Day12 : Window Functions

Aug 11, 2023

```sql
--FOPO - Function over Partition by order by
--FO : Function over : Table + Value of the function
SELECT *
FROM farmers_market.customer_purchases;

SELECT SUM(quantity)Total
FROM farmers_market.customer_purchases;

SELECT *, SUM(quantity) OVER() Total
FROM farmers_market.customer_purchases;
--------------------------------------------------------------------------------------

--FOP

SELECT *
FROM farmers_market.customer_purchases;

SELECT product_id,SUM(quantity)Total
FROM farmers_market.customer_purchases
GROUP BY product_id;


SELECT *, SUM(quantity) OVER(PARTITION BY product_id) Total
FROM farmers_market.customer_purchases;

--------------------------------------------------------------------------------------

--FOO

SELECT *, SUM(quantity) OVER(ORDER BY transaction_time) Total
FROM farmers_market.customer_purchases;

--Issue : if transaction_time is the same then calculation will not happen. We have
solution see it in future

--------------------------------------------------------------------------------------
```

```sql
--FOPO

SELECT product_id,SUM(quantity)Total
FROM farmers_market.customer_purchases
GROUP BY product_id;

SELECT *, SUM(quantity) OVER(PARTITION BY product_id ORDER BY transaction_time ) Total
FROM farmers_market.customer_purchases;

--------------------------------------------------------------------------------------

--Ranking functions

SELECT *, rank() OVER (ORDER BY quantity DESC) AS rnk,
          dense_rank() OVER (ORDER BY quantity DESC) AS d_rnk
FROM farmers_market.vendor_inventory
ORDER BY quantity DESC;

SELECT *, ROW_NUMBER() OVER (ORDER BY quantity DESC) AS rno
FROM farmers_market.vendor_inventory;

--------------------------------------------------------------------------------------

--Rank with filtration
SELECT * FROM (
  SELECT *,
      ROW_NUMBER() OVER (PARTITION BY vendor_id ORDER BY quantity DESC) AS rno
FROM farmers_market.vendor_inventory
) A
WHERE A.rno=1;
```