-- Refactored SQL Script for Joins and Subqueries (Oracle Live SQL Compatible)

-- Author: Trainer Support Script

-- Description: Demonstrates joins and subqueries in Oracle with clean syntax

-- =====================

-- JOIN CONCEPTS

-- =====================

-- LEFT JOIN: All records from LEFT table + matching records from RIGHT table

-- RIGHT JOIN: All records from RIGHT table + matching records from LEFT table

-- INNER JOIN: Only matching records from both tables

-- Join is always based on a common column/attribute.

-- Drop if already exists

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE transactions1';

EXECUTE IMMEDIATE 'DROP TABLE accounts1';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

-- Create accounts1 table (Primary)

CREATE TABLE accounts1 (

acc\_id INT,

t\_type VARCHAR2(30),

amount INT

);

-- Create transactions1 table (Secondary)

CREATE TABLE transactions1 (

acc\_id INT,

t\_id INT,

t\_date DATE

);

-- Insert sample data into accounts1

INSERT INTO accounts1 VALUES (101, 'withdraw', 2000);

INSERT INTO accounts1 VALUES (102, 'deposit', 5000);

INSERT INTO accounts1 VALUES (103, 'deposit', 6000);

-- Insert sample data into transactions1

INSERT INTO transactions1 VALUES (101, 2001, SYSDATE);

INSERT INTO transactions1 VALUES (102, 2002, SYSDATE);

-- LEFT JOIN Example

SELECT

a.acc\_id,

a.t\_type,

a.amount,

t.t\_id,

t.t\_date

FROM

transactions1 t

LEFT JOIN

accounts1 a ON a.acc\_id = t.acc\_id;

-- =====================

-- SELF JOIN EXAMPLE (EMPLOYEE-MANAGER)

-- =====================

-- Drop and recreate employeess table

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE employeess';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

CREATE TABLE employeess (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR2(200),

manager\_id INT

);

-- Insert employee-manager data

INSERT INTO employeess VALUES (1, 'Elakiya', NULL);

INSERT INTO employeess VALUES (2, 'Aasha', 1);

INSERT INTO employeess VALUES (3, 'Sowmiya', 1);

INSERT INTO employeess VALUES (4, 'Hariharan', 2);

INSERT INTO employeess VALUES (5, 'Suryanarayan', 2);

-- Self Join to show employee and their manager

SELECT

e.emp\_id,

e.emp\_name AS employee,

m.emp\_name AS manager

FROM

employeess e

LEFT JOIN

employeess m ON e.manager\_id = m.emp\_id;

-- =====================

-- SUBQUERY SECTION

-- =====================

-- Types:

-- 1. Single-row (MAX, MIN etc.)

-- 2. Multi-row (IN, ANY, ALL)

-- 3. Scalar (Single value)

-- 4. Correlated Subquery (uses outer query reference)

-- Example Tables: customers, orders, payments

-- Show customers who spent more than average amount

SELECT \*

FROM payments

WHERE amount > (SELECT AVG(amount) FROM payments);

-- Show customers with order amount > average

SELECT customernumber, customername

FROM customers

WHERE customernumber IN (

SELECT customernumber

FROM orders

WHERE amount > (SELECT AVG(amount) FROM orders)

);

-- Multi-row Subquery Example

SELECT \*

FROM customers

WHERE customernumber IN (

SELECT DISTINCT customernumber FROM orders

);

-- Scalar Subquery: Show each customer's latest order date

SELECT

c.customernumber,

c.customername,

(SELECT MAX(orderdate) FROM orders o WHERE o.customernumber = c.customernumber) AS latest\_order

FROM customers c;

-- =====================

-- SALARY SCALAR SUBQUERY EXAMPLE

-- =====================

-- Drop and recreate employee salary table

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE employeesss';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

CREATE TABLE employeesss (

id INT,

name VARCHAR2(100),

salary INT

);

-- Insert salary data

INSERT INTO employeesss VALUES (1, 'Niti', 1000);

INSERT INTO employeesss VALUES (2, 'Nitin', 2000);

INSERT INTO employeesss VALUES (3, 'Nitika', 3000);

INSERT INTO employeesss VALUES (4, 'Riya', 4000);

-- Scalar Query: Compare actual vs max salary

SELECT

name,

salary AS actual\_salary,

(SELECT MAX(salary) FROM employeesss) AS max\_salary

FROM employeesss;

-- Single-row Subquery

SELECT \*

FROM employeesss

WHERE salary < (SELECT MAX(salary) FROM employeesss);