MYSQL SUMMARY DAY 1

- Data we can store temporarily in memory, and to structure our data in memory we have Data structures algorithms, Now to store data permanently we need Databases and multiple databases we have present, which organize our data in the best way in the permanent storage.

- To store data in persistent storage, we need to write our data inside a file(table) and file we need to put inside a folder(Directory)

- DBMS - Database Management System, There are some standard database management systems that all the companies follow to manage their data, Users of the system are given facilities to perform several kinds of operations on such a system for either manipulation of the data in the database or the management of the database structure itself. Database Management Systems (DBMSs) are categorized according to their data structures or types. To implement DBMS we have a lot of product like MySql, MongoDB and many more.

- SQL- Structured query language, We use SQL to do queries and analyses over data

- Download MySQL server - https://dev.mysql.com/downloads/installer/

- Mysql workbench will be used as a client-side tool to query our data stored in the MySQL server.

- SQL is case insensitive, So we can write SQL queries either in capital or small letters.

- **SHOW DATABASES;** -> Tt will show available databases in the MySQL server.

- **CREATE DATABASE lwdb** -> It will create a database named lwdb

- **USE lwdb** -> We will go inside lwdb database now.

- **SHOW tables;** -> It will show all tables we have inside lwdb database.

- **CREATE table students(salary int ,name char(30),mobile char(13) ,remarks char(20));** - it will create a table name students with the following columns.

- name char(30) -> means we are putting a constraint that a maximum of only 30 characters we can type in the “name” column.

- **SELECT * from students**; -> It will show all the columns and rows in this table

- **insert into students values (40,"ram", "1234","good");** - it will insert one new row with these data in the students table.

- **Insert into students (name, mobile) values(“Rohit”,”12345”);** -> Now a new record will be created but only the “name” and “mobile” columns will have data, else other columns will be null.

- **2- Tier Architecture**- When the client/user directly inserts the data into the MySQL server then this architecture is called a 2-tier architecture. The 2-tier architecture is based on a client-server machine.

- **3- tier architecture-** When between the user and MySQL server we have some programs in between like web applications, the mobile applications, Which in behalf of user interact with MySQL server, then this architecture is known as 3-tier architecture.

- **Drop table students;** -> It will drop the students table.

- Create table students (
```
                    name char(30),   -> Maximum 30 characters we can insert
                    gender enum(‘M’,’ F’),  -> As gender could be only M or F
                    phone char(13),
                    course char(40),
                    price decimal(3,2),       -> decimal is a datatype
                    time char(30)  ->storing time in char datatype
                    );
```
- Database design is the organization of data according to a database model.

- *Insert into students values*
```
                      (“Vimal”,” M”,”22246”, “ML”,4.5,” Monday”),
                      (“Vimal”,” M”,”22246”, “ML”,3.5,” Tuesday”),
                       (“pop”,” M”,”3334546”, “ml”,4.5,” Monday” );
```

- **Select phone from students WHERE name=” pop”;** -> Now it will show the phone no of all rows where the name is “ram”.

- **Select sum(price) from students where course =” ML”** Now it will sum the total price of the course having ML(or ml). (4.5+3.5+4.5=13.5)

- Now here we have three problems
    - By Mistake, one record can be added twice, and then while the analytics time it will show wrong insights to us (like total earning will be shown more than what actually we have achieved) (Duplicasy problem)
    - Now for the “ML” program price is the same for everybody but by mistake, someone might write the price wrong(let’s say price =3.5), Again this will be an issue for us.
    - Now If 100K of students have registered for the “ML” program then in the MySQL database writing “ML” in the “course” column for 1000’s students will take a lot of storage to store just the “ML” word, So here we are wasting the money and storage, Instead what we can do is, we can define the course somewhere else, and give it’s reference everywhere, now it will save a lot of

space as well as no chance of writing wrong course name by the user while inserting data, and in future, if we have to change course name from “ML” to “Machine learning” then we can do it from one place only.

- Now to avoid the above three mistakes we have to organize our database effectively and that is called database design. Database design is the organization of data according to a database model.

- **Normalization** is the process of organizing data in a database This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

- Now let’s solve all the three problems one by one, Now to remove duplicity we can make any column “unique”, Now onward while inserting data, first of all, it will check if that column value which we want to insert is unique if yes, then only data will be inserted, Now as a DB guy we have to plan which of your column can be unique, “name” can’t be unique, but phone no can be unique, So while creating the table
Create table students (
name char(30),
gender enum(‘M’,’ F’), -> As gender could be only M or F
phone char(13) unique, -> ***We have made Phone no Unique***
course char(40),
price decimal(3,2), -> decimal is a datatype
time char(30) ->datetime to store date&time
);
- Above Changes to make the “phone no” unique will not completely solve our problem, If the user wanted to register for the “DevOps” course also, then he will be not able to write records in the Mysql server, which is not a good thing for our business, So here we need to make some other column unique, i.e “OrderId” We can add a new column in our table i.e “OrderId” which will be unique to every row (**OrderIdint unique auto_increment**) “auto_increment” will keep on incrementing the OrderId for further rows.

- We have to store data in such a way that we can write queries to get a single row at a time for eg -> **Select * from students where Orderid=1,** -> *It will show details about OrderId 1*

- **RDBMS** (Relational database management system) - Now to manage our database we are going to create multiple tables consisting of their respective columns, and then we will define some relations between them.

- The field/column which is capable of giving individual records on a query is known as a **key**, And if we give its a reference to some other table then it is called a **primary key**.

- A **foreign key** is the one that is taking reference from the **primary key** to show data in his column, So A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

- 1st Normalization form has 3 rules
  - None of the records should be same (No duplicity)
  - Manage data in multiple tables(primary key and foreign key)
  - Per column per record should have a single value (Means in the “course” column we can’t put “course” as “ML, DevOps” for a user if the student has bought both the programs, Because in this way query and doing analytics will be very confusing.

- Now Here For Managing our DB we have created 3 tables.

```
create table students (
     sid int   primary key   auto_increment,
    first_name char(30) not null,
    last_name char(30),
    gender enum("M", "F"),
    phone_number char(12) unique
);

describe students; - To describe your table

create table courses (
    cid int auto_increment,
    course_name varchar(40) unique,
    price decimal(4,2),
    teacher char(30),
    primary key (cid)
);

desc courses;


create table orders (
    oid int auto_increment primary key,
    course_id  int,
    student_id int,
    order_time datetime,
    foreign key (course_id)  references courses(cid),
    foreign key (student_id) references students(sid)
);
```

Now we have created three tables, and they are in relation.

- Schema is logical planning of DB, like how many tables we will create, how they will be connected, what columns we will have, and what datatypes we will use, all this planning is called Schema.