

Online Retail Database System

Name: Sakshi Vishwas Paralekar

Roll No: 459

College: Patkar Varde College, Mumbai

Date: 02-06-2025

Project Overview

To design and implement a relational database for a small online retail store that tracks customers, products, orders, and payments.

ER Diagram (Text Representation)

[Customers]		[Products]		[Orders]	[Order Items]		[Payment]
customer_id(PK)		product_id(PK)		order_id(PK)	order_item_id(PK)		payment_id(PK)
name		name		customer_id (FK)	order_id (FK)		order_id (FK)
email		category		order_date	product_id (FK)		amount
phone number		price			quantity		payment_date
city		stock					payment_method

Table Creation SQL

-- 1. Customers Table

```
CREATE TABLE Customer (  
    customer_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100),  
    email VARCHAR(100),  
    phone VARCHAR(15),
```

```
    city VARCHAR(50)
);

-- 2. Products Table

CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    category VARCHAR(50),
    price DECIMAL(10, 2),
    stock INT
);

-- 3. Orders Table

CREATE TABLE Orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    order_date DATE,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);

-- 4. Order_Items Table

CREATE TABLE Order_items (
    order_item_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT,
    product_id INT,
    quantity INT,
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);

-- 5. Payments Table

CREATE TABLE Payments (
```

```
payment_id INT PRIMARY KEY AUTO_INCREMENT,  
order_id INT,  
amount DECIMAL(10,2),  
payment_date DATE,  
payment_method VARCHAR(50),  
FOREIGN KEY (order_id) REFERENCES Orders(order_id)  
);
```

SQL Queries

-- 1. List all customers from Mumbai

```
SELECT * FROM Customers WHERE city = 'Mumbai';
```

-- 2. Show all orders with product names

```
SELECT o.order_id, c.name AS customer, p.name AS product, oi.quantity  
FROM Orders o  
JOIN Customers c ON o.customer_id = c.customer_id  
JOIN Order_Items oi ON o.order_id = oi.order_id  
JOIN Products p ON oi.product_id = p.product_id;
```

-- 3. Total sales per product

```
SELECT p.name, SUM(oi.quantity * p.price) AS total_sales  
FROM Order_Items oi  
JOIN Products p ON oi.product_id = p.product_id  
GROUP BY p.name;
```

-- 4. Top customer by total amount spent

```
SELECT c.name, SUM(p.amount) AS total_spent  
FROM Payments p
```

```
JOIN Orders o ON p.order_id = o.order_id
JOIN Customers c ON o.customer_id = c.customer_id
GROUP BY c.name
ORDER BY total_spent DESC
LIMIT 1;
```

Conclusion

The ER diagram models an online retail system where customers place orders containing products, each order has related order items and a single payment, ensuring efficient data organization and transaction tracking.