Entity Relationship Diagram - Document

Name of Database: **EcommerceDatabase**

Contributors:

Devansh Pratap Singh (055037)

Amartya Raj Singh (055053)

DESCRIPTION:

This database contains 13 tables.

TABLE DESCRIPTIONS ARE AS FOLLOWS:

1. Users Table:

- UserID (INT, PRIMARY KEY, AUTO_INCREMENT)
- Name (VARCHAR(100))
- Email (VARCHAR(100), UNIQUE)
- Phone (VARCHAR(15))
- Address (TEXT)
- CreatedAt (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)

2. Categories Table:

- CategoryID (INT, PRIMARY KEY, AUTO_INCREMENT)
- CategoryName (VARCHAR(100), UNIQUE)

3. Products Table:

- ProductID (INT, PRIMARY KEY, AUTO_INCREMENT)
- Name (VARCHAR(100))
- Description (TEXT)
- Price (DECIMAL(10,2))
- Stock (INT)
- CategoryID (INT, FOREIGN KEY referencing Categories(CategoryID))
- CreatedAt (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)

4. Orders Table:

- OrderID (INT, PRIMARY KEY, AUTO_INCREMENT)

- UserID (INT, FOREIGN KEY referencing Users(UserID))
- OrderDate (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
- Status (ENUM('Pending', 'Shipped', 'Delivered', 'Cancelled'), DEFAULT 'Pending')
- TotalAmount (DECIMAL(10,2))

5. OrderDetails Table:
   - OrderDetailID (INT, PRIMARY KEY, AUTO_INCREMENT)
   - OrderID (INT, FOREIGN KEY referencing Orders(OrderID))
   - ProductID (INT, FOREIGN KEY referencing Products(ProductID))
   - Quantity (INT)
   - Subtotal (DECIMAL(10,2))
6. Payments Table:
   - PaymentID (INT, PRIMARY KEY, AUTO_INCREMENT)
   - OrderID (INT, UNIQUE, FOREIGN KEY referencing Orders(OrderID))
   - PaymentMethod (ENUM('Credit Card', 'Debit Card', 'PayPal', 'Bank Transfer'), NOT NULL)
   - PaymentStatus (ENUM('Pending', 'Completed', 'Failed'), DEFAULT 'Pending')
   - PaymentDate (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
   - Amount (DECIMAL(10,2))
7. Reviews Table:
   - ReviewID (INT, PRIMARY KEY, AUTO_INCREMENT)
   - UserID (INT, FOREIGN KEY referencing Users(UserID))
   - ProductID (INT, FOREIGN KEY referencing Products(ProductID))
   - Rating (INT, CHECK (Rating BETWEEN 1 AND 5))
   - Comment (TEXT)
   - CreatedAt (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
8. Coupons Table:
   - CouponID (INT, PRIMARY KEY, AUTO_INCREMENT)
   - Code (VARCHAR(50), UNIQUE)
   - Discount (DECIMAL(5,2))
   - ExpirationDate (DATE)
9. Wishlist Table:
   - WishlistID (INT, PRIMARY KEY, AUTO_INCREMENT)

- UserID (INT, FOREIGN KEY referencing Users(UserID))
- ProductID (INT, FOREIGN KEY referencing Products(ProductID))
- CreatedAt (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)

10. Shipping Table:
    - ShippingID (INT, PRIMARY KEY, AUTO_INCREMENT)
    - OrderID (INT, FOREIGN KEY referencing Orders(OrderID))
    - ShippingAddress (TEXT)
    - TrackingNumber (VARCHAR(50))
    - Status (ENUM('Pending', 'Shipped', 'Delivered'), DEFAULT 'Pending')
    - EstimatedDeliveryDate (DATE)
11. Cart Table:
    - CartID (INT, PRIMARY KEY, AUTO_INCREMENT)
    - UserID (INT, FOREIGN KEY referencing Users(UserID))
    - ProductID (INT, FOREIGN KEY referencing Products(ProductID))
    - Quantity (INT)
    - CreatedAt (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
12. Admins Table:
    - AdminID (INT, PRIMARY KEY, AUTO_INCREMENT)
    - Name (VARCHAR(100))
    - Email (VARCHAR(100), UNIQUE)
    - PasswordHash (VARCHAR(255))
    - Role (ENUM('SuperAdmin', 'Manager', 'Support'), DEFAULT 'Manager')
    - CreatedAt (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
13. SupportTickets Table:
    - TicketID (INT, PRIMARY KEY, AUTO_INCREMENT)
    - UserID (INT, FOREIGN KEY referencing Users(UserID))
    - Issue (TEXT)
    - Status (ENUM('Open', 'In Progress', 'Resolved'), DEFAULT 'Open')
    - CreatedAt (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
14. Refunds Table:
    - RefundID (INT, PRIMARY KEY, AUTO_INCREMENT)

- OrderID (INT, FOREIGN KEY referencing Orders(OrderID))
- Reason (TEXT)
- Status (ENUM('Pending', 'Approved', 'Rejected'), DEFAULT 'Pending')
- RefundAmount (DECIMAL(10,2))
- CreatedAt (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)

Relationship descriptions

Understood! Here's the relationship description for your EcommerceDatabase tables (without the Coupons table), in the requested format:

1. Products and Categories:

- Relationship: One-to-many
- Cardinality: 1:N
    - Products: 1 (Each product must belong to one category)
    - Categories: 0..* (A category can have zero or many products)
- Description: Each product is associated with one specific category. One category can be associated with multiple products. The CategoryID in the Products table is a foreign key referencing the CategoryID in the Categories table.

2. Orders and Users:

- Relationship: One-to-many
- Cardinality: 1:N
    - Orders: 1 (Each order is placed by one user)
    - Users: 0..* (A user can place zero or many orders)
- Description: Each order is placed by one specific user. One user can place multiple orders. The UserID in the Orders table is a foreign key referencing the UserID in the Users table.

3. OrderDetails and Orders:

- Relationship: One-to-many
- Cardinality: 1:N
    - OrderDetails: 0..* (An order can have zero or many order details)
    - Orders: 1 (Each order detail belongs to one order)

- Description: Each order detail is associated with one specific order. One order can have multiple order details. The OrderID in the OrderDetails table is a foreign key referencing the OrderID in the Orders table.

4. OrderDetails and Products:

- Relationship: One-to-many
- Cardinality: 1:N
    - OrderDetails: 0..* (An order detail can be for zero or one product)
    - Products: 0..* (A product can be in zero or many order details)
- Description: Each order detail is associated with one specific product. One product can be included in multiple order details. The ProductID in the OrderDetails table is a foreign key referencing the ProductID in the Products table.

5. Payments and Orders:

- Relationship: One-to-one or One-to-zero-or-one
- Cardinality: 1:1 or 1:0..1
    - Payments: 0..1 (A payment is associated with zero or one order)
    - Orders: 1 (Each order has one payment)
- Description: Each payment is associated with one specific order. One order has one payment. The OrderID in the Payments table is a foreign key referencing the OrderID in the Orders table. The OrderId is also unique in the payment table.

6. Reviews and Users:

- Relationship: One-to-many
- Cardinality: 1:N
    - Reviews: 0..* (A user can write zero or many reviews)
    - Users: 1 (Each review is written by one user)
- Description: Each review is written by one specific user. One user can write multiple reviews. The UserID in the Reviews table is a foreign key referencing the UserID in the Users table.

7. Reviews and Products:

- Relationship: One-to-many
- Cardinality: 1:N
    - Reviews: 0..* (A product can have zero or many reviews)
    - Products: 1 (Each review is for one product)

- Description: Each review is for one specific product. One product can have multiple reviews. The ProductID in the Reviews table is a foreign key referencing the ProductID in the Products table.

8. Wishlist and Users:

- Relationship: One-to-many
- Cardinality: 1:N
    - Wishlist: 0..* (A user can have zero or many wishlist items)
    - Users: 1 (Each wishlist item belongs to one user)
- Description: Each wishlist item belongs to one specific user. One user can have multiple wishlist items. The UserID in the Wishlist table is a foreign key referencing the UserID in the Users table.

9. Wishlist and Products:

- Relationship: One-to-many
- Cardinality: 1:N
    - Wishlist: 0..* (A product can be in zero or many wishlists)
    - Products: 1 (Each wishlist item is for one product)
- Description: Each wishlist item is for one specific product. One product can be in multiple wishlists. The ProductID in the Wishlist table is a foreign key referencing the ProductID in the Products table.

10. Shipping and Orders:

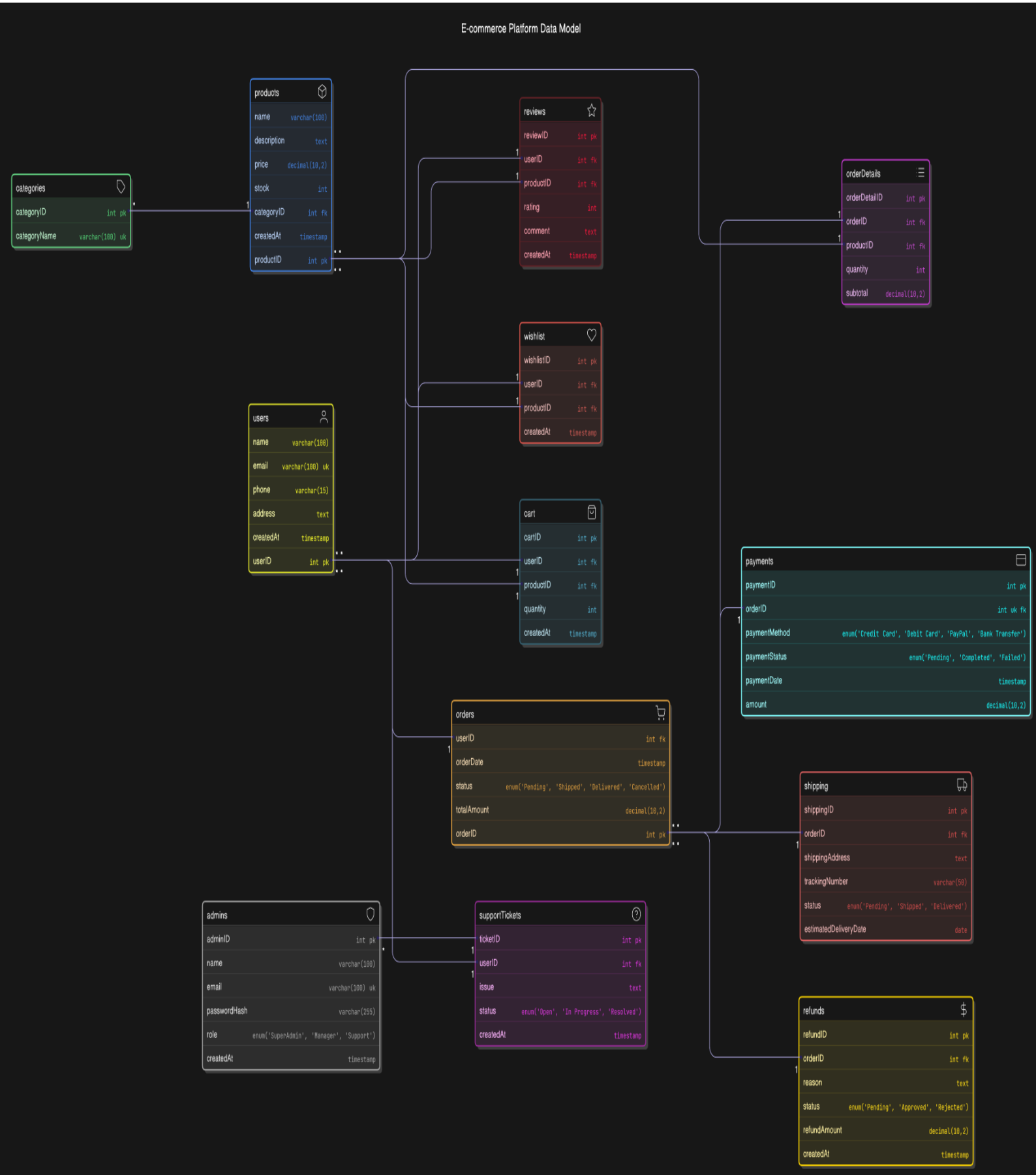
- Relationship: One-to-one or One-to-zero-or-one
- Cardinality: 1:1 or 1:0..1
    - Shipping: 0..1 (A shipping record is associated with zero or one order)
    - Orders: 1 (Each order has one shipping record)
- Description: Each shipping record is associated with one specific order. One order has one shipping record. The OrderID in the Shipping table is a foreign key referencing the OrderID in the Orders table.

11. Cart and Users:

- Relationship: One-to-many
- Cardinality: 1:N
    - Cart: 0..* (A user can have zero or many items in the cart)
    - Users: 1 (Each cart item belongs to one user)

- Description: Each cart item belongs to one specific user. One user can have multiple items in the cart. The UserID in the Cart table is a foreign key referencing the UserID in the Users table.

12. Cart and Products:
    - Relationship: One-to-many
    - Cardinality: 1:N
        - Cart: 0..* (A product can be in zero or many carts)
        - Products: 1 (Each cart item is for one product)
    - Description: Each cart item is for one specific product. One product can be in multiple carts. The ProductID in the Cart table is a foreign key referencing the ProductID in the Products table.
13. SupportTickets and Users:
    - Relationship: One-to-many
    - Cardinality: 1:N
        - SupportTickets: 0..* (A user can open zero or many support tickets)
        - Users: 1 (Each support ticket is opened by one user)
    - Description: Each support ticket is opened by one specific user. One user can open multiple support tickets. The UserID in the SupportTickets table is a foreign key referencing the UserID in the Users table.
14. Refunds and Orders:
    - Relationship: One-to-one or One-to-zero-or-one
    - Cardinality: 1:1 or 1:0..1
        - Refunds: 0..1 (A refund record is associated with zero or one order)
        - Orders: 1 (Each order has zero or one refund record)
    - Description: Each refund record is associated with one specific order. One order can have zero or one refund record. The OrderID in the Refunds table is a foreign key referencing the OrderID in the Orders table.


E-commerce Platform Data Model
categories
categoryID int pk
categoryName varchar(100) uk
products
name varchar(100)
description text
price decimal(10,2)
stock int
categoryID int fk
createdAt timestamp
productID int pk
reviews
reviewID int pk
userID int fk
productID int fk
rating int
comment text
createdAt timestamp
orderDetails
orderDetailID int pk
orderID int fk
productID int fk
quantity int
subtotal decimal(10,2)
wishlist
wishlistID int pk
userID int fk
productID int fk
createdAt timestamp
users
name varchar(100)
email varchar(100) uk
phone varchar(15)
address text
createdAt timestamp
userID int pk
cart
cartID int pk
userID int fk
productID int fk
quantity int
createdAt timestamp
payments
paymentID int pk
orderID int uk fk
paymentMethod enum('Credit Card', 'Debit Card', 'PayPal', 'Bank Transfer')
paymentStatus enum('Pending', 'Completed', 'Failed')
paymentDate timestamp
amount decimal(10,2)
orders
userID int fk
orderDate timestamp
status enum('Pending', 'Shipped', 'Delivered', 'Cancelled')
totalAmount decimal(10,2)
orderID int pk
shipping
shippingID int pk
orderID int fk
shippingAddress text
trackingNumber varchar(50)
status enum('Pending', 'Shipped', 'Delivered')
estimatedDeliveryDate date
admins
adminID int pk
name varchar(100)
email varchar(100) uk
passwordHash varchar(255)
role enum('SuperAdmin', 'Manager', 'Support')
createdAt timestamp
supportTickets
ticketID int pk
userID int fk
issue text
status enum('Open', 'In Progress', 'Resolved')
createdAt timestamp
refunds
refundID int pk
orderID int fk
reason text
status enum('Pending', 'Approved', 'Rejected')
refundAmount decimal(10,2)
createdAt timestamp