

Project Design Diagram

CS 157A Team 30

Aaron Warren, Phu Tran, Evan Ugarte

E/R Diagram Description

Entity Sets

Customer

- This entity set contains the users' general information such as user's name, address, phone number and member ID. The member ID is the primary key for the entity set Customer. It shared a one to one relationship with the entity set Account.

Account

- This entity set holds the user's account information where its key is account ID. This entity set shares 3 relationship with the other entity sets such as one to one relationship with entity set Customer, one to many relationship with entity set Card Info and another one to many relationship with the entity set Orders.

Card Info

- This entity set contains the user's card payments information where the key is the Card ID. This entity set shares an one to many relationship with the entity set Account.

Orders

- This entity set represent the user's shopping cart where the key will be its order ID attribute. This entity set sharder an one to many relationship with the entity set Account

Item

- This entity holds the relevant information for each item which includes the name, price, category id, and item id. The primary key for each entity is the item ID. The entity set shares a one-to-one relationship with the item stock levels entity set.

Item Stock Levels

- This entity set has a one to one relationship with the entity set items. This is because as an item can only have one stock level. The entities in the set are identified by the primary key stockID, and are paired to items by the Item's ID.

Item Categories

- This is an entity set that will hold the various categories for Items. The primary key Category ID identifies entities in this set, Category Name is also another attribute. An example of Category name can be "Chips" or "Soda".

Relationships

Owns

- This relationship has a one to one relationship between the entity sets Customer and Account. It is a one to one relationship because each customer can only own exactly one account and in the same hand, each account belongs to exactly one customer. Its attribute will have the keys from both entity sets which are Member ID and Account ID.

Holds

- This relationship has a one to many relationship between the entity sets Account and Card Info. It is a one to many relationship because each account can hold many different card information as a payment methods. On the other hand, we will assume that each card information can only be stored in exactly one account.

Make

- This relationship has a one to many relationship between the entity sets Account and Orders. It is a one to many relationship because each account can make many orders and each specific order can be made by one account.

Has

- This relationship has a one-to-one relation between the entity sets Item and Item Stock Levels. It is one-to-one because each item will only have a single stock level to maintain the quantity of each item in the warehouse.

contain

- This relationship has a one-to-many relation between the entity sets Orders and Item. It is one-to-many because each order can hold many different items. The reason for this is because a person is not restricted to buying only a single item.

Belongs

- This is a relationship between Item and Item Categories. The relationship is many to many, as Items can have multiple categories and categories have multiple items.

Schemas and Tuples

Entity Sets -> Relations

Customer(memberID, name, cellPhone, address)

- (1234, "Peter Pan", "123-456-1220", "145 Story Rd")
-

Account(accountID, memberID, email, password)

- ("A1234", 1234, "peterpan@gmail.com", mypassword)
-

CardInfo(cardID, cardHolder, cardNumber, CVV, zipcode)

- (1, "Peter Pan", 1234567890987654, 879, 95111)
-

Orders(orderID, itemID, memberID, price)

- (420, 731, 1234, 25.00)
- (???, 111, 1234, ???.??)
-

Item(itemID, name, price, category)

- (731, "black t-shirt", 15.00, "clothing")
- (111, "Hershey Chocolate", 10.00, "candy")
-

ItemCategories(categoryID, categoryName)

- (222, "clothing")
- (333, "candy")
-

ItemStockLevels(stockID, itemID, stockDate, quantity)

- (999, 731, 2019-02-14, 100)
- (879, 111, 2019-04-25, 57)
-

Relationships -> Relations

Owns(memberID, accountID)

- (1234, "A1234")
-

Holds(accountID, cardID)

- ("A1234", 1)
-

Make(accountID, orderID)

- ("A1234", 420)

Contain(orderID, itemID)

- (420, 731)
- (420 111)
-

Has(itemID, stockID)

- (731, 999)
- (111, 879)
-

Belongs(itemID, categoryID)

- (731, 222)
- (111, 333)
-



