**2. Trainer Script & Explanation**

**A. SQL Joins**

**Definition**: Combines rows from two or more tables based on related columns.

**1. INNER JOIN**

* Returns only matching rows from both tables.
* **Syntax**:

sql

RunCopy code

1SELECT columns FROM table1

2INNER JOIN table2 ON table1.column = table2.column;

* **Example**:

sql

RunCopy code

1SELECT e.employee\_name, d.department\_name

2FROM employees e

3INNER JOIN departments d ON e.dept\_id = d.dept\_id;

**2. LEFT JOIN (LEFT OUTER JOIN)**

* Returns all rows from the left table and matched rows from the right table (NULL if no match).
* **Syntax**:

sql

RunCopy code

1SELECT columns FROM table1

2LEFT JOIN table2 ON table1.column = table2.column;

* **Example**:

sql

RunCopy code

1SELECT c.customer\_name, o.order\_id

2FROM customers c

3LEFT JOIN orders o ON c.customer\_id = o.customer\_id;

**3. RIGHT JOIN (RIGHT OUTER JOIN)**

* Returns all rows from the right table and matched rows from the left table (NULL if no match).
* **Syntax**:

sql

RunCopy code

1SELECT columns FROM table1

2RIGHT JOIN table2 ON table1.column = table2.column;

* **Example**:

sql

RunCopy code

1SELECT o.order\_id, p.product\_name

2FROM orders o

3RIGHT JOIN products p ON o.product\_id = p.product\_id;

**4. FULL JOIN (FULL OUTER JOIN)**

* Returns all rows when there is a match in either table (NULL for non-matching rows).
* **Syntax**:

sql

RunCopy code

1SELECT columns FROM table1

2FULL JOIN table2 ON table1.column = table2.column;

* **Example**:

sql

RunCopy code

1SELECT e.employee\_name, d.department\_name

2FROM employees e

3FULL JOIN departments d ON e.dept\_id = d.dept\_id;

**5. SELF JOIN**

* Joins a table to itself (useful for hierarchical data).
* **Syntax**:

sql

RunCopy code

1SELECT a.column, b.column

2FROM table a, table b

3WHERE a.column = b.column;

* **Example**:

sql

RunCopy code

1SELECT e1.employee\_name, e2.manager\_name

2FROM employees e1, employees e2

3WHERE e1.manager\_id = e2.employee\_id;

**6. CROSS JOIN**

* Returns the Cartesian product of two tables (all possible combinations).
* **Syntax**:

sql

RunCopy code

1SELECT columns FROM table1

2CROSS JOIN table2;

* **Example**:

sql

RunCopy code

1SELECT p.product\_name, s.store\_name

2FROM products p

3CROSS JOIN stores s;

**B. Subqueries**

**Definition**: A query nested inside another query (SELECT, INSERT, UPDATE, DELETE).

**1. Single-Row Subquery**

* Returns only one row.
* **Example**:

sql

RunCopy code

1SELECT employee\_name FROM employees

2WHERE salary > (SELECT AVG(salary) FROM employees);

**2. Multi-Row Subquery**

* Returns multiple rows.
* **Example**:

sql

RunCopy code

1SELECT product\_name FROM products

2WHERE category\_id IN (SELECT category\_id FROM categories WHERE category\_name LIKE 'Electronics%');

**3. Correlated Subquery**

* References outer query columns.
* **Example**:

sql

RunCopy code

1SELECT e.employee\_name, e.salary

2FROM employees e

3WHERE salary > (SELECT AVG(salary) FROM employees WHERE dept\_id = e.dept\_id);

**C. DML Operations (Data Manipulation Language)**

**1. INSERT**

* Adds new rows to a table.
* **Syntax**:

sql

RunCopy code

1INSERT INTO table\_name (column1, column2) VALUES (value1, value2);

* **Example**:

sql

RunCopy code

1INSERT INTO employees (employee\_id, employee\_name, salary) VALUES (101, 'John Doe', 50000);

**2. UPDATE**

* Modifies existing records.
* **Syntax**:

sql

RunCopy code

1UPDATE table\_name SET column1 = value1 WHERE condition;

* **Example**:

sql

RunCopy code

1UPDATE employees SET salary = 55000 WHERE employee\_id = 101;

**3. DELETE**

* Removes rows based on a condition.
* **Syntax**:

sql

RunCopy code

1DELETE FROM table\_name WHERE condition;

* **Example**:

sql

RunCopy code

1DELETE FROM employees WHERE employee\_id = 101;

**4. TRUNCATE**

* Removes all rows (faster than DELETE, no rollback).
* **Syntax**:

sql

RunCopy code

1TRUNCATE TABLE table\_name;

* **Example**:

sql

RunCopy code

1TRUNCATE TABLE temp\_employees;

**3. Learner Exercises**

**Exercise 1: Joins**

**Problem**:  
Retrieve all customers and their orders (even if they haven’t placed any).

**Solution**:

sql

RunCopy code

1SELECT c.customer\_name, o.order\_id

2FROM customers c

3LEFT JOIN orders o ON c.customer\_id = o.customer\_id;

**Exercise 2: Subquery**

**Problem**:  
Find products with prices higher than the average price.

**Solution**:

sql

RunCopy code

1SELECT product\_name, price

2FROM products

3WHERE price > (SELECT AVG(price) FROM products);

**Exercise 3: DML Operations**

**Problem**:  
Insert a new employee and update their salary.

**Solution**:

sql

RunCopy code

1INSERT INTO employees (employee\_id, employee\_name, salary) VALUES (102, 'Jane Smith', 60000);

2UPDATE employees SET salary = 65000 WHERE employee\_id = 102;

**4. Oracle Live SQL Script**

sql

RunCopy code

1-- Create sample tables

2CREATE TABLE employees (

3 employee\_id NUMBER PRIMARY KEY,

4 employee\_name VARCHAR2(100),

5 salary NUMBER,

6 dept\_id NUMBER

7);

8

9CREATE TABLE departments (

10 dept\_id NUMBER PRIMARY KEY,

11 dept\_name VARCHAR2(100)

12);

13

14-- Insert sample data

15INSERT INTO departments VALUES (1, 'HR');

16INSERT INTO departments VALUES (2, 'IT');

17

18INSERT INTO employees VALUES (101, 'John Doe', 50000, 1);

19INSERT INTO employees VALUES (102, 'Jane Smith', 60000, 2);

20

21-- INNER JOIN example

22SELECT e.employee\_name, d.dept\_name

23FROM employees e

24INNER JOIN departments d ON e.dept\_id = d.dept\_id;

25

26-- LEFT JOIN example

27SELECT e.employee\_name, d.dept\_name

28FROM employees e

29LEFT JOIN departments d ON e.dept\_id = d.dept\_id;

30

31-- UPDATE example

32UPDATE employees SET salary = 55000 WHERE employee\_id = 101;

**5. Summary**

✅ **Joins** → Combine tables (INNER, LEFT, RIGHT, FULL, SELF, CROSS).  
✅ **Subqueries** → Nested queries for complex filtering.  
✅ **DML** → INSERT, UPDATE, DELETE, TRUNCATE for data manipulation.