* **CUSTOMER TABLE-**

CREATE TABLE Customers (customer\_id INT PRIMARY KEY, name VARCHAR(100), email VARCHAR(255), phone\_number VARCHAR(20), address\_id INT, FOREIGN KEY (address\_id) REFERENCES Addresses(address\_id) );

INSERT INTO Customers (customer\_id, name, email, phone\_number, address\_id)

VALUES

(101, 'John Smith', 'john.smith@example.com', '+1 (555) 123-4567', 201),

(102, 'Sarah Johnson', 'sarah.johnson@emailprovider.com', '+1 (555) 987-6543', 202),

(103, 'Michael Lee', 'mlee@gmail.com', '+1 (555) 222-3333', 203),

(104, 'Emily Chen', 'emily\_chen@email.com', '+1 (555) 777-8888', 204),

(105, 'Maria Rodriguez', 'marodriguez@emailprovider.com', '+1 (555) 444-5555', 205);

CREATE TABLE Product (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(255),

stock INT,

category\_id INT,

MRP DECIMAL(10, 2),

Brand VARCHAR(100)

);

INSERT INTO Product (product\_id, product\_name, stock, category\_id, MRP, Brand)

VALUES

(501, 'Laptop X1', 10, 701, 1299.99, 'TechMaster'),

(502, 'Smartphone S2', 20, 702, 899.00, 'ElectroTech'),

(503, 'TV T3', 5, 704, 1499.99, 'ViewSonic'),

(504, 'Earbuds E4', 30, 706, 249.00, 'AudioTech'),

(505, 'Running Shoes R5', 15, 707, 199.95, 'FitFoot');

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

shipping\_date DATE,

cart\_id INT,

order\_amount DECIMAL(10, 2),

order\_date DATE

);

INSERT INTO Orders (order\_id, shipping\_date, cart\_id, order\_amount, order\_date)

VALUES

(1001, '2024-03-20', 501, 1299.99, '2024-03-15'),

(1002, '2024-03-21', 502, 899.00, '2024-03-16'),

(1003, '2024-03-22', 503, 1499.99, '2024-03-17'),

(1004, '2024-03-23', 504, 498.00, '2024-03-18'),

(1005, '2024-03-24', 505, 199.95, '2024-03-19');

CREATE TABLE OrderItem (

order\_item\_id INT PRIMARY KEY,

order\_id INT,

product\_id INT,

quantity INT,

unit\_price DECIMAL(10, 2),

total\_price DECIMAL(10, 2)

);

INSERT INTO order\_item (shipping\_date, order\_id, cart\_id, order\_amount, order\_date)

VALUES

('2024-03-15', 1101, 201, 49.99, '2024-03-10'),

('2024-03-16', 102, 202, 29.99, '2024-03-11'),

('2024-03-17', 103, 203, 99.99, '2024-03-12'),

('2024-03-18', 104, 204, 79.99, '2024-03-13'),

('2024-03-19', 105, 205, 39.99, '2024-03-14');

CREATE TABLE payment (

customer\_id INT,

order\_id INT,

payment\_id INT,

payment\_mode VARCHAR(50),

payment\_date DATE

);

INSERT INTO payment (customer\_id, order\_id, payment\_id, payment\_mode, payment\_date)

VALUES

(101, 201, 301, 'Credit Card', '2024-03-15'),

(102, 202, 302, 'PayPal', '2024-03-16'),

(103, 203, 303, 'Cash On Delivery', '2024-03-17'),

(104, 204, 304, 'Debit Card', '2024-03-18'),

(105, 205, 305, 'Credit Card', '2024-03-19');

CREATE TABLE payment\_history (

payment\_history\_id INT,

payment\_id INT,

data VARCHAR(255),

status VARCHAR(50),

transaction\_id VARCHAR(100)

);

INSERT INTO payment\_history (payment\_history\_id, payment\_id, data, status, transaction\_id)

VALUES

(1, 301, 'Payment successful', 'Completed', 'TRX001'),

(2, 302, 'Payment pending verification', 'Pending', 'TRX002'),

(3, 303, 'Payment failed', 'Failed', 'TRX003'),

(4, 304, 'Payment refunded', 'Refunded', 'TRX004'),

(5, 305, 'Payment successful', 'Completed', 'TRX005');

CREATE TABLE Admins ( admin\_id INT PRIMARY KEY, username VARCHAR(50), password VARCHAR(255), email VARCHAR(255) );

INSERT INTO Admins (admin\_id, username, password, email, last\_login\_date)

VALUES

(1, 'admin1', 'hashed\_password\_1', 'admin1@example.com'),

(2, 'admin2', 'hashed\_password\_2', 'admin2@example.com'),

(3, 'admin3', 'hashed\_password\_3', 'admin3@example.com'),

(4, 'admin4', 'hashed\_password\_4', 'admin4@example.com'),

(5, 'admin5', 'hashed\_password\_5', 'admin5@example.com');

CREATE TABLE cart (

cart\_id INT,

total\_item INT,

grand\_total DECIMAL(10, 2),

customer\_id INT

);

INSERT INTO cart (cart\_id, total\_item, grand\_total, customer\_id)

VALUES

(201, 3, 129.97, 101),

(202, 2, 59.98, 102),

(203, 1, 99.99, 103),

(204, 2, 159.98, 104),

(205, 4, 189.96, 105);

CREATE TABLE Supplier ( name VARCHAR(100), email VARCHAR(255), phone\_number VARCHAR(20), address VARCHAR(255) );

INSERT INTO Supplier (supplier\_id, name, email, phone\_number, address)

VALUES

(701, 'TechSupply Inc.', 'info@techsupply.com', '+1 (555) 123-4567', '123 Tech Ave, Tech City'),

(702, 'ElectroWorld Suppliers', 'contact@electroworld.com', '+1 (555) 987-6543', '456 Circuit St, Electronics Town'),

(703, 'Gadget Source LLC', 'sales@gadgetsource.com', '+1 (555) 222-3333', '789 Innovation Blvd, Gadget Valley'),

(704, 'Fashion Fabrics Ltd.', 'info@fashionfabrics.com', '+1 (555) 777-8888', '321 Style Ave, Fashion District'),

(705, 'Sporting Gear Co.', 'sales@sportinggear.com', '+1 (555) 444-5555', '555 Fitness Plaza, Sportstown');

CREATE TABLE review (

review\_id INT,

description VARCHAR(255),

ratings DECIMAL(3, 2),

product\_id INT,

customer\_id INT

);

INSERT INTO review (review\_id, description, ratings, product\_id, customer\_id)

VALUES

(1, 'Great product, very satisfied with the quality!', 4.5, 101, 201),

(2, 'Fast shipping, item as described.', 4.0, 102, 202),

(3, 'Good value for the price, would recommend.', 3.8, 103, 203),

(4, 'Excellent customer service, resolved my issue quickly.', 5.0, 104, 204),

CREATE TABLE wishlist (

wishlist\_id INT,

customer\_id INT,

product\_id INT,

added\_at TIMESTAMP

);

INSERT INTO wishlist (wishlist\_id, customer\_id, product\_id, added\_at)

VALUES

(1, 101, 201, '2024-03-15 10:00:00'),

(2, 102, 202, '2024-03-16 11:30:00'),

(3, 103, 203, '2024-03-17 09:45:00'),

(4, 104, 204, '2024-03-18 14:20:00'),

(5, 105, 205, '2024-03-19 16:55:00');

CREATE TABLE shipment (

shipment\_id INT,

arrival\_date DATE,

estimate\_date DATE,

shipment\_date DATE,

order\_id INT

);

INSERT INTO shipment (shipment\_id, arrival\_date, estimate\_date, shipment\_date, order\_id)

VALUES

(1, '2024-03-20', '2024-03-25', '2024-03-18', 101),

(2, '2024-03-21', '2024-03-26', '2024-03-19', 102),

(3, '2024-03-22', '2024-03-27', '2024-03-20', 103),

(4, '2024-03-23', '2024-03-28', '2024-03-21', 104),

(5, '2024-03-24', '2024-03-29', '2024-03-22', 105);

CREATE TABLE category (

category\_id INT,

category\_name VARCHAR(50),

description VARCHAR(255)

);

INSERT INTO category (category\_id, category\_name, description)

VALUES

(1, 'Electronics', 'Includes all electronic devices such as smartphones, laptops, and cameras.'),

(2, 'Clothing', 'A wide range of clothing items for men, women, and kids.'),

(3, 'Books', 'Various genres of books including fiction, non-fiction, and educational.'),

(4, 'Home & Kitchen', 'Products for home decor, kitchen appliances, and accessories.'),

(5, 'Sports & Outdoors', 'Equipment and gear for sports, outdoor activities, and fitness.');

QUEIRES-

1. Find the total number of customers:

SELECT COUNT(\*) AS TotalCustomers FROM Customers;

1. List all customers with their email and phone number:

SELECT name, email, phone\_number FROM Customers;

1. Find the total stock of all products:

SELECT SUM(stock) AS TotalStock FROM Product;

1. List all products with their MRP and Brand:

SELECT product\_name, MRP, Brand FROM Product;

1. Find the total amount of all orders:

SELECT SUM(order\_amount) AS TotalOrdersAmount FROM Orders;

1. List all orders with their shipping date and order amount:

SELECT shipping\_date, order\_amount FROM Orders;

1. Find the total quantity of all order items:

SELECT SUM(quantity) AS TotalOrderItems FROM OrderItem;

1. List all order items with their product name and total price:

SELECT p.product\_name, oi.total\_price FROM OrderItem oi JOIN Product p ON oi.product\_id = p.product\_id;

1. Find the total number of payments made:

SELECT COUNT(\*) AS TotalPayments FROM payment;

1. List all payments with their payment mode and payment date:

SELECT payment\_mode, payment\_date FROM payment;

1. Find the total number of successful payments:

SELECT COUNT(\*) AS SuccessfulPayments FROM payment\_history WHERE status = 'Completed';

1. List all admins with their username and email:

SELECT username, email FROM Admins;

1. Find the total number of items in all carts:

SELECT SUM(total\_item) AS TotalCartItems FROM cart;

1. List all suppliers with their name and email:

SELECT name, email FROM Supplier;

1. Find the average rating of all products:

SELECT AVG(ratings) AS AverageRating FROM review;

1. List all reviews with their description and ratings:

SELECT description, ratings FROM review;

1. Find the total number of items in all wishlists:

SELECT COUNT(\*) AS TotalWishlistItems FROM wishlist;

1. List all shipments with their arrival date and order id:

SELECT arrival\_date, order\_id FROM shipment;

1. Find the total number of categories:

SELECT COUNT(\*) AS TotalCategories FROM category;

1. List all categories with their category name and description:

SELECT category\_name, description FROM category;

1. List all customers with their email and phone number.

SELECT name, email, phone\_number FROM Customers;

1. Find the total stock of all products.

SELECT SUM(stock) AS total\_stock FROM Product;

1. List all products with a stock less than 10.

SELECT product\_name, stock FROM Product WHERE stock < 10;

1. Find the total amount of orders placed in March 2024.

SELECT SUM(order\_amount) AS total\_orders FROM Orders WHERE MONTH(order\_date) = 3 AND YEAR(order\_date) = 2024;

1. List all customers who have placed an order.

SELECT DISTINCT c.name FROM Customers c JOIN Orders o ON c.customer\_id = o.cart\_id;

1. Find the total revenue generated from each product category.

SELECT c.category\_name, SUM(oi.total\_price) AS total\_revenue

FROM OrderItem oi

JOIN Product p ON oi.product\_id = p.product\_id

JOIN category c ON p.category\_id = c.category\_id

GROUP BY c.category\_name;

1. List all customers who have made a payment using Credit Card.

SELECT DISTINCT c.name FROM Customers c JOIN payment p ON c.customer\_id = p.customer\_id WHERE p.payment\_mode = 'Credit Card';

1. Find the total number of successful payments.

SELECT COUNT(\*) AS successful\_payments FROM payment\_history WHERE status = 'Completed';

1. List all admins who have never logged in.

SELECT username FROM Admins WHERE last\_login\_date IS NULL;

1. Find the total number of items in each customer's cart.

SELECT customer\_id, SUM(total\_item) AS total\_items FROM cart GROUP BY customer\_id;

1. List all suppliers who supply products in the 'Electronics' category.

SELECT s.name FROM Supplier s JOIN Product p ON s.supplier\_id = p.supplier\_id WHERE p.category\_id = 1;

1. Find the average rating for each product.

SELECT product\_id, AVG(ratings) AS average\_rating FROM review GROUP BY product\_id;

1. List all customers who have added a product to their wishlist.

SELECT DISTINCT c.name FROM Customers c JOIN wishlist w ON c.customer\_id = w.customer\_id;

1. Find the total number of shipments delivered in March 2024.

SELECT COUNT(\*) AS total\_deliveries FROM shipment WHERE MONTH(arrival\_date) = 3 AND YEAR(arrival\_date) = 2024;

1. List all products that have been reviewed by a customer with a rating of 5.

SELECT p.product\_name FROM Product p JOIN review r ON p.product\_id = r.product\_id WHERE r.ratings = 5;

1. Find the total number of items in each order.

SELECT order\_id, SUM(quantity) AS total\_items FROM OrderItem GROUP BY order\_id;

1. List all customers who have placed an order for a product from the 'Electronics' category.

SELECT DISTINCT c.name FROM Customers c JOIN Orders o ON c.customer\_id = o.cart\_id JOIN OrderItem oi ON o.order\_id = oi.order\_id JOIN Product p ON oi.product\_id = p.product\_id WHERE p.category\_id = 1;

1. Find the total revenue generated from orders placed in March 2024.

SELECT SUM(order\_amount) AS total\_revenue FROM Orders WHERE MONTH(order\_date) = 3 AND YEAR(order\_date) = 2024;

1. List all products that have been reviewed by a customer with a rating of 4 or higher.

SELECT p.product\_name FROM Product p JOIN review r ON p.product\_id = r.product\_id WHERE r.ratings >= 4;

1. Find the total number of products in each category.

SELECT category\_id, COUNT(\*) AS total\_products FROM Product GROUP BY category\_id;