Window functions:

Start @ 9:03

Recap:

Joins

L, R, F, O.

Self join

Cross join

Inequi Join

Multi-table join

Agenda:

1. HR Database
2. Window fn.
3. Types of window fn
   - Row_number()
   - Rank()
   - D.Rank()

Agg. window fn.

Ntile

Sales for each vendor.

V.id	Prod.id	Sales
A	P1	10
A	P2	15
A	P3	20
B	P4	8
B	P10	10

5

Vid	Sum(Sales)
A	45
B	18

2

↓ rows are collapsed

Select sum(Sales)
from...

group by v.id.

What if I want the agg to be in each row without collapsing?

because of window fn.

v.id	Prod.id	Sales
A	P1	10
A	P2	15
A	P3	20
B	P4	8
B	P10	10

⇒

v.id	Prod.id	Sum(Sales)
A	P1	45
A	P2	45
A	P3	45
B	P4	18
B	P10	18

What is the % Share of each month to total sales.

Month	Sales	Sum(Sales)	% Share
Jan	4	10	4/10 = 40%
Feb	5	10	5/10 = 50%
Mar	1	10	1/10 = 10%

Sal > Avg. Sales → in each dept

empid	Dept	Salary
1	A	10
2	A	20

find avg. in each dept

dept	Avg(Sal)
A	26.66

3	A	50
4	B	15
5	B.	25

B	20

Join back to the main table

empid	Dept	Salary	Avg in each dept
1	A	10	26.66
2	A	20	26.66
3	A	50	26.66 → 50 is > 26.66
4	B	15	20
5	B.	25	20. → 25 > 20

Window fn:

- Value
  - First value
  - last value
  - Nth value
  - lag
  - lead
- Ranking
  - cume_dis()
  - percent_rank()
  - Ntile()
  - Rank()
  - Dense Rank()
  - Row Number()
- Agg.
  - Sum
  - Count
  - Avg.
  - Min
  - Max

What is agg.?

Given a series of values, the output is a single value

[5, 10, 15, 20]

Sum → 50
Min → 5
Max → 20
Avg → 12.5
Count → 4

Syntax :

Function over (group to apply and what order)

Row-number over (Partition by dept-id order by salary)

- Row-number ↓ what func to apply.
- over: apply this on this
- Partition by dept-id ↓ on what group to apply (optional clause).
- order by salary ↓ in what sequence. (mandatory for ranking clause)

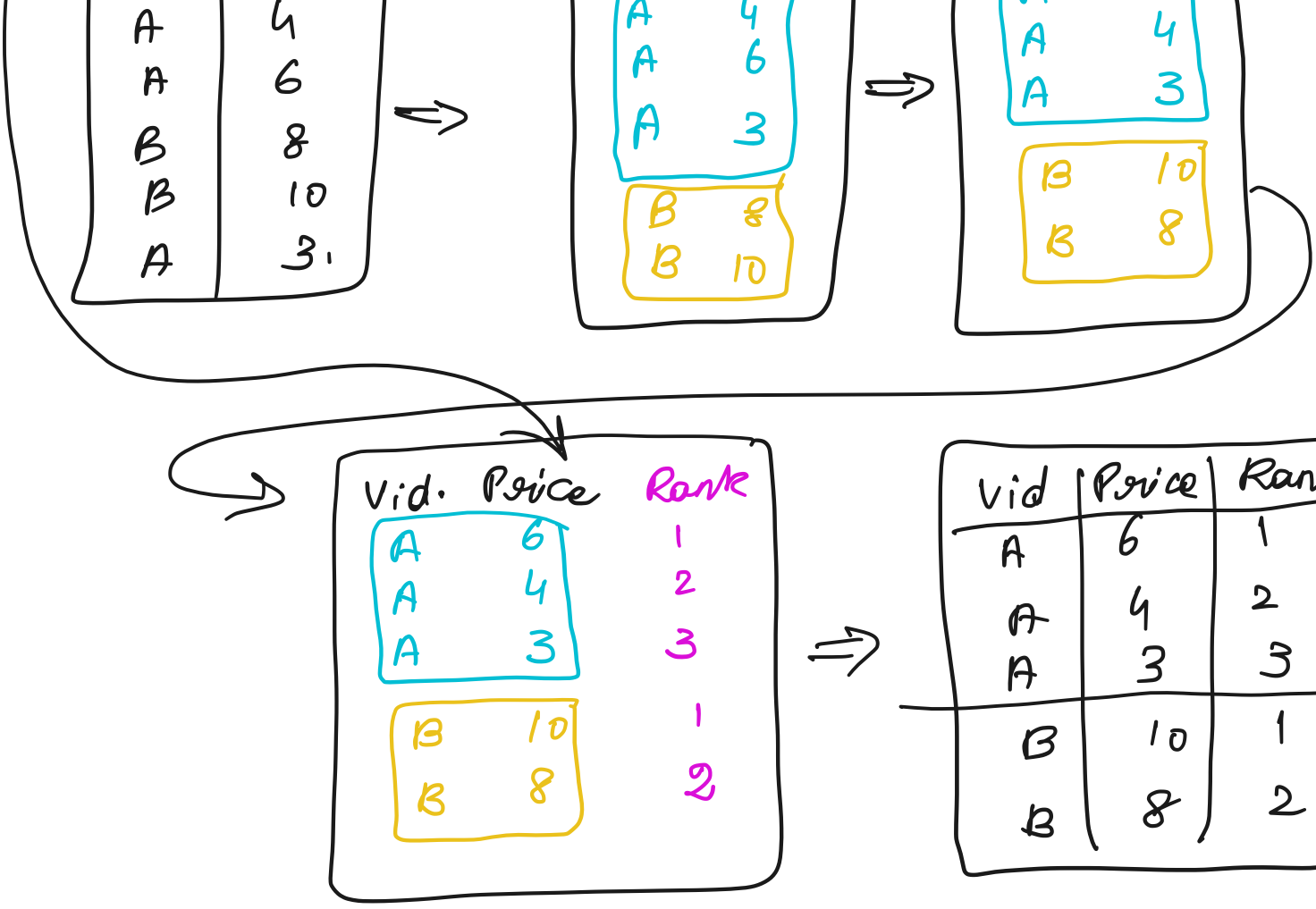
Rank the price in desc order within each vendor.

Row-number ( ) over (Partition by v.id order by price desc)

Vid	Price

Vid	Price

Vid.	Price
A	6



Row_number() over(order by price desc)

↓

The entire table is 1 partition

Vid	Price
A	4
A	6
B	8
B	10
A	3

⇒

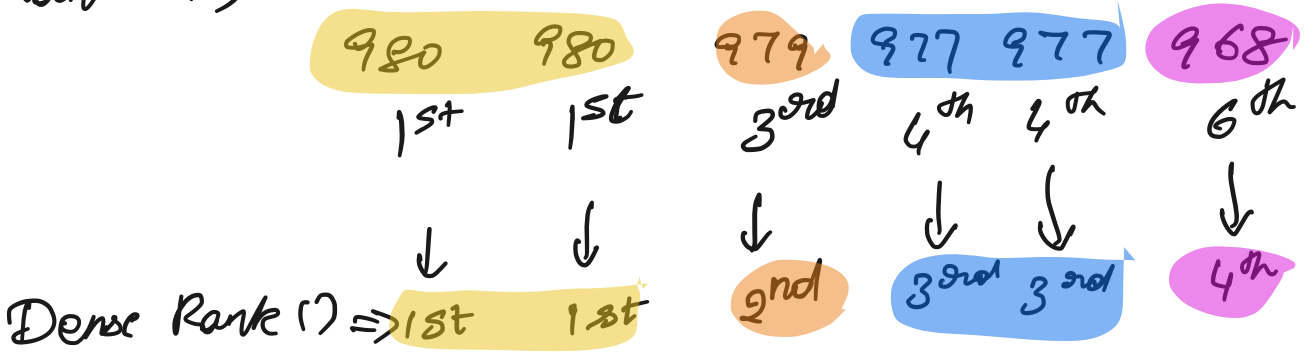
Vid	Price	Rank
B	10	1
B	8	2
A	6	3
A	4	4
A	3	5

Order of execution:

From, join
Where
Group by
Agg. func
Having
Window func.
Select
Distinct
Order by
Offset
limit

Row_num() → Creates a seq. of number.

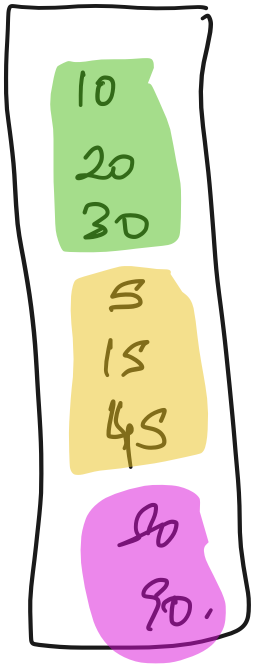
Rank()



Find the emp-id with ... highest salary

emp.id	Salary.	Row_num.	Rank	D.Rank
1	100	1	1	1
2	100	2 X	1	1
3	100	3	1	1
4	86 ✓	4	4	2
5	86 ✓	5	4	2
6	70	6	6	3

Ntile :



Create bins of people and evenly divide them into groups.

Create 3 bins.

→ Split the rows into 3 groups

$$\frac{\text{nrows.}}{\text{nbins}} = \frac{8.}{3} = 2.66.$$

2-bins will have 3 rows
1-bin will have 2 rows.

Salary of employee.

10 24 36 40 42 48 49 60 94

→ Create 4 groups based on salary

8 10 24 | 36 40 42 | 48 49 60 | 94 96 98

Ntile - splits the rows based on the count of rows and not the actual value of the column.

Syntax: for 4 bins

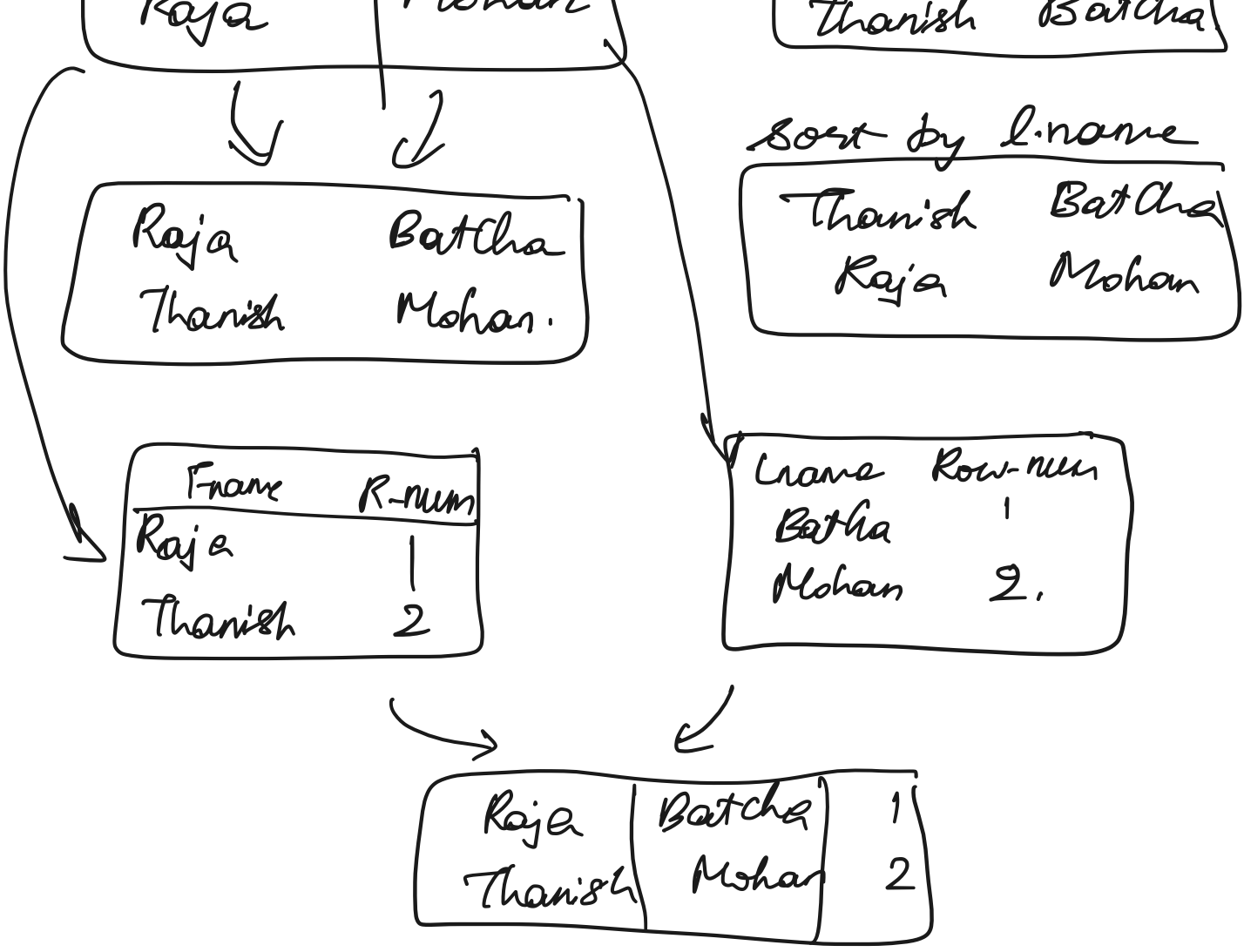
Ntile (4) over (Partition by order by <col>)

↓ optional. ↓ mandatory.

Fname	Cname
Thanish	Batch
Raja	Mohan

sort by Fname

Raja Mohan


Raja
Thanish Batcha
Sort by l.name
Thanish Batcha
Raja Mohan
Raja Batcha
Thanish Mohan.
Fname R-num
Raja 1
Thanish 2
Lname Row-num
Batcha 1
Mohan 2.
Raja Batcha 1
Thanish Mohan 2