**Date function**

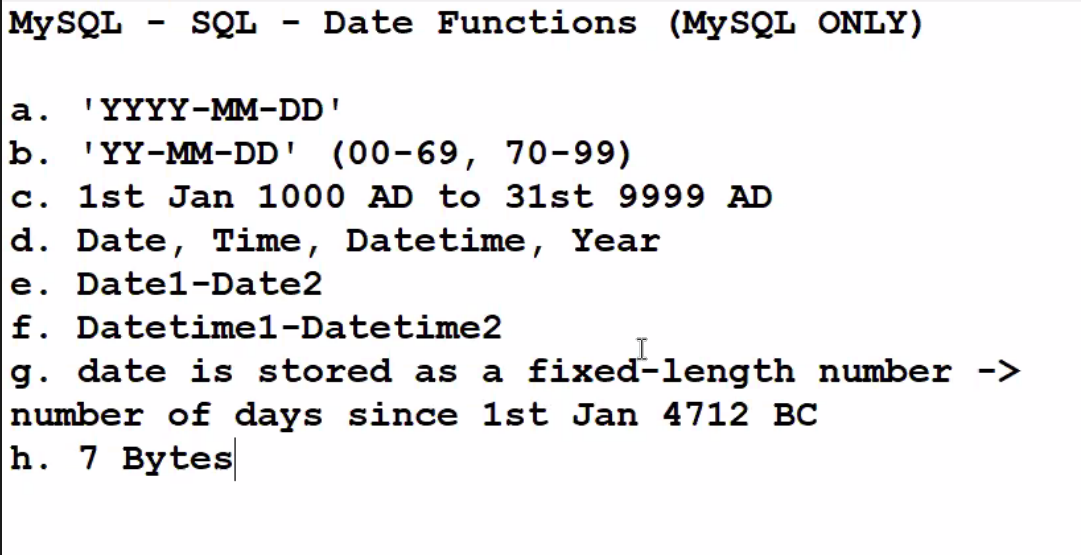
Emp table

Hiredate

2019-10-15

2019-12-31

2019-01-13



Select sysdate() from dual ; returns current date and time when the statement executed

* Sysdate() returns server date and time “***abhi abhi****”*

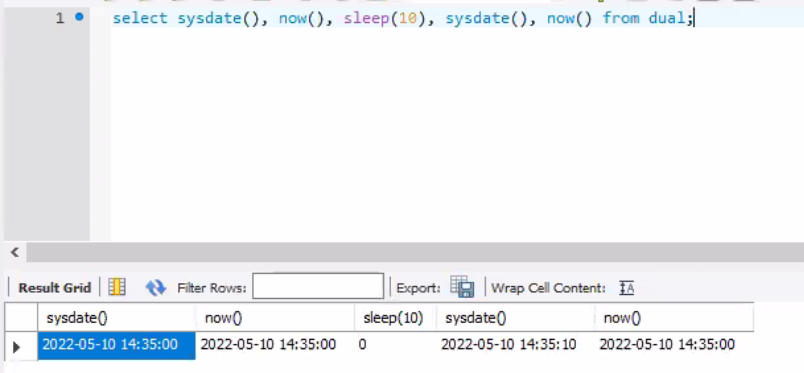
Select now() from dual; returns current date and time

When the statement began to execute

* now() returns server date and time

select sysdate(), now(), from dual;

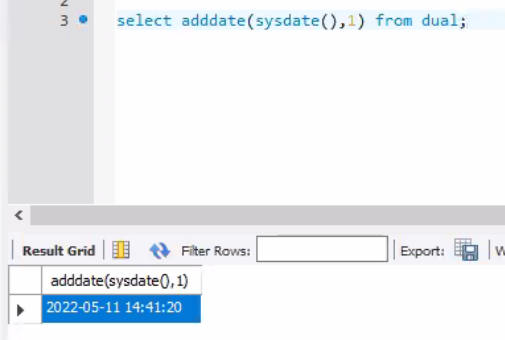
select sysdate(), now(), sleep(10), sysdate() ,now() from dual;



**Sysdate() -> date and time displays**

**Now()-> to maintain logs of operations of DML operations**

Select adddate(sysdate(),1) from dual; show next day adding 24 hrs



Select adddate(sysdate(),2) from dual; day after tomorrow

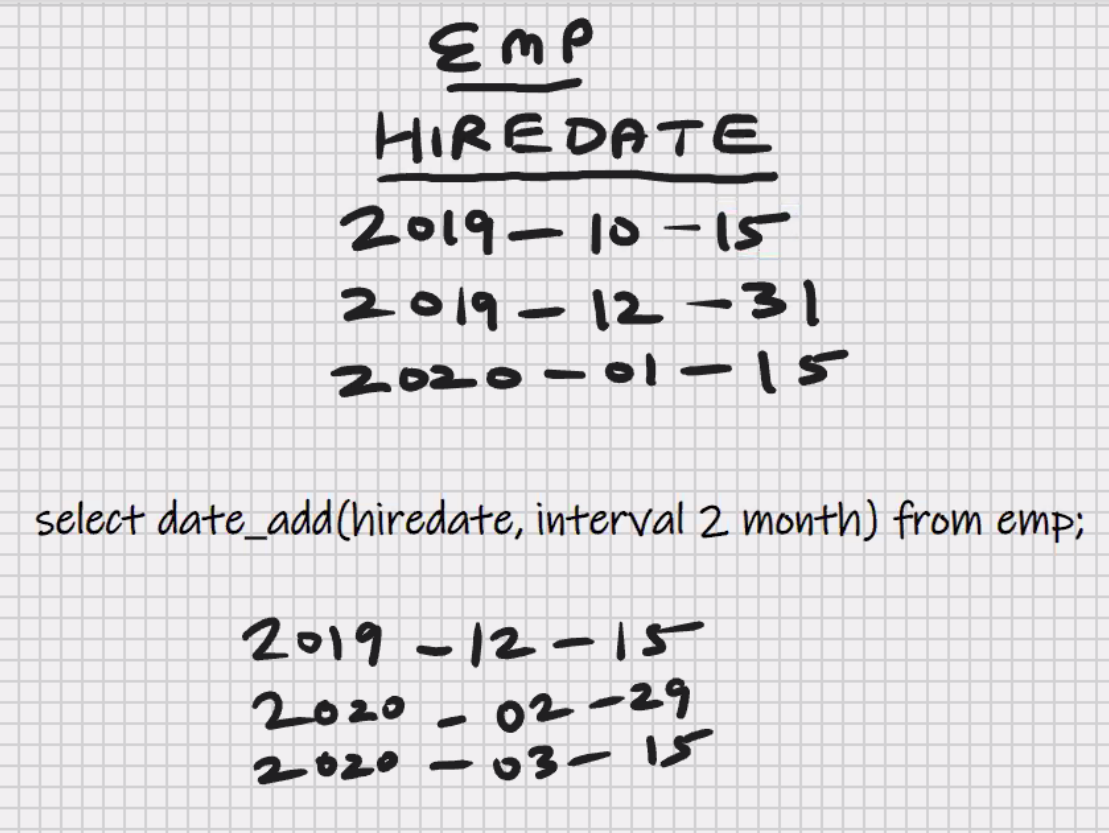
Select adddate(sysdate(),-1) from dual; yesterday

Select adddate(sysdate(),7) from dual; add week

Select datedif(sysdate(),hiredate) from emp;

🡪 Returns number of dates between 2 dates;

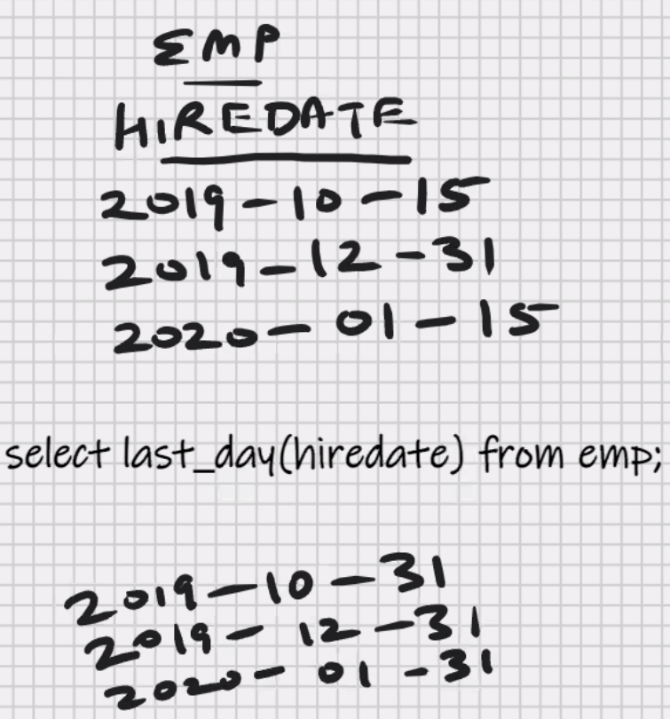
Select date\_add (hiredate, interval 2 month) from emp;



Select date\_add (hiredate, interval -2 month) from emp;

Select date\_add (hiredate, interval 1 year) from emp;

Select date\_add (hiredate, interval -1 year) from emp;



Select last\_day(hiredate) from emp;

**This function is present in only**

**Oracle and mysql**

Select dayname(sysdate()) from dual;

Tuesday

Select upper(dayname(sysdate())) from dual;

TUESDAY

Select addtime(‘2010-01-10’ 11:00:00’, ‘1’) from dual;

Add 1 second to specified date and time

Select addtime(‘2010-01-10’ 11:00:00’, ‘1:00:00’) from dual;

Add 1 hour to specified date and time

**List functions(independent of datatypes)**

Emp table

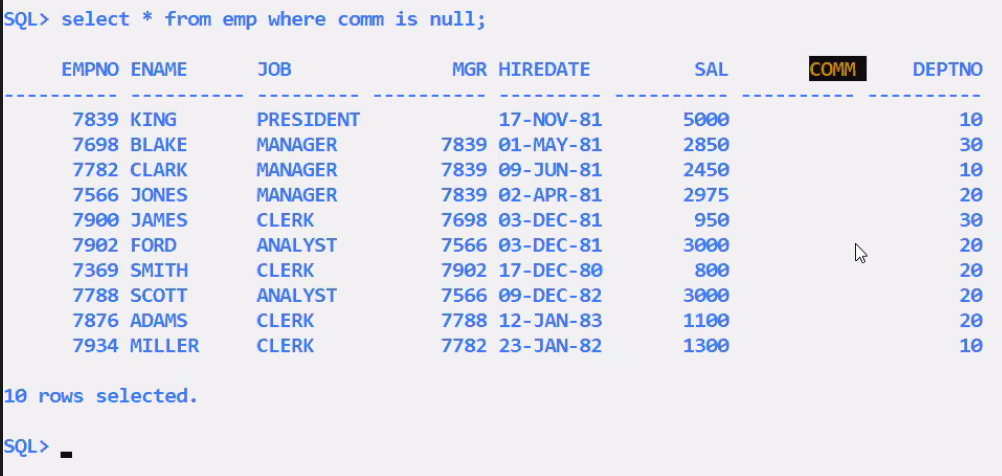
|  |  |  |
| --- | --- | --- |
| Ename | Sal | Comm |
| A | 5000 | 500 |
| B | 6000 |  |
| C |  | 700 |



Select \* from emp where comm=null;

* any comparisons done with null, returns null
* Pessimistict querying -> searching for null value

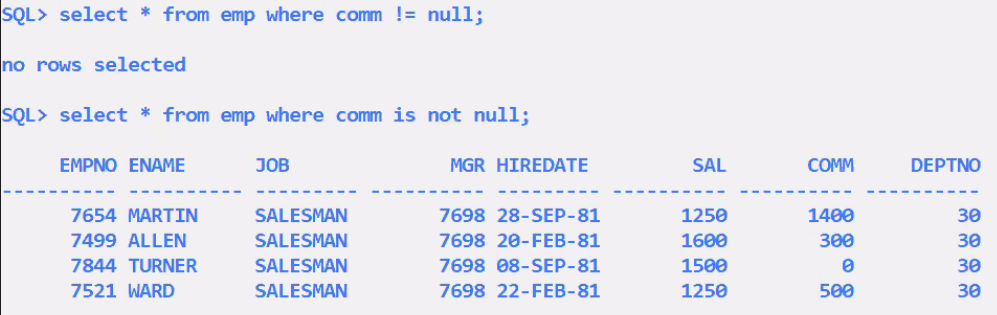
Select \* from emp where comm is null;



Select \* from emp where comm!=null;

* any comparisons done with null, returns null

select \* from emp where comm is not null;



* is null is a special operator

select sal+comm from emp;

5500

(Null)

(null)

* any operation done with null, returns null

in MySQL-

select sal+ ifnull(comm,0) from emp;

5500

6000

null

* if the comm is null returns 0;
* else return comm;
* end if;

select ifnull(sal,0) + ifnull(comm,0) from emp;

5500

6000

700

Ifnull(comm,0)

Ifnull(comm,100) -> if comm is null return 100

Ifnull(city,’Mumbai’)

Ifnull(orderdate,’2022-04-01’)

In oracle:-

Select nvl(sal,0) + nvl(comm, 0) from emp;

Nvl(comm,0)

Nvl(comm,100) -> if comm is null return 100

Nvl(city,’Mumbai’)

Nvl(orderdate,’2022-04-01’)

Emp table

Sal

1000

2000

3000

4000

5000

**Greatest function:**

Select greatest (sal,3000) from emp;

3000

3000

3000

4000

5000

Select greatest (sal,3000,4000,5000,6000,10000) from emp;

10000

10000

10000

10000

10000

Greatest (col1, col2, col3, ……….., col255)

Greatest (val1, val2, val3, ………., val255)

Greatest (num1, num2, num3)

Greatest (‘str1’,’str2’,’str3’,’str4’)

Greatest (date1, date2, date3)

Set x = greatest(a,b,c,d);

Least function:

Select least(sal,3000) from emp;

1000

2000

3000

3000

3000

Least (col1, col2, col3, ……….., col255)

Least (val1, val2, val3, ………., val255)

Least (num1, num2, num3)

Least (‘str1’,’str2’,’str3’,’str4’)

Least (date1, date2, date3)

Set x = least(a,b,c,d);

* greatest function used to set a lower limit on some value
* least function used to set a upper limit on some value

bonus= 10% sal, min bonus=300

select greatest(sal\*01,300) “BONUS” from emp;

cashback = 10% amt, max cashback = 1000;

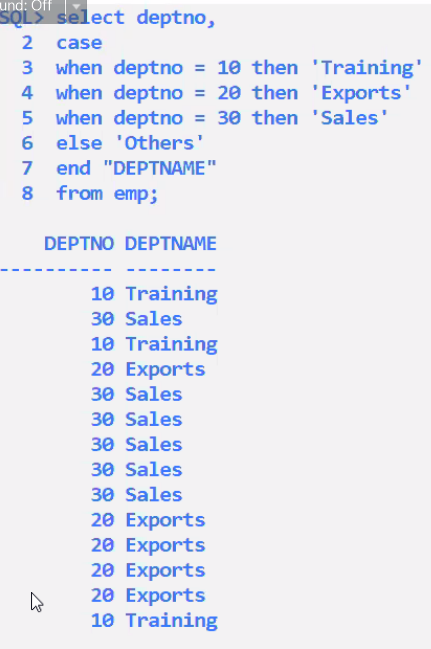
least(amt\*0.1,1000) from orders;

CASE expression

EMP TABLE

|  |  |
| --- | --- |
| SAL | DEPTNO |
| 1000 | 10 |
| 2000 | 10 |
| 3000 | 20 |
| 4000 | 30 |
| 5000 | 40 |

Select

Case

When dept=10 then ‘Training’

When dept=20 then ‘Exports’

When dept=30 then ‘Sales’

Else ‘Others’

End “DEPTNAME”

From emp;

Training

Training

Exports

Sales

Others

ELSE is optional in case expression

If else is not specified then for all other values,

It returns null values

**It is like switch case statement**

Select

Case

When dept = 10 then ‘Ten’

When dept = 20 then ‘Twenty’

When dept = 30 then ‘Thirty’

When dept = 40 then ‘Forty’

End “DEPTNO”

From emp;

If deptno = 10 then HRA = 40% annual

If deptno= 20 then HRA = 30% annual

If deptno = 30 then HRA = 20% annual

Select deptno, ename, sal\*12 annual,

Case

When deptno=10 then sal\*12\*0.4

When deptno=20 then sal\*12\*0.3

When deptno=30 then sal\*12\*0.2

Else sal\*12\*0.1

End “HRA”

From emp;

If sal>3000 then REMARK = ‘High Income’

If sal<3000 then REMARK = ‘Low Income’

If sal=3000 then REMARK = ‘Middle Income’

Select ename, sal

Case

When sign(sal-3000)=1 then ‘High Income’

When sign(sal-3000)=-1 then ‘Low Income’

Else ‘Middle Income’

End “REMARKS”

From emp

Order by 2;

**MySQL – SQL – Environment functions**

In MySQL-

Select user() from dual;

‘dishi@localhost’

* used to maintain the logs of DML operations

emp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ename | Sal | Col1 | Col2 | Col3 |
| King | 5000 | Dishi | 2022-05-10 | Etc |
|  |  |  |  |  |

Insert into emp values

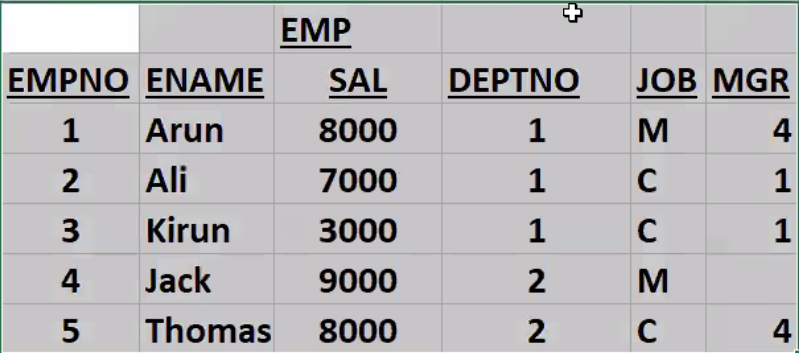
(‘King’,5000,user(),now(),etc);

Show character set;

In Oracle:

Select user from dual;

**Group / Aggregate functions**



Select

Case

When job = ‘M’ then ‘MANAGER’

When job = ‘C” then ‘CLERK’

End “JOB”

From emp;

**Single row functions:**

Operate on one row at a time:

Character, number, date, list, environment functions

e.g. upper(ename), round(sal), etc.

**Multi-Row functions:**

Operate on multiple rows at a time

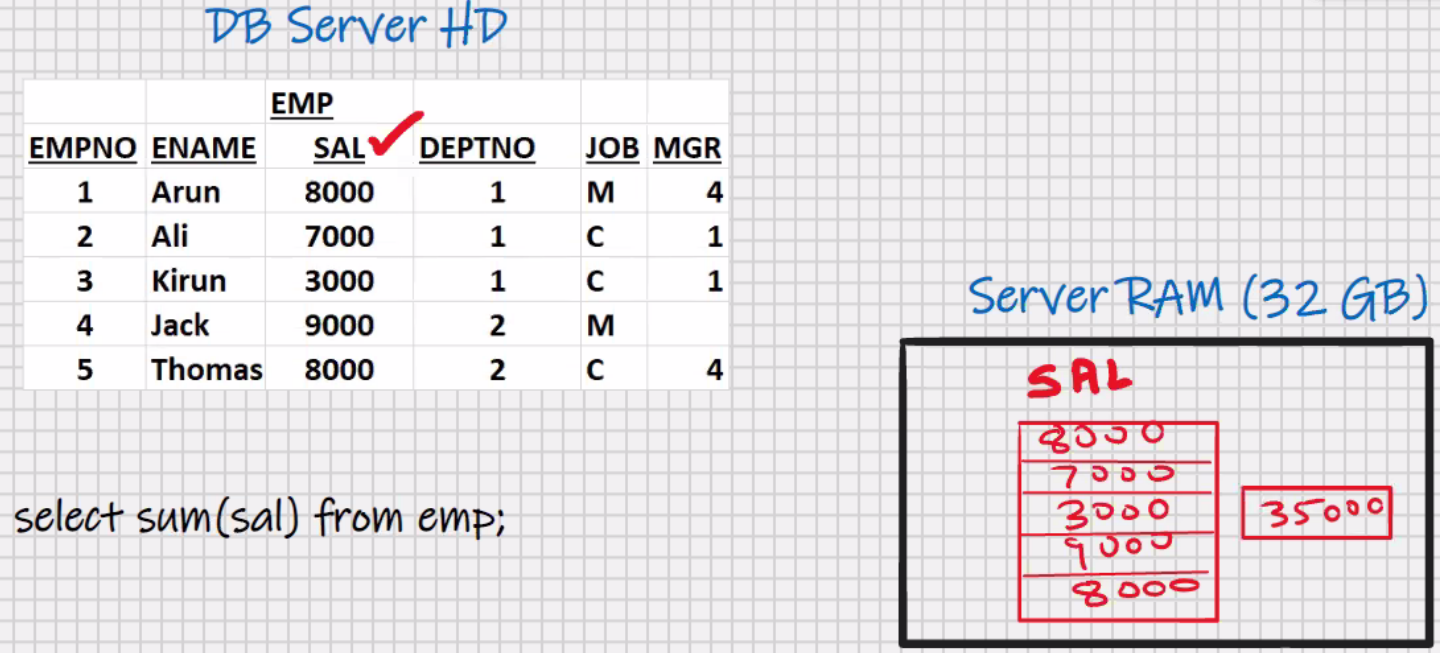
e.g sum(sal), min(sal), max(sal), etc.

**group functions:**

**sum function:**

select sum(sal) from emp;

35000



**Assumption 4th row sal is null**:-

**Sum function:**

Select sum(sal) from emp; <- null values are not counted by group functions

26000

Select sum(ifnull(sal,0)) from emp; <- you may do this but not required

26000

**Average function-avg:**

Select avg(sal) from emp;

26000/4=6500

Select avg(ifnull(sal,0)) from emp;

26000/4=5200

**Minimum function:**

Select min(sal) from emp;

3000

Select min(ifnull(sal,0)) from emp;

0

**Maximum function:**

Select max(sal) from emp;

8000

Select max(ifnull(sal,0)) from emp;

0

**Count function:**

Select count (sal) from emp;

* Return number of rows where sal is not having null value

4

Select count (\*) from emp;

* Returns count of total number of rows in the table

5

Select count(\*)-count(sal) from emp;

1

Select max(sal)/min(sal) from emp;

8000/3000=2.67

Select sum(sal)/count(\*) from emp;

26000/5=5200

Select avg(ifnull(sal,0)) from emp;

26000/5=5200

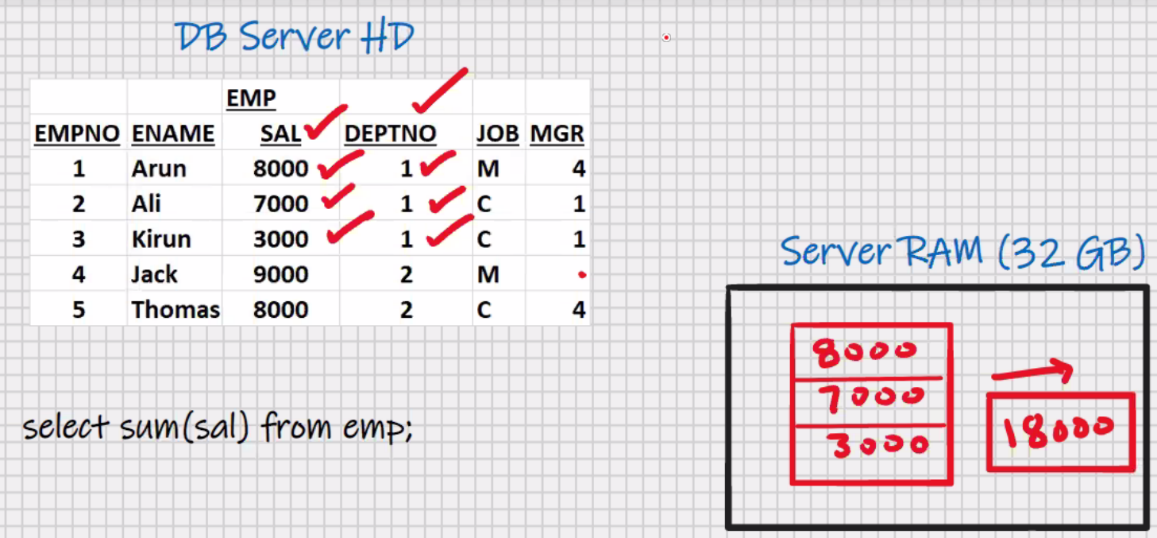
**Assumption 4th row sal Is 9000:-**

Select sum(sal) from emp

Where deptno=1;

18000

* Where clause is used for searching
* Searching takes place in DB server HD



Select avg(sal) from emp

Where job = ‘C’;

6000

**COUNT-QUERY:**

Counting the number of Query hits:-

Select count(\*) from emp ->**VERY IMPORTANT**

Where sal>7000;

3

Sum(column)

avg(column)

min(column)

max(column)

count(column)

count(\*)

stddev(column)

variance(column)

min(sal), min(ename), min(hiredate)

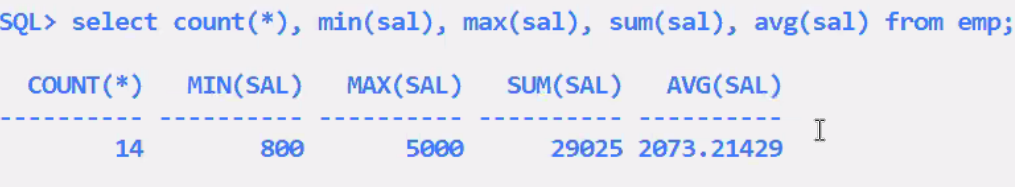
max(sal), max(ename), max(hiredate)

count(sal), count(ename), count(hiredate)

* used for all datatypes

select sum(sal), avg(sal), min(sal), max(sal) from emp;

* multiple group functions can be selected



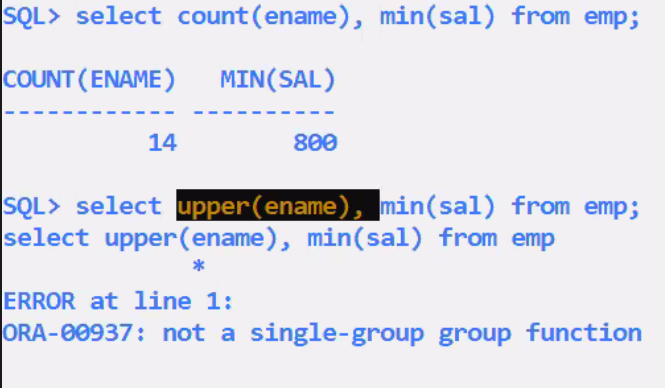
* summary of my table

**3 RESTRICTIONS:-**

Select ename, min(sal) from emp; <- error in oracle (works in MySQL but the output is meaningless)

1. you cannot select a column of table by itself, alongwith a group function

select upper(ename), min(sal) from emp; <- error in oracle (works in MySQL but the output is meaningless)



2.you cannot select a single row function alongwith a group function

Select \* from emp

Where sal>avg(sal); -> Error in MySQL and Oracle

3.you cannot use group function in the where clause

(Where clause is used for searching; the searching takes place in DB server HD at the time of searching in the table the avg(sal) doesn’t exist; the avg (sal) is calculated by MySQL afterwards (after the rows are brought into server ram))

**My-SQL- SQL-group by clause(Very Important)**

* used for grouping

select sum(sal) from emp;

35000

Select sum(sal) from emp

Where deptno=1;

18000

Select deptno, sum(sal) from emp

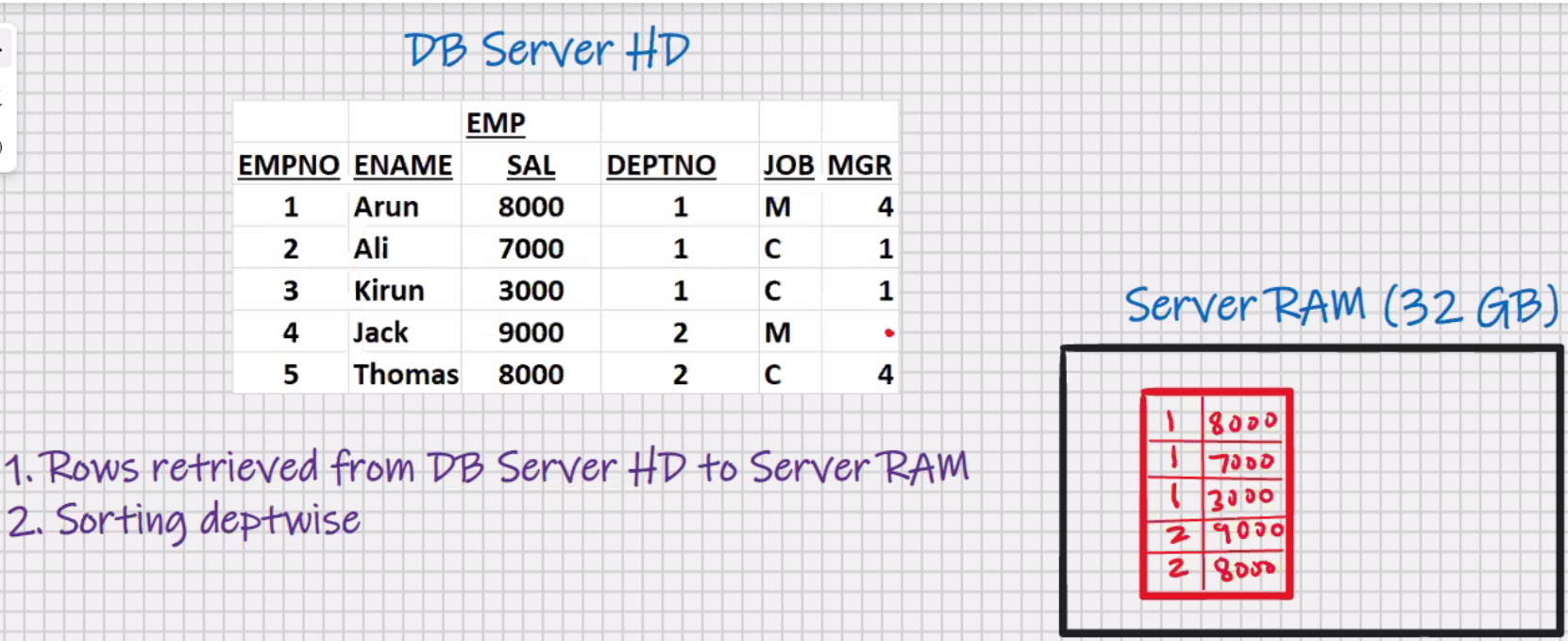
Group by deptno;

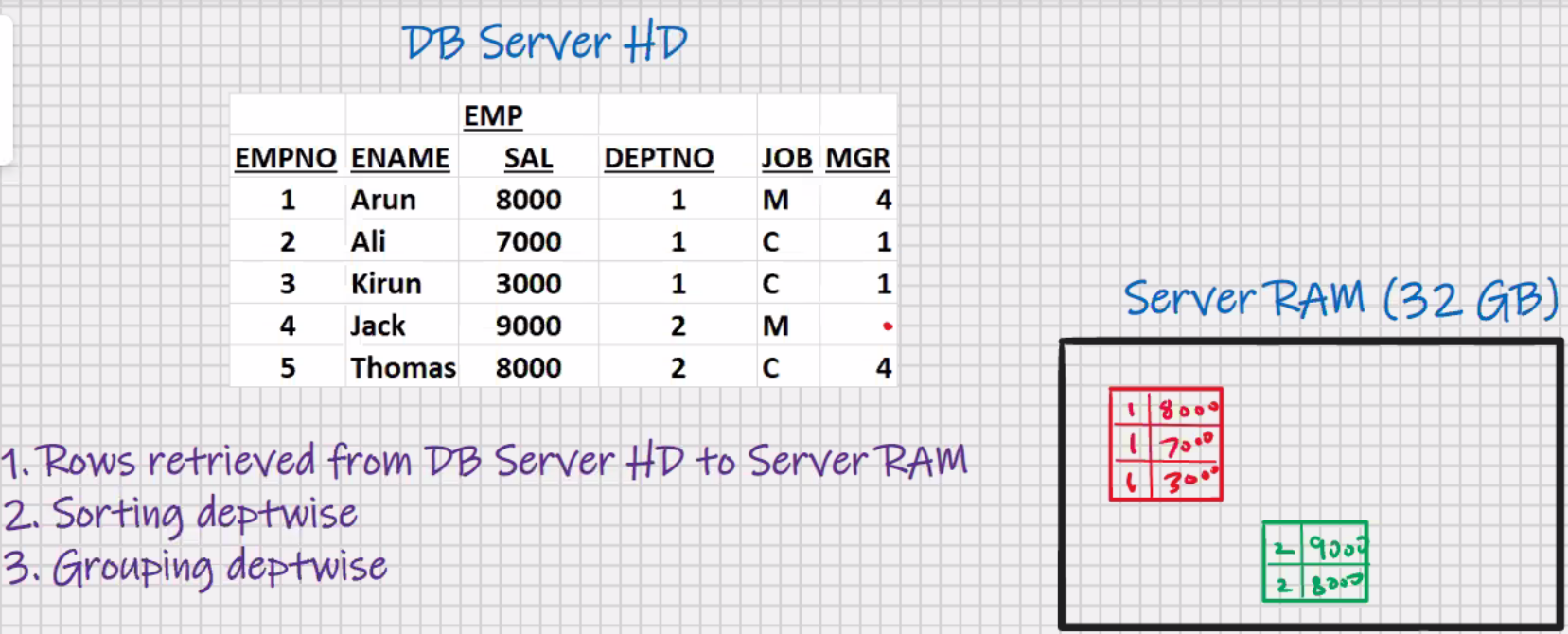
DEPTNO SUM(SAL)

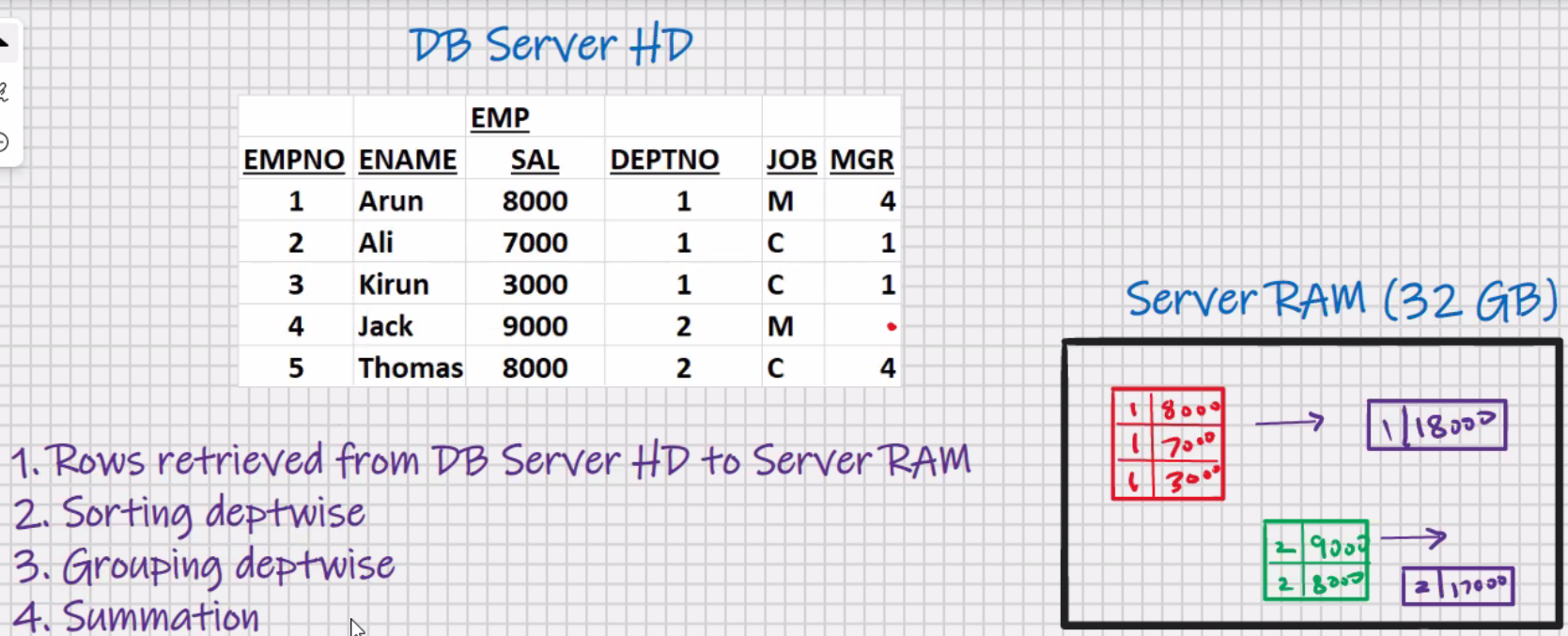
----------- -------------

1 18000

2 17000







1. Besides the group function, whichever column is present in select clause; it has to be present in group by clause

Select deptno, sum(sal) from emp; -> ; <- error in oracle (works in MySQL but the output is meaningless)

Select deptno, sum(sal) from emp

Group by deptno;

1. whichever column is present in group by clause it may or may not be present in select clause

Select sum(sal) from emp

Group by deptno;

SUM(SAL)

-------------

18000

8000

Select deptno, sum(sal) from emp

Group by deptno;

Select deptno, max(sal) from emp

Group by deptno;

Select deptno,min(sal) from emp

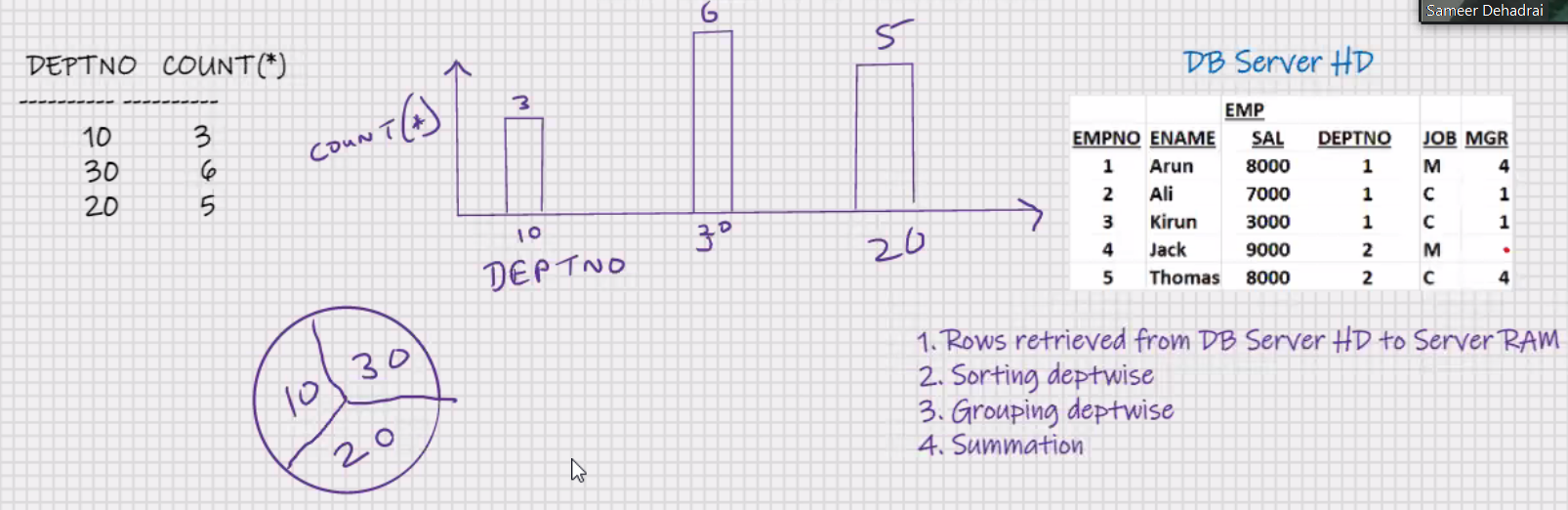
Group by deptno;

Select deptno, count(\*) from emp

Group by deptno;

2D query-> any select statement with a group by clause is known as a 2D query, because you can plot a graph from the output e.g. x-y grapgh, bar graph, pie chart, etc.

#1 Oracle Graphics



Select job, sum(sal) from emp

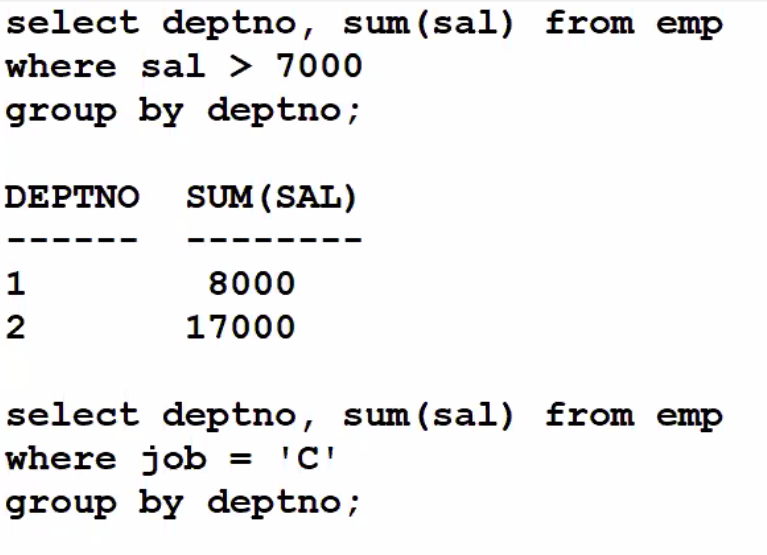
Group by job;

Select job, sum(sal) from emp

Where sal>7000

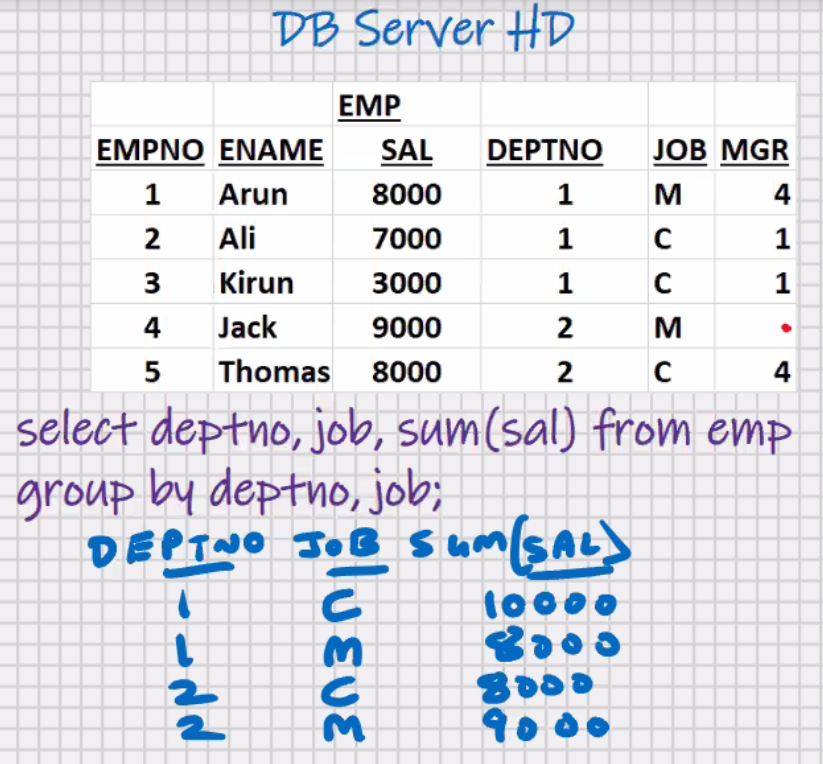
Group by job;

* **WHERE** **clause has to be specified before the group by clause**
* WHERE clause is used for searching ; the searching takes place in DB server HD
* WHERE clause is used to restrict the rows
* WHERE clause is used to retrieve the rows from DB server HD to server RAM



Select deptno, job, sum(sal) from emp

Group by deptno, job;



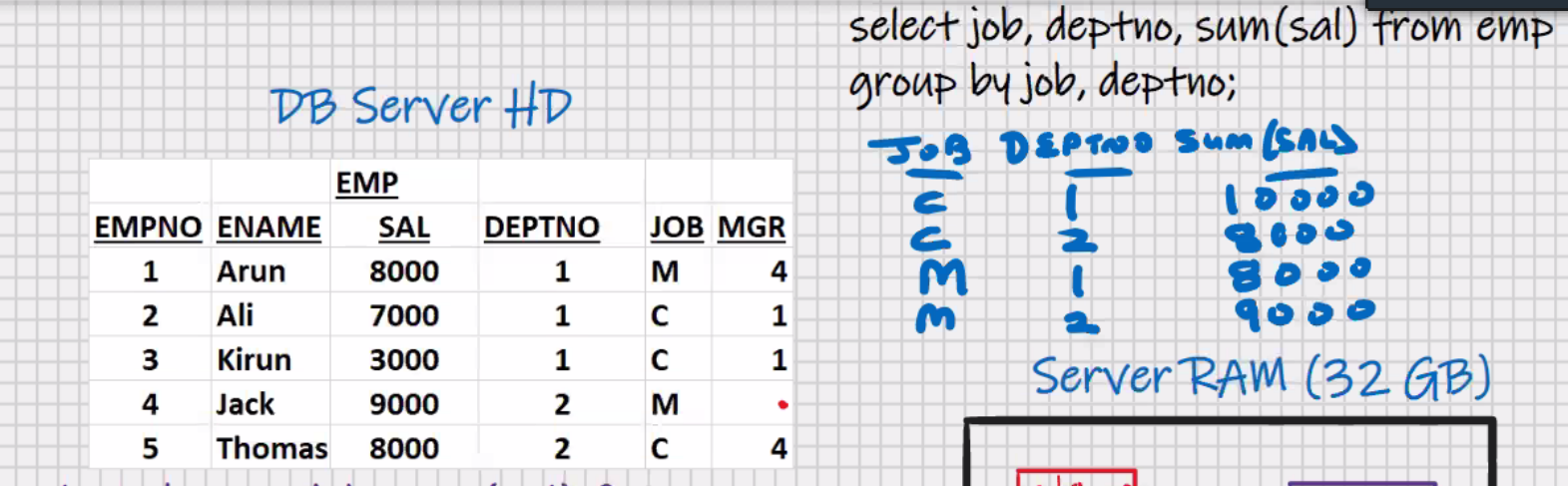
* No upper limit on the number of columns in Group by clause

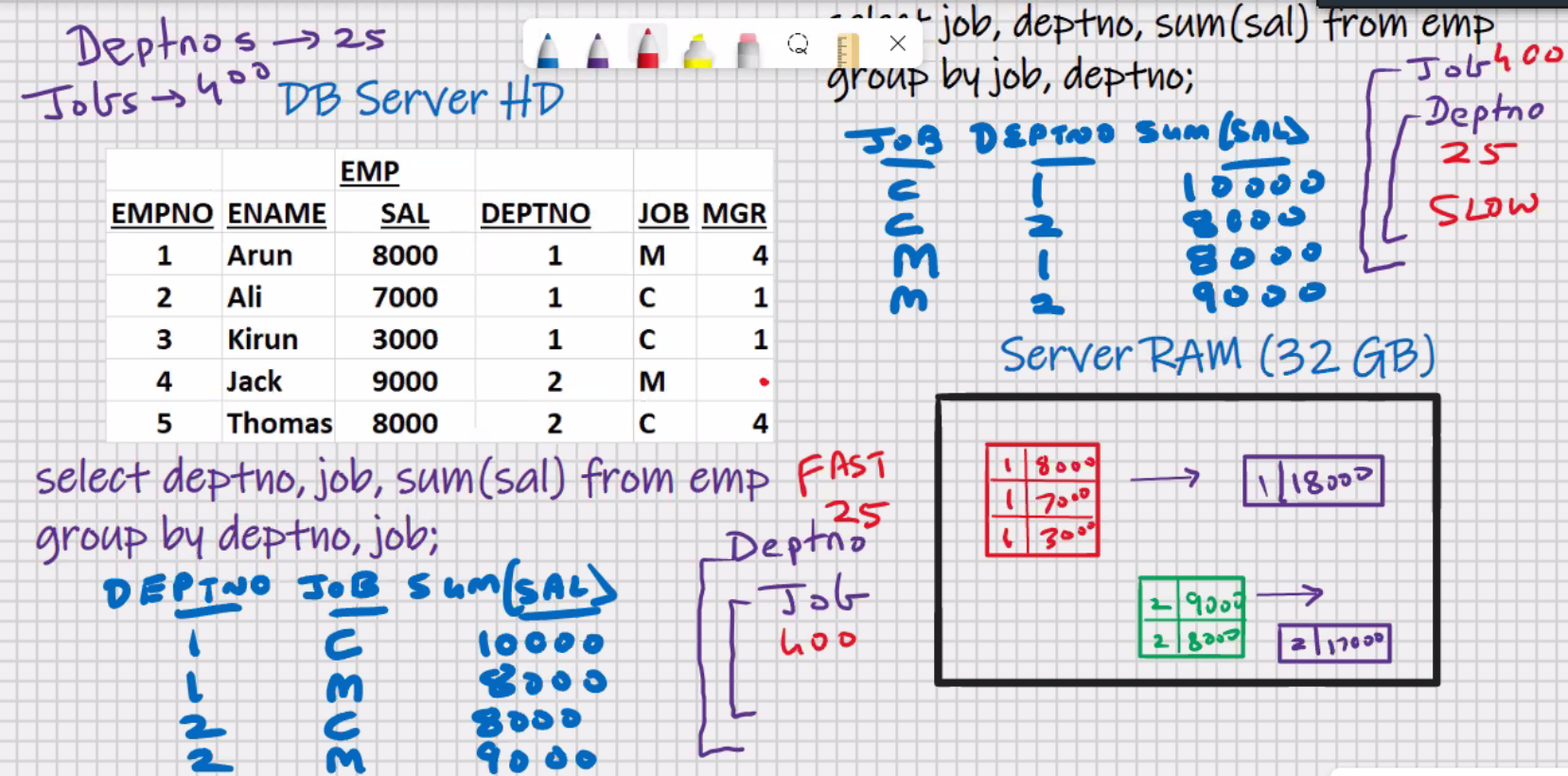
Select ………..

Group by country, state, city;

Select job, deptno sum(sal) from emp

Group by job, deptno;

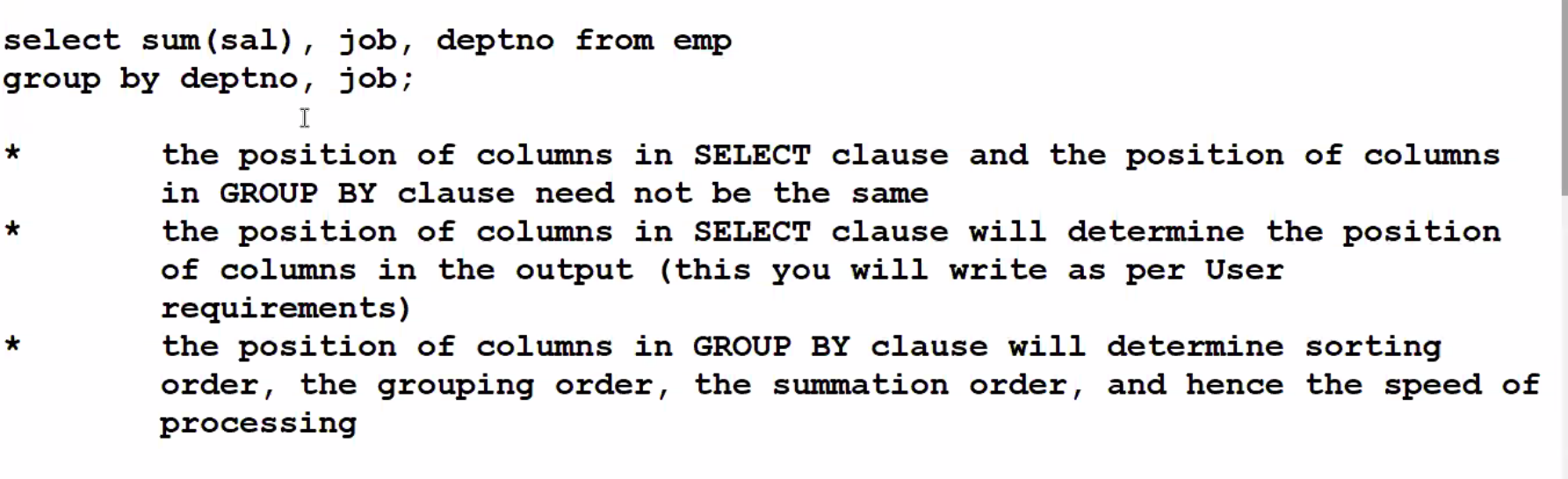


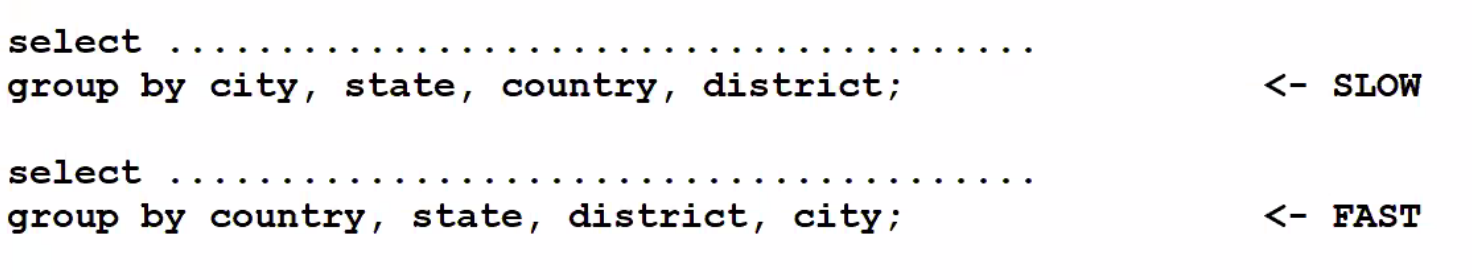


Select job, deptno sum(sal) from emp

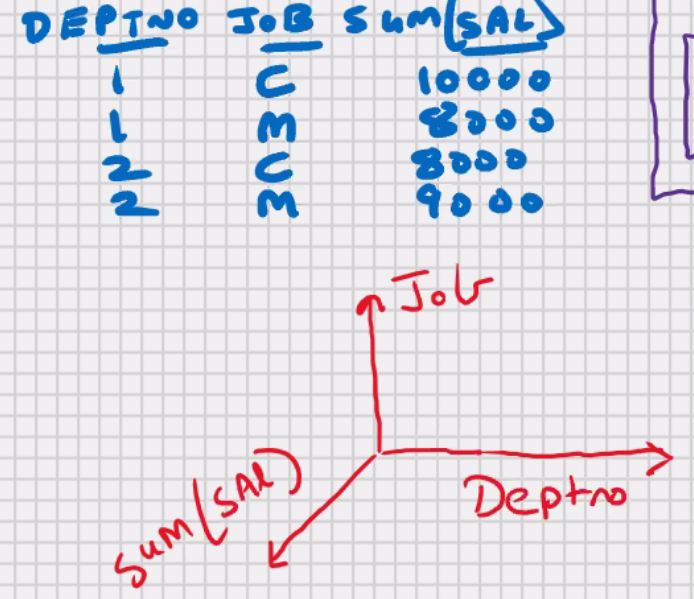
Group by deptno, job;

* The position of columns in select clause and the position of columns in group by clause need not be the same
* The position of columns in select clause will determine the position of columns in the output(this you will write as per user requirements)
* The position of columns in group by clause will determine sorting order, the grouping order, the summation order, and hence the speed of processing





Countries are less in number compare to states, districts, cities

Select deptno, job, sum(sal) from emp

Group by deptno, job;

* If you have 1 column in group by clause

It is 2D Query

* If you have 2 column in group by clause

It is 3D Query

* If you have 3 column in group by clause

It is 4D Query

* Known as **Multidimensional queries**(also known as **spatial queries**)