**SQL (**Structured Query Language**)**

SQL (Structured Query Language) is used to perform operations on the records stored in the database, such as updating records, inserting records, deleting records, creating and modifying database tables, views, etc.

SQL COMMANDS

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature –

* **DDL (**DATA DEFINITION LANGUAGE**)**
  + **CREATE**

Creates a new table, a view of a table, or other object in the database.

* + **ALTER**

Modifies an existing database object, such as a table.

* + **DROP**

Deletes an entire table, a view of a table or other objects in the database.

* **DML (**DATA MANIPULATION LANGUAGE**)**
  + **SELECT**

Retrieves certain records from one or more tables.

* + **INSERT**

Creates a record.

* + **UPDATE**

Modifies records.

* + **DELETE**

Deletes records.

* **DCL (**DATA CONTROL LANGUAGE**)**
* **GRANT**

Gives a privilege to user.

* **REVOKE**

Takes back privileges granted from user.

**EXAMPLES**

**DATABASE:**

A **database** is an organized collection of data, so that it can be easily accessed and managed.

CREATE DATABASE **TRAINING**;

**TABLES:**

* SQL Table is a collection of data which is organized in terms of rows and columns. In DBMS, the table is known as relation and row as a tuple.

**STUDENT**

CREATE TABLE STUDENT(

Regid INTEGER NOT NULL PRIMARY KEY,

NAME VARCHAR(20),

DOB DATE,

Deptid INT NOT NULL,

Address VARCHAR(80),

FOREIGN KEY(Deptid) REFERENCES DEPARTMENT(Deptid)

);

**DEPARTMENT**

CREATE TABLE DEPARTMENT(

Deptid INTEGER NOT NULL PRIMARY KEY,

Dept\_Name VARCHAR(20)

);

**LECTURERS**

CREATE TABLE LECTURERS(

Lid INT NOT NULL PRIMARY KEY,

LName VARCHAR(20),

Deptid INT NOT NULL,

FOREIGN KEY(Deptid) REFERENCES DEPARTMENT(Deptid)

);

**FEE**

CREATE TABLE FEE(

Fid INTEGER NOT NULL PRIMARY KEY,

Regid INT NOT NULL,

FOREIGN KEY(Regid) REFERENCES STUDENT(Regid),

Fee INT

);

**SBJ**

CREATE TABLE SBJ(

Sid INTEGER NOT NULL PRIMARY KEY,

SBJ\_Name VARCHAR(20),

Lid INT NOT NULL,

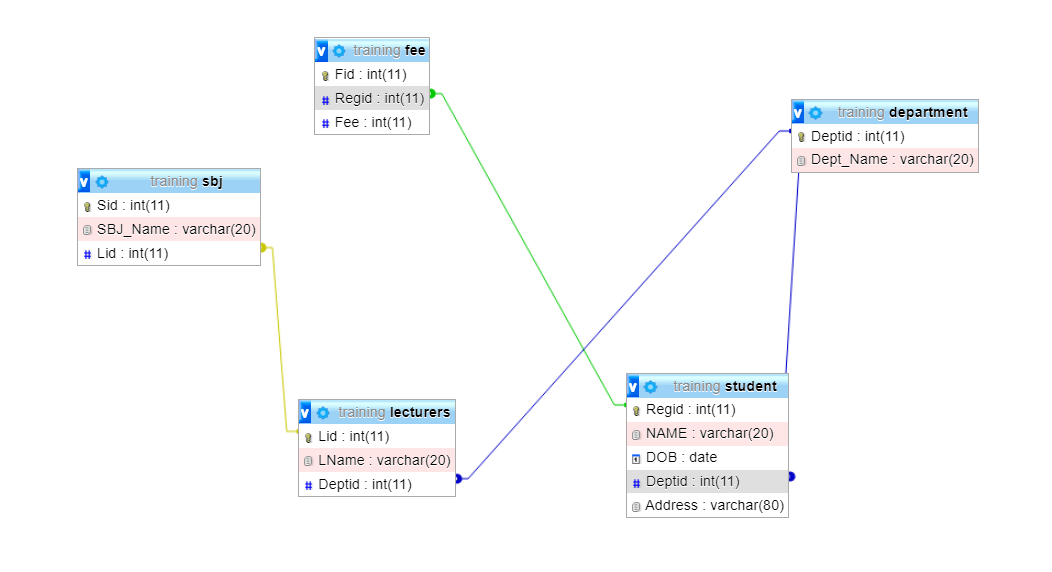
FOREIGN KEY(Lid) REFERENCES LECTURERS(Lid)

);

**SCHEMA**

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It’s the database designers who design the schema to help programmers understand the database and make it useful.

****

**STUDENT**

INSERT INTO STUDENT values(20,"Alex",'1996-12-08',1,"Idukki");

INSERT INTO STUDENT values(21,"Bibin",'1998-03-14',1,"Ernakulam");

INSERT INTO STUDENT values(22,"Anet",'1998-05-29',2,"Kottayam");

INSERT INTO STUDENT values(23,"Ajay",'1997-9-21',3,"Ernakulam");

INSERT INTO STUDENT values(24,"Alan",'1997-5-21',1,"Idukki");

INSERT INTO STUDENT values(25,"Adorn",'15-6-1997',3,"Alappuza");

INSERT INTO STUDENT values(26,"Don",'8-8-1991',1,"Idukki");

INSERT INTO STUDENT values(27,"Abhijith",'29-02-1995',1,"Kannoor");

INSERT INTO STUDENT values(28,"Jomin",'13-3-1998',2,"Trivandrum");

INSERT INTO STUDENT values(29,"Debin",'1-04-1999',1,"Kottayam");

INSERT INTO STUDENT values(30,"Sachin",'18-7-1993',3,"Idukki");

**DEPARTMENT**

INSERT INTO DEPARTMENT values(1,"SCIENCE");

INSERT INTO DEPARTMENT values(2,"COMMERCE");

INSERT INTO DEPARTMENT values(3,"HUMANITIES");

**FEE**

INSERT INTO FEE values(112,9000,20);

INSERT INTO FEE values(113,9000,21);

INSERT INTO FEE values(114,6500,22);

INSERT INTO FEE values(115,5000,23);

INSERT INTO FEE values(116,9500,24);

INSERT INTO FEE values(117,9000,25);

INSERT INTO FEE values(118,8000,26);

INSERT INTO FEE values(119,6500,27);

INSERT INTO FEE values(120,7000,28);

INSERT INTO FEE values(121,7500,29);

INSERT INTO FEE values(122,6000,30);

**LECTURERS**

INSERT INTO LECTURERS values(561,"Rajesh",1);

INSERT INTO LECTURERS values(562,"akhil",2);

INSERT INTO LECTURERS values(563,"Amal",3);

INSERT INTO LECTURERS values(564,"Bibin",1);

INSERT INTO LECTURERS values(565,"Alex",2);

INSERT INTO LECTURERS values(566,"Sumith",1);

**SBJ**

INSERT INTO SBJ values(144,"Maths",562);

INSERT INTO SBJ values(147,"Biology",561);

INSERT INTO SBJ values(148,"Accountancy",563);

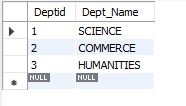
INSERT INTO SBJ values(149,"Econmics",564);

INSERT INTO SBJ values(150,"History",565);

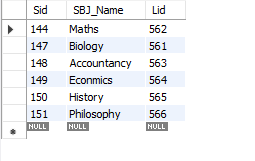
INSERT INTO SBJ values(151,"Philosophy",566);

**Output**

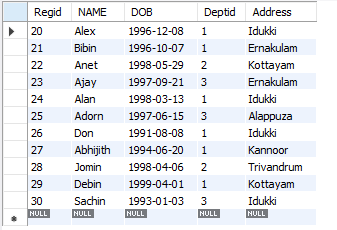
**DEPARTMENT**



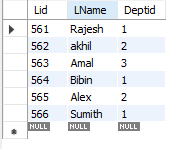
**SBJ**



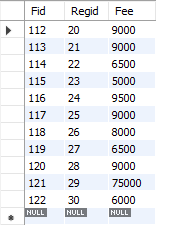
**STUDENT**



**LECTURERES**



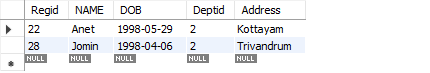
**FEE**



**WHERE**

**SELECT \* FROM student WHERE Deptid=2;**

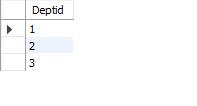
**Output**



**DISTINCT**

**SELECT DISTINCT Deptid FROM student;**

**Output**



**ALTER**

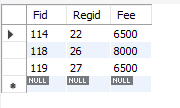
The ALTER TABLE statement in Structured Query Language allows you to add, modify, and delete columns of an existing table. This statement also allows database users to add and remove various SQL constraints on the existing tables.

**ALTER TABLE STUDENT CHANGE NAME NAME VARCHAR(45);**

**BETWEEN**

**SELECT \* FROM fee WHERE Fee BETWEEN 6500 AND 8000**

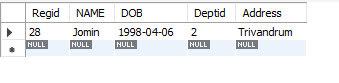
**Output**



**LIKE**

**SELECT \* FROM student WHERE NAME LIKE 'J%';**

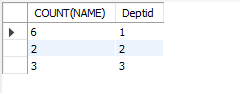
**Output**



**GROUP BY**

**SELECT COUNT(NAME), Deptid FROM student group by Deptid;**

**Output**



**HAVING**

**SELECT COUNT(NAME),Deptid FROM student group by Deptid having Deptid =3;**

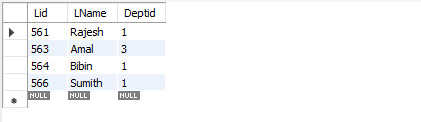
**Output**



**IN**

**SELECT \* FROM lecturers WHERE Deptid IN (1,3,5);**

**Output**



**TRUNCATE**

A truncate SQL statement is used to remove all rows (complete data) from a table. It is similar to the DELETE statement with no WHERE clause.

**TRUNCATE TABLE FEE;**

**Output**



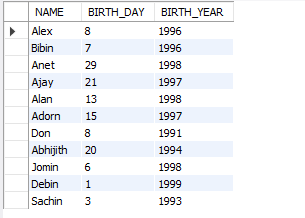
**DIFFERENCE BETWEEN UNIQUE AND DISTINCT**

The UNIQUE keyword in SQL plays the role of a database constraint; it ensures there are no duplicate values stored in a particular column or a set of columns. On the other hand, the DISTINCT keyword is used in the SELECT statement to fetch distinct rows from a table.

**DATE**

**SELECT NAME,extract(DAY from DOB)as BIRTH\_DAY,extract(YEAR from DOB)as BIRTH\_YEAR FROM student;**

**Output**



**JOINS**

 JOIN means **"to combine two or more tables”**. The SQL JOIN clause takes records from two or more tables in a database and combines it together.

Five types of JOINS:

1. inner join,
2. left outer join,
3. right outer join,
4. full outer join, and
5. cross join.

**LEFT JOIN**

**SELECT student.name,Department.Dept\_Name FROM student left join Department on Department.Deptid=student**

**RIGHT JOIN**

**SELECT student.name,Department.Dept\_Name FROM student right join Department on Department.Deptid=student**