1

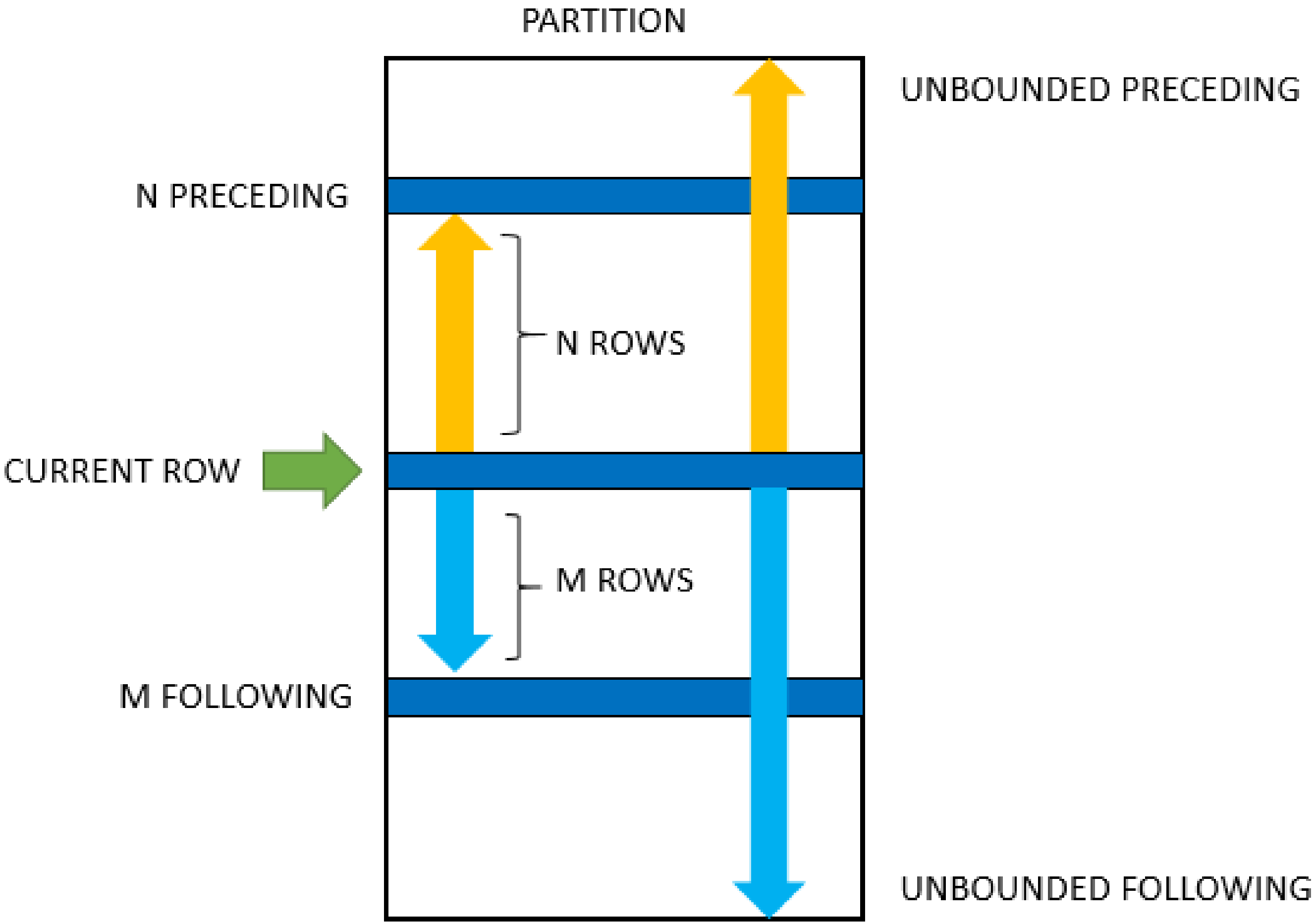
Syntax of Partion, Order, frame:

PARTITION BY expr1, expr2, ...

ORDER BY expression [ASC | DESC]
[NULL {FIRST| LAST}] ,...

{ RANGE | ROWS } BETWEEN
frame_start AND frame_end
or
{ RANGE | ROWS } frame_start

frame_start-
N PRECEDING, UNBOUNDED
PRECEDING , CURRENT ROW
frame_end-
CURRENT ROW, UNBOUNDED
FOLLOWING, N FOLLOWING


PARTITION
UNBOUNDED PRECEDING
N PRECEDING
N ROWS
CURRENT ROW
M ROWS
M FOLLOWING
UNBOUNDED FOLLOWING

2

Use cases

Running totals:

```
SELECT
   sale_date,
   SUM(amount)
   OVER (PARTITION BY EXTRACT(MONTH FROM sale_date)
                      ORDER BY sale_date) AS running_sales
FROM sales;
```

Window functions can be used to calculate running totals. This can be useful for tracking sales over time or for identifying trends.

sale_date	running_sales
2023-01-01	100.50
2023-01-15	175.70
2023-02-02	250.80
2023-02-18	376.20
2023-03-05	300.00
2023-03-20	450.75

https://dbfiddle.uk/tGl-q8lj

Moving averages:

```sql
SELECT
  stock_date,
  closing_price,
  AVG(closing_price) OVER (ORDER BY stock_date
                           ROWS BETWEEN 2 PRECEDING AND CURRENT ROW) AS moving_average
FROM
  stock_prices;
```

Window functions can be used to calculate moving averages. This can be useful for smoothing out data or for ***identifying trends***
https://dbfiddle.uk/ua1-Y5Gi

3

ROW_NUMBER()

employee_name	salary	department_name	max_salary	avg_salary	s_no
Jane Smith	6000.00	Finance	6000.00	5500.000000	1
John Doe	5000.00	Finance	6000.00	5500.000000	2
Emily Davis	4500.00	Sales	4500.00	4250.000000	1
Michael Johnson	4000.00	Sales	4500.00	4250.000000	2
Robert Miller	5200.00	Marketing	5500.00	5233.333333	1
Sarah Wilson	5000.00	Marketing	5500.00	5233.333333	2
David Brown	5500.00	Marketing	5500.00	5233.333333	3

Here s_no is a generated by using the ROW_NUMER()

https://dbfiddle.uk/-G4fCgWJ

Rank()

employee_id	employee_name	department	salary	department_rank
2	Jane Doe	Sales	6000.00	1
3	Mark Johnson	Finance	7000.00	1
5	Michael Wilson	HR	4500.00	1

GOAL: To fetch the employee details who got paid highest among his dept

https://dbfiddle.uk/mLH8wley

DENSE_rank()

employee_name	department	salary	department_rank	department_dense_rank
Robert Johnson	Finance	7500.00	1	1
Mark Johnson	Finance	7000.00	2	2
Jessica Davis	Finance	7000.00	2	2
Emily Davis	Finance	5500.00	4	3
Jennifer Smith	HR	5000.00	1	1
Michael Wilson	HR	4500.00	2	2
Adam Wilson	Sales	6500.00	1	1
Jane Doe	Sales	6000.00	2	2
John Smith	Sales	5000.00	3	3
Sarah Thompson	Sales	5000.00	3	3

DENSE_RANK() example:
https://dbfiddle.uk/Z34ZUlC2

LEAD() & LAG()

LAG() & LEAD() example: https://dbfiddle.uk/Z34ZUtC2

List of window functions:

- ⭐ Ranking Functions
  - row_number()
  - rank()
  - dense_rank()
- ⭐ Distribution Functions
  - percent_rank()
  - cume_dist()

⭐Analytic Functions

- lead()
- lag()
- ntile()
- first_value()
- last_value()
- nth_value()

Aggregate Functions:

- avg()
- count()
- max()
- min()
- sum()