show databases;

use shop;

desc shop.products;

#dealing with data base

create database sql\_practice;

drop database sql\_practice;

#dealing with tables using DDL commands

use student ;

desc student ;

create table test2(

id int primary key,

name varchar(90),

doj varchar(20),

salary int,

department varchar(20)

);

desc test;

ALTER TABLE test modify

id int primary key auto\_increment;

#adding column and making it as forign key

ALTER TABLE test ADD

id int primary key references student(id);

ALTER TABLE test add

age int default(20);

ALTER TABLE test add

department varchar(20) default("HR");

alter table test modify column

department varchar(30) default("IT");

ALTER TABLE test add

Salary int check(salary>2500) default(30000);

ALTER TABLE test modify

Salary int default(30000) check(salary>2500) ;

ALTER TABLE test add

department1 varchar(20) default("HR");

alter table test drop id;

# Deleting table

drop table test;

# Data Manipulation Language (DML)

insert into test values(

"Basharat",44,66000,116,"HR"),

("Khizar",30,82000,117,"IT"),

("Adil",40,53000,118,"BBA"),

("Tabarak",40,63000,119,"BBA")

;

select \* from test2;

truncate test2;

insert into test2 values(

101,"Basharat","2009-2-2",22000,"HR"),

(102,"Khizar","2007-4-14",65000,"IT"),

(103,"Adil","2005-1-2",50000,"BBA"),

(104,"Tabarak","2006-1-20",63000,"BBA")

;

select \* from test;

update test set salaray=93000

where id=106;

delete from test where id=104;

select department from test;

select distinct department as dep

from test;

select distinct age from test;

# agregate functions

select count(age) as counting\_age

from test;

select sum(salary) as total\_salary from test;

select min(salary) from test;

select max(salary) from test;

select avg(salary) from test;

# other built in functions

select name,length(name) from test;

select name,lower(name) from test;

select name,upper(name) from test;

select name,length(ltrim("India ")) from test;

select name,length(rtrim(" India ") )from test;

#date functions

select curdate();

select year(curdate());

select month(curdate());

select weekday(curdate());

select current\_time();

select hour(current\_time());

select minute(current\_time());

select second(current\_time());

select microsecond(current\_time());

select now(); #giving date and time both

#find average salary of each department with department name

select department,round(avg(salary),0) as average\_salary from test group by department;

select department,round(avg(salary),0) as average\_salary

from test

group by department # group by use to make group of similar things

order by department; # order by use to sort

select department, max(salary) from test group by department;

#for to use agrgate functions in conditions

# "having" clause used

select department, max(salary) from test

group by department

having avg(salary)>30000;

select department, min(salary) from test

group by department

order by department desc; # sorting in descending order

select name,salary from test where name="zain" or name="ahsan" or

name="Awon";

#for multiple "OR" like above, we use "in" operator

select name,salary from test where name in ("zain","Ahsan");

select name ,department from test where

age between 21 and 30;

select name from test where name like('a%');

select name from test where name like('%n'); # ends with n

select name from test where name like('%a%');

select name from test where name like('\_a%');

select name from test where name like('\_w%');

select st.address ,ts.name ,ts.department

from test as ts inner join

student as st

where ts.id=st.id;

select \*

from test as ts left join

student as st

on ts.id=st.id;

select \*

from test as ts right join

student as st

on ts.id=st.id;

#to find union of two tables we use "union"

select \* from test as ts left join student as st on ts.id=st.id

union

select \* from test as ts right join student as st on ts.id=st.id;

# print epmloyes name who is taking highest salary in his department

select name,department,salary from test where salary=( select max(salary) from test);

select name,department,salary from test

where salary=(

select max(salary) from test

group by department);

update test set salary=salary \* 0.35

where age in (select age from test

where age>21);

# print name of employ who is taking highest salary in his department

select name,department,max(salary) from test

group by department;

select t1.name,t1.department from test as t1, test as t2

where t1.name=t2.name

and t1.department!=t2.department;

update test set salary=salary\*0.35

where age in (select age

from test as t2 where t2.age>21);

select \* from test;

delete from test where id=102;

#deleting data inside the table

truncate table test;

create table customers(

id int,

age int

);

# before insert trigger

delimiter //

create trigger age\_verify

before insert

on customers

for each row

if new.age<0 then set new.age=0;

end if //

insert into customers values

(1,20),

(2,-30) ;

select \* from customers;

# after trigger

create table customer2(

id int, age int);

create table mesg(

id int,

mesg varchar(30)

);

delimeter //

create trigger check\_null\_dob

after insert

on customer2

for each row

begin

if new.dob is null then

insert into mesg values

(new.id,"hi please update date");

end if ;

end //

insert into customer2 values

(1,90),

(2,null),

(3,30),(4,null);

select \* from mesg;

# update trigger

delimiter //

create trigger update\_salary

before update

on test

for each row

if new.salary=20000 then set new.salary=26000;

elseif new.salary>26000 then set new.salary=30000;

end if; //

update test set salary=28000 where id=103;

select \* from test;

# delete trigger

create table salary(id int,salary int);

delimiter $$

create trigger after\_del

before delete

on test

for each row

begin

insert into salary values(old.id,old.salary);

end$$

delimiter ;

delete from test where id=110;

select \* from salarysalary;

#sub quieries practice

# print name of employs with salary who are taking greater salary then avg

select\* from test;

select avg(salary) from test;

select name,salary from test

where salary>(select avg(salary) from test);

#print name of employs taking more salary then umer

select name,salary,department from test where

salary>(select salary from test where name="Umer");

select name,salary,department from test where name!="Sami" and

salary>(select max(salary) from test where name="Sami");

# stored procedures (like we are making functions)

create table players(

id int,

name varchar(30),

goals int

);

insert into players values

(105,"Afzaal",5),

(106,"Umer",6),

(107,"Zaid",18),

(108,"Saad",8);

select \* from players;

delimiter &&

create procedure top\_player\_dynamicaly()

begin

select id,name, goals from players

where goals>6;

end &&

delimiter ;

call top\_player();

delimiter &&

create procedure top\_player\_dynamicaly(in var int)

begin

select id,name, goals from players

where goals>var;

end &&

delimiter ;

call top\_player\_dynamicaly(7);

#parametrize procedure (like parametrize fucntion)

delimiter //

create procedure sort\_by\_salary\_ascending(in var int)

begin

select \* from test

order by salary limit var;

end //

delimiter ;

# want to know only three records from table.

call sort\_by\_salary(3);

call sort\_by\_salary\_ascending(5);

#procedure to update salary

delimiter //

create procedure update\_salary(in temp\_id int,

in new\_salary int)

begin

update test set salary =new\_salary

where id=temp\_id;

end; //

delimiter ;

# to delete procedure

#drop procedure update\_salary;

# to remove safe update mode error

#set sql\_safe\_updates=1;

call update\_salary(101 ,280000);

select \* from test ;

# views (use to store data virtualy.)

create view customers\_viewcustomers\_view as

select name,salary, department from test2;

create view top\_players as

select name ,goals from players where goals>5 order by goals desc;

select \* from top\_players;

**Interview Questions practice**

#find max salary in every department

select department, max(salary) from test group by department;

#find unique departments

select distinct department from test;

#write sql query to fetch unique departments and print there length

select distinct department, length(department) from test;

#use if datediff. it is use to find differnce between two dates or time

select now();

select now(),datediff(current\_date(),'2022-04-14') as diff;

# write sql query to disply departments that having more then 2 employes.

select department ,count(id)from test group by department having count(id)>2;

# display detail of employes except IT department

select \* from test where department!="IT";

# write and sql query to print details if employes who joined before april 2010 and after may 2005

select \* from test2 where doj>'2005-3-3' and doj<'2009-1-1';

# find the employ having third highest salary from table

select \* from (select \* from test order by salary desc limit 3) as t

order by salary limit 1;

# above query explaination: inner query will return top 3 employes. then we order them ascending order and get only one record

# print all the alternate reocrds in table or printing ODD records in table

select \* from test where id%2=1;

#print Even records in table

select \* from test where id%2=0;

# write sql query to fetch all duplicate rows of table

select \* , count(id) as repeated from test3 group by id having count(name)>1;

#other version of above

select \* from test3 group by id having count(id)>1

and count(name)>1 and count(age)>1;

#display name of employes with exactly 2 A's in his name

select \* from test where length(name)-length(replace(upper(name),'A',''))=2;

#display name of employes with greater 2 A's in his name

select \* from test where length(name)-length(replace(upper(name),'A',''))>=2;

#from a given string how you will get 4 characters starting from 2nd position

select \* from test where substring(name,2,4)="ain";

# wite sql query to print one row twice in results from a table

select name, department from test where department="IT"

union all

select name,department from test as e1 where e1.department="IT";

# write sql query to add 100 where number is 0, 20 when number is 1. else print number

select \* from test3;

select id, case when id=101 then id+10

when id=102 then id+20

else id end as added

from test3;

# find sum of all positive and negative values

select sum(case when id>0 then id else 0 end) as sum\_positive,

sum(case when id<0 then id else 0 end ) as sum\_negative from test3;

# write a query to fetch values in table that are present in B but not present in table A

select id from test left join test3 using (id) where test3.id is null;

# find customers who don't have record in orders table (here assuming tesst3 as order table)

select \* from test where id not in (select id from test3);

#find all month end orders (not proper answer as eomonth not in mysql but in sql)

select \* from test2 where dayofmonth(doj)=31 or dayofmonth(doj)=30;

# display total number of products in a category with category name

select c.name, count(\*) from products as p inner join category as c

on p.category\_id=c.id group by c.id

order by count(\*) desc;

# find late orders deliverd list

select \* from orders where delivery\_Date>required\_date;