**Data Base**

1. It used to store data and maintains the data.
2. Using Database you get the results easily using a query.
3. This data will be maintain for a longer period of time.
4. There are different type of databases Present in the market
   1. **Relational Database**
      1. You can store a values in the form of rows and column, in the database and table format.
      2. There can be a multiple tables which will be related with each other using primary and foreign key
      3. This data is also known as structure data.
      4. Example:

**MySql**, Oracle, Sql Server, Postgrace etc.

* 1. Document Database
     1. It is use to store a data into a document format and those document are in the form of JSON.
     2. It is not in the structure type and in the table format.
     3. Example:

MongoDB

* 1. Graph Database
     1. The data store in the form of tables, rows. But it will be represented as a graph.
     2. 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.

**SQL (Structure Query Language)**

1. SQL is use to execute a query in the Database.
2. This is the query language which needs to be follow to interact with DB.
3. SQL Has different categories in which the queries are distributed.
   1. **DDL (Data Definition Language)**
      1. Using this you can create, modify and drop a structure of the table and the different database object like table, index, database, view. Procedure, triggers etc.
      2. SQL Operations like CREATE, ALTER, DROP can be performed in this type.
   2. **DML (Data Manipulation Language)**
      1. Using this you can insert, update, delete the records in the tables.
      2. SQL Operation such as INSERT, UPDATE, DELETE can be performed in this type.
   3. **DQL (Data Query Language)**
      1. Using this you can retrieve the records from the Database. There can be clause, Joins used in this retrieval operation.
      2. SQL operations such as SELECT can be perform in this type
   4. **TCL (Transaction Control Language)**
      1. Using this you can make group of queries and it can be manages using in a transaction. This transaction can be successfully completed or it can be reverted.
      2. SQL operation like COMMIT, ROLLBACK, SAVEPOINT in this type
   5. **DCL (Data Control Language)**
      1. Using this you can control which can access what from the database.
      2. You can manage the user permissions.
      3. SQL operation like GRANT, REVOKE is this type.

**Data Types in MYSQL**

<https://www.w3schools.com/mysql/mysql_datatypes.asp>

1. String data type
2. Numeric data Type
3. Date time data type

**DDL (Data Definition Language)**

1. Create database and table in MYSQL
   1. Create Data base

Syntax : CREATE DATABASE <DatabaseName>;

Example: CREATE DATABASE fsd16Aug;

* 1. To Enter inside a database

Syntax: USE <DatabaseName>

* 1. Create Table

Syntax: CREATE TABLE <tableName> (Columnname Datatype, \_ , \_ , \_ , );

Example: CREATE TABLE employee(id int, name varchar(20), city varchar(20), salary double);

1. View the list of database/table
   1. To get the list of all the available Database

SHOW DATABASES;

* 1. To get the list of available tables in a specific Datatbasee

SHOW TABLES;

* 1. To get the metadata(info) of the table

DESC <table\_name>

1. Alter Query
   1. These are use to modify the structure which is already exist.
   2. Add new Column in Table
      1. Syntax: ALTER TABLE <tableName> ADD COLUMN <ColumnName> <DataType>
      2. Example: ALTER TABLE employee ADD COLUMN doj date;
   3. Modify the existing column
      1. You can modify the column details like Datatype, length, default value etc.
      2. Syntax: ALTER TABLE <tableName> MODIFY COLUMN <ColumnName> <DataType
      3. Example: ALTER TABLE employee MODIFY COLUMN name varchar(50);
   4. Remove the existing column
      1. Syntax: ALTER TABLE <tableName> DROP COLUMN <column\_name>
      2. Example: ALTER TABLE employee DROP COLUMN age;
   5. Rename the column
      1. Syntax: ALTER TABLE <tableName> RENAME COLUMN <OldName> TO <New\_Name>
      2. Example: ALTER TABLE employee RENAME COLUMN id TO empid;
2. Deleting the Database Objects (table/database)
   1. You can use drop query to delete the any database object
   2. This will be permanently delete from the Database.
   3. Syntax: DROP <ObjectType> <ObjectName>;
   4. Example: DROP TABLE employee;
   5. Example: DROP DATABASE fsd16aug;

Data Manipulation Language (DML)

1. Is use to work with the records from the tables.
2. You can manipulate the records from the table.
3. You can perform Insert, Update and Delete operation on the Data.
4. While handling data make sure that all string and date and timestamp type of values must be in single quotes and all the numeric values must be without single quotes.
5. Insert Operation
   1. To create new record(row) inside table.
   2. Syntax:

INSERT INTO <tablename>(columnname , \_ , \_ , \_) VALUES(value1, value2, \_, \_,\_)

* 1. Example

INSERT INTO emp(id,name,city,salary,doj) VALUES(12,'Abc','Pune',64364.43,'2020-02-15');

INSERT INTO emp VALUES(1,'Xyz','Mumbai',88364.43,'2015-03-15');

INSERT INTO emp(id,name,city,doj) VALUES(2,'Pqr','Mumbai','2022-01-20');

INSERT INTO emp VALUES(11,'Test1','Mumbai',34534.43,'2020-06-22'),(13,'Test3','Nagpur',54534.43,'2018-05-11'),(14,'Test4','Pune',64534.43,'2019-07-2');

1. Update Operation
   1. You can update the existing record details using update query
   2. Syntax:

UPDATE <tablename> SET columname=updatedvalue, columname=updatedvalue WHERE <condition>;

* 1. Example:

UPDATE emp SET salary = salary+((salary/100)\*10);

UPDATE emp set salary=34343.56 WHERE id=2;

1. Delete Operation
   1. You an delete the data from the database.
   2. You can delete the data based on condition using Where clause.
   3. Syntax:

DELETE FROM <tablename> where <condition> ;

* 1. Example:

DELETE FROM emp WHERE id=15;

**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 Student Table with following column

1. id int – not null and unique
2. name varchar – not null
3. contact varchar – not null unique
4. email varchar – unique
5. city varchar – Default values must be ‘Pune’ if not provided
6. age int – must be between 5 to 25 only
7. gender varchar – must be “Male” or “Female”

create table student(

id int PRIMARY KEY,

name varchar(30) NOT NULL,

contact varchar(10) NOT NULL UNIQUE,

email varchar(50) UNIQUE,

city varchar(20) DEFAULT 'Pune',

age int,

gender varchar(10),

CONSTRAINT age\_chk CHECK (age BETWEEN 5 AND 25),

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

);

Data Query Language (DQL)

1. It is use to retrieve the data from the DB.
2. In this SELECT query is used with multiple variation such as using conditions, Clauses, join etc.
3. You can select all the rows and columns from the database using following query

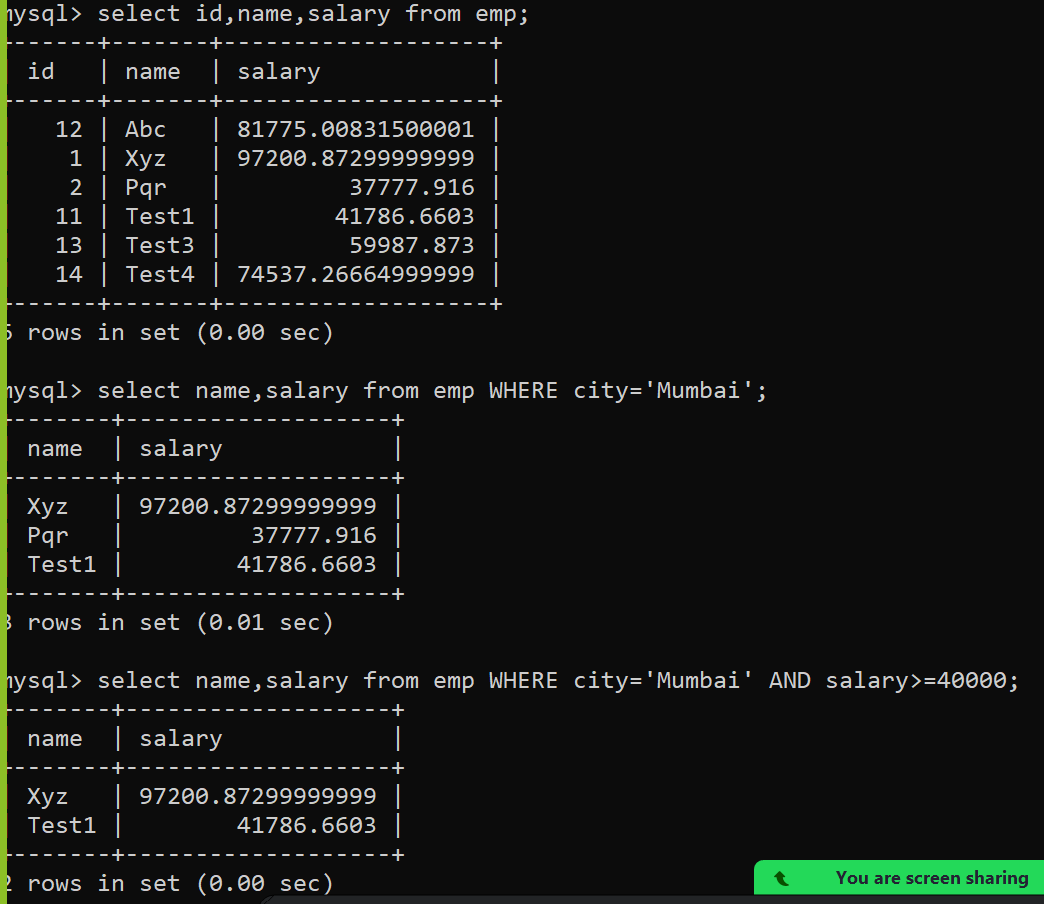
Syntax: **SELECT \* FROM <TableName>**

1. You can select a specific columns from the table

Syntax: **SELECT columnname, columnname FROM <tableName>**

1. You can select a specific rows from the table to do this you can use a clause in SQL
   1. Where clause
      1. Using where clause you can add a condition on the row selection.
      2. In the where clause you apply more than once condition and this conditions can be combine together with AND, OR clause.
      3. You can also select a records based on Range or Based on List of values. Using IN and BETWEEN clause
      4. There clause can be use while select, update and delete operation.
      5. Syntax:

SELECT \* FROM <tablename> WHERE condition AND/OR condition;



Select records using IN clause (you can specify a list of value)

select id,name,salary,city from emp WHERE **city IN ('Mumbai', 'Pune');**

Select records using BETWEEN clause (you can specify a range of value)

select id,name,salary,doj from emp where **doj BETWEEN '2018-01-01' AND '2021-12-31';**

* 1. Like clause
     1. Like clause is use to get the records based on the partial value.
     2. Like clause has to use with a where clause.
     3. In Like clause you can specify the value which start with some char ‘char%’ or ends with some char ‘%char’ or can search by any of the character occurred in between ‘%char%’

* 1. Order by clause
     1. You can select a records based in order.
     2. The order can be ascending (ASC) or descending (DESC).
     3. By default ascending order will be followed
     4. You can also specify the multiple column in the order by clause.
     5. Example

**select \* from emp ORDER BY id ASC;**

**select \* from emp ORDER BY id DESC;**

**select \* from emp ORDER BY city, name;**

* 1. Limit clause
     1. You can specify how may rows you wanted to select by specifying a limit clause.
     2. Exception

select \* from emp ORDER BY salary DESC **LIMIT 3;**

* 1. Group by clause
     1. To group the records and perform the operations on the grouped records.
     2. Mostly Operations like aggregate functions are use in group by clause
     3. Aggregate functions such as sum, avg, count, min, max
     4. Syntax:

Select function() from <tablename> GROUP BY <column\_name>

* + 1. Example:

SELECT city, count(id) AS emp\_count FROM emp **GROUP BY (city);**

* 1. Having clause
     1. Having clause is use to perform the filtration on the group result
     2. You cannot use where clause in the grouped result.
     3. Having clause must be used wit group by clause only.
     4. Example:

SELECT city, count(id) AS emp\_count FROM emp GROUP BY (city) **HAVING emp\_count>1**;

**Joins**

1. Joins are use to work with multiple table together.
2. Using this you can connect two or more table together.
3. The reference of one table will be store inside another table using joins.
4. This is possible with the help of foreign key and Primary key.
5. You can also retrieve the data from the multiple tables using joins.



create table student(id int PRIMARY KEY, name varchar(30),contact varchar(10),city varchar(20));

create table stud\_education(

id int PRIMARY KEY,

title varchar(20),

percent double(4,2),

passingyr varchar(4),

sid int,

**FOREIGN KEY(sid) REFERENCES student(id)**

);

There are different type of joins present

1. Inner Join

mysql> SELECT s.id,name, contact, city, title, passingyr, percent

-> FROM student s **INNER JOIN** stud\_education e

-> WHERE s.id=e.sid;

+----+------+------------+-------+-------+-----------+---------+

| id | name | contact | city | title | passingyr | percent |

+----+------+------------+-------+-------+-----------+---------+

| 2 | B | 6688776677 | Pune | BE | 2020 | 78.65 |

| 5 | E | 9898776677 | Delhi | BSC | 2021 | 79.65 |

+----+------+------------+-------+-------+-----------+---------+

1. Outer Join
   1. Left Outer Join

mysql> SELECT s.id,name, contact, city, title, passingyr, percent

-> FROM student s **LEFT JOIN** stud\_education e

-> **ON** s.id=e.sid;

+----+------+------------+--------+-------+-----------+---------+

| id | name | contact | city | title | passingyr | percent |

+----+------+------------+--------+-------+-----------+---------+

| 1 | A | 9988776677 | Pune | NULL | NULL | NULL |

| 2 | B | 6688776677 | Pune | BE | 2020 | 78.65 |

| 3 | C | 7878766677 | Delhi | NULL | NULL | NULL |

| 4 | D | 9988776454 | Mumbai | NULL | NULL | NULL |

| 5 | E | 9898776677 | Delhi | BSC | 2021 | 79.65 |

+----+------+------------+--------+-------+-----------+---------+

* 1. Right Outer Join

mysql> SELECT s.id,name, contact, city, title, passingyr, percent

-> FROM student s **RIGHT JOIN** stud\_education e

-> **ON** s.id=e.sid;

+------+------+------------+-------+-------+-----------+---------+

| id | name | contact | city | title | passingyr | percent |

+------+------+------------+-------+-------+-----------+---------+

| 2 | B | 6688776677 | Pune | BE | 2020 | 78.65 |

| 5 | E | 9898776677 | Delhi | BSC | 2021 | 79.65 |

| NULL | NULL | NULL | NULL | ME | 2022 | 67.65 |

| NULL | NULL | NULL | NULL | MCA | 2019 | 88.65 |

+------+------+------------+-------+-------+-----------+---------+

* 1. Full Outer Join (It is not supported in MySql but you can use in another databases)

