

1. display all employees who joined on 1-Jan-1983 or 2 feb 1984

```
select * from emp
where hiredate in ('1983-01-01','1984-02-02')
```

2. display all employees whose commission is not null and joined in year 1985

```
select * from emp
where comm is not null and hiredate between '1985-01-01' and '1985-12-31'
```

3. display all employees whose name starts with J or M or K

```
select * from emp
where ename regexp '^[JMK]';
```

```
select * from emp
where ename like 'J%' or ename like 'M%' or ename like 'K%'
```

4. display all employees who are working as clerk or salesman and working in department 10

```
select * from emp
where job in ('CLERK','SALESMAN') and deptno=10
```

5. display all employees whose mgr is 7902

```
select * from emp
where mgr=7902;
```

#### Distinct keyword

1. display distinct jobs in emp table

```
select distinct job
from emp;
```

1. distinct job and sal

```
select distinct job,sal
from emp;
```

#### limit clause

1. to display first 2 rows

```
select * from emp
limit 2;
```

2. display 3,4 and 5 row

following query will skip 2 rows and display 3 rows

```
select * from emp
limit 2,3;
```

To arrange data in sorted order

```
select *  
from emp  
order by sal
```

display all employees of dept 10 or 20 sorted on deptno

```
select *  
from emp  
where deptno in (10,20)  
order by deptno;
```

display all employees of dept 10 or 20 sorted on deptno in descending order

```
select *  
from emp  
where deptno in (10,20)  
order by deptno desc;
```

display employee with 5<sup>th</sup> highest salary

```
select *  
from emp  
order by sal desc  
limit 4,1
```

Function in mysql

functions are of 2 types

1. single row functions –if a function takes input from one row and gives the o/p, then it is called as single row functions

2. aggregate functions--- if a function takes input from multiple rows and gives the o/p, then it is called as aggregate functions  
to use aggregate functions, we use group by and having  
sum, count,min, max and avg these are aggregate functions

1. to display sum, min, max, average, count from all employees and also display not null values in comm column.

```
select sum(sal),min(sal),max(sal),avg(sal),count(*), count(comm)
from emp;
```

2. display sum min, max, avg for each department

```
select deptno, sum(sal),min(sal),max(sal),avg(sal),count(*), count(comm)
from emp
group by deptno
order by deptno; ----good to write
```

1. display sum min, max, avg for each department and for each job

```
select deptno,job, sum(sal),min(sal),max(sal),avg(sal),count(*), count(comm)
from emp
group by deptno,job
order by deptno,job; ----good to write
```

When we use aggregate function in query, then in select statement you can add only columns which are in group by clause, other than aggregate functions.

2. display sum of salary for all employees working under same mgr.

```
select mgr,sum(sal)
from emp
group by mgr;
```

3. display sum,min,max,avg of salary for all employees working in dept 10

```
select deptno,sum(sal),min(sal),max(sal),avg(sal)
from emp
where deptno =10;
```

4. display sum,min,max,avg of salary for all employees with sal > 2000, department wise,

```
select deptno,sum(sal),min(sal),max(sal),avg(sal)
```

```
from emp  
where sal>2000  
group by deptno;
```

5. find how many CLERKS are there in each department.

```
select job,deptno,count(*)  
from emp  
where job='CLERK'  
group by deptno;
```

6. find all departments in which 2 or more clerks are there

```
select job,deptno,count(*)  
-> from emp  
-> where job='CLERK'  
-> group by deptno  
-> having count(*)>=2  
Order by cout(*)
```

if the condition is based on column which is existing in the table then use condition in where clause, and if the condition is based on aggregate functions then use it in having cluse;

7. find sum of net sal, count, min net sal, max net sal for all clerks, if the sal <3000

```
select sum(sal+ifnull(comm,0)),count(*),min(sal+ifnull(comm,0)), max(sal+ifnull(comm,0))  
from emp  
where job='CLERK' and sal<3000;
```

```
select job,sum(sal)  
from emp  
where job='CLERK' and sal<3000;
```

8. display sum of net sal for all clerks, if the sal <3000 department wise

```
select deptno,sum(sal+ifnull(comm,0))
from emp
where job='CLERK' and sal<3000
group by deptno;
```

9. display sum, avg for netsalary of all employees department wise and arrange it in descending order of sum of netsal  
netsal can be calculated as sum+comm

```
select deptno,sum(sal),sum(sal+ifnull(comm,0)) sumsal, avg(sal+ifnull(comm,0))
average
from emp
group by deptno
order by sum(sal+ifnull(comm,0)) desc
```

### Single row functions

Functions which give 1 o/p for each row, are called as single row functions.

1. number
2. string
3. date

### Number functions

abs(num)	it will convert -ve value to +ve val
sqrt(num)	to find sqrt of the number
ceil(num)	It will always give the next minimum number ceil(3.12)=4 ceil(3.65)=4
floor(num)	It will always give the previous maximum number floor(3.12)=3 floor(3.76)=3
round(num,precesion)	it will round the value upto precision round(1.4567,2)=1.46

	round(1.4512,2)=1.45
truncate(num,precision)	It will truncate the value upto precision truncat(1.4567,2)=1.45 truncat(1.4512,2)=1.45
pow(num,raiseto)	will find num raiseto pow(3,2)=9
Mod(num,divisor)	will find mod value mod(5,2)=1

#### String functions

upper(val)	convert string into uppercase
lower(val)	convert string into lowercase
concat(str1,str2,str3,.....)	it combine all the strings to form single string
substr(str,start,length)	will retrieve portion of the string starting from start value, length number of characters, length is optional, if length is not given, then It will provide all characters till end
left(str,length)	will retrieve length number of leftmost characters
right(str,length)	will retrieve length number of rightmost characters
trim(str)	it will remove all leading and trailing spaces
rtrim(str)	it will remove all trailing spaces
ltrim(str)	it will remove all leading spaces
instr(str,substr)	it will return the position of first occurrence of substr in the given string
replace (str,oldstr,newstr)	It will replace all occurrence of oldstr with newstr in the given string
format(value,precision)	It displays thousand separators in the number 300,000,000.00
lpad(str,length,character)	add given character on left of the sting, so that the total length will be length select empno,ename,concat(ename,"-----",job),concat(rpad(ename,12,'-'),job) -> from emp;
rpadd(str,length,character)	add given character on right of the sting, so that the total length will be length
length(str)	number of characters in the given string

11-09-2025

#### date function

now()	it will display current date and time
curdate()	it will display current date

date_format(date,format)	to display the date in specified format Y --- will display 4 digit year y----will display 2 digit year M---- month name in character m----month in number d-----date in number D- display th or st after date b--- display months in 3 letter (jan, feb,...) r ---- to print time in 12 hrs (hh:mm:ss AM/PM) %W---- (Sunday..Saturday) %w--- Day of the week (0=Sunday..6=Saturday)
date_add(date,interval)	it will find the date after given interval date_add(curdate(),interval 2 day) date_add(curdate(),interval 2 month) date_add(curdate(),interval 2 year)
date_sub(date,interval)	it will find the date after given interval date_add(curdate(),interval 2 day) date_add(curdate(),interval 2 month) date_add(curdate(),interval 2 year)
datediff(date1,date2)	find the difference between 2 dates
timestampdiff(YEAR, date1, CURDATE())	find difference between 2 dates in terms of years  SELECT TIMESTAMPDIFF (YEAR, YOUR_COLUMN, CURDATE()) FROM YOUR_TABLE AS AGE
day(date) month(date) year(date) week(date) quarter(date)	day or month or year or week or quarter of the given date then use these functions
extract(date, fmt)	to find day month or year from date extract(month from curdate()) extract(year from curdate()) extract(day from curdate())
monthname(date)	it will find name of month in character
dayname(date)	it will display dayname of the given date
last_day(date)	it will display last day of the current month

to find years of experience for all employees

```
select empno,ename,hiredate,timestampdiff(YEAR,hiredate,curdate()) experience
```

```
-> from emp;
```

display all employee with experience > 42

```
select empno,ename,hiredate,timestampdiff(YEAR,hiredate,curdate()) experience
```

```
-> from emp
```

```
where timestampdiff(YEAR,hiredate,curdate())>=42
```



display all products which will expire after 3 months

```
select *
```

```
from perishableprod
```

```
where expdate >= date_add(curdate(),interval 3 month)
```

```
date_sub(expdate,interval 3 months)>=curdate()
```

1. display the meeting date which is 3 months 20 days from the date tomorrow.

```
select date_add(date_add(curdate(),interval 21 day),interval 3 month)
```

Case statement

if comm is null or 0 then poor performance

if comm <=300 then 'ok performance'

if >300 and <=500 then 'good performance'

otherwise excellent performance

```
select empno,ename,sal,comm,
```

```
case when comm is null or comm=0 then 'poor performance'
```

```
when comm<=300 then 'ok performance'
```

```
when comm<=500 then 'good performance'
```

```
else 'excellent performance' end comment
```

```
from emp;
```

2. dept no 10 the display admin

if 20 then display HR

otherwise display network

```
select empno,ename,sal,deptno,
```

```
case when deptno=10 then 'admin'
```

```
when deptno=20 then 'HR'
```

```
else 'network' end dname
```

```
from emp;
```

```
select empno,ename,sal,deptno,
```

```
-> case deptno when 10 then 'admin'
```

```
-> when 20 then 'HR'
```

```
-> else 'network' end dname
```

```
-> from emp;
```

