**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

Day 3 : 15-02-2023

Maven :

Maven is a build tool. Build tool mean the tool is responsible to compile the program, run program, creating jar or war file, help to download the dependencies base upon the project.

Maven it a tool which provide standard project structure which follow by All IDE.

Creating maven project in Eclipse IDE.

In Eclipse IDE by default maven created Java version 1.5 (old version).

And maven provided pom.xml file (Project Object Model). This file hold all configuration details for our maven project.

ORM : Object Relation Mapping :

ORM is a concept like OOPs which help to store, delete, update and retrieve data in the form of object from database.

Limitation of JDBC.

1. Using JDBC we can’t store java object as well as we can’t retrieve java object from database. In DAO layer we need convert object into sql format as will retrieving record from database we will get in string format with the help of sql which we need to convert object.
2. JDBC use SQL language to interact with database. SQL is database dependent query language. If we move from mysql or oracle or db 2 or sql server we need to change our query.
3. With the help of JDBC it is not mandatory in Java Side we need JavaBean class object. JDBC doesn’t follow standard.
4. JDBC doesn’t allow relationship like is a relationship(inheritance) as well as has relationship(inside one class we are creating another class object).

ORM : Object Relation Mapping

In ORM JavaBean class is known as entity class.

In RDBMS table is known as Relation

Programming side

class Employee { Employee -🡪 Table

id,name,salary ID,NAME,SALARY(columns)

variable

}

Object in programming side and relation in database side and we are proding mapping

Using xml

Using annotation

Employee (class/entity class)🡨---🡪 EMPLOYEE(TABLE)

Id 🡪ID (column) datatypes PK

Name,🡪Name (column)

Salary🡪Salary(column )

The implementation of ORM according to Java is

Hibernate

JPA : Java Persistence API (Application Programming interface).

iBaties

Hibernate is tool provided by jboss. Which help to achieve orm using Java technologies. Hibernate is an open source tool.

By default eclipse IDE doesn’t provide hibernate plugin. So we need to download those plugins.

In Database Employee table (present)

Now we need create entity class (javabean) with few annotation

@Entity : This annotation we will write on class level to make normal java bean class as entity class.

@Id : This annotation we have to write on that property which property consider as primary key.

**Day 4 : 02-16-2023**

**Now we need to create configuration file. This file hold the details about database connection.**

**In JDBC we are written the code as**

**Class.forName(“driverName”);**

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

**In same in hibernate we can write in different way**

1. **Using java class**
2. **Using xml file**
3. **Using properties file**

**In Hibernate we will create hibernate.cfg.xml file. This file hold database connection details. Like driverName,url,username,password + entity class details + hibernate properties (optional)**

**We need to load the hibernate.cgf.xml file. To this xml file hibernate provided Configuration class.**

After file loaded successfully we need to create SessionFactory interface reference. This SessionFactory interface is like a Connection in jdbc.

After sessionfactory created now we need to create Session object. Session is like as Statement and PreparedStatement. Session reference provided method which help to store, delete, update and retrieve entity object.

By default we if store any record using jdbc in mysql database. By default it apply default commit. (using transactional concept).

But if we do same thing using ORM like Hibernate or JPA. We need to commit manually using transactional concept.

TCL : Transactional control language.

Whenever we do any DML Operation like Insert, delete, and Update if all query execute successfully we can say commit if any query go wrong we can say rollback.

Update account set amount = amount – 100 where accno=123;

Update account set amount = amount + 100 where accno=567;

If in any query error generate we have to say rollback if everything execute successfully we have to say commit.

Whenever we display reference of any class using println internally it will call toString() method of object class. that class method return type is string. It will give the output as [packageName.className@code](mailto:packageName.className@code)

By default every java class extends Object class. So if you want to property output we have to override tostring method.

Retrieve more than one records

Hibernate provided their own query language ie HQL (hibernate query language)

SQL : structured query language : it is database dependent query language. It will give the data in string format.

select \* from employee (employee is table name and \* means all column). Generally in database table is not a case sensitive.

Select \* from employee where id=2; id is column name

Select name from employee

HQL : hibernate query language : it is dataset independent query language. It will give data in object format.

select e from Employee e (Employee is a entity class name and it is case sensitive and e is reference or object. E hold all property of entity class).

Select e from Employee e where e.id = 2; e is employee object and id is variable name

Select e.name from Employee e : e is object and name is property or variable name

Employees first\_name 10 column

Department department\_name 5 column

Location city 7 column

To get these information we need to use join concept.

**Day 5 : 02-17-2023**

**@GeneratedValue this annotation use to create auto\_increment number for pid.**

**Database 🡪 Table (Product / Employee)**

**In JDBC table Employee must be map to JavaBean class Employee.**

**Hibernate Relationship**

**Hibernate support is a (inheritance) as well as has (association or aggregation or composition) relationship**

**In RDBMS we can make the relationship using primary and foreign key.**

**Trainer (one) ----🡪 Students (many) relationship**

**Trainer (one)-----🡪Course (many) relationship**

**Trainer**

**PK**

**TId TName tech**

1 Raj Java

2 Ravi Python

**create table trainer(tid int primary key,tname varchar(10), tech varchar(10));**

**tid is primary key, doesn’t allow duplicate and doesn’t allow null value.**

**Student**

**PK FK**

SID SName Age TSID

100 Reeta 21 1

101 Meeta 22 1

102 Leeta 23 2

**create table student(sid int primary key,sname varchar(10), age int , tsid int, foreign key(tsid) references trainer(tid));**

**sid primary key, tsid is foreign key tsid id allow only those values which present in primary key trainer table it can allow duplicate as well as allow null value.**

**Course table**

**Day 6: 02-20-2023**

**JavaEE or JEE : Java Enterprise Edition**

**https://**[**www.google.com**](http://www.google.com) **🡪 URL (uniform resource locator)**

**http : protocol : hyper text transfer protocol : set of rules which help to communication more than one machine**

**www: word wide web**

**google : domain name**

**com : commercial**

**req(http/https)-----🡪**

**Client Server**

**🡨--res(http/https)**

**Html/html5**

**Css/css3**

**JavaScript**

**JEE**

**Servlet, JSP and EJB**

**JSP :Java Server Page**

**Enterprise Java Bean**

**Asp.net**

**Php**

**Python**

**Node Js**

**In Servlet, JSP and EJB no main method. After created servlet, jsp and ejb application we need to deploy this application in server.**

**Server is responsible to deploy the application like it load class, it will create the object, it will life cycle methods, it destroy the object. The execution of server side technologies is taken care by server.**

**According to application point of view server are divided into two types.**

1. **Web server**
2. **Application server**

**Container : Container is a part of server which is also known as engine. Container is responsible to take care the execution of server side technologies like Servlet, Jsp and EJB.**

**If server is type of web server which contains only one type of container ie web container and this container is responsible to execute servlet and jsp.**

**If server is type of application server which contains more than one type of container ie web container, ejb container, jms container. Web container is responsible to execute servlet and jsp and ejb container is responsible to execute ejb.**

**Application container provide more extra features like connection pooling, thread management, resource management, security etc.**

**Web server : tomcat part of apache company**

**Application server: web logic, jboss, glashfish etc.**

**Open source server tomcat :**

**Servlet : Servlet is a normal java program which help to create dynamic web page on server side.**

**servlet : servlet is a package part of javax root package. That package lot of classes and interfaces.**

**javax.servlet.\*;**

**Servlet : Servlet is a interface part of servlet package.**

**javax.servlet.Servlet;**

**this interface contains five methods**

**init only once at beginning**

**service call again and again when client send request**

**destroy life cycle method at last when application close.**

**getServletInfo**

**getServletConfig**

**public class Demo implements Servlet {**

**we need to override all five methods mandatory.**

**}**

**GenericServlet : it is a type of abstract class which internally implement Servlet interface and provided body for all abstract method except service method.**

**public class Demo extends GenericServlet {**

**we need to provide the body for only service method.**

**}**

**HttpServlet : it is a type of abstract class which internally extends GenericServlet and provided body for service method as well as provided some extra method in the form of doXXX like doGet, doPost, doPut, doDelete etc**

**public class Demo extends HttpServlet {**

**service (doGet or doGet()) : doGet or doPost internally call service method only.**

**}**

**DD : Deployment descriptor file**

[**http://localhost:portNumber/ProjectName/URLpattern**](http://localhost:portNumber/ProjectName/URLpattern) **: now client sending to request to server**

**in html default method is get and it will call server side technologies doGet method.**

**If method is get it call doGet method and information will send through url using URL Rewrite technique ie URL?key=value&key=value&key=value**

**Get method is not secure. Using doGet we can send max 255 character data.**

**If we want to achieve security we have to make method as post. If method is post it will call server service technologies doPost and if method is post data send through body part of request. Performance wise post method is slower than get method. In post method data will send to request body part.**

**Day 7 : 02-21-2023**

**Navigating from one page to another using Servlet and JSP.**

**RequestDispatcher: It is a interface which provide set of method which help to move or navigate from one page to another page with condition.**

**Syntax to create reference of RequestDispatcher.**

**RequestDispatcher rd1 = request.getRequestDispacher(“path”);**

**Path : if target page is servlet then path must target servlet page URL pattern**

**If target page is html or jsp. Then path must be pageName.html or pageName.jsp.**

**rd1.forward(request,response); if we use forward we can see the output of only target page.**

**or**

**rd1.include(request,response); if we use include we can see the output of source page as well as target page combine as one output.**

**Login details checking from database using JDBC (SignIn and SignUp)**

**If we are planning connect the database using Servlet we need to copy and paste jar file inside lib folder (that folder sub older WEB-INF folder).**

**create table login(emailid varchar(30) primary key,password varchar(30));**

**JSP : Java Server Pages :**

**JSP is a tag base object oriented scripting language which help to create the web page on server side.**

**Limitation of Servlet**

1. **Servlet is normal java program if we do any changes we need to re compile and re deploy the program.**
2. **Servlet is complex in sense. If we want to display simple message through servlet. We need to create normal class and that class must be extends or implement type of Servlet. Then we need to override method ie doGet or doPost. Then we need to create PrintWriter class object and we need to provide servlet configuration details in web.xml file or using annotation.**
3. **If we want to write any presentation logic ie html code in servlet we need to write inside pw.println(“<h1>Welcome </h1>”);**

**To write html code inside println is more complex because it consider as a string.**

**JSP tags**

1. **Scripting tag** 
   1. **Scriplet tag**

**Syntax**

**<% opening tag**

**We can write java code or the code which we**

**Write inside doGet or doPost**

**%> closing tag**

* 1. **Declarative tag**

**<%!**

**We can create instance variable**

**%>**

* 1. **Expression tag**

**<%=**

**Expression**

**%>**

* 1. **<%-- --%> comment in JSP**

1. **Implicit object** 
   1. **out : in JSP out is implicit object. It a reference of PrintWriter class. In Servlet we were creating this reference like**

**PrintWriter pw = response.getWriter();**

**But in out is like pw we can’t change the name.**

* 1. **request : it is consider as HttpServletRequest interface reference.**
  2. **response : it is consider as HttpServletResponse interface reference.**

1. **Action tags** 
   1. **Jsp include**
   2. **Jsp forward**
2. **Jsp directive tags**