PROJECT - Online Retail Application Database

Reason for opting this database design:
Online shopping is everywhere these days, and I wanted to dig deeper into how things work behind the scenes. Understanding how the data is organized in the background - the tables, connections, and how everything fits together - is essential to see how an online store really runs. So I opted for this database design to get a better picture of how information like customer details, products, and purchases is handled. This hands-on exploration and research will help me to learn how all these pieces interact and make the online shopping experience smooth for users. Goal was to get a grasp on how user actions on the website connect to what happens in the database.

I have designed a database design for an online application that sells cosmetic products:

Process Flow Description

- **Account Creation** : Users will create an account on site and add an address. One user can have multiple addresses and at time of checkout can select which address should the order be shipped to.

- **Product Categories :** Site has hierarchical product categories. A product category can have further subcategories. Example : Skin Care has subcategories : Face Care, Hair Care, Body Care and these have further subcategories. Face care -> Skin Care, Lip Care,Eye Care. Users can choose from multiple products.

- **Checkout** : When a user proceeds to checkout, they select a delivery address from their saved addresses. Upon confirming the order, a new entry is created in the orders table. An order_id is generated at this stage. This order_id is then updated in the relevant rows of the cart table to link the order with the items in the cart.

- **Order items**: Once order is placed an entry is created for each item in this table with a unique id for each item for that specific order alongwith price at time of purchase and quantity. One item can have multiple quantities.

- **Payment Confirmation**: Once the user completes the payment, an entry is created in the payments table, which records the payment details related to that specific order.

DATABASE DESIGN

- **customers Table:**

Purpose: Stores customer information such as their names, email addresses, and phone numbers.
Primary Key: customer_id
Relationships: Connected to other tables (addresses,cart) through the customer_id field.

- **addresses Table:**

Purpose: Contains customer addresses with details like address lines, city, state, and postal code. One customer can have multiple addresses.
Primary Key: address_id
Relationships: Connected to orders table via address_id field to identify the shipping address for an order.

- **product_category Table:**

Purpose: Stores different categories of products and their sub-categories. Stores category name and parent_category_id
Primary Key: category_id
Relationships: parent_category_id acts as a foreign and references category_id in this tableforming self relation. Category_id acts as foreign key in products table

- **products Table:**

Purpose: Holds details about individual products like category, name, price, and quantity in stock.
Primary Key: product_id
Foreign Key Constraint: category_id associates each product with a specific category from the Product_category table.
Relationships: Connected to cart table via product_id

- **order_status Table:**

Purpose: Contains various order statuses (e.g., confirmed, processing, shipped, delivered etc).
Primary Key: order_status_id
Relationships: Connected to orders table via order_status_id

- **orders Table:**

Purpose: Stores order details, including customer ID, status, shipping address, and order date.
Primary Key: order_id
Foreign Key Constraints: Connects Addresses, and order_status tables via shipping_address_id, and order_status_id, respectively.

- **Order_items Table:**

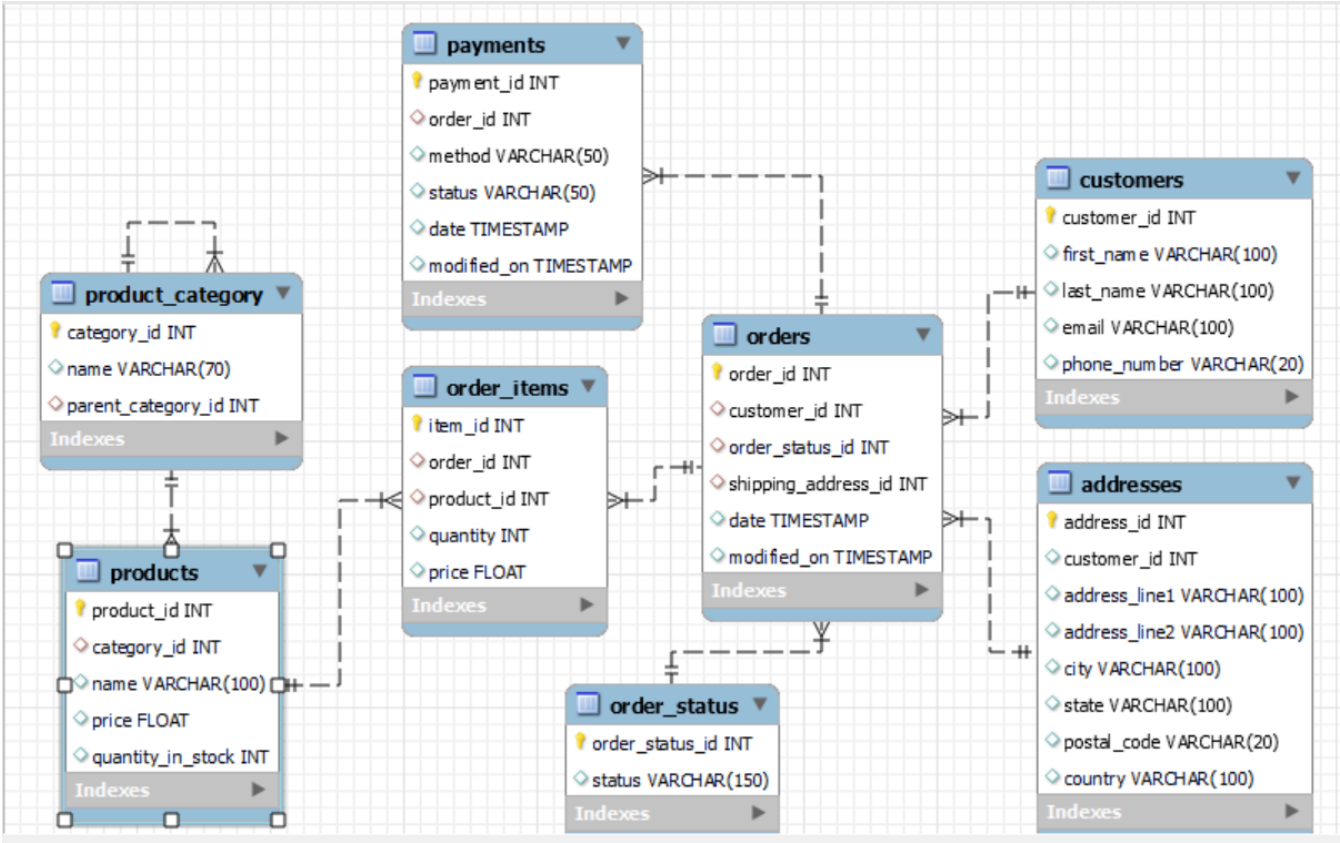
Purpose: Stores items ordered by user including quantity and price at time of purchase
Primary Key: item_id
Foreign Key: product_id,order_id acts as foreign key

- **payments Table:**

Purpose: Stores payment details associated with orders such as payment_method,status and date

Primary Key: payment_id
Foreign Key Constraint: Connects to the orders table through the order_id field.

EER Diagram



Relationships between tables

A **customer** can have **multiple addresses** saved in the system and at time of checkout can decide which address the specific order needs to be shipped to. Product category has a parent and child category forming a hierarchical relationship. One **product category** can have multiple **products** and each product has its name,price and quantity in stock. Users can place orders for products and each product can have multiple quantities. So an **order** can have multiple products and each product can be a part of many orders. **Order status** table is being used to keep track of orders and status id is being used as foreign key in orders table. **Payments** table has reference to order_id alongwith payment method, date and status. Amount field is not used as the amount can be calculated by multiplying the quantity of items and price from the order items table.