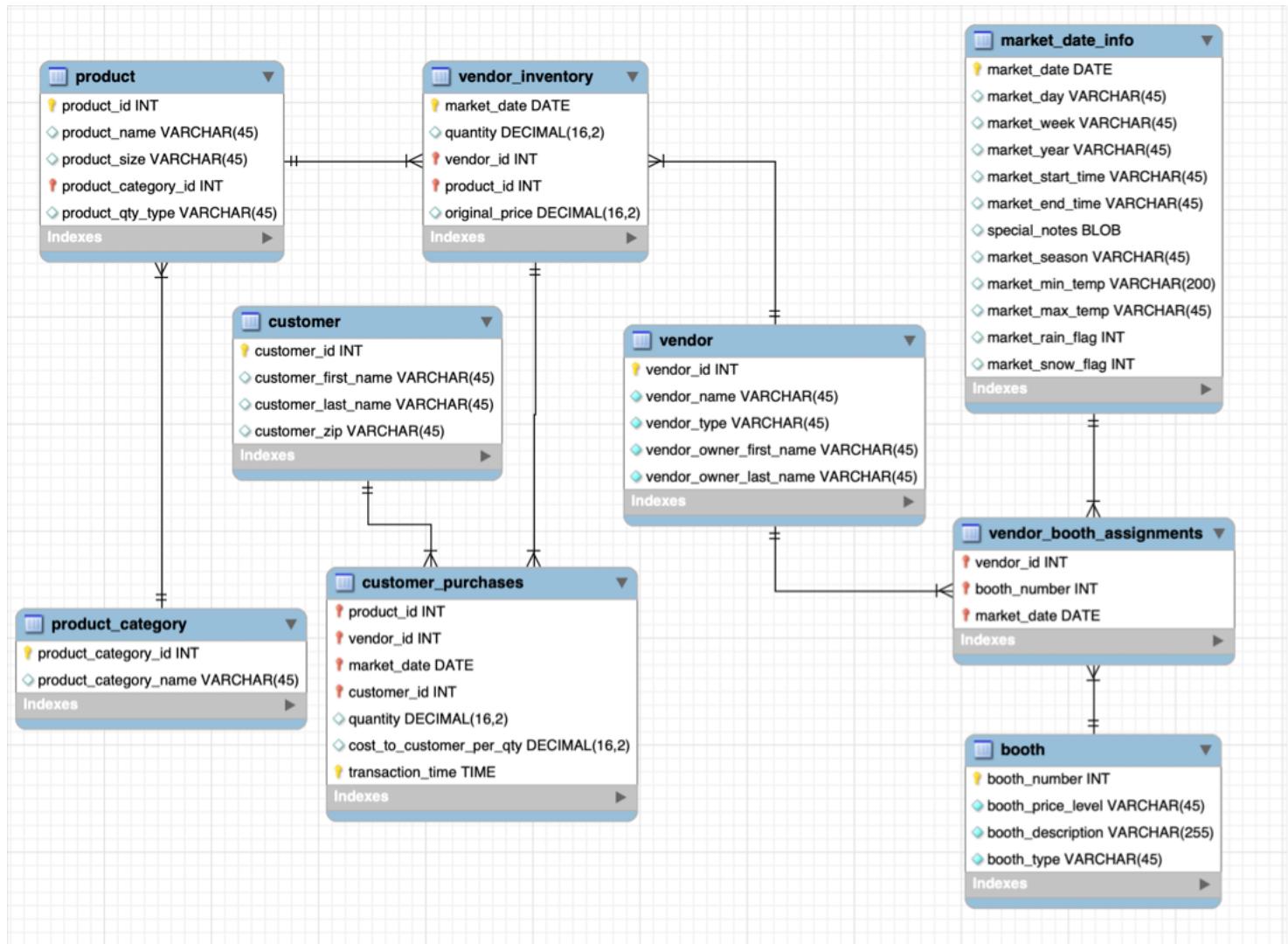


Agenda

- a. Functions in sql 
 - b. Where Clause 
 - c. AND,OR operators 
 - d. Like Operator 
 - e. usage of NULL
 - f. Subqueries
 - g. CASE statement



①

Offset

employee

eid	name	Sal	Dept
0	A	400	IT
1	B	200	IT
2	C	700	AD
3	D	100	HR
4	E	900	AD

↓

0	900
1	700
2	400
3	200
4	100

Qn. Get me second highest salary from employee?

An.

Select * from employee

order by salary desc

limit 1

offset 2

900
700

↓
700

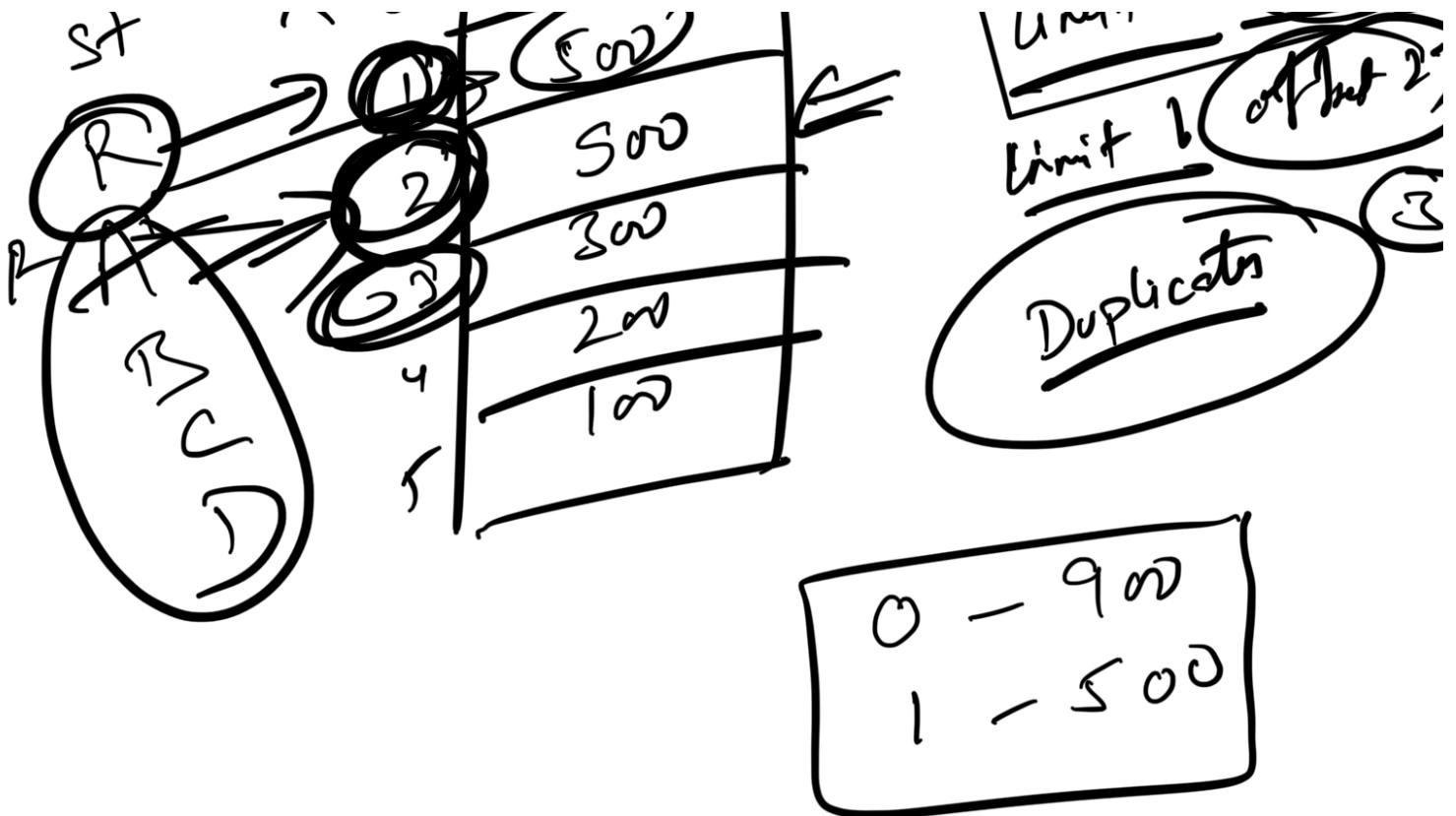
0 900
1 700
2 500
3 200
4 100

X₂

Limit 1 offset 1

X₀ 700

limit 1
1...+1 offset 1



Functions

Round(col, 1)

→ Ceil

→ Floor

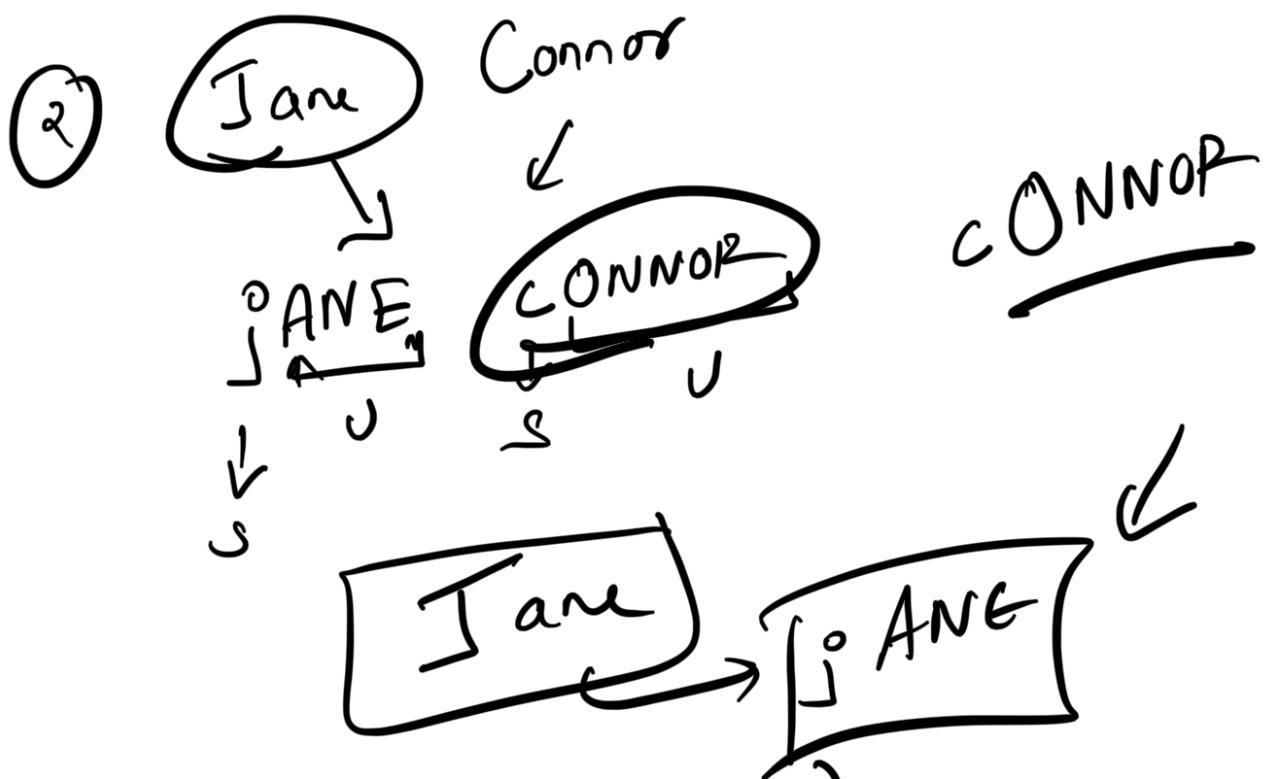
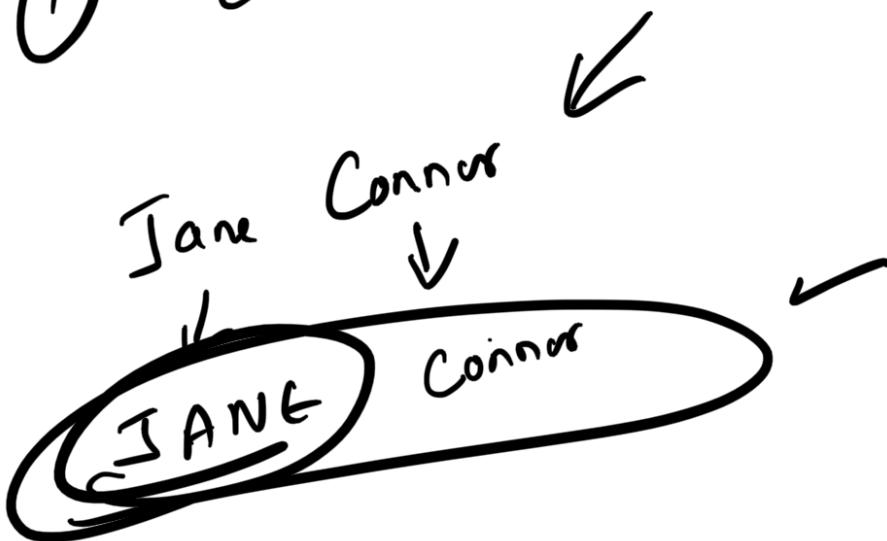
→ Least

→ greatest

Select least $\{7, 4, 2, 1\}$

$$\begin{aligned}
 &\downarrow \\
 &\text{greatest} \\
 &= ? 1 \\
 &= ?
 \end{aligned}$$

String
① Concat



→ Substring (lippet, lct, lct)
Concat
, r Jane i. i), = IX(̄)

lower (Substring L⁻, , , ")
 upper (Substring (Jane (2)))
)
 ↓
 = one
 ANE

OR

	0	1
0	F	F
1	T	F
	T	T
	T	T

AND

	0	1
0	T	F
1	F	T
	F	F
	F	F

IN (1, 2, 3)
 IN (, , , ,)

Pattern Matching
 like operator



U LO ~



(Jer) LO

Case Series

Jersey ✓
Jermey ✓
Jereemay ✓
Jeremy X

aksh LO ↗

0 - More likely

T ↘ ↘
① akshay

②

Mo-akshii

③ Akshay

④

Rudraksh i

one & only 1

- mit

① Amrit ✓

②

Cherit

④

legit X

③ Sharit X

An-

⑥ - tumn ↗

↖

✓

- Latumn

(a) ~~Autumn~~
 (b) ~~Rat~~
 (c) ~~Off~~
 (d) ~~Autumn~~

$$\begin{array}{r}
 1000 \\
 20^{\circ} \\
 \hline
 100^{\circ} + 2^{\circ} = 12^{\circ}
 \end{array}
 \Rightarrow \frac{2^{\circ}}{100^{\circ}} \times 1000 \Rightarrow 2^{\circ}$$

$$A \div \div \div \div = N$$

- (1) Ashmin X
- (2) Autumn X
- (3) Angoyan
- (4) Athiranam X

offset → has to be used only with unit
 → limit 0 offset 0
 → integer
 → pointer



limit 1 offset (1)
 (2)

1	1
1	2
2	3
3	4
4	5

Day - 3

filtering & Subqueries-II

- ① No Size
- ② null
- ③ Blank

④ → Spaces Blank (1, 2, 3)

⑤ → ASCII = 0

⑥ → Does take memory
and disk

⑦ → TRIM() + all & leading
trailing

⑧ 1 Anut = name
Select name from T
from Anut

- null
- No Ascii
- no memory / no
Disk
- Size is null ;
- if null (col, replace)

⑨ \leq Anut Singh \leq
Anut

~~① \Rightarrow Amit Singh @ Singh~~
~~② \Rightarrow Amit Singh~~
 e) name = ' ' ;
 ↑

① Select * from Customer 100
 where last_name IN [
 ↓
 a
 b
 c]
 ↗ 26 → Sub Queries
 [a, b, c, d, e, f, g]
 ... 26

Outer Query Sub Queries Inner Query
 ↓ ↗ ↓
 End [] First

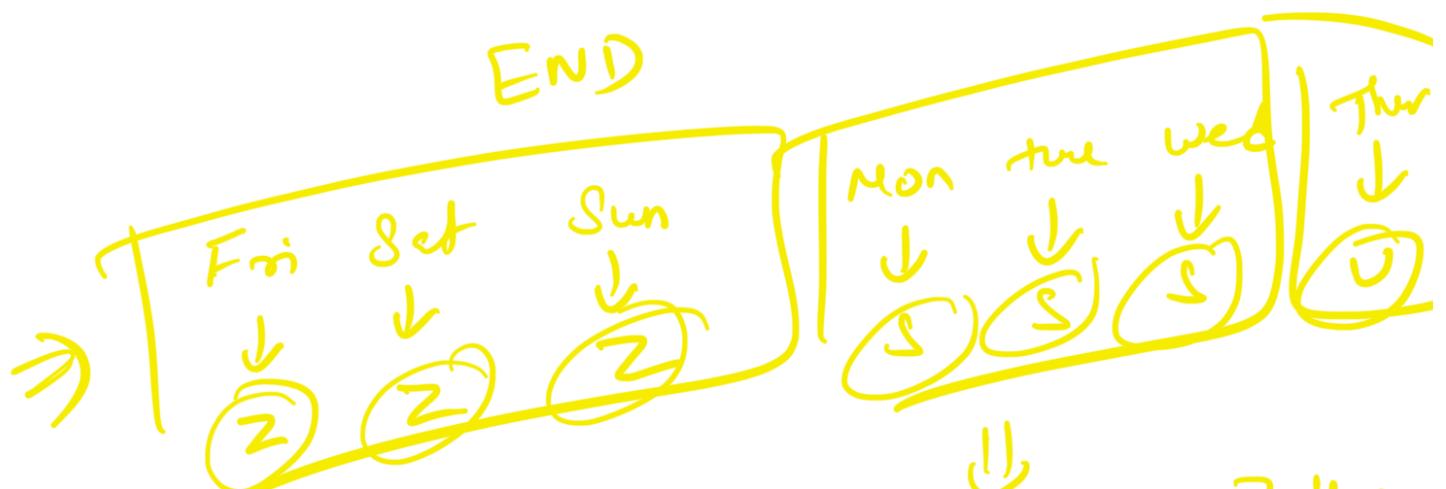
O C I Q , Q , Q , Q

TQ)
)) , TQ(1)

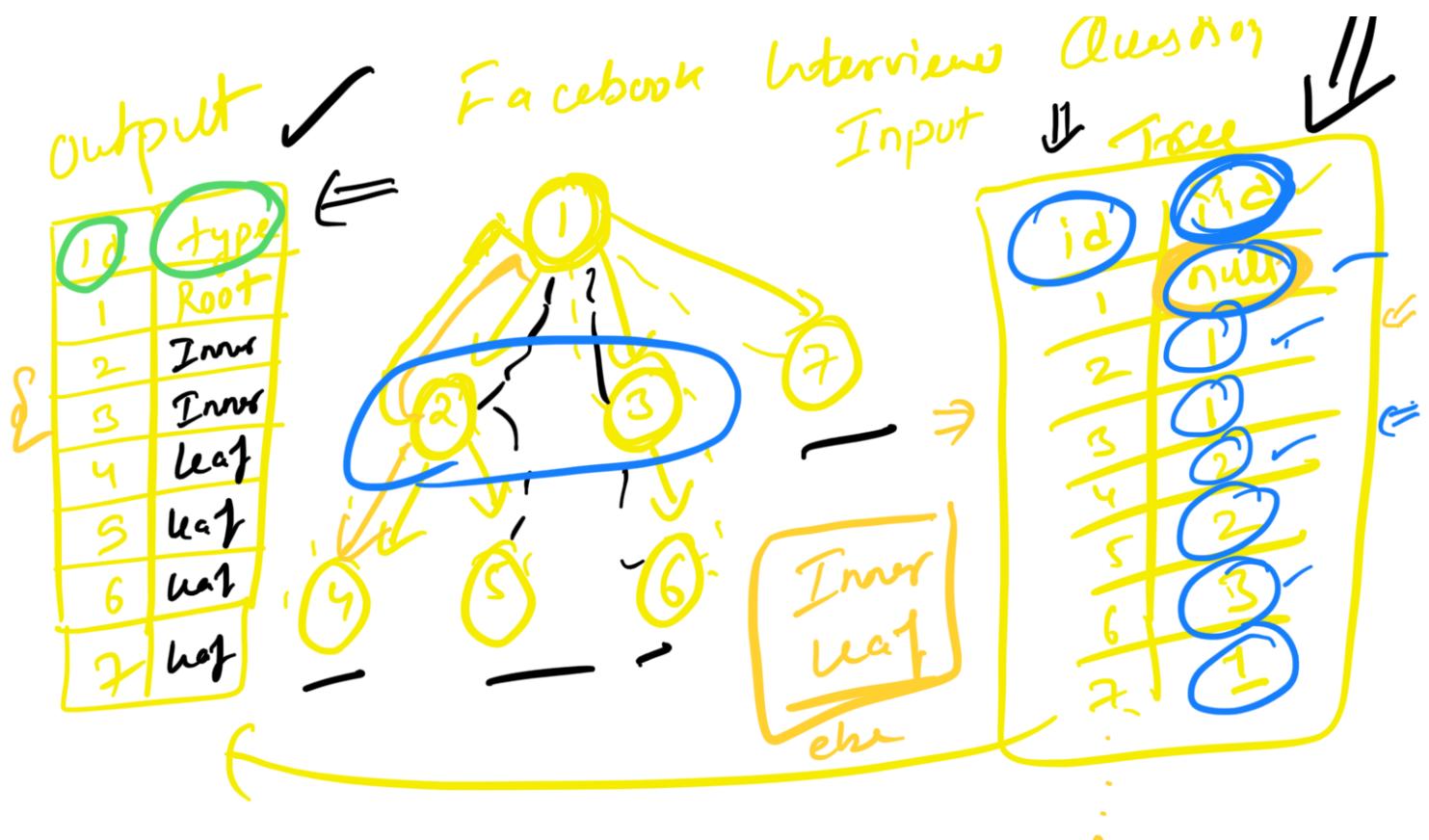
Choices
→ if else
for → CASE statements

Syntax

CASE when (condition is true) then
 do this;
 when (---) then do my
 thing
 else (do this)



Schult \Rightarrow Case
when days in $[F, S, J]$ then
zomato
when days in $[M, T, W]$ then
swiggy
else ubereats
End as (my-like-my-eaten)



Answer :

Select id

Case
when $\text{id} \in \text{IN}$ (Select id from tree
where p_id is null)
then 'Root'

End of type
from tree;

	id	p_id
→	1	null
→	2	1
→	3	1
→	4	2

Case
X X when $\text{p_id} = \text{null}$
then 'Root'
X when $\text{id} \in \text{IN}$ (Select
 p_id from tree)
else sum over

