==========

MySQL **Group by** clause

==========

Group by :Used **for grouping**
Where clause->used **for** searching
Order by clause->used **for** sorting
For Update->used **for** locking **table row** manully

```
mysql> select * from emp;
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
|     1 | arun   | 8000 |      1 | M    | 4    |
|     2 | ali    | 7000 |      1 | C    | 1    |
|     3 | kiran  | 3000 |      1 | C    | 1    |
|     4 | jack   | 9000 |      2 | M    | NULL |
|     5 | Thomas | 8000 |      2 | C    | 4    |
+-------+--------+------+--------+------+------+

mysql> select sum(sal) from emp;
+----------+
| sum(sal) |
+----------+
|    35000 |
+----------+

mysql> select sum(sal) from emp where deptno=1;
+----------+
| sum(sal) |
+----------+
|    18000 |
+----------+
```

to see ouput **Sum of** sal departmentwise

```
mysql> select deptno,sum(sal) from emp group by deptno;
+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|      1 |    18000 |
|      2 |    17000 |
+--------+----------+
```

==========

Important **SQL** Rules:

steps:
1. 5 **rows** will loaded **in** Server ram **from** Datbase Hardisk**(only** needed **data are in** ram**)**
2. **(**sorting deptwise**)**it will goes two dimensional **ARRAY and** sorting will be done departmentwise
3. **Grouping is** done departmentwise
4. Seperate **array for each** department **and** summation done**(**total**)**
5. **HAVING** clause **if** present
6. **Order by** clause **after having**

==========

---

* Two rules **for group by** clause

```
Select clause->select deptno, sun(sal)
From clause->from emp
Group by clause->group by deptno
```

Rule 1:besides the **group** functions,whicheve colum **is** present **in select** clause has **to** be present **in Group by** clause

```
select deptno,sum(sal) from emp;----->error
mysql> select deptno,sum(sal) from emp;
+--------+----------+
```

```
| deptno | sum(sal) |
+--------+----------+
|      1 |    35000 |
+--------+----------+
```

this work is mysql but error in oracle.but the output is meaning less.

use group by clause

```
mysql> select deptno,sum(sal) from emp group by deptno;
+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|      1 |    18000 |
|      2 |    17000 |
+--------+----------+
```

Rule 2:which ever column is present in group by clause ,
it may or may not be present in select clause.

```
mysql> select sum(sal) from emp group by deptno;
+----------+
| sum(sal) |
+----------+
|    18000 |
|    17000 |
+----------+
```

it will not department for whichever

```
mysql> select deptno,min(sal) from emp group by deptno;
+--------+----------+
| deptno | min(sal) |
+--------+----------+
|      1 |     3000 |
|      2 |     8000 |
+--------+----------+
```

```
mysql> select deptno,max(sal) from emp group by deptno;
+--------+----------+
| deptno | max(sal) |
+--------+----------+
|      1 |     8000 |
|      2 |     9000 |
+--------+----------+
```

```
mysql> select deptno,count(*) from emp group by deptno;
+--------+----------+
| deptno | count(*) |
+--------+----------+
|      1 |        3 |
|      2 |        2 |
+--------+----------+
```

```
mysql> select count(*),min(sal),max(sal),sum(sal) from emp;
+----------+----------+----------+----------+
| count(*) | min(sal) | max(sal) | sum(sal) |
+----------+----------+----------+----------+
|        5 |     3000 |     9000 |    35000 |
+----------+----------+----------+----------+
```

```
mysql> select count(deptno),min(sal),max(sal),sum(sal) from emp;
+---------------+----------+----------+----------+
| count(deptno) | min(sal) | max(sal) | sum(sal) |
+---------------+----------+----------+----------+
|             5 |     3000 |     9000 |    35000 |
+---------------+----------+----------+----------+
```

---

* where clause is specified before the group by caluse
* where clause is used for searching
* searching takes place in DB server HD
* where clause is used to restrict the ROWS
* where clause is used to retrive the rows from datbase server
Hard disk to server RAM

```
mysql> select deptno,sum(sal) from emp where sal>7000 group by deptno;
+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|      1 |     8000 |
|      2 |    17000 |
+--------+----------+
```

1. error if. we write group first then. where clause.

Department and jobwise group by clause

```
mysql> select deptno,job,sum(sal) from emp group by deptno,job;
+--------+------+----------+
| deptno | job  | sum(sal) |
+--------+------+----------+
|      1 | M    |     8000 |
|      1 | C    |    10000 |
|      2 | M    |     9000 |
|      2 | C    |     8000 |
+--------+------+----------+
```

i.e. work like nested for loop.
this will loaded is server ram select depto,job,sum(sal) Deptno job Sum(sal)

* no upper limit on the number of column in group by clause

```
SQL> select .......
     group by country,state,city;
     if we have 3 colums in SQL cmd then.
```

* 3levels nested loop.
  Country....inside...State..inside...city...inside
  i.e. select statement will be slow(that much sorting has to take place)

*if interchange the order of colums THEN.

```
select job,deptno, sum(sal) from emp group by job,deptno;
mysql> select job,deptno, sum(sal) from emp group by job,deptno;
+------+--------+----------+
| job  | deptno | sum(sal) |
+------+--------+----------+
| M    |      1 |     8000 |
| C    |      1 |    10000 |
| M    |      2 |     9000 |
| C    |      2 |     8000 |
+------+--------+----------+
```

this is jobwise departmentwise sorting sum.
in this job is outter loop and deptno is inner LOOP but output is same
moral...order doesn't amtter'.... .difference makes in processing speed.
40 times faster...(see video later 3.19 pm)

```
mysql> select deptno,job,sum(sal) from emp group by job,deptno;
+--------+------+----------+
| deptno | job  | sum(sal) |
+--------+------+----------+
```

```
|      1 | M    |     8000 |
|      1 | C    |    10000 |
|      2 | M    |     9000 |
|      2 | C    |     8000 |
+--------+------+----------+
```

* the position of select clause and order of group by need not be the same.
* the positon of select determines will be in output(user requirement)
* the order of columns in group by clause will determin the sorting order,
  the grouping order, the summation order and hence the speed of processing.

```
select.............
group by city, country, district, state;---<slower


select.............
group by country, state,district, city; <---faster
epends no of rows


mysql> select deptno,sum(sal) from emp group by deptno;
+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|      1 |    18000 |
|      2 |    17000 |
+--------+----------+

mysql> select deptno,sum(sal) from emp group by deptno having sum(sal)>17000;
+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|      1 |    18000 |
+--------+----------+


------------------------------------------------------
* HAVING clause works AFTER the summation is done.
* order by works after having
------------------------------------------------------

select.....from......
where.....
group by.....
having.....
order by.....

mysql> select sum(sal) from emp
    -> group by deptno;
+----------+
| sum(sal) |
+----------+
|    18000 |
|    17000 |
+----------+


select max(sum(sal)) from emp group by deptno;
Nesting of group function is allowed only in
oracle RDBMS, not in mySQL


select sum(sal) from emp group by deptno;

mysql> select sum(sal) from emp group by deptno;
+----------+
| sum(sal) |
+----------+
|    18000 |
|    17000 |
```

```
+----------+
```

not showing deptno using alice

```
mysql> (select sum(sal) from emp group by deptno) as tempp;
+----------+
| sum(sal) |
+----------+
|    18000 |
|    17000 |
+----------+
```

Nesting select stament ..

```
mysql> select max(sum_sal) from(select sum(sal) sum_sal from emp group by deptno) as tempp;
+--------------+
| max(sum_sal) |
+--------------+
|        18000 |
+--------------+
```

it will takes double time as compare to oracle.

==========

MySQL JOIN Operations

==========

```
CREATE table dept(deptno int(4),dname varchar(20),loc VARCHAR(20));
insert into dept values(1,'TRN','Bby');
insert into dept values(2,'EXP','Dlh');
insert into dept values(3,'MKTG','Cal');
```

```
mysql> select * from dept;
+--------+-------+------+
| deptno | dname | loc  |
+--------+-------+------+
|      1 | TRN   | Bby  |
|      2 | EXP   | Dlh  |
|      3 | MKTG  | Cal  |
+--------+-------+------+
```

```
mysql> select * from emp;
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
|     1 | arun   | 8000 |      1 | M    | 4    |
|     2 | ali    | 7000 |      1 | C    | 1    |
|     3 | kiran  | 3000 |      1 | C    | 1    |
|     4 | jack   | 9000 |      2 | M    | NULL |
|     5 | Thomas | 8000 |      2 | C    | 4    |
+-------+--------+------+--------+------+------+
```

* to void data redundancy->unnecessary duplication of data(wastage of Hard disk space)
  we use join operation.want to see columns of two three table.data is stored in multiple TABLE
* to view the columns of 2 or more table,then you will have to write a join
  write a join.

Normalisation:1NF, 2NF, 3NF BCNF

```
mysql> select ename,deptno from emp;
+--------+--------+
| ename  | deptno |
+--------+--------+
| arun   |      1 |
| ali    |      1 |
| kiran  |      1 |
| jack   |      2 |
| Thomas |      2 |
+--------+--------+
```

* select dname,ename from emp,dept where dept.deptno=emp.deptno;
* dept.deptno->tablename.columnname

last table dept->driving TABLE
emp-> driven TABLE <----------order

```
mysql> select dname,ename from emp,dept where dept.deptno=emp.deptno;
+-------+--------+
| dname | ename  |
+-------+--------+
| TRN   | arun   |
| TRN   | ali    |
| TRN   | kiran  |
| EXP   | jack   |
| EXP   | Thomas |
+-------+--------+
```

is entire table is search known full table scan.

where dept.deptno=emp.deptno;
where emp.deptno=where dept.deptno
order doesnt matter. output will be same.

from clasue dept,emp...speed excetion matter
dept,emp<-slower
emp,dept<-faster-dept outer table--emp inner table
emp driving table----outerloop--department will inner low--

department as driving TABLE
Deptno no 1 Arun ali kiran
Deptno no 2 jack thomas
Deptno no 3 where condition is not statifiy hence it rejected
Input DB server HD and server RAM---IO RAM 5 Times

faster search operation:
Outer loop must be LESS Inner loop must more

* searching is always on HDD

* in order for your join to work faster, preferably the driving TABLE should be table with lesser number of rows.

Order Of output changing:

```
mysql> select ename,dname from emp,dept where dept.deptno=emp.deptno;
+--------+-------+
| ename  | dname |
+--------+-------+
| arun   | TRN   |
| ali    | TRN   |
| kiran  | TRN   |
| jack   | EXP   |
| Thomas | EXP   |
+--------+-------+
```

select ename,dname from emp,dept where dept.deptno=emp.deptno order by 1;

```
mysql> select ename,dname from emp,dept where dept.deptno=emp.deptno order by 1;
+--------+-------+
| ename  | dname |
+--------+-------+
| ali    | TRN   |
| arun   | TRN   |
| jack   | EXP   |
```

```
| kiran   | TRN    |
| Thomas | EXP    |
+--------+-------+
ename sort aplhabetically


mysql> select ename,loc,dname,job,sal from emp,dept where dept.deptno=emp.deptno order by 1;
+--------+------+-------+------+------+
| ename  | loc  | dname | job  | sal  |
+--------+------+-------+------+------+
| ali    | Bby  | TRN   | C    | 7000 |
| arun   | Bby  | TRN   | M    | 8000 |
| jack   | Dlh  | EXP   | M    | 9000 |
| kiran  | Bby  | TRN   | C    | 3000 |
| Thomas | Dlh  | EXP   | C    | 8000 |
+--------+------+-------+------+------+


mysql> select * from emp,dept where dept.deptno=emp.deptno order by 1;
+-------+--------+------+--------+------+------+--------+-------+------+
| empno | ename  | sal  | deptno | job  | mgr  | deptno | dname | loc  |
+-------+--------+------+--------+------+------+--------+-------+------+
|     1 | arun   | 8000 |      1 | M    | 4    |      1 | TRN   | Bby  |
|     2 | ali    | 7000 |      1 | C    | 1    |      1 | TRN   | Bby  |
|     3 | kiran  | 3000 |      1 | C    | 1    |      1 | TRN   | Bby  |
|     4 | jack   | 9000 |      2 | M    | NULL |      2 | EXP   | Dlh  |
|     5 | Thomas | 8000 |      2 | C    | 4    |      2 | EXP   | Dlh  |
+-------+--------+------+--------+------+------+--------+-------+------+
common COLUMNS may have different meaning

* the common colum that is present in both the tables,that
  coluns name need not be the same in both the tables. because the same colum may have a
  different meaning in some other table

select dname,loc,ename,job,sal from emp,dept
where dept.x=emp.y order by 1;
-------------------------------------------------------------------------
mysql> select emp.deptno,dept.dname,dept.loc,emp.empno,emp.ename,emp.job,emp.sal from
emp,dept where emp.deptno=dept.deptno order by 1;
+--------+-------+------+-------+--------+------+------+
| deptno | dname | loc  | empno | ename  | job  | sal  |
+--------+-------+------+-------+--------+------+------+
|      1 | TRN   | Bby  |     1 | arun   | M    | 8000 |
|      1 | TRN   | Bby  |     2 | ali    | C    | 7000 |
|      1 | TRN   | Bby  |     3 | kiran  | C    | 3000 |
|      2 | EXP   | Dlh  |     4 | jack   | M    | 9000 |
|      2 | EXP   | Dlh  |     5 | Thomas | C    | 8000 |
+--------+-------+------+-------+--------+------+------+


mysql> select dname,sum(sal) from emp,dept where dept.deptno=emp.deptno group by dname;
+-------+----------+
| dname | sum(sal) |
+-------+----------+
| TRN   |    18000 |
| EXP   |    17000 |
+-------+----------+


mysql> select dname,sum(sal) from emp,dept where dept.deptno=emp.deptno group by dname;
+-------+----------+
| dname | sum(sal) |
+-------+----------+
| TRN   |    18000 |
| EXP   |    17000 |
+-------+----------+


mysql> select upper(dname),sum(sal) from emp,dept where dept.deptno=emp.deptno group by
upper(dname) having sum(sal)>10000 order by 1;
+--------------+----------+
```

```
| upper(dname) | sum(sal) |
+--------------+----------+
| EXP          |    17000 |
| TRN          |    18000 |
+--------------+----------+
```

==========

JOIN Types

==========

```
Select dname,ename from emp,dept where dept.deptno=emp.deptno;

mysql> Select dname,ename from emp,dept where dept.deptno=emp.deptno;
+-------+--------+
| dname | ename  |
+-------+--------+
| TRN   | arun   |
| TRN   | ali    |
| TRN   | kiran  |
| EXP   | jack   |
| EXP   | Thomas |
+-------+--------+

Cycle1: deptno(driving table less colums) 1 it will goto emp deptno 1 entrie table deptnp
1,1,1, TRN arun ali kiran
Cycle2: deptno 2 it will goto emp dept 2,2 Exp jack thomas
Cycle3: deptno 2 deptno=3 false stop loop.show output

mysql> Select dname,ename from emp,dept where dept.deptno>emp.deptno;
mysql> Select dname,ename from emp,dept where dept.deptno!=emp.deptno;
```

==========

Equijoin(Natural join)

==========

```
* join based on equality conditon(===)
* shows matching rows of both the tables
* use :
  a. view the colums of both the tables
  e,g, Dname and ename custname and orders deatils etc
* this is the most frequenlty used join(>90%) and hence it also known as natural join.
   92% in my computer....most frequenlty used more 90%
```

==========

Inequijoin(Non-equijoin)

==========

```
join based on inequality condition
Select dname,ename from emp,dept where dept.deptno!=emp.deptno;

Cycle1: deptno(driving table less colums)!=1 it will goto emp deptno!=1 entrie table deptnp
1,1,1, TRN arun ali kiran
Cycle2: deptno 2 it will goto emp dept!=2,2 Exp jack thomas
Cycle3: deptno 2 deptno!=3 all will come true stop loop.show

who are emp not belong to training
who are emp not belong to marketing
who are emp not belong to Export
* shows non matching rows of the both rows
Use:  a. not belong to particular thing(absent,not delivery,not made paymeent)
      b. used in  EXCEPTION reports(rarely used)

mysql> Select dname,ename from emp,dept where dept.deptno!=emp.deptno;
+-------+--------+
| dname | ename  |
+-------+--------+
| MKTG  | arun   |
| EXP   | arun   |
| MKTG  | ali    |
```

```
| EXP    | ali     |
| MKTG   | kiran   |
| EXP    | kiran   |
| MKTG   | jack    |
| TRN    | jack    |
| MKTG   | Thomas  |
| TRN    | Thomas  |
+-------+--------+
```

```
================================================================================
   Outer Join joined with plus sign in oracle(equijoin also)
================================================================================
Oracle:
select dname,ename from emp,dept
where dept.deptno=emp.deptno(+);

where condition

 LHS                          RHS
dept.deptno                emp.deptno(+)
cycle 1 deptno=1......1,1,1, goto emp return dname trn ename arun ali kiran(old thing)
cycle 2 deptno=1......1,1,1, goto emp return dname exports ename arun jack,thomas(old thing)
cycle 3 deptno=3.......false  even if false condition it showns me null value...matching
rows and also right side

* you canot put plus both side
* shows matching rows of both the tables plus non-matching rows of OUTER tab;each
OUTER table->table which is on opposite side of(+) SIGN known outter join
* a.master Deatils child(Parent-child reports)

//Child table--details tables
//Parent table-->master table

* practical of do while loop is outer join


================================================================================
Outer Join joined with plus sign in oracle(equijoin also)
================================================================================
select dname,ename from emp,dept
where dept.deptno (+)=emp.deptno;
dname  ename

1. Half Outerjoin((+)sign is on any side i.e. LHS and RHS)
     Left==left side(+)
     right==right side(+)
2. Full Outerjoin((+) both side sign)

* if we put Plus sign both side then nested do while loop...becomes full outer join

select dname,ename from emp,dept
where dept.deptno=emp.deptno (+);<------Right outer join

select dname,ename from emp,dept
where dept.deptno (+)=emp.deptno; <------left outer join



================================================================================
  Full outerjoin
================================================================================
*  shows matching rows of both the tables
   plus
   non matching rows of both the tables
*  based on nested Do-while loop


select dname,ename from emp,dept where dept.deptno=emp.deptno (+)
union                                                    <------Full outer join
select dname,ename from emp,dept where dept.deptno=emp.deptno (+);
```

```
-------------------------------------------------------------------------------------------------------
---------------------
* ANSI syntax for full outerjoin:(not working in MySQL)
* works all RDBMS not works only MySQL
  select dname,ename from emp full outer join dept on (dept.deptno=emp.deptno);
-------------------------------------------------------------------------------------------------------
--------------------
* ANSI syntax for Right outerjoin:
* works all RDBMS including MySQL
  select dname,ename from emp Right outer join dept on (dept.deptno=emp.deptno);
-------------------------------------------------------------------------------------------------------
--------------------
* ANSI syntax for Left outerjoin:
* works all RDBMS including MySQL
  select dname,ename from emp left outer join dept on (dept.deptno=emp.deptno);

-------------------------------------------------------------------------------------------------------
--------------------
mysql> select * from emp;
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
|     1 | arun   | 8000 |      1 | M    | 4    |
|     2 | ali    | 7000 |      1 | C    | 1    |
|     3 | kiran  | 3000 |      1 | C    | 1    |
|     4 | jack   | 9000 |      2 | M    | NULL |
|     5 | Thomas | 8000 |      2 | C    | 4    |
+-------+--------+------+--------+------+------+

mysql> select * from dept;
+--------+-------+------+
| deptno | dname | loc  |
+--------+-------+------+
|      1 | TRN   | Bby  |
|      2 | EXP   | Dlh  |
|      3 | MKTG  | Cal  |
+--------+-------+------+
mysql>  select dname,ename from emp Right outer join dept on (dept.deptno=emp.deptno);
+-------+--------+
| dname | ename  |
+-------+--------+
| TRN   | kiran  |
| TRN   | ali    |
| TRN   | arun   |
| EXP   | Thomas |
| EXP   | jack   |
| MKTG  | NULL   |
+-------+--------+

mysql>  select dname,ename from emp left outer join dept on (dept.deptno=emp.deptno);
+-------+--------+
| dname | ename  |
+-------+--------+
| TRN   | arun   |
| TRN   | ali    |
| TRN   | kiran  |
| EXP   | jack   |
| EXP   | Thomas |
+-------+--------+

select dname,ename from emp Right outer join dept on (dept.deptno=emp.deptno)
union
select dname,ename from emp left outer join dept on (dept.deptno=emp.deptno);
+-------+--------+
| dname | ename  |
+-------+--------+
| TRN   | kiran  |
| TRN   | ali    |
| TRN   | arun   |
```

```
| EXP    | Thomas |
| EXP    | jack   |
| MKTG   | NULL   |
+-------+--------+
```

```
mysql> select * from emp left outer join dept on (dept.deptno=emp.deptno)  where emp.deptno;
+-------+--------+------+--------+------+------+--------+-------+------+
| empno | ename  | sal  | deptno | job  | mgr  | deptno | dname | loc  |
+-------+--------+------+--------+------+------+--------+-------+------+
|     1 | arun   | 8000 |      1 | M    | 4    |      1 | TRN   | Bby  |
|     2 | ali    | 7000 |      1 | C    | 1    |      1 | TRN   | Bby  |
|     3 | kiran  | 3000 |      1 | C    | 1    |      1 | TRN   | Bby  |
|     4 | jack   | 9000 |      2 | M    | NULL |      2 | EXP   | Dlh  |
|     5 | Thomas | 8000 |      2 | C    | 4    |      2 | EXP   | Dlh  |
+-------+--------+------+--------+------+------+--------+-------+------+
```

```
mysql> select * from emp right outer join dept on (dept.deptno=emp.deptno)  where dept.deptno;
+-------+--------+------+--------+------+------+--------+-------+------+
| empno | ename  | sal  | deptno | job  | mgr  | deptno | dname | loc  |
+-------+--------+------+--------+------+------+--------+-------+------+
|     3 | kiran  | 3000 |      1 | C    | 1    |      1 | TRN   | Bby  |
|     2 | ali    | 7000 |      1 | C    | 1    |      1 | TRN   | Bby  |
|     1 | arun   | 8000 |      1 | M    | 4    |      1 | TRN   | Bby  |
|     5 | Thomas | 8000 |      2 | C    | 4    |      2 | EXP   | Dlh  |
|     4 | jack   | 9000 |      2 | M    | NULL |      2 | EXP   | Dlh  |
|  NULL | NULL   | NULL |   NULL | NULL | NULL |      3 | MKTG  | Cal  |
+-------+--------+------+--------+------+------+--------+-------+------+
```

Inner Join

* by default every join is as an Inner join
  putting a (+) sign is what makes it an Outerjoin
* do not mention in interviews unless explicitly by interviewer