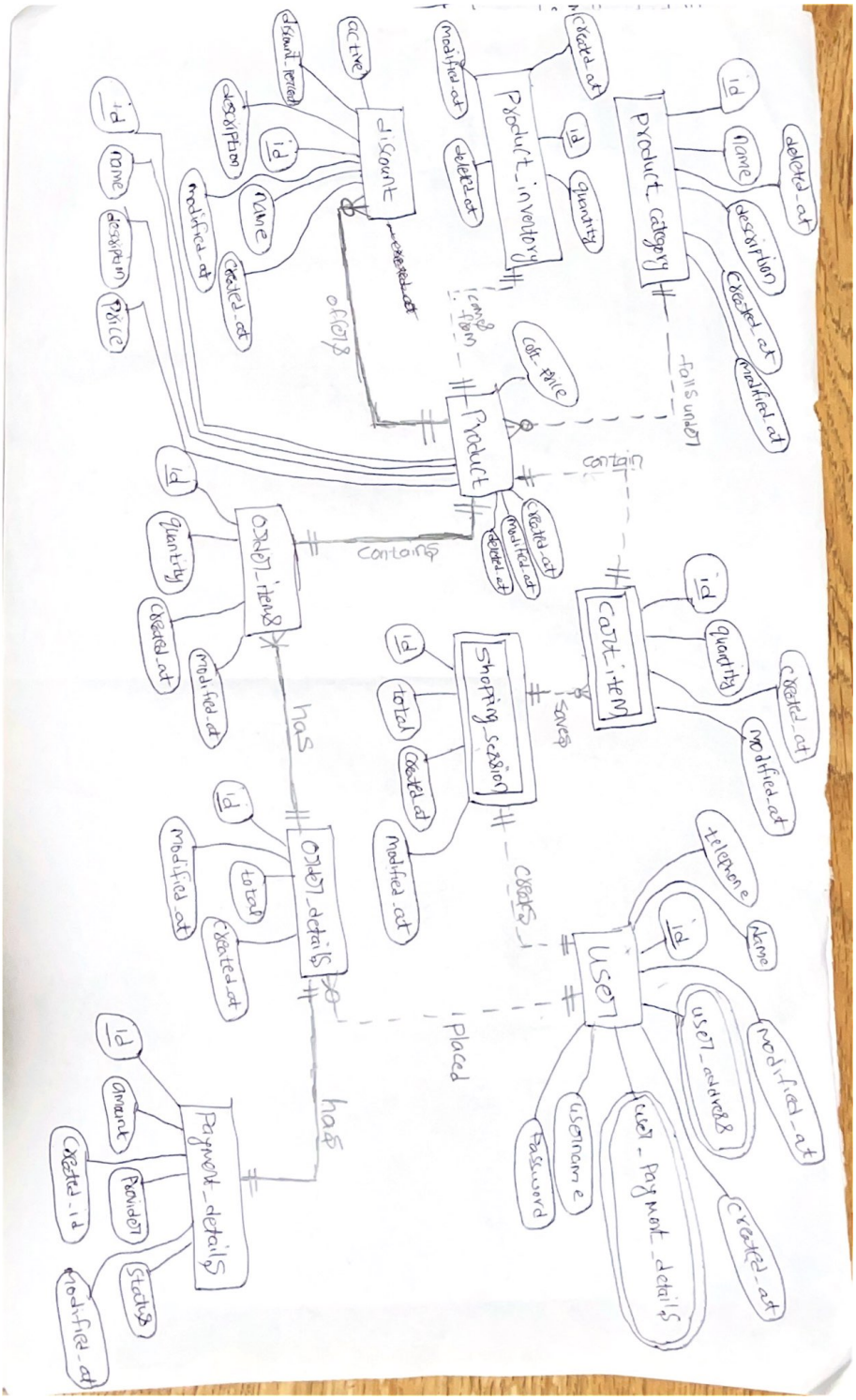
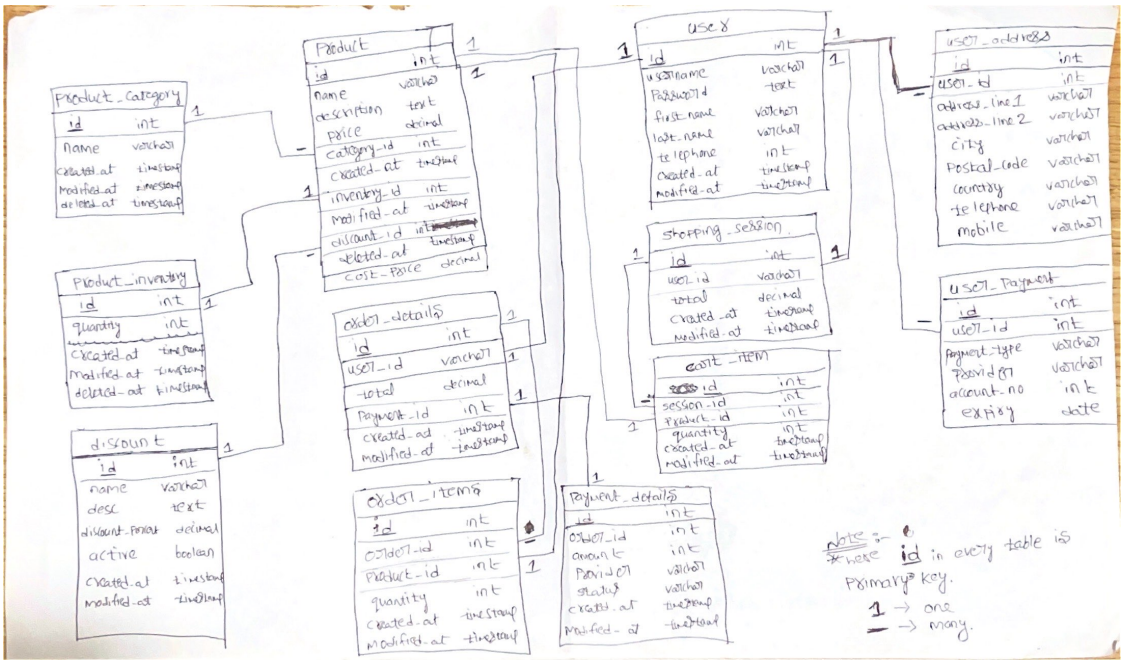
VIJAY MOHAN YEDDU(02008709) and SUNIL REDDY MARAM(02045584)

1. ER- Diagram



2. Relation-Schema:



We have created a user table that contains all the user details along with user_payment and user_address tables to store multiple addresses and payment details of users.

Managing products is not simply about maintaining a list of products. You also must manage the inventory, discounts, categories, and other attributes of the products.

shopping_session and cart_item as temporary data stores that only store the shopping session information of the current user until the order is confirmed and the data is moved to permanent storage tables with the payment details (order_details, order_items, and payment_details).

3) & 4)...FUNCTIONAL DEPENDECIES AND NORMALIZATION ON EACH TABLE:

First Normal Form (1NF)

This initial set of rules sets the fundamental guidelines for keeping your database properly organized.

- Remove any repeating groups of data *(i.e. beware of duplicative columns or rows within the same table)*
- Create separate tables for each group of related data
- Each table should have a primary key (*i.e. a field that identifies each row with a non-null, unique value*)

Second Normal Form (2NF)

This next set of rules builds upon those outlined in 1NF.

- Meet every rule from 1NF
- Remove data that doesn't depend on the table's primary key *(either move the data to the appropriate table or create a new table and primary key)*
- Foreign keys are used to identify table relationships

Third Normal Form (3NF)

This set of rules takes those outlined in 1NF and 2NF a step further.

- Meet every rule from 1NF and 2NF
- Remove attributes that rely on other non-key attributes *(i.e. remove columns that depend on columns that aren't foreign or primary keys)*

1.product_category

Product_category(id,name,created_at,modified_at,deleted_at)

Functional Dependencies:

ID → NAME, CREATED_AT, MODIFIED_AT, DELETED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

2.product_inventory

Product_inventory(id,quantity,created_at,modified_at,deleted_at)

Functional Dependencies:

ID → QUANTITY, CREATED_AT, MODIFIED_AT, DELETED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

3.discount

Discount (id, name, description, discount_percent ,active, created_at, modified_at)

Functional Dependencies:

ID → NAME,DESCRIPTION, DISCOUNT_PERCENT, ACTIVE, CREATED_AT, MODIFIED_AT,DELETED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

4.product

Product (id, name, description, price, category_id, inventory_id, discount_id, created_at, modified_at, deleted_at, cost_price);

Functional Dependencies:

ID → NAME, DESCRIPTION, PRICE, CATEGORY_ID, INVENTORY_ID, DISCOUNT_ID, CREATED_AT, MODIFIED_AT, DELETED_AT, COST_PRICE

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

5.user

User(id,username,password,firstname,lastname,telephone,created_at, modified_at)

Functional Dependencies:

ID → USERNAME, PASSWORD, FIRSTNAME, LASTNAME, TELEPHONE, CREATED_AT, MODIFIED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

6.user_address

User_address(id,user_id,address_line1, address_line2,city,postalcode,country,telephone,mobile);

Functional Dependencies:

ID → USER_ID, ADDRESS_LINE1, ADDRESS_LINE2, POSTALCODE, COUNTRY,CITY, TELEPHONE, MOBILE.

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

7.user_payment

User_payment(id,user_id,payment_type,provider,account_no,expiry)

Functional Dependencies:

ID → USER_ID, PAYMENT_TYPE, PROVIDER, ACCOUNT_NO, EXPIRY.

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

8.shopping session

Shopping_session(id,user_id,total,created_at,modified_at)

Functional Dependencies:

ID →USER_ID, TOTAL, CREATED_AT, MODIFIED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

9.cart_item

Cart_item(id,session_id,product_id,quantity,created_at,modified_at)

Functional Dependencies:

ID → ID, SESSION_ID, PRODUCT_ID, QUANTITY, CREATED_AT, MODIFIED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

10.order_details

Order_details(id,user_id,total,payment_id,created_at,modified_id)

Functional Dependencies:

ID → USER_ID, TOTAL_PAYMENT_ID, CREATED_AT, MODIFIED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

11.order_items

Order_items(id,order_id,product_id,quantity,created_at,modified_at);

Functional Dependencies:

ID → ORDER_ID, PRODUCT_ID, QUANTITY, CREATED_AT, MODIFIED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

12.payment_details

Payment_details(id,order_id,amount,provider,status,created_at,modified_at)

Functional Dependencies:

ID → ORDER_ID, AMOUNT, PROVIDER, STATUS, CREATED_AT, MODIFIED_AT

Normalization:

From the above normalization rules it satisfying 1NF, 2NF, 3NF.

5). Possible Queries:

1) Top 5 selling products & product categories during Thanksgiving Season i.e from (11/23/2022 – 11/28/2022)
2) Create a list of inventories available by product category and product level after the end of Thanksgiving Sales (To fill up for the Christmas season)
3) A query to calculate profit margin (Selling_price – (Cost_Price) – (Discount)) for Top 10 Product categories and their products
4) Top 3 Service Providers by Sales
5)Top 100 customers by sales and profit margin
6) What is the average session time per user and Top customer with high session time
7) Top products at user level (for top 100 customers) with high shopping session time but no conversion
8) Create a query to get the avg shopping sessions by cart value per customer
9) Create a query to get discount names and profit/loss generated for each discount
10) Create a query to get customer Rank based on profit generated

6). Queries that cannot be derived from the database

- Create a query to get conversion rate from session table to payment
- Distribution of order for user address
- Distribution of orders by user payment details