

Agenda

a. Window Analytical Functions

- i. Row Number
- ii. Rank
- iii. Dense Rank
- iv. Lag Lead
- v. First Value
- vi. Nth Value
- vii. NTile

b. Date and Time Functions

① Aggregation window function

→ SUM \downarrow \rightarrow Salay) over (Partition by
employee
order by
date)

fn

Q1

Analytical window fn

a) ROW number

	PK	Emp_id	Name	Sal	Dept_id	row_no
P1	101	Antie	100	10	1	1
P2	102	Beta	200	10	2	2
P3	103	C	250	10	3	3
P4	105	D	300	10	4	4
P5	108	E	150	10	5	5
P6	110	Anant	250	11	6	6
P7	201	Baba	175	11	7	7

employee

Q2

over()

dept_id

row_no

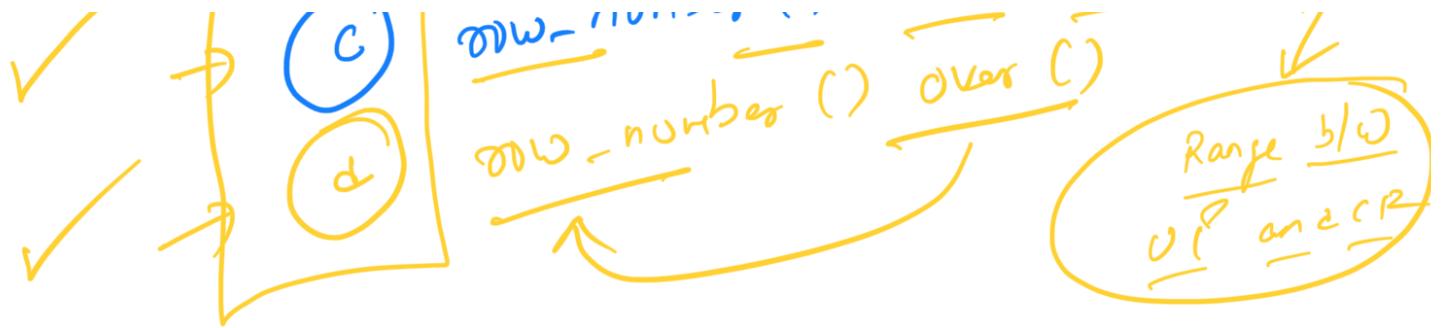
over(partition by dept_id)

row_number()

X a ~~DDW-number()~~ over (partition by emp-id)

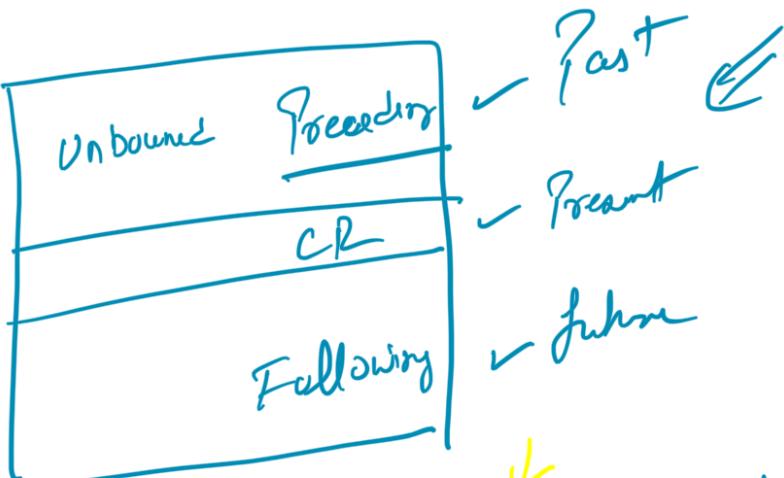
X b ~~DDW-number(emp-id) over ()~~

X c ~~number() over (order by eid)~~



Range $\frac{b/w}{U/P \text{ and } CR}$

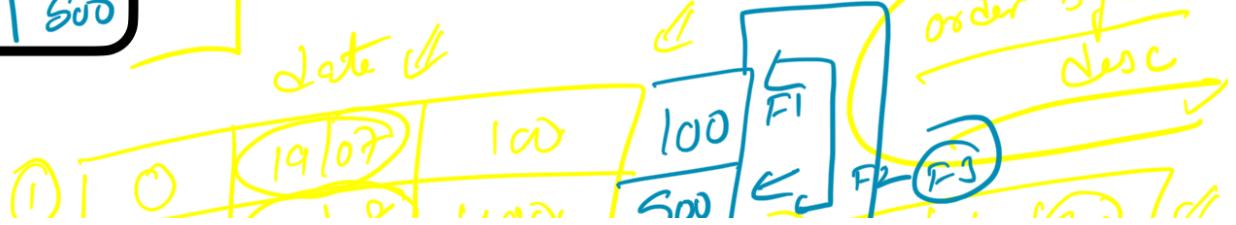
- ✓ Range between Unbounded
 ⇒ Proceeding and Current Row)
 ⇒ over (Position by ~~exp-name~~
order by Date



①
 name date sale
 O 19/07 100
 O 19/08 400
 O 19/09 300
 T 19/07 200
 T 19/08 400
 T 19/09 300
 T 19/10 500

②
 Sum(Sale) over (partition by name
 order by date)

⇒ Range between
 F1 and F2



Q1 ↗

0	(19/08)	400	...	F1
0	(19/09)	300	...	F2
0	(19/09)	300	800	F3
0	(19/09)	300	800	...

$$UP = 0 + 100$$

$$UP = 100 + CR = 400$$

$$F3 = \frac{UP}{CR} = \frac{100, 400}{300}$$

Q1

100	1900
400	1900
300	1900
200	1900
400	1900
500	1900



$$UP = 0 \\ UF = \frac{400, 300}{200, 400} \\ 500$$

$$CR = 100 \\ UP = \frac{300}{400} | \frac{400}{100} \\ UF = \frac{400}{500} | CR = 200$$

Range between UP and UF
Sum(sales)

↓

N	d	S.	Name	Date	Sale	X
O	19/09	200		19/07	100	-
O	19/09	200		19/08	200	-
O	19/08	200		19/09	200	-
I	19/08	400		19/07	300	-
I	19/07	100		19/08	400	-
O	19/07	300		19/09	200	-

400
400

$$UP = 0 \\ CR = \frac{200, 200}{0} \\ UP = 0 \text{ or } CR = 200$$

Sum(sales) ←
over ←
(order by date desc)

Range b/w UP and CR
T1, ..., Tn

$$UP = 0 \text{ or } CR = 200$$

Diagram illustrating a query execution plan for a table with columns: Date, OrderID, Quantity, and Price.

The table has 4 rows and 4 columns. The first column is Date (O/T), the second is OrderID (19/08/09/10/11), the third is Quantity (200/400/200/400), and the fourth is Price (100/100/100/100).

Range: Rows 1, 2, 3, 4.

Rows: Row 1, Row 2, Row 3, Row 4.

CR: Total quantity = 100 + 100 + 100 + 100 = 400.

UP: Total price = 100 + 100 + 100 + 100 = 400.

IF: Total quantity = 200 + 200 + 200 + 200 = 800.

Sum: Total price = 100 + 100 + 100 + 100 = 400.

Find query → Sum(scl) over (order by date desc b/w UP and IF)

Range

Diagram illustrating a query execution plan for a table with columns: Date, OrderID, and Price.

The table has 7 rows and 3 columns. The first column is Date (O/T), the second is OrderID (19/03/02/02/01/01), and the third is Price (100/200/300/200/400/500).

Range: Rows 1, 2, 3, 4, 5, 6.

Rows: Row 1, Row 2, Row 3, Row 4, Row 5, Row 6.

CR: Total price = 100 + 200 + 300 + 200 + 400 + 500 = 1500.

UP: Total price = 100 + 200 + 300 + 200 + 400 + 500 = 1500.

IF: Total price = 200 + 300 + 400 + 500 = 1000.

Range = logical
Rows = physical

$$UP =$$

$$CR =$$

$$UP = \frac{100}{2nd} \quad CR = \frac{200}{3rd}$$

$$IF = \frac{300}{4th}$$

$$UP = 300, 200, 300$$

$$CR = 400$$

$$IF = 500$$

Q1

O	19/03	200
O	19/02	400
O	19/01	500
T	19/03	300
T	19/02	200
T	19/01	400

1	600
2	
3	

Sum (Salary)

OVER (PARTITION BY

~~EMP_NAME~~order by
date descRows between
UP and CR

PL

$$\text{UP} = 200 \\ \text{CR} = 400$$

$$\text{UP} = 200, 300 \\ \text{CR} = 400$$

100 / Rows

①
②
③

O	19/03	100 -	600
O	19/03	200 -	600
O	19/03	300 -	600
O	19/03	400 + 400	

Avg

Sum()

① over
(CPB emplo
OB date);Avg (jd) $\Rightarrow \frac{9}{1000}$ Range
Row

Row-number	Rank	Dense rank
1	1	1

② ②

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
1	1	100	1	200	2	300	3	300	4	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
1	1								2																			
1	1								3																			
1	1								4																			
2	2									5																		
3	3									6																		
1	1																											

Row number ()
 → over () ; ✓
 → Row-number()
 over (partition by
 sel)
 → Row-number()
 over (partition by
 sel order
 by name);

Rank
 → over
 order by
 name)
 → over (order
 by sel)

→ over (order by
 sel)

Range b/w
 UP and CP2

→

C1	C2
4	2
2	3

data

C1	C2
1	4
2	3

3	1
1	4

3	2
4	1

→ ~~Select C1, C2 from data order by C1 asc, C2 desc;~~

Select C1, C2 from

1	4
2	3
3	1
4	2

Select C1, ~~row-number()~~ over(order by C1) as X
From data
order by C1) T1)

T1.x

4	1
2	2
3	3
1	4

Join

Union

T2.x

Select C2, ~~row-number()~~ over(order by C2 desc)
from data
order by C2 desc)

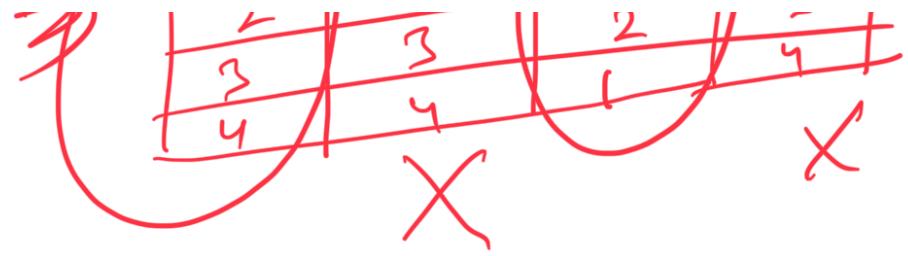
2	4
3	3
1	2
4	1

1
2
3
4

On T1.x = T2.x

C1	x	C2	x
1	1	4	1
2	2	3	2

4
 3
 2
 1



(2)

			0	null	
1	w	1 1	0	w	1
1	w	2 2	0	w	1
1	w	3 3	3	w	1
1	L	4 1	1	L	0
1	w	5 4	1	w	1
1	w	6 5	1	w	1
1	w	7 6	1	w	1
1	w	8 7	1	w	1
1	w	9 8		w	1

→ 3
↑

(4)