**Hibernate**

1) create the **Dynamic web project**

2) Download mysql and hibernate jar file and configure them to program

OR

**convert our project into Maven**

i) Right click on project name ---> configure ---> convert to maven Project---->

Finish

this will create a file **pom.xml**

in this file we write all dependencies for jar file.

**Maven have 3 Repository :**

1) Local repository : - c:\user\..\m2\repository---> online download hone ke

baad yhi save hota hai hmara dependencies.

2)central repository : - Jha se hum log download kar rhe hai net se

3)remote repository : - if i want to download it by remotely .

AGAR **POM.XML** ME FIRST LINE ME ERROR AARHA HAI TO WINDOWS--> PREFERENCE-->MAVEN--> SELECT THE DOWNLOAD ARTIFACT JAVADOC. AND APPLY AND CLOSE.

**We have to download the dependences which where required**

**search in google :** -> MySQL maven dependences

open the Maven Repository

And download the dependence

which where required for it

1) select the version of maven

2)copy the maven code and paste it into pom.xml

3)Pom.xml file store the Dependencies of a project

4)pom->Project object Model

<packaging>war</packaging>// iske niche

**<dependencies>**

yha paste karna hai ye

dependencies ka tag bnake

uske ander paste karna hai

**</dependencies>**

**pom.xml :**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>hibernate\_Project</groupId>

<artifactId>hibernate\_Project</artifactId>

<version>0.0.1-SNAPSHOT</version>

**<packaging>war</packaging>**

**<dependencies>**

*<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->*

***<dependency>***

***<groupId>mysql</groupId>***

***<artifactId>mysql-connector-java</artifactId>***

***<version>8.0.33</version>***

***</dependency>***

*<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->*

***<dependency>***

***<groupId>org.hibernate</groupId>***

***<artifactId>hibernate-core</artifactId>***

***<version>5.6.15.Final</version>***

***</dependency>***

*<!-- https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper -->*

***<dependency>***

***<groupId>org.apache.tomcat.embed</groupId>***

***<artifactId>tomcat-embed-jasper</artifactId>***

***<version>10.1.1</version>***

***</dependency>***

**</dependencies>**

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<release>21</release>

</configuration>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.3</version>

</plugin>

</plugins>

</build>

</project>

**ii) iske baad hme hibernate.cgf.xml file bnana hai**

src--> main--> java folder per right click kro aur new --> other-->xml File-->

next-->

name that file as hibernate.cfg.xml and next --> finish .

this file store the database information.

**Hibernate.cgf.xml file :-**

<?xml version="1.0" encoding="UTF-8"?>

<hibernate-configuration>

**<session-factory>**

**<property name="dialect">org.hibernate.dialect.MySQLDialect</property>**

**<property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver**

**</property>**

**<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hms**

**</property>**

**<property**

**name="hibernate.connection.username">root</property>**

**<property name="hibernate.connection.password">Harshit@4141</property>**

**<mapping class="com.entity1.Adminlogin"/>** // yha hum map kar rhe hai

adminlogin class se

**</session-factory>**//com.entity--> package name hai aur adminlogin -->class

name hai

</hibernate-configuration>

*// isse database se connection hogya hai ........*

**iii) iske baad hme aab entity class bnana hoga jisko hum POJO file bolte hai**

src --> main --> java per right click krenge aur ek class bnayenge and package

name is entity and class name is same as the table name

**package com.Entity**;

import javax.persistence.\*;

**@Entity**

@Table**(name="adminlogin")//** agar table ka name aur class ka name same nhi

hoga tab hum isko use krenge ye name hi table ka name hai

public class Adminlogin {

**@Id**

int id;// ye primary key hai iss liye hum iske uper @id likhe hai

**@Column(name="user")**

String user;

**@Column(name="pass")**// agar column name aur table ke column ke name

same nhi hoga tab lgayenge hum ye

String pass;

public int getId()

return id;

}

public void setId(int id) {

this.id = id;

}

public String getUser() {

return user;

}

public void setUser(String user) {

this.user = user;

}

public String getPass() {

return pass;

}

public void setPass(String pass) {

this.pass = pass;

}

}

**Now create the JSP page**

webapp-->new --> jsp file --> name the file and next--> finish;

file name = > viewadmindetail.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@page import="org.hibernate.\*" %>

<%@page import="org.hibernate.cfg.\*" %>

<%@page import="java.util.\*" %>

**<%@page import="com.entity1.Adminlogin" %>**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<%

// hme apna hibernate.cfg.xml ko iss page per load karne ke liye hum ye likhte hai kyo ki iss file ke ander hi hmara database se connect karne ka code likha hai

**SessionFactory sf= new Configuration().configure().buildSessionFactory();**

// we also write is like this

//Configuration conf= new Configuration();

// Configuration config = conf.configure();

// SessionFactory sessionFactory = config.buildSessionFactory();

// ye apne se hi hibernate.cfg.xml file ko load kar dega

// agar hmare hibernate.cfg.xml file ka kuch aur name hai jaise ki har.cfg.xml tab hum ye likhenge

//SessionFactory sf= new //Configuration("Har.xml").configure().buildSessionFactory();

// it help to communicate with the database

**Session k=sf.openSession();//** it create the connection between program

**Query rs=k.createQuery("from Adminlogin");//** Adminlogin ye class ka name hai yha per.. it is a HQL --> hibernate Query Language.// hum apne pure project me khi bhi table ka name nhi denge sirf class ka name denge

// hum table se data nikal rhe hai mtlab ki yha per hmara select query run ho rha hai background me

//ye nhi kaam kar rha hai pta nhi kyo

//Iterator i=rs.iterate();

//while(i.hasNext())

//{

//out.println(i.next());

//}

// ye bhi pta nhi kyo kaam nhi kar rha hai

//for(Iterator it=rs.iterate();it.hasNext();)

//{

//

// Adminlogin b=(Adminlogin) it.next();

// out.println(b.getId());

// out.println(b.getUser());

// out.println(b.getPass());

//

//}

// isliye hum list ke help se print kra rhe hai table ke data ko

**List<Adminlogin> it=rs.list();**

**for(Adminlogin b:it)**

**{**

**out.print(b.getId()+" ");**

**out.print(b.getUser()+" ");**

**out.print(b.getPass()+" <br>");**

**}**

%>

</body>

</html>

**Save the data in the database(insert the data in database).**

**<%**

*SessionFactory sf= new Configuration().configure().buildSessionFactory();*

*Session s= sf.openSession();*

*Transaction t= s.beginTransaction();*

*Student st=new Student();*

*st.setRollno(3);*

*st.setName("Ashu kumari");*

*s.save(st);*

*t.commit();*

**%>**

**Hibernate Association Mappings : -**

**@OneTOOne**

**@OneTOMany**

**@ManyTOOne**

**@ManyToMany**

**This annotation help to join the table .**

**@One TO One**

**Patient.java PatientDetails.java**

*PatientDetails d*

**Yha hum Patient class(table) me PatientDetails class(table) ka Reference bna rhe hai . ye unidirectional ban jayega mtlab ki hum Patient class(table) ke help se PatientDetails ko find kar sakte hai .(*One direction*).**

**OneToOne : -**

**Patient.java PatientDetails.java**

*@ID*

*Int pid;*

*String Name;*

*@OneToOne(mappedBy =” k”)*

*PatientDetails d;*

*@ID*

*Int pid;*

*@OneToOne*

*@JoinColumn(name=”pid”)*

*// pid=>PK(of Patient table)*

*Patient k;*

**Yha hum aab Patient table se PatientDetails ko access kar sakte hai aur PatientDetails table se Patient ko access kar sakte hai .(*unidirectional mapping*).(cross – reference).**

**@JoinColumn(name=pid) : ->***join kar rhe hai table ko aur pid Patient table ka Primary key hai.*

*mappedBy =k* **and** *Patient k; 🡪* **both value must be same like here it is k.**

**yha per pid hi PK hai aur FK hai…**

**Add Patient data in the table………………..**

*<%@ page language="java" contentType="text/html; charset=UTF-8"*

*pageEncoding="UTF-8"%>*

*<%@page import="org.hibernate.\*" %>*

*<%@page import="org.hibernate.cfg.\*" %>*

*<%@page import="java.util.\*" %>*

*<%@page import="com.entity.\*" %>*

*<!DOCTYPE html>*

*<html><head>*

*<meta charset="UTF-8">*

*<title>Insert title here</title>*

*</head><body>*

*<%*

***SessionFactory sf=new Configuration().configure().buildSessionFactory();***

***Session s=sf.openSession();***

*//agar hme insert//update//delete opertation perform karna hai toh hume //Transaction ko add karna hoga*

***Transaction t=s.beginTransaction();***

*// iske bina hum commit nhi kar payenge aur jab tak hum commit nhi krenge //tab tak database me data store nhi hoga*

***PatientDetails d=new PatientDetails();***

***d.setPatient\_id(102);***

***d.setName("rajan");***

***d.setAddress("Delhi");***

***Patient p=new Patient();***

***p.setPatient\_id(102);***

***p.setUsername("rajan@123");***

***p.setD(d);***

***s.save(d);****// most important line*

***s.save(p);****// isse mera sara data table me insert ho jayega database me*

*// hibernate background me insert Query likhega hmare liye*

*//s.delete(p);// it delete that Patient record*

*//s.update(p);// it perform the update operation*

***t.commit();***

*%>*

*</body></html>*

***AddPatient***

***<%***

*SessionFactory sf=new Configuration().configure().buildSessionFactory();*

*Session s=sf.openSession();*

*Transaction t=s.beginTransaction();*

*PatientDetails d=new PatientDetails();*

*d.setPatient\_id(102);*

*d.setName("rajan");*

*d.setAddress("Delhi");*

*Patient p=new Patient();*

*p.setPatient\_id(102);*

*p.setUsername("rajan@123");*

*p.setD(d);*

*s.save(d);*

*s.save(p);*

*t.commit();*

***%>***

***View Patients :***

***<%***

*SessionFactory sf=new Configuration().configure().buildSessionFactory();*

*Session s=sf.openSession();*

*Patient p1=s.get(Patient.class,100);*

*PatientDetails d=p1.getD();*

*out.println(p1.getPatient\_id()+" "+p1.getUsername()+" "+d.getAddress()+" "+d.getMobile()+" "+d.getName()+" "+d.getPatient\_id()+" "+d.getGender());*

***%>***

**To view the Patients details :**

*<%@ page language="java" contentType="text/html; charset=UTF-8"*

*pageEncoding="UTF-8"%>*

*<%@ page import="org.hibernate.cfg.\*" %>*

*<%@ page import="org.hibernate.\*" %>*

*<%@ page import="java.util.\*" %>*

*<%@ page import="com.entity.\*" %>*

*<!DOCTYPE html>*

*<html><head>*

*<meta charset="UTF-8">*

*<title>Insert title here</title></head><body>*

*<%*

***SessionFactory sf=new Configuration().configure().buildSessionFactory();***

***Session s=sf.openSession();***

*//****Approach 1*** *agar ek table se data ko nikalna hai tab ye krenge*

*//Patient p=s.get(Patient.class,100);// background me ye query execute kerga --> select \* from Patient where Patient\_id=100;*

*// Patient is a table and 100 is the Primary key value (Patient\_id);*

*//out.println(p.getPatient\_id()+" "+p.getUsername());*

*//****Approach 2*** *// agar hme Patient aur Patientdetail dono se data ko fatch karna hai tab hume ye karna hoga : -*

***Patient p1=s.get(Patient.class,100);***

***PatientDetails d=p1.getD();***

***out.println(p1.getPatient\_id()+" "+p1.getUsername()+" "+d.getAddress()+" "+d.getMobile()+" "+d.getName()+" "+d.getPatient\_id()+" "+d.getGender());***

*%>*

*</body></html>*

**@One To Many**

**Doctor.java Department.java**

*@ID*

*Int deptid;*

*String name;*

*@OneToMany(mappedBy = “dept”)*

*List<Doctor> d;*

*@ID*

*Int docid;*

*String Name;*

*@ManyToOne*

*@joinColumn(name=”deptid”)*

*Department dept;*

**Yha per ek department me ek se jada doctor ho sakte hai iss liye isme one to many relationship hoga .**

**Agar ek department me ek hi doctor hota toh hum simple uss class ka reference de dete(Doctor k) in Department table per agar ek class se jada hai tab hme List of collection bnana hoga Doctor ka (***List<Doctor> d;***) in the Department table.**

***Imp:***

Jis table me hme foreign key bnana hai usme hi **@joinColumn()** likhenge aur aur department table ke kis column se hme join karna hai vo **name=”deptid”** likha denge.

Yha per hum doctor table me foreign key bna rhe hai toh doctore table me ek aur column ban jayega deptid ka jo ki foreign key hoga aur vhi department table ka primary key hoga.

Jis table me hum **@joinColumn** likha rhe usme hi Foreign key bnega

Jis table me hum ***@OneToMany(mappedBy = “dept”)*** *likha rhe hai uska hi reference liya jar ha hai .*

***mappedBy =”dept”*** *ka “dept” aur Doctor table me jo department ka reference bna ha* ***Department dept****; ye dono same hona chahiye…*

*@joinColumn****(name=”deptid”****) : - iss line se doctor table me ek column ban jayega iska naam* ***deptid*** *hoga aur* ***ye foregin key*** *ban jayega table ka.*

*Doctor table: Department table*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *docid* | *int* | *NO* | *PRI* |  |
| *Name* | *varchar(255)* | *YES* |  |  |
| *deptid* | *int* | *YES* | *MUL* |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| deptid | Int | NO | PRI |  |  |
| name | varchar(255) | YES |  |  |  |

Note :

Jab bhi hme data insert karna hoga indono table mer toh hme sabse pahle Department table me data set karna hoga aur jis table me hum foregin key bnaye hai uss table me baad me data insert hoga……

**Insert data into table :**

<%@ **page** language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<%@**page** import=*"com.dbase.\*,com.One\_to\_Many.\*,java.util.\*,org.hibernate.\*,org.hibernate.cfg.\*,com.entity.\*"* %>

<!DOCTYPE html>

<html><head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head><body>

<%

**Dbconnect d=new Dbconnect();**

**Session s=d.getSession();**

**Transaction t=s.beginTransaction();**

**Department dept=new Department();**

**dept.setDeptid(300);**

**dept.setName("Ortho");**

**s.save(dept);**

**Doctor doc=new Doctor();**

**doc.setName("Harsht");**

**doc.setDocid(11);**

**doc.setDept(dept);**

**s.save(doc);**

**t.commit();**

%>

</**body**></**html**>

**Search.jsp:----------------------------------------------------**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html><head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head><body>

<form action=*"deleteDoctor.jsp"*>

Doctor id<input type=*"text"* name=*"id"*>

<input type=*"submit"*>

</form></body></html>

**deleteDoctor.jsp**

<%

Dbconnect d=new Dbconnect();

Session s=d.getSession();

int id=Integer.parseInt(request.getParameter("id"));

Transaction t=s.beginTransaction();

Doctor doc=new Doctor();

doc.setDocid(id);

s.delete(doc);

t.commit();

%>

**Search doctor : ---------------------------------------------------------**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html><html><head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head><body>

<form action=*"viewDepartmentandDoctor.jsp"*>

department id<input type=*"text"* name=*"id"*>

<input type=*"submit"*>

</form></body></html>

***viewDepartmentandDoctor.jsp : ------------------------------------------------------***

<%

Dbconnect d=new Dbconnect();

Session s=d.getSession();

int id=Integer.parseInt(request.getParameter("id"));

Department dept=s.get(Department.class,id);

out.println(dept.getDeptid()+" "+dept.getName());

List<Doctor> listdoc=dept.getD();// jis dr ki department id 300 hoga vo sare dr List me aajayenge

out.println("<table border 2px>");

for(Doctor doc : listdoc)

{

out.println("<tr><td>"+doc.getDocid()+" "+doc.getName()+"</td></tr>");

}

out.println("</table>");

%>

**@Many To Many**

**Patient2.java**

*@Id*

*int pid;*

*String pname;*

*@ManyToMany*

*@JoinTable(name="patienttest",joinColumns=@JoinColumn(name="pid"),inverseJoinColumns=*

*@JoinColumn(name="testid"))*

*// table name = patienttest , primary key of Patient2, Primary key of test*

*Set<Test>tests=new HashSet<>();*

**Test.java**

*@Id*

*int testid;*

*String name;*

*@ManyToMany(mappedBy="tests")*

*Set <Patient2> patient=new HashSet<>();*

*Note : -*

*Hme yha aab many to many mapping karna hai.*

*Ex : - ek Patient ka ek se jada test ho sakta hai aur ek test kai patients ka ho sakta hai toh yha per many to many relationship ban rha hai Patient apne kai sare test kra sakta hai aur ek test kai sare Patients kra sakte hai isse ye smaj aata hai ki yha many to many relationship hoga..*

*Agar hme many to many relationship karna hai toh hme ek nya table bnana hoga jisme ki ho ki kon sara parient konsa test kraya hai hme apne se vo table nhi bnana hoga vo khud apne se ban jayega* ***joinTable*** *ke help se..*

*Yha per many to many relation ban rha hai toh hme List ya set ka use karna hoga kyo ki ek patient ke bahot sare test ho sakte hai aur test bahot sare patient ka ho sakta hai…*

*Yha hum set aur hashset use kar rhe hai taki ki unique data store ho ..*

***@JoinTable(name="patienttest",joinColumns=@JoinColumn(name="pid"),inverseJoinColumns=@JoinColumn(name="testid"))***

*Yha per hum ek nya table bna rhe hai jisme do hi column rhega ek column patient2 table se aaya hai aur ek column Test table se aaya hai dono column Primary key hona chahiye dono table ka aur table ka name=”****patienttest”*** *aur dono column ka name hum yha per diye hai ek ka name* ***pid*** *hai aur ek column ka name* ***testid*** *hai dono Primary key honge nye table me aur apne apne table me bhi.*

*Isme hum foregin key nhi bna rhe hai yha hum direct nya table bna rhe hai aur mapping ka use kar rhe hai jis class me hum table nhi bna rhe hai usme hum mapping krenge apne dusare class ke reference variable ko…*

***@ManyToMany(mappedBy="tests")***

**HQL(Hibernate Query Language)**

***-------------------------------------- Named Parameter--------------------------------------***

*dbconnect c= new dbconnect();*

*Session s= c.getSession();*

*Query <Student> q= s.createQuery("from Student where rollno =:roll");*

*q.setParameter("roll",1);*

*List<Student> p= q.list();*

*for(Student r:p){*

*out.println(r.getRollno() + " "+ r.getName());*

*}*

***------------------------------------------- HCQL-------------------------------------------***

*dbconnect c= new dbconnect();*

*Session s= c.getSession();*

*Criteria q= s.createCriteria(Student.class);*

*q.add(Restrictions.eq("name","neha"));*

*q.addOrder(Order.desc("rollno"));*

*List<Student> p= q.list();*

*for(Student r:p){*

*out.println(r.getRollno() + " "+ r.getName());*

*}*

***--------------------------------------*** *Named Queries****--------------------------------------***

*step1*

*@NamedQueries( {*

*@NamedQuery(name="s1" , query="from Student where rollno=:roll"),*

*@NamedQuery(name="s2",query="from Student where name=:name)*

*})*

*@Entity*

*@Table(name="student")*

*public class Student {*

*....}*

*step 2 -----------jsp page ---------*

*Query <Student> q= s.getNamedQuery("s1");*

*q.setParameter("roll",2);*

*List<Student> p= q.list();*

*for(Student r:p){*

*out.println(r.getRollno() + " "+ r.getName());*

*}*