**what is SQL ?**

SQL is stands for structured query language

SQL is used to create database | table structure

SQL is a case in-sensitive language

Ex: INSERT , insert

SQL is create some command or query to make database or table structured.

**Types of SQL query or command**

a) DDL (data definition language)

b) DML (data manipulation language)

c) DQL (data query language)

d) TCL (Transactional control language)

**DDL(data definition language) :** create a database and table structures

Ex: a) create

b) alter

c) rename

d) change

e) drop

f) truncate

**a) how to create a database**

**what is database ?**

database is where we stored something in form of tables or information i.e called database.

**syntax :** create database databasename;

examples : create database ecommerceapp;

**b) how to create a table**

**what is table ?**

tables are in form of columns and rows where we stored information’s of users

**syntax :**

create table tablename

(

Columnname datatype(size) primary key auto\_increment,

Columnname datatype(size)

);

**Examples :**

create table tbl\_user

(

userid int primary key AUTO\_INCREMENT,

username varchar(155),

password varchar(200),

photo varchar(200),

gender varchar(155),

hobby varchar(200),

phone bigint,

countryname varchar(255)

)

create table tbl\_contact

(

contactid int AUTO\_INCREMENT primary key,

name varchar(255),

fname varchar(255),

lname varchar(255),

mobile bigint,

subject varchar(255),

message text

)

**b) Alter :** alter is used to add, modify , change a column name after create a table structured.

1) alter table tbl\_contact change fname firstname varchar(255);

2) alter table tbl\_user add statename varchar(255);

3) alter table tbl\_user add cityname varchar(255);

4) alter table tbl\_user add email varchar(255) after username;

**c) change :** change is used to change any column name after create any tables.

1) alter table tbl\_contact change fname firstname varchar(255);

**d) rename :** rename is used to rename a table after create a table structured.

Examples : rename table tbl\_user to users;

rename table tbl\_country to country;

rename table tbl\_contact to contact;

**e) drop :** drop is used to drop database structured and tables structured

a) drop database databasename;

ex: drop database ecommerceapp;

b) drop table tablename;

ex: drop table country;

**f) truncate :** truncate table tablename;

truncate is used to delete data from tables after truncate we never rollback our data from tables.

Ex: truncate table contact;

**2) DML: data manipulation language:**

DML stands for data manipulation language it means how to store data, delete data and update data.

**a) insert**

**b) delete**

**c) update**

**a)** insert: how to insert data in table there we used insert

syntax : insert into tablename(columnname) values(‘value’);

examples : insert into contact(name,firstname,lname,mobile,subject,message) values('brijesh','brijesh','pandya','9311212121','customer care contact','hi i am brijesh');

or

multiple data insert

insert into contact(name,firstname,lname,mobile,subject,message) values('yash','yash','pandya','9311212121','customer care contact','hi i am yash'),('akshay','akshay','pandya','9311212121','customer care contact','hi i am akshay');

or

insert into contact values('null','badi','badi','badi','9311212121','customer care contact','hi i am badi'),('null','mukesh','mukesh','patel','9311212121','customer care contact','hi i am patel');

**b) delete :** delete is used to delete all data or rows | particulars data | alternate data

1) all data delete

Examples : delete from tablename;

Delete from contact;

2) particular one data deleted

Examples : delete from tablename where columnname=’name’;

delete from tablename where id=’id’;

delete from contact where name='mukesh';

delete from contact where contactid='3';

**3) alternate data delete**

delete from contact where contactid in (1,4);

4) set range to delete data

Delete from contact where contactid between 5 and 10;

**c) update :** update is used to update data from tables

update tablename set columnname=’values’ where id=’id’;

examples : update users set username='akshay',email='akshay@gmail.com',password='ak545454',photo='akshay.png',hobby='teaching',phone='634565651',cityname='porbandar' where userid=1;

**c) DQL :** dql stands for data query language

syntax : select

a) select all table data

select \* from contact;

b) particular data select

select \* from contact where contactid=1;

b) particular data select with name

select \* from contact where name=’brijesh’;

c) alternate data select

SELECT \* from contact WHERE contactid in (1,3,5);

d) range data select using between

SELECT \* from contact WHERE contactid between 1 and 5;

e) range data select using range

SELECT \* from contact where contactid LIMIT 0,6;

e) select data using columns

select contactid,name,mobile from contact;

e) select data using columns using where

select contactid,name,mobile from contact where contactid=1;

select contactid,name,mobile from contact where contactid BETWEEN 1 and 2;

select contactid,name,mobile from contact where contactid limit (0,3);

select contactid,name,mobile from contact where contactid in (1,2);

**SQL function : sql function are two types**

**a) aggrigate function**

1) max()

Ex: select max(salary) as max\_salary from employee;

2) min()

Ex: select min(salary) as min\_salary from employee;

3) count()

Ex: select count(empid) as Total\_Numberof\_Employee from employee;

4) avg()

Ex: select avg(salary) as Average\_salary from employee;

5) sum()

Ex: select sum(salary) as Sumof\_Salary from employee;

**b) scalar function**

1) first() : note mysql is no more support first

Ex: select first(name) from employee;

2) last() : note mysql is no more support first

Ex: select last(name) from employee;

3) ucase()

Ex: select ucase(name) from employee;

4) lcase()

Ex: select lcase(name) from employee;

**4) TCL (transactional control language)**

a) commit

b) rollback

**a) commit** : commit is used to save query after delete any data

**examples :**

start TRANSACTION;

delete from contact where name='neha';

commit;

b) rollback : rollback is used to rollback data after delete

**examples :**

start TRANSACTION;

select \* from contact where name='neha';

rollback;

Note : mysql is not support rollback structure oracle will support rollback.

**SQL view :** view is used to create a duplicate tables or clone of table **.**

**Examples :**

CREATE VIEW view\_name AS SELECT column1, column2 FROM table\_name  
WHERE condition;

create view emp\_view as select empid,name,email,phone from employee where empid in (1,3,5);

we can manage data from view

a) we create view

b) we delete data from view

c) we update data from view

examples : update emp\_view set name='tushar',email='tushar@gmail.com',phone='6356584545' where empid=3;

examples : delete from emp\_view where empid=3;

SQL index : we create sql index to perform speed of sql tables or it is optimized speed of sql tables.

Or

Sql index is used to optimized speed and fast data search from tables

Index does not show result it is manually set and optimized speed of tables;

Examples :

CREATE INDEX index\_name ON table\_name (column1, column2, ...);

Ex: create index employee\_index on employee(empid,name,email,phone);

**SQL key constraints :** key constraints is used to set limits on your tables

Types of key contstraints in SQL

a) primary key

b) foreign key

c) unique key

**a) primary key :** A pk is used to provides only one time in a tables.

A pk is never return null values.

A pk is always auto\_increments

A pk always stored a unique values.

create table tbl\_feedback

(

fid int primary key AUTO\_INCREMENT,

name varchar(155),

email varchar(255),

phone bigint,

comment text

)

**b) unique key : A**  unique key is used to provides multiples column in a tables

A uk is uniquely identified users.

A uk is stored one times of null values and its never stored dublicate values.

ALTER TABLE `tbl\_feedback` ADD UNIQUE(`email`);

**c) Foreign key :** A fk is provides more than two columns.

A fk provides relationship between one tables to another tables.

A fk is provides relationship via common field with one tables to another tables.

**SQL joins :**

**Types of joins**

**a) inner join**

**b) join**

**c) outer join**

**a) left join**

**b) right join**

**c) full join**

**d) cross join**