### DBMS

MySQL is a database....

* SQL is a structure query language using for perform MySQL....
* RDBMS => Relational Data Base Management System....

* Work on tabular form....

DATABASE => it is a space : work store data like image ,tables etc ....

🡺 A database is an organized collection of data that is store and managed in a way

Type of sql language 🡺🡺

* **DDL**– Data Definition Language
* **DQL**– Data Query Language
* **DML**– Data Manipulation Language
* **DCL**– Data Control Language
* **TCL**– Transaction Control Language

CREATE DATABASE => CREATE DATABASE database-name

Types of Databases

* There are several types of databases, that are briefly explained below.

1. Relational Database Management System (RDBMS)
2. Non-Relational Database Management System (NoSQL or Non-SQL)
3. Hierarchical databases
4. Network databases
5. Object-oriented databases
6. Relational databases
7. Cloud Database
8. Centralized Database
9. Operational Database
10. NoSQL databases

* USE DATABASE 🡺🡺

use database-name

* CREATE TABLE 🡺🡺

CREATE TABLE Course(course\_id INT PRIMARY KEY,course\_name varchar(15) not null)

* SHOW TABLE STRUCTURE 🡺🡺

DESCRIBE course(table name);

* INSERT DATA INTO TABLE =>
* INSERT INTO course(course\_id,course\_name)

VALUES

(1,'Mathematic'),

(2,'Physics'),

(3,'Chemistry');

DDl ( Data Definition Language)=(delete, alter,update)🡺🡺🡺

CREATE STUDENT TABLE 🡺

CREATE TABLE student(name varchar(20) not null,

address varchar(50) not null,

contact varchar(10) not null ,

stud\_id int PRIMARY key

)

DML {Data Manipulation Language}==(insert)🡺🡺🡺

INSERT DATA INTO STUDENT TABLE 🡺

INSERT INTO student(name,address,contact,stud\_id)

VALUES

('Yogendra Patidar','Atal Dwar,Indore','8746352345',101),

('Saksham Malviya','Vijay Nagar,Indore','7838237264',102),

('Mani Agrawal','ByPass,Indore','6573836463',103),

('Abhishek Umre','Kalani Nagar,Indore','7643568976',104),

('Atishay Jain','Gita Bhawan,Indore','8654936785',105);

* DataType 🡺🡺
* Int
* Decimal
* Varchar
* Text
* Date
* Date and time
* Boolean
* Const

Constent type🡺🡺

* Primary key == For fetch
* Foreign key == all data store in object
* Not null ==Required
* Default ==
* Enum == choice based fix data
* Unique key == Unique value
* Check == conditional

Add one more new field in table 🡺🡺

ALTER TABLE course ADD COLUMN fees int

Modify table field datatype 🡺🡺

ALTER TABLE course MODIFY fees varchar(30)

Delete specific table field 🡺🡺

ALTER TABLE course DROP fee

Create new database 🡺 🡺

CREATE DATABASE empdb;

Use data base🡺🡺

Use empdb;

Create new emptable 🡺🡺

CREATE TABLE emptable(Empid int PRIMARY key not null,

First\_name varchar(50) not null,

Last\_name varchar(50) NOT null,

Age int not null,

Hire\_date date,

Address varchar(100)

);

Insert data into epmtable 🡺🡺

INSERT into emptable(Empid,First\_name,Last\_name,Age,Hire\_date,Address)

VALUES

(101,'Yogendra','Patidar',20,'2024-11-10,'Atal Dwar,Indore');

Change fiend name 🡺🡺

ALTER TABLE tbalename CHANGE oldfeildname newfieldname datatype constent;

ALTER TABLE student CHANGE contact mobile varchar(10);

Delete DataBAse 🡺🡺

DROP DATABASE database\_name

Delete Particular table 🡺🡺

DROP TABLE table\_name;

Database 🡺🡺

CREATE DATABASE new\_empdb;

Department 🡺

CREATE TABLE Department(dept\_id int PRIMARY key not null,dept\_name varchar(50) not null)

Designation🡺

CREATE TABLE Designation(desg\_id int PRIMARY key not null,desg\_name varchar(50) not null)

Create Employee table or foreign key 🡺🡺

CREATE TABLE Employee(emp\_id int PRIMARY key not null ,

emp\_name varchar(50) not null ,

emp\_age int NOT null,

emp\_salary int ,

dept\_id int,

desg\_id int,

FOREIGN KEY ( dept\_id )REFERENCES department (dept\_id),

FOREIGN KEY ( desg\_id) REFERENCES designation(desg\_id)

)

How to create Foreign key 🡺🡺 FOREIGN KEY (field name which you to create foreign key) REFERENCES table name which to get primary key (reference table field key name)

Insert data into department table 🡺🡺

INSERT INTO department(dept\_id,dept\_name)

VALUES

(1,'Manager'),

(2,'Sales'),

(3,'IT');

Insert data into designation table 🡺🡺

INSERT INTO designation(desg\_id,desg\_name)

VALUES

(1,'Developer'),

(2,'Sales Excutive'),

(3,'HR');

Insert data into employee table🡺🡺

INSERT INTO employee(emp\_id , emp\_name , emp\_age , emp\_salary , dept\_id , desg\_id)

VALUES

(1,'Saksham Malviya',30,27000,1,1),

(2,'Yogendra Patidar',20,25000,2,2),

(3,'Ravi',25,20000,3,3);

Show tablel data 🡺🡺

SELECT \* from department

Fetch data from foreign name 🡺🡺

SELECT \* FROM student WHERE course\_id IN ( SELECT course\_id FROM course WHERE course\_name = 'Hindi' );

Fetch student by course name 🡺🡺

SELECT \* FROM student

JOIN course ON student.course\_id = course.course\_id

WHERE course.course\_name = 'Hindi';

Update data in table🡺🡺

UPDATE student SET stud\_name='Sakku' WHERE stud\_name='Saksham'

For fetch particular fild data with conditions 🡺🡺

SELECT stud\_name , stud\_age from student WHERE stud\_name ="Saksham"

For fetch all field data with conditions 🡺🡺

SELECT \* from student WHERE stud\_name ="Saksham"

* Where you to update database structure where we use(DDL) 🡺🡺 ALTER

Syntax 🡺

ALTER TABLE tbalename CHANGE oldfeildname newfieldname datatype constent;

* ALTER TABLE student CHANGE contact mobile varchar(10);
* Where you to update data of table where we use (DML)🡺🡺

UPDATE🡺🡺

Syntax🡺

UPDATE table\_name SET col\_name=VALUE Where

Whichh Reccord

* UPDATE student SET stud\_name='Sakku' WHERE stud\_name='Saksham';

Delete Particular data row from table 🡺🡺

Syntax🡺🡺

* DELETE FROM table\_name WHERE row\_id or name = data
* DELETE from student WHERE stud\_name = "Abhishek"

Like query🡺🡺

SELECT \* from employee WHERE city LIKE "I%"

% this use for after any length

SELECT \* from employee

WHERE city LIKE "\_\_\_\_\_"

(\_) underscore is use for one char

SELECT \* from employee

WHERE city LIKE "B\_\_\_\_\_"

SELECT \* from employee

WHERE city LIKE "B\_\_\_%"

SELECT \*

FROM employee

WHERE city LIKE 'B%'

OR city LIKE 'C%'

OR city LIKE 'D%'

OR city LIKE 'E%'

OR city LIKE 'F%'

OR city LIKE 'G%'

OR city LIKE 'H%'

OR city LIKE 'I%';

SELECT \*

FROM employee

WHERE city LIKE [A-I]

SELECT \*

FROM employee

WHERE city LIKE 'B%'

OR first\_name ="Ram"

Between Query(from to from ) 🡺🡺

SELECT \* FROM employee

WHERE emp\_salary BETWEEN 20000 AND 30000

IN Query(specific data fetch) 🡺🡺

SELECT \* from employee WHERE emp\_salary in (20000,30000,50000)

**Query🡺🡺**

* SELECT \* from employee WHERE emp\_salary = 20000 AND emp\_age=20
* UPDATE employee SET emp\_salary = 40000 WHERE emp\_salary = 20000 and emp\_age =20
* UPDATE employee SET emp\_salary = 60000 WHERE emp\_salary = 40000 and emp\_age =20 AND first\_name LIKE "R%"
* UPDATE employee SET emp\_salary = 60000 WHERE emp\_salary = 40000 and emp\_age =20 AND first\_name LIKE "%m"
* UPDATE employee SET emp\_salary = 60000 WHERE emp\_salary = 40000 and emp\_age =20 AND first\_name LIKE "%a%"
* UPDATE employee SET emp\_salary = 60000 WHERE emp\_age =19 AND first\_name LIKE "%k%"
* UPDATE employee SET emp\_salary = emp\_salary + 200 where emp\_age = 20;

**Order by Query 🡺🡺**

* SELECT \* from employee ORDER BY first\_name ( output will Acending order )
* SELECT \* from employee ORDER BY first\_name DESC ( output will Decending order )
* SELECT \* from employee ORDER BY emp\_id DESC
* SELECT \* from employee ORDER BY emp\_id

**For fetch unique data 🡺🡺**

* SELECT DISTINCT dept\_name from department

**Avrage🡺🡺**

* SELECT AVG(emp\_salary) from employee

**Total of column value 🡺🡺**

* SELECT SUM(emp\_salary) from employee

**Fetch MAX valule of column 🡺🡺**

* SELECT Max(emp\_salary) from employee

**Fetch conditional data 🡺🡺**

* SELECT \* FROM employee WHERE emp\_salary=40000 AND emp\_age=22 and city="Indore"
* SELECT \* from employee WHERE city !="Indore" AND desg\_id = 3 AND emp\_age= 20 and emp\_salary=20000
* SELECT \* from employee WHERE city !="Indore" AND emp\_age= 20 and emp\_salary=20000 AND(SELECT desg\_name from designation WHERE desg\_name ="HR")
* SELECT \* from employee WHERE city !="Indore" AND desg\_id = 3 AND emp\_age= 20 and emp\_salary=20000

**Join query 🡪**

* Create database database\_name(empdb1)
* Create table CREATE TABLE department(dept\_id int PRIMARY key not null,dept\_name varchar(20) not null)
* CREATE TABLE employee(emp\_id int PRIMARY key not null,emp\_name varchar(20) not null, dept\_id int)

**Insert data🡪**

INSERT Into department(dept\_id,dept\_name)

VALUE

(101,"HR"),

(102,"Sales"),

(103,"IT"),

(104,"Account")

INSERT Into employee(emp\_id,emp\_name,dept\_id)

VALUE

(1,"Sarthak",101),

(2,"Yogendra",103),

(3,"Saksham",103),

(4,"Abhishek",102),

(5,"Aditya",104)

Fetch join data 🡪🡪

INNER JOIN 🡺Fetch join data

* SELECT e.emp\_id,e.emp\_name,d.dept\_name,d.dept\_id FROM employee e INNER JOIN department d ON e.dept\_id = d.dept\_id

LEFT JOIN 🡪🡪show all left table data if data is null show null

* SELECT e.emp\_id,e.emp\_name,d.dept\_name,d.dept\_id FROM employee e LEFT JOIN department d ON e.dept\_id = d.dept\_id

Right JOIN 🡪🡪Oppsite to LEFT JOIN

LEFT JOIN 🡪🡪show all left table data if data is null show null

* SELECT e.emp\_id,e.emp\_name,d.dept\_name,d.dept\_id FROM employee e RIGHT JOIN department d ON e.dept\_id = d.dept\_id

Create New database 🡪🡪

Cratate database empdb2;

Create table1🡪🡪

* CREATE table management(mg\_name varchar(20) not null, mg\_id int PRIMARY key not null)
* CREATE TABLE emp\_list(emp\_id int PRIMARY key not null,
* emp\_name varchar(50) not null ,
* mg\_id int,
* FOREIGN KEY (mg\_id) REFERENCES management(mg\_id)
* )

**Insert data into tables 🡪🡪**

* INSERT into management(mg\_id,mg\_name)

VALUES

(1,"Sales"),

(2,"HR"),

(3,"Marketing"),

(4,"Accounts")

INSERT into emp\_list(emp\_id,emp\_name,mg\_id)

VALUES

(1,"Sales",2),

(2,"HR",3),

(3,"Marketing",1),

(4,"Accounts",4)

* CREATE TABLE emp\_list(emp\_id int not null PRIMARY key,emp\_name varchar(50) not null , mg\_id int ,

FOREIGN KEY (mg\_id) REFERENCES emp\_list(emp\_id))

* INSERT INTO EMPLIST (emp\_id,emp\_name,mg\_id,)
* VALUES (1,'Ajay',1),(2,'Yogendra',2),(3,'Sarthak',2),(4,'Abhishek',1){Insert Data silege-single}

**UNION JOIN🡪🡪🡪**

SELECT e.emp\_id,e.emp\_name,d.dept\_name,d.dept\_id FROM employee e LEFT JOIN department d ON e.dept\_id = d.dept\_id

UNION

SELECT e.emp\_id,e.emp\_name,d.dept\_name,d.dept\_id FROM employee e RIGHT JOIN department d ON e.dept\_id = d.dept\_id

***CROS JOIN🡺🡺***

* SELECT e.emp\_id,e.emp\_name,d.dept\_name,d.dept\_id FROM employee e CROSS JOIN department d

***SELF JOIN🡺🡺***

* SELECT e1.emp\_name employee,e2.emp\_name manager from emp\_list e1 LEFT join emp\_list e2 ON e2.emp\_id = e2.mg\_id

***Steps for create Views Table🡺🡺***

CREATE VIEW emp\_view AS

SELECT emp\_id,first\_name,last\_name,emp\_age,emp\_salary FROM employee

Imp🡺🡺

SELECT DEPT\_ID ,COUNT(emp\_id) employee FROM employee

GROUP BY DEPT\_ID

SELECT DEPT\_ID ,COUNT(emp\_id) emp\_view FROM emp\_view WHERE(emp\_salary < 30000)

GROUP BY DEPT\_ID

***AS🡺🡺***

* The **SQL AS** keyword is used to create an alias for a column or table. This alias exists only for the duration of the query and helps in making the output more readable.

***GROUP BY🡺🡺***

🡪The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

***View Query 🡺🡺***

CREATE VIEW emp\_avg\_salary AS

SELECT emp\_id,first\_name,last\_name,emp\_salary FROM employee

* CREATE VIEW emp\_view AS

SELECT employee\_id,first\_name,last\_name,department,designation,salary FROM employee

* CREATE VIEW it\_emp\_high\_salary AS

SELECT employee\_id,first\_name,last\_name,department,designation,salary FROM employee WHERE department ="IT" and salary>60000

* CREATE VIEW recent\_hires AS

SELECT employee\_id,first\_name,last\_name,department,designation,salary FROM employee WHERE hire\_date =”2019-01-01”

***Update query🡺🡺***

UPDATE employee set salary= salary\*1.10 WHERE department ="IT"

UPDATE emp\_view set department= "Marketing",salary = 70000 WHERE employee\_id =5;

Deleti Query🡺🡺

DELETE FROM employee WHERE employee\_id IN(SELECT employee\_id FROM emp\_above\_30 WHERE age>50)