show databases;

create database batch11\_1;

use batch11\_1;

drop database batch11\_1;

create table student(id int ,name varchar(20),age int ,

address varchar(20),marks float,dob date);

desc student;

insert into student values(1,"harry",20,"mumbai",70.89,"1999-08-09");

insert into student values(2,"jack",25,"pune",78.89,"1991-08-09");

insert into student values(3,"sam",22,"thane",80.89,"1995-08-09");

insert into student values(4,"Rohan",24,"vashi",90.89,"1998-08-09");

insert into student values(5,"raj",25,"dehli",60.89,"1999-08-09");

select \* from student;

/\* ddl commands

alter--

add new column

change column name

chnage table name

change datatypes

drop colun

\*/

/\* Add column to the existing table-

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

=> alter table tablename

add colname1 datatype(size) ,

.

.

.add colnameN datatype(size) ;

\*/

alter table student add contactno long;

desc student;

select \* from student;

/\*Add column to the existing table at first position-

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

=>alter table tablename

add colname datatype(size) first;\*/

alter table student add column roll\_no int first;

/\* chnage column name

=>alter table tablename change oldcolumnname newcolumnname dataype;\*/

alter table student change address location varchar(20);

/\* rename table name

alte table oldtablename rename to newtablename;\*/

alter table student rename to stud;

select \* from stud;

desc stud;

/\* modify--change the datatype of your column

alter table table\_name modify columnname newdatatype(newsize)\*/

alter table stud modify age float;

alter table stud modify location varchar(50);

drop database batch11\_1;

drop table stud;

alter table stud drop column roll\_no;

delete from stud where id=5;

truncate table stud;

/\* constraints--- set of rules defines at the time of table creation

1.primary key--not accept duplicate data and null value,only on PK

2.unquie---not allow duplicate value but allow null

3.not null--not allow null value but duplicate

4.default---set any value bydefault

5.auto\_increment--that column value auto matically incremented

6.check---check condition and accordingly value will be added

7.foreign key---used to create relation ship multiple tables\*/

create table employee(

id int primary key auto\_increment,

name varchar(20) unique,

salary int not null,

comp\_name varchar(20) default "itvedant",

age int check(age>18)

);

desc employee;

insert into employee values(1,"Ram",434345,"IBM",20);

insert into employee values(2,"sam",65665,"TCS",30);

insert into employee values(3,"soham",45665,"TCS",35);

insert into employee values(4,"jack",55665,"Wipro",24);

insert into employee values(null,"harry",95665,"infosys",27);

select \* from employee;

/\*DQL---data query lang

retrive data

filter data according to condition

ordering data(asc/desc)

select query-----

all field retrive\*/

select \* from employee;

/\* retrive perticular column\*/

select name,salary from employee;

/\*arithmetic operators ----- +,-,\*,/,% -----\*/

select name,salary,salary+5000 as result from employee;

select name,salary,salary/5 as result from employee;

/\*comparision operator----->,<.>=,<=,==\*/

select \* from employee where age>25;

select \* from employee where id<4;

select name,salary ,comp\_name from employee where age>=30;

/\* logical operators---- and,or

and----if all conditions are true then it will return true values

or---if any one value will be true then its true\*/

select \* from employee where salary>50000 and age<25;

select \* from employee where comp\_name="TCS" and age>30;

select \* from employee where salary>50000 or age<25;

select \* from employee where comp\_name="TCS" or age>30;

select \* from employee where comp\_name!="TCS";

/\*range operator--between clause\*/

select \* from employee where id between 2 and 5;

/\*list operator----in,not in\*/

select \* from employee where name="Ram" or name="soham"or name="harry";

select \* from employee where name in("Ram","soham","harry");

select \* from employee where name not in("Ram","soham","harry");

/\* distinct clause----it retrives only unique records--\*/

select distinct comp\_name from employee;

/\* order by clause -----sort data according to order(asc/desc)

by default order is asc

\*/

select \* from employee order by age ;

select \* from employee order by age asc ;

select \* from employee order by salary desc ;

select name ,comp\_name from employee order by comp\_name desc;

/\*retriving 1st 4 recorsd in table\*/

select \* from employee limit 4;

/\*retriving last 4 record \*/

select \* from employee order by id desc limit 4;

/\* null and not null values\*/

select \* from employee where salary=null;

select \* from employee where salary is not null;

/\* like keyword

pattern- 'a%' - which starts with a

pattern- '%a' - which ends with a

pattern- '%a%' - which contains a

pattern - '\_a%' - second character is a

pattern - '%a\_' - second last character is a

pattern - '%a\_\_' - third last character is a

\*/

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

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

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

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

select \* from employee where name like '%a\_';

select \* from employee where name like '%h\_\_';

/\* inbuilt functions in sql\*/

/\* string funcion---

1)concat()---join tow or more string\*/

select concat("hello","\_","world")as msg;

select id,name,concat(id,"\_",name)as result from employee ;

/\*2)lower()---disp data in lower case\*/

select lower(comp\_name)as result from employee;

/\*3)upper()---disp data in upper case\*/

select upper(comp\_name)as result from employee;

/\*4) replace()--('string','oldstring','newstring')\*/

select replace('good morning','morning','afternnon')as result;

select comp\_name,replace(comp\_name,'IBM','IBM1')as result from employee;

/\*substr---syntax substr(m,n)

m--starting positio ,n--no of char extracted\*/

select substr('itvedant',3,6)as result;

/\*length---count number of charcter in string\*/

select name,length(name)as result from employee;

/\* math functions---

abs()---absoulte value\*/

select abs(10) as result;

select abs(-10) as result;

/\* ceil()--maximum value in that range\*/

select ceil(10.4) as result;

select ceil(10.9) as result;

select ceil(10.5) as result;

/\*floor ()--minimum value in that range\*/

select floor(10.4) as result;

select floor(10.9) as result;

select floor(10.5) as result;

select floor(10) as result;

/\*round---based on decimal points\*/

select round(10.4) as result;

select round(10.9) as result;

select round(10.5) as result;

/\* mod()---genrate reminder value\*/

select mod(10,4) as result;

/\*pow()---genrate power\*/

select pow(4,2)as result;

/\*sqrt()---genrate sqrt value\*/

select sqrt(64)as result;

/\* date and time function

date format--'yyyy-mm-dd' '2023-09-08'

curdate()--gives current system date

curtime()--gives current sys time

now(),sysdate()\*/

select curdate()as result;

select curtime()as result;

select now()as result;

select sysdate()as result;

select year('2022-08-09')as result;

select year(now()) as result;

select month('2022-08-09')as result;

select month(now()) as result;

select day('2022-08-09')as result;

select day(now()) as result;

select hour(now()) as result;

select datediff('2095-09-09',now())as result;

select date\_format(now(),'%D %M %Y')as result;

select date\_format(now(),'%d %m %y')as result;

select \* from employee;

/\* aggrigate function(group by caluse)---count,min,max,avg,sum,\*/

select comp\_name,count(comp\_name)as result from employee group by comp\_name;

select comp\_name,min(salary)as result from employee group by comp\_name;

select comp\_name,max(salary)as result from employee group by comp\_name;

select comp\_name,avg(salary)as result from employee group by comp\_name;

select comp\_name,sum(salary)as result from employee group by comp\_name;

/\*having caluse----\*/

select comp\_name,count(comp\_name)as result from employee

group by comp\_name having count(comp\_name)>1;

/\*foregin key---it used to make relationships between two or more tables

fk also pk of another table

we can add duplicate value in fk

we can add null into fk

one table will having multiple fk\*/

create table department

(did int primary key,

dname varchar(100),

dlocation varchar(100),

id int,

foreign key(id)references employee(id));

desc department;

insert into department values(1000,"HR","mumbai",2);

insert into department values(2000,"Admin","thane",1);

insert into department values(3000,"Production","vashi",1);

insert into department values(4000,"IT","pune",null);

insert into department values(15000,"civil","deli",4);

select \* from department;

create table product(

pid int primary key,

pname varchar(20),

pquntity int,

id int,

did int,

foreign key(id)references employee(id),

foreign key(did)references department(did));

desc product;

insert into product values(5001,"book",4,1,1000);

insert into product values(5002,"mobile",10,2,2000);

insert into product values(5003,"camera",40,null,3000);

insert into product values(5004,"pen",50,3,null);

insert into product values(5005,"dress",9,4,1000);

select \* from product;

Create table abc select \* from product limit 0;

/\* joins---joins are basically used to retrive the data from two or more tables

types---inner join,left outer,right outer join,cross join ,self join\*/

select \* from employee;

select \* from department;

select employee.id,name,salary,comp\_name,dname,dlocation

from employee inner join department

on employee.id=department.id;

/\*left outer join --it retrives all data from your left table and which is common data from right table i

if data not matches then by default it set the value null\*/

select employee.id,name,salary,comp\_name,dname,dlocation

from employee left outer join department

on employee.id=department.id;

/\*right outer join--it retrives all data from right table and which is common data

from left table if data not present then it set null value\*/

select employee.id,name,salary,comp\_name,did,dname,dlocation

from employee right outer join department

on employee.id=department.id;

/\* Q.write query to rerive salary,comp\_name,did,dname,dlocation from you tables and apply the condition

whose did >5000;\*/

select salary,comp\_name,did,dname,dlocation

from employee inner join department

on employee.id=department.id where did>5000;

/\* cross join---\*/

select \* from employee cross join department;

/\* self join\*/

select a.id,a.name,b.salary,b.comp\_name

from employee as a,employee as b

where a.id=b.id;

/\* joins with 3 tables---\*/

select employee.id,name,salary,comp\_name,dname,dlocation,pname,pquntity

from employee inner join department

on employee.id=department.id

inner join product

on department.id=product.id

;

select salary,comp\_name,did,dname,dlocation

from employee inner join department

on employee.id>department.id ;

create view v as select employee.id,name,salary,comp\_name,dname,dlocation,pname,pquntity

from employee inner join department

on employee.id=department.id

inner join product

on department.id=product.id

;

select \* from v;