

SQL INTERVIEW PREPARATION PART 4.2

WINDOWS FUNCTIONS QUESTIONS CONTINUED:

11. First and Last Sale for Each Product

Scenario:

You have a Sales table with columns ProductID, SaleDate, and SaleAmount.

Question:

Write a query to find the first and last sale amount for each product.

Solution:

```
WITH ProductSaleAmount AS (  
    SELECT  
        ProductID,  
        SaleDate,  
        SaleAmount,  
        ROW_NUMBER() OVER (PARTITION BY ProductID ORDER BY SaleDate ASC) AS FirstSale,  
        ROW_NUMBER() OVER (PARTITION BY ProductID ORDER BY SaleDate DESC) AS LastSale  
    FROM Sales  
)  
SELECT  
    ProductID,  
    MAX(CASE WHEN FirstSale = 1 THEN SaleAmount END) AS First_Sale_Amount,  
    MAX(CASE WHEN LastSale = 1 THEN SaleAmount END) AS Last_Sale_Amount  
FROM ProductSaleAmount  
GROUP BY ProductID;
```

12. Median Salary by Department

Scenario:

You have an Employees table with EmpID, DepartmentID, and Salary.

Question:

Write a query to calculate the median salary for each department using window functions.

Solution:

```
WITH RankedSalaries AS (  
    SELECT  
        DepartmentID,  
        Salary,  
        ROW_NUMBER() OVER (PARTITION BY DepartmentID ORDER BY Salary ASC) AS  
Row_Num,  
        COUNT(*) OVER (PARTITION BY DepartmentID) AS Total_Rows  
    FROM Employees  
)  
SELECT
```

```

DepartmentID,
CASE
    WHEN Total_Rows % 2 = 1 THEN
        MAX(CASE WHEN Row_Num = (Total_Rows / 2) + 1 THEN Salary END)
    ELSE
        AVG(CASE WHEN Row_Num IN (Total_Rows / 2, (Total_Rows / 2) + 1) THEN Salary
END)
    END AS Median_Salary
FROM RankedSalaries
GROUP BY DepartmentID;

```

13. Find Employees Above Team Average

Scenario:

You have an Employees table with EmpID, TeamID, and Salary.

Question:

Write a query to find employees whose salaries are above their team's average salary.

Solution:

```

SELECT EmpID,
       TeamID,
       Salary
FROM (
    SELECT EmpID,
           TeamID,
           Salary,
           AVG(Salary) OVER (PARTITION BY TeamID) AS Average_Salary
    FROM Employees
) AS TeamSalary
WHERE Salary > Average_Salary;

```

14. Dense Ranks in Sales by Quarter

Scenario:

You have a Sales table with SaleID, SaleAmount, and SaleDate.

Question:

Write a query to assign a dense rank to each sale by quarter and year.

Solution:

```

SELECT SaleID,
       SaleAmount,
       DENSE_RANK() OVER (PARTITION BY YEAR(SaleDate) ORDER BY SaleAmount DESC) AS
Rank_by_Year,
       DENSE_RANK() OVER (PARTITION BY YEAR(SaleDate), QUARTER(SaleDate) ORDER BY
SaleAmount DESC) AS Rank_by_Quarter
FROM Sales;

```

15. Find Gaps in Ranks

Scenario:

You have a Scores table with StudentID and Score.

Question:

Write a query to identify gaps in ranks when students have identical scores.

Solution:

```
WITH RankedScores AS (  
    SELECT  
        StudentID,  
        Score,  
        RANK() OVER (ORDER BY Score DESC) AS Score_Rank  
    FROM Scores  
)  
ContinuousRanks AS (  
    SELECT  
        Score_Rank,  
        ROW_NUMBER() OVER (ORDER BY Score_Rank) AS Continuous_Rank  
    FROM RankedScores  
)  
SELECT  
    Score_Rank,  
    Continuous_Rank,  
    Score_Rank - Continuous_Rank AS Rank_Gap  
FROM ContinuousRanks  
WHERE Score_Rank - Continuous_Rank > 0;
```