SQL-04 | Window Functions

Lecture Queries

WINDOW Functions

Window Functions

Window fns give the ability to put the values from one row of data into context compared to a group of rows, or partition.

We can answer questions like

- If the dataset were sorted, where would this row land in the results?
- How does a value in this row compare to a value in the prior row?
- How does a value in the current row compare to the average value for its group?

So, window functions **return group aggregate calculations alongside individual row-level** information for items in that group, or partition.

New Question: Extract the most expensive items and the **product_id** they are associated with per vendor.

```
SELECT
        vendor_id,
        market_date,
        product_id,
        original_price,
    ROW_NUMBER() OVER (PARTITION BY vendor_id ORDER BY original_price DESC)
AS price_rank
FROM farmers_market.vendor_inventory
```

RANK()

The RANK function numbers the results just like ROW_NUMBER does, but gives rows with the same value the same ranking.

```
SELECT
    vendor_id,
    market_date,
    product_id,
    original_price,
    RANK() OVER (PARTITION BY vendor_id ORDER BY
original_price DESC) AS
 price_rank
    FROM farmers_market.vendor_inventory
```

DENSE_RANK()

If you don’t want to skip rank numbers for tied values like in case of RANK, use the DENSE_RANK function.

```
SELECT
    vendor_id,
    market_date,
    product_id,
    original_price,
    DENSE_RANK() OVER (PARTITION BY vendor_id ORDER
BY original_price DESC) AS
 price_rank
    FROM farmers_market.vendor_inventory
```

Return the “**top tenth**” of the inventory, when sorted by price?

The dynamic solution is to use the NTILE function.

```
SELECT
        vendor_id,
        market_date,
        product_id,
        original_price,
        NTILE(10) OVER (ORDER BY original_price DESC) AS price_ntile
FROM farmers_market.vendor_inventory
ORDER BY original_price DESC
```

Question: As a farmer, you want to figure out which of your products were above the average price per product on each market date?

```
SELECT
        vendor_id,
        market_date,
        product_id,
        original_price,
        AVG(original_price) OVER (PARTITION BY market_date ORDER BY
       market_date) AS average_cost_product_by_market_date
FROM farmers_market.vendor_inventory
```

Extract the farmer's products that have prices above the market date's average product cost.

- Using a **subquery**, we can filter the results to a single vendor, with **vendor_id 8**, and
- only **display products that have prices above the market date's average product** cost.

```
SELECT * FROM
        (
            SELECT
                vendor_id,
                    market_date,
                    product_id,
                    original_price,
                    ROUND(AVG(original_price) OVER (PARTITION BY market_date ORDER
        BY market_date), 2) AS average_cost_product_by_market_date
FROM farmers_market.vendor_inventory )x
WHERE x.vendor_id = 8
                    AND x.original_price > x.average_cost_product_by_market_date
```

Question: Count how many different products each vendor brought to market on each date, and displays that count on each row.

```
SELECT
    vendor_id,
    market_date,
    product_id,
    original_price,
    COUNT(product_id) OVER (PARTITION BY market_date, vendor_id)
 vendor_product_count_per_market_date
    FROM farmers_market.vendor_inventory
 ORDER BY vendor_id, market_date, original_price DESC
```