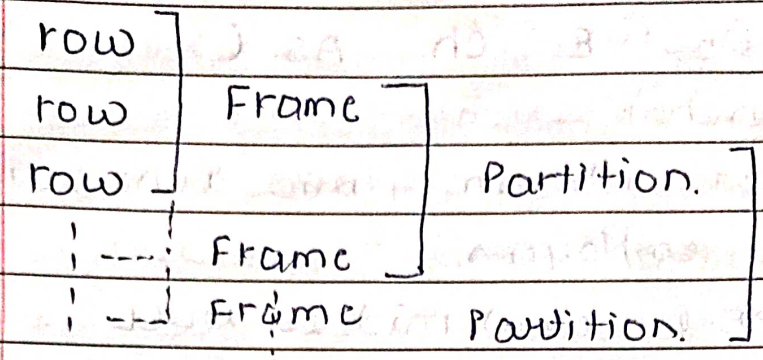
advance # window fcn
concept

DOMS | Page No.
Date / /

Window Functions.



- running rows makes a frame
  1 row = 1 frame
  2 row = frame 2
  3 row = frame 3 etc
  or it can be depend on condition.
  but the point is 0 — ith current frame
  0 — (i+1)th next frame.

- window fcns are used to perform calculations across a set of table (window) that are related to current row

OVER	PARTITION BY	ORDER BY
↓	↓	↓
Defines the window to operate on.	like a group by for window over which other fcns can be applied.	order rows with in partition.

window fns

DOMS | Page No.
Date / /

1. RANK()
   - assign rank to each row in partition.
   - assign same rank to duplicates but skip next values.

   Eg:

value:	10	20	20	43
RANK:	1	2	2	3

2. DENSE_RANK()
   - same as Rank fn) but do not skip values. It goes like 1 2 2 3

3. ROW_NUMBER()
   - assign each row a unique row number

4. Aggregate fns can also be used using OVER clause to apply them over a partition.

5. LAG() — access data from previous rows.
   - LAG(column, N)

   LEAD() — access data from next rows.
   - LEAD(column, N)

6. FIRST_VALUE(column)
   LAST_VALUE(column)
   NTH_VALUE(column, N)
   — respective value from each partition.

7. NTILE(N): SQL try to divide in equal partition.

window fun) # concept
frame clause

DOMS | Page No.
Date / /

8. CUME_DIST() | PERCENT_RANK()

↓ | ↓

It gives the cumulative distribution of row within set of rows. i.e. How much distribution current row till now holds. | It gives the relative rank of row within partition i.e. How many rows are below or higher relative to current row.

FORMULA: (No. of row with value ≤ current) / total rows. | (current ROW NO. - 1) / (total - 1)

Range: 0 < value ≤ 1 | 0 ≤ value ≤ 1

Note: In case of duplicate records, row of last record is considered.

9. FRAME Clause.

- FRAME word is not used.
- used at end of OVER clause as last arguments.

Syntax:

(A) BETWEEN (B) PRECEDING AND (C) Following.

window fns
frame clause

DOMS | Page No.
Date / /

(A) RANGE : take account of duplicates.
eg: last 2 : 5 [4 4] 3 2
ROWS : specific to rows only.
eg: last 2 : 5 [4 4] 3 2

(B) → unbounded : from starting of partition.
→ n : from last n of partition. (range/rows)

* (there are many more)

(C) unbounded : till end of current partition.
n : next n (range / rows) of current partition.
current row : till current row (omit word following)

WHY?

Because in fns like last_value()
frame considers the running rows.

	C1	C2	O	We want
Partition.	A	1	1	4
	B	2	2	4
	C	3	3	4
	D	4	4	4

F1 : A ∴ O = 1
F2 : AB ∴ O = 2
F3 : ABC ⋮
F4 : ABCD ↓

window fcts
concept

DOMS | Page No.
Date / /

as these fcts are applied frame by frame. even though they are applied over partition, as per saying, but executed frame by frame.

By default FRAME clause of LAST_VALUE is
* Range between unbounded preceding and current row.
∴ it will generate wrong results.
Hence, we need frame clause.

Examples:

1. Select id, name, salary,
   RANK() OVER (ORDER BY salary desc)as R
   FROM employees;

2. Select id, date, amount,
   AVG(amount) OVER (
   order By date
   ROWS between 2 preceding and current row
) as moving_avg
   FROM orders;

3. Select id, salary,
   PERCENT_RANK() OVER() as PR
   FROM employees.

window fun
window keyword

Note: if you have common window condition. i.e. inside over brackets. than you can omit common conditions and write it in window keyword fun

Example:

At end of query mention.

:

WINDOW W as (conditions)

URC.

Query.

```
Select *,
FIRST_value (Pn) OVER W as FV,
Last_value (Pn) OVER W as LV
From products.
where PC = 'Phone'
WINDOW W as ( partition by PC
              order by price desc
              range between 2 preceding
              and 2 following );
```