**Data base**

1. Database is use to store a data permanently and can work with a large data.
2. To interact with Database you must know the SQL language.
3. To perform operations on the large data can be possible in a minimum time using a SQL.
4. Different type of DB
   1. **Relational Database**
      1. In this type the data will be store in a structure way such as table column format.
      2. There will be a multiple table present and they will be related with each other.
      3. Example: Mysql, Oracle, MSSql server, Postgres, DB2 etc…
   2. Document Database
      1. In this type o database you an store a JSON data.
      2. The values can be inserted and retrieve in the form of json only.
      3. This is not a structure database.
      4. Example: MongoDB
   3. Graph Database
      1. The data representation will be in the form of tree structure.
      2. In this you will get visualization of the data.
      3. Example: Neo4J

Database Download and installation

MySql:

<https://dev.mysql.com/downloads/installer/>



Install MySql:

<https://www.youtube.com/watch?v=OM4aZJW_Ojs>

**Important**

Note down the port number (3306), username (root) and password during the Setup.

****

**Structure Query Language (SQL)**

1. SQL is a query language which is used to interact with the database.
2. This language is a case insensitive.
3. In the SQL you can execute a single query at a time to perform operations on the database.
4. To execute a group of queries you can use the PL/SQL which considers as an advance part of the SQL.
5. SQL queries are categorized into 5 parts
   1. DDL
      1. Data Definition Language (DDL)
      2. The queries are used to work with a database schemas, structures such as table, database, constrains etc.
      3. Queries: **Create, alter, drop**.
   2. DML
      1. Data Manipulation Language (DML)
      2. The queries are use to perform operations on the data from the table such as create data, updating data, deleting data.
      3. Queries: **insert, Update, Delete**
   3. DCL
      1. Data Control Language (DCL)
      2. Using this queries you can decide which user can perform which operations can be setup.
      3. Queries: **Grant, Revoke**
   4. TCL
      1. Transaction Control Language (TCL)
      2. It is use to perform the database transaction. That’s is one transaction can be of multiple queries.
      3. Queries: Commit, Rollback, savepoint
   5. DQL
      1. Data Query Language (DQL)
      2. Is use to retrieve the data from the table.
      3. Query: **Select**

**DataType**

1. There are major 3 types of Data type
   1. Numeric DataType
   2. Text Datatype
   3. Date DataType

<https://www.w3schools.com/mysql/mysql_datatypes.asp#:~:text=In%20MySQL%20there%20are%20three,numeric%2C%20and%20date%20and%20time>

**DDL Queries**

**Create**

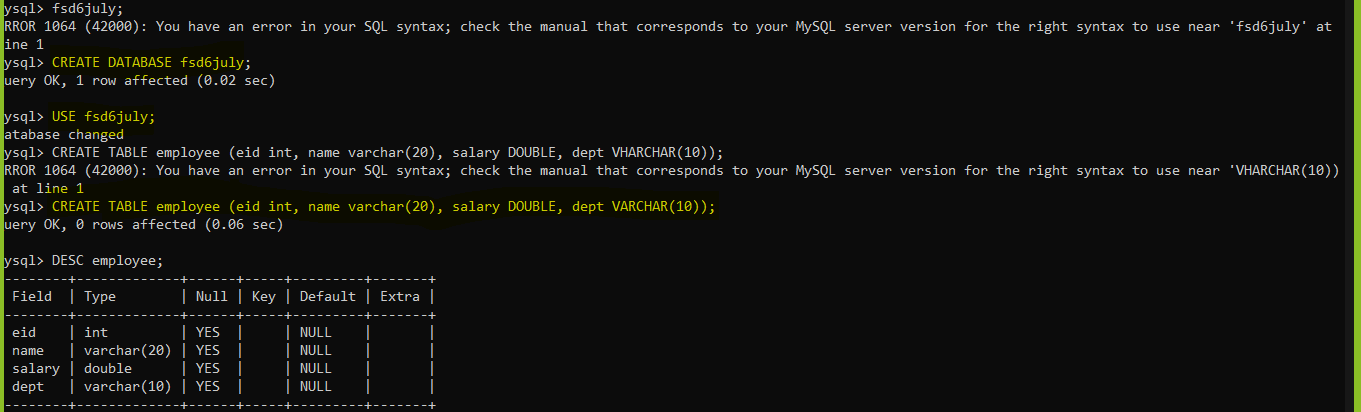
1. It is used to create new object into DB (Database, table, index, constraints)
2. Syntax:

Create <Object> <Name>;

Example:

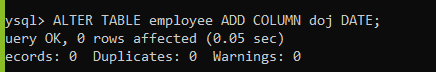
Create database test;

Create table employee(column data type)

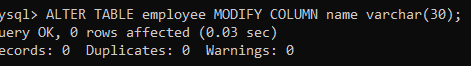


**Alter**

1. Using this you can modify the object structure.
2. There are different categories inside alter
   1. Add New Column

****

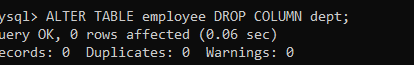
* 1. Modify the column

****

* 1. Rename the column

****

* 1. Drop Column (this will delete the records from table also)

****

**Drop**

1. It is used to drop the database Objects like table, Database, constraints

Syntax

DROP <OBJECT\_TYPE> <NAME>;



**Sql Constraints**

1. It is a way to apply a restriction on the columns of the table.
2. This restriction will be applied at the time of inserting data.
3. Using constraint, you can make sure that the valid data is getting inserted into DB.
4. There are some constraints are as follows
   1. **Not Null**: You can make sure that value are not null, and user is inserting value for the column.
   2. **Unique**: you can make sure that the values are unique and no duplicate values are allowed for the column.
   3. **Check**: you can apply a check on the values of the column. Check like the range of value or can allowed specific list of values.
   4. **Default**: To provide a default value for the column. By default the NULL values will be used as a default value.
   5. **Primary Key**: It is a combination of Not Null and Unique. Primary key column can be used as a foreign key in another table to make a relation between 2 tables. By Default, the indexes (Used for the faster searching/retrieval) will be applied on the Primary key column.
   6. **Foreign Key**: Foreign key values are the values refer from the primary key. In foreign key column values can be duplicate or values can be null. Using this key you can relate with the table having primary Key.

Create table Employee

Id – Primary Key

Name – Not null

Email – not null and unique

Country – default ‘IN’

Age – must be between 22-60

Gender – must be any one of the male,female, others

CREATE TABLE employee (

id int PRIMARY KEY,

name varchar(20) NOT NULL,

email varchar(30) NOT NULL UNIQUE,

country varchar(10) DEFAULT ‘IN’,

age int,

gender varchar(10),

CONSTRAINT age\_chk CHECK (age between 22 AND 60),

CONSTRAINT gender\_chk CHECK (gender IN (‘Male’,’Female’,’Other’))

);

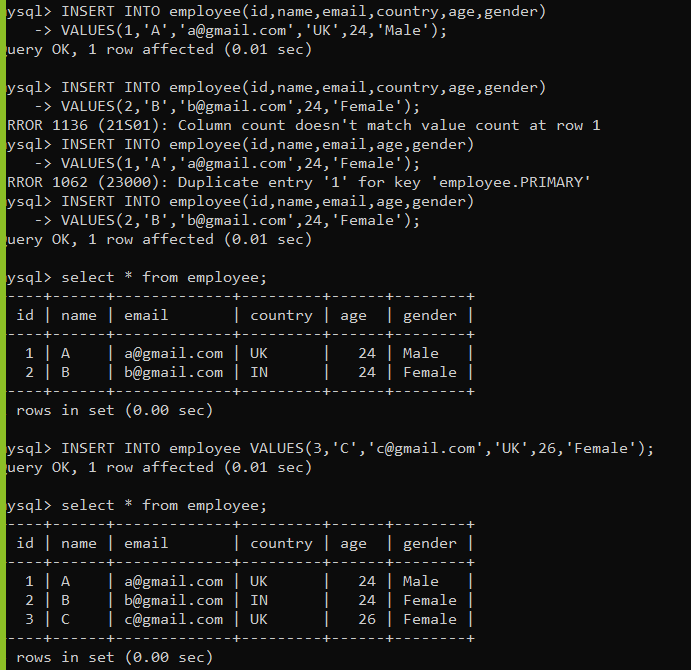
**DML**

**Insert Data Into Table**

Using This you can insert a record into table.

Syntax:

INSERT INTO <TABLE\_NAME>(column) VALUES(values);



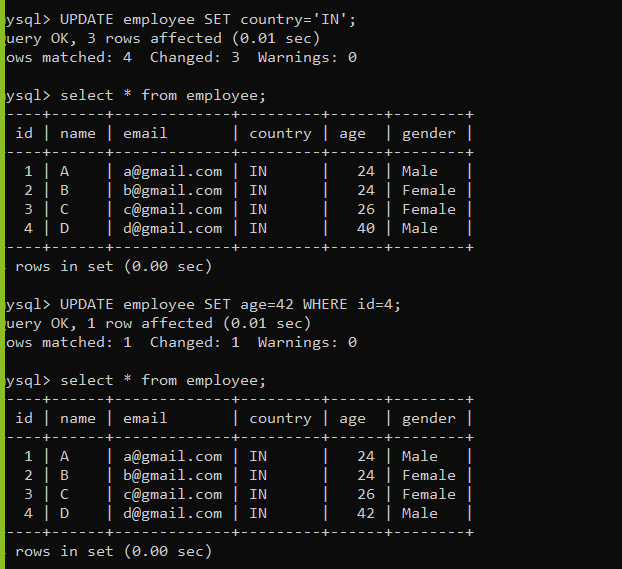
**Update Data from the table**

To update the details of the existing record.

Syntax:

UPDATE <table\_name> SET column=value, column=value, column=value;

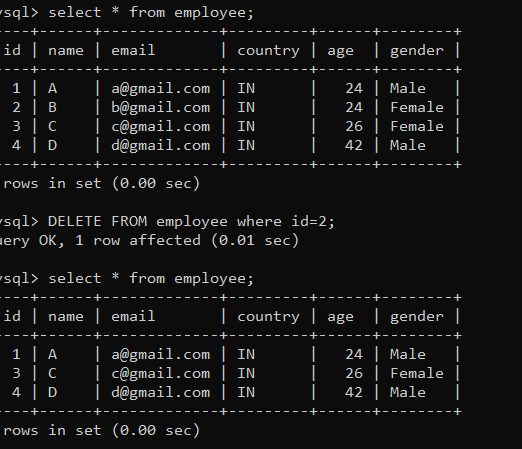
UPDATE <table\_name> SET column=value, column=value WHERE <Condition>;



**Delete Data From Table**

Syntax:

DELETE FROM <table\_name> where <Condition>;



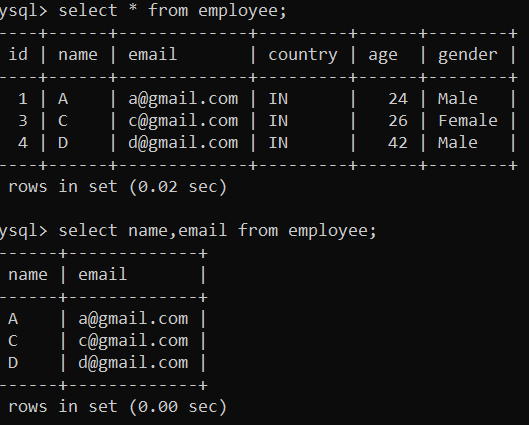
DQL:

1. Data Query Language
2. Using this you can select a record from the table.
3. You can select a record for specific row and column
4. To filter out the data you can use clauses.
5. Syntax:

**SELECT<COLUMNS> from the TABLE\_NAME Clause Condition.**

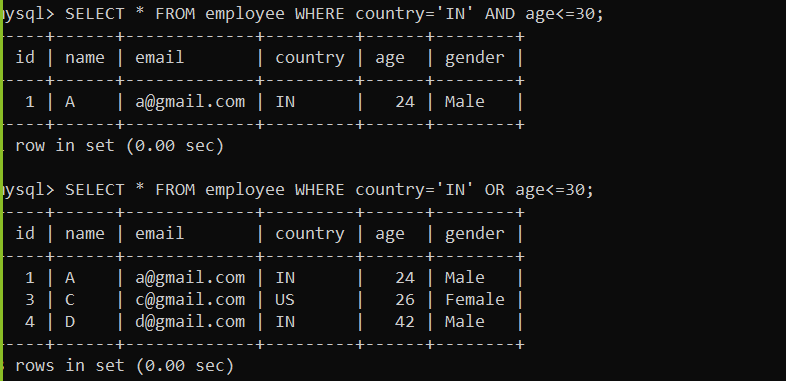
In Please of COLUMNS you can also use \* to get all the columns

To Select a Specific row clause and condition is used.

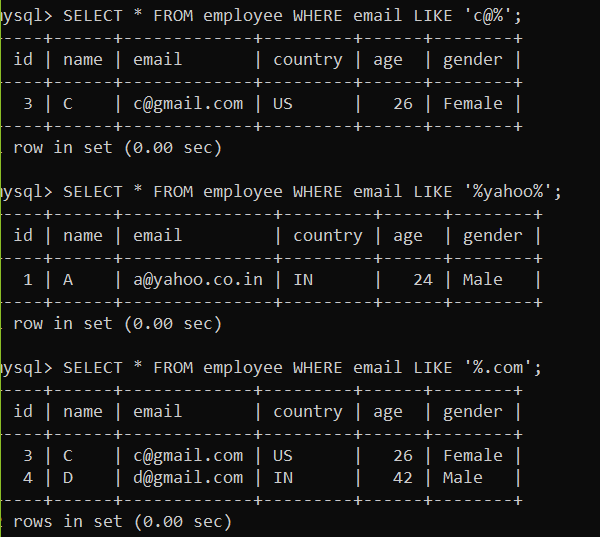


**Clauses**

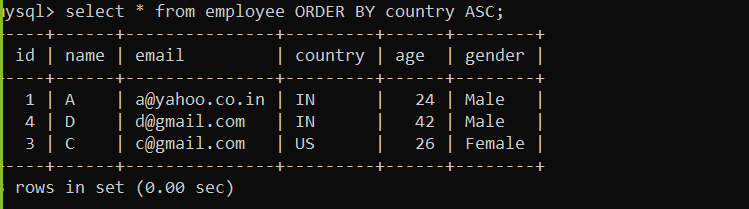
1. Clauses are used to filter out the data.
2. Clauses can be used for select, update, delete query.
3. Some of the mostly used clauses.
   1. Where Clause
      1. You can apply a condition on the data.
      2. Here you can also combine more than one conditions at a time.
      3. To the multiple conditions you can AND, OR operators.

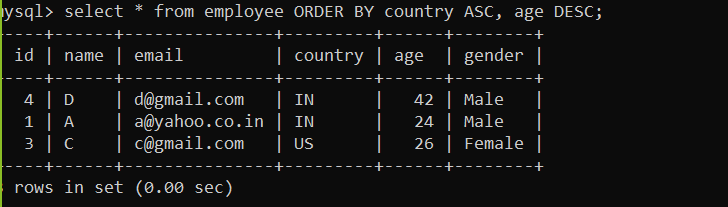


* 1. Like Clause
     1. This clause is use to get the records based on partial value.

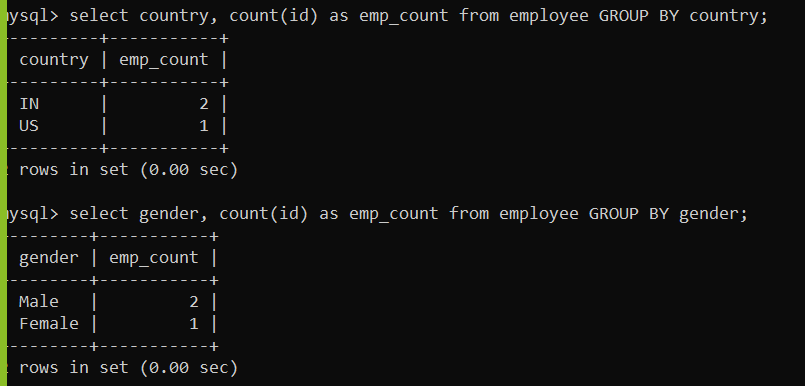


* 1. Order by clause
     1. It is use to get the data in the sorted format.
     2. The data can be sort using ASC (Ascending) or DESC (Descending)

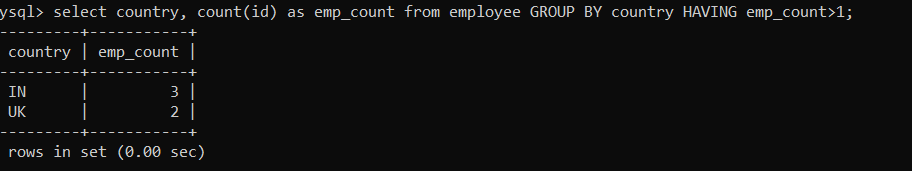




* 1. Group by
     1. It is used to group the records.
     2. This clause is mostly used for an aggregate function such as SUM, AVG, COUNT, MIN, MAX.



* 1. Having clause
     1. Having Clause is always use with a group by clause.
     2. This clause is use to apply condition on the grouped columns to get a result of specific rows.



Entity Relational Diagram (ERD)



CREATE TABLE studinfo(

sid int PRIMARY KEY,

name VARCHAR(20),

contact VARCHAR(10),

email VARCHAR(30)

)

CREATE TABLE studadd(

aid int PRIMARY KEY,

city VARCHAR(20),

pincode VARCHAR(10),

state VARCHAR(30),

sid int,

FOREIGN KEY (sid) REFERENCES studinfo(sid)

)

CREATE TABLE edudetails(

eid int PRIMARY KEY,

title VARCHAR(20),

percent DOUBLE,

passingyear INT,

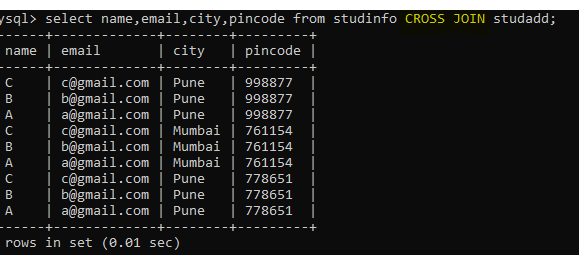
sid int,

FOREIGN KEY (sid) REFERENCES studinfo(sid)

)

JOIN

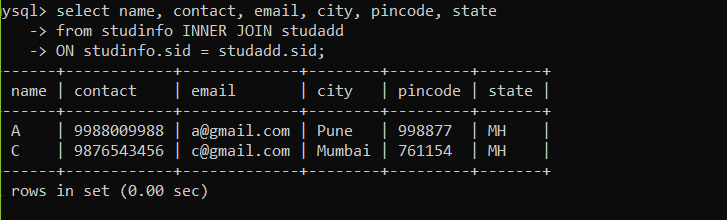
1. Joins are use to work with multiple table.
2. Join are use to select a records from a multiple table.
3. Types of Joins
   1. Cross/Certation Join
      1. It is use without where clause and it will produce an output in the multiplies of the 2 table rows.



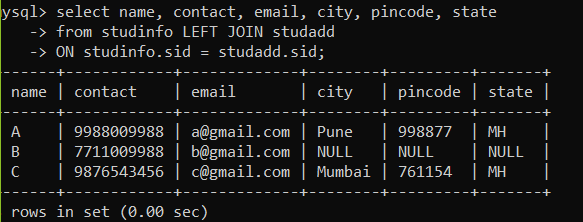
* 1. Inner Join
  2. Outer Join
     1. Left Outer Join
     2. Right Outer Join
     3. Full Join (Not supported in MYSQL but can be use on Oracle, Postgres etc.



Inner Join Example:



Left Outer Join Example:



Right Outer Join Example :

