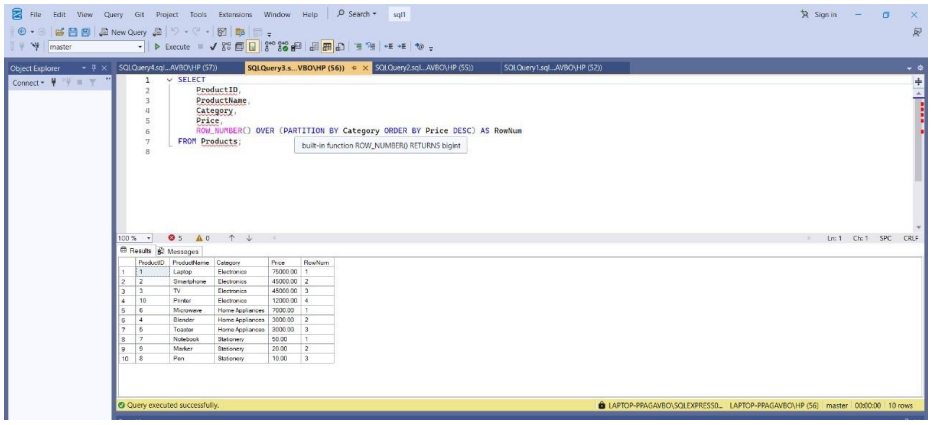
# SQL Window Functions: Ranking and Top-N Queries

## 1. Use ROW_NUMBER() to Assign a Unique Rank Within Each Category

The ROW_NUMBER() function assigns a unique number to each row within a partition. Even if values are tied, each row receives a distinct number.

```
SELECT
   ProductID,
   ProductName,
   Category,
   Price,
   ROW_NUMBER() OVER (PARTITION BY Category ORDER BY Price
DESC) AS RowNum
FROM Products;
```



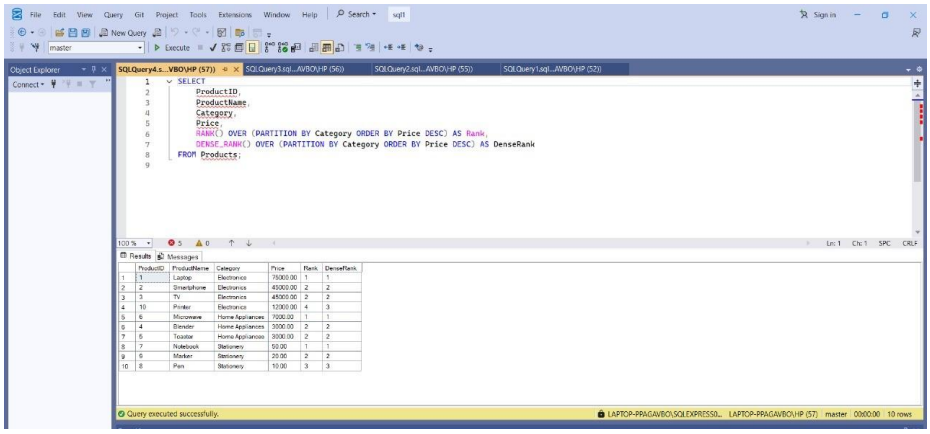

---

ROW_NUMBER() gives a unique number to each row — even if prices are the same (no ties allowed).

## 2. Use RANK() and DENSE_RANK() to Compare Tie Handling

```
SELECT
   ProductID,
   ProductName,
```

```
    Category,
    Price,
    RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS Rank,
    DENSE_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRank
FROM Products;
```




---

Difference:

RANK(): Skips numbers after ties.
Example: If two products tie at 1st place, the next one is 3rd.

DENSE_RANK(): No skipping.
Example: If two products tie at 1st, next one is 2nd.

## 3. Find Top 3 Most Expensive Products in Each Category Using ROW_NUMBER()

```
WITH RankedProducts AS (
    SELECT
        ProductID,
        ProductName,
        Category,
        Price,
        ROW_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum
    FROM Products
)
```

***SELECT ****
***FROM RankedProducts***
***WHERE RowNum <= 3;***