Online Retail Database

Mark Kim

Student ID: 918204214

GitHub Username: mkim797

Milestone/Version	**Date**
M1V1	09/11/2021
M1V2	10/30/2021
M2V1	10/31/2021
M2V1.5	11/1/2021

Contents

I Project Description 2

II Use Cases 3

A Use Case 1 3

B Use Case 2 3

C Use Case 3 3

D Use Case 4 4

E Use Case 5 4

III Database Requirements 6

IV Detailed List of Main Entities, Attributes and Keys 10

V Entity Relationship Diagram (ERD) 14

VI Testing Table 15

VII Enhanced Entity Relationship Model (EER) 17

VIII Forward Engineering 25

IX Inserting Data 25

X Testing 25

XI Testing Table 26

Section I: Project Description

The goal of this project is to design and implement an online retail database that functions similarly to the online portals of major retailers such as Walmart, The Home Depot, and Amazon. This database system is to manage all aspects of online retail operations from product acquisition and warehousing, to customer order fulfillment and followup. By implementing this database, the online retailer will be able to better understand product turnover, supply chain, human resourcing, fulfillment efficiency, and many other key performance indicators.

From the vendor side of the business, the database will manage the supply chain. This will allow the business to ensure that its resources are applied as efficiently as possible. The business will be able to understand how long it will take for each incoming order to be fulfilled so they can plan for the correct amount of warehouse space to be allocated for each order. Additionally, the business will be able to choose multiple different vendors to aggregate supply capacity in the case that there are supply chain breakdowns. This will minimize the chance for an interruption in the business's ability to fulfill orders to their customers.

Of course, as an online retail business, customers need to be carefully managed. This database will facilitate the management of customer accounts, transactions, order fulfillment, and purchase habits. By tracking these items, the business will be better equipped to manage their inventory, vendors, and sales trends. By managing customer transactions, the business will be able to make informed decisions about how to manage its supply chain; the business will know how much of each product to order from its vendors. As an added value to customers, the database will also allow customers to comment and review the products they order and their general experience with the business.

Lastly, and possibly most importantly, this project will manage internal operations. This includes, but is not limited to, inventory management, warehousing and distribution, human resources and capital resources. With this system, the business will be able to manage its staffing requirements and responsibilities, which warehouses to use when fulfilling customer orders, management and administrative hierarchy, and working capital. This will allow the business to maintain smooth operations by ensuring proper staffing at all of its locations. Furthermore, the business will be able to resource its distribution network to ensure timely and efficient fulfillment of orders. Finally, the database will be able to support multiple roles for employees to ensure a separation of duties.

Section II: Use Cases

Use Case 1

Title: Product Purchase from Website

Actors: Customer (Leia Organa)

Description: Leia is someone who is looking for a new circular saw. She is not happy with the last one she bought since it lacks the feature set she wants. She wants to browse to see what kind of saws that the website has available. She finds that there are categories that she can use to limit her search and since the saw is a type of tool, she chooses the tool category. Nevertheless, tools have a lot more other subcategories, so she continues drilling down the menus until the list is limited to only circular saws. Now that she has a nice succinct list of saws, she is able to compare them and find the one she wants.

Once found, Leia adds the saw she wants to her shopping cart. She continues browsing a little longer to see if there is anything else she wants, but decides that she doesn't need anything else and checks out. After entering in her shipping address and billing address, she chooses a shipping method and completes the transaction. Once completed, Leia receives an order confirmation email and when the item ships, she receives her receipt with all the relevant shipping details.

Use Case 2

Title: Product Order from Supplier

Actors: Buyer (Han Solo)

Description: Han is a buyer for the cutlery category of products at one of the warehouses of our online retailer. Han's responsibilities include tracking sales and inventory so that he knows exactly what products to buy and when so that the retailer has enough supply to handle demand at any given moment. There are many suppliers and manufacturers of cutlery, so he needs to ensure that fulfillment of the orders happen in a timely fashion. If not, he needs to either find another supplier that can fulfill the order or find a substitute item that will satisfy the customers. With all this in mind, Han logs into the system and checks the inventory of all the cutlery in stock at this warehouse. He then contacts the supplier and orders many knives to be stocked in the warehouses. He then enters the pending order in so that he can track the status of it. Once completed, he continues ordering more cutlery from other vendors until he is sure that the warehouse will continue to have enough supply to handle demand.

Use Case 3

Title: Customer opens an account

Actors: Customer (Lando Calrissian)

Description: Lando shops at our online retailer regularly. Because of this, he wants us to save his information to make his shopping experience more effortless. He knows that once he opens an account, he can also sign up for our rewards program, where he can redeem points for credit on future purchases. He is also aware of our premium subscription where he receives free expedited shipping for all his orders, but he is not sure if he wants to pay for this yet since he doesn't really need expedited shipping and all his orders qualify for free ground shipping anyways.

Once he opens his account, he can save multiple shipping addresses and multiple payment methods, which allows him to purchase items quickly and effortlessly. He can also view all his purchase history and track the order status of all his orders. Once he receives enough points in the rewards program, he can redeem the points for things like gift cards and special discounts or gifts.

Use Case 4

Title: Customer Order is Fulfilled

Actors: Warehouse Employee (Kylo Ren)

Description: This use case begins as Kylo goes to his station. Once there, he logs into his computer terminal which shows him what orders he needs to box up and ship out. The computer terminal gives him the order number with a packing list of all the items to be packed up. All the work of locating and pulling products has been automated, so his job is to double check the bin prepared by the robots and pack them securely in a box. If there is an error, it is his responsibility to remove any incorrect products to be returned to the warehouse and request any products that may be missing. Each of these errors are entered into the terminal along with any corrective action taken. For example, if the warehouse shows that there is no inventory of a product, he must note that on his terminal so that it can either be fulfilled by another warehouse or notify the customer of a backorder or cancellation. Once he verifies and packs all the items listed on his terminal, he completes the order by including a packing list inside and labeling the box with the shipping label provided to him. Once he completes the order, he scans the box and sends it down the conveyor belt to be routed to the correct pickup area of the corresponding courier for delivery.

Use Case 5

Title: Product Return

Actors: Customer (Boba Fett), Warehouse Employee (Padme Amidala)

Description: Boba bought a microwave from the online retail store but it arrived broken, so she wants a replacement. Since he is an account holder, he can log into his account and begin his return process from his order history. After providing the reason for his return and once the system authorizes his return, he is given options for how he would like the return processed. He chooses to get a replacement and bring the broken item to a participating brick and mortar retailer to pack and ship the product back because that is most convenient for him. After completing the transaction, he is provided with a return ID that he is to supply the retailer along with the item to be returned. He then gives the item to the retailer within the allowable time limit. The retailer then scans the item, labels the return, and aggregates all the items that were received during the day to be picked up with a contracted courier to be returned to the closest warehouse. Once the item is scanned, the fulfillment center creates a new order to replace the broken item.

Once the warehouse receives the item, Padme is tasked with finishing the return process. Padme already is logged into her work terminal. She takes an item from the bin full of returns and scans it. Once scanned, the item is flagged as received. Padme then reviews the product and determines the cause of its inability to operate (i.e. damage from shipment, manufacturer defect, etc). Once the cause is determined, she enters it into her terminal so that the correct entity is notified and corrective action is taken.

Section III: Database Requirements

1. General User (Strong)
   (a) A general user shall create only one account with a unique email.
   (b) A general user shall be able to register to many mailing lists
   (c) A general user shall have many addresses
   (d) A general user shall have many payment methods
   (e) A general user shall create many orders
   (f) A general user is a registered user
2. Registered User (Strong - Recursive)
   (a) A registered user is a general user.
   (b) A registered user shall be allowed to login to their account from many supported devices simultaneously.
3. Device (Strong)
   (a) A device shall log into zero or one account
   (b) A device shall be used by many registered users
4. Account
   (a) An account shall be created by one and only one user
   (b) An account is a Basic Account
   (c) An account is a Premium Account
   (d) An account shall be logged into by many devices
   (e) An account shall accrue at least one reward
5. Basic Account
   (a) A Basic account is an account
6. Premium Account
   (a) A Premium account is an account
7. Reward (Strong)
   (a) A reward shall be accrued by many accounts
8. Mailing List (Strong)
   (a) A mailing list shall have many users
9. Order (Weak)
   (a) An order is a purchase

    (b) An order is a return

    (c) An order ships to at most one shipping address

    (d) An order has one order details

    (e) An order creates at least one fulfillment action

10. Purchase

    (a) A purchase is an order

11. Return

    (a) A return is an order

12. Order Details (Weak)

    (a) An order detail generates one receipt

    (b) An order detail receives payment from at least one payment method

    (c) An order detail contains at least one product

    (d) An order detail sends payment details to one accounting department

13. Receipt

    (a) A receipt shall be generated by one and only one order details

14. Accounting Dept (Strong)

    (a) An accounting dept shall receive payment details from many order details

    (b) An accounting dept shall pay many payees

15. Payee (Strong)

    (a) A payee shall be payed by many accounting departments

16. Address (Strong)

    (a) An address shall be associated with many general users

    (b) An address is a billing address

    (c) An address is a primary address

    (d) An address is a shipping address

17. Primary Address

    (a) A primary address is an address

18. Billing Address

    (a) A billing address is an address

    (b) A billing address has at least one payment method

19. Shipping Address

    (a) A shipping address is an address

    (b) A shipping address is shipped to by many orders

20. Payment method (Strong)
    (a) A payment method is a Bank Account
    (b) A payment method is a Credit Card
    (c) A payment method is Cryptocurrency
    (d) A payment method pays for many orders
    (e) A payment method has one billing address
    (f) A payment method has at most one general user
21. Bank Account
    (a) A bank account is a payment method
    (b) A bank account is a checking account
    (c) A bank account is a savings account
22. Checking account
    (a) A checking account is a bank account
23. Savings account
    (a) A saving account is a bank account
24. Credit card
    (a) A credit card is a payment method
25. Crypto
    (a) Crypto is a payment method
26. Fulfillment Action (Strong)
    (a) A fulfillment action is created by many orders
    (b) A fulfillment action uses at most one courier
    (c) A fulfillment action has at most one region
27. Region (Strong)
    (a) A region has many fulfillment actions
    (b) A region has many warehouses
28. Courier (Strong)
    (a) A courier is used by many fulfillment actions
29. Warehouse (weak?)
    (a) A warehouse has one region
    (b) A warehouse requests many inventory items from many suppliers
30. Product (Strong)
    (a) A product is ordered from many order details

(b) A product is stocked in many inventories

(c) A product has many categories

(d) A product has many promotions

31. Promotion (Strong)

    (a) A promotion is associated with many products

32. Category (Strong)

    (a) A category has many products

33. Inventory (Strong)

    (a) An inventory contains many products

    (b) An inventory receives many items from many suppliers

    (c) An inventory stocks many warehouses

34. Supplier (Strong)

    (a) A supplier sends many items to inventory

    (b) A supplier is requested by a warehouse to send many items

Section IV: Detailed List of Main Entities, Attributes and Keys

1. General User (Strong)
   - user_id: key, numeric
   - name: composite, alphanumeric
   - email: alphanumeric
   - phone: numeric
2. Registered User (Strong)
   - registered_user_id: key, numeric
   - login: alphanumeric
   - account_id: foreign key, numeric
3. Address (Strong)
   - address_id: key, numeric
   - address: composite, alphanumeric
   - city: alphanumeric
   - state: alphanumeric
   - zip_code: numeric
   - country: alphanumeric
   - address_type: numeric
4. Device (Strong)
   - device_id: key, numeric
   - device_type: alphanumeric
   - device_os: alphanumeric
   - device_ip: numeric
5. Mailing List (Strong)
   - mailinglist_id: key, numeric
   - mailinglist_name: alphanumeric
   - description: alphanumeric
   - mailinglist_category: alphanumeric
6. Account (Weak)
   - account_id: key, numeric
   - registered_user_id: foreign key, numeric

   - account_type: numeric

7. Region (Strong)
   - region_id: key, numeric
   - location: composite, alphanumeric
   - service_area: numeric
8. Promotion (Strong)
   - promotion_id: key, numeric
   - promotion_type: numeric (e.g. discount, free item, etc)
   - promotion_limit: numeric (e.g. applies only once, or maximum 3 items, etc)
9. Reward (Strong)
   - reward_id: key, numeric
   - reward_type: numeric (e.g. discount, free item, etc)
   - reward_limit: numeric (e.g. applies only once, or maximum 3 items, etc)
10. Payment Method (Strong)
    - payment_id: key, numeric
    - billing_address_id: foreign key, numeric
    - payment_type: alphanumeric
11. Order (Weak)
    - order_id: key, numeric
    - user_id: foreign key, numeric
    - shipping_address_id: foreign key, numeric
    - order_type: numeric
12. Order Details (Weak)
    - order_details_id: key, numeric
    - order_id: foreign key, numeric
    - payment_method_id: foreign key, numeric
    - receipt_id: foreign, numeric
13. Receipt (Weak)
    - receipt_id: key, numeric
    - order_details_id: key, numeric
    - receipt_generated: timestamp
14. Product (Strong)
    - product_id: key, numeric

- UPC_code: unique, numeric
- product_name: alphanumeric
- product_description: alphanumeric
- product_price: numeric
- product_cost: numeric

15. Category (Strong)
    - category_id: key, numeric
    - category_name: alphanumeric
    - category_description: alphanumeric
16. Inventory (Strong)
    - inventory_id: key, numeric
    - category_name: alphanumeric
    - category_description: alphanumeric
17. Supplier (Strong)
    - supplier_id: key, numeric
    - location: multivalue, alphanumeric
    - description: alphanumeric
18. Warehouse (Weak)
    - warehouse_id: key, numeric
    - region_id: foreign key, numeric
    - location: multivalue, alphanumeric
19. Fulfillment Action (Strong)
    - fulfillment_id: key, numeric
    - order_id: foreign key, numeric
    - package_dimensions: composite, numeric
    - package_weight: numeric
    - shipping_priority: alphanumeric
    - fulfillment_status: numeric
20. Courier (Strong)
    - courier_id: key, numeric
    - courier_account_id: alphanumeric
    - service_area: alphanumeric
    - service_limit: alphanumeric (e.g. maximum/minimum package size/weight)

21. Accounting Dept (Strong)
    - accounting_dept_id: key, numeric
    - order_details_id: foreign key, numeric
    - payment_date: timestamp
22. Payees (Strong)
    - payee_id: key, numeric
    - payment_amount: numeric
    - payment_status: alphanumeric

Section V: Entity Relationship Diagram (ERD)

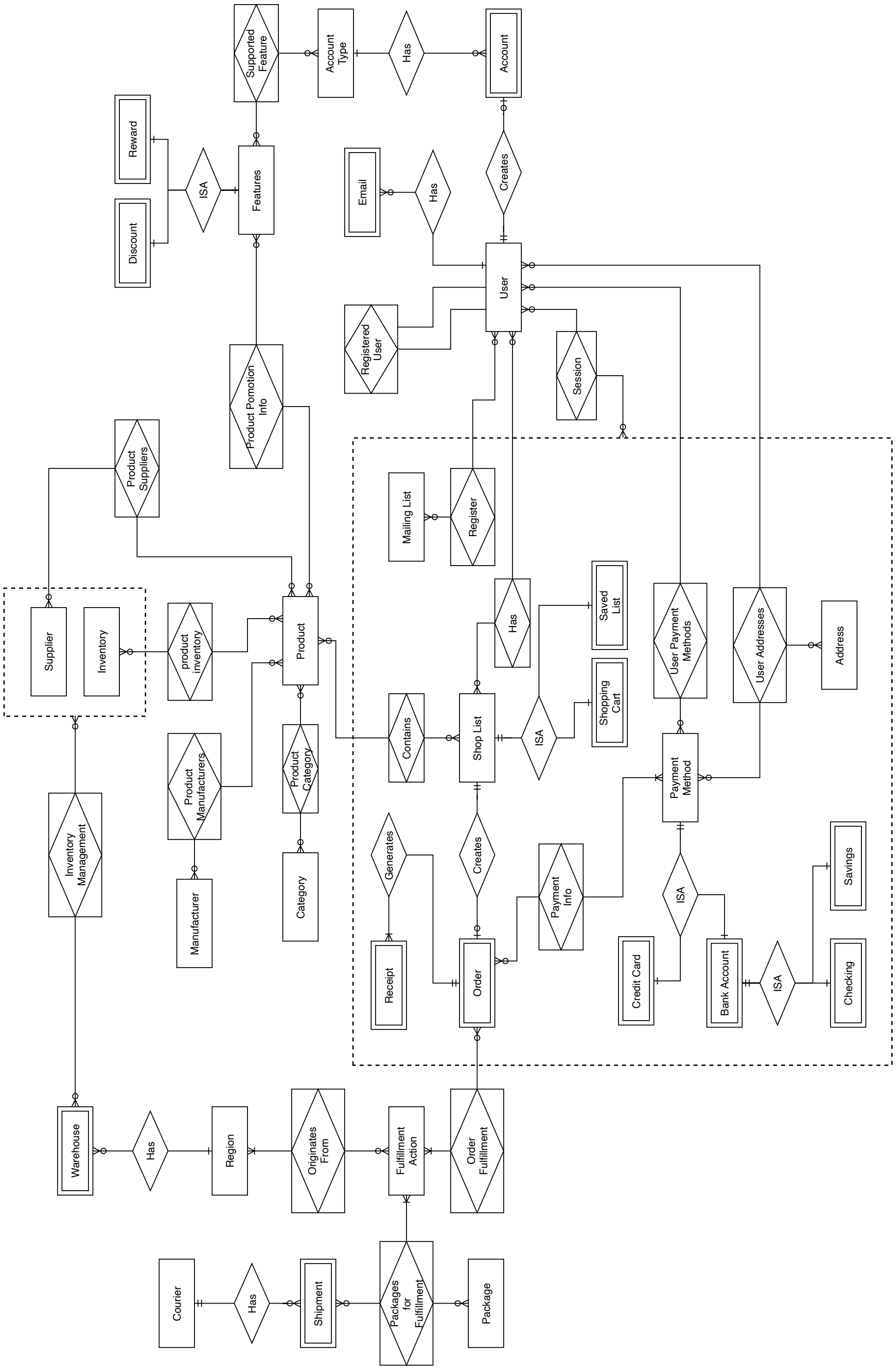

Figure 1: ERD

Section VI: Testing Table

Table 1: Entity Test

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error Description
1	General User	Create	Account	1-to-1	Pass	None
2	General User	User Address Info	Address	M-to-M	Pass	None
3	General User	User Payment Info	Payment Method	M-to-M	Fail	A payment method should be associated with at most one user. If a payment method can be associated with several users, there could be security issues.
4	General User	Registers	Mailing List	M-to-M	Pass	None
5	General User	Creates	Order	1-to-1	Fail	A user should be able to order more than once.
6	Registered User	Open Session	Device	1-to-M	Fail	Device can be used by multiple users.
7	Device	Logs to	Account	M-to-1	Fail	Device does not necessarily have to be logged in. Can't force a device to be logged in.
8	Account	Reward Info	Reward	M-to-M	Pass	None
9	Account	Receives	Promotion	M-to-M	Pass	None
10	Order	Ships to	Shipping Address	1-to-1	Fail	A single order should be shipped to at most one shipping address. To ship to a different address, another order should be generated. Also, the shipping address does not have to be shipped to by an order.
11	Order	Has	Order Details	M-to-M	Fail	A single order should have only one order detail associated with it. An order detail should be associated with one and only one order.
12	Order	Create	Fulfillment action	M-to-M	Pass	None

Table 1: Entity Test

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error Description
13	Order Details	Payed by	Payment Method	M-to-M	Pass	None
14	Order Details	Generate	Receipt	1-to-1	Pass	None
15	Order Details	Send	Accounting Dept	M-to-M	Fail	Order details must send details to one accounting department. And Accounting department does not necessarily have to receive details from Order details.
16	Order Details	Has	Product	M-to-M	Pass	None
17	Fulfillment action	Use	Courier	M-to-M	Fail	A fulfillment action should use at most one courier.
18	Fulfillment action	Has	Region	M-to-M	Fail	A fulfillment action should be associated with at most one region
19	Region	Has	Warehouse	1-to-M	Pass	None
20	Warehouse	Requests	Inventory	M-to-M	Pass	None
21	Warehouse	Requests from	Supplier	M-to-M	Pass	None
22	Product	Contains	Inventory	M-to-M	Pass	None
23	Product	Has	Category	M-to-M	Pass	None
24	Product	Promotion Info	Promotion	M-to-M	Pass	None
25	Supplier	Sends	Inventory	M-to-M	Pass	None
26	Accounting Dept	Pays	Payees	M-to-M	Pass	The accounting department should not be forced to be paying payees, but payees do not require the accounting department to always be paying them.

Section VII: Enhanced Entity Relationship Model (EER)

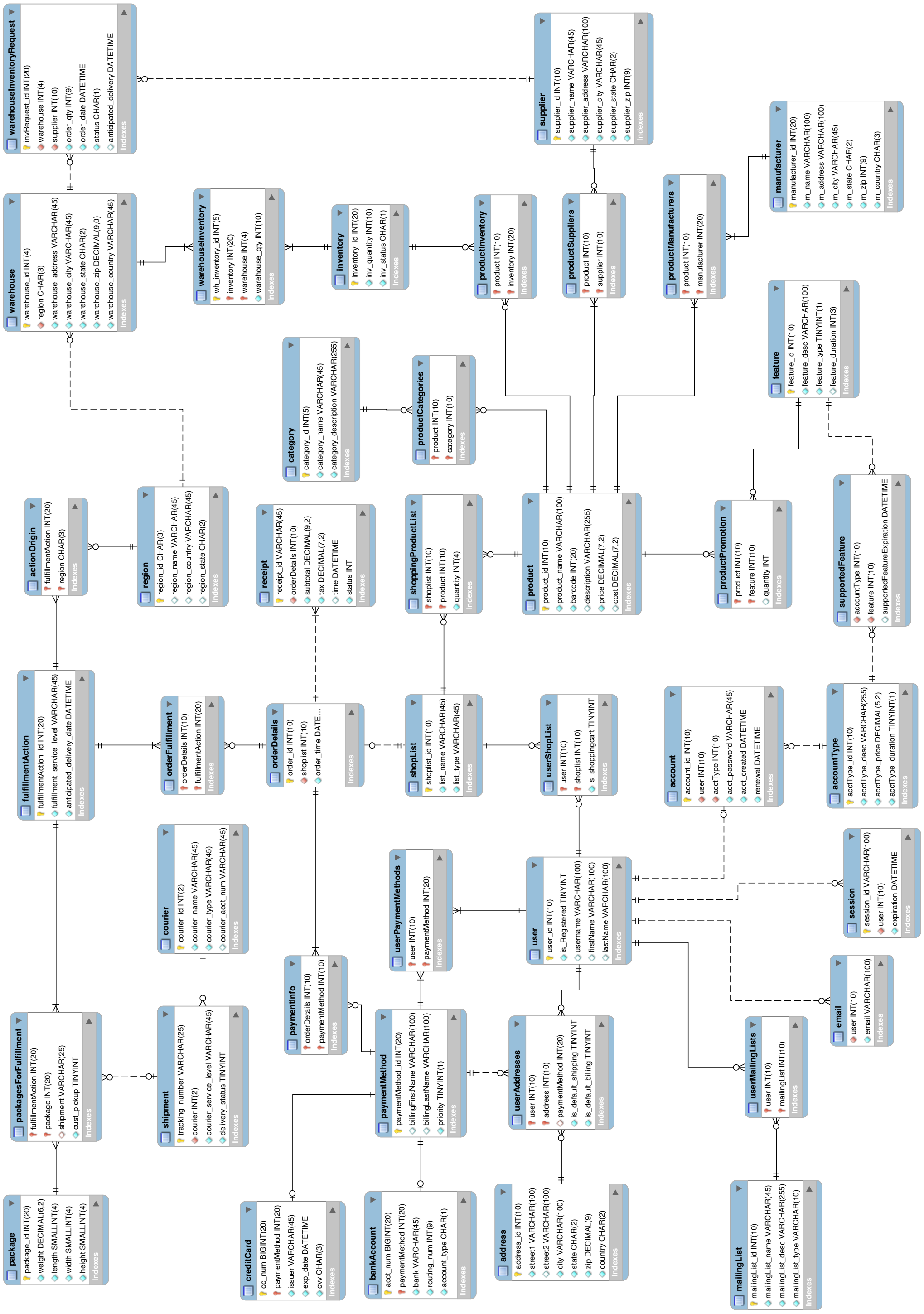

Figure 2: EER Diagram

Table 2: Implementation for Foreign Keys

Table	FK	ON DELETE	ON UPDATE	Comments
packagesForFulfillment	fulfillmentAction	CASCADE	CASCADE	If a fulfillment action is deleted/updated, then the packages for fulfillment should be deleted/updated as well
packagesForFulfillment	package	CASCADE	CASCADE	If a package is deleted/updated, then the packages for fulfillment should be deleted/updated as well
packagesForFulfillment	shipment	SET NULL	CASCADE	If a shipment is updated, then the packages for fulfillment should be updated as well. If a shipment is deleted, then the shipment info in packagesForFulfillment should be set to null and cust_pickup flag set to true; this is the case where the customer chooses to pick up the order at the warehouse.
actionOrigin	fulfillmentAction	CASCADE	CASCADE	If a fulfillment action is deleted/updated, then the action origin should be deleted/updated as well

Table 2: Implementation for Foreign Keys

Table	FK	ON DELETE	ON UPDATE	Comments
actionOrigin	region	RESTRICT	RESTRICT	A region should be restricted from deletion/update since the action is non-trivial. Otherwise, many fulfillment actions associated with a particular region will end up being orphaned or changed to a region that may not be appropriate for the action. Hence, deletion/update of a region should only occur when there is a new region that can replace it or if the region still overlaps with the previous scope.
warehouse	region	RESTRICT	RESTRICT	A region should be restricted from deletion/update since the action is non-trivial. Deletion/update of a region should only occur when there is a new region that can replace it or if the region still overlaps with the previous version. Otherwise, warehouses would end up being deleted/updated along with the region. Because they are physical spaces, they cannot move if the scope of a region is changed.

Table 2: Implementation for Foreign Keys

Table	FK	ON DELETE	ON UPDATE	Comments
warehouseInventoryRequest	warehouse	CASCADE	CASCADE	A deletion/update of a warehouse should delete/update an inventory request.
warehouseInventoryRequest	supplier	CASCADE	CASCADE	A deletion/update of a supplier should delete/update an inventory request.
warehouseInventory	inventory	CASCADE	CASCADE	A deletion/update of an inventory should delete/update a warehouse inventory.
warehouseInventory	warehouse	CASCADE	CASCADE	A deletion/update of a warehouse should delete/update a warehouse inventory.
receipt	orderDetails	CASCADE	CASCADE	A deletion/update of an order should delete/update a receipt.
orderFulfillment	orderDetails	CASCADE	CASCADE	A deletion/update of an order should delete/update an order fulfillment.
orderFulfillment	fulfillmentAction	CASCADE	CASCADE	A deletion/update of a fulfillment action should delete/update an order fulfillment.

Table 2: Implementation for Foreign Keys

Table	FK	ON DELETE	ON UPDATE	Comments
shipment	courier	RESTRICT	CASCADE	A deletion of a courier should be restricted since many shipments are associated with a particular courier. Deletion of a courier should only occur if a replacement courier has already been considered for all shipments associated with the courier. An update of a courier should update a shipment.
creditCard	paymentMethod	CASCADE	CASCADE	A deletion/update of the payment method should delete/update the credit card.
bankAccount	paymentMethod	CASCADE	CASCADE	A deletion/update of the payment method should delete/update the bank account.
paymentInfo	orderDetails	CASCADE	CASCADE	A deletion/update of an order should delete/update the payment information.
paymentInfo	paymentMethod	CASCADE	CASCADE	A deletion/update of a payment method should delete/update the payment information.
orderDetails	shopList	CASCADE	CASCADE	A deletion/update of a shopping list should delete/update the order.
productCategories	product	CASCADE	CASCADE	A deletion/update of a product should delete/update the product categories.

Table 2: Implementation for Foreign Keys

Table	FK	ON DELETE	ON UPDATE	Comments
productCategories	category	CASCADE	CASCADE	A deletion/update of a category should delete/update the product categories.
shoppingProductList	shopList	CASCADE	CASCADE	A deletion/update of a shopList should delete/update the shopping product list.
shoppingProductList	product	CASCADE	CASCADE	A deletion/update of a product should delete/update the shopping product list.
userPaymentMethods	user	CASCADE	CASCADE	A deletion/update of a user should delete/update the user payment method.
userPaymentMethods	paymentMethod	CASCADE	CASCADE	A deletion/update of a paymentMethod should delete/update the user payment method.
userAddresses	user	CASCADE	CASCADE	A deletion/update of a user should delete/update the user addresses.
userAddresses	address	CASCADE	CASCADE	A deletion/update of an address should delete/update the user addresses.
userShopList	user	CASCADE	CASCADE	A deletion/update of a user should delete/update the user shop list.
userShopList	shoplist	CASCADE	CASCADE	A deletion/update of a shopping list should delete/update the user shop list.

Table 2: Implementation for Foreign Keys

Table	FK	ON DELETE	ON UPDATE	Comments
productInventory	product	RESTRICT	CASCADE	The deletion of a product should be restricted since the existence of inventory implies the existence of a product; deletion should only occur if there is no more product inventory. An update of a product should update the product inventory.
productInventory	inventory	CASCADE	CASCADE	A deletion/update of an inventory item should delete/update the product inventory.
productSuppliers	product	CASCADE	CASCADE	A deletion/update of a product should delete/update the product supplier.
productSuppliers	supplier	CASCADE	CASCADE	A deletion/update of a supplier should delete/update the product supplier.
productManufacturers	product	CASCADE	CASCADE	A deletion/update of a product should delete/update the product manufacturer.
productManufacturers	manufacturer	CASCADE	CASCADE	A deletion/update of a manufacturer should delete/update the product manufacturer.
productPromotion	product	CASCADE	CASCADE	A deletion/update of a product should delete/update the product promotion.
productPromotion	feature	CASCADE	CASCADE	A deletion/update of a feature should delete/update the product promotion.

Table 2: Implementation for Foreign Keys

Table	FK	ON DELETE	ON UPDATE	Comments
supportedFeature	accountType	CASCADE	CASCADE	A deletion/update of an account type should delete/update the supported feature.
supportedFeature	feature	CASCADE	CASCADE	A deletion/update of a feature should delete/update the supported feature.
account	user	CASCADE	CASCADE	A deletion/update of a user should delete/update the account.
account	accountType	RESTRICT	CASCADE	The deletion of an accountType should be restricted since such a deletion would delete many accounts. Deletion should only occur if there does not exist accounts of the account type. An update of an accountType should update the account.
session	user	CASCADE	CASCADE	A deletion/update of a user should delete/update the session.
email	user	CASCADE	CASCADE	A deletion/update of a user should delete/update the email.
userMailingLists	user	CASCADE	CASCADE	A deletion/update of a user should delete/update the user mailing lists.
userMailingLists	mailingList	CASCADE	CASCADE	A deletion/update of a mailing list should delete/update the user mailing lists.

Section VIII: Forward Engineering

Due to bugs in mySQL workbench, edits were made to the databasemodel.sql file which are outlined below.

Changed TINYINT to BOOLEAN data type:

1. user.is_Registered to BOOLEAN from TINYINT
2. userAddresses.is_default_shipping to BOOLEAN from TINYINT
3. userAddresses.is_default_billing to BOOLEAN from TINYINT
4. userShopList.is_shoppingcart to BOOLEAN from TINYINT
5. packagesForFulfillment.cust_pickup to BOOLEAN from TINYINT

Section IX: Inserting Data

This section is intentionally empty. No documentation necessary.

Section X: Testing

This section is intentionally empty. No documentation necessary.

Section XI: Testing Table

Table 3: Testing Table

Entity	SQLQuery	Pass/Fail	Error Description	Possible Solution
user	DELETE	Pass	None	None
user	UPDATE	Pass	None	None
address	DELETE	Pass	None	None
address	UPDATE	Pass	None	None
paymentMethod	DELETE	Pass	None	None
paymentMethod	UPDATE	Pass	None	None
shopList	DELETE	Pass	None	None
shopList	UPDATE	Pass	None	None
product	DELETE	Fail	FK constraint fails (restrict)	Need to create trigger or procedure to deal with this error
product	UPDATE	Pass	None	None
accountType	DELETE	Fail	FK constraint fails (restrict)	Need to create trigger or procedure to deal with this error
accountType	UPDATE	Pass	None	None
feature	DELETE	Pass	None	None
feature	UPDATE	Pass	None	None
category	DELETE	Pass	None	None
category	UPDATE	Pass	None	None
mailingList	DELETE	Pass	None	None
mailingList	UPDATE	Pass	None	None
fulfillmentAction	DELETE	Pass	None	None
fulfillmentAction	UPDATE	Pass	None	None
package	DELETE	Pass	None	None
package	UPDATE	Pass	None	None
courier	DELETE	Pass	None	None
courier	UPDATE	Pass	None	None
region	DELETE	Fail	FK constraint fails (restrict)	Need to create trigger or procedure to deal with this error
region	UPDATE	Pass	None	None
inventory	DELETE	Pass	None	None
inventory	UPDATE	Pass	None	None
supplier	DELETE	Pass	None	None
supplier	UPDATE	Pass	None	None

Table 3: Testing Table

Entity	SQLQuery	Pass/Fail	Error Description	Possible Solution
manufacturer	DELETE	Pass	None	None
manufacturer	UPDATE	Pass	None	None
productManufacturers	DELETE	Pass	None	None
productManufacturers	UPDATE	Pass	None	None
session	DELETE	Pass	None	None
session	UPDATE	Pass	None	None
userAddresses	DELETE	Pass	None	None
userAddresses	UPDATE	Pass	None	None
email	DELETE	Pass	None	None
email	UPDATE	Pass	None	None
userShopList	DELETE	Pass	None	None
userShopList	UPDATE	Pass	None	None
shoppingProductList	DELETE	Pass	None	None
shoppingProductList	UPDATE	Pass	None	None
orderDetails	DELETE	Pass	None	None
orderDetails	UPDATE	Pass	None	None
receipt	DELETE	Pass	None	None
receipt	UPDATE	Pass	None	None
account	DELETE	Pass	None	None
account	UPDATE	Pass	None	None
supportedFeature	DELETE	Pass	None	None
supportedFeature	UPDATE	Pass	None	None
userPaymentMethods	DELETE	Pass	None	None
userPaymentMethods	UPDATE	Pass	None	None
paymentInfo	DELETE	Pass	None	None
paymentInfo	UPDATE	Pass	None	None
creditCard	DELETE	Pass	None	None
creditCard	UPDATE	Pass	None	None
bankAccount	DELETE	Pass	None	None
bankAccount	UPDATE	Pass	None	None
productCategories	DELETE	Pass	None	None
productCategories	UPDATE	Pass	None	None
productPromotion	DELETE	Pass	None	None
productPromotion	UPDATE	Pass	None	None
userMailingLists	DELETE	Pass	None	None
userMailingLists	UPDATE	Pass	None	None
orderFulfillment	DELETE	Pass	None	None
orderFulfillment	UPDATE	Pass	None	None
shipment	DELETE	Pass	None	None

Table 3: Testing Table

Entity	SQLQuery	Pass/Fail	Error Description	Possible Solution
shipment	UPDATE	Pass	None	None
packagesForFulfillment	DELETE	Pass	None	None
packagesForFulfillment	UPDATE	Fail	FK constraint fails (set null)	Need to create trigger or procedure to deal with this error
actionOrigin	DELETE	Pass	None	None
actionOrigin	UPDATE	Pass	None	None
warehouse	DELETE	Pass	None	None
warehouse	UPDATE	Pass	None	None
productInventory	DELETE	Pass	None	None
productInventory	UPDATE	Pass	None	None
productSuppliers	DELETE	Pass	None	None
productSuppliers	UPDATE	Pass	None	None
warehouseInventoryRequest	DELETE	Pass	None	None
warehouseInventoryRequest	UPDATE	Pass	None	None
warehouseInventory	DELETE	Pass	None	None
warehouseInventory	UPDATE	Pass	None	None
productManufacturers	DELETE	Pass	None	None
productManufacturers	UPDATE	Pass	None	None