



Online Retail Store Database

GROUP MEMBERS:

جمانه المزید
وصایف السلامہ
جنی الريس

CONTENTS

Introduction

Tables

Insert

Queries

ER Model

Thank you !



Introduction

This project presents the design of a database for an online store.

The goal is to organize and manage essential data such as customers, products, orders, and payments.

A well-structured database helps the store operate efficiently, improves data accuracy, and supports smooth online shopping experiences.



TABLES

01

PRODUCT

02

CARTS ITEM

03

CUSTOMER

04

CATEGORY

05

ORDER ITEM

06

ORDER

07

CARTS

08

SHIPMENTS

PRODUCT TABLE:



```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    Description TEXT,
    Price DECIMAL(10, 2) NOT NULL CHECK (Price >= 0),
    StockQuantity INT NOT NULL CHECK (StockQuantity >= 0),
    CategoryID INT NOT NULL,
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID));
```

CARTITEM TABLE:



```
CREATE TABLE CartItems(
    CartItemID INT PRIMARY KEY AUTO_INCREMENT,
    CartID INT NOT NULL,
    ProductID INT NOT NULL,
    Quantity INT NOT NULL CHECK (Quantity > 0), --Ensures a product is only listed once in a given cart
    UNIQUE KEY (CartID, ProductID),
    FOREIGN KEY (CartID) REFERENCES Carts(CartID) ON DELETE CASCADE,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID));
```

CUSTOMER TABLE:



```
CREATE TABLE Customers ( CustomerID INT PRIMARY KEY AUTO_INCREMENT,  
FirstName VARCHAR(100) NOT NULL, LastName VARCHAR(100) NOT NULL, Email  
VARCHAR(255) NOT NULL UNIQUE, PasswordHash VARCHAR(255) NOT NULL, --  
Stores the secure hash of the password Address VARCHAR(255), City  
VARCHAR(100), Country VARCHAR(100));
```

CATEGORY TABLE:

```
CREATE TABLE Categories ( CategoryID INT  
PRIMARY KEY AUTO_INCREMENT,  
CategoryName VARCHAR(100) NOT NULL UNIQUE,  
Description TEXT);
```



ORDER ITEM TABLE:

```
CREATE TABLE OrderItems(
```

```
    OrderItemID INT PRIMARY KEY AUTO_INCREMENT, OrderID INT NOT NULL,  
    ProductID INT NOT NULL,  
    Quantity INT NOT NULL CHECK (Quantity > 0),  
    UnitPrice DECIMAL(10, 2) NOT NULL, --Price at the time of purchase  
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID));
```



ORDER TABLE:

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY AUTO_INCREMENT,  
    CustomerID INT NOT NULL,  
    OrderDate DATETIME DEFAULT CURRENT_TIMESTAMP,  
    TotalAmount DECIMAL(10, 2) NOT NULL CHECK (TotalAmount >= 0),  
    OrderStatus VARCHAR(50) NOT NULL, --e.g., 'Pending', 'Processing', 'Shipped', 'Delivered'  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));
```





CARTS TABLE:

```
CREATETABLE Carts (
    CartID INT PRIMARY KEY AUTO_INCREMENT,
    CustomerID INT NOT NULL UNIQUE, --Only one active cart per customer
    DateCreated DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE);
```



SHIPMENTS TABLE:

```
CREATETABLE Shipments (
    ShipmentID INT PRIMARY KEY AUTO_INCREMENT,
    OrderID INT NOT NULL UNIQUE, --One shipment per order
    ShipmentDate DATETIME,
    TrackingNumber VARCHAR(100) UNIQUE,
    Shipper VARCHAR(100),
    DeliveryStatus VARCHAR(50) NOT NULL, --e.g., 'In Transit', 'Out for Delivery',
    'Delivered'
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE);
```

Insert





Insert code

```
INSERT INTO Categories (CategoryName, Description) VALUES  
('Electronics', 'Devices like phones, laptops, and accessories.'),  
('Apparel', 'Clothing, shoes, and fashion accessories.'),  
('Home Goods', 'Furniture, kitchenware, and decor.');
```

```
INSERT INTO Customers (FirstName, LastName, Email,  
PasswordHash, Address, City, Country) VALUES  
('Alice', 'Johnson', 'alice.j@example.com', 'secure_hash_for_alice',  
'123 Main St', 'Anytown', 'USA'),  
('Bob', 'Smith', 'bob.s@example.com', 'secure_hash_for_bob',  
'456 Oak Ave', 'Otherville', 'Canada');
```





Insert code

```
INSERT INTO Products (Name, Description, Price, StockQuantity, CategoryID) VALUES
('Smartphone X', 'Latest model with high-resolution camera.', 799.99, 50, 1),
('Wireless Headphones', 'Noise-cancelling over-ear headphones.', 149.50, 120, 1),
('T-Shirt -Blue', '100% Cotton, Large size.', 25.00, 200, 2),
('Coffee Maker', '12-cup automatic programmable coffee maker.', 45.99, 80, 3);
```

--Alice's first order (Phone X + Headphones)

```
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount, OrderStatus) VALUES(1, NOW(), 949.49, 'Processing');
```





Insert code

```
--Alice's second order(BlueT-Shirt) INSERT INTO Orders  
(CustomerID, OrderDate, TotalAmount, OrderStatus) VALUES (1,  
NOW(), 25.00, 'Processing');
```

```
INSERT INTO OrderItems(OrderID, ProductID, Quantity,  
UnitPrice) VALUES  
(1, 1, 1, 799.99), --1x Smartphone X for Order 1  
(1, 2, 1, 149.50), --1x Wireless Headphones for Order 1  
(2, 3, 1, 25.00); --1x T-Shirt -Blue for Order 2
```





Insert code

```
INSERT INTO Shipments (OrderID, ShipmentDate,  
TrackingNumber, Shipper, DeliveryStatus) VALUES  
(1, DATE_ADD(NOW(), INTERVAL 1 DAY), 'TRK123456789',  
'SwiftLogistics', 'In Transit');
```

```
INSERT INTO Carts (CustomerID, DateCreated) VALUES  
(2, NOW());
```

```
INSERT INTO CartItems(CartID, ProductID, Quantity) VALUES  
(1, 4, 1);
```



Insert result

ProductID	Name	Description	Price	StockQuantity	CategoryID
1	Smartphone X	Latest model with high-resolution camera.	799.99	50	1
2	Wireless Headphones	Noise-cancelling over-ear headphones.	149.50	120	1
3	T-Shirt - Blue	100% Cotton, Large size.	25.00	200	2
4	Coffee Maker	12-cup automatic programmable coffee maker.	45.99	80	3

CustomerID	FirstName	LastName	Email	PasswordHash	Address	City	Country
1	Alice	Johnson	alice.j@example.com	secure_hash_for_alice	123 Main St	Anytown	USA
2	Bob	Smith	bob.s@example.com	secure_hash_for_bob	456 Oak Ave	Otherville	Canada

CategoryID	CategoryName	Description
1	Electronics	Devices like phones, laptops, and accessories.
2	Apparel	Clothing, shoes, and fashion accessories.
3	Home Goods	Furniture, kitchenware, and decor.

OrderID	CustomerID	OrderDate	TotalAmount	OrderStatus
1	1	2025-11-02 17:26:11	949.49	Processing
2	1	2025-11-02 17:26:11	25.00	Processing





Insert result

OrderItemID	OrderID	ProductID	Quantity	UnitPrice
1	1	1	1	799.99
2	1	2	1	149.50
3	2	3	1	25.00

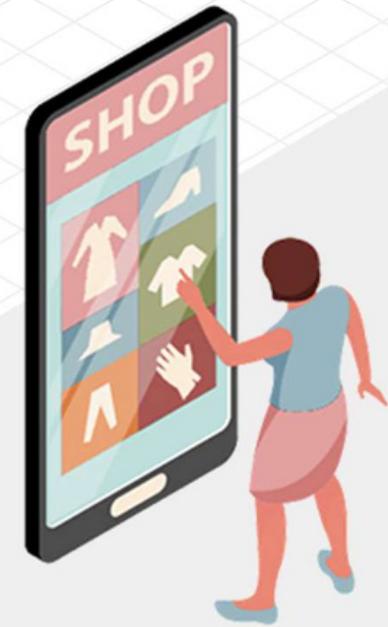
ShipmentID	OrderID	ShipmentDate	TrackingNumber	Shipper	DeliveryStatus
1	1	2025-11-03 17:26:11	TRK123456789	SwiftLogistics	In Transit

CartItemID	CartID	ProductID	Quantity
1	1	4	1

CartID	CustomerID
1	2



Queries



1) Retrieve all products with their category names

This query joins the Products and Categories tables to show what category each product belongs to.

```
SELECT
    p.ProductID,
    p.NameAS ProductName,
    c.CategoryName,
    p.Price,
    p.StockQuantity
FROM Products p
JOIN Categories c ON p.CategoryID= c.CategoryID;
```

ProductID	ProductName	CategoryName	Price	StockQuantity
1	Smartphone X	Electronics	799.99	50
2	Wireless Headphones	Electronics	149.50	120
3	T-Shirt - Blue	Apparel	25.00	200
4	Coffee Maker	Home Goods	45.99	80

2) List all orders placed by a specific customer (e.g., Alice Johnson)

This joins Orders and Customers to show Alice's orders.

```
SELECT  
    o.OrderID,  
    o.OrderDate,  
    o.TotalAmount,  
    o.OrderStatus  
FROM Orders o  
JOIN Customers c ON o.CustomerID= c.CustomerID  
WHERE c.FirstName= 'Alice' AND c.LastName= 'Johnson';
```

OrderID	OrderDate	TotalAmount	OrderStatus
1	2025-11-02 17:26:11	949.49	Processing
2	2025-11-02 17:26:11	25.00	Processing

3) Get detailed order information including products in each order

This query combines Orders, OrderItems, and Products to display the contents of each order.

```
SELECT
    o.OrderID,
    p.Name AS ProductName,
    oi.Quantity,
    oi.UnitPrice,
    (oi.Quantity * oi.UnitPrice) AS Subtotal
FROM OrderItem oi
JOIN Orders o ON oi.OrderID = o.OrderID
JOIN Products p ON oi.ProductID = p.ProductID
ORDER BY o.OrderID;
```

OrderID	ProductName	Quantity	UnitPrice	Subtotal
1	Smartphone X	1	799.99	799.99
1	Wireless Headphones	1	149.50	149.50
2	T-Shirt - Blue	1	25.00	25.00

4) Show all shipments and their delivery status

This query retrieves order shipment details.

```
SELECT
    s.ShipmentID,
    s.TrackingNumber,
    s.Shipper,
    s.DeliveryStatus,
    o.OrderID,
    c.FirstName AS CustomerName
FROM Shipments s
JOIN Orders o ON s.OrderID = o.OrderID
JOIN Customers c ON o.CustomerID = c.CustomerID;
```

ShipmentID	TrackingNumber	Shipper	DeliveryStatus	OrderID	CustomerName
1	TRK123456789	SwiftLogistics	In Transit	1	Alice

5. Display products currently in a customer's cart (e.g., Bob Smith)

This query finds the items in Bob's active cart.

```
SELECT cu.FirstName, cu.LastName,  
p.NameAS ProductName,  
ci.Quantity,  
p.Price,  
(ci.Quantity* p.Price) AS TotalPriceFROM CartItems ci  
JOIN Carts ca ON ci.CartID= ca.CartID  
JOIN Customers cu ON ca.CustomerID= cu.CustomerID  
JOIN Products p ON ci.ProductID= p.ProductID  
WHERE cu.FirstName= 'Bob' AND cu.LastName= 'Smith';
```

FirstName	LastName	ProductName	Quantity	Price	TotalPrice
Bob	Smith	Coffee Maker	1	45.99	45.99

6. Find the total number of orders made by each customer

Useful to see how active each customer is.

```
SELECT  
c.CustomerID,  
CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,  
COUNT(o.OrderID) AS TotalOrders  
FROM Customers c  
LEFT JOIN Orders o ON c.CustomerID= o.CustomerID  
GROUP BY c.CustomerID;
```

CustomerID	CustomerName	TotalOrders
1	Alice Johnson	2
2	Bob Smith	0

7) Display products with low stock

Great for inventory monitoring.

```
SELECT  
ProductID,  
Name AS ProductName,  
StockQuantity  
FROM Products  
WHERE StockQuantity < 100;
```

ProductID	ProductName	StockQuantity
1	Smartphone X	50
4	Coffee Maker	80

8. Calculate total revenue generated from each product

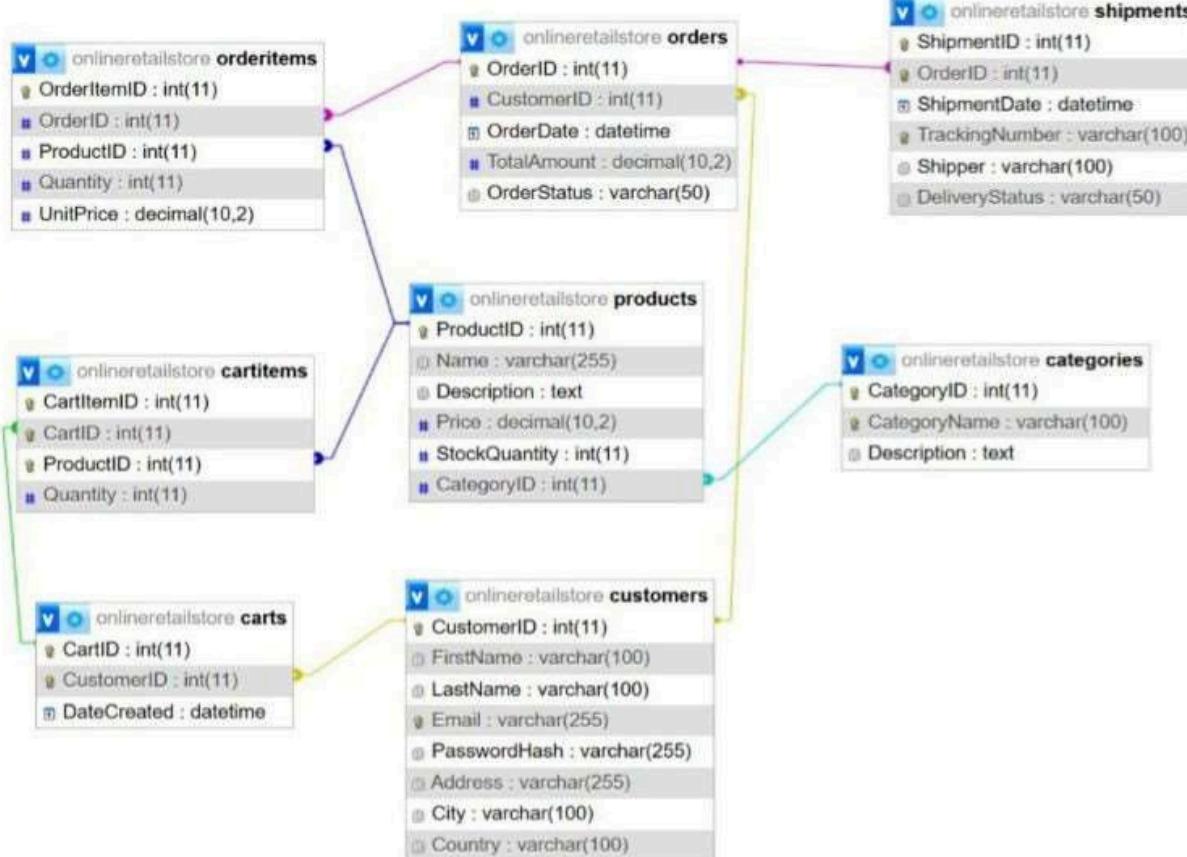
This uses OrderItems to compute sales revenue per product.

```
SELECT  
    p.ProductID,  
    p.Name AS ProductName,  
    SUM(oi.Quantity * oi.UnitPrice) AS TotalRevenue  
FROM Products p  
JOIN OrderItem oi ON p.ProductID = oi.ProductID  
GROUP BY p.ProductID;
```

ProductID	ProductName	TotalRevenue
1	Smartphone X	799.99
2	Wireless Headphones	149.50
3	T-Shirt - Blue	25.00

ER Model





THANK YOU!

