Window Functions Decoded: Part 1

Section 1: Learn

What are Window Functions in SQL?

Window functions **perform calculations across a set of rows** related to the current row, **without collapsing rows** like aggregate functions.

Why Use Window Functions?

- **Retains Individual Rows**: Unlike GROUP BY, window functions **preserve row-level data**.
- **Ranks and Sorts Data Efficiently**: Enables numbering and ranking.
- **Improves Analytical Queries**: Works well for reporting and trends analysis.

Key Window Functions Covered

1. **ROW_NUMBER()** – Assigns a unique row number.
2. **RANK()** – Assigns ranks, with gaps if there are ties.
3. **DENSE_RANK()** – Assigns ranks, without gaps for ties.

Interesting Fact:

Window functions are heavily used in **data analytics and reporting systems** to rank and filter records dynamically!

Section 2: Practice

1. Understanding and Applying Window Functions

Basic Syntax of Window Functions

```
SELECT column_name, window_function() OVER (PARTITION BY
column_name ORDER BY column_name)
FROM table_name;
```

- **PARTITION BY**: Defines the group of rows the function operates on.
- **ORDER BY**: Defines the ordering within the partition.

---

2. ROW_NUMBER, RANK, DENSE_RANK in Practical Use

Using ROW_NUMBER() to Assign Unique Row Numbers

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

- **Assigns a unique row number** within each department, ordered by **salary**.

Using RANK() to Assign Ranks (With Gaps)

```
SELECT name, department, salary,
    RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS
rank_num
FROM Employees;
```

- If two employees have the same salary, **they get the same rank, and the next rank is skipped**.

Using DENSE_RANK() to Assign Ranks (Without Gaps)

```
SELECT name, department, salary,
    DENSE_RANK() OVER (PARTITION BY department ORDER BY salary
DESC) AS dense_rank_num
FROM Employees;
```

- If two employees have the same salary, **they get the same rank, but the next rank is not skipped**.

Section 3: Know More

Frequently Asked Questions (FAQs)

1. What is the difference between RANK() and DENSE_RANK()?

- **RANK()** leaves **gaps** in ranking when there are ties.
- **DENSE_RANK()** does not leave gaps.

2. Can I use multiple window functions in one query?

Yes! Example:

```
SELECT name, salary,
    ROW_NUMBER() OVER (ORDER BY salary DESC) AS row_num,
    RANK() OVER (ORDER BY salary DESC) AS rank_num
FROM Employees;
```

3. How do I filter only the top-ranked record per group?

Use ROW_NUMBER() in a subquery:

```
SELECT * FROM (
    SELECT name, department, salary,
        ROW_NUMBER() OVER (PARTITION BY department ORDER BY salary
DESC) AS row_num
    FROM Employees
) AS RankedEmployees WHERE row_num = 1;
```

4. Do window functions improve performance?

They can **simplify queries** but may slow down **on large datasets** if indexing is not used properly.

5. Can window functions be used with GROUP BY?

No, window functions **work independently** of GROUP BY.
