**Phase 2**

JDBC : Java Database Connectivity :

Class.forName(“driverName”)

4 types of driver

Type 1 or jdbc odbc bridge driver : from java 8 onward removed

Type 2 or jdbc native api driver

Type 3 or jdbc net protocol driver

Type 4 or jdbc thin or pure driver

com.mysql.cj.jdbc.Driver type 4 for mysql database

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

Statement stmt = con.createStatement();

int res = stmt.executeUpdate(“DML Operation”)

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

Using while loop retrieve the records one by one

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

PreparedStatement pstmt = con.prepareStatement(“insert into employee values(?,?,?)”);

pstmt.setIt(1,100);

pstmt.setString(2,”Ravi”);

pstmt.setFloat(3,12000.50);

int res = pstmt.executeUpdate();

Limitation of JDBC :

1. Using JDBC we can’t store and retrieve object directly. (DAO Layer) : Data Access object : In this class we can write pure jdbc code.

In DAO layer we need to convert Java object into sql query or vice-versa.

1. JDBC use SQL and SQL is database dependent.
2. JDBC doesn’t support is a and has relationship. Inheritance and has a relationship (one to one or one to many or many to one or many to many)

ORM : Object relationship mapping.

@Entity

@Table(name=”employee”)

class Employee { (entity class) table

@Id Employee

Id,

@Column(name=”empInfo)

name,salary ID,Name,Salary

}

Mapping

Using xml file

Using annotation

In ORM

We can write database details using

1. Using normal java classes
2. Using xml
3. Using properties file

hibernate.cfg.xml or properties file

we need to provide database details like driver name, url, username, password, mapping classes details(classes with annotation).

Hibernate : Hibernate is one of the implementation of ORM.

JPA : Spring framework with Hibernate or JPA.

iBaties

more

Configuration con = new Configuration();

con.configure(“hibernate.cfg.xml”);

SessionFactory sf = con.buildSessionFactory(); // Connection con;

Session session = sf.openSession(); //PreparedStatement

Transaction tran = session.getTransaction();

Employee emp =new Employee();

emp.setId(100);

emp.setName(“Ravi”);

emp.setSalary(12000);

tran.begin();

session.save(emp); // insert query

tran.commit();

Employee e = session.get(Employee.class,1);

If(e==null) {

System.out.println(“Record not present”);

}else {

Tran.begin

Session.delete(e);

Tran.commit();

}

Employee e = session.get(Employee.class,1);

If(e==null) {

System.out.println(“Record not present”);

}else {

Tran.begin

e.setSalary(45000);

Session.update(e);

Tran.commit();

}

We required server to run servlet and jsp program.

Web server tomcat ,

Application server web logic, jboss, WAS etc.

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

Servlet interface : five methods

Init, service, destroy, getServletInfo, getServletConfig

GenericServlet : this class internally implements Servlet interface and provided body for all methods except service method.

HttpServlet : this class internally extends GenericServlet and provided some extra methods in the form of doXXX like doGet or doPost

Class MyServletDemo implements Servlet {

We need to override all five methods.

}

Class MyServletDemo extends GenericServlet {

We need to override only service method.

}

Class MyServletDemo extends HttpServlet {

doGet(req,res),

PrintWriter pw = res.getWriter();

String name = req.getParameter(“name”);

RequestDispatcher rd = req.getRequestDispatcher(“Home”)

rd.include(req,res);

or

rd.forward(req,res);

doPost()

}

Session Tracking : session is a collection of request and response within a particular period of time is known as session.

Cookies : cookies is small text file created by server which client send first request to server.

URLRewriting

HttpSession HttpSesssion hs = request.getSession();

Hidden Form field

JSP

View : JSP and HTML

Servlet (Controller )

Service and DAO

DAO layer using JDBC or Hibernate

Index.jsp

Source

Destination

Date of Travel

Number of Ticket

Please display all plane details.

Book

Book

PlainTable : Pid, Source, Destination , Price , NumberOrTicket etc

Once we click on book button or hyper link etc

Then display another page

Plane name, source, destination, price, number of ticket

FirstName,LastName,age, PhoneNumber

Confirm for the booking

Credit card number

Pin Number

Your ticket booked successfully.

[admin@gmail.com](mailto:admin@gmail.com) admin@123

**Admin for Learner’s Academy.**

**Master list for subject**

**Master list for class**

**Master list of teacher**

**SID**

**Subject**

**ClassId**

**classSection**

**TId**

**TeacherName**

**Teacher going to assign the class : one to many**

**Teacher going to take the subject : one to many**

**Student to class : one to one and one to many for subject**

**1st link to add teacher details.**

**2nd link to add subject details**

**3rd link to add class details**

**4th link to view all student details**

**5th link to view all class details**

**6th link to view all teacher details.**