Non-Correlated Subqueries

A non-correlated subquery is an independent subquery that can be executed separately from the outer query. It does not reference columns from the outer query.

Example

1. **Creating Tables:**

```
CREATE TABLE Departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50)
);

CREATE TABLE Employees (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(50),
    department_id INT,
    salary DECIMAL(10, 2),
    FOREIGN KEY (department_id) REFERENCES
Departments(department_id)
);
```

2. **Inserting Records:**

```
INSERT INTO Departments (department_id, department_name)
VALUES
(1, 'Engineering'),
(2, 'Marketing');

INSERT INTO Employees (employee_id, employee_name,
department_id, salary) VALUES
(1, 'Ravi Kumar', 1, 75000),
(2, 'Anjali Sharma', 2, 65000),
(3, 'Amit Verma', 1, 80000),
(4, 'Neha Gupta', 2, 70000),
(5, 'Vijay Singh', 1, 72000);
```

3. **Non-Correlated Subquery Example**:

```
SELECT employee_name, salary
FROM Employees
WHERE department_id = (SELECT department_id FROM
Departments WHERE department_name = 'Engineering');
```

- This query finds employees who work in the 'Engineering' department.

Correlated Subqueries

A correlated subquery is dependent on the outer query. It references columns from the outer query and is executed once for each row selected by the outer query.

Example

1. **Correlated Subquery Example**:

```
SELECT e1.employee_name, e1.salary
FROM Employees e1
WHERE e1.salary > (SELECT AVG(e2.salary) FROM Employees
e2 WHERE e2.department_id = e1.department_id);
```

- This query finds employees whose salary is above the average salary of their respective departments.

Summary

- **Non-Correlated Subquery**:
  - Independent subquery that can be executed separately.
  - Example: Finding employees in the 'Engineering' department.
- **Correlated Subquery**:
  - Dependent subquery that references columns from the outer query.
  - Example: Finding employees with a salary above the department average.

Putting it All Together

```
-- Create Departments table
CREATE TABLE Departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50)
);

-- Create Employees table
CREATE TABLE Employees (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(50),
    department_id INT,
    salary DECIMAL(10, 2),
    FOREIGN KEY (department_id) REFERENCES
Departments(department_id)
);

-- Insert records into Departments table
INSERT INTO Departments (department_id, department_name)
VALUES
(1, 'Engineering'),
(2, 'Marketing');

-- Insert records into Employees table
INSERT INTO Employees (employee_id, employee_name,
department_id, salary) VALUES
(1, 'Ravi Kumar', 1, 75000),
(2, 'Anjali Sharma', 2, 65000),
(3, 'Amit Verma', 1, 80000),
(4, 'Neha Gupta', 2, 70000),
(5, 'Vijay Singh', 1, 72000);

-- Non-Correlated Subquery Example
SELECT employee_name, salary
```

```
FROM Employees
WHERE department_id = (SELECT department_id FROM
Departments WHERE department_name = 'Engineering');

-- Correlated Subquery Example
SELECT e1.employee_name, e1.salary
FROM Employees e1
WHERE e1.salary > (SELECT AVG(e2.salary) FROM Employees e2
WHERE e2.department_id = e1.department_id);
```

These examples demonstrate how to use correlated and non-correlated subqueries in SQL, along with table creation and record insertion.