**what is SQL ?**

SQL stands for structured query language

SQL is a language

SQl is a case insensitive language

Ex: INSERT , insert , Insert

SQL is used to create database | table structured

SQL is used to create database | table | delete | update information’s in database.

SQL types of command ?

1. DDL
2. DML
3. DQL
4. TCL
5. DDL : stands for data definition language

Ex: create | alter | rename | truncate | drop |change.

1. DML : stands for data manipulation language

Ex: insert | update | delete

1. DQL: stands for data query language

Ex: select

1. TCL: stands for transactional control language

Ex: commit | rollback

**DDL : a) create a database**

**Syntax :** create database databasename;

Ex: create database flipkart;

<http://localhost/phpmyadmin/>

create database flipkart\_db;

**b) create a table**

**Syntax :**

Create table tablename

(

Id int(default size) auto\_increment primary key,

Name varchar(255),

Email varchar(255),

Password varchar(255),

gender varchar(255),

hobby varchar(255),

mobile bigint,

message text

);

Ex: create TABLE users

(

userid int AUTO\_INCREMENT PRIMARY key,

name varchar(255),

password varchar(255),

firstname varchar(255),

lastname varchar(255),

gender varchar(255),

hobby varchar(255),

mobile bigint,

address text

);

create TABLE category

(

category\_id int AUTO\_INCREMENT PRIMARY key,

name varchar(255),

added\_date varchar(255)

);

create TABLE subcategory

(

subcategory\_id int AUTO\_INCREMENT PRIMARY key,

subcategoryname varchar(255),

added\_date varchar(255)

);

create TABLE products

(

product\_id int AUTO\_INCREMENT PRIMARY key,

category varchar(255),

subcategory varchar(255),

pname varchar(255),

pimage varchar(255),

price float,

qty int,

descriptions text

);

Alter : alter is used to add | modify | change column name is tables .

Ex: alter table products CHANGE descriptions product\_decriptions text;

Ex: alter table products add added\_date varchar(255);

Ex: alter table products add offer\_price float after price;

Change :

alter table products CHANGE descriptions product\_decriptions text;

Rename : rename is used to rename the table name

rename table category to flip\_category;

rename table products to flip\_products;

**truncate** : truncate is used to remove or empty all tables data after truncate we never rollback data

truncate table tablename;

ex: truncate table flip\_category;

**drop :** drop is used to drop database and table structure and data both after drop we can not rollback data

**drop a database**

drop database databasename ;

ex: drop database flipkart\_db;

**drop a table**

drop table tablename;

ex: drop table flip\_category;

drop table flip\_products;

<http://localhost/phpmyadmin/>

create database flipkart\_db;

**DML :** dml stands for data manipulation language

Query : a) insert

b) delete

c) update

how to insert data :

1. Insert :

Ex: insert into flip\_category (name,added\_date) values ('mobile','09/01/2023');

Or insert multiple data

insert into flip\_category (name,added\_date) values ('electronics','09/01/2023'),('furnitures','09/01/2023'),('kitchen ware','09/01/2023');

or another way to insert query without taking columns

insert into flip\_category values ('null','kids items','09/01/2023'),('null','Mens items','09/01/2023'),('null','womens items','09/01/2023');

1. Delete : delete is used to delete all tables data | or particular data | or alternate data

1. Delete from flip\_category; deleted all data from tables
2. Delete from flip\_category where category\_id=1;
3. Delete from flip\_category where name=’mobile’;
4. Delete from flip\_category where name in ('mens items','electronics','kitchen ware');
5. Delete from flip\_category where category\_id BETWEEN 1 and 100;
6. Update :

Update tablename set columnname=’value’ where id=’id’;

update flip\_category set name='electronics',added\_date='07/01/2023' where category\_id=3;

DQL : dql stands for data query language

Command : select

1. select \* from flip\_category
2. select \* from flip\_category where category\_id=3;
3. select \* from flip\_category where name=’mens items’;
4. select category\_id,name from flip\_category;
5. select \* from flip\_category order by name asc;
6. select \* from flip\_category order by name;
7. select \* from flip\_category order by name desc;

Like operartor : searching an name with keyword or alphabetic character

1. select \* from flip\_category where name like 'e%';
2. select \* from flip\_category where name like '%e%';
3. select \* from flip\_category where name like '%e';

**SQL function :**

**A function is a group of code or set of instructions i.e used to complte any task.**

**Or**

**A function is block of code or group of code i.e used to copleted any task .**

**Typs of SQL function :**

1. **aggrigate function**
2. **min()**
3. **max()**
4. **avg()**
5. **count()**
6. **Sum()**

**B) scalar function**

**a) first()**

**b) last ()**

**c) ucase()**

**d) lcase()**

1. Max () : find a max values

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

1. min () : find a min values

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

Find second highest salary

Find subquery in SQL : query within another query

select max(salary) from flip\_employee where salary < (SELECT max(salary) from flip\_employee);

1. select avg(salary) as AVG\_SALARY from flip\_employee;
2. select sum(salary) as SUM\_SALARY from flip\_employee;
3. select count(emplid) as Total\_number\_employee from flip\_employee;

b) scalar :

1. first : get first data from table

select first(empname) from flip\_employee;

1. last : get last data from table

select last(empname) from flip\_employee;

1. ucase() : get uppercase data from table

select ucase(empname) from flip\_employee;

1. lcase() : get lowercase data from table

select lcase(empname) from flip\_employee;

TCL : transaction control language

1. commit () :

commit () : commit is used to save any data after delete

START TRANSACTION;

delete from flip\_employee where emplid=1;

commit;

1. rollback() :

Note : mysql structured does not support

START TRANSACTION;

Select \* from flip\_employee where emplid=1;

rollback;

SQL constraints : sql constraints provides a limit on tables

Types of SQL constraints :

1. primary key
2. unique key
3. foreign key
4. compound key

primary key : A PK is used to assign only once time in a table

A PK never returns null value

A PK is always auto\_increments

Country

Country\_id(pk) country\_name

1 india

unique key : A UK is used to assign more than once time in a table

A UK at least once time return null value.

A UK never returns a dublicate values

users

user\_id(pk) name email(uk) phone

1 ABC [abc@gmail.com](mailto:abc@gmail.com) 9173357215

ALTER TABLE `users` ADD UNIQUE(`email`);

Foreign key : A FK is used to provide relationship between one table to another table

A FK provides relationship with its common field

A FK provides more than one times in a tables

Tbl\_country

country\_id (PK) countryname

1. india
2. usa
3. pakistan

Tbl\_state

state\_id (PK) statename country\_id (FK)

1. gujrat 1
2. california 2
3. karachi 3
4. uP 1
5. MP 1
6. Lahore 3

Tbl\_city

city\_id (PK) cityname country\_id (FK) state\_id(fk)

rajkot 1 1

1. navada 2 2
2. pkcity 3 3

create table tbl\_country

(

country\_id int PRIMARY key AUTO\_INCREMENT,

country\_name varchar(255)

)

Create a foreign key

create table tbl\_state

(

state\_id int PRIMARY key AUTO\_INCREMENT,

country\_id int REFERENCES tbl\_country(country\_id),

state\_name varchar(255)

)

create table tbl\_city

(

city\_id int PRIMARY key AUTO\_INCREMENT,

country\_id int REFERENCES tbl\_country(country\_id),

state\_id int REFERENCES tbl\_state(state\_id),

city\_name varchar(255)

)

**SQL joins : joins are used to join more than one tables if data matched one table with another tables.**

Types of join :

1. inner join
2. join
3. outer join
4. left join
5. right join
6. full join
7. cross join

Inner join : inner join are used to join more than one tables if data matched fromt first table with second table join all data.

Syntax : select firsttablename.\*,columnname from firsttablename inner join secondtablename on firsttable.commonfield=secondtable.commonfield;

Ex: select tbl\_state.\*,country\_name from tbl\_state inner join tbl\_country on tbl\_state.country\_id=tbl\_country.country\_id;

Or

Select tbl\_city.\*,country\_name,state\_name from tbl\_city inner join tbl\_country on tbl\_city.country\_id=tbl\_country.country\_id inner join tbl\_state on tbl\_city.state\_id=tbl\_state.state\_id;

Or

Select city\_id,city\_name,country\_name,state\_name from tbl\_city inner join tbl\_country on tbl\_city.country\_id=tbl\_country.country\_id inner join tbl\_state on tbl\_city.state\_id=tbl\_state.state\_id;

1. Join : join are used to join more than one tables if data matched from first table with second table if data are matched join all data.

Syntax : select firsttablename.\*,columnname from firsttablename join secondtablename on firsttable.commonfield=secondtable.commonfield;

Ex: select tbl\_state.\*,country\_name from tbl\_state join tbl\_country on tbl\_state.country\_id=tbl\_country.country\_id;

Or

Select tbl\_city.\*,country\_name,state\_name from tbl\_city join tbl\_country on tbl\_city.country\_id=tbl\_country.country\_id join tbl\_state on tbl\_city.state\_id=tbl\_state.state\_id;

Or

Select city\_id,city\_name,country\_name,state\_name from tbl\_city join tbl\_country on tbl\_city.country\_id=tbl\_country.country\_id join tbl\_state on tbl\_city.state\_id=tbl\_state.state\_id;

1. Outer join :
2. Left join : left join are used to join more than one table if data matched from first table of left rows with second table of left rows join all data otherwise return null value.

Syntax : select firsttablename.\*,columnname from firsttablename left join secondtablename on firsttable.commonfield=secondtable.commonfield;

Ex: select tbl\_state.\*,country\_name from tbl\_state left join tbl\_country on tbl\_state.country\_id=tbl\_country.country\_id;

Or

Select tbl\_city.\*,country\_name,state\_name from tbl\_city left join tbl\_country on tbl\_city.country\_id=tbl\_country.country\_id left join tbl\_state on tbl\_city.state\_id=tbl\_state.state\_id;

Or

Select city\_id,city\_name,country\_name,state\_name from tbl\_city left join tbl\_country on tbl\_city.country\_id=tbl\_country.country\_id left join tbl\_state on tbl\_city.state\_id=tbl\_state.state\_id;

b) right join : right join are used to join more than one table if data matched from second table of right rows with first table of right rows join all data otherwise return null value.

Syntax : select firsttablename.\*,columnname from firsttablename right join secondtablename on firsttable.commonfield=secondtable.commonfield;

Ex: select tbl\_state.\*,country\_name from tbl\_state right join tbl\_country on tbl\_state.country\_id=tbl\_country.country\_id;

Or

Select tbl\_city.\*,country\_name,state\_name from tbl\_city right join tbl\_country on tbl\_city.country\_id=tbl\_country.country\_id right join tbl\_state on tbl\_city.state\_id=tbl\_state.state\_id;

Or

Select city\_id,city\_name,country\_name,state\_name from tbl\_city right join tbl\_country on tbl\_city.country\_id=tbl\_country.country\_id right join tbl\_state on tbl\_city.state\_id=tbl\_state.state\_id;

c)Full join =left join + right join

**Note : fill join is not supported in mysql**

**d)cross join :** cross join are used either data matched or not matched return all cross or multiply of data.

select \* from tbl\_state cross join tbl\_country;

or

select \* from tbl\_city CROSS join tbl\_country cross join tbl\_state;

select tbl\_users.\*,country\_name,state\_name,city\_name from tbl\_users join tbl\_country on tbl\_users.country\_id=tbl\_country.country\_id join tbl\_state on tbl\_users.state\_id=tbl\_state.state\_id join tbl\_city on tbl\_users.city\_id=tbl\_city.city\_id;