**SQL PROJECT (Sales Database) – Soham Chavan**

1. Create table

-- Create Products Table

CREATE TABLE products (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(255),

category VARCHAR(50),

price DECIMAL(10, 2)

);

-- Create Customers Table

CREATE TABLE customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255),

email VARCHAR(100),

registration\_date DATE

);

-- Create Sales Table

CREATE TABLE sales (

sale\_id INT PRIMARY KEY,

product\_id INT,

customer\_id INT,

sale\_date DATE,

sales\_amount DECIMAL(10, 2),

FOREIGN KEY (product\_id) REFERENCES products(product\_id),

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

1. Inserting data

Insert Values into Products Table-

INSERT INTO products (product\_id, product\_name, category, price)

VALUES

(1, 'Laptop', 'Electronics', 1200.00),

(2, 'Smartphone', 'Electronics', 800.00),

(3, 'Desk Chair', 'Furniture', 150.00),

(4, 'Coffee Table', 'Furniture', 200.00),

(5, 'Headphones', 'Electronics', 100.00),

(6, 'Bookshelf', 'Furniture', 180.00),

(7, 'Blender', 'Appliances', 50.00),

(8, 'Dining Table', 'Furniture', 300.00),

(9, 'Fitness Tracker', 'Electronics', 120.00),

(10, 'Toaster', 'Appliances', 30.00),

(11, 'Office Desk', 'Furniture', 250.00),

(12, 'Digital Camera', 'Electronics', 600.00),

(13, 'Microwave', 'Appliances', 80.00),

(14, 'Couch', 'Furniture', 500.00),

(15, 'Vacuum Cleaner', 'Appliances', 70.00);

Insert Values into Customers Table-

INSERT INTO customers (customer\_id, customer\_name, email, registration\_date)

VALUES

(101, 'John Doe', 'john.doe@email.com', '2022-01-01'),

(102, 'Jane Smith', 'jane.smith@email.com', '2022-02-15'),

(103, 'Bob Johnson', 'bob.johnson@email.com', '2022-03-10'),

(104, 'Emily White', 'emily.white@email.com', '2022-04-05'),

(105, 'Michael Brown', 'michael.brown@email.com', '2022-05-20'),

(106, 'Sophia Lee', 'sophia.lee@email.com', '2022-06-12'),

(107, 'David Clark', 'david.clark@email.com', '2022-07-25'),

(108, 'Olivia Taylor', 'olivia.taylor@email.com', '2022-08-18'),

(109, 'William Hall', 'william.hall@email.com', '2022-09-03'),

(110, 'Ava Allen', 'ava.allen@email.com', '2022-10-09');

Insert Values into Sales Table-

INSERT INTO sales (sale\_id, product\_id, customer\_id, sale\_date, sales\_amount)

VALUES

(1, 1, 101, '2022-01-05', 1200.00),

(2, 2, 102, '2022-02-20', 800.00),

(3, 3, 103, '2022-03-15', 150.00),

(4, 4, 101, '2022-04-01', 200.00),

(5, 5, 104, '2022-05-10', 100.00),

(6, 6, 105, '2022-06-25', 180.00),

(7, 7, 106, '2022-07-18', 50.00),

(8, 8, 107, '2022-08-30', 300.00),

(9, 9, 108, '2022-09-15', 120.00),

(10, 10, 109, '2022-10-03', 30.00),

(11, 11, 110, '2022-11-12', 250.00),

(12, 12, 101, '2022-12-05', 600.00),

(13, 13, 102, '2023-01-08', 80.00),

(14, 14, 103, '2023-02-20', 500.00),

(15, 15, 104, '2023-03-10', 70.00);

1. Simple Aggregations:

Total Sales Amount:

SELECT SUM(sales\_amount) AS total\_sales FROM sales;

Average Price of Products:

SELECT AVG(price) AS average\_price FROM products;

1. Filtering and Sorting:

Products in the Electronics Category:

SELECT \* FROM products WHERE category = 'Electronics';

Top 5 Highest Priced Products:

SELECT \* FROM products ORDER BY price DESC LIMIT 5;

1. Grouping and Aggregating:

Total Sales by Product Category:

SELECT category, SUM(sales\_amount) AS total\_sales FROM sales GROUP BY category;

Number of Sales per Customer:

SELECT customer\_id, COUNT(sale\_id) AS num\_sales FROM sales GROUP BY customer\_id;

1. Conditional Filtering:

Products with Price Greater Than 200:

SELECT \* FROM products WHERE price > 200;

Customers with Total Spending Above 1000:

SELECT c.customer\_name, SUM(s.sales\_amount) AS total\_spending

FROM customers c

JOIN sales s ON c.customer\_id = s.customer\_id

GROUP BY c.customer\_name

HAVING total\_spending > 1000;

1. Date-related Queries:

Sales in March 2022:

SELECT \* FROM sales WHERE sale\_date >= '2022-03-01' AND sale\_date < '2022-04-01';

Most Recent Customer Registrations:

SELECT \* FROM customers ORDER BY registration\_date DESC LIMIT 5;

Identify Months with the Highest Total Revenue:

SELECT EXTRACT(MONTH FROM s.sale\_date) AS month, SUM(s.sales\_amount) AS total\_revenue

FROM sales s

GROUP BY month

ORDER BY total\_revenue DESC

LIMIT 1;

1. Calculate Average Sales per Customer:

SELECT c.customer\_id, c.customer\_name, AVG(s.sales\_amount) AS avg\_sales

FROM customers c

JOIN sales s ON c.customer\_id = s.customer\_id

GROUP BY c.customer\_id, c.customer\_name;

1. Joins and Subqueries:

List Customers Who Purchased Products from Multiple Categories:

SELECT c.customer\_id, c.customer\_name

FROM customers c

WHERE c.customer\_id IN (

SELECT DISTINCT s.customer\_id

FROM sales s

JOIN products p ON s.product\_id = p.product\_id

GROUP BY s.customer\_id

HAVING COUNT(DISTINCT p.category) > 1

);