DAY 12 – SQL Window Functions

Topics: Window Functions, ROW_NUMBER, RANK, DENSE_RANK

1. What are Window Functions?

Window functions perform calculations across a set of table rows related to the current row. Unlike GROUP BY, window functions do not reduce the number of rows returned by the query.

General Syntax:

```
SELECT column, window_function() OVER (PARTITION BY column ORDER BY column) FROM table;
```

2. ROW_NUMBER()

ROW_NUMBER assigns a unique sequential number to each row within a partition. Even if rows have the same values, each row gets a different number.

Query Example:

```
SELECT name, dept, salary, ROW_NUMBER() OVER (PARTITION BY dept ORDER BY salary DESC) AS row_num FROM employees;
```

Use Case:

Used when each row must be uniquely identified, such as removing duplicates or pagination.

3. RANK()

RANK assigns the same rank to rows with equal values. However, it skips the next rank value when duplicates exist.

Query Example:

```
SELECT name, dept, salary, RANK() OVER (PARTITION BY dept ORDER BY salary DESC) AS rank_val FROM employees;
```

Important Note:

If two employees share rank 1, the next employee receives rank 3.

4. DENSE_RANK()

DENSE_RANK is similar to RANK, but it does not create gaps in ranking. Ranks increase sequentially.

Query Example:

```
SELECT name, dept, salary, DENSE_RANK() OVER (PARTITION BY dept ORDER BY salary DESC) AS dense_rank FROM employees;
```

5. Difference Summary

- ROW_NUMBER: Unique number, no duplicates, no gaps
- RANK: Same rank for same values, gaps allowed
- DENSE_RANK: Same rank for same values, no gaps

6. Interview Query – Top N per Group

```
SELECT * FROM ( SELECT name, dept, salary, DENSE_RANK() OVER (PARTITION BY dept
ORDER BY salary DESC) AS rnk FROM employees ) t WHERE rnk <= 2;
```

7. Key Points to Remember

- Window functions do not reduce rows
- PARTITION BY acts like grouping inside a window
- ORDER BY defines ranking or calculation order
- DENSE_RANK is preferred for leaderboard-style ranking