**Name: Pratik Sambhaji Pawar**

**Day 21**

**MYSQL**

**Database:**

It is the collection of data in a format that can be easily accessed.

**DBMS:**

A software application used for database management

**User-🡪DBMS🡪Database**

**Types of database:**

* **Relational:** Data stored in the table. Uses sql and RDMS. Eg. Mysql, PostgreSQL
* **Non-relational**: Data stored in the table format .uses no – sql . Eg. Mongodb

**SQL:**

**-**Structured Query language

-SQL is a programming language used to interact with relational database

-It used the CRUD operations:

Create

Read

Update

Delete

**What is a table**:

-Table is the combination of rows and columns.

-Row are horizontals, which show the individual information.

-columns are verticals, Which show the structure and schema.

**Create Our First Database**:

**Syntax**

**Eg.** CREATE DATABASE temp1;

**Delete database:**

**Syntax:**

DROP DATABASE db\_name;

Eg.

DROP DATABASE temp1:

**Create table:**

**Syntax:**

CREATE TABLE *table* \_name(

Col\_name1 datatype constraint,

Col\_name2 datatype constraint,

Col\_name3 datatype constraint

)

**Eg.**

CREATE TABLE student(

Id INT PRIMARY KEY,

Name VARCHAR (50),

Age int NOT NULL

)

**SQL datatype:**

-CHAR -VARCHAR

-INT -BIGINT

-BIT -FLOAT

-DOUBLE -BOOLEAN

-DATE -YEAR

**Types of SQL Commands:**

**DDL(Data Definition Language):** create, alter, rename, truncate, drop.

**DML(Data Manipulation Language);** select, insert, update, delete.

**DCL(Data Control Language**): grant, revoke**.**

**TCL(Transaction Control Language):** commit, rollback**.**

**Database Related Queries:**

-CREATE DATABASE *db \_name***;**

**-**CREATE DATABASE IF EXIST NOT EXIST*db \_name***;**

-DROP DATABASE *db \_name*;

-DROP DATABASE IF NOT EXIST *db \_name*;

-SHOW DATABASE;

-SHOW TABLE;

**Table Related Queries:**

-**Create:**

**Syntax:**

CREATE TABLE *table* \_name(

Col\_name1 datatype constraint

)

**Eg.**

TABLE student(

Id CREATE INT PRIMARY KEY,

Name VARCHAR (50),

Age int NOT NULL)

-**Select and View ALL Columns:**

Syntax:

SELECT \* FROM *table \_name;*

**Eg.**

SELECT \* FROM student;

-**Insert:**

Syntax:

INSERT INTO *table \_name(col\_1,col\_2optional)* VALUES(col1\_v1,col2\_v2),(col1\_v2,col2\_v2);

Eg.

INSERT INTO student (roll \_no ,names) VALUES(11,”Pratik”),(12,”Somanath”);

**Where Clause:**

To define some condition

Syntax:

SELECT  *col1,col2*  FROM *table \_name* WHERE *condition;*

Eg.

SELECT \* FROM student WHERE city =”Akluj”;

Using Operators in WHERE:

**Arithmetic operators**: +,-,\*,/,%

**Comparison operators:**=,!=,>,<

**Logical operators:** AND, OR, NOT, IN, BETWEEN, ALL, LIKE, ANY

**Bitwise operators:**&,|

**Operators:**

**1)AND:** To check both conditions are to be true.

Eg.

SELECT \* FROM student WHERE marks>80 AND city=”Akluj ”;

**2)OR:** To check at least one condition to be true.

Eg.

SELECT \* FROM student WHERE mark >90 OR city=”Akluj”

3)**BETWEEN:** To select from a given range.

Eg.

SELECT \* FROM student WHERE marks BETWEEN 80 AND 90;

4)**IN:** To match any value in the list.

Eg.

SELECT \* FROM student WHERE city IN (“Akluj”,” Pandharpur”);

4)**NOT:** To neglect the given condition.

Eg.

SELECT \* FROM student WHERE city NOT IN(“Akluj”,”Pandharpur”);

**LIKE operator:**

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

The percent sign (%) represents zero, one, or multiple characters

The underscore sign (\_) represents one, single character

|  |  |
| --- | --- |
| LIKE Operator | Description |
| WHERE CustomerName LIKE 'a%' | Finds any values that start with "a" |
| WHERE CustomerName LIKE '%a' | Finds any values that end with "a" |
| WHERE CustomerName LIKE '%or%' | Finds any values that have "or" in any position |
| WHERE CustomerName LIKE '\_r%' | Finds any values that have "r" in the second position |
| WHERE CustomerName LIKE 'a\_%' | Finds any values that start with "a" and are at least 2 characters in length |
| WHERE CustomerName LIKE 'a\_\_%' | Finds any values that start with "a" and are at least 3 characters in length |
| WHERE ContactName LIKE 'a%o' | Finds any values that start with "a" and ends with "o" |

**Limit clause:**

Set an upper limit on the number of (tuples) rows to be returned.

Syntax:

SELECT \* FROM *table \_name* LIMIT  *number*;

Eg.

SELECT \* FROM student LIMIT 3;

**Order By Clause:**

To sort in ascending(ASC) or descending order(DESC)

Syntax:

SELECT *col1,col2* FROM table \_name

ORDER BY col \_name(s) ASE/DESC;

Eg.

SELECT \* FROM student ORDER BY city ASC;

**Aggregate Functions:**

Aggregate functions perform a calculation on a set of values and return a single value.

-COUNT()

-MAX()

-MIN()

-SUM()

-AVG()

**Get Maximum Marks:**

SELECT max(Marks) FROM student;

**Get Average marks:**

SELECT avg (marks) FROM student;

**Table related queries:**

**Update:**

**-**To update an existing row

**Syntax:**

UPDATE *table \_name*

SET *col1=val1,col2=val2*

WHERE *condition;*

**Eg.**

UPDATE student

SET grade=”0”

WHERE grade=” A”;

**Delete:**

**-**To delete existing rows

**Syntax:**

DELETE FROM *table \_name*

WHERE *condition;*

**Eg.**

DELETE FROM student

WHERE marks<33;

**Table Related Queries:**

**Alter:**

**ADD Column:**

ALTER TABLE  *table \_name*

ADD COLUMN *col \_name datatype constraint*;

Eg.

ALTER TABLE  *Student*

ADD COLUMN age int NOT NULL DEFAULT 19;

**Drop Column:**

ALTER TABLE  *table \_name*

DROP COLUMN c*ol \_name;*

*Eg.*

ALTER TABLE  *student*

DROP COLUMN stud \_age;

**RENAME Table**

ALTER TABLE *table \_name*

RENAME TO *new \_tabel \_name;*

Eg.

ALTER TABLE *student*

RENAME TO *stud;*

**CHANGE Column(rename):**

ALTER TABLE  *table­­­ \_name*

CHANGE COLUMN  *old \_name new \_name new \_datatype new \_constaint;*

Eg.

ALTER TABLE  *student*

CHANGE COLUMN *age stud \_age* INT*;*

**MODIFY Column(modify datatype/constraint):**

ALTER TABLE*table \_name*

MODIFY *col \_name new \_datatype new \_constraint;*

Eg.

ALTER TABLE*student*

MODIFY *age* VARCHAR (2)*;*

**Truncate:**

**-**To delete table data

TRUNCATE TABLE *table \_name;*