**SQL – Structured Query Language**

**There are 4 types Types of sql which are:-**

1. **DDL (Data Definition Language)**

* **Used to Create Database & table structure.**
* **To Create DATABASE:-**

**Syntax - create database databasename;**

**Example – create database blood\_bank;**

* **Chart for Creating Table:-**

**Columnname datatype(size) keyconstraints**

**Id int(defaultsize(11)) primary key(auto\_increments)**

**Name,firstname, char,varchar(0-255) not null**

**lastname,email char,varchar(0-255) not null**

**Mobile bigint(defaultsize(20))**

**Date date**

**Datetime datetime**

**Message text**

**Photo blob,varchar(0-255)**

**Salary int,float**

* **To CREATE Table:**

**Syntax - create table tablename (**

**columnname datatype(size) pk auto\_increment,**

**.**

**columnname datatype(size)**

**);**

**Example 1 –**

**create table bbank\_donor(**

**donor\_id int PRIMARY KEY AUTO\_INCREMENT,**

**donor\_firstname varchar(55),**

**donor\_lastname varchar(55),**

**mobile bigint,**

**city varchar(15),**

**state varchar(20)**

**)**

**Example 2 –**

**create table bbank\_doctor(**

**doctor\_id int PRIMARY KEY AUTO\_INCREMENT,**

**doctor\_firstname varchar(10),**

**doctor\_lastname varchar(10),**

**city varchar (50),**

**apointment\_date date**

**)**

* **To ALTER table - used to add, update or modify a column after create table.**

**Example:-**

**alter table bbank\_donor add blood\_group int;**

**alter table bbank\_doctor add doctor\_degree int;**

**alter table bbank\_donor add adhar\_num bigint after donor\_lastname;**

**alter table bbank\_doctor add state varchar(15) after city;**

**alter table bbank\_donor change adhar\_num adharcard\_num bigint;**

**alter table bbank\_doctor change doctor\_degree doctor\_qualification bigint;**

* **To CHANGE table**

**Example:-**

**alter table bbank\_donor change adhar\_num adharcard\_num bigint;**

**alter table bbank\_doctor change doctor\_degree doctor\_qualification bigint;**

* **To RENAME table – used to rename tables.**

**Example:-**

**rename table bbank\_donor to bloodbank\_donor;**

**rename table bbank\_doctor to bloodbank\_doctor;**

* **To DROP table - used to delete database structure and table structures and its data also.**

**Syntax –drop table tablename;**

**Example:- drop table bloodbank\_doctor;**

**Syntax – drop database databasename;**

**Example:- drop database blood\_bank;**

* **To TRUNCATE table - used to delete or empty data from table but structure remain same.**

**Example:-**

**TRUNCATE table bloodbank\_name;**

1. **DML (Data Manipulation Language)**

* **Used to Manipulate data inside of tables.**
* **To INSERT table:-**

**INSERT INTO bloodbank\_doctor (doctor\_firstname)values('Ramesh');**

**INSERT INTO bloodbank\_doctor (doctor\_firstname)values('Suresh'),('Hitesh');**

**insert into bloodbank\_doctor values('null','Sagar'),('null','Surya');**

* **To UPDATE table:-**

**update bloodbank\_doctor SET doctor\_firstname="suryakumar" WHERE doctor\_id=7;**

**update bloodbank\_doctor SET doctor\_firstname='sagarSir' WHERE doctor\_firstname='sagar';**

* **To DELETE table:-**

**DELETE FROM bloodbank\_doctor WHERE doctor\_id='4';**

**DELETE FROM bloodbank\_doctor WHERE doctor\_firstname='sagarSir';**

**DELETE FROM bloodbank\_doctor WHERE doctor\_id in (8,9);**

**DELETE FROM bloodbank\_doctor WHERE doctor\_firstname in ('suryakumar');**

**DELETE from bloodbank\_servant;**

1. **DQL (Data Query Language)**

* **Used to select data from tables or fetch data from tables.**
* **To SELECT table:-**

**SELECT \* from bloodbank\_doctor;**

**SELECT servant\_name FROM bloodbank\_servant;**

**SELECT \* FROM bloodbank\_servant WHERE servant\_id=2;**

**SELECT \* FROM bloodbank\_servant WHERE servant\_name='terence';**

**SELECT \* FROM bloodbank\_servant WHERE servant\_name in ('umesh','terence');**

**SELECT \* FROM bloodbank\_servant WHERE servant\_id BETWEEN 2 AND 5;**

**SELECT \* FROM bloodbank\_servant WHERE servant\_name like '%h';**

**SELECT \* FROM bloodbank\_servant WHERE servant\_name like '%e%';**

**SELECT \* FROM bloodbank\_servant WHERE servant\_name like 'r%';**

**Now we will learn ORDER BY & GROUP BY:-**

* **ORDER BY - used to filter data from table in ASC or DESC order.**

**select \* FROM bloodbank\_servant order by servant\_name;**

**select \* FROM bloodbank\_servant order by servant\_name ASC;**

**select \* FROM bloodbank\_servant order by servant\_name DESC;**

* **GROUP BY- used to filter or select group of columns data.**

**SELECT sum(servant\_sallary),servant\_dept FROM bloodbank\_servant GROUP BY servant\_dept;**

**ALIAS - SELECT sum(servant\_sallary) as sumofservant\_salary,servant\_dept FROM bloodbank\_servant GROUP BY servant\_dept;**

1. **TCL (Transactional Query Language)**

* **Used to save data after delete or rollback data after delete.**
* **To COMMIT table:-**

**DELETE FROM bloodbank\_servant WHERE servant\_name='yashpal';**

**COMMIT;**

**START TRANSACTION;**

**DELETE FROM bloodbank\_servant WHERE servant\_name='terence';**

**commit;**

* **To ROLLBACK table:-**

**START TRANSACTION;**

**SELECT \* FROM bloodbank\_servant WHERE servant\_name='terence';**

**COMMIT;**

**Mysql structure does not support rollback query.**

* **SQL KEYCONSTRAINTS**

**There are 3 types of sql keyconstraints.**

1. **Primary Key (pk)**

**- never return a null value.**

**- only defined once time in a table and a pk key should alway auto\_increment. It stores a Unique value.**

**Bloodbank\_servant**

**Servant\_id(pk) servant\_name servant\_mobile servant\_adharcard servant\_dept servant\_sallary**

1. **Unique Key (uk)**

* **Returns at least on null value.**
* **is stored a null values but once time.**
* **it is provided to more than one columns.**
* **uk never store repeated data or duplicate data.**

**Example – alter table ‘bloodbank\_servant’ add unique(‘servant\_mobile’);**

1. **Foreign Key (fk)**

* **It is used to provides relation between tables.**
* **It is used in more than one times in a table.**
* **It is used to provide relationship with common field.**

**Example**

**servant\_country**

**Country\_id (pk) country\_name**

**1 India**

**2 uk**

**3 india**

**servant\_state**

**State\_id (pk) state\_name country\_id (fk)**

**1 gujarat 1**

**2 london 2**

**3 uttarpradesh 1**

**servant\_city**

**city\_id(pk) city\_name country\_id(fk) state\_id(fk)**

**1 rajkot 1 1**

**2 southwales 2 2**

**3 locknow 1 1**

**Query:-**

**create table servant\_city**

**(**

**state\_id int primary key AUTO\_INCREMENT,**

**state\_name varchar(255),**

**country\_id int REFERENCES servant\_country(country\_id)**

**)**

**create table servant\_city**

**(**

**city\_id int primary key AUTO\_INCREMENT,**

**city\_name varchar(255),**

**country\_id int REFERENCES servant\_country(country\_id),**

**state\_id int REFERENCES servant\_state(state\_id)**

**)**

**SQL JOINS:-**

* **There are 4 Types ofsql joins.**
* **Inner Join- inner join & join both are same and join more than one table with common field if data are matched join all data.**

**Syntax – select firsttable\_name .\*,column\_name from firsttable\_name inner join secondtable\_name on firsttablename.common\_field=secondtablename.common\_field**

**Example – select servant\_city from \*,country\_name from servant\_city inner join servant\_country on servant\_city.country\_id=servant\_country.country\_id;**

**Example1 - select servant\_city .\*,country\_name, state\_name from servant\_city inner join servant\_country on servant\_city.country\_id=servant\_country.country\_id inner join servant\_state on servant\_city.state\_id=servant\_state.state\_id;**

**(it shows data by countryid & name , stated&name)**

**Example2 - select city\_id,city\_name,country\_name,state\_name from servant\_city inner join servant\_country on servant\_city.country\_id=servant\_country.country\_id inner join servant\_state on servant\_city.state\_id=servant\_state.state\_id;**

**(it shows the data of city\_name state\_name & country\_name)**

* **Join – Join & inner join both are same .**

**Example - select city\_id,city\_name,country\_name,state\_name from servant\_city join servant\_country on servant\_city.country\_id=servant\_country.country\_id join servant\_state on servant\_city.state\_id=servant\_state.state\_id;**

* **Outer Join: – It is of 3 Types**
* **LEFT JOIN - used to join from firsttable of left column with second table of left column if data are match join all data otherwise return null value.**

**Example - select city\_id,city\_name,country\_name,state\_name from servant\_city left join servant\_country on servant\_city.country\_id=servant\_country.country\_id left join servant\_state on servant\_city.state\_id=servant\_state.state\_id;**

* **RIGHT JOIN - used to join from firsttable of right column with second table of right column if data are match join all data otherwise return null value.**

**Example -** **select city\_id,city\_name,country\_name,state\_name from servant\_city right join servant\_country on servant\_city.country\_id=servant\_country.country\_id right join servant\_state on servant\_city.state\_id=servant\_state.state\_id;**

**Example1 –**

* **FULL JOIN – It is made of Left join + right join = full join.**

**Mysql does not support the full join.**

* **Cross Join – It is used to multiply each of rows and return matched data.**

**Example – select \* from bloodbank\_user cross join bloodbank\_servant;**

**select \* from bloodbank\_user cross join servant\_country;**

**\*SQL FUNCTIONS\***

* **There are 2 Types of SQL functions.**

1. **Aggrigate function**
2. **MAX:-**

**Example - select max(servant\_sallary) from bloodbank\_servant;**

1. **MIN:-**

**Example - select min(servant\_sallary) from bloodbank\_servant;**

1. **AVG:-**

**Example - select avg(servant\_sallary) from bloodbank\_servant;**

1. **COUNT:-**

**Example - select count(servant\_id) from bloodbank\_servant;**

1. **SUM:-**

**Example - select sum(servant\_sallary) from bloodbank\_servant;**

**To Find 2nd Highest Sallary from Table.**

**select max(servant\_sallary) from bloodbank\_servant where servant\_sallary < (select max(servant\_sallary) from bloodbank\_servant);**

1. **Scalar function**
2. **FIRST:-**

**EXAMPLE - select first (servant\_name) FROM bloodbank\_servant;**

1. **LAST:-**

**EXAMPLE - select last (servant\_name) FROM bloodbank\_servant;**

1. **UCASE:-**

**EXAMPLE - select ucase (servant\_name) FROM bloodbank\_servant;**

1. **LCASE:-**

**EXAMPLE - select lcase (servant\_name) FROM bloodbank\_servant;**

**\*SQL VIEWS\***

**SQL VIEW - sql view create a duplicate table or virtual table of main tables.**

**There are 2 types of Views**

1. **Simple View:- when we create view on simple column we call it as simple view.**

**Example - create view View\_doctor as select doctor\_id from bloodbank\_doctor where doctor\_id=1;**

**Example1 - create view view\_donor as select donor\_firstname from bloodbank\_donor where donor\_firstname='ramesh';**

1. **Complex View:- when we create view on multiple column we call it as complex view.**

**Example - create view multiview\_servant as select servant\_id, servant\_name, servant\_mobile, servant\_sallary, servant\_dept from bloodbank\_servant;**

**Example1 - delete from multiview\_servant where servant\_name='terence';**

**\*SQL INDEXER\***

**Sql indexer – It does not provides any output but indexer is used to improved speed or efficiency of table.**

**Example - create index servant\_index on bloodbank\_servant(servant\_id,servant\_name,servant\_mobile,servant\_dept);**