**Top 3 Departments with Highest Average Salary**

**SQL Query:**

sql

SELECT DepartmentID, DepartmentName,

AVG(Salary) AS AvgSalary

FROM Departments d

LEFT JOIN Employees e ON d.DepartmentID = e.DepartmentID

GROUP BY DepartmentID, DepartmentName

ORDER BY AvgSalary DESC

FETCH FIRST 3 ROWS ONLY;

```

**question 2: Retrieving Hierarchical Category Paths**

**SQL Query:**

WITH RECURSIVE CategoryPaths AS (

SELECT CategoryID, CategoryName,

CAST(CategoryName AS VARCHAR(255)) AS Path

FROM Categories

WHERE ParentCategoryID IS NULL

UNION ALL

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;

```

### Question 3: Total Distinct Customers by Month

SQL Query:

WITH MonthNames AS (

SELECT TO\_CHAR(ADD\_MONTHS(TRUNC(SYSDATE, 'YEAR'), LEVEL - 1), 'Month') AS MonthName,

LEVEL AS MonthNumber

FROM DUAL

CONNECT BY LEVEL <= 12

),

CustomerCounts AS (

SELECT TO\_CHAR(OrderDate, 'Month') AS MonthName,

COUNT(DISTINCT CustomerID) AS CustomerCount

FROM Orders

WHERE EXTRACT(YEAR FROM OrderDate) = EXTRACT(YEAR FROM SYSDATE)

GROUP BY TO\_CHAR(OrderDate, 'Month')

)

SELECT m.MonthName, NVL(c.CustomerCount, 0) AS CustomerCount

FROM MonthNames m

LEFT JOIN CustomerCounts c ON m.MonthName = c.MonthName

ORDER BY m.MonthNumber;

```

**Question 4: Finding Closest Locations**

SQL Query:

SELECT LocationID, LocationName, Latitude, Longitude,

(6371 \* ACOS(COS(RADIANS(:latitude)) \* COS(RADIANS(Latitude)) \*

COS(RADIANS(Longitude) - RADIANS(:longitude)) +

SIN(RADIANS(:latitude)) \* SIN(RADIANS(Latitude)))) AS Distance

FROM Locations

ORDER BY Distance

FETCH FIRST 5 ROWS ONLY;

```

**Question 5: Optimizing Query for Orders Table**

**SQL Query:**

SELECT OrderID, OrderDate, CustomerID, TotalAmount

FROM Orders

WHERE OrderDate >= SYSDATE - 7

ORDER BY OrderDate DESC;

CREATE TABLE Orders (

OrderID NUMBER PRIMARY KEY,

OrderDate DATE,

CustomerID NUMBER,

TotalAmount NUMBER

)

PARTITION BY RANGE (OrderDate) (

PARTITION p1 VALUES LESS THAN (TO\_DATE('2023-01-01', 'YYYY-MM-DD')),

PARTITION p2 VALUES LESS THAN (TO\_DATE('2023-02-01', 'YYYY-MM-DD')),

...

);