**Lab 1: Basic Scalar Subquery**

**Task:** Retrieve the names of employees whose salaries are above the average salary.

**Solution:**

SELECT EmployeeName, Salary

FROM Employees

WHERE Salary > (SELECT AVG(Salary) FROM Employees);

**Lab 2: Subquery in the SELECT Clause (Scalar Subquery)**

**Task:** Display each employee’s name along with the total number of employees in the company.

**Solution:**

SELECT EmployeeName,

(SELECT COUNT(\*) FROM Employees) AS TotalEmployees

FROM Employees;

**Lab 3: Subquery with EXISTS**

**Task:** Retrieve a list of customers who have placed at least one order.

**Solution:**

SELECT CustomerID, CustomerName

FROM Customers C

WHERE EXISTS (SELECT 1 FROM Orders O WHERE O.CustomerID = C.CustomerID);

**Lab 4: Subquery with NOT EXISTS**

**Task:** Retrieve employees who have never made any sales.

**Solution:**

SELECT EmployeeID, EmployeeName

FROM Employees E

WHERE NOT EXISTS (SELECT 1 FROM Sales S WHERE S.EmployeeID = E.EmployeeID);

**Lab 5: Correlated Subquery**

**Task:** Retrieve orders where the order amount is greater than the average order amount for that customer.

**Solution:**

SELECT OrderID, CustomerID, Amount

FROM Orders O1

WHERE Amount > (SELECT AVG(Amount)

FROM Orders O2

WHERE O2.CustomerID = O1.CustomerID);

**Lab 6: Subquery with IN**

**Task:** Retrieve the names of employees who work in departments located in 'New York'.

**Solution:**

SELECT EmployeeName

FROM Employees

WHERE DepartmentID IN (SELECT DepartmentID FROM Departments WHERE Location = 'New York');

**Lab 7: Subquery with ANY**

**Task:** Retrieve products whose price is greater than at least one product in category 'Electronics'.

**Solution:**

SELECT ProductName, Price

FROM Products

WHERE Price > ANY (SELECT Price FROM Products WHERE Category = 'Electronics');

**Lab 8: Subquery with ALL**

**Task:** Retrieve products whose price is higher than all products in the 'Books' category.

**Solution:**

SELECT ProductName, Price

FROM Products

WHERE Price > ALL (SELECT Price FROM Products WHERE Category = 'Books');

**Lab 9: Nested Subqueries**

**Task:** Retrieve customers who have placed orders with a total amount greater than $5000.

**Solution:**

SELECT CustomerID, CustomerName

FROM Customers

WHERE CustomerID IN

(SELECT CustomerID

FROM Orders

WHERE OrderID IN

(SELECT OrderID FROM OrderDetails GROUP BY OrderID HAVING SUM(Amount) > 5000));

**Lab 10: Common Table Expression (CTE) as an Alternative to Subqueries**

**Task:** Retrieve employees who have a salary greater than the average salary in their department.

**Solution:**

WITH AvgSalaryPerDept AS (

SELECT DepartmentID, AVG(Salary) AS AvgSalary

FROM Employees

GROUP BY DepartmentID

)

SELECT E.EmployeeID, E.EmployeeName, E.Salary, E.DepartmentID

FROM Employees E

JOIN AvgSalaryPerDept A ON E.DepartmentID = A.DepartmentID

WHERE E.Salary > A.AvgSalary;