

⇒

vendor_id	Name	Type	if_fresh	Action
	→	<u>Fresh</u> focus	1	←
		Prep food	0	←
		—	0	←
	→	Egg s <u>frsh</u>	1	

↑
CASE & WHEN

WHERE → Action? X

Syntax ?

SELECT ✓
 {
 _____,
 _____,

 → CASE
 WHEN
 THEN
 WHEN
 .

(condⁿ) ✓
 ? ✓

↓
ELSE Action
END AS alias

FROM _____
↓ Wt _____

"Total fresh items" → LIKE ✓
↑

WHEN ^{lower}(vendor_type) LIKE " % fresh % "
THEN "fresh"
ELSE "Not fresh"
↓
Binary Action

↑↑
FRESH
fresh
fresh
FRESH

SELECT

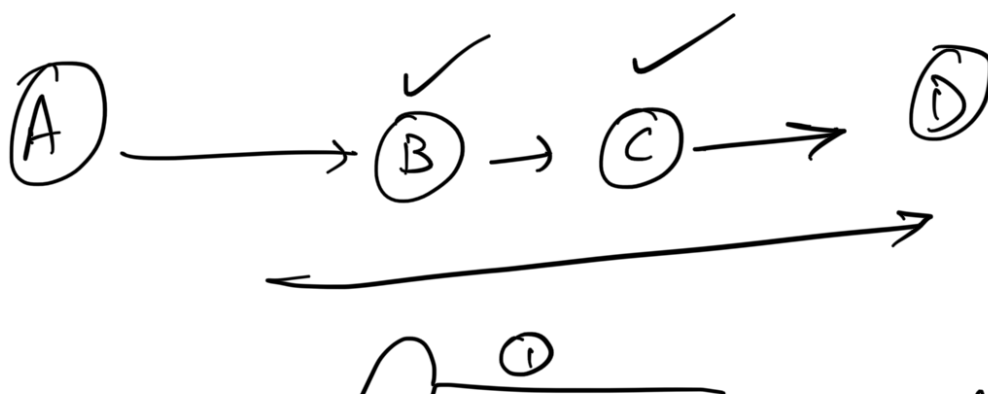
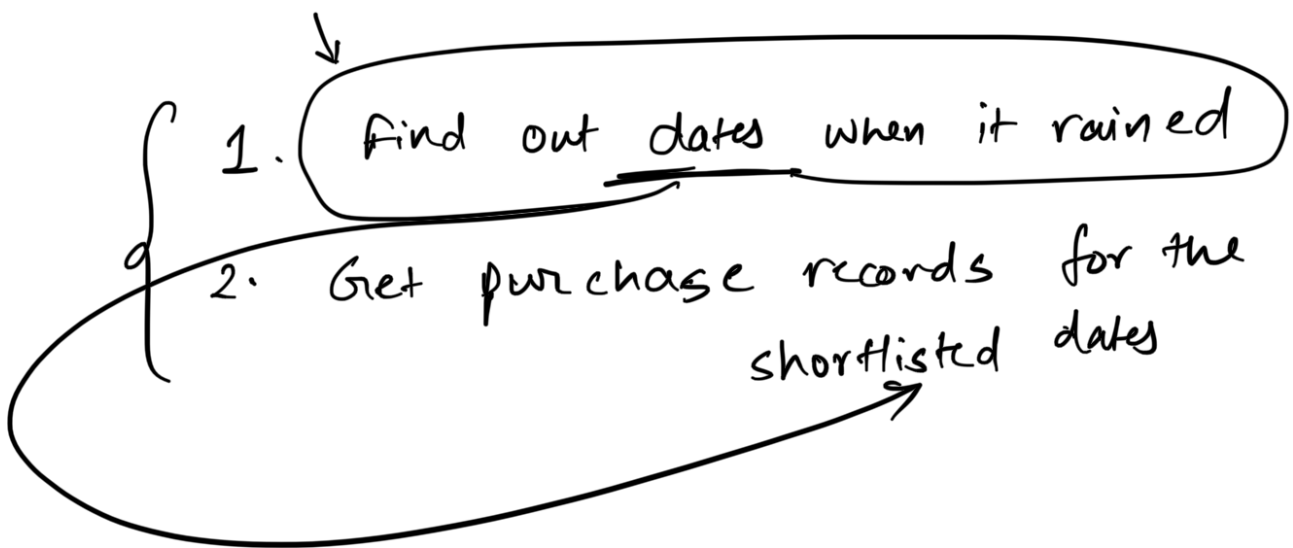
... / ... in ... if true val_if_true val_if_false)

IF (cond , val-if-true , val-if-false)
As —

IF \rightarrow Binary/Two possible Actions (1/0)

CASE \rightarrow Multiple (more than 2 possible actions)
WHEN

Q: Days when it rained



Q: Get details of all vendors selling fresh products along with product names and the qty of that product in their inventory.

③

v-id	v_name	product	pcat	qty	date

Product (L)

P-id	P-name	Pcat-id
1	A	1
2	B	2
3	C	2
4	D	1
5	E	3
6	F	10 ←

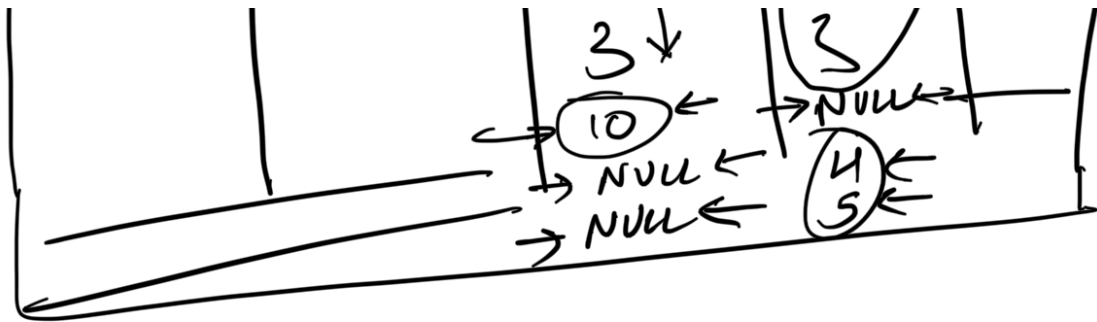
⑤

Product - cat (R)

Pcat-id	pcat_name
1	M
2	N
3	O
4 ←	P
5 ←	Q

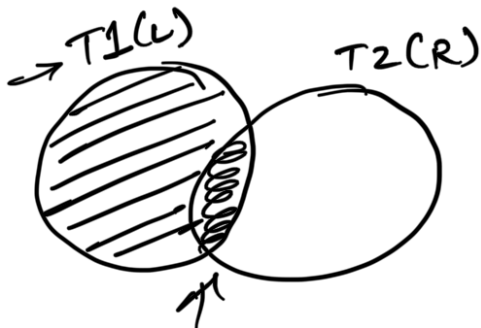
outer Join

P-id	P-name	P. pcat-id	PC.pcat-id	Cat-name
		1	1	
		2	2	
		2	2	
		1	1	

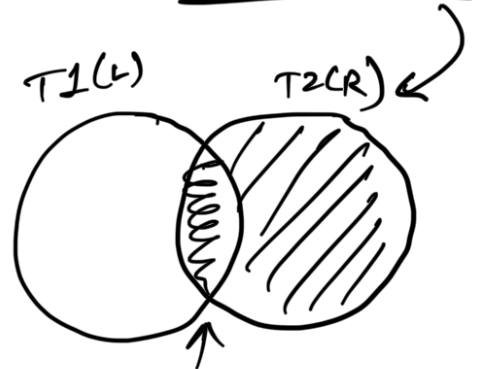


Joins

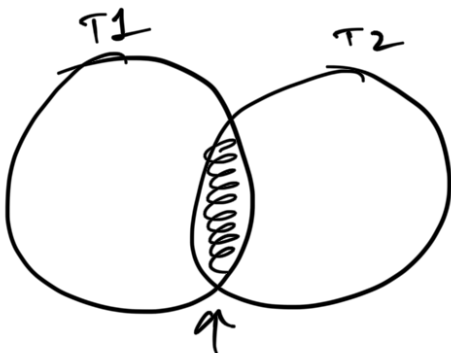
1. Left Join



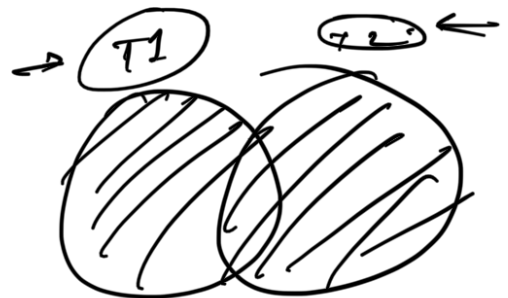
2. Right Join



3. Inner Join



④ Full/outer Join



Syntax:

SELECT

_____ ,

_____ ,

==>

FROM left_table ✓

→ [TYPE OF JOIN] right_table ✓

ON left_table. [field_to_match] = right_table. [field]

WHERE

✓ Customer (L) ✓

C-id		
1		
2		
3		
4		
5		

CP (R)

C-id		
1		
2		
2		
3		

✓ Left Join

L-C-id	R-C-id
1	1
1	1
2	2
2	2
3	3
4	NULL



→ WHERE Cp. customer_id IS NULL

"%@scler.com"

NULL → Absence of value

5 = NULL ←