Agenda

a. Window Function
    i. Analytical function
    ii. Aggregation function

100

Window Function

↳ SUM → 1 row

↳ SUM group by = 11 rows

↳ SUM ( ) → 100 ← Subquery

→ Table level and performs any aggregation / analytical

(1) 5th highest salary from each department :

→ Max (max (max (max)))

→ order by limit offset

→ Subquery → Time & complexity

→ Duplicates

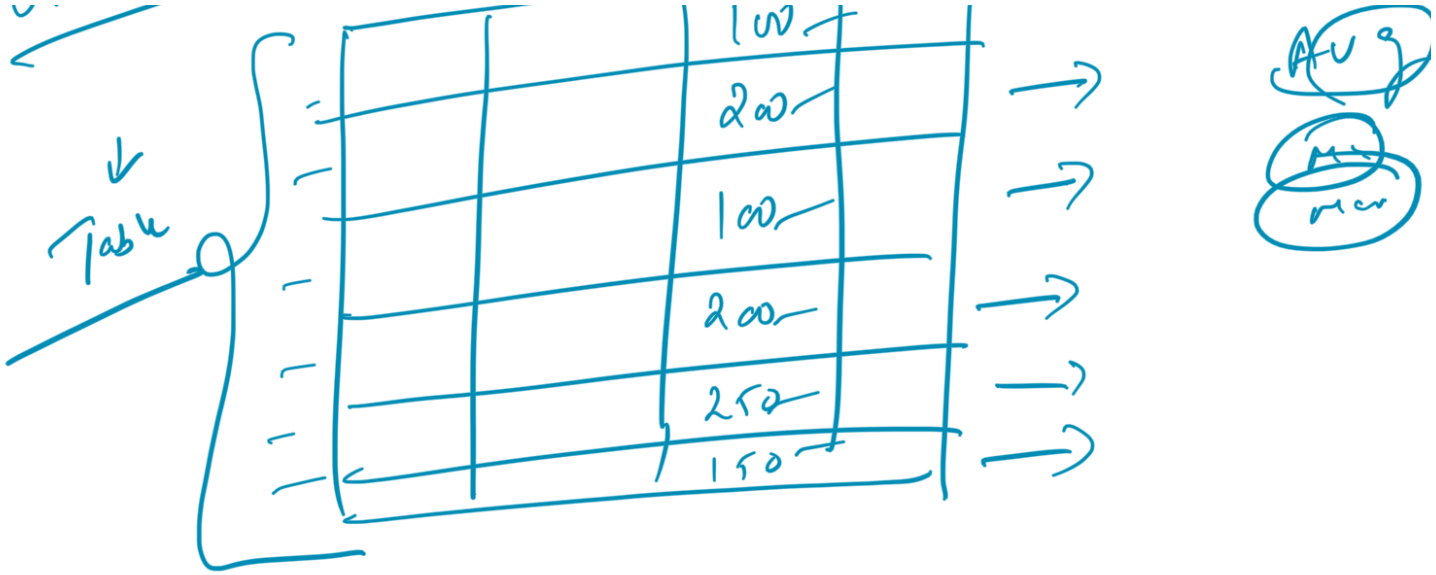
(2) OVER ( )

(i) Partition By

(ii) order by

OVER ( ) →

Sal → Employee → Sum



(a) Partition by Dept

ed	Sal	Dept	
1	100	AD	175
2	200	AD	175
3	100	AD	175
4	300	AD	175
5	900	IT	420
6	500	IT	420
7	600	IT	420

AD	175
IT	420

Avg (Sal) Dept wise

Over (partition by Dept)

Group by at row level

Sala	Deptid	Job		new_Avg
100	10	IT	261.66	300

2	2ω	10	IT	"	3ω
3	3ω	10	IT	"	"
4	4ω	10	IT	"	"
5	500	10	IT	"	"
6	100	10	AD	"	100

1600 / 6

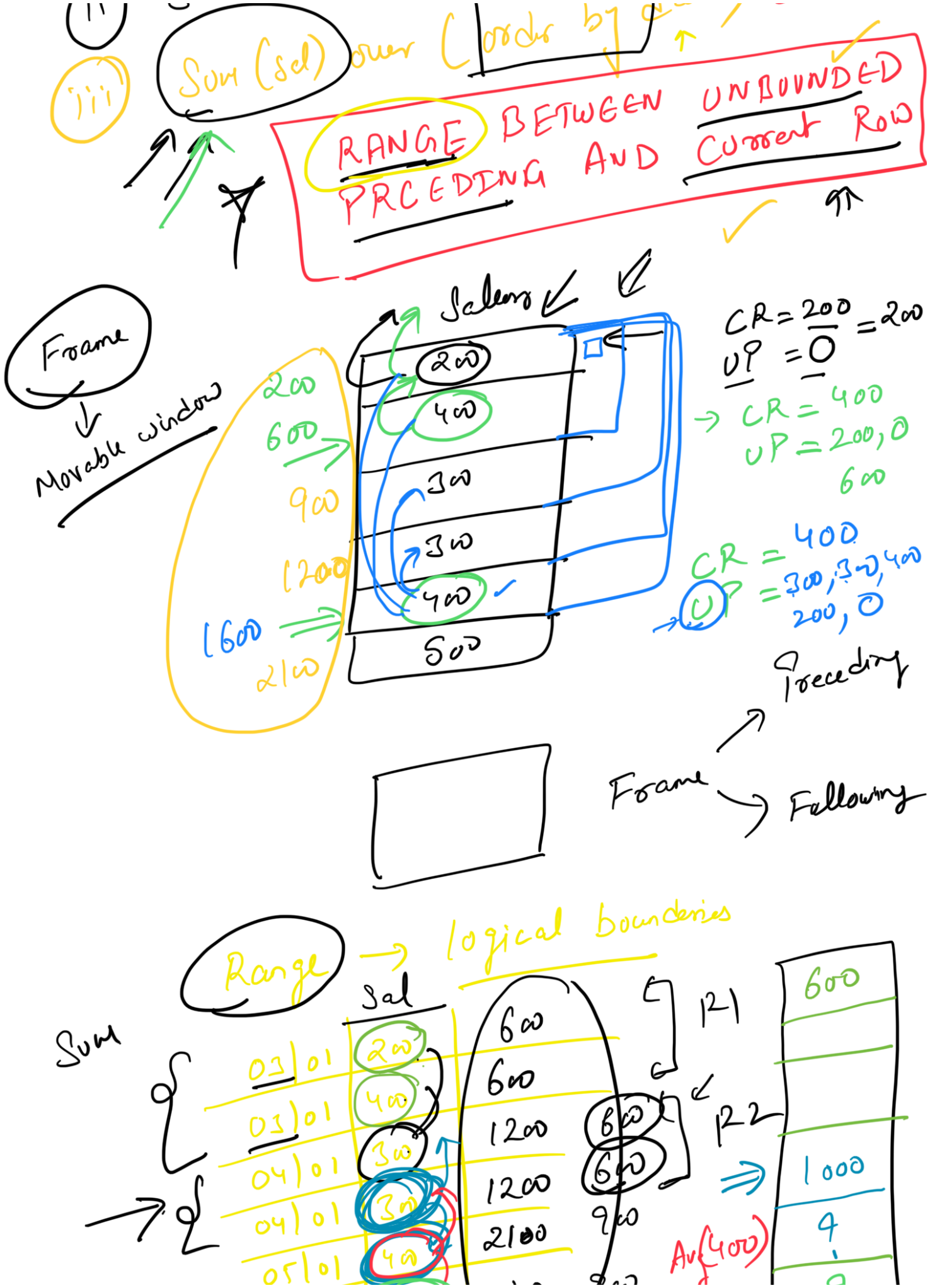
1. Avg(sal) over (Partition by dept_id) ⇒ 26666
2. Avg(sal) over (Partition by dept_id, job_id

ii) order by Sales

After Sorting

sal	date	emp	date	sal			
200	03/01	odin	2017-03-01	200	2100	900	600
400	03/01	odin	-04-01	300	2100	900	600
300	04/01	odin	05-10	400	2100	900	1200
300	04/01	thor	03-01	400	2100	1200	1200
400	05/01	thor	04-01	300	2100	1200	2100
500	05/01	thor	05-01	500	2100	1200	2100

1. Sum(sal) OVER()
2. Sum(sal) over (PD employee ... date)


Sum (sal) over (order by
RANGE BETWEEN UNBOUNDED PRECEDING AND Current Row
Frame
Movable window
Salary
200
400
300
300
400
500
200
600
900
1200
1600
2100
CR = 200
UP = 0
= 200
CR = 400
UP = 200, 0
600
CR = 400
UP = 300, 300, 400
200, 0
Preceding
Frame
Following
Range → logical boundaries
Sum
sal
03/01
03/01
04/01
04/01
05/01
200
400
300
300
400
600
600
1200
1200
2100
121
122
600
1000
Avg(400)

Sum (sal) over (order by date
Rows between 1 preceding and 1 following)

Avg (sal)

Max (sal) = 500

Min (sal) = ?

1P = 0
1F = 400
CR = 200

1P = 300
1F = 500
CR = 400

1P = 300 CR = 300
1F = 400

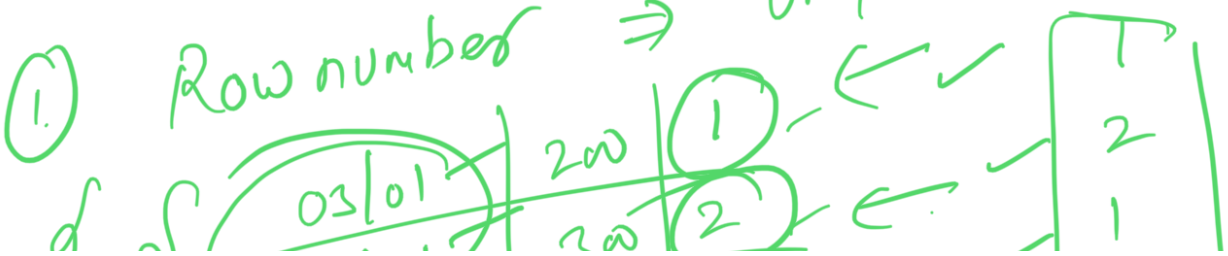
1P = 400 CR = 500
1F = 0

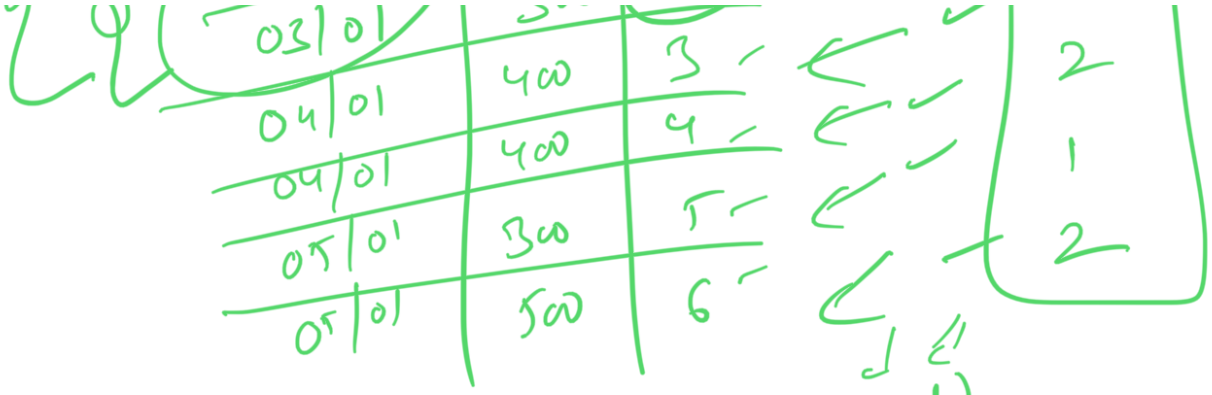
Aggregation Sum | Count | Min | Max | Avg

⇒ Analytical functions

1. Row number
2. Rank
3. Dense rank

1. Row number ⇒ unique number of records

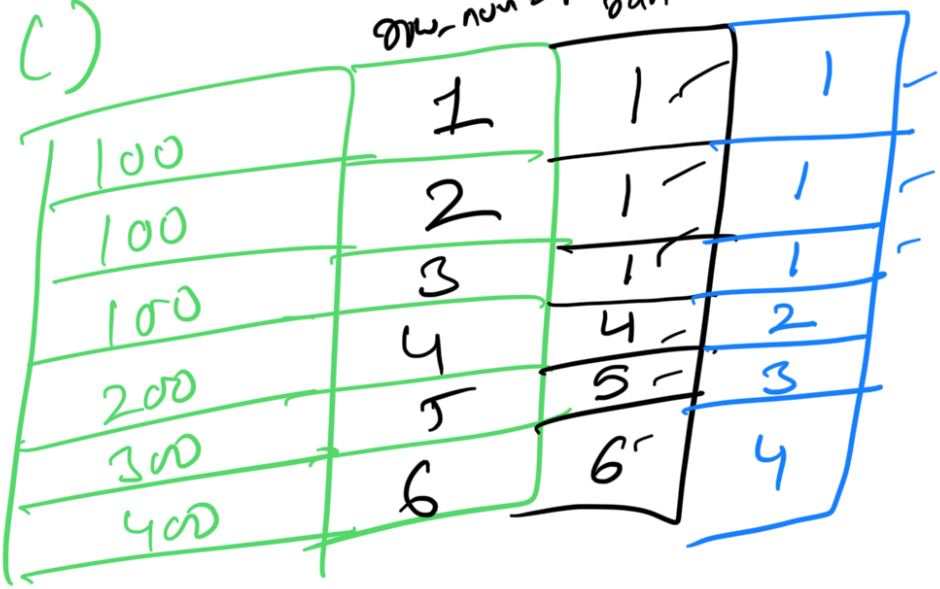




row_number() over (order by date)

Range b/w UP and CR

→ Rank()



→ Dense-rank()

Doubt clearing Session

1

① Substring ()

" Merlin 15123 "

" [illegible] 23456 "

Merlin 1 S 1 2 3
1 2 3 4 5 6 7 8 9 10 11

Substring(name, 1, 6) = Merlin

over (order by)
over (PBA order by)

Range between Unbounded Preceding and Current row

(3) Aman

employee & department
dept_id dept_id

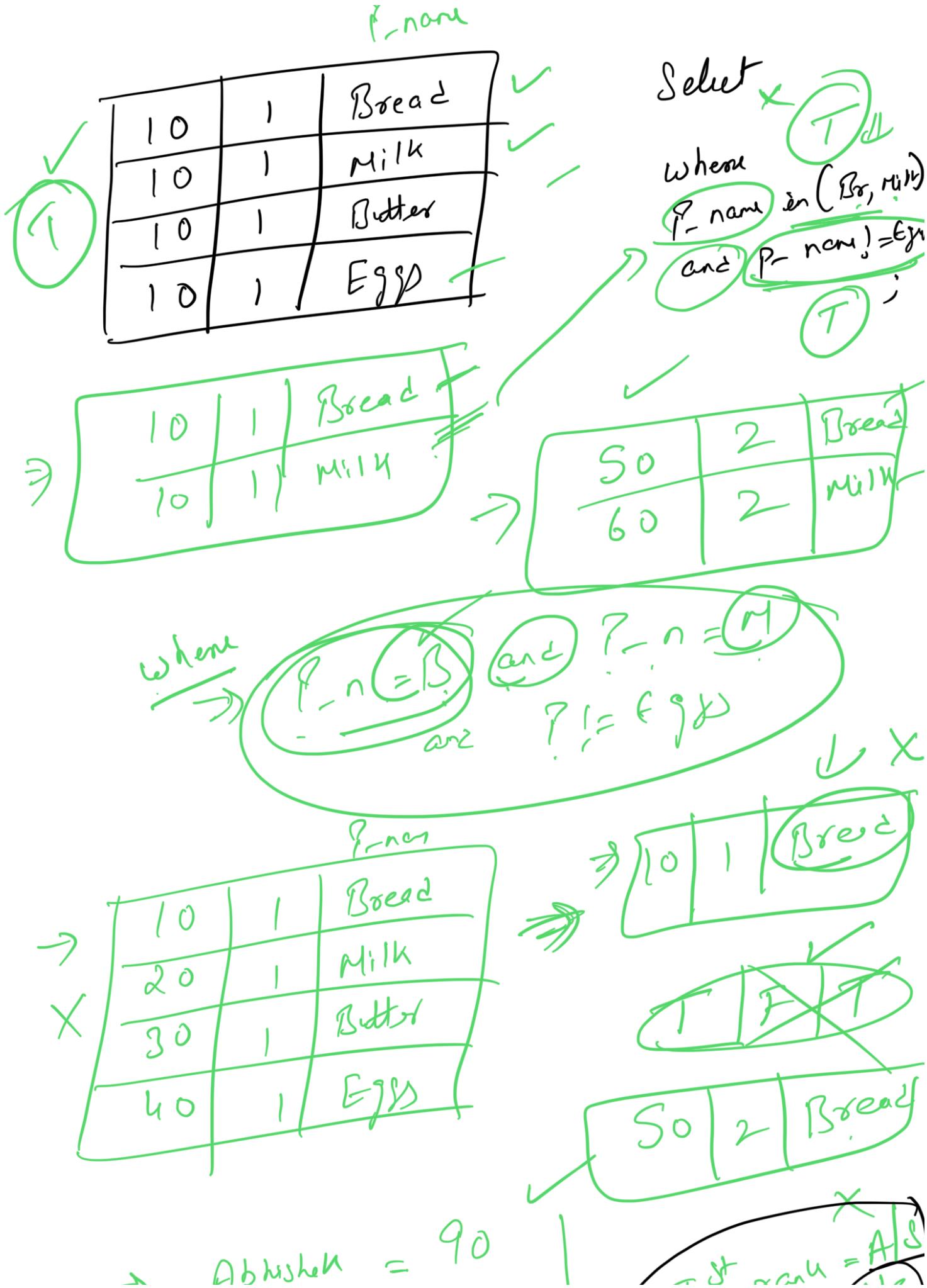
(1) Select * from
(2) employee e join department d
(3) → ON e.dept_id = d.dept_id ;
using (dept_id) ;

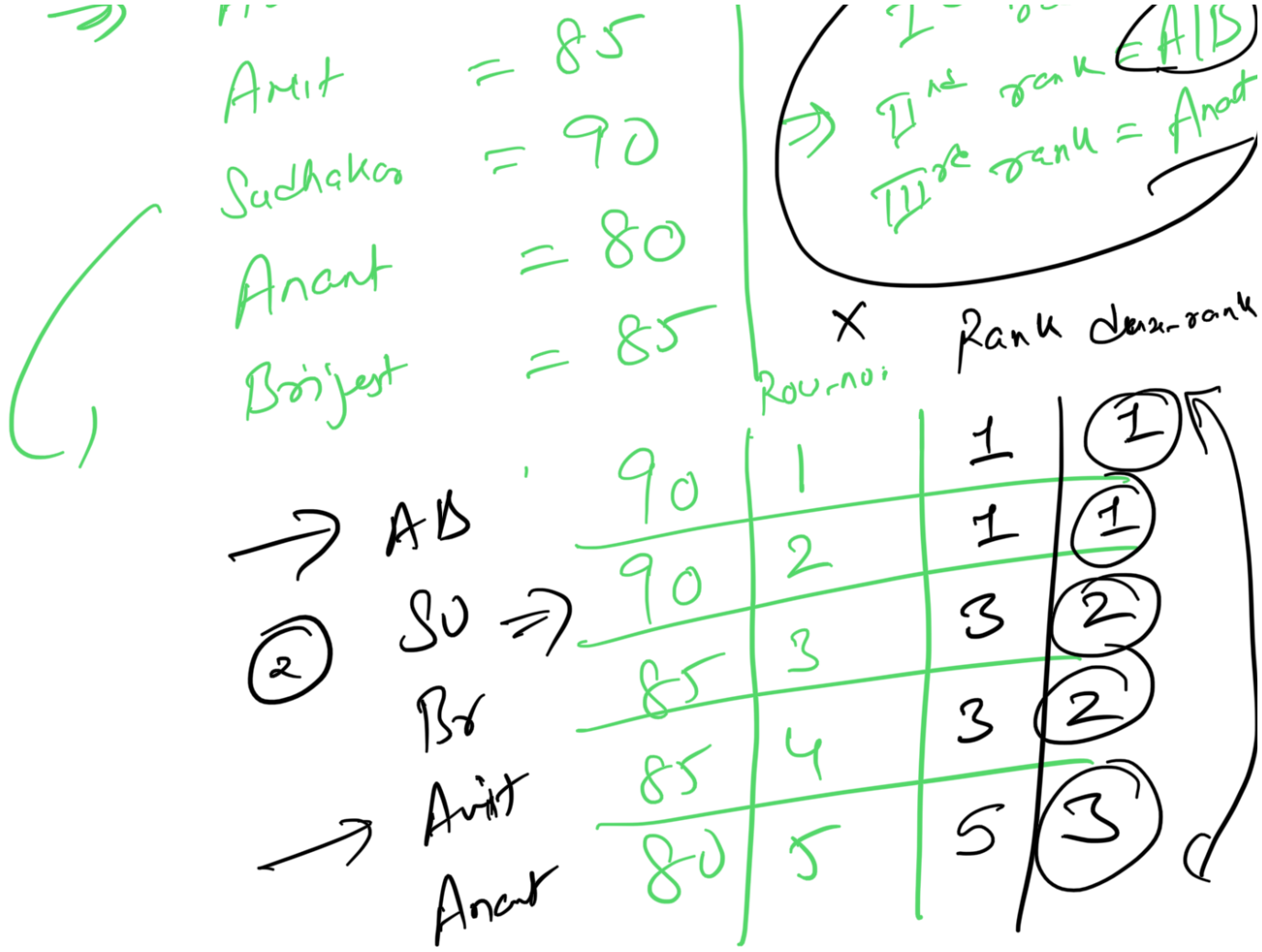
(4) Sudhakar

Cid, Cname

Breed & Milk

× Eggs


P_name
10 1 Bread
10 1 Milk
10 1 Butter
10 1 Eggs
Select
where
P_name in (Br, Milk)
and P_name != Eggs
10 1 Bread
10 1 Milk
50 2 Bread
60 2 Milk
where
P_n = B
and
P_n = M
or
P != Eggs
P_name
10 1 Bread
20 1 Milk
30 1 Butter
40 1 Eggs
10 1 Bread
T F T
50 2 Bread


Amit = 85
Sudhakar = 90
Anant = 80
Brijesh = 85
II nd rank ∈ A|B
III rd rank = Anant
X
Roll-no:
Rank
dense-rank
AB
SU
Br
Amit
Anant
90
90
85
85
80
1
2
3
4
5
1
1
3
3
5
1
1
2
2
3