**Question 2:**

**Question 1: Top 3 Departments with Highest Average Salary**

**Task:**

* + - 1. Write a SQL query to find the top 3 departments with the highest average salary of employees. Ensure departments with no employees show an average salary of NULL.

**Deliverables:**

* + - 1. SQL query that retrieves DepartmentID, DepartmentName, and AvgSalary for the top 3 departments.
      2. Explanation of how the query handles departments with no employees and calculates average salary.

**Question 2: Retrieving Hierarchical Category Paths**

**Task:**

* + - 1. Write a SQL query using recursive Common Table Expressions (CTE) to retrieve all categories along with their full hierarchical path (e.g., Category > Subcategory > Sub-subcategory).

**Deliverables:**

* + - 1. SQL query that uses recursive CTE to fetch CategoryID, CategoryName, and hierarchical path.
      2. Explanation of how the recursive CTE works to traverse the hierarchical data.

**SQL QUERY: (QUESTION 1)**

WITH DeptAvg AS (

SELECT

d.DepartmentID,

d.DepartmentName,

AVG(e.Salary) AS AvgSalary

FROM

Departments d

LEFT JOIN

Employees e ON d.DepartmentID = e.DepartmentID

GROUP BY

d.DepartmentID, d.DepartmentName

)

SELECT

DepartmentID,

DepartmentName,

AvgSalary

FROM

DeptAvg

ORDER BY

AvgSalary DESC

LIMIT 3;

**SQL QUERY : (Question 2)**

WITH RECURSIVE CategoryPaths AS (

-- Anchor member: start with top-level categories

SELECT

CategoryID,

CategoryName,

CAST(CategoryName AS VARCHAR(255)) AS Path

FROM

Categories

WHERE

ParentCategoryID IS NULL

UNION ALL

-- Recursive member: append subcategories

SELECT

c.CategoryID,

c.CategoryName,

CONCAT(cp.Path, ' > ', c.CategoryName) AS Path

FROM

Categories c

INNER JOIN

CategoryPaths cp ON c.ParentCategoryID = cp.CategoryID

)

SELECT

CategoryID,

CategoryName,

Path

FROM

CategoryPaths;