**1.Ecommerce\_SQL\_Database**

CREATE TABLE Customerrs

(

customer\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

country VARCHAR(50)

);

INSERT INTO Customerrs VALUES(1,'Alice Johnson', 'alice@example.com', 'USA')

INSERT INTO Customerrs VALUES(2,'Bob Smith', 'bob@example.com', 'Canada')

INSERT INTO Customerrs VALUES(3,'Carlos Gomez', 'carlos@example.com', 'Mexico')

INSERT INTO Customerrs VALUES(4,'Diana Wu', 'diana@example.com', 'China')

INSERT INTO Customerrs VALUES(5,'Ethan Brown', 'ethan@example.com', 'UK')

select\*from customerrs



CREATE TABLE Productt

(

product\_id INT PRIMARY KEY,

product\_name VARCHAR(100) NOT NULL,

category VARCHAR(50),

price DECIMAL(10, 2) NOT NULL

);

INSERT INTO Productt VALUES(1,'Wireless Mouse', 'Electronics', 25.99)

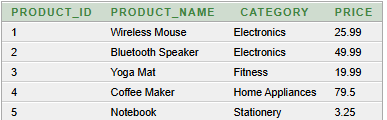
INSERT INTO Productt VALUES(2,'Bluetooth Speaker', 'Electronics', 49.99)

INSERT INTO Productt VALUES(3,'Yoga Mat', 'Fitness', 19.99)

INSERT INTO Productt VALUES(4,'Coffee Maker', 'Home Appliances', 79.50)

INSERT INTO Productt VALUES(5,'Notebook', 'Stationery', 3.25)

select\*from productt



CREATE TABLE Orderc

(

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE NOT NULL,

total\_amount DECIMAL(12, 2),

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

);

INSERT INTO Orderc VALUES (101, 1,'20-Mar-2025', 150.50)

INSERT INTO Orderc VALUES (102, 2,'21-Mar-2025', 80.00)

select\*from Orderc



CREATE TABLE OrderDetails

(

order\_detail\_id INT PRIMARY KEY,

order\_id INT,

product\_id INT,

quantity INT NOT NULL,

price DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

INSERT INTO OrderDetails VALUES(1, 1, 2, 25.99,101)

INSERT INTO OrderDetails VALUES(2, 2, 1, 49.99,102)

select\*from OrderDetails



**2. Basic Queries (SELECT, WHERE, ORDER BY, GROUP BY)**

SELECT \* FROM Customerrs

WHERE country = 'USA';



SELECT name, email

FROM Customerrs

WHERE country = 'India'

ORDER BY name ASC;



SELECT customer\_id, COUNT(order\_id) AS total\_orders

FROM Orderc

GROUP BY customer\_id;



**3. JOINS**

INNER JOIN:

SELECT o.order\_id, o.order\_date, c.name, c.country

FROM Orderc o

INNER JOIN Customerrs c ON o.customer\_id = c.customer\_id;

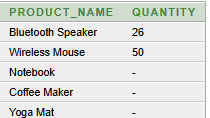


LEFT JOIN:

SELECT p.product\_name, od.quantity

FROM Productt p

LEFT JOIN OrderDetails od ON p.product\_id = od.product\_id;



RIGHT JOIN:

SELECT od.order\_id, od.quantity, p.product\_name, p.category

FROM OrderDetails od

RIGHT JOIN Productt p ON od.product\_id = p.product\_id;



**4. Aggregate Functions and GROUP BY**

SELECT c.country, SUM(o.total\_amount) AS total\_sales

FROM Orderc o

JOIN Customerrs c ON o.customer\_id = c.customer\_id

GROUP BY c.country

ORDER BY total\_sales DESC;



SELECT p.product\_name, AVG(od.quantity) AS avg\_quantity

FROM OrderDetails od

JOIN Productt p ON od.product\_id = p.product\_id

GROUP BY p.product\_name;



**5. Subqueries**

SELECT name

FROM Customerrs

WHERE customer\_id IN (

SELECT customer\_id

FROM Orderc

GROUP BY customer\_id

HAVING COUNT(order\_id) > 3

);



**6. Views**

CREATE VIEW SalesSummary AS

SELECT p.category, SUM(od.price \* od.quantity) AS total\_revenue

FROM OrderDetails od

JOIN Productt p ON od.product\_id = p.product\_id

GROUP BY p.category;

SELECT \* FROM SalesSummary;

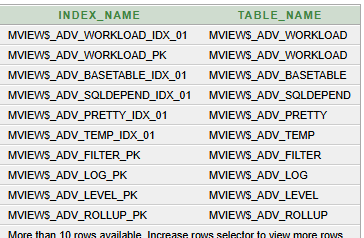


**7.Indexing**

CREATE INDEX idx\_customer\_country ON Customerrs(country);

SELECT index\_name, table\_name

FROM user\_indexes;



**Summary**:

**Objective**: Use SQL queries to extract and analyze data from a database.

**Tools**: Oracle Database 10g Express Edition

**Deliverables**: SQL queries in a .sql file + screenshots of output

**Commands**:

\* **SELECT**: Retrieves all columns from the specified table.

\* **WHERE**: Filters records based on the condition.

\* **ORDER BY**: Sorts the results based on a column.

\* **LIMIT 10**: Restricts the output to the top 10 results.

\* **INNER** **JOIN**: Retrieves only matching records from both tables.

\* **LEFT** **JOIN**: Retrieves all records from the left table and matching records from the right.

\* **RIGHT** **JOIN**: Retrieves all records from the right table and matching records from the left.

\* **GROUP** **BY**: Groups rows that have the same values in specified columns.

\* **SUM(), AVG():** Aggregate functions to calculate the sum and average.

\* **Subquery**: A query within another query to filter or aggregate data.

\* **HAVING**: Used with GROUP BY to filter grouped results.

\* **CREATE** **VIEW**: Creates a virtual table from the result of a query.

\* **Indexes**: Speed up search queries, especially on columns that are frequently queried or used in joins.

**Outcome**

\* Objective Achieved: Learn to manipulate and query structured data using SQL.

\* Key Skills Gained:

o Writing basic and advanced SQL queries.

o Using aggregation functions like SUM() and AVG().

o Mastering JOINs (INNER, LEFT, RIGHT).

o Writing and using subqueries.

o Creating views for data analysis.

o Optimizing queries using indexes.