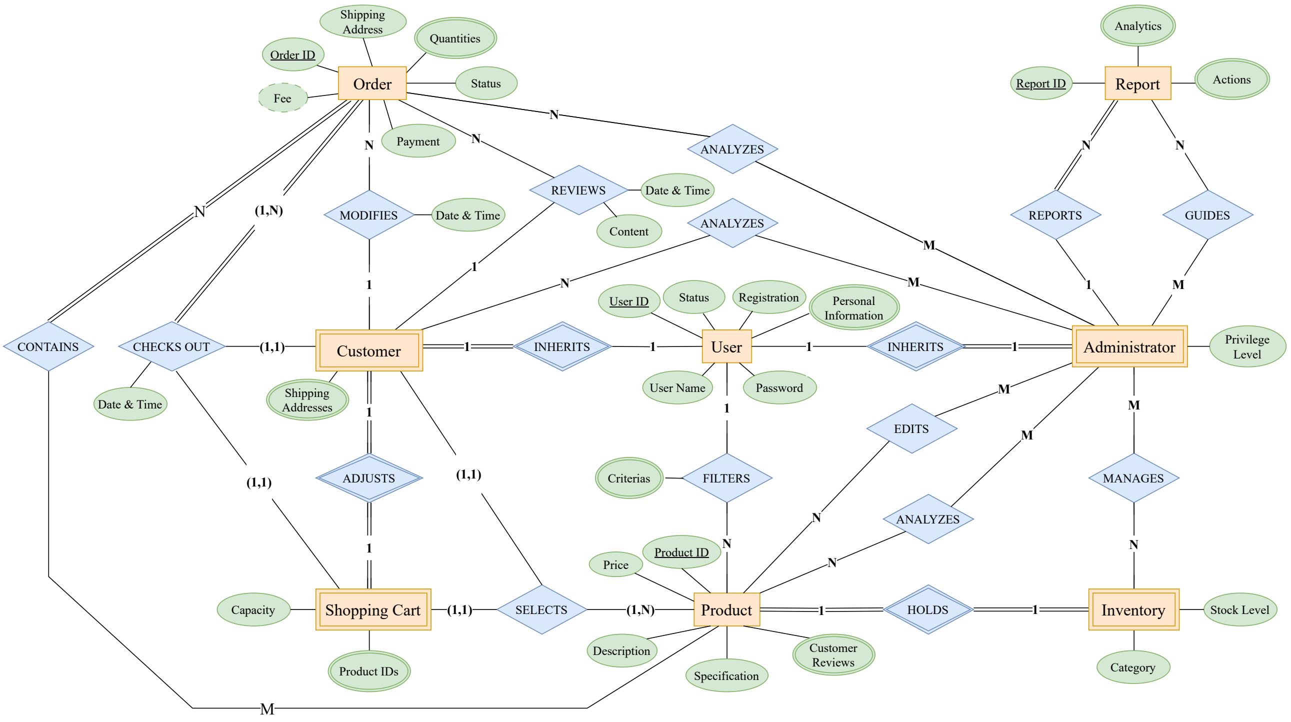
12. If *Status* of *Order* is not turned to be shipped, *Customer* can modify the *Order* via *MODIFIES*, and similarly, *Date & Time* will be recorded.

13. After *Status* turns to be received, *Customer* can review the *Order* (and the products purchased in the order). Then *REVIEWS* is constructed with *Date & Time* and *Content*.

14. *Administrator* has access to analyze *Order*, *Customer* and *Product* with his or her *Privilege Level* showing his or her visibility for *EDITS*. Afterwards, *Administrator* can generates *Report* with *Analytics* (multi-valued) and *Actions* (multi-valued), guiding other administrators or himself or herself. Please note that *Report* is uniquely identified by its own *Report ID*. Meanwhile, *Report* can only be written by one administrator. Thus, *Report* is with the foreign key *User ID*. Also, no report can be generated before *REPORTS*, hence the relationship REPORTS should be with *Report's* fully participation.

2 Entity Relationship Diagram

Based on the proposed assumptions, our entity relationship diagram is as follows:



3 Relational Schema

3.1 Relational schemata for entities

1. *User (User ID, Password, User Name, User Status, Registration, Personal Information)*
2. *Customer (User ID, Shipping Addresses)*
3. *Administrator (User ID, Privilege Level)*
4. *Product (Product ID, Price, Description, Specification, Customer Reviews)*
5. *Inventory (Product ID, Category, Stock Level)*
6. *Shopping Cart (User ID, Product IDs, Capacity)*
7. *Order (Order ID, Fee, Shipping Address, Quantities, Status, Payment)*
8. The foreign keys of *Order: User ID, Product ID*
9. *Report (Report ID, Analytics, Actions)*
10. The foreign key of *Report: User ID*

3.2 Relational schemata for relationships

1. *FILTERS (User ID, Product ID, Criteria)*
2. *CONTAINS (Product ID, User ID, Order ID)*
3. *CHECKS OUT (Order ID, Product ID, User ID, Date & Time)*
4. *MODIFIES (Order ID, Product ID, User ID, Date & Time)*
5. *REVIEWS (Order ID, Product ID, User ID, Content, Date & Time)*
6. *SELECTS (User ID, Product ID)*
7. *ANALYZES*$_{Administrator-Order}$ *(User ID*$_{Administrator}$ *, Order ID, Product IDs, User ID*$_{Customer}$*)*
8. *ANALYZES*$_{Administrator-Customer}$ *(User ID*$_{Administrator}$*, User ID*$_{Customer}$*)*
9. *ANALYZES*$_{Administrator-Product}$ *(User ID, Product ID)*
10. *EDITS (User ID, Product ID)*
11. *MANAGES (User ID, Product ID)*
12. *REPORTS (User ID, Report ID)*
13. *GUIDES (User ID*$_{Administrator\_\_reporter}$*, Report ID, User ID*$_{Administrator\_\_guidee}$*)*
14. Please note that we do not give the schemata for relationships of *INHERITS*$_{Customer\text{-}User}$, *INHERITS*$_{Administrator\text{-}User}$, *HOLDS* and *REPORTS*, because they are all binary identifying relationships.