**Phase 2 :** Become a back-end expert

**9 days**

**Day 1 : 20-06-2022**

**JDBC**

**Hibernate : These two modules is use to connect the databases.**

**We are going to use MySQL Database.**

**Servlet**

**JSP : These two modules is use to create the web application using Java technologies.**

**In Virtual Lab to connect the database**

**mysql –u root –p**

**password : Simplilearn**

**MySQL is open source and type of RDBMS (Relational Database Management System).**

**Syntax to create the database**

**create database databaseName**

**create database myjavadb; This command is use to create the database**

**show databases; This command is use to view all databases present in your account**

**use myjavadb; using this command we can switch from one database to another database.**

**Syntax to create the table**

**create table tableName(columnName datatype, columnName datatype)**

**Employee**

**Id Name Salary**

**int varchar(10) float**

**Primary key**

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

**show tables : this command is use to show all table present in that database**

**desc employee : This command is use to show table structure**

**Syntax to insert the records in table**

**insert into tableName values(v1,v2,v3);**

**insert into employee values(1,’Ravi’,15000);**

**View the records from a table**

**Select \* from tableName**

**select \* from employee**

**select columnName,columnName from tableName**

**Filter the records with conditions (where clause)**

1. **select \* from employee where salary >= 15000;**
2. **select \* from employee where id =1;**
3. **select \* from employee where name = 'Ravi';**
4. **select \* from employee where name like 'Ravi';**
5. **select \* from employee where salary between 15000 and 18000;**
6. **select \* from employee where id in (1,5)**

**update the records with conditions**

**syntax**

**update tableName set columnName = value where columnName = value**

1. **update employee set salary = 25000**
2. **update employee set salary = 22000 where id =1;**
3. **update employee set salary = 1800 where name = ‘Ramesh’;**
4. **update employee set name = ‘Ravi Kumar ‘ where name =’Ravi’;**

**delete records**

**syntax**

**delete from tableName**

1. **delete from employee;**
2. **delete from employee where id=1;**
3. **delete from employee where name =’Ravi’;**
4. **delete from employee where salary > 12000;**

**Database It is use to store the data in table format.**

**Java mysql**

**JDBC**

**JDBC : Java Database Connectivity :**

**JDBC is an API (Application Programming Interface) which provide lot of pre-defined classes and interface which help to connect the any RDMBS database(mysql, oracle, postgres sql, db2 or sql server) to store, retrieve, update and delete the records using Java technologies.**

**Steps to connect the database through Java technologies.**

1. **Import the package sql.**
2. **Jdbc always throw checked exception so we have to write the method it may be main method or user defined with exception handling concept ie try-catch or throws exception concept.**
3. **Load the Driver : Driver is a pre-defined class provided by vendor (database) which help to connect the database.**

**Class.forName(“driverName”);**

**Class is a pre-defined class name itself is a Class and it is a part of lang package. Which contains forName is a static method. Which help to load the class.**

**JDBC provided totally 4 types of driver.**

1. **Jdbc odbc bridge driver : from java8 onward this type of driver removed.**
2. **Jdbc native api driver**
3. **Jdbc net protocol driver**
4. **Jdbc thin or pure driver.**

**We are using type 4 ie jdbc thin driver.**

**For MySQL Database**

**com.mysql.jdbc.Driver -🡪 if database is 5.x version**

**com.mysql.cj.jdbc.Driver 🡪 if database is 8.x version**

**We have to download mysql connector jar which contains Driver pre-defined class which help to connect the database.**

1. **Establish the connection : DriverManager is a pre-defined class which contains getConnection is a static method which takes 3 parameter url, username and password. This method return type is Connection interface reference.**

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

**Day 2 : 21-06-2022**

1. **We have to create types of Statement**
2. **Statement**
3. **PreparedStatement**

**Both are interfaces which provide set of method which help to do the operation on table like insert, delete, update and retrieve.**

**Statement : syntax to create the Statement interface reference.**

**Statement stmt = con.createStatement();**

1. **Insert operation :**

**int res = stmt.executeUpdate(“insert query”);**

**executeUpdate method return type is int. if record inserted successfully it will return 1 or it will generate exception.**

1. **Delete operation**

**int res = stmt.executeUpdate(“delete query”)**

1. **Update operation**

**int res = stmt.executeUpdate(“update query”)**

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

**executeUpdate() method return type is int.**

**executeQuery() method return type is ResultSet. ResultSet is a interface which provide method to retrieve the records one by one.**

**Statement is use to do static query. PreparedStatement is use to dynamic query. PreparedStatement support parameterized query concept. Using this concept we can pass the dynamic value. PreparedStatement is faster than Statement Because in Statement whenever we run the program each time program get compile in java side and execute in database side again and again. Using PreparedStatement query compile only once and execute n number of times. Meaning of the PreparedStatement is pre compiled query.**

1. **Close the resource**

**stmt.close();**

**con.close();**

**We will do CRUD Operation using Standard**

**Maven : maven is known as build tool, which is responsible to compile, run, creating, jar or war file, download the dependencies.**

**In maven project we can see the pom.xml file project object model . This file hold project description details.**

**While developing the application using JDBC table must be map to JavaBean class.**

**Table -🡪 JavaBean**

**Dao (Data Access Object ) : This is a type of class which contains pure database logic.**

**Service layer : This class is responsible to write the pure business logic.**

**Day 3**

**22-06-2022**

**Limitation of JDBC**

1. **Using JDBC we can’t store as well as can’t retrieve java object from database. We have to convert Java object into sql query and vice-versa.**
2. **JDBC use SQL language. SQL is database query language. If we move from mysql database to oracle or sql server or db2 we have to change the query.**

**Because sql is database dependent query language.**

1. **JDBC throw checked exception so while writing jdbc code we have to use try-catch block mandatory. And all exception hierarchy is database dependent.**
2. **JDBC doesn’t support is a and has a relationship.**

**ORM : Object Relation mapping**

**Object – on programming side**

**Relation 🡪 database**

**Mapping**

**class Employee { Table Employee**

**id, name, salary ID,NAME,SALARY**

**}**

**Mapping**

**Employee ---- EMPLOYEE**

**id --- ID PK**

**name --- NAME**

**salary --- SALARY**

**we can do mapping using**

**using xml file**

**using annotation**

**configuration details**

**we are writing using**

**using xml file**

**using properties file**

**using java classes**

**This file provide the details about driverName, url, username, password and dialect class(this class is responsible to convert java object to sql and vice-versa).**

**ORM tools**

1. **Hibernate**
2. **JPA(Java Persistence API)**
3. **iBaties**
4. **JDO**

**Hibernate : Hibernate is a open source framework. Which implements the ORM features. Hibernate is a part of JBoss.**

**Using Hibernate we are going to improve DAO layer.**

**Table -🡪 Employee 🡪 Id(PK), name, salary**

**In ORM JavaBean class is known as entity class.**

**Through JDBC if we do any DML (Data Manipulation Language ie insert delete and update). By default auto commit.**

**In My SQL by default all DML operation are auto commit.**

**In Oracle database all DML operation not auto commit.**

**If we do any operation through ORM like Hibernate by default all DML operation not auto commit. So we have to use transaction concept mandatory.**

**Hibernate use its own query language ie HQL (Hibernate Query Language).**

**SQL : Structured Query language :**

**Database Dependent**

**select \* from employee (employee is a table name and it is not a case sensitive). \* means we are retrieve all columns**

**select \* from employee where id = 100;**

**select \* from employee where name like ‘Ramesh’;**

**select \* from employee where salary > 12000;**

**HQL : Hibernate Query language**

**Database independent**

**select emp from Employee emp (Employee is a entity class name which is case sensitive and emp is a object). emp means object we are retrieving all property.**

**Select emp from Employee emp where emp.id =100;**

**Select emp from Employee emp where emp.name like ‘Ramesh’;**

**Select emp from Employee emp where emp.salary > 12000**

**Day 5 : 24-06-2022**

**Hibernate Relationship :**

**Hibernate support is a as well as has a relationship.**

**Is a relationship : Inheritance**

**Has a relationship : inside one class creating the object of another class.**

**Has a relationship :**

**4 types**

1. **One to many trainer to students**
2. **Many to one employees to department/project**
3. **One to one person to passport**
4. **Many to many students to technologies**

**TrainerAndStudents**

**TId TName tech Sid SName age**

**1 Raj Java 100 Seeta 21**

**1 Raj Java 101 Meeta 22**

**1 Raj Java 103 Veeta 23**

**Data redundancy (duplicate records)**

**Trainer**

**PK**

**TId TName tech**

**1 Raj Java**

**2 Ravi Python**

**Students**

**Pk FK (Foreign Key)**

**SId SName age TSId**

**100 Seeta 21 1**

**101 Meeta 22 1**

**102 Veeta 23 2**

**103 Leeta 23 2**

**Foreign key : it use to connect the primary key. If the column foreign key it allow only those value which present in primary key. It can allow duplicate.**

**JPA : Java persistence API JPA is a specification.**

**Hibernate : Hibernate is a implementation.**

**We will create the Trainer and Student tables with Primary key and foreign key.**

**create table trainer(**

**tid int primary key,**

**tname varchar(25),**

**tech varchar(25));**

**create table student(**

**sid int primary key,**

**sname varchar(25),**

**age int, tsid int,**

**foreign key(tsid) references trainer(tid));**

**SQL Join : Join is use to retrieve the more than one column from more than one table with or without conditions.**

**We can do Join using SQL or HQL.**

**Many to Many relationship**

**Students**

**Sid SName**

1. **Ravi**
2. **Ramesh**
3. **Rajesh**

**Technologies**

**TId Technologies**

**100 Java**

**101 Python**

**102 Angular**

**Third table which maintain the relationship**

**srno Sid TId**

**1 1 100**

**2 1 101**

**3 1 102**

**4 2 100**

**5 2 101**

**Day 6**

**27-06-2022**

**JavaEE or J2EE or JEE**

**Java Enterprise Edition**

**Servlet, JSP (Java Server Pages) and EJB (Enterprise Java Bean).**

**Using JEE module we can create the dynamic we application using Java technologies.**

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

**http: hypertext transfer protocol and s secure.**

**www : word wide web**

**google : domain**

**com : commercial**

**req (http/https)---🡪**

**Client Server**

**🡨----Res (http/https)**

**HTML/HTML5**

**CSS/CSS3**

**JavaScript**

**JEE**

**Servlet, jsp or ejb**

**Asp.net**

**Php**

**Python**

**Node JS**

**To run the server side technologies we require server.**

**Two types of server**

1. **Web server : tomcat (apache company)**
2. **Application server : web logic, jboss, glashfish etc.**

**No main method in servlet, jsp or ejb. Server contains container. Container is a part of Server which is also known as engine. Container is responsible to run the servlet, jsp and ejb application like load the class, create the object, call the life cycle method, destroy the object.**

**If server is a type of web server it contains only one type of container ie web container. Web Container is responsible to execute servlet and jsp not ejb.**

**If server is a type of application server it contain different type of container like web container, ejb container, jms container. Web container is responsible to execute servlet and jsp and ejb container is responsible to execute ejb program. Application server provide some extra services like connection pooling, thread management, resource management, security etc.**

**We are going to run servlet and jsp program using web server ie tomcat.**

**Default port number for tomcat is 8080**

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

**import javax.servlet.\*;**

**servlet is a package which contains set of classes and interfaces.**

**import javax.servlet.Servlet;**

**Servlet is a interface which contains set of methods.**

**This interface contains 5 method**

**init**

**service**

**destroy 3 life cycle methods.**

**getServletInfo**

**getServletConfig**

**1st approach to create the servlet program**

**class MyDemo implements Servlet {**

**we have to override all five methods.**

**}**

**GenericServlet : it is a pre-defined abstract class which internally implements Servlet interface. This class provided body for all methods except service method.**

**2nd approach to create the servlet program**

**class MyDemo extends GenericServlet {**

**here we have to override only service method.**

**}**

**HttpServlet : it is a type of abstract class which internally extends GenericServlet class. This class provided body for service method and provide few extra method in the form of doXXX like doGet, doPost etc. doGet or doPost method internally call service method only.**

**3rd approach to create the Servlet program**

**class MyDemo extends GenericServlet {**

**service() or doGet() or doPost()**

**}**

**Once you deploy the application on server**

**Now we have access this application using browser.**

[**http://localhost:8080/SimpleServletApplication/Demo**](http://localhost:8080/SimpleServletApplication/Demo)

**By default every html form method consider as a get. If method is get it will call doGet method. If method is get it will send the information through URL using Query param concept. Like**

**URL?key1=value&key2=value&key=value**

**Get method is not a secure and we can send maximum 255 character data through get method.**

**If we want to make secure we can use post method. If method is post then it will call doPost of servlet program. In post method data will send through body part of request. We can send more than 255 character data through post method.**

**Day 6 :**

**28-06-2022**

**RequestDispatcher : it is a interface which provide set of method which help to navigate from one page to another page (servlet or jsp or html) base upon our requirements.**

**Syntax to create the RequestDispatcher interface reference.**

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

**If target page is servlet then path must be url pattern of servlet page.**

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

**rd1.forward(request,response);**

**we can see output of only target page.**

**Only target page**

**or**

**rd1.include(request,response);**

**we can see output of current page as well as target page.**

**Source + target page**

**Limitation of Servlet**

1. **Servlet is normal java program if we do any changed we have recompile and redeploy the application once again.**
2. **Inside a servlet if we want to write any html code it must be in double quotation. Html code embedded inside a servlet.**
3. **Servlet is a complex. If we want to write simple message through servlet. We have to make normal class that class must be extends or implements type of servlet. Then we have to override life cycle method. Then create printwriter class object and we have to provide the servlet details in web.xml file.**

**JSP : Java Server Pages : JSP is tag base interpreter scripting language which help to create dynamic web page on server side.**

**JSP provided lot of pre-defined tags**

1. **Scripting tag** 
   1. **Scriptlet tag**

**<%**

**Java coding or the code which we write inside a main method. The code which we write inside doGet or doPost**

**%>**

* 1. **Declarative tag**

**<%!**

**Variable declaration**

**%>**

* 1. **Expression tag**

**<%=**

**%>**

* 1. **Comment tags**

**<%-- --%>**

1. **Implicit object** 
   1. **out : it is pre-defined implicit object which is consider as PrintWriter class object.**
   2. **request : it is equal to HttpServletReqeust object.**
   3. **response : it is equal to HttpServletResponse object.**
2. **Action tags**
   1. **Jsp forward**
   2. **Jsp include**
3. **Directive tags**

**Limitation of JSP**

1. **JSP is also type of servlet when we run the jsp program on browser internally it will convert into servlet. This phase is known as page translation. So performance wise jsp is slower than servlet.**
2. **Jsp is not secure. If we any business logic or database logic in jsp. When we deliver this application to client. Client can see our code very easily. But if we write the code in java class ie may be servlet or normal java class we can provide only .class file not source code.**
3. **In JSP we can do re-usability of code. We can include or forward whole jsp page but not a part of jsp code. Because in jsp we can’t write. But if we write the code in java classes that code we can do re-usability.**

**While developing any web application we develop the application using MVC architecture.**

**MVC :**

**Model View Controller**

**View : Presentation Logic : HTML and JSP -🡪**

**Console replace by browser.**

**Controller -🡪 Servlet**

**Main class replace by Servlet**

**Model -🡪 JavaBean class : map to table**

**Service class : pure business logic**

**Dao class : pure data base logic**

**Using jdbc or hibernate**

**Resource class : database connectivity**