Task 6: Subqueries and Nested Queries

Objective:

Use subqueries in SELECT, WHERE, and FROM clauses to enhance SQL query logic.

Tools:

- DB Browser for SQLite
- MySQL Workbench

Hints / Mini Guide

1. Scalar Subquery:

Returns a single value (one row, one column).

Example:

```
SELECT name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

2. Correlated Subquery:

A subquery that references columns from the outer query.

Example:

```
SELECT name, department_id, salary
FROM employees e
WHERE salary > (
SELECT AVG(salary)
FROM employees
WHERE department_id = e.department_id
```

);

3. Subquery in FROM Clause:

Used to create temporary tables or derived tables.

Example:

```
SELECT dept_avg.department_id, dept_avg.avg_salary
FROM (
SELECT department_id, AVG(salary) AS avg_salary
FROM employees
GROUP BY department_id
) AS dept_avg
WHERE avg_salary > 50000;
```

4. Subquery with IN:

Example:

```
SELECT name
FROM employees
WHERE department_id IN (
SELECT department_id
FROM departments
WHERE location = 'New York'
);
```

5. Subquery with EXISTS:

Example:

```
SELECT department_id, department_name
FROM departments d
```

```
WHERE EXISTS (
SELECT 1
FROM employees e
WHERE e.department_id = d.department_id
);
```

6. Subquery using "=" operator:

Example:

```
SELECT name
FROM employees
WHERE salary = (SELECT MAX(salary) FROM employees);
```

Outcome:

- Mastery in using nested SQL queries
- Ability to apply advanced logic in SELECT, WHERE, and FROM clauses
- Understanding the difference between scalar, correlated, and non-correlated subqueries