Lab cycle 2

Expt no: 4

IMPLEMENTATION OF SUBQUERY, JOINS, VIEWS AND SET OPERATIONS

AIM:

Consider the following Database Schema given at the left and execute queries to familiarize with subqueries, joins, view and set operations.

QUERIES:

1. Create table **regions**.
   regions (region_id INT PRIMARY KEY,region_name VARCHAR(50));
2. Create table **countries**.
   countries (country_id CHAR(2) PRIMARY KEY,country_name VARCHAR(50),region_id INT,FOREIGN KEY (region_id) REFERENCES regions(region_id));
3. Create table **locations**.
   locations (location_id INT PRIMARY KEY,street_address VARCHAR(100),postal_code VARCHAR(20),city VARCHAR(50),state_province VARCHAR(50),country_id CHAR(2),FOREIGN KEY (country_id) REFERENCES countries(country_id));
4. Create table **departments**.
   departments (department_id INT PRIMARY KEY,department_name VARCHAR(50),location_id INT,FOREIGN KEY (location_id) REFERENCES locations(location_id));
5. Create table **employees**.
   employees (employee_id INT PRIMARY KEY,first_name VARCHAR(50),last_name VARCHAR(50),email VARCHAR(100),phone_number VARCHAR(20),hire_date DATE,job_id VARCHAR(10),salary DECIMAL(10, 2),manager_id INT,department_id INT,FOREIGN KEY (job_id) REFERENCES jobs(job_id),FOREIGN KEY (manager_id) REFERENCES employees(employee_id),FOREIGN KEY (department_id) REFERENCES departments(department_id));

6. Create table **dependents**.
   dependents (dependent_id INT PRIMARY KEY,first_name VARCHAR(50),last_name VARCHAR(50),relationship VARCHAR(50),employee_id INT,FOREIGN KEY (employee_id) REFERENCES employees(employee_id));

7. Create table **jobs**.
   jobs (job_id VARCHAR(10) PRIMARY KEY,job_title VARCHAR(50),min_salary DECIMAL(10, 2),max_salary DECIMAL(10, 2));

8. Insert records into table **regions**.
   (1, 'Europe'),(2, 'Americas'),(3, 'Asia'),(4, 'Middle East and Africa');

9. Insert records into table **countries**.
   ('IT', 'Italy', 1),('US', 'United States', 2),('JP', 'Japan', 3),('IN', 'India', 3),('ZA', 'South Africa', 4);

10. Insert records into table **locations**.

(1000, 'Via Roma 100', '00100', 'Rome', 'Lazio', 'IT'),(1700, '2000 Broadway', '94111', 'San Francisco', 'California', 'US'),
(1800, 'Nishi Shinjuku 3-2', '160-0023', 'Tokyo', 'Tokyo', 'JP'),
(1900, 'Block B, ITPL', '560066', 'Bangalore', 'Karnataka', 'IN'),
(2000, '1 Mandela Way', '8001', 'Cape Town', 'Western Cape', 'ZA'),
(2100, '123 Connaught Place', '110001', 'Delhi', 'Delhi', 'IN');

11. Insert records into table **departments**.

(10, 'Administration', 1000),(20, 'Marketing', 1700),(30, 'Sales', 1700),(40, 'Finance', 1800),(50, 'IT', 1900);

12. Insert records into table **employees**.

(101, 'John', 'Doe', 'jdoe@example.com', '1234567890', '2020-01-15', 'J01', 12000, NULL, 10),
(102, 'Jane', 'Smith', 'jsmith@example.com', '9876543210', '2019-03-10', 'J02', 8000, 101, 20),
(103, 'Alice', 'Johnson', 'ajohnson@example.com', '5556667777', '2018-11-25', 'J03', 15000, 101, 30),
(104, 'Robert', 'Taylor', 'rtaylor@example.com', '8889990000', '2021-07-01', 'J04', 11000, 102, 40),
(105, 'Linda', 'Williams', 'lwilliams@example.com', '4445556666', '2022-05-12', 'J05', 9500, 103, 50),
(106, 'Mark', 'Brown', 'mbrown@example.com', '9998887777', '2024-01-10', 'J05', 11000, 105, 50),
(109, 'Zara', 'Ali', 'zali@example.com', '3334445555', '2024-02-01', 'J01', 9000, NULL, 10),
(110, 'Ravi', 'Sharma', 'rsharma@example.com', '9998887777', '2024-03-15', 'J01', 8500, NULL, 20),
(111, 'Smith', 'David', 'dsmith@example.com', '8887776666', '2024-04-01', 'J03', NULL, NULL, 40);

13. Insert records into table **dependents**.

(1, 'Anna', 'Doe', 'Daughter', 101),
(2, 'Michael', 'Smith', 'Son', 102),
(3, 'Emily', 'Johnson', 'Spouse', 103),
(4, 'Sophia', 'Taylor', 'Daughter', 104),
(5, 'James', 'Williams', 'Son', 105);

14. Insert records into table **jobs**.

('J01', 'Administrator', 10000, 15000),
('J02', 'Marketing Specialist', 7000, 12000),
('J03', 'Sales Manager', 12000, 18000),
('J04', 'Finance Analyst', 10000, 14000),
('J05', 'IT Specialist', 9000, 13000);

15. Display all the records from the above tables.
16. Find all employees who are located in the location with ID **1700**.
17. Find all employees who are **not** located in location **1700**.

18. Find the employees who have the **highest salary**. (SUBQUERIES)
19. Find all employees whose salaries are **greater than the average salary** of all employees. (SUBQUERIES)
20. Find all departments (Department ID, Name) which have at least one employee with a salary **greater than 10,000**. (SUBQUERIES)
21. Find all departments (Department ID, Name) that **do not** have any employee with a salary greater than **10,000**. (SUBQUERIES)
22. Find all employees whose salaries are **greater than the lowest salary** of every department. (SUBQUERIES)
23. Find all employees whose salaries are **greater than or equal to the highest salary** of every department. (SUBQUERIES)
24. Calculate the **average of average salaries** of departments. **(Hint: SQL subquery in the FROM clause)** (SUBQUERIES)
25. Find the salaries of all employees, their **average salary**, and the **difference** between each employee's salary and the average salary. **(Hint: SQL Subquery in the SELECT clause)** (SUB)
26. Find all employees whose salary is **higher than the average salary of their department**. **(Hint: Use Correlated Subquery).**
27. Return all employees who **have no dependents**.
28. Display **first name, last name, and department name** of employees from departments **1, 2, and 3**. (JOIN)
29. Display **first name, last name, job title, and department name** of employees from departments **10, 20, and 30** whose salary is greater than **10,000**. (JOIN)
30. Display **department name, street address, postal code, country name, and region name** of all departments. (JOIN)