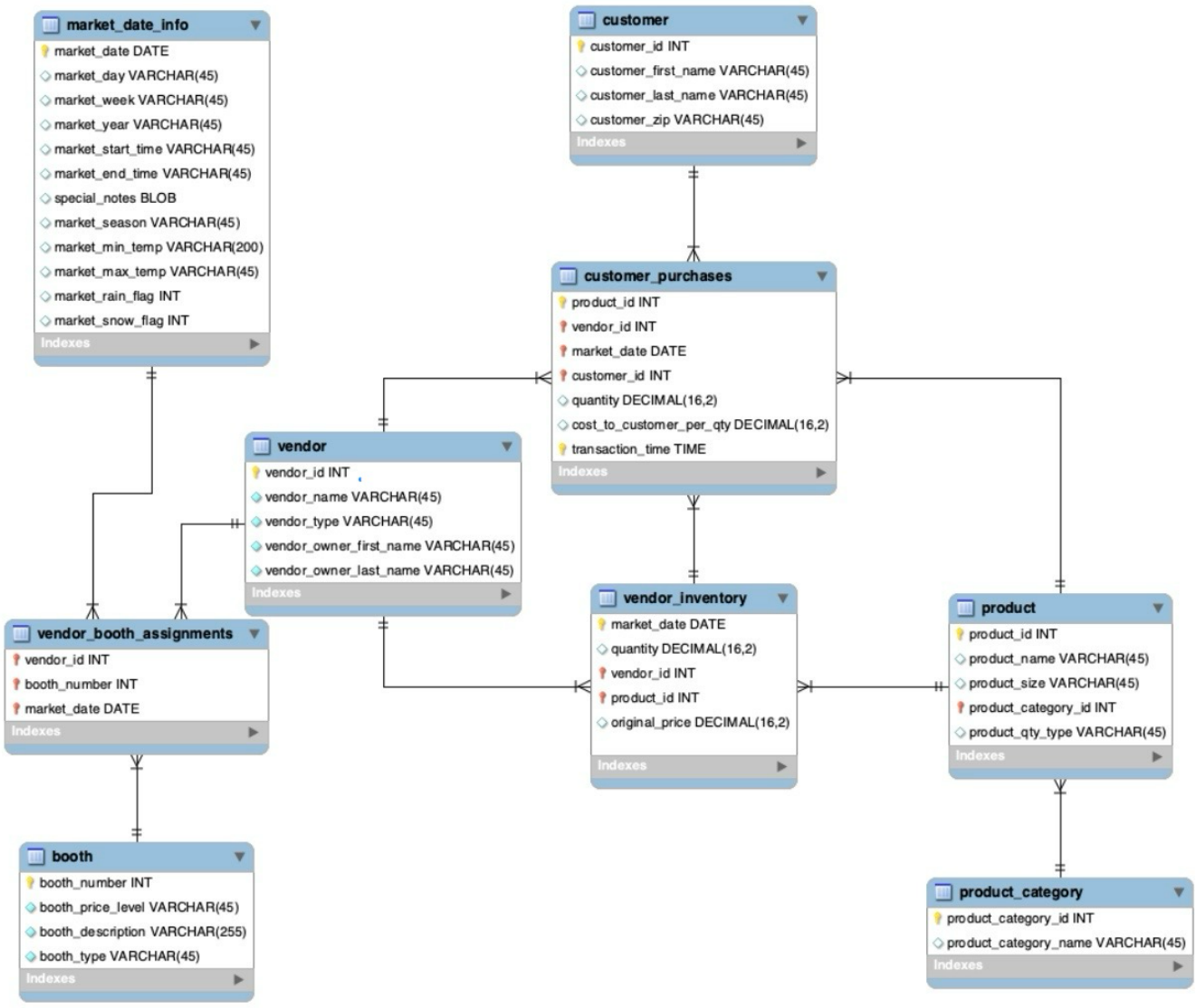
Agenda

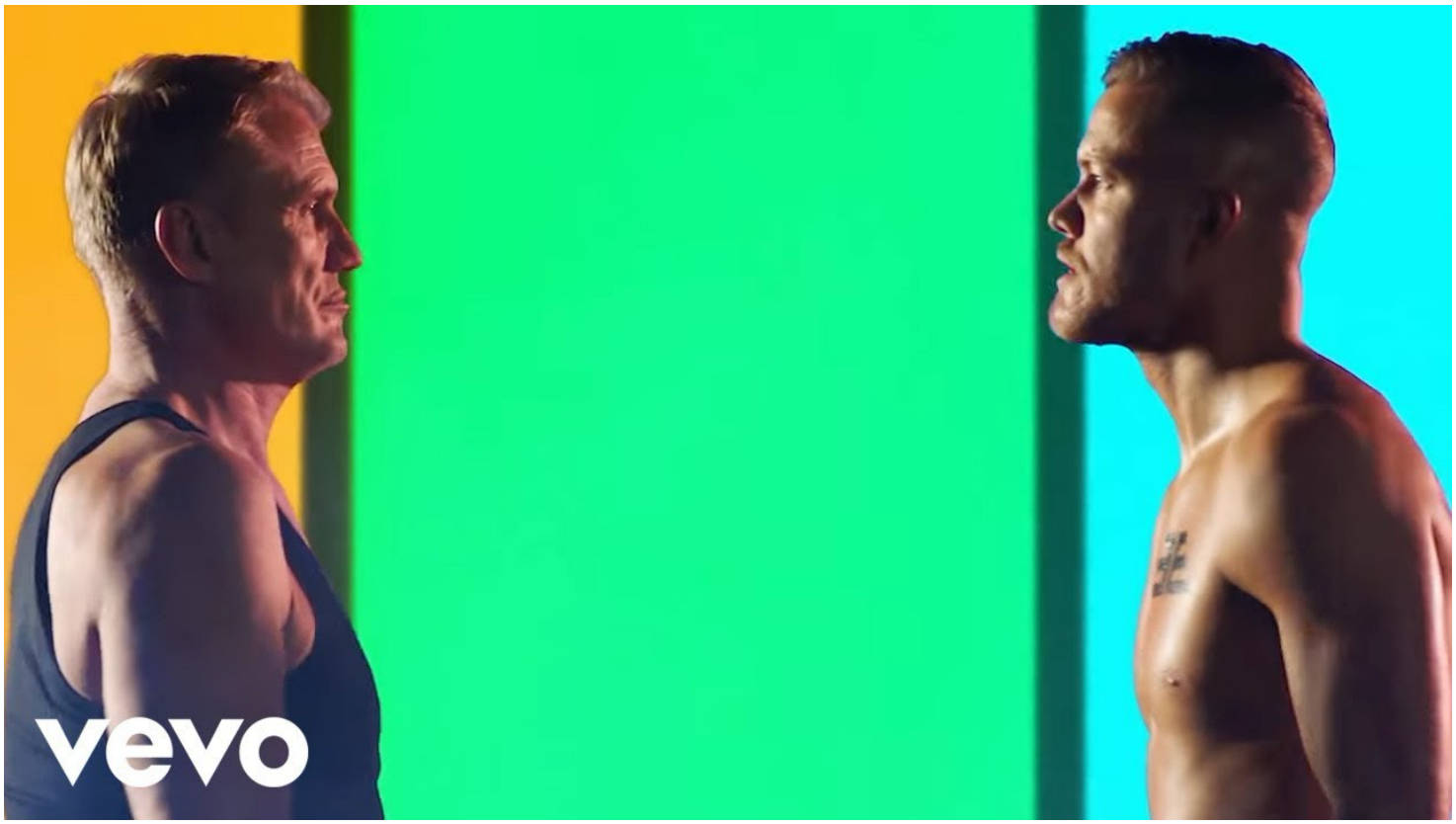
Full Join

Group by

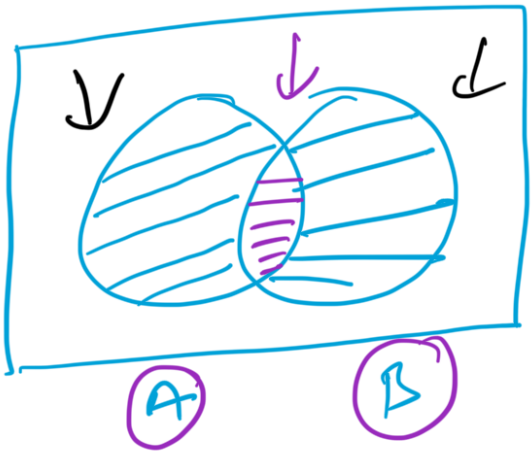
Having clause

Having vs Where Clause




vevo

Full join / Full outer join / outer join



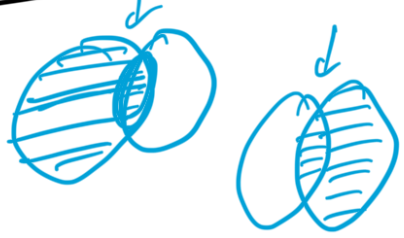
A = 1, 2, 3, 4, 5

B = 4, 5, 6, 7, 8

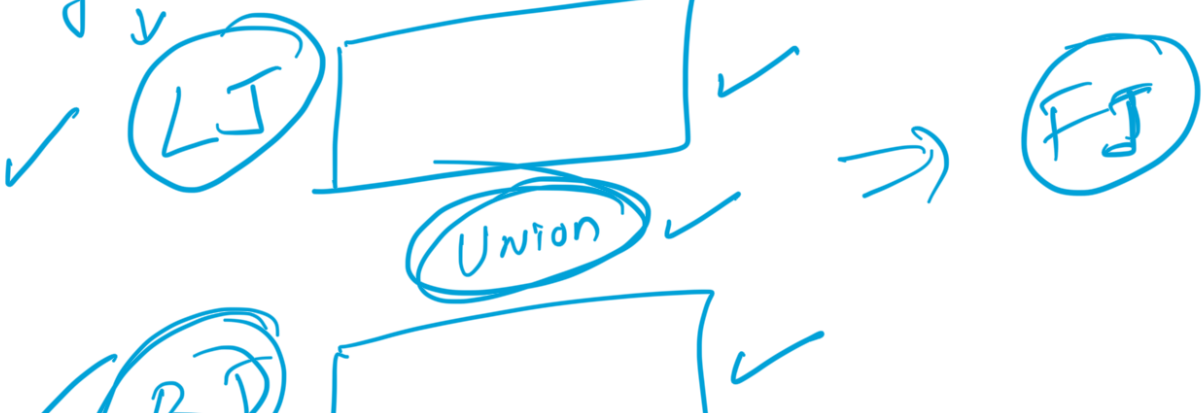
A ∪ B

Syntax :-

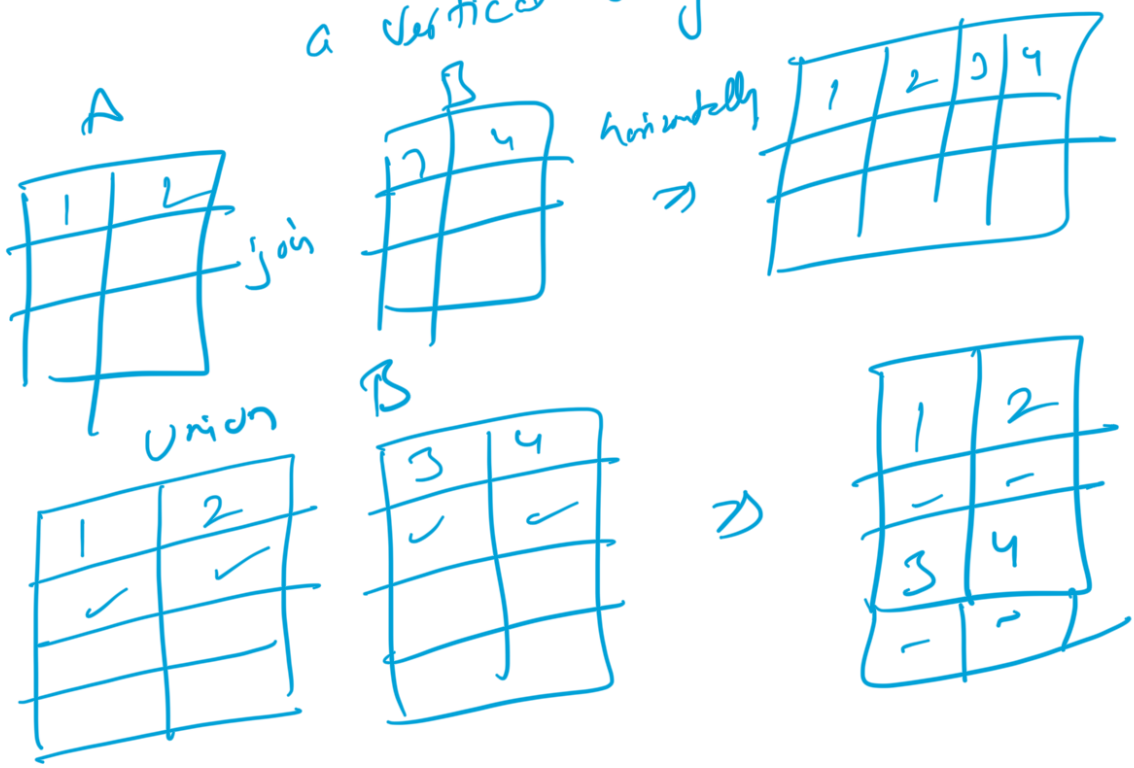
```
Select *
from A a
full join / full outer join B b
ON a.id = b.id ;
```



MySQL full join ⇒



Union : is used to Combine the result of two or more Select statement in a vertical way.



Union rules

→ (1) Every Select statement within UNION must have same no. of columns.

→ (2) The columns must have same/similar datatypes.

Union

A = 1, 2, 3, 4, 5

Union All

A = 1, 2, 3, 4, 5

B = 4, 5, 6, 7, 8

B = 4, 5, 6, 7, 8

A union B = {1, 2, 3, 4, 5, 6, 7, 8}

A union All B =

1 2 3 4, 5, 4, 5 6 7 8

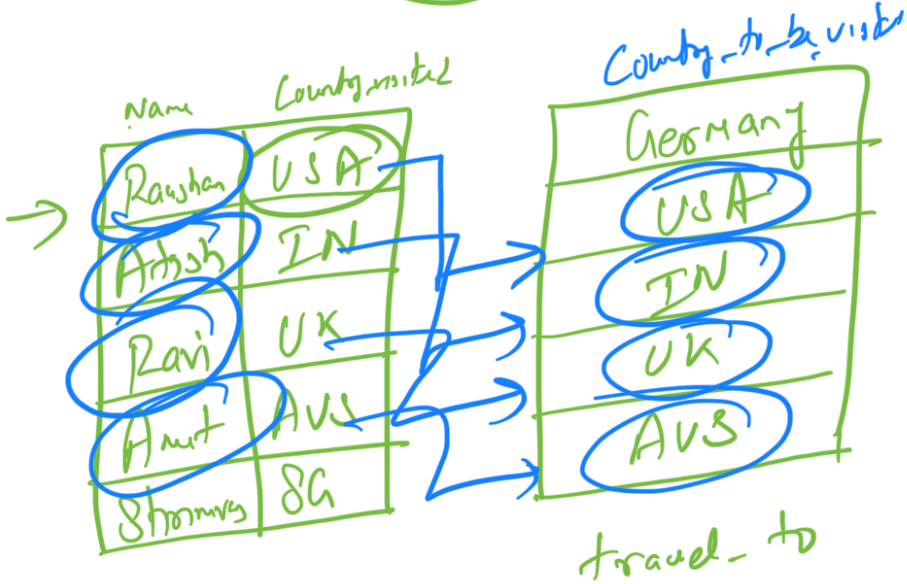
---

equi and non equi join

equi join ⇒ ON a.id = b.id;

Non equi join ⇒ ON a.id < > <> != <= >= b.id;

XYZ



Select *
from Users u
join travel_to t
ON u.cv = t.ct
;

!=

AUS

R ⇒ G, IN, UN,

Roushan	Germany
11	India

*

(2) Correlated Subquery :-

OQ

→ Select * from A where a.id IN

→ IQ Select id from B where condition

)

⇒ a Subquery that uses the values from outer query.

emp

eid	name	address
1	A	DL
2	B	PU
3	C	Chd
4	D	Mumbai
5	E	Hyd

Dept

D'id	Dname	Eid
D1	IT	1
D2	HR	2
D3	Testing	3

F

F

T

Qn. Find all employee detail who work in

a department.

An. (1)

```
Select * from employees e
Left join departments d
ON e.eid = d.eid
where d.dept_id is not null;
```

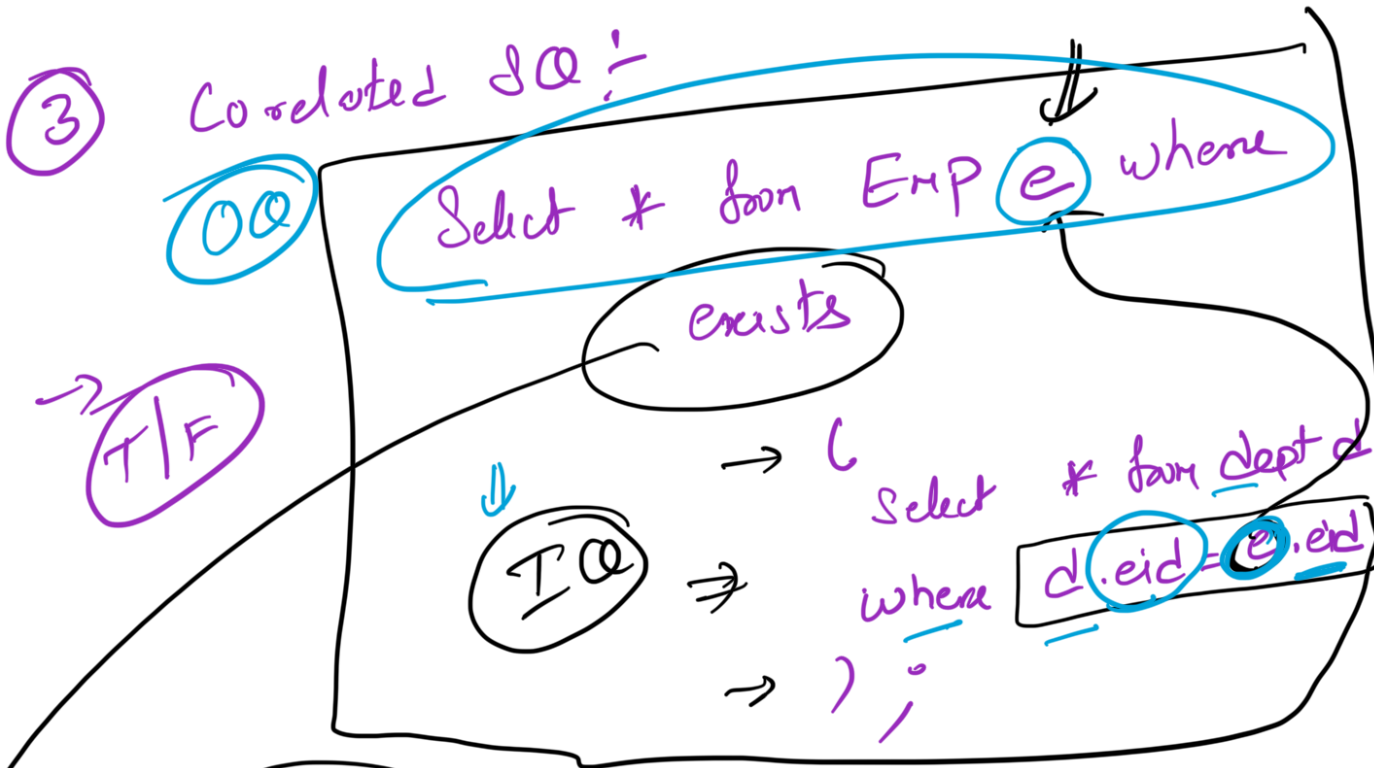
(2) Subquery :-

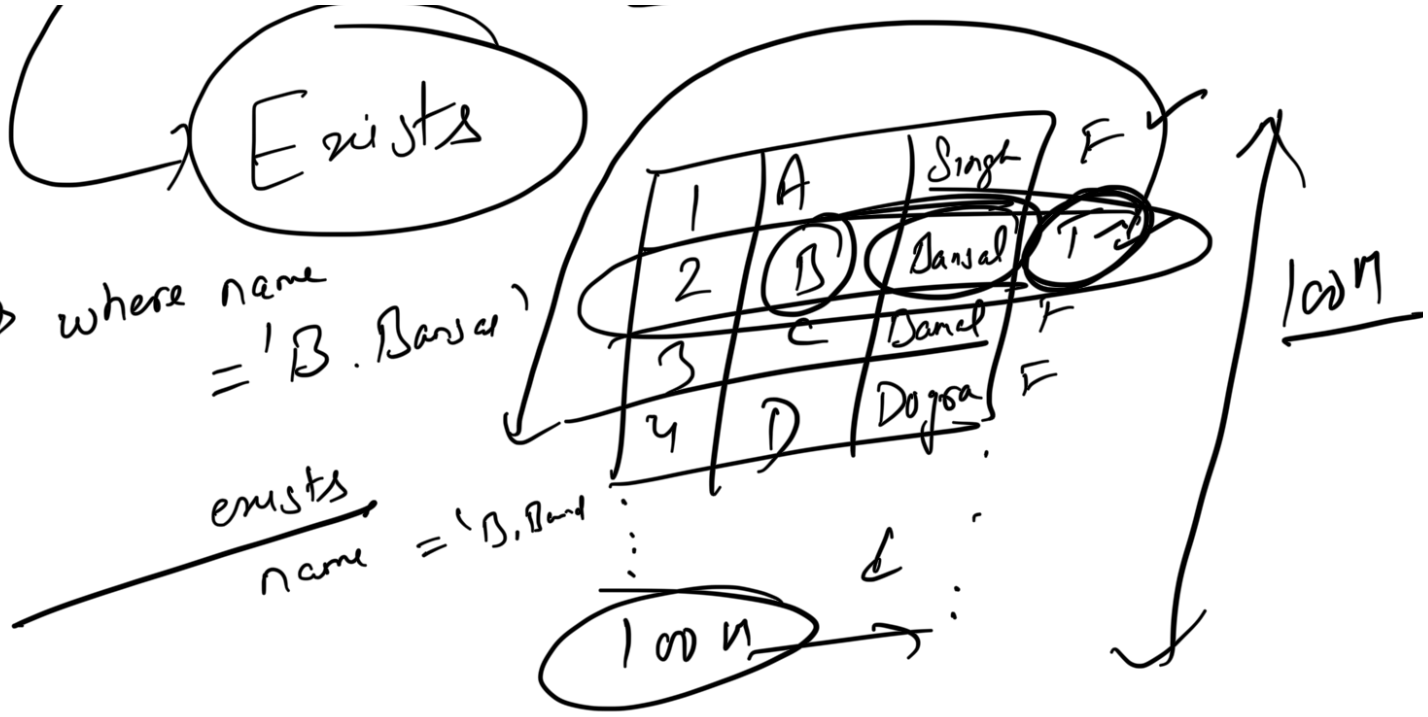
OQ

```
Select (*) from employees
where eid IN (1,2,3)
(
Select eid from departments
);
```

IQ

(3) Correlated SQ :-





Aggregation functions

① Min ② Max ③ Sum ④ Count ⑤ Avg

Name	Salary
A	10ω
B	3ω
C	4ω
D	6ω
E	2ω
F	7ω

→ Min (Salary)
→ 10ω

→ Max (Salary)
→ 7ω

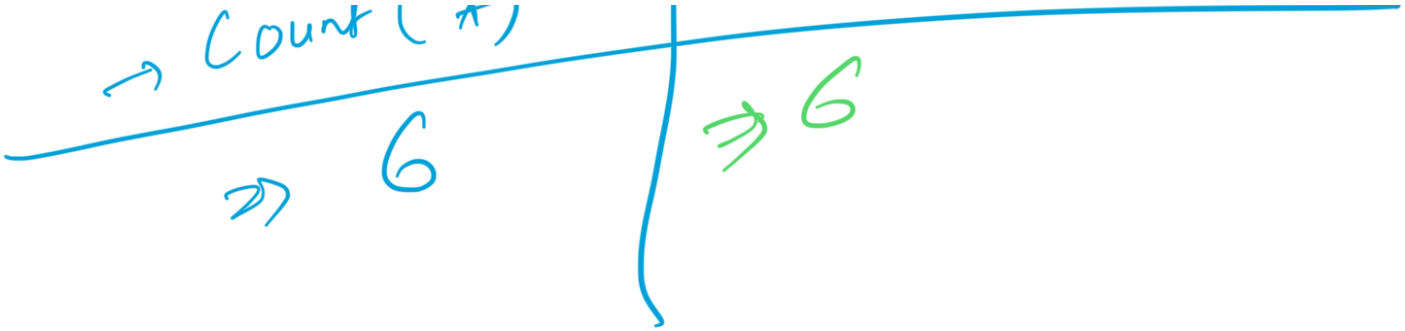
→ Sum (Sal) → 23ω

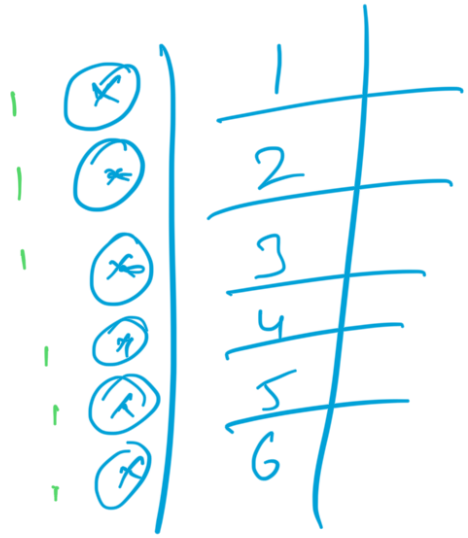
→ Count (*) = 6

→ Avg (Sa) = Σ / C = 23ω / 6 → (4ω)

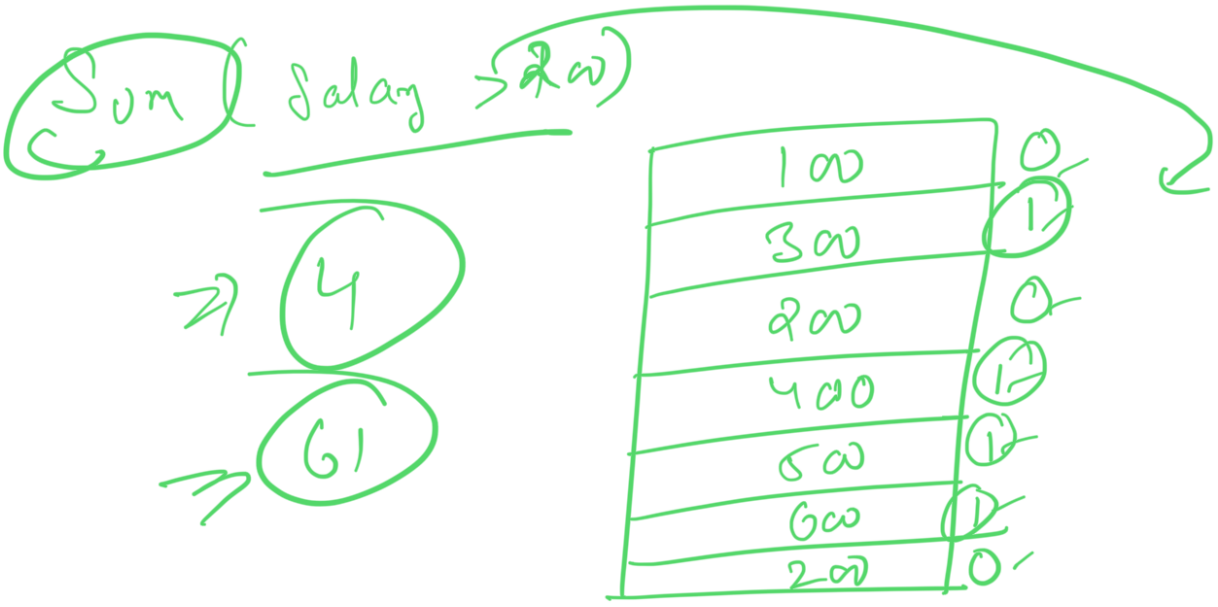
Count ( ) → arg

Count (*) | Count (1)

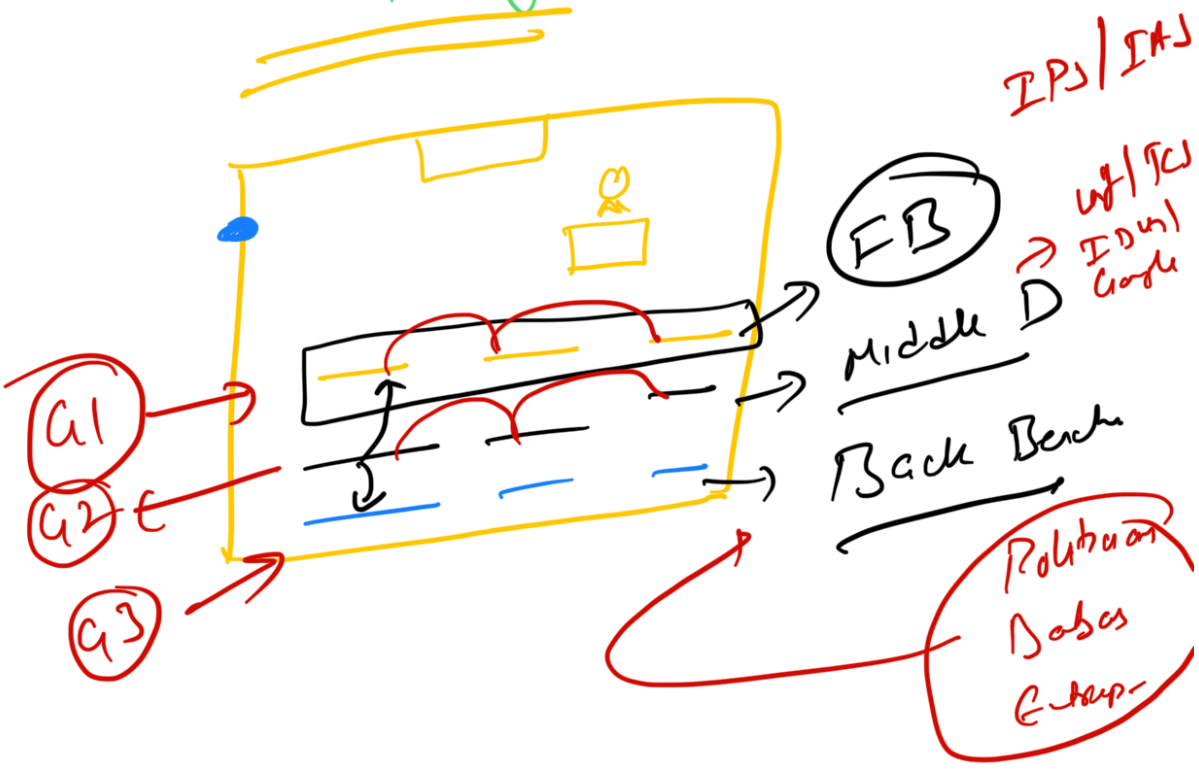




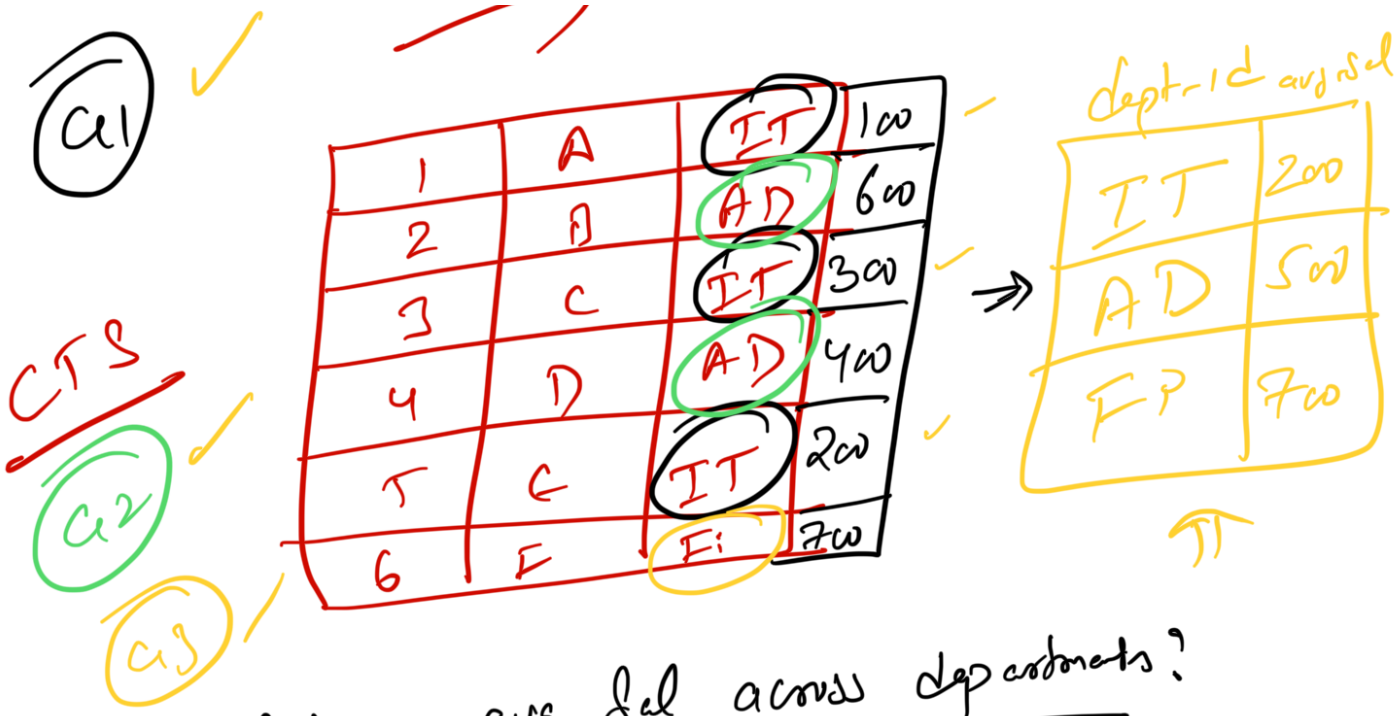
Count(*)	Count(1)	Count(column_name)	Count(DISTINCT col_name)
Counts NULL - commonly used - less confusing	Exactly same as count(*). Better to use count(*)	Counts duplicates & ignores NULL.	Neither duplicates nor NULLS



Group by



Syntax:- Select *
→ from A
→ where
→ Group by Col or Cols



⇒ Get me avg sal across departments?

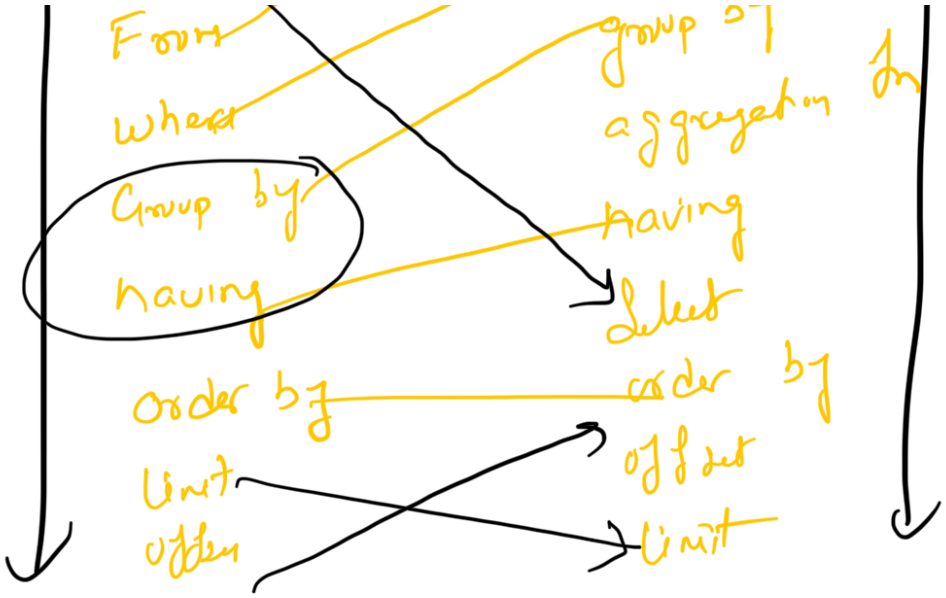
```
Select dept_id, avg(sal) as avg_sal
From employees
Group by dept_id;
```

(1) whatever u have in Group by, make sure they are present in Select statement.

*(2) if you are putting more uniqueness in Select statement, make sure that is also a part of Group by

---

Having clause

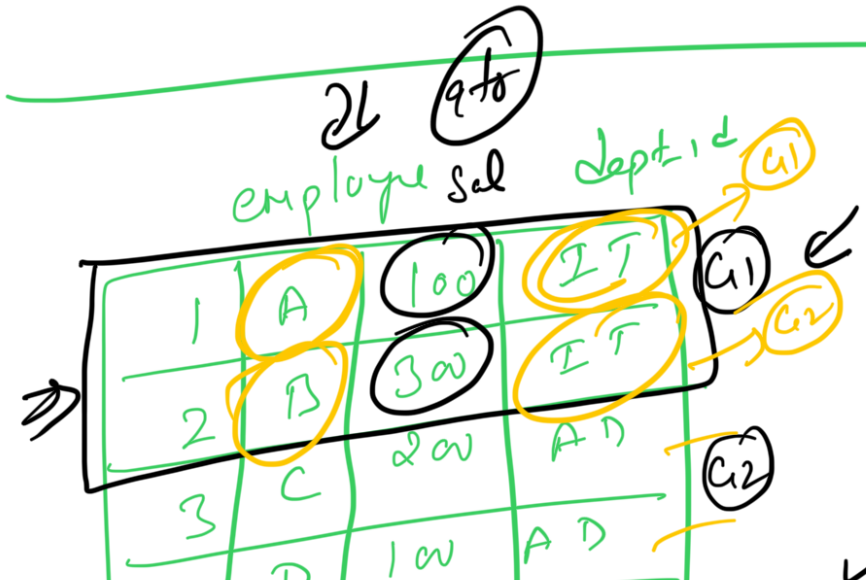


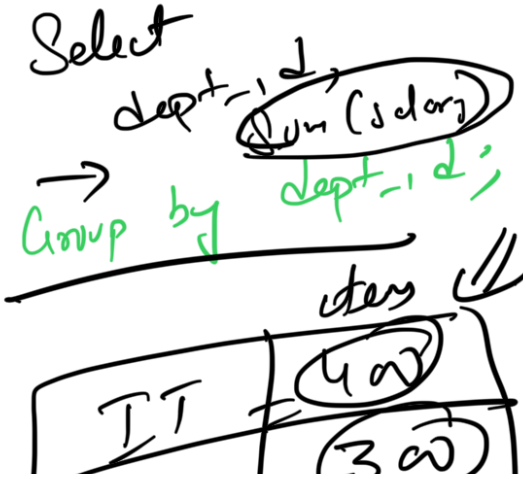
having = filter clause

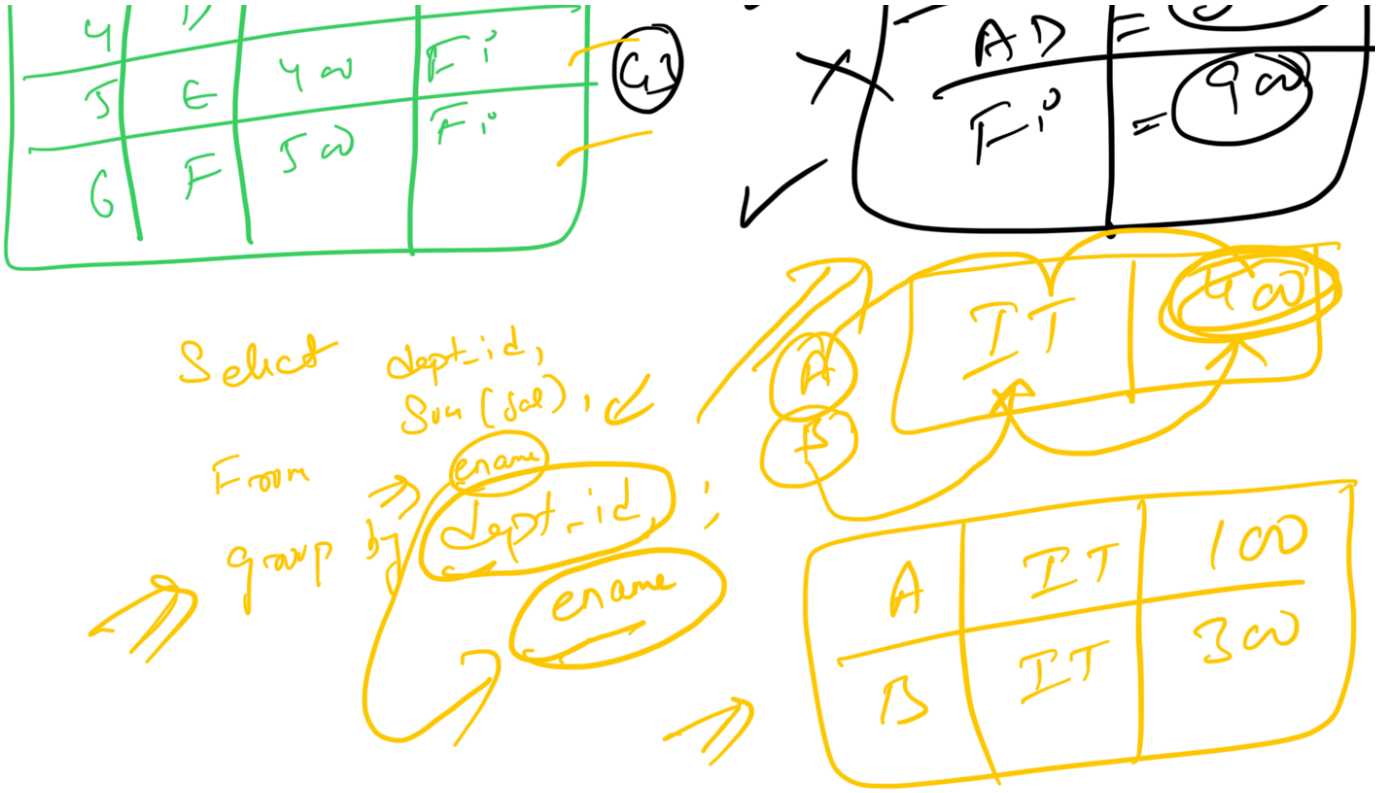
where = filter clause

where ⇒ always works with Columns which are in table.

having ⇒ always works with aggregation Columns, which are natively not present in table.






Select dept_id, Sum(sal),
From
group by
ename
dept_id
ename
A
B
IT
100
A
IT
100
B
IT
300