**DQL**

create database my\_db;

use my\_db;

create table employee(

id int primary key auto\_increment,

name varchar(20),

per\_day int,

total\_leaves int,

bouns int

);

alter table employee rename column bouns to bonus;

show tables;

desc employee;

insert into employee

(name,per\_day,total\_leaves,bonus)

values

("raj",1000,5,4500),

("rani",2500,3,1000),

("aniket",3000,8,null),

("sumit",500,2,800);

select \* from employee;

-- SELECT WITH ARITHMATIC OPERATORS : +,-,\*,/

-- show monthly salary of each employee

select name,per\_day\*30 as salary from employee;

-- show salary after giving bonus

select name

, per\_day\*30+bonus as salary\_bonus from employee;

select name,per\_day\*30 as salary, per\_day\*30+bonus as salary\_bonus from employee;

-- show salary after removing leaves

select name, per\_day\*(30-total\_leaves) as final\_salary from employee;

-- make salary half of each employee

select name,(per\_day\*30)/2 as half\_salary from employee;

-- SELECT WITH COMPARISION OP

select \* from employee where per\_day<1000;

select \* from employee where per\_day>1000;

select \* from employee where per\_day<=1000;

select \* from employee where per\_day>=1000;

select \* from employee where per\_day=1000;

select \* from employee where per\_day!=1000;

select \* from employee where per\_day<>1000;

-- show employees who are getting salary more than 30k

select \* from employee where per\_day\*30>30000;

-- SELECT WITH LOGICAL OPRATOR : and, or, not

-- show employees who are getting per\_day more than 1000 as well as has leaves more than 5 days

select \* from employee where per\_day>1000 and total\_leaves>5;

-- show employees who are getting either per\_day more than 1000 or has leaves more than 5 days

select \* from employee where per\_day>1000 or total\_leaves>5;

-- show employees who are getting either per\_day more than 5000 or has leaves more than 5 days

select \* from employee where per\_day>5000 or total\_leaves>5;

-- show me employees whos name is not raj

select \* from employee where not name = "raj";

-- SLEECT WITH IN , NOT IN OPERATOR

-- show empoyees getting per\_day either 1000 or 2500 or 3000

select \* from employee where per\_day=1000 or per\_day=2500 or per\_day=3000;

-- instead of using multiple OR statements, use IN operator

select \* from employee where per\_day in(1000,2500,3000);

select \* from employee where per\_day not in(1000,2500,3000);

-- SELECT WITH RANGE OPERATOR (BETWEEN)

-- show employees getting per\_day in range 500-2500

select \* from employee where per\_day between 500 and 2500;

-- IMPORTANT : here 500 and 2500 both values will be included

-- show employees who are not getting per\_day in range 500-2500

select \* from employee where per\_day not between 500 and 2500;

-- SELECT WITH LIKE

-- % means any characher with any number

-- show me employees whos name starts with 'r'

select \* from employee where name like 'r%';

-- show me employees whos name starts with 'ra'

select \* from employee where name like 'ra%';

-- show me employees whos name ends with 't'

select \* from employee where name like '%t';

-- show me employees whos name ends with 'et'

select \* from employee where name like '%et';

-- when we want to specify number of characters (not characters) then use \_

select \* from employee where name like 'r\_j';

-- exactly 4 characters before 't'

select \* from employee where name like '\_\_\_\_t';

-- exactly 5 characters before 't'

select \* from employee where name like '\_\_\_\_\_t';

-- name must have 'a' as second character

select \* from employee where name like '\_a%';

-- SELECT WITH LIMIT

-- below query will return only top 2 recods

select \* from employee limit 2;

select \* from employee where name like 'r%' limit 1;

-- SELECT WITH ORDER BY (defaulr oeder is ascending)

-- sort employees based on per\_day ascending order

select \* from employee order by per\_day;

-- sort employees based on per\_day descending order

select \* from employee order by per\_day desc;

-- sort employees based on name ascending order

select \* from employee order by name;

-- order by with where condition

select \* from employee where total\_leaves>2 order by total\_leaves;

-- SELECT WITH IS NULL

select \* from employee where bonus is null;

select \* from employee where bonus is not null;