WINDOW FUNCTIONS

GROUP BY ——— Summarizing data.

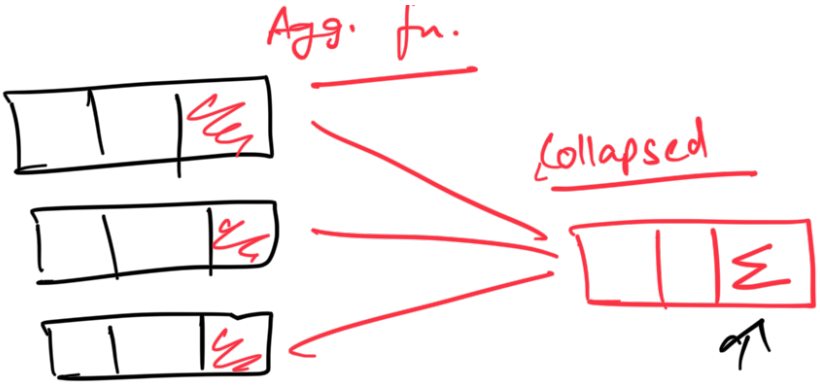
Q. Get the price of the most expensive item per vendor.

```
SELECT
    vendor_id,
    MAX (original_price)
FROM ———
GROUP BY v_id;
```

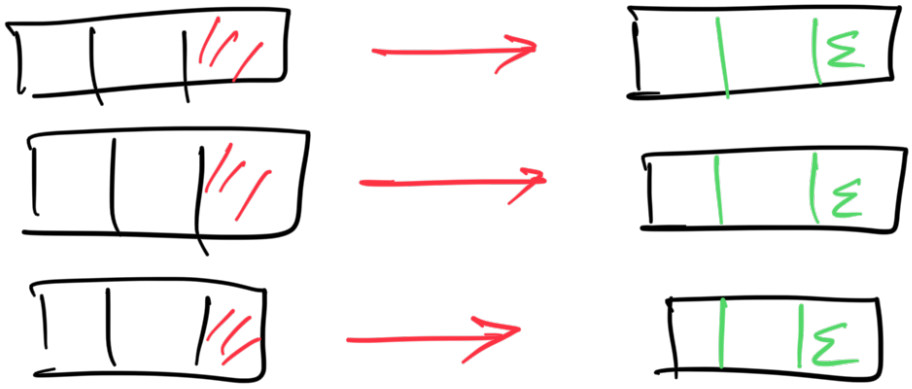
Q. Rank the products in each vendor's inventory. Expensive products get a lower rank.

Window functions

GROUP BY



Window Functions



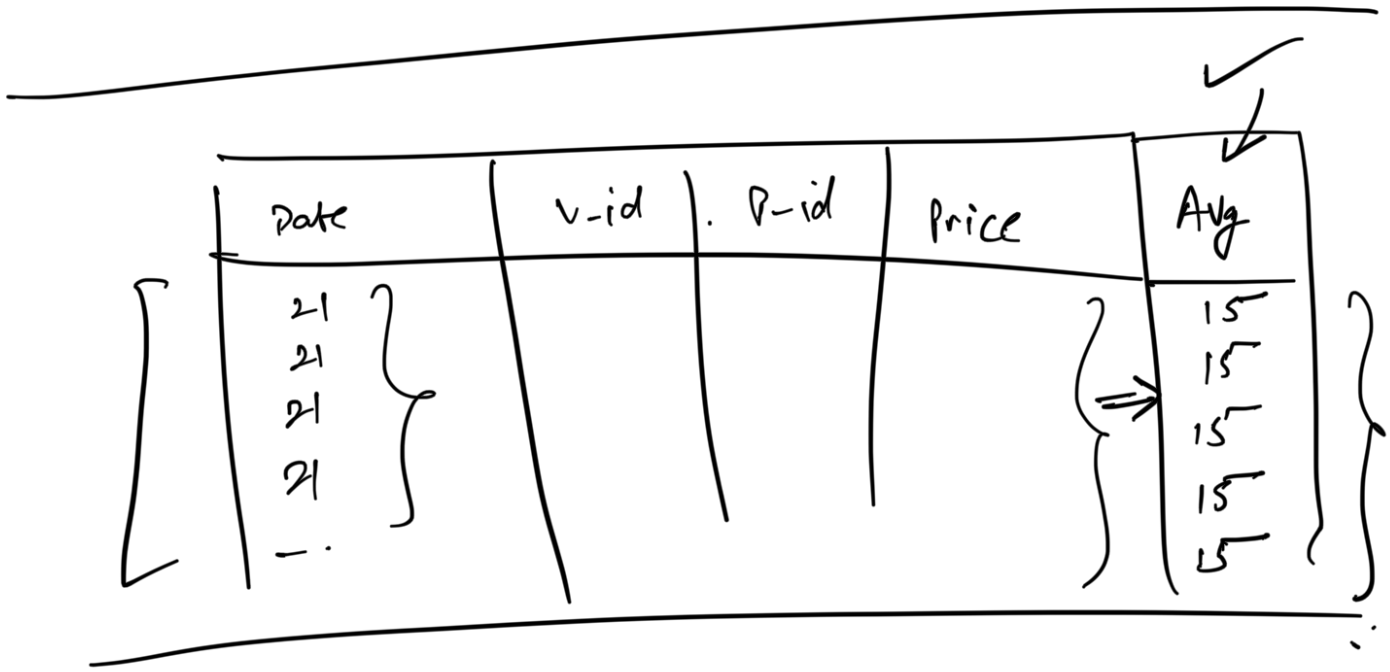
Syntax:

1. ROW_NUMBER()

```
SELECT
    ROW_NUMBER() OVER ( PARTITION BY ___, ___, ___
                        ORDER BY ___ )
FROM ___
```

NL

Q: Figure out which products were above the average price on each market_date.



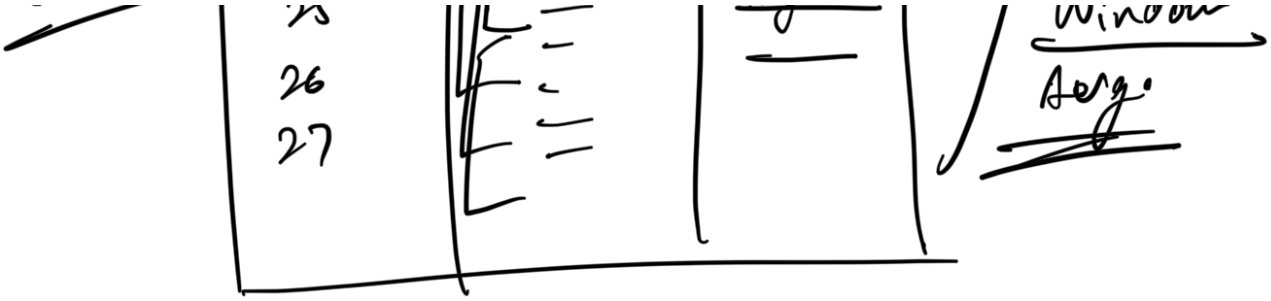
Q: Count how many different products each vendor brought to the market on each date and display that count on each row.

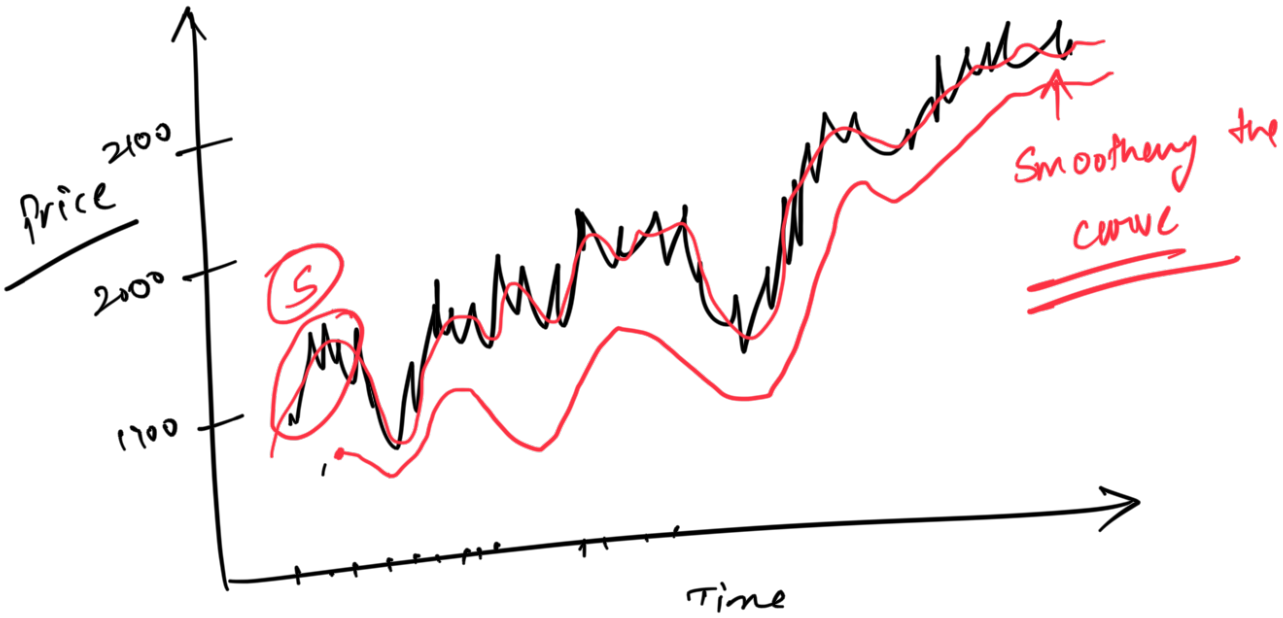
Reliance Stock Price Data

Date	close_price	5-Day Avg
21	1950	NULL
22	—	NULL
23	—	NULL
24	—	NULL

Cumulative
200
350
450
−200
−150
−100

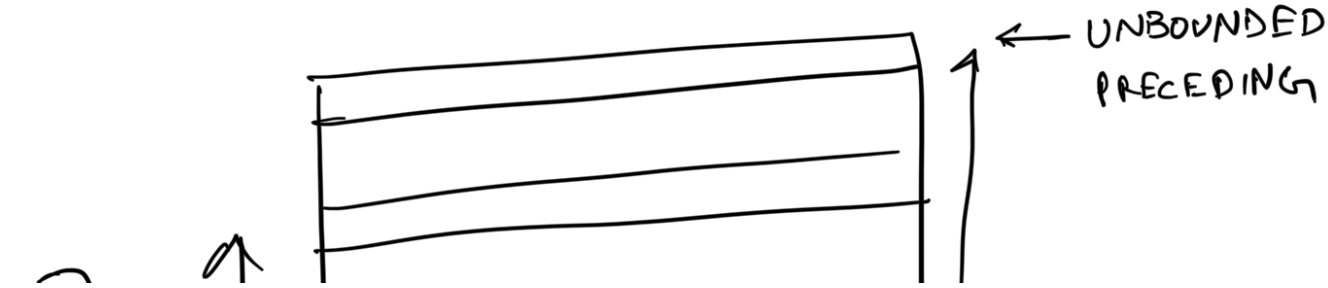
Moving

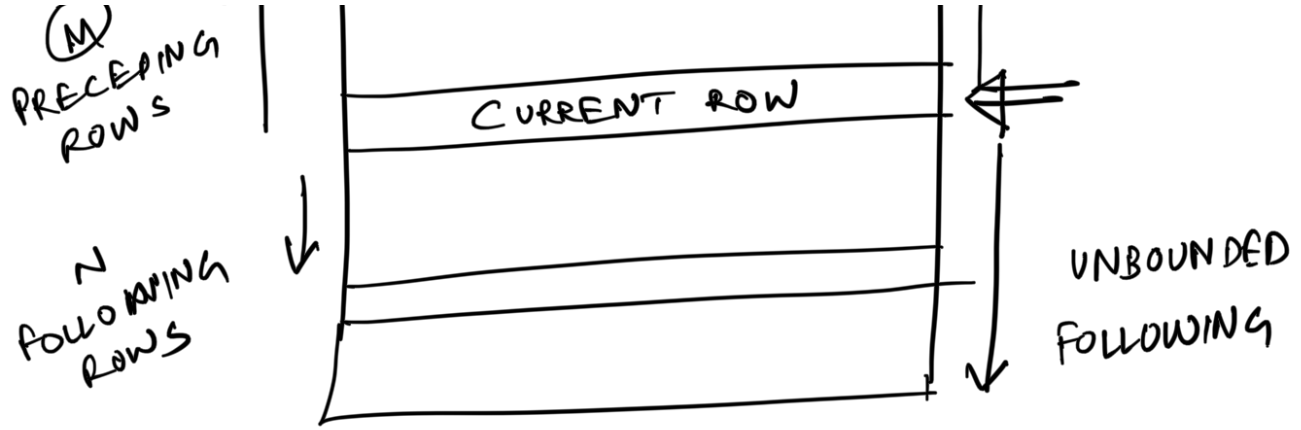




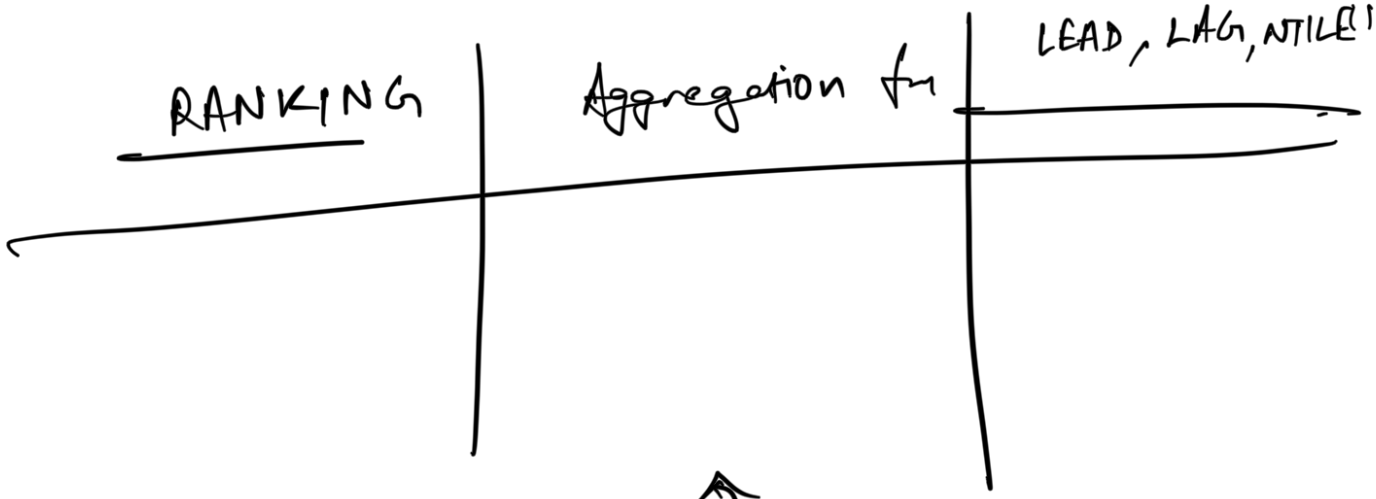
d. Calculate the monthly cumulative sales over the entire sales table.

WINDOW FRAMES





Q Calculate the moving average on a window frame of 1 preceding + 1 following.



(2)

sale	LAG	LEAD
200	NULL	400
400	NULL	300
300	200	200
200		NULL

Syntax:

LEAD (expr, N, default) OVER ( )

Q. Display each vendor's booth assignment for each market_date alongside their previous booth assignments.

Follow up

⇒ Find out the vendors who are either new or are changing the booth.

To make the transition smooth.

Q NTILE ( )

Employees

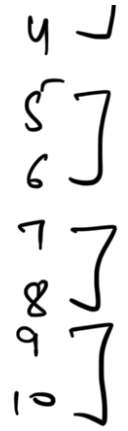
Quartile

E_ID	Salary	
1	25	
2	50	
3	75	
4	80	
.		
.		
10	45	

10

NTILE (5)

1
2
3



NTH_VALUE(), FIRST_VALUE(), LAST_VALUE()

NTH_VALUE (expr, N) OVER (___)