SQL Window Functions: Interview Questions & Case Studies

Introduction to SQL Window Functions

Window functions in SQL allow you to perform calculations across a subset of rows related to the current row.

Unlike aggregate functions (SUM, COUNT, etc.), which collapse multiple rows into a single output row, window functions retain individual rows while providing additional calculated values.

Common Window Functions in SQL

- ROW_NUMBER(): Assigns a unique sequential integer to rows within a partition.
- RANK(): Assigns a ranking to each row within a partition, with gaps for duplicates.
- DENSE_RANK(): Similar to RANK(), but without gaps in ranking.
- NTILE(N): Distributes rows into N number of groups.
- LEAD(): Fetches the value from the next row within the partition.
- LAG(): Fetches the value from the previous row within the partition.
- FIRST_VALUE(): Retrieves the first value in an ordered partition.
- LAST_VALUE(): Retrieves the last value in an ordered partition.
- SUM(), AVG(), MIN(), MAX() (with OVER() clause): Performs aggregate calculations while retaining row-level details.

Scenario-Based SQL Window Function Case Studies

Case Study 1: Sales Ranking by Region

Scenario: A retail company wants to rank its stores based on their total sales in each region.

SQL Query:

```sql
SELECT StoreID, Region, SalesAmount,
       RANK() OVER (PARTITION BY Region ORDER BY SalesAmount DESC) AS SalesRank
FROM SalesData;
```

Case Study 2: Identifying Top 3 Highest Salaries in Each Department

Scenario: A company wants to identify the top 3 highest salaries in each department without removing duplicate values.

SQL Query:

```sql
SELECT EmployeeID, Department, Salary,
       DENSE_RANK() OVER (PARTITION BY Department ORDER BY Salary DESC) AS SalaryRank
FROM Employees
WHERE SalaryRank <= 3;
```

Case Study 3: Finding Monthly Sales Growth

Scenario: A company wants to track monthly revenue growth for each product.

SQL Query:

```sql
SELECT ProductID, SalesMonth, Revenue,
       LAG(Revenue, 1, 0) OVER (PARTITION BY ProductID ORDER BY SalesMonth) AS PreviousMonthRevenue,
       Revenue - LAG(Revenue, 1, 0) OVER (PARTITION BY ProductID ORDER BY SalesMonth) AS RevenueGrowth
FROM Sales;
```

Common SQL Window Function Interview Questions

1. **What is the difference between RANK() and DENSE_RANK()?**
2. **How does the NTILE() function work?**
3. **When would you use LAG() and LEAD()?**
4. **How can you get a running total using window functions?**

```sql
SELECT EmployeeID, Salary,
      SUM(Salary) OVER (ORDER BY EmployeeID) AS RunningTotal
FROM Employees;
```

5. **Can you use multiple window functions in the same query?** Yes, you can.

Conclusion

Mastering SQL window functions is essential for handling analytical queries efficiently.

Understanding their use cases through real-world scenarios makes it

easier to tackle interview questions.