

WINDOW FUNCTIONS

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

} SELECT
 vendor_id,
 MAX(original_price) AS most_expensive
 FROM vi
GROUP BY vendor_id

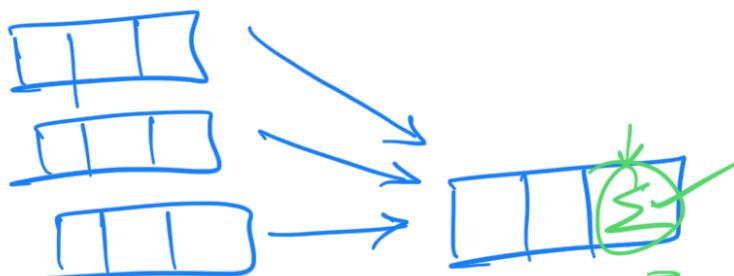
Orders

Date	C-ID	Art
21	1	4
21	1	
21	1	
21	2	
21	2	
22	1	
22	2	
21 - 1	2	
21 - 2		
22 - 1		
22 - 2		

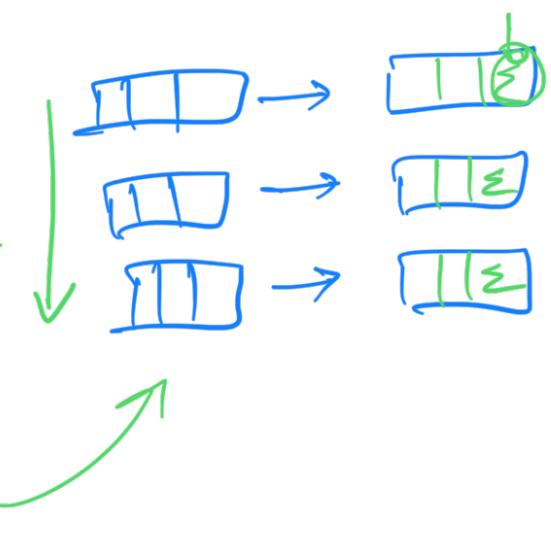
Q: Rank the products in each vendor's inventory. Expensive products should get a lower rank.

	v-id	X	P_id	price(j)	Rank
Not collapsing	1	21	4	5	2
X	1	22	4	9	1
collapsing	1	22	7	4	3
	1	22	8	3	4
	2	23	2	2	5
	2	23	5	1	6
	2				

Aggregate Functions



Window Functions



→ ROW_NUMBER() →

Window :

SELECT

Vendor-id,
Market-date,
Product-id,
Original-price)

→ Row-number()

V.id	price	Rownum
1	5	1
1	5	2
1	4	3
2	2	1
2	2	2

OVER (PARTITION BY vendor-id
ORDER BY original-price
DESC)

① Partitioning

② Sorting

Date	V.id	P.id	Price	Avg(price)
21	1	2	5	3
21	1	2	1	3
21	2	3	2	3
21	3	4	4	3
21	4	2	3	3

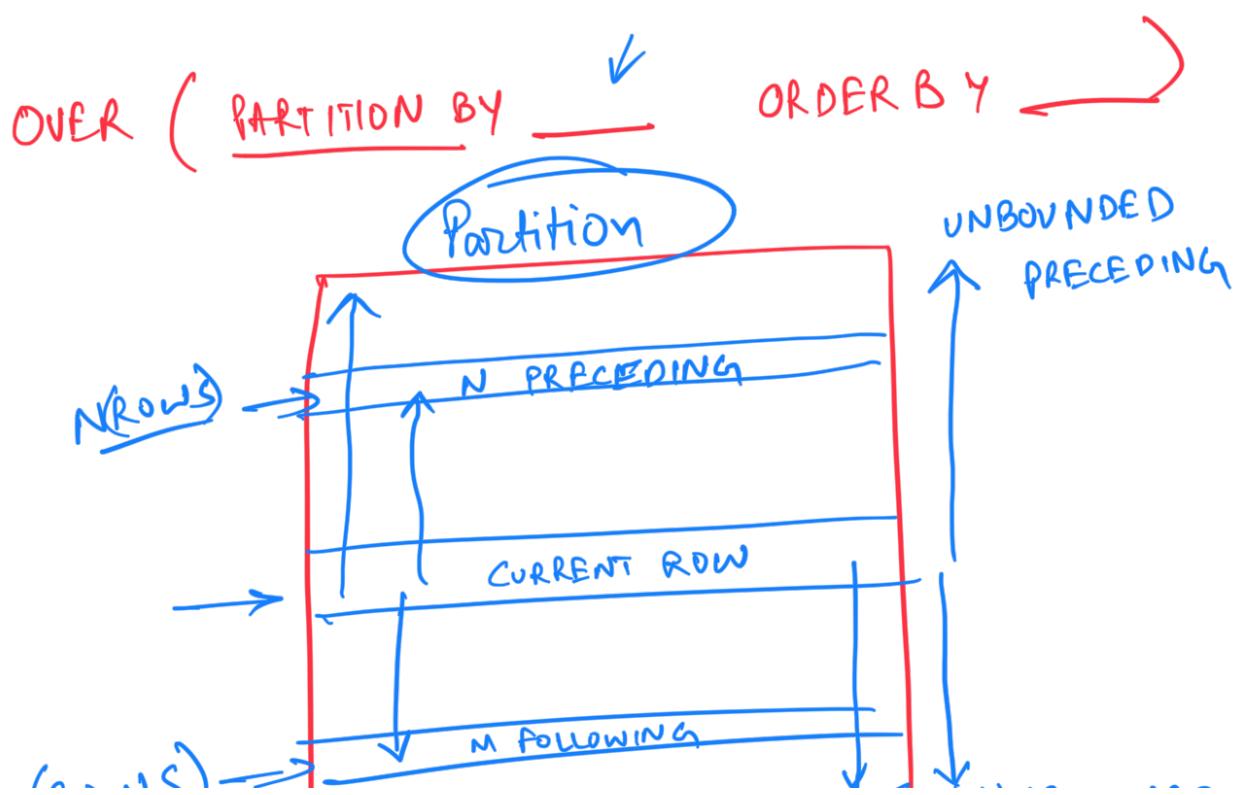
Avg(Original-price) OVER (_____)

Partition By market_date DRP Br By
date —

① ↗ ↘

	Date	v-id	p-id	Count	
1	21	1	1	3	X
	21	1	2	3	
	21	1	3	3	
	21	2	4	3	
2	21	2	5	3	
	21	2	2	3	

② ↗ ↘



M(ROWS)



UNBOUNDED FOLLOWING

Reliance		S-DAY
Date	closing-price	Moving Avg.
21	1900	NA
22	1905	NA
23	1910	NA
24	1907	AA
25	1900	1906
28	1910	—
29	1915	—
30	:	—