**Phase 2 :**

**Day 1 : 02-13-2023**

**Web Application**

**Database : MySQL : Self learning**

**JDBC : Java Database Connectivity**

**ORM (Object Relation Mapping ) Using Hibernate /JPA**

**Servlet**

**JSP : Java Server Pages**

**Please connect virtual lab**

**Open the terminal and type as**

**mysql -u root -p**

**Simplilearn**

**username is root**

**password is Simplilearn**

We can store the data permanently

1. Using file base system
2. Database system

Limitation of file base system

1. Data redundancy means same record we can store again and again
2. Data inconsistency : example : format of the file.

Employee.txt

Id,name,salary

1. To do CRUD operation in file complex task. Create, Read, Update and Delete.
2. Security

Database :

Data : it is a raw fact.

Information : processed data or meaningful data.

Database : it is use to store the data in table format using row and column.

DBMS : database management system : it is a software which help to store the data in table format.

RDBMS

Employee Record in excel sheet

Id Name Salary

100 John 12000

101 Lex 14000

100 Steven 16000

TrainerStudent

TId TName Tech Sid Sname Age

1 Raj Java 100 Steven 27

2 Raj Java 101 Lex 28

3 Raj Java 102 Neena 31

Relational database management system

Trainer

PK(primary key)

TId Tname tech

1 Raj Java

2 Ravi Python

Student

PK FK (Foreign Key)

Sid SName age TSId

100 Steven 24 1

101 Lex 25 1

102 Leeta 26 2

103 Reeta 28

MySQL, Oracle, DB2, Postgres, SQL Server 2022 etc are all RDBMS Database.

To interact with these all database we need to learn one English statement language ie SQL (Structured Query Language)

Syntax to view the database : plz login to database using username and password.

show database

use databasename; this command is use to switch to existing database.

show tables; this command is use to show all table present in that database.

Syntax to create the database.

create database databasename;

create database mydb;

we will create table with name as Employee

id,name,salary are column with

id as number without decimal

name as string value

salary as number with decimal

and id must unique PK

DDL (Data Definition Language)

create table employee(id int primary key,name varchar(10), salary float);

to view the table structure we need to run the command as

desc employee

DML (Data Manipulation Language)

insert into tablename values(v1,v2,v3);

insert into employee(1,’Ravi’,12000);

to view the records from database

select \* from tableName; \* means all columns

select \* from employee;

select columnname,columnname from tableName;

filter the records using where clause

select \* from employee where salary > 12000;

>, >=, <, <=, =, != (relational operator which we can use with number values)

Select \* from employee where name =’Raj’; single condition value

Select \* from employee where id in(1,4,8); condition with multiple value.

Select \* from employee where salary between 5000 and 10000 (range value)

Update query

Update tableName set columnName = value;

update employee set salary = 35000; in table all record salary will update with 35000

update employee set salary = 34000 where id=1;

update employee set salary = 32000 where salary > 25000;

update employee set name =’Ravi Kumar’ where name =’Ravi’;

Delete query

Delete from tablename; all record delete from table.

delete from employee;

delete from employee where id=1;

delete from employee where name =’Ravi’;

delete from employee where salary < 15000;

JDBC : Java Database Connectivity :

JDBC is a API (Application programming interface) which provided lot of pre defined classes and interfaces which help to connect any database like mysql or oracle using Java technology to do the operation like insert, delete, update and retrieve.

Steps to connect database using JDBC.

1. Import sql package. : Java provided classes and interfaces which help to connect the database and all those classes part of sql package.
2. Jdbc throw checked exception it SQLException. So while writing jdbc code inside main method or use defined method that method is responsible to handle to exception using try-catch or throws.
3. Load the driver : Driver is a pre fined class in the form of jar or exe format provided by vendor whose database we are going to connect.

4 types of driver

Type 1 : jdbc odbc bridge driver

Type 2 : jdbc native api driver

Type 3 : jdbc net protocol driver

Type 4 : jdbc pure or thin driver

From Java8 onward type1 driver removed.

Type 4 driver come in the form jar file. So we need to add that jar file base upon which database we are connecting. We can add manually or using build tool like maven or gradle.

jar file .java and .class file mainly for core java project

war file .java, .class, .html, .css, .xml, .js

ear file .java, .class, .html, .css, .xml, .js with ejb programs

to load the driver in java we need to write the code as

Class.forName(“driverName”);

Class is a pre defined the name itself is a Class.

This class contains forName static method which help to load the class.

1. Establish the connection :

DriverManager is a pre defined class which contains lot of pre defined method like getConnection() which takes three parameter 1st url, 2nd username, 3rd password.

DriverManager.getConnection(url,username,password);

This method return type is Connection interface reference

Connection con = DriverManager.getConnection(url,username,password);

1. After connected successfully we need to create Statement interface reference. Which provided set of methods which help to insert, delete, update and retrieve the records from database.

Statement stmt = con.createStatement();

stmt.executeUpdate(“insert/delete/update”)

and

ResultSet rs = stmt.executeQuery(“select clause”);

while(rs.next()){

System.out.println(“id is ”+rs.getInt(1)+” Name is ”+rs.getString(2)+” Salary is ”+rs.getFloat(3));

}

**Day 2 : 02-14-2023**

int temp = stmt.executeUpdate(“DML Query (Insert/Delete/Update)”);

the return type of excuteUpdate() method is int. if query executed successfully and base upon query how many records get updated that number of records stored in temp type of int variable.

stmt.executeQuery(“select query”); return type ResultSet like Iterator

stmt.executeUpdate(“insert/delete/update”); return type int if record insert, delete or update successfully then it return greater than zero else zero.

PreparedStatement :

Statement and PreparedStatement both are interfaces.

Statement is use to execute static query and PreparedStatement is use to execute dynamic query with help of parameterized query concept.

If we use Statement reference whenever we execute the program each time query compile java side, then query send to database and execute and get the acknowledgement. But if we PreparedStatement query compile only once and execute again and again.

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(“select query);

int res = stmt.executeUpdate(“insert/delete/update);

PreparedStatement pstmt = con.prepareStatement(“query”);

Query can be insert, delete, update or retrieve.

ResultSet rs = pstmt.executeQuery();

int temp = rs.executeUpdate();

Database :mydb

Employee -🡪 id, name, salary (columns)

In Database table must be map with Java class that class as of now is JavaBean class.

JavaBean class is normal class with few rules we need to follow.

1. class must be public
2. All variable inside JavaBean class must be private
3. For each variable we need to write a method ie setter and getter
4. Setter method is use set the value and getter method is use to get the value.
5. Set method name is set followed by variable and getter method name is get followed by variableName.

public class Employee {

private int id;

private String name;

public void setId(int id) {

this.id = id;

}

public void setName(String name) {

this.name =name;

}

public int getId() {

return this.id;

}

public int getName() {

return name;

}

}

According to standard

In Database table

In Java JavaBean must map

Table JavaBean

Employee Employee

ID,NAME,SALARY id,name,salary

Setter and getter methods

Map

On JavaBean class we will do some operation ie service or business logic that class is known as service class. Service class contains pure business logic.

Service class name must be JavaBean class name followed by Serice

EmployeeService

CustomerService

AccountService

ProductService

Dao class (Data Access Object ) : it is a type of normal class which contains database logic written using Jdbc or ORM.

This class contains pure database logic.

EmployeeDao

CustomerDao

AccountDao

ProductDao

Inside DAO or service class don’t create Scanner class object or don’t ask the value through keyboard.

Dao and service layer not responsible to interact with input device.

In Service layer we can write business logic

before calling DAO method.

After calling DAO method

Before as well as After DAO method

Not Service logic