**<https://www.tutorialspoint.com/hibernate/hibernate_query_language.htm>**

[**https://github.com/topics/hackerrank-sql-solutions**](https://github.com/topics/hackerrank-sql-solutions)

[**https://github.com/Java-aid**](https://github.com/Java-aid)

[**https://github.com/topics/hackerrank-solutions**](https://github.com/topics/hackerrank-solutions)

**---------------03/01/2024—Wednesday---Class-1------------------------------------**

**Projects: Hibernate,Hibernate2, HibernatePdt**

**Packages: see below three for Hibernate(modified in spring), com.student, com.product**

**Programs: 1)2)3) below 4)student.java 5)stuDao.java 6)Test.java 7)product.java 8)pdtDao.java 9)Test.java**

**What Are actually Session Factory& Session mean and does?**

[**https://youtu.be/sz\_TeaYz8B8?si=y92HsOa2zMd\_eKi6**](https://youtu.be/sz_TeaYz8B8?si=y92HsOa2zMd_eKi6)

**1)employee.java**

**package** com.cisco;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

@Entity

**public** **class** employee {

@Id

@GeneratedValue

**private** **int** eid;

**private** String ename;

**private** **double** esal;

**private** String edes;

@Override

**public** String toString() {

**return** "eid=" + eid + ", ename=" + ename + ", esal=" + esal + ", edes=" + edes + " ";

}

**public** **int** getEid() {

**return** eid;

}

**public** **void** setEid(**int** eid) {

**this**.eid = eid;

}

**public** String getEname() {

**return** ename;

}

**public** **void** setEname(String ename) {

**this**.ename = ename;

}

**public** **double** getEsal() {

**return** esal;

}

**public** **void** setEsal(**double** esal) {

**this**.esal = esal;

}

**public** String getEdes() {

**return** edes;

}

**public** **void** setEdes(String edes) {

**this**.edes = edes;

}

}

2)empDao.java

**package** com.cisco;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.AnnotationConfiguration;

**import** org.hibernate.cfg.Configuration;

//driver class

//url, pass,mapp, @id donot give, transaction

//@xml file

**public** **class** empDao {

**public** **void** savedata(employee em) {

System.***out***.println("Saving the emp data to database partially");

// System.out.println(em);

Configuration con = **new** AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

Transaction t = s.beginTransaction();

s.save(em);

t.commit();

s.close();

}

}

3)Test.java

**package** com.cisco;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

employee e = **new** employee();

e.setEid(1);

e.setEname("abc");

e.setEsal(345.6);

e.setEdes("se");

empDao d = **new** empDao();

d.savedata(e);

}

}

EXCEPTIONS:

WHEN YOU GIVE WRONG ANY OTHER CLASS INSTEAD OF DRIVER CLASS?

EXCEPTION:

Exception in thread "main" org.hibernate.boot.registry.classloading.spi.ClassLoadingException: Unable to load class [com.mysql.cj.jdbc.hujd]

WHEN YOU GIVE WRONG SCHEMA NAME?

EXCEPTION:

Exception in thread "main" org.hibernate.exception.SQLGrammarException: Error calling Driver#connect

WHEN YOU GIVE WRONG USERNAME?

Exception in thread "main" org.hibernate.exception.JDBCConnectionException: Error calling Driver#connect

WHEN YOU GIVE WRONG PASSWORD?

Exception in thread "main" org.hibernate.exception.JDBCConnectionException: Error calling Driver#connect(same)

WHEN YOU GIVE WRONG DIALECT?

Exception in thread "main" org.hibernate.boot.registry.selector.spi.StrategySelectionException: Unable to resolve name [org.hibernate.dialect.MySQDialect] as strategy [org.hibernate.dialect.Dialect]

WHEN YOU GIVE WRONG MAPPING CLASS?

Exception in thread "main" org.hibernate.MappingException: Unable to load class [ com.product.employee] declared in Hibernate configuration <mapping/> entry

Caused by: java.lang.ClassNotFoundException: com.product.employee

WHEN YOU DID NOT GIVE ANNOTATION?

Exception in thread "main" org.hibernate.AnnotationException: No identifier specified for entity: com.cisco.employee

-----------------04/01/2024-------------Thursday---CLASS-2---------------------

Morning class

Topic: How retrieve the data by ID. How to update a row. How to delete a row.

Afternoonclass

Topic:getAll data, How to add multiple classes in xml file

Project:Hibernate

Packages: pojo, controller, com.test

Programs: 1)employee.java

2)department.java

3)product.java

4)getCon.java

5)empDao.java

6)depDao.java

7)productDao.java

8)Test.java

9)Testde.java

10)Testpdt.java

1)hibernate.cfg.xml

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

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

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

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

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

<property name="hibernate.connection.password">root</property>

<property name="hbm2ddl.auto">update</property>

<property name="hibernate.show\_sql">true</property>

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

<mapping class="com.cisco.employee" />

</session-factory>

</hibernate-configuration>

2)employee.java

package com.cisco;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class employee {

@Id

@GeneratedValue

private int eid;

private String ename;

private double esal;

private String edes;

@Override

public String toString() {

return "eid=" + eid + ", ename=" + ename + ", esal=" + esal + ", edes=" + edes + " ";

}

public int getEid() {

return eid;

}

public void setEid(int eid) {

this.eid = eid;

}

public String getEname() {

return ename;

}

public void setEname(String ename) {

this.ename = ename;

}

public double getEsal() {

return esal;

}

public void setEsal(double esal) {

this.esal = esal;

}

public String getEdes() {

return edes;

}

public void setEdes(String edes) {

this.edes = edes;

}

}

3)empDao.java

package com.cisco;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

//driver classs

//url, pass,mapp, @id donot give, transaction

//@xml file

public class empDao {

public void savedata(employee em) {

System.out.println("Saving the emp data to database partially");

// System.out.println(em);

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

Transaction t = s.beginTransaction();

s.save(em);

t.commit();

s.close();

}

public void getByID(int eid) {

System.out.println("Saving the emp data to database partially");

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

employee e =(employee)s.get(employee.class, eid);

System.out.println(e);

s.close();

}

public void updatedata() {

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

Transaction t = s.beginTransaction();

employee e = (employee)s.get(employee.class, 1);

e.setEname("siva");

e.setEsal(456.7);

e.setEdes("se");

s.update(e);

t.commit();

s.close();

}

public void deletedata() {

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

Transaction t = s.beginTransaction();

employee e = (employee)s.get(employee.class, 4);

s.delete(e);

t.commit();

s.close();

}

}

4)Test.java

**package** com.cisco;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

employee e = **new** employee();

empDao d = **new** empDao();

e.setEid(1);

e.setEname("abc");

e.setEsal(345.6);

e.setEdes("se");

//d.savedata(e);

//d.updatedata();

d.deletedata();

}

}

--------------------05/01/2024—Friday---Class -3--------------------------------------

TOPIC:

getAllselect();

getAllwhere();

getAllwheredynamic(2, 800.00);

updateselect(3,367.5,"mred");deleteusinghql("ghi");

\*\*\*\*\*\*\*\*\*\*\*\*AGGREGATE FUNCTIONS[COUNT,AVG,MAX,MIN,SUM,GROUPBY, ORDERBY(ASC,DESC), HAVING AFTER THIS JOINS]\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **void** getByAll() {

System.***out***.println("Retrieving all the data");

Session s= getCon.*con*();

Query t =s.createQuery("from employee");

List<employee>data=t.list();

**for**(employee p:data) {

System.***out***.println(p);

}

s.close();

}

🡪🡪In this the query works when it is like below

Select\*from employee

But when I need to write with where& and I want to retrieve only ename, esal data we cannot do it here like this . we have another style and nature

Learn How to google first?

Hql queries

<https://docs.jboss.org/hibernate/core/3.5/reference/en/html/queryhql.html>

<https://www.tutorialspoint.com/hibernate/hibernate_query_language.htm>

**public** **void** getAllselect() {

Session s = getCon.*con*();

Query q =s.createQuery("select e.ename(PROPERTY NAMES NOT THE COLUMN NAMES), e.esal from employee e"); [POJO CLASS NAME NOT TABLE NAME]

List<Object[]> g = q.list();

**for**(Object[] o: g) {

System.***out***.println(o[0]+" "+o[1]);

}

s.close();

}

The below all are static.

**public** **void** getAllwhere() {

Session s = getCon.*con*();

Query q =s.createQuery("select e.ename, e.esal, e.edes from employee e where e.eid>=2 and e.esal>=800");

List<Object[]> g = q.list();

**for**(Object[] o: g) {

System.***out***.println(o[0]+"-----"+o[1]+"-----"+o[2]);

}

s.close();

}

To change it to dynamic.

We have set parameter()

Commonly by session we can perform

Insert

Perform following operations by session

Save

Update

Delete

Get

Load

Create query

Hql queries

Update

Delete

Select

Select and where

Orderby

aggregator

groupby

---aggregator and groupby and orderby

1)empDao.java

package controller;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

import pojo.employee;

public class empDao {

public void savedata(employee em) {

System.out.println("Saving the emp data to database partially");

// System.out.println(em);

Session s= getCon.con();

Transaction t = s.beginTransaction();

s.save(em);

t.commit();

s.close();

}

public void getByID(int eid) {

System.out.println("Saving the emp data to database partially");

Session s= getCon.con();

employee e =(employee)s.get(employee.class, eid);

System.out.println(e);

s.close();

}

public void updatedata() {

Session s= getCon.con();

Transaction t = s.beginTransaction();

employee e = (employee)s.get(employee.class, 1);

e.setEname("siva");

e.setEsal(456.7);

e.setEdes("se");

s.update(e);

t.commit();

s.close();

}

public void deletedata() {

Session s= getCon.con();

Transaction t = s.beginTransaction();

employee e = (employee)s.get(employee.class, 4);

s.delete(e);

t.commit();

s.close();

}

public void getByAll() {

System.out.println("Retrieving all the data");

Session s= getCon.con();

Query t =s.createQuery("from employee");

List<employee>data=t.list();

for(employee p:data) {

System.out.println(p);

}

s.close();

}

//05/01/2024

public void getAllselect() {

Session s = getCon.con();

Query q =s.createQuery("select e.ename, e.esal,e.edes from employee e");

List<Object[]> g = q.list();

for(Object[] o: g) {

System.out.println(o[0]+"---"+o[1]+"----"+o[2]);

}

s.close();

}

public void getAllwhere() {

Session s = getCon.con();

Query q =s.createQuery("select e.ename, e.esal, e.edes from employee e where e.eid>=2 and e.esal>=800");

List<Object[]> g = q.list();

for(Object[] o: g) {

System.out.println(o[0]+"-----"+o[1]+"-----"+o[2]);

}

s.close();

}

public void getAllwheredynamic(int eid , double esal) {

Session s = getCon.con();

Query q =s.createQuery("select e.ename, e.esal, e.edes from employee e where e.eid>=:eid and e.esal>=:esal");

q.setParameter("eid", eid);

q.setParameter("esal", esal);

List<Object[]> g = q.list();

for(Object[] o: g) {

System.out.println(o[0]+"-----"+o[1]+"-----"+o[2]);

}

s.close();

}

public void updateselect(int eid,double esal, String edes) {

Session s = getCon.con();

Transaction t = s.beginTransaction();

String h ="update employee set esal=:esal, edes=:edes where eid=:eid";

Query q = s.createQuery(h);

q.setParameter("esal",esal);

q.setParameter("edes",edes);

q.setParameter("eid",eid);

int r=q.executeUpdate();

System.out.println("updated row:"+r);

t.commit();

s.close();

}

public void deleteusinghql(String ename) {

Session s = getCon.con();

Transaction t = s.beginTransaction();

String h =" delete from employee where ename=:ename";

Query q = s.createQuery(h);

q.setParameter("ename",ename);

int r=q.executeUpdate();

System.out.println("deleted row:"+r);

t.commit();

s.close();

}

//Aggregate Functions

public void orderby() {

System.out.println("Retrieving all the data");

Session s= getCon.con();

//Query t =s.createQuery("from employee order by esal");

/\*List<employee>data=t.list();

for(employee p:data) {

System.out.println(p);

}

s.close();\*/

Query t =s.createQuery("select ename, edes from employee order by esal");

List<Object[]>data=t.list();

for(Object []p:data) {

String ename = (String) p[0];

String edes = (String) p[1];

System.out.println("Name: " + ename + ", Designation: " + edes);

}

s.close();

}

public void count() {

Session s = getCon.con();

Query q =s.createQuery("select count(\*) from employee");

List<Object> g = q.list();

System.out.println(g);

s.close();

}

public void max() {

Session s = getCon.con();

Query q =s.createQuery("select max(esal) from employee");

List<Object> g = q.list();

System.out.println(g);

s.close();

}

public void min() {

Session s = getCon.con();

Query q =s.createQuery("select min(esal) from employee");

List<Object> g = q.list();

System.out.println(g);

s.close();

}

public void sum() {

Session s = getCon.con();

Query q =s.createQuery("select sum(esal) from employee");

List<Object> g = q.list();

System.out.println(g);

s.close();

}

public void getAllagg() {

Session s = getCon.con();

Query q =s.createQuery("select count(\*),max(esal),min(esal),sum(esal) from employee");

List<Object[]> g = q.list();

for(Object[] p: g ) {

System.out.println(p[0]+"---"+p[1]+"---"+p[2]+"---"+p[3]);

}

s.close();

}

public void groupby() {

Session s = getCon.con();

Query q =s.createQuery("select count(\*),max(esal),min(esal),sum(esal),edes from employee group by edes ");

List<Object[]> g = q.list();

for(Object[] p: g ) {

System.out.println(p[0]+"---"+p[1]+"---"+p[2]+"---"+p[3]+"---"+p[4]);

}

s.close();

}

public void groupbyandorderby() {

Session s = getCon.con();

Query q =s.createQuery("select max(esal),min(esal),sum(esal),count(\*),edes from employee group by edes order by esal desc ");

List<Object[]> g = q.list();

for(Object[] p: g ) {

System.out.println(p[0]+"---"+p[1]+"---"+p[2]+"---"+p[3]+"---"+p[4]);

}

s.close();

}

}

2)employee.java

package pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class employee {

@Id

@GeneratedValue

private int eid;

private String ename;

private double esal;

private String edes;

@Override

public String toString() {

return "eid=" + eid + ", ename=" + ename + ", esal=" + esal + ", edes=" + edes + " ";

}

public int getEid() {

return eid;

}

public void setEid(int eid) {

this.eid = eid;

}

public String getEname() {

return ename;

}

public void setEname(String ename) {

this.ename = ename;

}

public double getEsal() {

return esal;

}

public void setEsal(double esal) {

this.esal = esal;

}

public String getEdes() {

return edes;

}

public void setEdes(String edes) {

this.edes = edes;

}

}

3)Test.java

package com.test;

import controller.empDao;

import pojo.employee;

public class Test {

public static void main(String[] args) {

employee e = new employee();

empDao d = new empDao();

e.setEid(1);

e.setEname("abc");

e.setEsal(345.6);

e.setEdes("se");

//d.savedata(e);

//d.updatedata();

//d.deletedata();

//d.getByAll();

//d.getAllselect();

//d.getAllwhere();

//d.getAllwheredynamic(2, 800.00);

//d.updateselect(3,367.5,"mred");

//d.deleteusinghql("ghi");

//d.orderby();

//\*\*\*\*\*\*\*\*\*\*\*\*AGGREGATE FUNCTIONS[COUNT,AVG,MAX,MIN,SUM,GROUPBY, ORDERBY(ASC,DESC), HAVING AFTER THIS JOINS]\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//d.orderby();

//d.count();

//d.max();

//d.min();

//d.sum();

d.getAllagg();

//d.groupby();

//d.groupbyandorderby();

}

}

**-----------------------08-01-2024---MONDAY-----------Class -4----------------------**

TOPIC: HIBERNATE RELATIONS(ONE TO ONE RELATION COMPLETED -- ONLY IN SINGLE DIRECTION NOT IN BIDIRECTION)

NOTE: THE TABLE WHICH CONSISTS OF JOIN RELATION IS CALLED THE PARENT (HERE PERSON IS THE PARENT, PASSPORT IS THE CHILD)

* IF YOU WANT TO DELETE PERSON , FIRST YOU NEED TO DELETE IT’S CHILD PASSPORT

PROJECT: Hibernate Relations, HRelations

Packages: com.pojo, com.controller

Programs:

1)Person.java

2)Passport.java

3)OneToOne.java

4)getCon.java

1)Person.java

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.CascadeType;

import javax.persistence.\*;

@Entity

public class Person {

@Id

@GeneratedValue

private int pid;

private String pname;

private int page;

private String pqual;

@OneToOne(cascade =CascadeType.ALL)

@JoinColumn(name = "fkpid")

private Passport ppt;

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public String getPqual() {

return pqual;

}

public void setPqual(String pqual) {

this.pqual = pqual;

}

public Passport getPpt() {

return ppt;

}

public void setPpt(Passport ppt) {

this.ppt = ppt;

}

}

2)Passport.java

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Passport {

@Id

@GeneratedValue

private int ppid;

private String pnum;

private String pexp;

public int getPpid() {

return ppid;

}

public void setPpid(int ppid) {

this.ppid = ppid;

}

public String getPnum() {

return pnum;

}

public void setPnum(String pnum) {

this.pnum = pnum;

}

public String getPexp() {

return pexp;

}

public void setPexp(String pexp) {

this.pexp = pexp;

}

}

3)getCon.java

package com.controller;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class getCon {

public static Session con() {

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

return s;

}

}

4)OneToOne.java

package com.controller;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import com.pojo.\*;

public class OneToOne {

public void savedata() {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Person p = new Person();

/\*p.setPname("Ram");

p.setPage(33);

p.setPqual("B.Tech");\*/

p.setPname("Lokesh");

p.setPage(37);

p.setPqual("M.Tech");

/\*Passport q = new Passport();

q.setPnum("AB123");

q.setPexp("2060");

p.setPpt(q);\*/

Passport q = new Passport();

q.setPnum("B123Q");

q.setPexp("2070");

p.setPpt(q);

r.save(p);

t.commit();

r.close();

}

//GET BY ID

public void getbyId(int pid) {

Session r = getCon.con();

Person p = (Person) r.get(Person.class, pid);

System.out.println(p.getPid()+", "+p.getPage()+", "+p.getPqual()+", "+p.getPpt().getPnum()+", "+p.getPpt().getPexp());

}

//GETALL

public void getAll() {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Query q=r.createQuery("from Person");

List<Person> data =q.list();

for(Person p:data) {

System.out.println(p.getPid()+", "+p.getPage()+", "+p.getPqual()+", "+p.getPpt().getPnum()+", "+p.getPpt().getPexp());

}

}

//UPDATE

public void update(int pid) {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Person p = (Person) r.get(Person.class, pid);

p.setPname("klin");

p.setPage(55);

p.setPqual("MA");

p.getPpt().setPexp("2045");

r.update(p);

t.commit();

r.close();

}

//DELETE

public void delete() {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Person p = (Person) r.get(Person.class, 4);

r.delete(p);

t.commit();

r.close();

}

public static void main(String[] args) {

OneToOne l = new OneToOne();

//l.savedata();

//l.getbyId(1);

l.getAll();

// l.update(1);

//l.delete();

}

}

**-----------09/01/2024------------------Tuesday---------Class -5-------------------**

TOPIC: HIBERNATE RELATIONS(ONE TO MANY RELATION COMPLETED -- ONLY IN SINGLE DIRECTION NOT IN BIDIRECTION)

Project: HibernateRelations\_OneToMany

Packages:

1)Pojo

2)Controller

Programs:

1)Person.java

2)Passport.java

3)Mobile.java

4)getCon.java

5)OM.java(Important)

6)OneToMany.java(Revise)

1)Person.java

package Pojo;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.OneToOne;

@Entity

public class Person {

@Id

@GeneratedValue

private int pid;

private String pname;

private int page;

private String pqua;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "fkpid")

private Passport pt;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name = "mkfid")

private List<Mobile> mb;

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public String getPqua() {

return pqua;

}

public void setPqua(String pqua) {

this.pqua = pqua;

}

public Passport getPt() {

return pt;

}

public void setPt(Passport pt) {

this.pt = pt;

}

public List<Mobile> getMb() {

return mb;

}

public void setMb(List<Mobile> mb) {

this.mb = mb;

}

}

2)Passport.java

package Pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Passport {

@Id

@GeneratedValue

private int ptid;

private String ptnum;

private String pexp;

public int getPtid() {

return ptid;

}

public void setPtid(int ptid) {

this.ptid = ptid;

}

public String getPtnum() {

return ptnum;

}

public void setPtnum(String ptnum) {

this.ptnum = ptnum;

}

public String getPexp() {

return pexp;

}

public void setPexp(String pexp) {

this.pexp = pexp;

}

}

3)Mobile.java

package Pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Mobile {

@Id

@GeneratedValue

private int mid;

private long mnum;

private String msim;

public Mobile() {

}

public Mobile( long mnum, String msim) {

this.mnum = mnum;

this.msim = msim;

}

public int getMid() {

return mid;

}

public void setMid(int mid) {

this.mid = mid;

}

public long getMnum() {

return mnum;

}

public void setMnum(long mnum) {

this.mnum = mnum;

}

public String getMsim() {

return msim;

}

public void setMsim(String msim) {

this.msim = msim;

}

}

4)getCon.java

package Controller;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class getCon {

public static Session con() {

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

return s;

}

}

5)OM.java

// important

package Controller;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import Pojo.Person;

import Pojo.Mobile;

import Pojo.Passport;

import java.util.\*;

public class OM {

public void savedata() {

Session g = getCon.con();

Transaction t = g.beginTransaction();

Person p = new Person();

p.setPname("Ram");

p.setPage(100);

p.setPqua("MCA");

Passport q = new Passport();

q.setPtnum("AB123C");

q.setPexp("2056");

p.setPt(q);

List<Mobile> mo = new ArrayList<>();

mo.add(new Mobile(9274881973l,"dcomo"));

mo.add(new Mobile(7626157831l,"idea"));

mo.add(new Mobile(67365378168l,"Jio"));

p.setMb(mo);

g.save(p);

t.commit();

g.close();

}

public void getbyId(int pid) {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Person k =(Person)r.get(Person.class, pid);

System.out.println(k.getPid()+","+k.getPname()+","+k.getPqua());

Passport m=k.getPt();

if( m !=null) {

System.out.println(m.getPtid()+","+m.getPtnum()+","+m.getPexp());

}

else {

System.out.println("No Passport");

}

List<Mobile> mo = k.getMb();

if(mo!=null) {

for(Mobile mob:mo) {

System.out.println(mob.getMid()+","+mob.getMnum()+","+mob.getMsim());

}

}

else {

System.out.println("No Mobile");

}

}

public void getAll() {

Session r = getCon.con();

Transaction t=r.beginTransaction();

Query d =r.createQuery("from Person");

List<Person>data =d.list();

for(Person k:data) {

System.out.println(k.getPid()+","+k.getPname()+","+k.getPqua());

Passport m=k.getPt();

if( m !=null) {

System.out.println(m.getPtid()+","+m.getPtnum()+","+m.getPexp());

}

else {

System.out.println("No Passport");

}

List<Mobile> mo = k.getMb();

if(mo!=null&& !mo.isEmpty()) {

for(Mobile mob:mo) {

System.out.println(mob.getMid()+","+mob.getMnum()+","+mob.getMsim());

}

}

else {

System.out.println("No Mobile");

}

}

}

public void update(int pid) {

Session r = getCon.con();

Transaction t=r.beginTransaction();

Person p=(Person)r.get(Person.class, pid);

List<Mobile> mob =p.getMb();

for(Mobile mo: mob) {

if(mo.getMsim().equals("airtel")) {

mo.setMnum(765432197l);

}

}

r.update(p);

t.commit();

r.close();

}

public void delete(int pid) {

Session r = getCon.con();

Transaction t=r.beginTransaction();

Person p=(Person)r.get(Person.class, pid);

r.delete(p);

t.commit();

r.close();

}

public static void main(String[] args) {

OM j = new OM();

//j.savedata();

//j.getbyId(1);

j.getAll();

//j.update(1);

//j.delete(6);

}

}

/\*The use of mo.isEmpty() is to check whether the list of mobiles (mo) is empty. In Java, the isEmpty() method is a part of the List interface, and it returns true if the list contains no elements; otherwise, it returns false.

Here's the logic breakdown:

mo != null: Checks if the list reference (mo) is not null.

!mo.isEmpty(): Checks if the list is not only not null but also not empty.

Without the mo.isEmpty() check, the condition mo != null would be true even if the list is empty. By adding mo.isEmpty(), you ensure that the block of code inside the if statement is executed only when the list is not null and not empty.

If the list is null, it means there are no mobiles associated with the person. If the list is not null but empty, it also means there are no mobiles associated with the person. In both cases, you want to print the "No Mobile" message.\*/

6) OneToMany.java

package Controller;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import Pojo.Person;

import Pojo.\*;

import java.util.List;

import java.util.ArrayList;

public class OneToMany {

public void savedata() {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Person p = new Person();

p.setPid(3);

p.setPage(23);

p.setPname("hij");

p.setPqua("MTECH");

/\*Passport k = new Passport();

k.setPtnum("FG123P");

k.setPexp("2070");\*/

List<Mobile> d =new ArrayList<>();

d.add(new Mobile(6428948476l,"vodo"));

d.add(new Mobile(786278368264l,"docomo"));

d.add(new Mobile(67249827428l,"idea"));

p.setMb(d);

//p.setPt(k);

r.save(p);

t.commit();

r.close();

}

public void getbyId(int pid) {

Session r = getCon.con();

Person p = (Person) r.get(Person.class, pid);

System.out.println(p.getPid()+", "+p.getPage()+", "+p.getPqua());

Passport n = p.getPt();

if(n!=null) {

System.out.println(n.getPtid()+","+n.getPtnum()+","+n.getPexp());

}

else {

System.out.println("No Passport");

}

List<Mobile> mo =p.getMb();

if(mo!=null) {

for(Mobile m:mo) {

System.out.println(m.getMid()+", "+m.getMnum()+", "+m.getMsim());

}

}

else {

System.out.println("No Mobile");

}

}

public void getAll() {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Query q=r.createQuery("from Person");

List<Person> data =q.list();

for(Person p:data) {

System.out.println(p.getPid()+", "+p.getPage()+", "+p.getPqua());

Passport n = p.getPt();

if(n!=null) {

System.out.println(n.getPtid()+","+n.getPtnum()+","+n.getPexp());

}

else {

System.out.println("No Passport");

}

List<Mobile> mo =p.getMb();

if(mo!=null) {

for(Mobile m:mo) {

System.out.println(m.getMid()+", "+m.getMnum()+", "+m.getMsim());

}

}

else {

System.out.println("No Mobile");

}

System.out.println("-----------Next---------------");

}

}

public void updatedata(int pid) {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Person p = (Person) r.get(Person.class, pid);

List<Mobile> mo =p.getMb();

for(Mobile n:mo) {

if(n.getMsim().equals("airtel")) {

n.setMnum(9876543210l);

}

}

r.update(p);

t.commit();

r.close();

}

public void deletedata(int pid) {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Person p = (Person) r.get(Person.class, pid);

r.delete(p);

t.commit();

r.close();

}

public static void main(String[] args) {

OneToMany m = new OneToMany();

m.savedata();

//m.getbyId(1);

//m.getAll();

//m.updatedata(2);

//m.deletedata(2);

}

}

**-------------10/01/2024----------------Wednesday-------Class -6----------------------**

Topic: HIBERNATE RELATIONS(ONE TO MANY RELATION -- ONLY IN SINGLE DIRECTION NOT IN BIDIRECTION)

Sir given task

Project: HibRel

Packages:

1)Pojo

2)Controller

Programs:

1)Product.java

2)Vendor.java

3)Items.java

4)getCon.java

5)OneToMany\_P\_to\_V.java

6)OneToMany\_V\_to\_I.java

1)Product.java

package Pojo;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

@Entity

public class Product {

@Id

@GeneratedValue

private int pid;

private String pname;

private double pprice;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name = "vkfid")

private List<Vendor> ven;

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public double getPprice() {

return pprice;

}

public void setPprice(double pprice) {

this.pprice = pprice;

}

public List<Vendor> getVen() {

return ven;

}

public void setVen(List<Vendor> ven) {

this.ven = ven;

}

}

2) Vendor.java

package Pojo;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

@Entity

public class Vendor {

public Vendor() {

}

@Id

@GeneratedValue

private int vid;

public Vendor(String vname, String vloc) {

super();

this.vname = vname;

this.vloc = vloc;

}

private String vname;

private String vloc;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name = "itfid")

private List<Items> it;

public int getVid() {

return vid;

}

public void setVid(int vid) {

this.vid = vid;

}

public String getVname() {

return vname;

}

public void setVname(String vname) {

this.vname = vname;

}

public String getVloc() {

return vloc;

}

public void setVloc(String vloc) {

this.vloc = vloc;

}

public List<Items> getIt() {

return it;

}

public void setIt(List<Items> it) {

this.it = it;

}

}

3)Items.java

package Pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Items {

@Id

@GeneratedValue

private int iid;

public Items( String iname, String iexp, String iddate) {

super();

this.iname = iname;

this.iexp = iexp;

this.iddate = iddate;

}

private String iname;

private String iexp;

private String iddate;

public int getIid() {

return iid;

}

public void setIid(int iid) {

this.iid = iid;

}

public String getIname() {

return iname;

}

public void setIname(String iname) {

this.iname = iname;

}

public String getIexp() {

return iexp;

}

public void setIexp(String iexp) {

this.iexp = iexp;

}

public String getIddate() {

return iddate;

}

public void setIddate(String iddate) {

this.iddate = iddate;

}

}

4)getCon.java

package Controller;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class getCon {

public static Session con() {

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

return s;

}

}

5) OneToMany\_P\_to\_V.java

package Controller;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

import Pojo.\*;

import java.util.\*;

public class OneToMany\_P\_to\_V {

public void savedata() {

Session r = getCon.con();

Transaction t=r.beginTransaction();

Product p = new Product();

p.setPid(1);

p.setPname("IPHONE");

p.setPprice(1.5);

List<Vendor> ve = new ArrayList<>();

ve.add((new Vendor("ABC","HYD")));

ve.add((new Vendor("DEF","SCD")));

ve.add((new Vendor("GHI","KNR")));

p.setVen(ve);

r.save(p);

t.commit();

r.close();

}

public void getbyId(int pid) {

Session r = getCon.con();

Transaction t = r.beginTransaction();

Product k =(Product)r.get(Product.class,pid);

System.out.println(k.getPid()+","+k.getPname()+","+k.getPprice());

List<Vendor> ve = k.getVen();

if(ve!=null) {

for(Vendor ven:ve) {

System.out.println(ven.getVid()+","+ven.getVname()+","+ven.getVloc());

}

}

else {

System.out.println("No Vendors for this product");

}

}

public void getAll() {

Session r = getCon.con();

Transaction t=r.beginTransaction();

Query d =r.createQuery("from Product");

List<Product>data =d.list();

for(Product k:data) {

System.out.println(k.getPid()+","+k.getPname()+","+k.getPprice());

List<Vendor> ve = k.getVen();

if(ve!=null) {

for(Vendor ven:ve) {

System.out.println(ven.getVid()+","+ven.getVname()+","+ven.getVloc());

}

}

else {

System.out.println("No Vendors for this product");

}

}

}

public void update(int pid) {

Session r = getCon.con();

Transaction t =r.beginTransaction();

Product p=(Product)r.get(Product.class, pid);

List<Vendor> v =p.getVen();

for(Vendor ve: v) {

if(ve.getVloc().equals("KNR")) {

ve.setVname("Jai");

}

}

r.update(p);

t.commit();

r.close();

}

public void delete(int pid) {

Session r = getCon.con();

Transaction t =r.beginTransaction();

Product p=(Product)r.get(Product.class, pid);

r.delete(p);

t.commit();

r.close();

}

public void getbyv(String vname) {

Session s = getCon.con();

Query q=s.createQuery("SELECT v FROM Vendor v WHERE v.vname = :vendorName");

List<Object[]> g =q.list();

for(Object [] o:g) {

System.out.println(o[0]+"---"+o[1]+"----"+o[2]);

}

s.close();

}

public static void main(String[] args) {

OneToMany\_P\_to\_V o = new OneToMany\_P\_to\_V();

//o.savedata();

// o.getbyId(1);

//o.getAll();

//o.update(1);

//o.delete(5);

}

}

6) OneToMany\_V\_to\_I.java

package Controller;

import java.util.\*;

import org.hibernate.Session;

import org.hibernate.Transaction;

import Pojo.\*;

public class OneToMany\_V\_to\_I {

public void savedata() {

Session r = getCon.con();

Transaction t=r.beginTransaction();

Vendor p = new Vendor();

p.setVid(5);

p.setVname("IPHONE");

p.setVloc("Hyd");

List<Items> ite = new ArrayList<>();

ite.add(new Items("cho","2024","2025"));

ite.add(new Items("bis","2027","2029"));

ite.add(new Items("cho","2030","2035"));

p.setIt(ite);

r.save(p);

t.commit();

r.close();

}

public static void main(String[] args) {

OneToMany\_V\_to\_I h = new OneToMany\_V\_to\_I();

h.savedata();

}

}

------11/01/2024—class not taken-------

-----12/01/2024 To 18/01/2024—sankranti holidays------

-------19/01/2024---Friday-----class not taken-----

--------20/01/2024----Saturday -----class not taken------

-----22/01/2024---------Monday---------Stringliterals and String class taken-----

----23/01/2024----Tuesday(Morning)--------How to create our own immutable class-------

**---------------23/01/2024------------------Tuesday-----Class -7-----------------------**

AFTERNOON

Topic: Relations

One to one, One to many, many to many

Project: HibernateRelations OTO,OTM,MTM

Package: 1)com.pojo

2)com.controller

Programs: 1)Person.java

2)Passport.java

3)course.java

4)mobile.java

5)getCon.java

6)OneToOne.java

7)OneToMany.java

**------------------------------24/01/2024--------Wednesday---Class -8-----------------**

Topic: Relations

One to one, one to many, many to many

Project: Hibernate\_All

Package: 1)com.pojo

2)com.controller

Programs: 1)Person.java

2)Passport.java

3)course.java

4)mobile.java

5)All.java

6)GetSessionFactory.java

7)OneToOne.java

8)OneToMany.java

9)ManyToMany.java

10)All.java

WE DONE ONLY SAVE DATA

**------------------------25/01/2024--------Thursday------Class -9--------------------**

Topic: Relations

One to one, one to many, many to many

Project: Hibernate\_All

Package: 1)com.pojo

2)com.controller

Programs: 1)Person.java

2)Passport.java

3)course.java

4)mobile.java

5)All.java

6)GetSessionFactory.java

7)OneToOne.java

8)OneToMany.java

9)ManyToMany.java

10)All.java

WE DID GETBYID, UPDATE DATA(Add passport to existing person, Add mobiles to existing person, Add courses to existing person).

Next perform update data for existing person,passport, mobiles, courses

**---------------------26/01/2024-------Friday------Class -10----------------------------**

Topic: GETALL() IN DIFFERENT WAY,HQL QUERIES

Project:Hibernate\_All, EmpHib

Package:

1)com.controller

2)com.pojo

1)com.pojo

2)com.controller

Programs:

1)Person.java

2)Passport.java

3)mobile.java

4)course.java

5)GetSessionFactory.java

6)All.java(See getByAll1() & getByAll2() All methods not required)

1)Employee.java

2)GetSessionFactory.java

3)UnitTest.java

1)Person.java

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.CascadeType;

import java.util.List;

import javax.persistence.\*;

@Entity

public class Person {

@Id

@GeneratedValue

private int pid;

private String pname;

private int page;

private String pqual;

@OneToOne(cascade =CascadeType.ALL)

@JoinColumn(name = "fkpid")

private Passport ppt;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name = "fkpersid")

private List<mobile> mob;

@ManyToMany(cascade = CascadeType.ALL)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkpid")},

inverseJoinColumns = {@JoinColumn(name = "fkcid")})

private List<course> cou;

public Person() {

}

public Person(int pid, String pname, int page, String pqual, Passport ppt, List<mobile> mob, List<course> cou) {

super();

this.pid = pid;

this.pname = pname;

this.page = page;

this.pqual = pqual;

this.ppt = ppt;

this.mob = mob;

this.cou = cou;

}

/\*@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt + ", mob="

+ mob + ", cou=" + cou + "]";

}\*/

@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + "]";

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public String getPqual() {

return pqual;

}

public void setPqual(String pqual) {

this.pqual = pqual;

}

public Passport getPpt() {

return ppt;

}

public void setPpt(Passport ppt) {

this.ppt = ppt;

}

public List<mobile> getMob() {

return mob;

}

public void setMob(List<mobile> mob) {

this.mob = mob;

}

public List<course> getCou() {

return cou;

}

public void setCou(List<course> cou) {

this.cou = cou;

}

}

2)Passport.java

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Passport {

@Id

@GeneratedValue

private int ppid;

private String pnum;

private String pexp;

@Override

public String toString() {

return "Passport [ppid=" + ppid + ", pnum=" + pnum + ", pexp=" + pexp + "]";

}

public int getPpid() {

return ppid;

}

public void setPpid(int ppid) {

this.ppid = ppid;

}

public String getPnum() {

return pnum;

}

public void setPnum(String pnum) {

this.pnum = pnum;

}

public String getPexp() {

return pexp;

}

public void setPexp(String pexp) {

this.pexp = pexp;

}

}

3)mobile.java

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class mobile {

@Id

@GeneratedValue

private int mid;

private long mnum;

private String msim;

@Override

public String toString() {

return "mobile [mid=" + mid + ", mnum=" + mnum + ", msim=" + msim + "]";

}

public mobile() {

}

public mobile( long mnum, String msim) {

this.mnum = mnum;

this.msim = msim;

}

public int getMid() {

return mid;

}

public void setMid(int mid) {

this.mid = mid;

}

public long getMnum() {

return mnum;

}

public void setMnum(long mnum) {

this.mnum = mnum;

}

public String getMsim() {

return msim;

}

public void setMsim(String msim) {

this.msim = msim;

}

}

4)course.java

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class course {

@Id

@GeneratedValue

private Integer cid;

private String cname;

private Double cfee;

public Integer getCid() {

return cid;

}

public void setCid(Integer cid) {

this.cid = cid;

}

public String getCname() {

return cname;

}

public void setCname(String cname) {

this.cname = cname;

}

public Double getCfee() {

return cfee;

}

public void setCfee(Double cfee) {

this.cfee = cfee;

}

public course(Integer cid, String cname, Double cfee) {

super();

this.cid = cid;

this.cname = cname;

this.cfee = cfee;

}

public course() {

}

@Override

public String toString() {

return "course [cid=" + cid + ", cname=" + cname + ", cfee=" + cfee + "]";

}

}

6)All.java

package com.controller;

import java.util.ArrayList;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.pojo.Passport;

import com.pojo.Person;

import com.pojo.course;

import com.pojo.mobile;

public class All {

public void saveddata(Person p) {

SessionFactory sf = GetSessionFactory.con();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

s.save(p);

t.commit();

s.close();

}

public void updatedata(int pid) {

SessionFactory sf = GetSessionFactory.con();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

Person p =(Person)s.get(Person.class, pid);

p.setPname("Chinna");

s.update(p);

t.commit();

s.close();

sf.close();

}

public void deletedata(int pid) {

SessionFactory sf = GetSessionFactory.con();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

Person p =(Person)s.get(Person.class, pid);

s.delete(p);

t.commit();

s.close();

sf.close();

}

public void getById(int pid) {

SessionFactory sf = GetSessionFactory.con();

Session s = sf.openSession();

Person p=(Person)s.get(Person.class, pid);

System.out.println(p);

}

public void getByAll() {

SessionFactory sf = GetSessionFactory.con();

Session s = sf.openSession();

Query q=s.createQuery("from Person");

List<Person> per =q.list();

for(Person k:per) {

System.out.println(k);

}

}

public static void getByAll2() {

SessionFactory sf = GetSessionFactory.con();

Session s = sf.openSession();

Query q=s.createQuery("from Person p");

List<Person> pers =q.list();

{

for(Person p:pers) {

if(p!=null&&p.getPpt()!=null&&!(p.getMob().isEmpty()&&p.getCou().isEmpty())) {

System.out.println(p+"\n"+p.getPpt()+"\n"+p.getMob()+"\n"+p.getCou());

}

}

}

}

public static void getByAll1() {

SessionFactory sf = GetSessionFactory.con();

Session s = sf.openSession();

Query q=s.createQuery("from Person p");

List<Person> pers =q.list();

for(Person g: pers) {

System.out.println(g);

if(g.getPpt()!=null) {

System.out.println(g.getPpt());

}

else {

System.out.println("No passport");

}

if(g.getMob().isEmpty()) {

System.out.println("No Mobile");

}else {

for(mobile mo:g.getMob()) {

System.out.println(mo);

}

if(g.getCou().isEmpty()) {

System.out.println("No Course");

}

else {

for(course co:g.getCou()) {

System.out.println(co);

}

}

}

}

}

public void getById1(int pid) {

SessionFactory sf = GetSessionFactory.con();

Session s = sf.openSession();

Person p = (Person)s.get(Person.class, pid);

System.out.println(p.getPid()+", "+p.getPname()+", "+p.getPage()+", "+p.getPqual());

Passport k = p.getPpt();

if(k!=null) {

System.out.println(k);

}

else {

System.out.println("No passport for"+p.getPname());

}

List<mobile> m = p.getMob();

if(m!=null) {

System.out.println(m);

for(mobile l:m) {

System.out.println(l);

}

}

else {

System.out.println("No mobile");

}

List<course> c = p.getCou();

if(c!=null) {

//System.out.println(c);

for(course d: c) {

System.out.println(d);

}

}

else {

System.out.println("No courses ");

}

}

public static void main(String[] args) {

All t = new All();

Person p =new Person();

p.setPname("SITA");

p.setPage(24);

p.setPqual("MCOM");

Passport q = new Passport();

q.setPnum("AB123Z");

q.setPexp("3090");

p.setPpt(q);

mobile m = new mobile();

m.setMsim("jio");

m.setMnum(986645342l);

mobile m1 = new mobile();

m1.setMsim("Airtel");

m1.setMnum(9786645342l);

mobile m2 = new mobile();

m2.setMsim("bsnl");

m2.setMnum(954645342l);

List<mobile> mob = new ArrayList<>();

mob.add(m);

mob.add(m1);

mob.add(m2);

p.setMob(mob);

course m3 = new course();

m3.setCname("Python");

m3.setCfee(30.5);

course m4 = new course();

m4.setCname("C");

m4.setCfee(40.5);

course m5 = new course();

m5.setCname("C++");

m5.setCfee(50.5);

List<course> cou = new ArrayList<>();

cou.add(m3);

cou.add(m4);

cou.add(m5);

p.setCou(cou);

//t.saveddata(p);

//t.updatedata(3);

//t.getById(3);

t.getByAll1();

//t.getByAll2();

//t.getById1(5);

}

}

Project: EmpHib

1)Employee.java

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "EmpHib")

public class Employee {

@Id

@GeneratedValue

private int eid;

private String ename;

private double esal;

private String edes;

private String eloc;

public Employee() {

}

public Employee(int eid, String ename, double esal, String edes, String eloc) {

super();

this.eid = eid;

this.ename = ename;

this.esal = esal;

this.edes = edes;

this.eloc = eloc;

}

public int getEid() {

return eid;

}

public void setEid(int eid) {

this.eid = eid;

}

public String getEname() {

return ename;

}

public void setEname(String ename) {

this.ename = ename;

}

public double getEsal() {

return esal;

}

public void setEsal(double esal) {

this.esal = esal;

}

public String getEdes() {

return edes;

}

public void setEdes(String edes) {

this.edes = edes;

}

public String getEloc() {

return eloc;

}

public void setEloc(String eloc) {

this.eloc = eloc;

}

@Override

public String toString() {

return "Employee [eid=" + eid + ", ename=" + ename + ", esal=" + esal + ", edes=" + edes + ", eloc=" + eloc

+ "]";

}

}

2)GetSessionFactory.java

package com.controller;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class GetSessionFactory {

public static SessionFactory con() {

Configuration con = new AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

return sf;

}

}

3)UnitTest.java

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** com.pojo.\*;

**public** **class** UnitTest {

**public** **static** **void** savedata(Employee p) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t= s.beginTransaction();

s.save(p);

t.commit();

s.close();

sf.close();

}

**public** **static** **void** getByID(**int** eid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

//Query p =s.createQuery("from Employee where eid =:eid");

Query p =s.createQuery("from Employee where eid >=:eid");

p.setParameter("eid", eid);

List<Employee> emp =p.list();

**for**(Employee em: emp) {

System.***out***.println(em);

}

}

**public** **static** **void** getByName(String ename) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Query p =s.createQuery("from Employee where ename =:ename");

p.setParameter("ename", ename);

List<Employee> emp =p.list();

**for**(Employee em: emp) {

System.***out***.println(em);

}

}

**public** **static** **void** getAll() {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Query p =s.createQuery("select e.eid, e.ename, e.esal from Employee e");

List<Object[]> emp =p.list();

**for**(Object[] em: emp) {

System.***out***.println(em[0]+", "+em[1]+", "+em[2]);

}

}

**public** **static** **void** update(**int** eid,String ename, **double** esal, String edes) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction k =s.beginTransaction();

Query p =s.createQuery("update Employee set ename=:ename, esal=:esal, edes=:edes where eid=:eid");

p.setParameter("ename", ename);

p.setParameter("esal", esal);

p.setParameter("edes", edes);

p.setParameter("eid", eid);

**int** result =p.executeUpdate();

System.***out***.println(result);

//s.update(p);

k.commit();

s.close();

sf.close();

}

**public** **static** **void** delete(String ename) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction k =s.beginTransaction();

Query p =s.createQuery("delete from Employee where ename=:ename");

p.setParameter("ename", ename);

**int** result =p.executeUpdate();

System.***out***.println(result);

//s.delete(p);

k.commit();

s.close();

sf.close();

}

**public** **static** **void** main(String[] args) {

UnitTest a = **new** UnitTest();

Employee e = **new** Employee();

e.setEname("anjan");

e.setEsal(5.6);

e.setEloc("chennai");

e.setEdes("jd");

//a.savedata(e);

//a.getByID(1);

//a.getByName("lilly");

//a.update(1,"anj", 77.7,"dev");

//a.delete("hj");

a.*getAll*();

}

}

**-------27/01/2024-------Saturday----------Class -11--------------------------------**

**Morning**

Topic: Aggregate Functions by HQL Queries Orderby(ASC, DESC), MIN, MAX, AVG, SUM, COUNT(\*), Groupby, Groupby and having

Always use having when you use aggregate functions. Do not use where in aggregate functions.

Projects: Hib\_AggregateFunctions

**Afternoon**

**Topic: Joins**

**Project: Hib\_Joins(One to One relation**[**"E:\html\Slokam\28 12 2023 O-O,O-M,M-O,M-M Relations.sql"**](file:///E:\html\Slokam\28%2012%202023%20O-O,O-M,M-O,M-M%20Relations.sql)**)**

**Package:**

**1)com.pojo**

**2)com.controller**

**Programs:**

**1)Person.java**

**2)Passport.java**

**3)getCon.java**

**4)OneToOne.java**

**1)Person.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.CascadeType;

import javax.persistence.\*;

@Entity

public class Person {

@Id

@GeneratedValue

private int pid;

private String pname;

private int page;

private String pqual;

@OneToOne(cascade =CascadeType.ALL)

@JoinColumn(name = "fkpid")

private Passport ppt;

@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt + "]";

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public String getPqual() {

return pqual;

}

public void setPqual(String pqual) {

this.pqual = pqual;

}

public Passport getPpt() {

return ppt;

}

public void setPpt(Passport ppt) {

this.ppt = ppt;

}

}

**2)Passport.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Passport {

@Id

@GeneratedValue

private int ppid;

@Override

public String toString() {

return "Passport [ppid=" + ppid + ", pnum=" + pnum + ", pexp=" + pexp + "]";

}

private String pnum;

private String pexp;

public int getPpid() {

return ppid;

}

public void setPpid(int ppid) {

this.ppid = ppid;

}

public String getPnum() {

return pnum;

}

public void setPnum(String pnum) {

this.pnum = pnum;

}

public String getPexp() {

return pexp;

}

public void setPexp(String pexp) {

this.pexp = pexp;

}

}

**3)getCon.java**

package com.controller;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class getCon {

public static Session con() {

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

return s;

}

}

**4)OneToOne.java**

package com.controller;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.pojo.\*;

public class OneToOne {

public static void Joins(int pid) {

Session r = getCon.con();

Query p =r.createQuery("from Person p inner join p.ppt where p.pid=:pid");

p.setParameter("pid", pid);

List<Object[]> emp =p.list();

for(Object[] em: emp) {

System.out.println(em[0]);

}

}

public static void Joinsbyselect() {

Session r = getCon.con();

Query p =r.createQuery("select p.pid, p.pname, pt.pnum, pt.pexp from Person p inner join p.ppt pt");

List<Object[]> l =p.list();

for(Object[] pe: l) {

Integer e =(Integer)pe[0];

String s = (String)pe[1];

String g = (String)pe[2];

String h = (String)pe[3];

System.out.println(e+", "+s+", "+g+", "+h);

}

}

public static void main(String[] args) {

OneToOne l = new OneToOne();

//l.Joins(2); /\*It is not using by using select\*/

l.Joinsbyselect();/\*It is using by using select\*/

}

}

**---------------------29/01/2024-------Monday--------Class -12------------------------**

**Morning**

**Topic: Joins**

* **One To One**
* **One To Many**
* **Many To Many**
* **Many To One**

**Project: HibnJoins**

**Package:**

**1)com.pojo**

**2)com.controller**

**Programs:**

**1)getCon.java**

**2)OneToOne.java**

**3)ManyToOne.java**

**4)ManyToMany.java**

**Afternoon**

**5)All.java**

**1)Person.java**

package com.pojo;

import java.util.List;

import javax.persistence.\*;

@Entity

public class Person {

@Id

@GeneratedValue

private int pid;

private String pname;

private int page;

private String pqual;

@OneToOne(cascade =CascadeType.ALL)

@JoinColumn(name = "fkpid")

private Passport ppt;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name = "fkpersid")

private List<mobile> mob;

@ManyToMany(cascade = CascadeType.ALL)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkpid")},

inverseJoinColumns = {@JoinColumn(name = "fkcid")})

private List<course> cou;

public Person() {

}

@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt

+ ", mob=" + mob + ", cou=" + cou + "]";

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public String getPqual() {

return pqual;

}

public void setPqual(String pqual) {

this.pqual = pqual;

}

public Passport getPpt() {

return ppt;

}

public void setPpt(Passport ppt) {

this.ppt = ppt;

}

public List<mobile> getMob() {

return mob;

}

public void setMob(List<mobile> mob) {

this.mob = mob;

}

public List<course> getCou() {

return cou;

}

public void setCou(List<course> cou) {

this.cou = cou;

}

public Person(int pid, String pname, int page, String pqual, Passport ppt, List<mobile> mob, List<course> cou) {

super();

this.pid = pid;

this.pname = pname;

this.page = page;

this.pqual = pqual;

this.ppt = ppt;

this.mob = mob;

this.cou = cou;

}

}

**2)Passport.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Passport {

@Id

@GeneratedValue

private int ppid;

@Override

public String toString() {

return "Passport [ppid=" + ppid + ", pnum=" + pnum + ", pexp=" + pexp + "]";

}

private String pnum;

private String pexp;

public int getPpid() {

return ppid;

}

public void setPpid(int ppid) {

this.ppid = ppid;

}

public String getPnum() {

return pnum;

}

public void setPnum(String pnum) {

this.pnum = pnum;

}

public String getPexp() {

return pexp;

}

public void setPexp(String pexp) {

this.pexp = pexp;

}

}

**3) mobile.java**

package com.pojo;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.OneToMany;

@Entity

public class mobile {

@Id

@GeneratedValue

private int mid;

private long mnum;

private String msim;

@Override

public String toString() {

return "mobile [mid=" + mid + ", mnum=" + mnum + ", msim=" + msim + "]";

}

public mobile(int mid, long mnum, String msim) {

super();

this.mid = mid;

this.mnum = mnum;

this.msim = msim;

}

public int getMid() {

return mid;

}

public void setMid(int mid) {

this.mid = mid;

}

public long getMnum() {

return mnum;

}

public void setMnum(long mnum) {

this.mnum = mnum;

}

public String getMsim() {

return msim;

}

public void setMsim(String msim) {

this.msim = msim;

}

public mobile() {

}

}

**4)course.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class course {

@Id

@GeneratedValue

private Integer cid;

private String cname;

private Double cfee;

public Integer getCid() {

return cid;

}

public void setCid(Integer cid) {

this.cid = cid;

}

public String getCname() {

return cname;

}

public void setCname(String cname) {

this.cname = cname;

}

public Double getCfee() {

return cfee;

}

public void setCfee(Double cfee) {

this.cfee = cfee;

}

public course(Integer cid, String cname, Double cfee) {

super();

this.cid = cid;

this.cname = cname;

this.cfee = cfee;

}

public course() {

}

@Override

public String toString() {

return "course [cid=" + cid + ", cname=" + cname + ", cfee=" + cfee + "]";

}

}

**5)getCon.java**

package com.controller;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class getCon {

public static Session con() {

Configuration con = new AnnotationConfiguration().configure("hibernate.cfg.xml");

SessionFactory sf =con.buildSessionFactory();

Session s=sf.openSession();

return s;

}

}

**6)OneToOne.java**

package com.controller;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

public class OneToOne {

public static void join() {

String query = "select p.pname, p.page, p.pqual, h.pnum, h.pexp from Person p inner join p.ppt h where p.pname ='Janaki'";

Session f = getCon.con();

Query p=f.createQuery(query);

List<Object[]>l=p.list();

for(Object[] d: l) {

System.out.println(d[0]+"---"+d[1]+"---"+d[2]+"---"+d[3]+"---"+d[4]);

}

}

public static void main(String[] args) {

OneToOne k = new OneToOne();

k.join();

}

}

**2)OneToMany.java**

package com.controller;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import com.pojo.Person;

public class OneToMany {

public static void join() {

String query = "select p.pname, p.page, p.pqual, h.mnum, h.msim from Person p inner join p.mob h where p.pname='Janaki'";

//[Many To One]

//String query = "select p.pname, p.page, p.pqual, h.mnum, h.msim from Person p inner join p.mob h where h.msim ='Jio'";

Session f = getCon.*con*();

Query p=f.createQuery(query);

List<Object[]>l=p.list();

for(Object[] d: l) {

System.*out*.println(d[0]+"---"+d[1]+"---"+d[2]+"---"+d[3]+"---"+d[4]);

}

}

public static void main(String[] args) {

OneToMany k = new OneToMany();

k.*join*();

}

}

**3)Many To Many.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**public** **class** ManyToMany {

**public** **static** **void** join() {

String query = "select p.pname, p.page, p.pqual, h.cname, h.cfee from Person p inner join p.cou h where h.cname ='Java'";

//String query = "select p.pname, p.page, p.pqual, h.cname, h.cfee from Person p inner join p.cou h where p.pname ='Janaki'";

Session f = getCon.*con*();

Query p=f.createQuery(query);

List<Object[]>l=p.list();

**for**(Object[] d: l) {

System.***out***.println(d[0]+"---"+d[1]+"---"+d[2]+"---"+d[3]+"---"+d[4]);

}

}

**public** **static** **void** main(String[] args) {

ManyToMany k = **new** ManyToMany();

k.*join*();

}}

**4)All.java**

package com.controller;

import java.util.List;

import java.util.HashSet;

import org.hibernate.Query;

import org.hibernate.Session;

public class All {

public static void join() {

String query = "select p.pname, p.page, p.pqual,t.pnum, t.pexp,m.mnum, m.msim, h.cname, h.cfee from Person p inner join p.ppt t inner join p.mob m inner join p.cou h where p.pname ='Janaki'";

Session f = getCon.*con*();

Query p=f.createQuery(query);

List<Object[]>l=p.list();

HashSet<String> k = new HashSet<>();

HashSet<String> k1 = new HashSet<>();

HashSet<String> k2 = new HashSet<>();

for(Object[] d: l)

{

k.add(d[0]+"---"+d[1]+"---"+d[2]+"-Person Passport--"+d[3]+"---"+d[4]);

}

for(Object[] d: l)

{

k1.add(d[5]+"-Mobile--"+d[6]);

}

for(Object[] d: l)

{

k2.add(d[7]+"-course--"+d[8]);

}

System.*out*.println(k);

System.*out*.println(k1);

System.*out*.println(k2);

HashSet<Object> n = new HashSet<>();

n.add(k);

n.add(k1);

n.add(k2);

System.*out*.println(n);

}

public static void main(String[] args) {

All k = new All();

k.*join*();

}

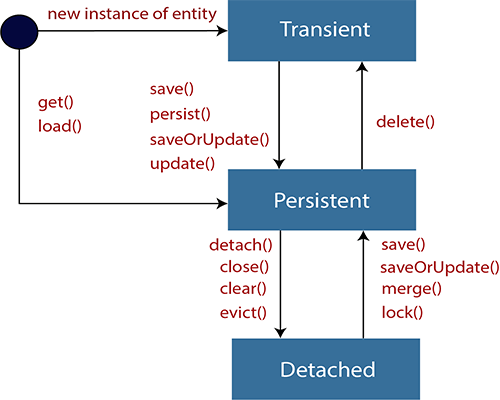
}

**----------------30/01/2024—Tuesday---------Class -13------------------------------**

**Topic: Hibnerate Lifecycle**

**Morning:**

**Notes Explanation, Hibnerate Lifecycle**



**Afternoon: Test the working of save(), persist(), saveOrUpdate(), update()**

**Project: HibnerateLifeCyCle(Transient, Persistent, Detached)**

**Packages:**

**1)com.pojo**

**2)com.controller**

**Programs:**

**1)Person.java**

**2)Passport.java**

**3)mobile.java**

**4)course.java**

**5)GetSessionFactory.java**

**6)unitperson.java**

**1)Person.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.CascadeType;

import java.util.List;

import javax.persistence.\*;

@Entity

public class Person {

@Id

@GeneratedValue

private int pid;

private String pname;

private int page;

private String pqual;

@OneToOne(cascade =CascadeType.*ALL*)

@JoinColumn(name = "fkpid")

private Passport ppt;

@OneToMany(cascade = CascadeType.*ALL*)

@JoinColumn(name = "fkpersid")

private List<mobile> mob;

@ManyToMany(cascade = CascadeType.*ALL*)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkpid")},

inverseJoinColumns = {@JoinColumn(name = "fkcid")})

private List<course> cou;

public Person() {

}

public Person(int pid, String pname, int page, String pqual, Passport ppt, List<mobile> mob, List<course> cou) {

super();

this.pid = pid;

this.pname = pname;

this.page = page;

this.pqual = pqual;

this.ppt = ppt;

this.mob = mob;

this.cou = cou;

}

/\*@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt + ", mob="

+ mob + ", cou=" + cou + "]";

}\*/

@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + "]";

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public String getPqual() {

return pqual;

}

public void setPqual(String pqual) {

this.pqual = pqual;

}

public Passport getPpt() {

return ppt;

}

public void setPpt(Passport ppt) {

this.ppt = ppt;

}

public List<mobile> getMob() {

return mob;

}

public void setMob(List<mobile> mob) {

this.mob = mob;

}

public List<course> getCou() {

return cou;

}

public void setCou(List<course> cou) {

this.cou = cou;

}

}

**2)Passport.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Passport {

@Id

@GeneratedValue

private int ppid;

private String pnum;

private String pexp;

@Override

public String toString() {

return "Passport [ppid=" + ppid + ", pnum=" + pnum + ", pexp=" + pexp + "]";

}

public int getPpid() {

return ppid;

}

public void setPpid(int ppid) {

this.ppid = ppid;

}

public String getPnum() {

return pnum;

}

public void setPnum(String pnum) {

this.pnum = pnum;

}

public String getPexp() {

return pexp;

}

public void setPexp(String pexp) {

this.pexp = pexp;

}

}

**3)mobile.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class mobile {

@Id

@GeneratedValue

private int mid;

private long mnum;

private String msim;

@Override

public String toString() {

return "mobile [mid=" + mid + ", mnum=" + mnum + ", msim=" + msim + "]";

}

public mobile() {

}

public mobile( long mnum, String msim) {

this.mnum = mnum;

this.msim = msim;

}

public int getMid() {

return mid;

}

public void setMid(int mid) {

this.mid = mid;

}

public long getMnum() {

return mnum;

}

public void setMnum(long mnum) {

this.mnum = mnum;

}

public String getMsim() {

return msim;

}

public void setMsim(String msim) {

this.msim = msim;

}

}

**4)course.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class course {

@Id

@GeneratedValue

private Integer cid;

private String cname;

private Double cfee;

public Integer getCid() {

return cid;

}

public void setCid(Integer cid) {

this.cid = cid;

}

public String getCname() {

return cname;

}

public void setCname(String cname) {

this.cname = cname;

}

public Double getCfee() {

return cfee;

}

public void setCfee(Double cfee) {

this.cfee = cfee;

}

public course(Integer cid, String cname, Double cfee) {

super();

this.cid = cid;

this.cname = cname;

this.cfee = cfee;

}

public course() {

}

@Override

public String toString() {

return "course [cid=" + cid + ", cname=" + cname + ", cfee=" + cfee + "]";

}

}

**5)GetSessionFactory.java**

package com.controller;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class GetSessionFactory {

public static SessionFactory con() {

Configuration con = new AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

return sf;

}

}

**6)unitperson.java**

package com.controller;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import com.pojo.\*;

public class unitperson {

public void savedata() {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t= s.beginTransaction();

Person p = new Person();

/\*p.setPname("Manju");

p.setPage(22);

p.setPqual("MCA");\*/

p.setPid(3);

p.setPname("Manju");

p.setPage(24);

p.setPqual("BSC");

/\*int r =(int) s.save(p);

System.out.println(r);\*/

//s.persist(p);

//s.saveOrUpdate(p);

//[save(), persist(), saveOrUpdate(), Update()]

t.commit();

s.close();

sf.close();

}

public void updatedata(int pid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

Person p =(Person)s.get(Person.class, pid);

p.setPname("SRIRAMA");

s.update(p);

t.commit();

s.close();

sf.close();

}

public static void main(String[] args) {

unitperson t = new unitperson();

t.savedata();

t.updatedata(2);

}

}

**-----------31/01/2024-------------------Wednesday---Class -14-------------**

**Topic:**

**Save(), Evict(), Merge(), clear(), Update()[Evict(), clear() will work with update only not with save]**

**Session.save():** Entity Objects will be saved in to database by calling the database connection. It just saves, after that you have duty to commit by t.commit();

**session.persist(entity):** This method tells Hibernate to remember the object for future saving in the database, usually done when the transaction is committed.

[**https://chat.openai.com/c/7198dbe5-64e5-4649-b70b-8b22989a80bd**](https://chat.openai.com/c/7198dbe5-64e5-4649-b70b-8b22989a80bd)

**Session.clear():** session will not be closed, but all objects within the session will be removed.

**Session.evict():** same as clear(), but only specified object will be removed from the session.

**public** **void** clear() {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t= s.beginTransaction();

Person p = **new** Person();

p.setPid(5);

p.setPname("Manjula");

p.setPage(22);

p.setPqual("BCOM");

Session s1 = sf.openSession();

Transaction t1 = s1.beginTransaction();

Person p1 = **new** Person();

p1.setPid(1);

p1.setPname("lly");

p1.setPage(23);

p1.setPqual("MA");

s.update(p);

s1.update(p1);

//s1.evict(p1);

//s.clear();

t.commit();

t1.commit();

s.close();

s1.close();

sf.close();

}

**1)unitperson.java**

**package** com.controller;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** java.util.List;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** com.pojo.\*;

**public** **class** unitperson {

**public** **void** savedata() {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t= s.beginTransaction();

Person p = **new** Person();

/\*p.setPname("Manju");

p.setPage(22);

p.setPqual("MCA");\*/

p.setPname("Manju");

p.setPage(24);

p.setPqual("BSC");

**int** r =(**int**) s.save(p);

System.***out***.println(r);

//s.persist(p);

//s.saveOrUpdate(p);

//[save(), persist(), saveOrUpdate(), Update()]

t.commit();

s.close();

sf.close();

}

**public** **void** updatedata(**int** pid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

//Transaction t = s.beginTransaction();

Person p =(Person)s.get(Person.**class**, pid);

p.setPname("SRIRAMA");

s.close();

SessionFactory sf1 = GetSessionFactory.*con*();

Session s1 = sf.openSession();

Transaction t = s1.beginTransaction();

Person p1 =(Person)s1.get(Person.**class**, pid);

p1.setPname("SRI");

//s1.update(p);

s1.merge(p);

t.commit();

s1.close();

sf1.close();

}

**public** **void** clear() {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t= s.beginTransaction();

Person p = **new** Person();

p.setPid(5);

p.setPname("Manjula");

p.setPage(22);

p.setPqual("BCOM");

Session s1 = sf.openSession();

Transaction t1 = s1.beginTransaction();

Person p1 = **new** Person();

p1.setPid(1);

p1.setPname("lly");

p1.setPage(23);

p1.setPqual("MA");

s.update(p);

s1.update(p1);

//s1.evict(p1);

//s.clear();

t.commit();

t1.commit();

s.close();

s1.close();

sf.close();

}

**public** **void** savedata1() {

//By save method evit(), clear(), close() are not working. But update they are working properly.

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t= s.beginTransaction();

Person p = **new** Person();

p.setPname("lilly flower");

p.setPage(25);

p.setPqual("MSC");

//s.evict(p);

s.save(p);

s.evict(p);

//s.clear();

//s.close();

t.commit();

//s.evict(p);

s.close();

sf.close();

}

**public** **static** **void** main(String[] args) {

unitperson t = **new** unitperson();

//t.savedata();

//t.updatedata(2);

//t.savedata1();

//t.updatedata(4);

t.clear();

}

}

**2) Person.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.CascadeType;

import java.util.List;

import javax.persistence.\*;

@Entity

public class Person {

@Id

@GeneratedValue

private int pid;

private String pname;

private int page;

private String pqual;

@OneToOne(cascade =CascadeType.*ALL*)

@JoinColumn(name = "fkpid")

private Passport ppt;

@OneToMany(cascade = CascadeType.*ALL*)

@JoinColumn(name = "fkpersid")

private List<mobile> mob;

@ManyToMany(cascade = CascadeType.*ALL*)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkpid")},

inverseJoinColumns = {@JoinColumn(name = "fkcid")})

private List<course> cou;

public Person() {

}

public Person(int pid, String pname, int page, String pqual, Passport ppt, List<mobile> mob, List<course> cou) {

super();

this.pid = pid;

this.pname = pname;

this.page = page;

this.pqual = pqual;

this.ppt = ppt;

this.mob = mob;

this.cou = cou;

}

/\*@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt + ", mob="

+ mob + ", cou=" + cou + "]";

}\*/

@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + "]";

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getPname() {

return pname;

}

public void setPname(String pname) {

this.pname = pname;

}

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public String getPqual() {

return pqual;

}

public void setPqual(String pqual) {

this.pqual = pqual;

}

public Passport getPpt() {

return ppt;

}

public void setPpt(Passport ppt) {

this.ppt = ppt;

}

public List<mobile> getMob() {

return mob;

}

public void setMob(List<mobile> mob) {

this.mob = mob;

}

public List<course> getCou() {

return cou;

}

public void setCou(List<course> cou) {

this.cou = cou;

}

}

-------------01/02/2024------Thursday------Noclass--------------------------------

**--------02/02/2024-----------Friday-----CLASS --15-------------------------------**

**TOPIC:**

Flush() method is discussed when

Pagination : When you open pagination Gmail t.ge

Bidirectional – if we save from passport then person should also change

Remove tostring(),

Now one to one

I will save person into passport

Topic: Bidirectional: OneToOne, ManyToOne, ManyToMany

Project: Hibnerate\_All\_Bidirectional

Packages:

1. com.pojo
2. com.controller

Programs:

1)Person.java

2)Passport.java

3)course.java

4)mobile.java

5)GetSessionFactory.java

6) OTO\_Bidirectional.java

7)MTO.java

8)MTM.java

AFTERNOON:

Topic: Criteria.

Criteria is similar to Query but Criteria increases the speed

Project: Hibnerate\_All\_Bidirectional

Packages: 1)com.pojo

2)com.controller

Programs:

1)GetSessionFactory.java

2)Criteriacl.java

1)Person.java

**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.OneToOne;

**import** javax.persistence.CascadeType;

**import** java.util.List;

**import** javax.persistence.\*;

@Entity

**public** **class** Person {

@Id

@GeneratedValue

**private** **int** pid;

**private** String pname;

**private** **int** page;

**private** String pqual;

@OneToOne(cascade =CascadeType.***ALL***)

@JoinColumn(name = "fkpid")

**private** Passport ppt;

@OneToMany(cascade = CascadeType.***ALL***)

@JoinColumn(name = "fkpersid")

**private** List<mobile> mob;

@ManyToMany(cascade = CascadeType.***ALL***)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkpid")},

inverseJoinColumns = {@JoinColumn(name = "fkcid")})

**private** List<course> cou;

**public** Person() {

}

**public** Person(**int** pid, String pname, **int** page, String pqual, Passport ppt, List<mobile> mob, List<course> cou) {

**super**();

**this**.pid = pid;

**this**.pname = pname;

**this**.page = page;

**this**.pqual = pqual;

**this**.ppt = ppt;

**this**.mob = mob;

**this**.cou = cou;

}

/\*@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt + ", mob="

+ mob + ", cou=" + cou + "]";

}\*/

/\*@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + "]";

}\*/

**public** **int** getPid() {

**return** pid;

}

**public** **void** setPid(**int** pid) {

**this**.pid = pid;

}

**public** String getPname() {

**return** pname;

}

**public** **void** setPname(String pname) {

**this**.pname = pname;

}

**public** **int** getPage() {

**return** page;

}

**public** **void** setPage(**int** page) {

**this**.page = page;

}

**public** String getPqual() {

**return** pqual;

}

**public** **void** setPqual(String pqual) {

**this**.pqual = pqual;

}

**public** Passport getPpt() {

**return** ppt;

}

**public** **void** setPpt(Passport ppt) {

**this**.ppt = ppt;

}

**public** List<mobile> getMob() {

**return** mob;

}

**public** **void** setMob(List<mobile> mob) {

**this**.mob = mob;

}

**public** List<course> getCou() {

**return** cou;

}

**public** **void** setCou(List<course> cou) {

**this**.cou = cou;

}

}

2)Passport.java

package com.pojo;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToOne;

@Entity

public class Passport {

@Id

@GeneratedValue

private int ppid;

private String pnum;

private String pexp;

/\*@Override

public String toString() {

return "Passport [ppid=" + ppid + ", pnum=" + pnum + ", pexp=" + pexp + "]";

}\*/

@OneToOne(cascade=CascadeType.ALL)

private Person pers;

public Person getPers() {

return pers;

}

public void setPers(Person pers) {

this.pers = pers;

}

public int getPpid() {

return ppid;

}

public void setPpid(int ppid) {

this.ppid = ppid;

}

public String getPnum() {

return pnum;

}

public void setPnum(String pnum) {

this.pnum = pnum;

}

public String getPexp() {

return pexp;

}

public void setPexp(String pexp) {

this.pexp = pexp;

}

}

3)mobile.java

package com.pojo;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

@Entity

public class mobile {

@Id

@GeneratedValue

private int mid;

private long mnum;

private String msim;

@Override

public String toString() {

return "mobile [mid=" + mid + ", mnum=" + mnum + ", msim=" + msim + "]";

}

public mobile() {

}

@ManyToOne(cascade =CascadeType.ALL)

private Person pm;

public Person getPm() {

return pm;

}

public void setPm(Person pm) {

this.pm = pm;

}

public mobile( long mnum, String msim) {

this.mnum = mnum;

this.msim = msim;

}

public int getMid() {

return mid;

}

public void setMid(int mid) {

this.mid = mid;

}

public long getMnum() {

return mnum;

}

public void setMnum(long mnum) {

this.mnum = mnum;

}

public String getMsim() {

return msim;

}

public void setMsim(String msim) {

this.msim = msim;

}

}

4)course.java

package com.pojo;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

@Entity

public class course {

@Id

@GeneratedValue

private Integer cid;

private String cname;

private Double cfee;

public Integer getCid() {

return cid;

}

public void setCid(Integer cid) {

this.cid = cid;

}

public String getCname() {

return cname;

}

public void setCname(String cname) {

this.cname = cname;

}

public Double getCfee() {

return cfee;

}

public void setCfee(Double cfee) {

this.cfee = cfee;

}

public course(Integer cid, String cname, Double cfee) {

super();

this.cid = cid;

this.cname = cname;

this.cfee = cfee;

}

public course() {

}

@Override

public String toString() {

return "course [cid=" + cid + ", cname=" + cname + ", cfee=" + cfee + "]";

}

@ManyToMany(cascade = CascadeType.ALL)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkcid")},

inverseJoinColumns = {@JoinColumn(name = "fkpid")})

private List<Person> per;

public List<Person> getPer() {

return per;

}

public void setPer(List<Person> per) {

this.per = per;

}

}

5)GetSessionFactory.java

package com.controller;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class GetSessionFactory {

public static SessionFactory con() {

Configuration con = new AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

return sf;

}

}

6) OTO\_Bidirectional.java

**package** com.controller;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** com.pojo.Passport;

**import** com.pojo.Person;

**public** **class** OTO\_Bidirectional {

**public** **static** **void** savedata(Passport p) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

s.save(p);

t.commit();

s.close();

}

**public** **static** **void** main(String[] args) {

OTO\_Bidirectional l = **new** OTO\_Bidirectional();

Passport pt = **new** Passport();

pt.setPpid(1);

pt.setPnum("sder123");

pt.setPexp("2025");

Person k = **new** Person();

//k.setPid(1);

k.setPname("klin");

k.setPage(55);

k.setPqual("MCA");

k.setPpt(pt);

pt.setPers(k);

l.*savedata*(pt);

}

}

7)MTO.java

package com.controller;

import java.util.ArrayList;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.pojo.Person;

import com.pojo.mobile;

public class MTO {

public void saveddata(Person p) {

SessionFactory sf = GetSessionFactory.con();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

s.save(p);

t.commit();

s.close();

}

public static void main(String[] args) {

MTO t = new MTO();

Person p =new Person();

p.setPname("Lucky");

p.setPage(23);

p.setPqual("BCA");

mobile m = new mobile();

m.setMsim("jio");

m.setMnum(986645342l);

m.setPm(p);

mobile m1 = new mobile();

m1.setMsim("Airtel");

m1.setMnum(9786645342l);

m1.setPm(p);

mobile m2 = new mobile();

m2.setMsim("bsnl");

m2.setMnum(954645342l);

m2.setPm(p);

List<mobile> mob = new ArrayList<>();

mob.add(m);

mob.add(m1);

mob.add(m2);

p.setMob(mob);

t.saveddata(p);

}

}

8)MTM.java

package com.controller;

import java.util.ArrayList;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.pojo.Person;

import com.pojo.course;

public class MTM {

public void saveddata(course p) {

SessionFactory sf = GetSessionFactory.con();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

s.save(p);

t.commit();

s.close();

}

public static void main(String[] args) {

MTM j = new MTM();

course m = new course();

m.setCid(1);

m.setCname("JAVA");

m.setCfee(30.5);

Person h = new Person();

h.setPname("AAA");

h.setPage(21);

h.setPqual("BTECH");

Person h1 = new Person();

h1.setPname("BBB");

h1.setPage(22);

h1.setPqual("MTECH");

Person h2 = new Person();

h2.setPname("CCC");

h2.setPage(23);

h2.setPqual("MCA");

List<Person> per = new ArrayList<>();

per.add(h);

per.add(h1);

per.add(h2);

m.setPer(per);

j.saveddata(m);

//t.updatedata(1);

//t.getById1(4);

//t.getByAll();

}

}

AFTERNOON:

2)Criteriacl.java

package com.controller;

import java.util.List;

import org.hibernate.Criteria;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.criterion.Order;

import org.hibernate.criterion.Restrictions;

import com.pojo.Person;

public class Criteriacl {

public static void main(String[] args) {

SessionFactory sf = GetSessionFactory.con();

Session s = sf.openSession();

Criteria c= s.createCriteria(Person.class);

//c.setFirstResult(3);

//c.setMaxResults(2);

//c.add(Restrictions.between("page", 23, 60));

//c.add(Restrictions.gt("page", 50));

//c.add(Restrictions.lt("page", 23));

//c.add(Restrictions.like("pname", "%c%"));

//c.addOrder(Order.asc("page"));

/\*String dt[] = {"Lucky","AAA", "CCC"};

c.add(Restrictions.in("pname", dt));\*/

//c.add(Restrictions.ge("page", 50));

//c.add(Restrictions.le("page", 25));

List<Person> per =c.list();

for(Person pers: per) {

System.out.println(pers);

}

}

}

**------------------03/02/2024-------------Saturday----------Class -16------------**

Topic: OneToOne, OneToMany, ManyToMany Relations Using Criteria

Project: Hibnerate\_Criteria\_OTO,OTM,MTM1

Packages:

1)com.pojo

2)com.controller

Programs:

1)Person.java

2)Passport.java

3)course.java

4)mobile.java

5) GetSessionFactory.java

6) OTO\_Criteria.java

7) OTM\_Criteria.java

8) MTM\_Criteria.java

**1)Person.java**

**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.OneToOne;

**import** javax.persistence.CascadeType;

**import** java.util.List;

**import** javax.persistence.\*;

@Entity

**public** **class** Person {

@Id

@GeneratedValue

**private** **int** pid;

**private** String pname;

**private** **int** page;

**private** String pqual;

@OneToOne(cascade =CascadeType.***ALL***)

@JoinColumn(name = "fkpid")

**private** Passport ppt;

@OneToMany(cascade = CascadeType.***ALL***)

@JoinColumn(name = "fkpersid")

**private** List<mobile> mob;

@ManyToMany(cascade = CascadeType.***ALL***)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkpid")},

inverseJoinColumns = {@JoinColumn(name = "fkcid")})

**private** List<course> cou;

**public** Person() {

}

**public** Person(**int** pid, String pname, **int** page, String pqual, Passport ppt, List<mobile> mob, List<course> cou) {

**super**();

**this**.pid = pid;

**this**.pname = pname;

**this**.page = page;

**this**.pqual = pqual;

**this**.ppt = ppt;

**this**.mob = mob;

**this**.cou = cou;

}

@Override

**public** String toString() {

**return** "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt + ", mob="

+ mob + ", cou=" + cou + "]";

}

/\*@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + "]";

}\*/

**public** **int** getPid() {

**return** pid;

}

**public** **void** setPid(**int** pid) {

**this**.pid = pid;

}

**public** String getPname() {

**return** pname;

}

**public** **void** setPname(String pname) {

**this**.pname = pname;

}

**public** **int** getPage() {

**return** page;

}

**public** **void** setPage(**int** page) {

**this**.page = page;

}

**public** String getPqual() {

**return** pqual;

}

**public** **void** setPqual(String pqual) {

**this**.pqual = pqual;

}

**public** Passport getPpt() {

**return** ppt;

}

**public** **void** setPpt(Passport ppt) {

**this**.ppt = ppt;

}

**public** List<mobile> getMob() {

**return** mob;

}

**public** **void** setMob(List<mobile> mob) {

**this**.mob = mob;

}

**public** List<course> getCou() {

**return** cou;

}

**public** **void** setCou(List<course> cou) {

**this**.cou = cou;

}

}

**2)Passport.java**

package com.pojo;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToOne;

@Entity

public class Passport {

@Id

@GeneratedValue

private int ppid;

private String pnum;

private String pexp;

@Override

public String toString() {

return "Passport [ppid=" + ppid + ", pnum=" + pnum + ", pexp=" + pexp + "]";

}

@OneToOne(cascade=CascadeType.ALL)

private Person pers;

public Person getPers() {

return pers;

}

public void setPers(Person pers) {

this.pers = pers;

}

public int getPpid() {

return ppid;

}

public void setPpid(int ppid) {

this.ppid = ppid;

}

public String getPnum() {

return pnum;

}

public void setPnum(String pnum) {

this.pnum = pnum;

}

public String getPexp() {

return pexp;

}

public void setPexp(String pexp) {

this.pexp = pexp;

}

}

**3)mobile.java**

**package** com.pojo;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.ManyToOne;

@Entity

**public** **class** mobile {

@Id

@GeneratedValue

**private** **int** mid;

**private** **long** mnum;

**private** String msim;

@Override

**public** String toString() {

**return** "mobile [mid=" + mid + ", mnum=" + mnum + ", msim=" + msim + "]";

}

**public** mobile() {

}

@ManyToOne(cascade =CascadeType.***ALL***)

**private** Person pm;

**public** Person getPm() {

**return** pm;

}

**public** **void** setPm(Person pm) {

**this**.pm = pm;

}

**public** mobile( **long** mnum, String msim) {

**this**.mnum = mnum;

**this**.msim = msim;

}

**public** **int** getMid() {

**return** mid;

}

**public** **void** setMid(**int** mid) {

**this**.mid = mid;

}

**public** **long** getMnum() {

**return** mnum;

}

**public** **void** setMnum(**long** mnum) {

**this**.mnum = mnum;

}

**public** String getMsim() {

**return** msim;

}

**public** **void** setMsim(String msim) {

**this**.msim = msim;

}

}

**4)course.java**

**package** com.pojo;

**import** java.util.List;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.JoinColumn;

**import** javax.persistence.JoinTable;

**import** javax.persistence.ManyToMany;

@Entity

**public** **class** course {

@Id

@GeneratedValue

**private** Integer cid;

**private** String cname;

**private** Double cfee;

**public** Integer getCid() {

**return** cid;

}

**public** **void** setCid(Integer cid) {

**this**.cid = cid;

}

**public** String getCname() {

**return** cname;

}

**public** **void** setCname(String cname) {

**this**.cname = cname;

}

**public** Double getCfee() {

**return** cfee;

}

**public** **void** setCfee(Double cfee) {

**this**.cfee = cfee;

}

**public** course(Integer cid, String cname, Double cfee) {

**super**();

**this**.cid = cid;

**this**.cname = cname;

**this**.cfee = cfee;

}

**public** course() {

}

@Override

**public** String toString() {

**return** "course [cid=" + cid + ", cname=" + cname + ", cfee=" + cfee + "]";

}

@ManyToMany(cascade = CascadeType.***ALL***)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkcid")},

inverseJoinColumns = {@JoinColumn(name = "fkpid")})

**private** List<Person> per;

**public** List<Person> getPer() {

**return** per;

}

**public** **void** setPer(List<Person> per) {

**this**.per = per;

}

}

**5)GetSessionFactory.java**

**package** com.controller;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.AnnotationConfiguration;

**import** org.hibernate.cfg.Configuration;

**public** **class** GetSessionFactory {

**public** **static** SessionFactory con() {

Configuration con = **new** AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

**return** sf;

}

}

**6) OTO\_Criteria.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.criterion.Restrictions;

**import** com.pojo.Person;

**public** **class** OTO\_Criteria {

//Purpose of criteria: Only fetching the data

//We have done fetching from single table.

//Now we will test how data can be fetched when there are multiple tables and there is a join relation in between them.

**public** **static** **void** main(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

/\*Criteria c =s.createCriteria(Person.class, "p");

c.createAlias("p.ppt", "pt");

c.add(Restrictions.eq("pt.pnum","sder123"));\*/

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.ppt", "pt");

c.add(Restrictions.*eq*("p.pname","klin"));

List<Person> data = c.list();

**for**(Person n: data) {

System.***out***.println(n);

}

}

}

**7)OTM\_Criteria.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.criterion.Restrictions;

**import** com.pojo.Person;

**public** **class** OTM\_Criteria {

**public** **static** **void** main(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

/\*Criteria c =s.createCriteria(Person.class, "p");

c.createAlias("p.mob", "m");

c.add(Restrictions.eq("p.pname","Lucky"));\*/

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.mob", "m");

c.add(Restrictions.*eq*("m.msim","jio"));

List<Person> data = c.list();

**for**(Person n: data) {

System.***out***.println(n);

}

}

}

**8)MTM\_Criteraia.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.criterion.Restrictions;

**import** com.pojo.Person;

**public** **class** MTM\_Criteria {

**public** **static** **void** main(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.cou", "c");

c.add(Restrictions.*eq*("p.pname","CCC"));

//ManyTOMany

/\*Criteria c =s.createCriteria(Person.class, "p");

c.createAlias("p.cou", "c");

c.add(Restrictions.eq("c.cname","Java"));\*/

List<Person> data = c.list();

**for**(Person n: data) {

System.***out***.println(n);

}

}

}

**----------05/02/2024---------------Monday---------Class -17--------------------**

Topic:

We applied OneToOne, OneToMany, ManyToMany with joins.

In yesterday class only we learnt about Restrictions. If we want we can apply again such as orderby, greaterthan, lessthan.

PROJECTIONS: Instead of getting entire object we can get specific columns(Properties) using projections.

projections with and or in hibernate criteria example

Where we will use Criterion?

Where we will use criteria?

Where we will use logical expression?

Where we will get projectionList?

Projects:

* Hibnerate\_Restrictions\_with\_joins
* Hibnerate Projections

Packages:

Project-1: Hibnerate\_Restrictions\_with\_joins

* com.pojo
* Person.java
* Passport.java
* mobile.java
* course.java
* com.controller
* GetSessionFactory.java
* OTO\_Criteria.java
* OTM\_Criteria.java
* MTM\_Criteria.java

Project-2: Hibnerate Projections

* com.pojo
* Person.java
* Passport.java
* mobile.java
* course.java
* com.controller
* GetSessionFactory.java
* ProjectionIntroduction.java
* Projectionwith\_AND\_OR.java

Package -1:com.pojo(The classes in com.pojo is same in both the projects)

**1)Person.java**

**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.OneToOne;

**import** javax.persistence.CascadeType;

**import** java.util.List;

**import** javax.persistence.\*;

@Entity

**public** **class** Person {

@Id

@GeneratedValue

**private** **int** pid;

**private** String pname;

**private** **int** page;

**private** String pqual;

@OneToOne(cascade =CascadeType.***ALL***)

@JoinColumn(name = "fkpid")

**private** Passport ppt;

@OneToMany(cascade = CascadeType.***ALL***)

@JoinColumn(name = "fkpersid")

**private** List<mobile> mob;

@ManyToMany(cascade = CascadeType.***ALL***)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkpid")},

inverseJoinColumns = {@JoinColumn(name = "fkcid")})

**private** List<course> cou;

**public** Person() {

}

**public** Person(**int** pid, String pname, **int** page, String pqual, Passport ppt, List<mobile> mob, List<course> cou) {

**super**();

**this**.pid = pid;

**this**.pname = pname;

**this**.page = page;

**this**.pqual = pqual;

**this**.ppt = ppt;

**this**.mob = mob;

**this**.cou = cou;

}

@Override

**public** String toString() {

**return** "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt + ", mob="

+ mob + ", cou=" + cou + "]";

}

/\*@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + "]";

}\*/

**public** **int** getPid() {

**return** pid;

}

**public** **void** setPid(**int** pid) {

**this**.pid = pid;

}

**public** String getPname() {

**return** pname;

}

**public** **void** setPname(String pname) {

**this**.pname = pname;

}

**public** **int** getPage() {

**return** page;

}

**public** **void** setPage(**int** page) {

**this**.page = page;

}

**public** String getPqual() {

**return** pqual;

}

**public** **void** setPqual(String pqual) {

**this**.pqual = pqual;

}

**public** Passport getPpt() {

**return** ppt;

}

**public** **void** setPpt(Passport ppt) {

**this**.ppt = ppt;

}

**public** List<mobile> getMob() {

**return** mob;

}

**public** **void** setMob(List<mobile> mob) {

**this**.mob = mob;

}

**public** List<course> getCou() {

**return** cou;

}

**public** **void** setCou(List<course> cou) {

**this**.cou = cou;

}

}

**2)Passport.java**

**package** com.pojo;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.OneToOne;

@Entity

**public** **class** Passport {

@Id

@GeneratedValue

**private** **int** ppid;

**private** String pnum;

**private** String pexp;

@Override

**public** String toString() {

**return** "Passport [ppid=" + ppid + ", pnum=" + pnum + ", pexp=" + pexp + "]";

}

@OneToOne(cascade=CascadeType.***ALL***)

**private** Person pers;

**public** Person getPers() {

**return** pers;

}

**public** **void** setPers(Person pers) {

**this**.pers = pers;

}

**public** **int** getPpid() {

**return** ppid;

}

**public** **void** setPpid(**int** ppid) {

**this**.ppid = ppid;

}

**public** String getPnum() {

**return** pnum;

}

**public** **void** setPnum(String pnum) {

**this**.pnum = pnum;

}

**public** String getPexp() {

**return** pexp;

}

**public** **void** setPexp(String pexp) {

**this**.pexp = pexp;

}

}

**3)course.java**

package com.pojo;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

@Entity

public class course {

@Id

@GeneratedValue

private Integer cid;

private String cname;

private Double cfee;

public Integer getCid() {

return cid;

}

public void setCid(Integer cid) {

this.cid = cid;

}

public String getCname() {

return cname;

}

public void setCname(String cname) {

this.cname = cname;

}

public Double getCfee() {

return cfee;

}

public void setCfee(Double cfee) {

this.cfee = cfee;

}

public course(Integer cid, String cname, Double cfee) {

super();

this.cid = cid;

this.cname = cname;

this.cfee = cfee;

}

public course() {

}

@Override

public String toString() {

return "course [cid=" + cid + ", cname=" + cname + ", cfee=" + cfee + "]";

}

@ManyToMany(cascade = CascadeType.ALL)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkcid")},

inverseJoinColumns = {@JoinColumn(name = "fkpid")})

private List<Person> per;

public List<Person> getPer() {

return per;

}

public void setPer(List<Person> per) {

this.per = per;

}

}

**4)mobile.java**

**package** com.pojo;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.ManyToOne;

@Entity

**public** **class** mobile {

@Id

@GeneratedValue

**private** **int** mid;

**private** **long** mnum;

**private** String msim;

@Override

**public** String toString() {

**return** "mobile [mid=" + mid + ", mnum=" + mnum + ", msim=" + msim + "]";

}

**public** mobile() {

}

@ManyToOne(cascade =CascadeType.***ALL***)

**private** Person pm;

**public** Person getPm() {

**return** pm;

}

**public** **void** setPm(Person pm) {

**this**.pm = pm;

}

**public** mobile( **long** mnum, String msim) {

**this**.mnum = mnum;

**this**.msim = msim;

}

**public** **int** getMid() {

**return** mid;

}

**public** **void** setMid(**int** mid) {

**this**.mid = mid;

}

**public** **long** getMnum() {

**return** mnum;

}

**public** **void** setMnum(**long** mnum) {

**this**.mnum = mnum;

}

**public** String getMsim() {

**return** msim;

}

**public** **void** setMsim(String msim) {

**this**.msim = msim;

}

}

**5)GetSessionFactory.java**

**package** com.controller;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.AnnotationConfiguration;

**import** org.hibernate.cfg.Configuration;

**public** **class** GetSessionFactory {

**public** **static** SessionFactory con() {

Configuration con = **new** AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

**return** sf;

}

}

So we applied joins on single table and also we applied with joins

By Restrictions we can apply orderby, greaterthan, lessthan restrictions also

**6) OTO\_Criteria.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.criterion.Restrictions;

**import** com.pojo.Person;

**public** **class** OTO\_Criteria {

**public** **static** **void** main(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.ppt", "pt");

c.add(Restrictions.*eq*("pt.pnum","sder123"));

/\*Criteria c =s.createCriteria(Person.class, "p");

c.createAlias("p.ppt", "pt");

c.add(Restrictions.eq("p.pname","klin"));\*/

List<Person> data = c.list();

**for**(Person n: data) {

System.***out***.println(n);

}

}

}

**7) OTM\_Criteria.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.criterion.Restrictions;

**import** com.pojo.Person;

**public** **class** OTM\_Criteria {

**public** **static** **void** main(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

/\*Criteria c =s.createCriteria(Person.class, "p");

c.createAlias("p.mob", "m");

c.add(Restrictions.eq("p.pname","Lucky"));\*/

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.mob", "m");

c.add(Restrictions.*eq*("m.msim","jio"));

List<Person> data = c.list();

**for**(Person n: data) {

System.***out***.println(n);

}

}

}

**8)MTM\_Criteria.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.criterion.Restrictions;

**import** com.pojo.Person;

**public** **class** MTM\_Criteria {

**public** **static** **void** main(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

/\*Criteria c =s.createCriteria(Person.class, "p");

c.createAlias("p.cou", "c");

c.add(Restrictions.eq("p.pname","CCC"));\*/

//ManyTOMany

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.cou", "c");

c.add(Restrictions.*eq*("c.cname","Java"));

List<Person> data = c.list();

**for**(Person n: data) {

System.***out***.println(n);

}

}

}

**Project-2: Hibnerate Projections**

**Package-1: com.pojo is same as above**

**Same programs**

**1)Person.java**

**2)Passport.java**

**3)mobile.java**

**4)course.java**

**Package -2:com.controller**

**PROJECTIONS:** Instead of getting entire object we can get specific columns(Properties) using projections.

<http://javainsimpleway.com/criteria-with-projection/>

projections with and or in hibernate criteria example

<https://www.tutorialspoint.com/hibernate/hibernate_criteria_queries.htm>

Criteria is a interface – It is used fetch the data

What we can add in to criteria?

We can add restrictions,restrictions can be taken as greaterthan, lessthan

**5) GetSessionFactory.java**

**package** com.controller;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.AnnotationConfiguration;

**import** org.hibernate.cfg.Configuration;

**public** **class** GetSessionFactory {

**public** **static** SessionFactory con() {

Configuration con = **new** AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

**return** sf;

}

}

**6)ProjectionIntroduction.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.criterion.Projection;

**import** org.hibernate.criterion.ProjectionList;

**import** org.hibernate.criterion.Projections;

**import** org.hibernate.criterion.Restrictions;

**import** com.pojo.Person;

**public** **class** ProjectionIntroduction {

//This gives person ,passport,mobile,course data only

**public** **static** **void** main1(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.ppt", "pt");

c.createAlias("p.mob", "m");

c.createAlias("p.cou", "c");

List<Person> data = c.list();

**for**(Person n: data) {

System.***out***.println(n);

}

}

**public** **static** **void** main(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.ppt", "pt");

c.createAlias("p.mob", "m");

c.createAlias("p.cou", "c");

//Now I want to keep restrictions on data which have all passport, mobile, courses

ProjectionList pl =Projections.*projectionList*();

pl.add(Projections.*property*("p.pname"));

pl.add(Projections.*property*("pt.pnum"));

pl.add(Projections.*property*("m.msim"));

pl.add(Projections.*property*("c.cfee"));

pl.add(Projections.*property*("c.cname"));

c.setProjection(pl);

c.add(Restrictions.*gt*("c.cfee",30.0));

List<Object[]> data = c.list();

**for**(Object[] n: data) {

System.***out***.println(n[0]+"---"+n[1]+"-----"+n[2]+"---"+n[3]+"----"+n[4]);

}

}

}

If we take individual restriction, it will return criterion, we will add this criterion to logicalExpression.

Again this logicalExpression is added to criteria.

Logical Expression is used when we need to apply and/Or operation on multiple restrictions.

**7) Projectionwith\_AND\_OR.java**

**package** com.controller;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.criterion.Criterion;

**import** org.hibernate.criterion.LogicalExpression;

**import** org.hibernate.criterion.Projection;

**import** org.hibernate.criterion.ProjectionList;

**import** org.hibernate.criterion.Projections;

**import** org.hibernate.criterion.Restrictions;

**import** com.pojo.Person;

**public** **class** Projectionwith\_AND\_OR {

**public** **static** **void** main(String[] args) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Criteria c =s.createCriteria(Person.**class**, "p");

c.createAlias("p.ppt", "pt");

c.createAlias("p.mob", "m");

c.createAlias("p.cou", "c");

ProjectionList pl =Projections.*projectionList*();

pl.add(Projections.*property*("p.pname"));

pl.add(Projections.*property*("pt.pnum"));

pl.add(Projections.*property*("m.msim"));

pl.add(Projections.*property*("c.cfee"));

pl.add(Projections.*property*("c.cname"));

c.setProjection(pl);

//c.add(Restrictions.gt("c.cfee",30.0));

//c.add(Restrictions.gt("m.msim","jio"));

//Now I am going to perform and & or,

Criterion cfee = Restrictions.*gt*("c.cfee", 30.0);

Criterion msim = Restrictions.*ilike*("m.msim","jio");

// ,by using Logical Expression

LogicalExpression andExp = Restrictions.*and*(cfee, msim);

c.add(andExp );

// LogicalExpression orExp = Restrictions.or(cfee, msim);

//c.add(orExp);

List<Object[]> data = c.list();

**for**(Object[] n: data) {

System.***out***.println(n[0]+"---"+n[1]+"-----"+n[2]+"---"+n[3]+"----"+n[4]);

}

}

}

Projection is used when we want the particular column name information.

So, we have learnt about criteria, criterion, restriction, LogicalExpression, Projections, ProjectionList

NextTopic Projections with Aggregations---------

Aggregations can be applied on single table

**---------06/02/2024-------Tuesday--------------Class -18------------------------------**

**Morning**

**Topic: Hibnerate Projections with Aggregations**

* **Aggregation Functions Using the Queries(Query) and without using Queries(Criteria)**

**Forward Engineering: From ui part go back. How is UI part in that we will design the tables.**

**Reverse Engineering: If you know the functionality well. Tables and data is available from that we are going to design UI. Fom the project we need to design the UI.**

**Project: Hibnerate Projections with Aggregations**

**Packages:**

**1)com.pojo**

**2)com.controller**

**Programs:**

**1)Person.java**

**2)Passport.java**

**3)mobile.java**

**4)course.java**

**5)** **ProjectionAggregationFunctions.java**

AFTERNOON:

TOPIC:

1)One Pojo Properties should be inserted into Two Tables

2)Two Pojo Class Properties should be inserted into One Table

Project: HibnerateOnepojoToManyTables,ManypojoToOneTables

Packages:

1)com.pojo

2)com.controller

Programs:

1)Address.java

2) OneClassIntoTwoTables.java

3)One.java

4)Two.java

5) TwoClassesIntoSingleTable.java

**Morning**

**Programs:**

1)Person.java

2)Passport.java

3)mobile.java These Pojo Classes are same as above class . See 5/2/24

4)course.java

**5)** **ProjectionAggregationFunctions.java**

/\*--------Aggregation Functions Using the Queries(Query) and without using Queries(Criteria)-------\*/

package com.controller;

import java.util.List;

import org.apache.poi.sl.usermodel.PlaceableShape;

import org.hibernate.Criteria;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.criterion.Order;

import org.hibernate.criterion.Projection;

import org.hibernate.criterion.ProjectionList;

import org.hibernate.criterion.Projections;

import org.hibernate.criterion.Restrictions;

import org.springframework.security.access.method.P;

import com.pojo.Person;

public class ProjectionAggregationFunctions {

public static void AggusingQueriesOnlycount() {

//This method is by using Queries

SessionFactory sf = GetSessionFactory.con();

Session s =sf.openSession();

Query p = s.createQuery("select count(\*)from Person ");

List<Object[]> data = p.list();

System.out.println(data);

}

public static void AggusingQueriesAll() {

//This method is by using Queries

SessionFactory sf = GetSessionFactory.con();

Session s =sf.openSession();

Query p = s.createQuery("select count(\*), max(page),min(page), sum(page), avg(page) from Person p group by page having page>=40 order by page desc ");

List<Object[]> data = p.list();

for(Object[] n: data) {

System.out.println("Count: "+n[0]+"\n"+"Maximum Age: "+n[1]+"\n"+"Minimum Age: "+n[2]+"\n"+"Sum of All ages: "+n[3]+"\n"+"Avg of all ages: "+n[4]);

}

}

public static void AggUsingCriteria() {

//This method is without using Queries. we will use Criteria.

SessionFactory sf = GetSessionFactory.con();

Session s =sf.openSession();

Criteria c =s.createCriteria(Person.class, "p");

//Now I want to keep restrictions on data which have all passport, mobile, courses

ProjectionList pl =Projections.projectionList();

//pl.add(Projections.countDistinct("page"));

pl.add(Projections.max("page"));

pl.add(Projections.min("page"));

pl.add(Projections.avg("page"));

pl.add(Projections.sum("page"));

pl.add(Projections.count("page"));

pl.add(Projections.groupProperty("page"));

c.setProjection(pl);

c.addOrder(Order.desc("page"));

c.add(Restrictions.gt("page",30));

List<Object[]> data = c.list();

//System.out.println(data);

for(Object[] n: data) {

System.out.println(n[0]+"---"+n[1]+"-----"+n[2]+"---"+n[3]+"----"+n[4]+"---"+n[5]);

}

}

public static void main(String[] args) {

ProjectionAggregationFunctions k = new ProjectionAggregationFunctions();

//k.AggusingQueriesOnlycount();

//k.AggusingQueriesAll();

k.AggUsingCriteria();

}

}

AFTERNOON:

TOPIC:

1)One Pojo Properties should be inserted into Two Tables

2)Two Pojo Class Properties should be inserted into One Table

Project: HibnerateOnepojoToManyTables,ManypojoToOneTables

Packages:

1)com.pojo

2)com.controller

Programs:

1)Address.java

2) OneClassIntoTwoTables.java

3)One.java

4)Two.java

5) TwoClassesIntoSingleTable.java

**1)Address.java**

package com.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.SecondaryTable;

import javax.persistence.Table;

@Entity

@Table(name ="Physicaladdress")

@SecondaryTable(name = "Mailingaddress")

public class Address {

@Id

@GeneratedValue

private int aid;

@Column(table ="Physicaladdress")

private String plocation;

@Column(table ="Physicaladdress")

private String pstreet;

@Column(table ="Physicaladdress")

private String pphone;

@Column(table ="Mailingaddress")

private String mlocation;

@Column(table ="Mailingaddress")

private String mstreet;

@Column(table ="Mailingaddress")

private String mphone;

public Address(String plocation, String pstreet, String pphone, String mlocation, String mstreet, String mphone) {

super();

this.plocation = plocation;

this.pstreet = pstreet;

this.pphone = pphone;

this.mlocation = mlocation;

this.mstreet = mstreet;

this.mphone = mphone;

}

@Override

public String toString() {

return "Address [aid=" + aid + ", plocation=" + plocation + ", pstreet=" + pstreet + ", pphone=" + pphone

+ ", mlocation=" + mlocation + ", mstreet=" + mstreet + ", mphone=" + mphone + "]";

}

public int getAid() {

return aid;

}

public void setAid(int aid) {

this.aid = aid;

}

public String getPlocation() {

return plocation;

}

public void setPlocation(String plocation) {

this.plocation = plocation;

}

public String getPstreet() {

return pstreet;

}

public void setPstreet(String pstreet) {

this.pstreet = pstreet;

}

public String getPphone() {

return pphone;

}

public void setPphone(String pphone) {

this.pphone = pphone;

}

public String getMlocation() {

return mlocation;

}

public void setMlocation(String mlocation) {

this.mlocation = mlocation;

}

public String getMstreet() {

return mstreet;

}

public void setMstreet(String mstreet) {

this.mstreet = mstreet;

}

public String getMphone() {

return mphone;

}

public void setMphone(String mphone) {

this.mphone = mphone;

}

public Address() {

}

}

**2)OneClassIntoTwoTables.java**

package com.controller;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.pojo.Address;

public class OneClassIntoTwoTables {

public static void savemethod() {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

Address p = new Address("Hyd","srnagar","1234","scd","aa","1234");

s.save(p);

t.commit();

s.close();

}

public static void GETBYID(int aid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

Address l =(Address)s.get(Address.class, aid);

System.*out*.println(l);

}

public void getById1(int pid) {

SessionFactory sf = GetSessionFactory.*con*();

Session d = sf.openSession();

Address p = (Address) d.get(Address.class, pid);

if (p != null) {

if (p.getPlocation() != null && p.getPstreet() != null && p.getPphone() != null) {

System.*out*.println("Physical Address: ");

System.*out*.println("Location: " + p.getPlocation());

System.*out*.println("Street: " + p.getPstreet());

System.*out*.println("Phone: " + p.getPphone());

} else {

System.*out*.println("No Physical Address ");

}

if (p.getMlocation() != null && p.getMstreet() != null && p.getMphone() != null) {

System.*out*.println("Mailing Address: ");

System.*out*.println("Location: " + p.getMlocation());

System.*out*.println("Street: " + p.getMstreet());

System.*out*.println("Phone: " + p.getMphone());

} else {

System.*out*.println("No Mailing Address ");

}

} else {

System.*out*.println("No Address Found");

}

d.close();

}

public void getById2(int pid) {

SessionFactory sf = GetSessionFactory.*con*();

Session d = sf.openSession();

Address p = (Address) d.get(Address.class, pid);

if (p != null) {

if (p.getPlocation() != null || p.getPstreet() != null || p.getPphone() != null) {

System.*out*.println("Physical Address: ");

if (p.getPlocation() != null) {

System.*out*.println("Location: " + p.getPlocation());

}

if (p.getPstreet() != null) {

System.*out*.println("Street: " + p.getPstreet());

}

if (p.getPphone() != null) {

System.*out*.println("Phone: " + p.getPphone());

}

} else {

System.*out*.println("No Physical Address");

}

if (p.getMlocation() != null || p.getMstreet() != null || p.getMphone() != null) {

System.*out*.println("Mailing Address: ");

if (p.getMlocation() != null) {

System.*out*.println("Location: " + p.getMlocation());

}

if (p.getMstreet() != null) {

System.*out*.println("Street: " + p.getMstreet());

}

if (p.getMphone() != null) {

System.*out*.println("Phone: " + p.getMphone());

}

} else {

System.*out*.println("No Mailing Address");

}

} else {

System.*out*.println("No Address Found");

}

d.close();

}

public static void getByName(String plocation) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Query p =s.createQuery("select a.pphone , a.pstreet from Address a where plocation =:plocation");

p.setParameter("plocation",plocation);

/\*List<Address> Adr =p.list();

for(Address em: Adr) {

System.out.println(em);

}\*/

List<Object[]> Adr =p.list();

for(Object[] em: Adr) {

System.*out*.println(em[0]+", "+em[1]);

}

}

public void updatedata(int aid)

{

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

Address p =(Address)s.get(Address.class, aid);

p.setPlocation("cherry");

p.setPphone("123456");

p.setMlocation("KNR");

p.setMphone("7890");

p.setMstreet("hk");

s.update(p);

t.commit();

s.close();

}

public void delete(int aid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

Address p = (Address) s.get(Address.class, aid);

s.delete(p);

t.commit();

s.close();

}

public void getByAll() {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Query q=s.createQuery("from Address");

List<Address> Ad =q.list();

for(Address k:Ad) {

System.*out*.println(k);

}

}

public static void main(String[] args) {

OneClassIntoTwoTables n = new OneClassIntoTwoTables();

//n.savemethod();

//n.GETBYID(3);

//n.getById1(1);

//n.getByName("fr");

//n.updatedata(3);

n.getByAll();

//n.delete(6);

}

}

**3)One.java**

**package** com.pojo;

**import** javax.persistence.Embedded;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

@Entity

**public** **class** One {

@Id

@GeneratedValue

**private** **int** oid;

**private** String oname;

**private** String oloc;

@Embedded

**private** Two to;

**public** One() {

}

@Override

**public** String toString() {

**return** "One [oid=" + oid + ", oname=" + oname + ", oloc=" + oloc + ", to=" + to + "]";

}

**public** One(String oname, String oloc, Two to) {

**super**();

**this**.oname = oname;

**this**.oloc = oloc;

**this**.to = to;

}

**public** **int** getOid() {

**return** oid;

}

**public** **void** setOid(**int** oid) {

**this**.oid = oid;

}

**public** String getOname() {

**return** oname;

}

**public** **void** setOname(String oname) {

**this**.oname = oname;

}

**public** String getOloc() {

**return** oloc;

}

**public** **void** setOloc(String oloc) {

**this**.oloc = oloc;

}

**public** Two getTo() {

**return** to;

}

**public** **void** setTo(Two to) {

**this**.to = to;

}

}

**4)Two.java**

**package** com.pojo;

**import** javax.persistence.Embeddable;

@Embeddable

**public** **class** Two {

**public** Two() {

}

**private** String tname;

**private** String tloc;

@Override

**public** String toString() {

**return** "Two [tname=" + tname + ", tloc=" + tloc + "]";

}

**public** Two(String tname, String tloc) {

**super**();

**this**.tname = tname;

**this**.tloc = tloc;

}

**public** String getTname() {

**return** tname;

}

**public** **void** setTname(String tname) {

**this**.tname = tname;

}

**public** String getTloc() {

**return** tloc;

}

**public** **void** setTloc(String tloc) {

**this**.tloc = tloc;

}

}

**5) TwoClassesIntoSingleTable.java**

package com.controller;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.pojo.\*;

public class TwoClassesIntoSingleTable {

public static void savemethod() {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

One p = new One();

p.setOloc("Hyd");

p.setOname("Manju");

Two q = new Two();

q.setTname("Siva");

q.setTloc("SCD");

p.setTo(q);

s.save(p);

t.commit();

s.close();

}

public static void GETBYID(int oid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

One g =(One)s.get(One.class, oid);

System.*out*.println(g);

}

public static void getByName(String oloc) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Query p =s.createQuery("select a.oloc , k.tloc, k.tname from One a inner join a.to k where a.oloc =:oloc");

p.setParameter("oloc",oloc);

/\*List<One> On =p.list();

for(One em: On) {

System.out.println(em);

}\*/

List<Object[]> Onead =p.list();

for(Object[] em: Onead) {

System.*out*.println(em[0]+", "+em[1]+","+em[2]);

}

}

public void updatedata(int oid)

{

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

One p =(One)s.get(One.class, oid);

p.setOname("cherry");

Two p1 = new Two();

p1.setTloc("Delhi");

p1.setTname("Tina");

p.setTo(p1);

s.update(p);

t.commit();

s.close();

}

public void delete(int oid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

One p = (One) s.get(One.class, oid);

s.delete(p);

t.commit();

s.close();

}

public void getByAll() {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Query q=s.createQuery("from One");

List<One> o =q.list();

for(One k:o) {

System.*out*.println(k);

}

}

public static void main(String[] args) {

TwoClassesIntoSingleTable h= new TwoClassesIntoSingleTable();

//h.savemethod();

//h.GETBYID(2);

h.*getByName*("Hyd");

//h.updatedata(2);

//h.delete(3);

//h.getByAll();

}

}

**----------07/02/2024---------------Wednesday--------Class -19---------------------**

**Topic: Composite Key&Version Concept for knowing the how many times we updated the record.**

**Morning**

**Topic:**

**What is PrimaryKey?**

**What is Foreign key?**

**What is Composite key?**

**11**

**12**

**22**

**AFTERNOON**

**Project:** **Hibnerate\_CompositeKey**

**Packages:**

**1)com.pojo**

**2)com.controller**

**Programs:**

**1)Product.java**

**2)Vendor.java**

**3)GetSessionFactory.java**

**4)CompositeTest.java**

**Afternoon**

**Topic: I must know how many times you are updating the records(I want the exact number)**

**Project:** **Hibnerate\_Version\_Concept**

**Packages:**

**1)com.pojo**

**2)com.controller**

**Programs:**

**1)Aadharcard.java**

**2)GetSessionFactory.java**

**3)AadharTest.java**

**MORNING**

**1)Product.java**

**package** com.pojo;

**import** javax.persistence.Embedded;

**import** javax.persistence.EmbeddedId;

**import** javax.persistence.Entity;

@Entity

**public** **class** Product {

@Embedded

@EmbeddedId

**private** Vendor ve;

**private** String pname;

**private** **double** pprice;

**private** String ploc;

@Override

**public** String toString() {

**return** "Product [ve=" + ve + ", pname=" + pname + ", pprice=" + pprice + ", ploc=" + ploc + "]";

}

**public** Vendor getVe() {

**return** ve;

}

**public** **void** setVe(Vendor ve) {

**this**.ve = ve;

}

**public** String getPname() {

**return** pname;

}

**public** **void** setPname(String pname) {

**this**.pname = pname;

}

**public** **double** getPprice() {

**return** pprice;

}

**public** **void** setPprice(**double** pprice) {

**this**.pprice = pprice;

}

**public** String getPloc() {

**return** ploc;

}

**public** **void** setPloc(String ploc) {

**this**.ploc = ploc;

}

}

**2)Vendor.java**

**package** com.pojo;

**import** java.io.Serializable;

**import** javax.persistence.Embeddable;

@Embeddable

**public** **class** Vendor **implements** Serializable{

**private** **int** pid;

**private** **int** vid;

@Override

**public** String toString() {

**return** "Vendor [pid=" + pid + ", vid=" + vid + "]";

}

**public** **int** getPid() {

**return** pid;

}

**public** **void** setPid(**int** pid) {

**this**.pid = pid;

}

**public** **int** getVid() {

**return** vid;

}

**public** **void** setVid(**int** vid) {

**this**.vid = vid;

}

}

1. **GetSessionFactory.java(It is same for two projects)**

package com.controller;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class GetSessionFactory {

public static SessionFactory con() {

Configuration con = new AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

return sf;

}

}

**4)CompositeTest.java**

package com.controller;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.pojo.Product;

import com.pojo.Vendor;

public class CompositeTest {

public static void savemethod() {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

Product h = new Product();

h.setPname("TV");

h.setPprice(56.7);

h.setPloc("Hyd");

Vendor j = new Vendor();

j.setPid(1);

j.setVid(1);

h.setVe(j);

s.save(h);

t.commit();

s.close();

}

public void updatedata(int pid, int vid)

{

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

Vendor v = new Vendor();

v.setPid(pid);

v.setVid(vid);

Product p =(Product)s.get(Product.class,v);

p.setPname("cherry");

s.update(p);

t.commit();

s.close();

}

public static void getAll() {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Query q =s.createQuery("from Product");

List<Product> d=q.list();

for (Product p : d) {

System.*out*.println(p);

}

}

public static void getByid(int pid, int vid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Vendor v = new Vendor();

v.setVid(vid);

v.setPid(pid);

Product p =(Product) s.get(Product.class, v);

System.*out*.println(p);

s.close();

}

public static void main(String[] args) {

CompositeTest n = new CompositeTest();

//n.savemethod();

//n.updatedata(1, 1);

//n.getAll();

n.*getByid*(1, 1);

}

}

/\*package com.controller;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.pojo.Product;

import com.pojo.Vendor;

public class CompositeTest {

public static void savemethod() {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

Product h = new Product();

h.setPname("TV");

h.setPprice(56.7);

h.setPloc("Hyd");

Vendor j = new Vendor();

j.setPid(1);

j.setVid(1);

h.setVe(j);

s.save(h);

t.commit();

s.close();

}

public static Product getByCompositeKey(int pid, int vid) {

SessionFactory sf = GetSessionFactory.*con*();

Session session = sf.openSession();

Transaction transaction = null;

Product product = null;

try {

transaction = session.beginTransaction();

// Create an instance of the composite key

Vendor compositeKey = new Vendor();

compositeKey.setPid(pid);

compositeKey.setVid(vid);

// Retrieve the Product by composite key

product = (Product) session.get(Product.class, compositeKey);

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

session.close();

}

return product;

}

public static void main(String[] args) {

CompositeTest n = new CompositeTest();

//n.savemethod();

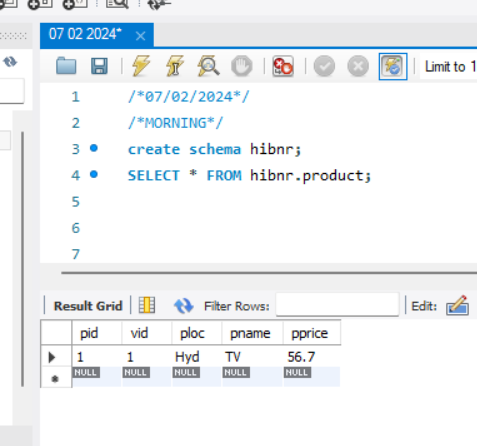
//n.getByCompositeKey(1, 1);

Product product = CompositeTest.*getByCompositeKey*(1, 1);

System.*out*.println("Retrieved Product: " + product);

}

}\*/



**AFTERNOON**

**1)** **AadharCard.java**

**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.Version;

@Entity

**public** **class** AadharCard {

@Id

@GeneratedValue

**private** Integer aid;

@Override

**public** String toString() {

**return** "AadharCard [aid=" + aid + ", anum=" + anum + ", aname=" + aname + ", aaddress=" + aaddress

+ ", Version=" + Version + "]";

}

**private** String anum;

**private** String aname;

**private** String aaddress;

@Version

**private** Long Version;

**public** Integer getAid() {

**return** aid;

}

**public** **void** setAid(Integer aid) {

**this**.aid = aid;

}

**public** String getAnum() {

**return** anum;

}

**public** **void** setAnum(String anum) {

**this**.anum = anum;

}

**public** String getAname() {

**return** aname;

}

**public** **void** setAname(String aname) {

**this**.aname = aname;

}

**public** String getAaddress() {

**return** aaddress;

}

**public** **void** setAaddress(String aaddress) {

**this**.aaddress = aaddress;

}

}

**2) AadharTest.java**

**package** com.controller;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** com.pojo.AadharCard;

**public** **class** AadharTest {

**public** **static** **void** savemethod() {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

AadharCard p = **new** AadharCard();

p.setAname("lucky");

p.setAnum("75342465");

p.setAaddress("knr");

s.save(p);

t.commit();

s.close();

}

**public** **void** updatedata(**int** aid)

{

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Transaction t = s.beginTransaction();

AadharCard p =(AadharCard)s.get(AadharCard.**class**, aid);

p.setAname("Radhika");

p.setAnum("75erw9739");

p.setAaddress("KNR");

s.update(p);

t.commit();

s.close();

}

**public** **static** **void** main(String[] args) {

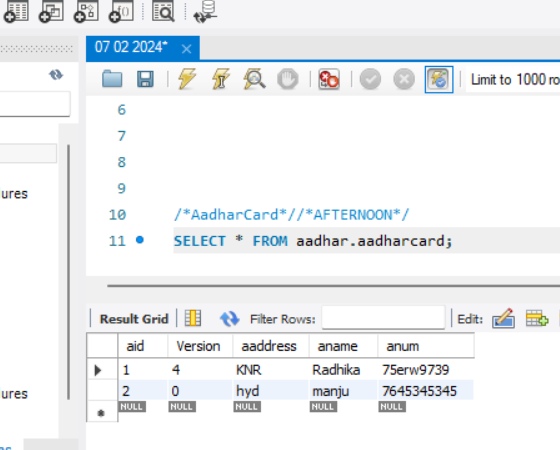
AadharTest m = **new** AadharTest();

//m.savemethod();

m.updatedata(1);

}

}



**----08/02/2024-------------Thursday------------Class - 20--------------------------**

**TOPIC: INHERITANCE**

* **SINGLE\_TABLE**
* **JOINED**
* **TABLE\_PER\_CLASS**

**PROJECT: Hibnerate\_Inheritance**

**PACKAGES:**

**1)com.pojo**

**Programs:**

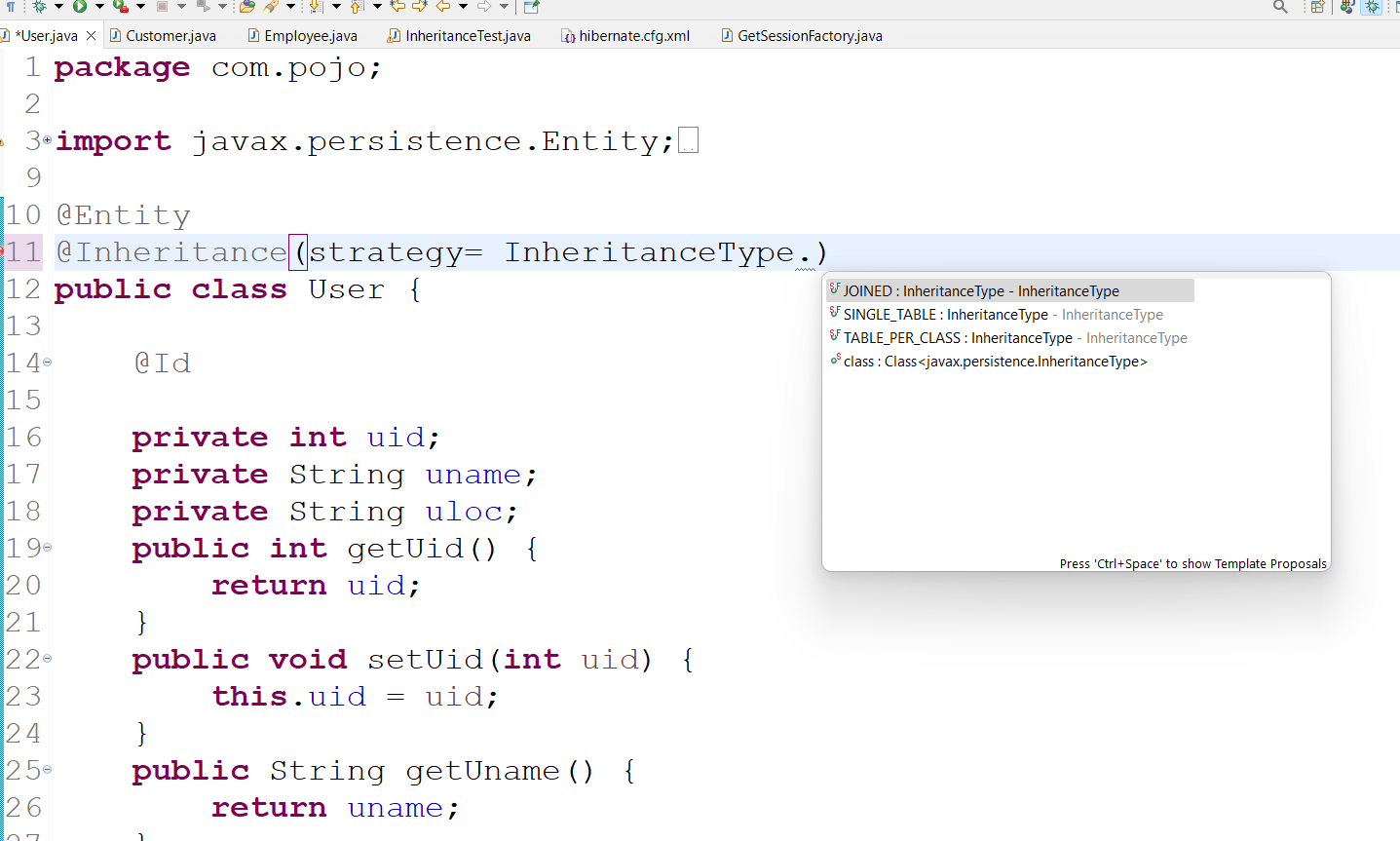
1. **User.java**
2. **Customer.java**
3. **Employee.java**

**2)com.controller**

**d. GetSessionFactory.java**

**e. InheritanceTest.java**

@Inheritance(strategy= InheritanceType.***TABLE\_PER\_CLASS***)



**a.User.java**

**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Inheritance;

**import** javax.persistence.InheritanceType;

@Entity

@Inheritance(strategy= InheritanceType.***TABLE\_PER\_CLASS***)

**public** **class** User {

@Id

**private** **int** uid;

**private** String uname;

**private** String uloc;

**public** **int** getUid() {

**return** uid;

}

**public** **void** setUid(**int** uid) {

**this**.uid = uid;

}

**public** String getUname() {

**return** uname;

}

**public** **void** setUname(String uname) {

**this**.uname = uname;

}

**public** String getUloc() {

**return** uloc;

}

**public** **void** setUloc(String uloc) {

**this**.uloc = uloc;

}

}

b.Employee.java

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Employee extends User {

private String ename;

private String eloc;

public String getEname() {

return ename;

}

public void setEname(String ename) {

this.ename = ename;

}

public String getEloc() {

return eloc;

}

public void setEloc(String eloc) {

this.eloc = eloc;

}

}

c.Customer.java

**package** com.pojo;

**import** javax.persistence.Entity;

@Entity

**public** **class** Customer **extends** User {

**private** String cname;

**private** String cloc;

**public** String getCname() {

**return** cname;

}

**public** **void** setCname(String cname) {

**this**.cname = cname;

}

**public** String getCloc() {

**return** cloc;

}

**public** **void** setCloc(String cloc) {

**this**.cloc = cloc;

}

}

d.GetSessionFactory.java

**package** com.controller;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**import** org.hibernate.cfg.AnnotationConfiguration;

**public** **class** GetSessionFactory {

**public** **static** SessionFactory con() {

Configuration con = **new** AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

**return** sf;

}

}

e.InheritanceTest.java

**package** com.controller;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** com.pojo.\*;

**public** **class** InheritanceTest {

**public** **static** **void** savemethod() {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

Employee h = **new** Employee();

h.setUid(1);

h.setEloc("HYD");

h.setEname("SIVA");

h.setUloc("SCD");

h.setUname("RAM");

Customer j = **new** Customer();

j.setUid(2);

j.setCloc("ZHB");

j.setCname("ROSY");

j.setUloc("KTB");

j.setUname("LILLY");

s.save(h);

s.save(j);

t.commit();

s.close();

}

**public** **static** **void** main(String[] args) {

InheritanceTest n = **new** InheritanceTest();

n.*savemethod*();

}

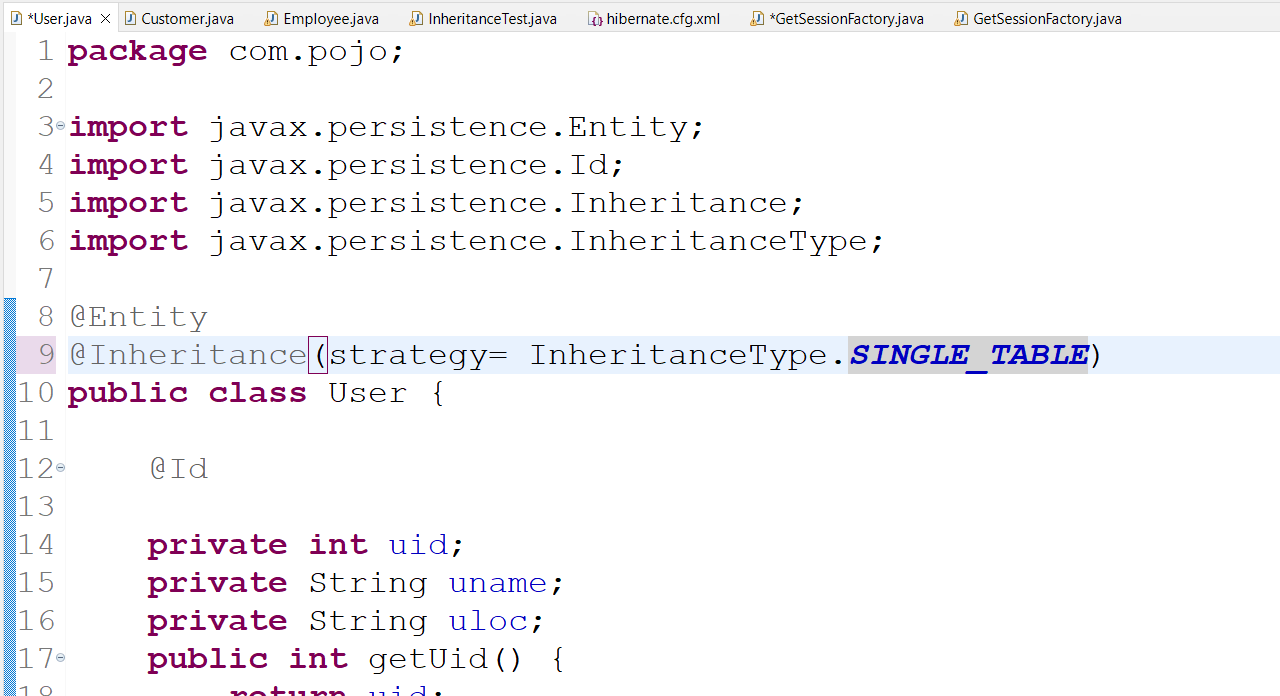
}

3cases :

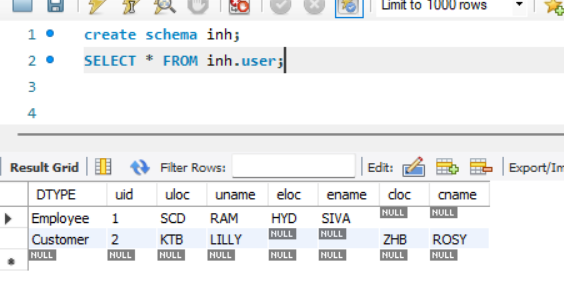
* **SINGLE\_TABLE**
* **JOINED**
* **TABLE\_PER\_CLASS**

**OUTPUTS:**

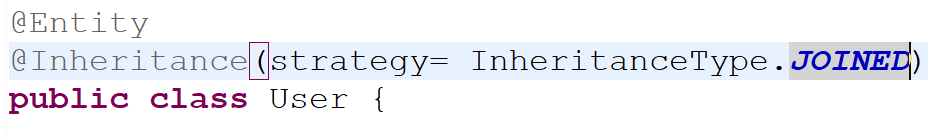
Case-1: **SINGLE\_TABLE**



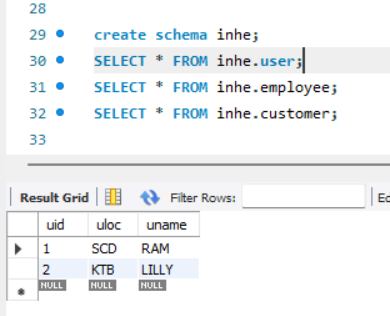
**1)**

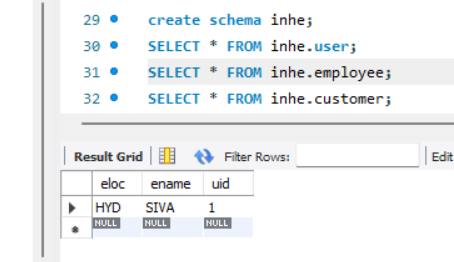


Case-2: **JOINED**

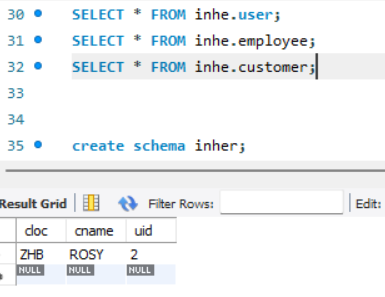


1)

  
2)



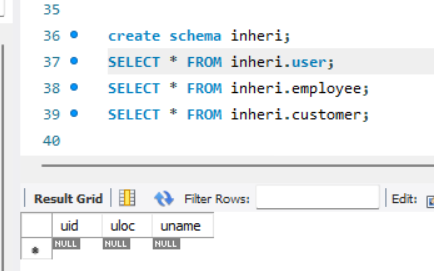
3)



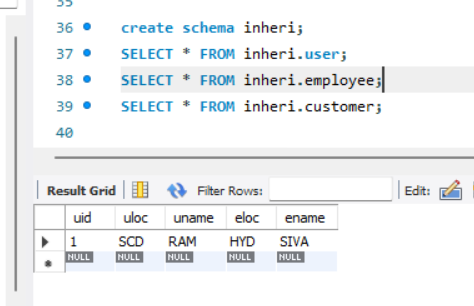
**Case-3: TABLE\_PER\_CLASS**

@Inheritance(strategy= InheritanceType.***TABLE\_PER\_CLASS***)

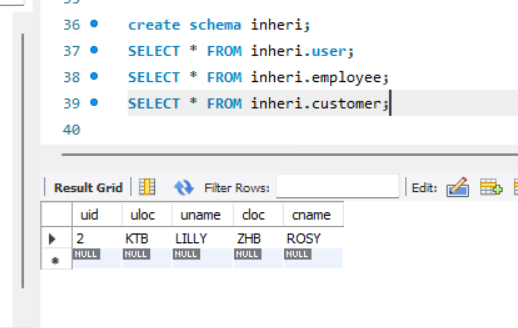
1)



2)



3)



**------------------09-02-2024-----------------FRIDAY--------CLASS-21----------**

(morning no class)

**(AFTERNOON)**

TOPIC:

* LAZY LOADING
* EAGER LOADING
* IMPORTANCE OF FLUSH() METHOD(See down for programs in detail)
* DIFFERENCE BETWEEN FLUSH() AND COMMIT() METHOD

Note: shortcutkey to keep the data in comments **CTRL+SHIFT+/**

TOPIC-1:

Difference between Lazy loading and Eager loading

<https://youtu.be/TejT8H81aVI?si=dNjufMdB65CNzypi>

Project: Hibnerate\_LazyLoadingAndEagerLoading

Packages:

1)com.pojo

Programs:

a) Person.java

b) Passport.java

c) mobile.java

d) course.java

2)com.controller

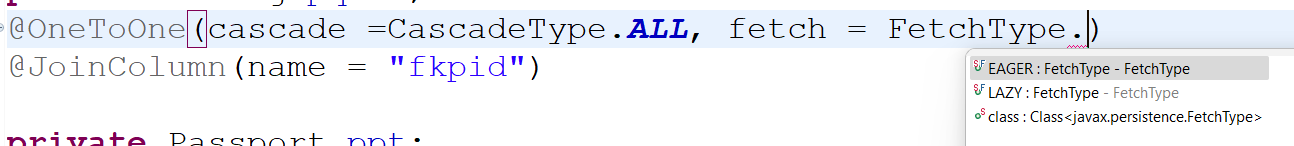
e) GetSessionFactory.java

f) All.java

NOTE:

* Lazy and Eager Loading are called as Fetching Strategies.
* One to One By default it has Eager Loading
* Many To One, Many To Many By default they have Lazy Loading

a)Person.java



**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.OneToOne;

**import** javax.persistence.CascadeType;

**import** java.util.List;

**import** javax.persistence.\*;

@Entity

**public** **class** Person {

@Id

**private** **int** pid;

**private** String pname;

**private** **int** page;

**private** String pqual;

@OneToOne(cascade =CascadeType.***ALL***, fetch = FetchType.***EAGER***)

@JoinColumn(name = "fkpid")

**private** Passport ppt;

@OneToMany(cascade = CascadeType.***ALL***, fetch = FetchType.***EAGER***)

@JoinColumn(name = "fkpersid")

**private** List<mobile> mob;

@ManyToMany(cascade = CascadeType.***ALL***, fetch = FetchType.***EAGER***)

@JoinTable(name = "percou",

joinColumns = {@JoinColumn(name = "fkpid")},

inverseJoinColumns = {@JoinColumn(name = "fkcid")})

**private** List<course> cou;

**public** Person() {

}

**public** Person(**int** pid, String pname, **int** page, String pqual, Passport ppt, List<mobile> mob, List<course> cou) {

**super**();

**this**.pid = pid;

**this**.pname = pname;

**this**.page = page;

**this**.pqual = pqual;

**this**.ppt = ppt;

**this**.mob = mob;

**this**.cou = cou;

}

/\*@Override

public String toString() {

return "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + ", ppt=" + ppt + ", mob="

+ mob + ", cou=" + cou + "]";

}\*/

@Override

**public** String toString() {

**return** "Person [pid=" + pid + ", pname=" + pname + ", page=" + page + ", pqual=" + pqual + "]";

}

**public** **int** getPid() {

**return** pid;

}

**public** **void** setPid(**int** pid) {

**this**.pid = pid;

}

**public** String getPname() {

**return** pname;

}

**public** **void** setPname(String pname) {

**this**.pname = pname;

}

**public** **int** getPage() {

**return** page;

}

**public** **void** setPage(**int** page) {

**this**.page = page;

}

**public** String getPqual() {

**return** pqual;

}

**public** **void** setPqual(String pqual) {

**this**.pqual = pqual;

}

**public** Passport getPpt() {

**return** ppt;

}

**public** **void** setPpt(Passport ppt) {

**this**.ppt = ppt;

}

**public** List<mobile> getMob() {

**return** mob;

}

**public** **void** setMob(List<mobile> mob) {

**this**.mob = mob;

}

**public** List<course> getCou() {

**return** cou;

}

**public** **void** setCou(List<course> cou) {

**this**.cou = cou;

}

}

b) Passport.java

**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

@Entity

**public** **class** Passport {

@Id

@GeneratedValue

**private** **int** ppid;

**private** String pnum;

**private** String pexp;

@Override

**public** String toString() {

**return** "Passport [ppid=" + ppid + ", pnum=" + pnum + ", pexp=" + pexp + "]";

}

**public** **int** getPpid() {

**return** ppid;

}

**public** **void** setPpid(**int** ppid) {

**this**.ppid = ppid;

}

**public** String getPnum() {

**return** pnum;

}

**public** **void** setPnum(String pnum) {

**this**.pnum = pnum;

}

**public** String getPexp() {

**return** pexp;

}

**public** **void** setPexp(String pexp) {

**this**.pexp = pexp;

}

}

**c)mobile.java**

**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

@Entity

**public** **class** mobile {

@Id

@GeneratedValue

**private** **int** mid;

**private** **long** mnum;

**private** String msim;

@Override

**public** String toString() {

**return** "mobile [mid=" + mid + ", mnum=" + mnum + ", msim=" + msim + "]";

}

**public** mobile() {

}

**public** mobile( **long** mnum, String msim) {

**this**.mnum = mnum;

**this**.msim = msim;

}

**public** **int** getMid() {

**return** mid;

}

**public** **void** setMid(**int** mid) {

**this**.mid = mid;

}

**public** **long** getMnum() {

**return** mnum;

}

**public** **void** setMnum(**long** mnum) {

**this**.mnum = mnum;

}

**public** String getMsim() {

**return** msim;

}

**public** **void** setMsim(String msim) {

**this**.msim = msim;

}

}

**d) course.java**

**package** com.pojo;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

@Entity

**public** **class** course {

@Id

@GeneratedValue

**private** Integer cid;

**private** String cname;

**private** Double cfee;

**public** Integer getCid() {

**return** cid;

}

**public** **void** setCid(Integer cid) {

**this**.cid = cid;

}

**public** String getCname() {

**return** cname;

}

**public** **void** setCname(String cname) {

**this**.cname = cname;

}

**public** Double getCfee() {

**return** cfee;

}

**public** **void** setCfee(Double cfee) {

**this**.cfee = cfee;

}

**public** course(Integer cid, String cname, Double cfee) {

**super**();

**this**.cid = cid;

**this**.cname = cname;

**this**.cfee = cfee;

}

**public** course() {

}

@Override

**public** String toString() {

**return** "course [cid=" + cid + ", cname=" + cname + ", cfee=" + cfee + "]";

}

}

e) **GetSessionFactory.java**

**package** com.controller;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.AnnotationConfiguration;

**import** org.hibernate.cfg.Configuration;

**public** **class** GetSessionFactory {

**public** **static** SessionFactory con() {

Configuration con = **new** AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

**return** sf;

}

}

**f) All.java**

**package** com.controller;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** com.pojo.Passport;

**import** com.pojo.Person;

**import** com.pojo.course;

**import** com.pojo.mobile;

**public** **class** All {

**public** **void** saveddata(Person p) {

SessionFactory sf = GetSessionFactory.*con*();

Session s =sf.openSession();

Transaction t = s.beginTransaction();

s.save(p);

t.commit();

s.close();

}

**public** **void** getById(**int** pid) {

SessionFactory sf = GetSessionFactory.*con*();

Session s = sf.openSession();

Person p=(Person)s.get(Person.**class**, pid);

System.***out***.println(p);

}

**public** **static** **void** main(String[] args) {

All t = **new** All();

Person p =**new** Person();

p.setPname("SITA");

p.setPage(24);

p.setPqual("MCOM");

Passport q = **new** Passport();

q.setPnum("AB123Z");

q.setPexp("3090");

p.setPpt(q);

mobile m = **new** mobile();

m.setMsim("jio");

m.setMnum(986645342l);

mobile m1 = **new** mobile();

m1.setMsim("Airtel");

m1.setMnum(9786645342l);

mobile m2 = **new** mobile();

m2.setMsim("bsnl");

m2.setMnum(954645342l);

List<mobile> mob = **new** ArrayList<>();

mob.add(m);

mob.add(m1);

mob.add(m2);

p.setMob(mob);

course m3 = **new** course();

m3.setCname("Python");

m3.setCfee(30.5);

course m4 = **new** course();

m4.setCname("C");

m4.setCfee(40.5);

course m5 = **new** course();

m5.setCname("C++");

m5.setCfee(50.5);

List<course> cou = **new** ArrayList<>();

cou.add(m3);

cou.add(m4);

cou.add(m5);

p.setCou(cou);

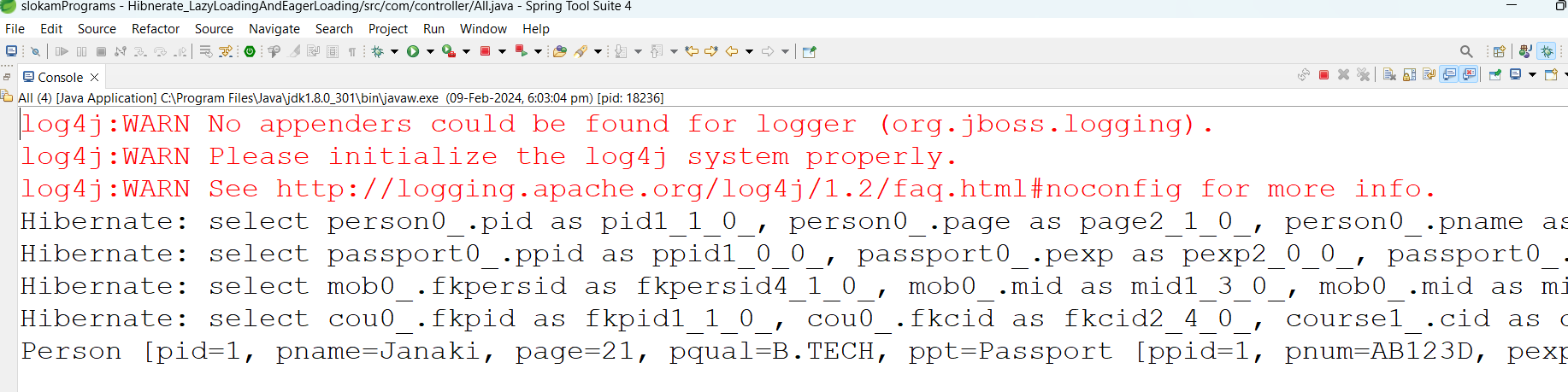
//t.saveddata(p);

t.getById(1);

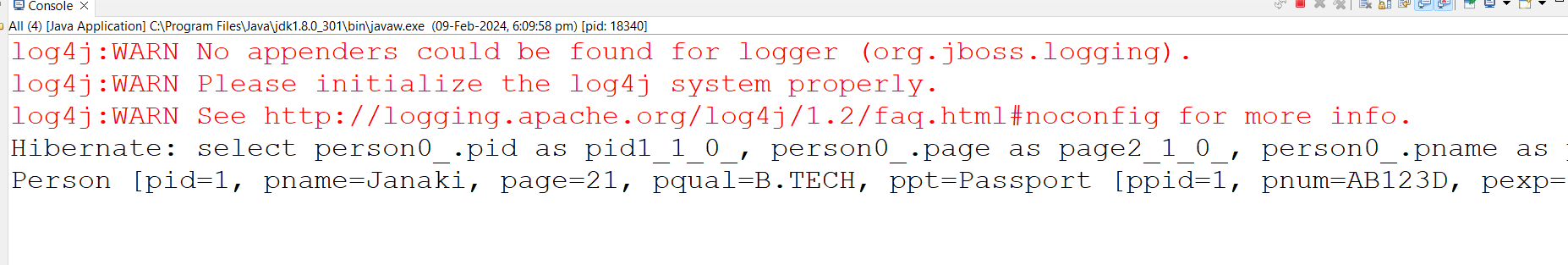
}

}

Output it is LAZY LOADING:



Output if it is EAGER LOADING:



**TOPIC-2:**

**Project: Hibernate\_Flush() Method And Commit() Method**

Flush(): In simple words, suppose there are 1lakh records , sending whole 1lakh records at a time to database, whole records will be in session. So, by using flush method we will send 10,000 records at a time. Sending 10,000 records at a time is better than Sending 1 lakh records at a time. Flush method just sinks the data in to database, but not commit. By this, process becomes easy and fast**.**

**Packages:**

**1)com.pojo**

**a) slkstd.java**

**2)com.controller**

**b) GetSessionFactory.java**

**c)Test.java**

[**https://www.javapedia.net/module/Hibernate/Hibernate-interview-questions-II/1166**](https://www.javapedia.net/module/Hibernate/Hibernate-interview-questions-II/1166)

**Note:**

**Difference commit() vs flush().**

1. **flush(): Flushing is the process of synchronizing the underlying persistent store with persistable state held in memory.it will update or insert into your tables in the running transaction, but it may not commit those changes.**
2. **Commit(): Commit will make the database commit.When you have a persisted object and you change a value on it, it becomes dirty and hibernate needs to flush these changes to your persistence layer.So You should commit but it also ends the unit of work.**

**a)slkstd.java**

package com.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class slkstd {

@Id

@GeneratedValue

private Integer sid;

private String sname;

private Integer sage;

private String squal;

private double smarks;

@Override

public String toString() {

return "slkstd [sid=" + sid + ", sname=" + sname + ", sage=" + sage + ", squal=" + squal + ", smarks=" + smarks

+ "]";

}

public Integer getSid() {

return sid;

}

public void setSid(Integer sid) {

this.sid = sid;

}

public String getSname() {

return sname;

}

public void setSname(String sname) {

this.sname = sname;

}

public Integer getSage() {

return sage;

}

public void setSage(Integer sage) {

this.sage = sage;

}

public String getSqual() {

return squal;

}

public void setSqual(String squal) {

this.squal = squal;

}

public double getSmarks() {

return smarks;

}

public void setSmarks(double smarks) {

this.smarks = smarks;

}

}

1. **GetSessionFactory.java**

package com.controller;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.AnnotationConfiguration;

import org.hibernate.cfg.Configuration;

public class GetSessionFactory {

public static SessionFactory con() {

Configuration con = new AnnotationConfiguration().configure();

SessionFactory sf =con.buildSessionFactory();

return sf;

}

}

**c)Test.java**

**package** com.controller;

**import** java.util.ArrayList;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** java.io.\*;

**import** com.pojo.slkstd;

**public** **class** Test {

**public** **static** ArrayList<slkstd> readfile() {

ArrayList<slkstd> stdd = **new** ArrayList();

**try** {

FileReader fr = **new** FileReader("E:\\Slokam Files\\slkstddata.txt");

BufferedReader br = **new** BufferedReader(fr);

**for**(String s = br.readLine(); s!=**null**;s = br.readLine()) {

String[] n = s.split(",");

slkstd std = **new** slkstd();

std.setSid(Integer.*parseInt*(n[0]));

std.setSname(n[1]);

std.setSage(Integer.*parseInt*(n[2]));

std.setSqual(n[3]);

std.setSmarks(Double.*valueOf*(n[4]));

stdd.add(std);

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** stdd;

}

**public** **static** **void** main(String[] args) {

ArrayList<slkstd> dd=Test.*readfile*();

SessionFactory s=GetSessionFactory.*con*();

Session ss= s.openSession();

Transaction t= ss.beginTransaction();

**int** i=1;

**for** (slkstd sl : dd) {

System.***out***.println(sl);

ss.save(sl);

**if**(i%5==0) {

ss.flush();

}

i++;

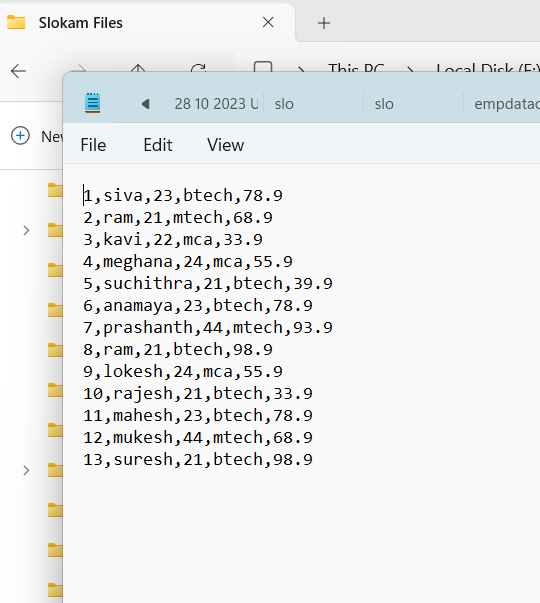
}

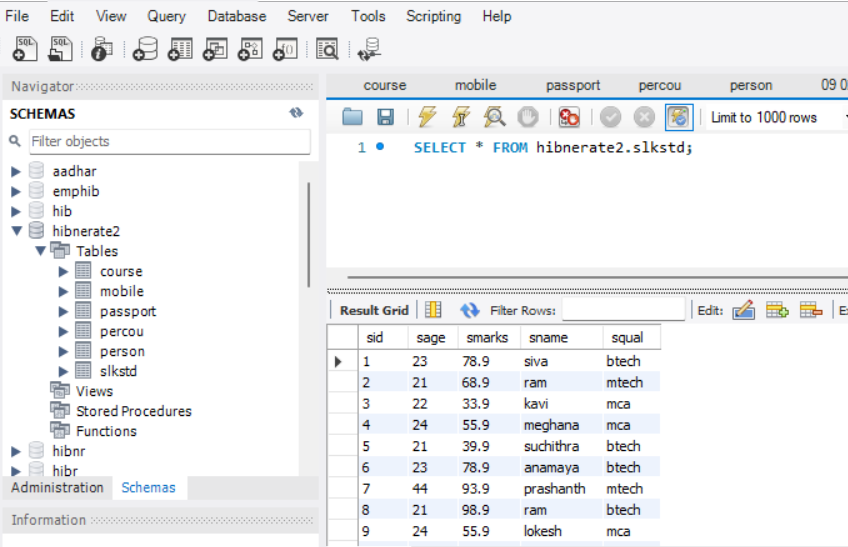
t.commit();

s.close();

}

}





**-------------10/02/2024-----Saturday----No Class--------------------------------------------12/02/2024--------Monday------SPRING STARTED------------------**

**----------13/02/2024--------Tuesday------Class-22---------------------------------**

**TOPIC:**

**AN**

**FIRST LEVEL CACHE, SECOND LEVEL CACHE, NO PROGRAMS**

**---------14/02/2024-----------Wednesday-----Class-23---------------------------**

**TOPIC:**

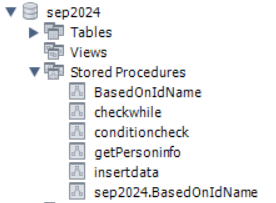
Morning no class

After Noon

Topic:

STORED PROCEDURES IN MYSQL

In procedures



1)insertdata

CREATE DEFINER=`root`@`localhost` PROCEDURE `insertdata`(in pid int ,in pname varchar(255), in page int, in pqual varchar(255))

BEGIN

insert into sep2024.person values(pid, pname, page, pqual);

END

2)BasedOnIdName

CREATE DEFINER=`root`@`localhost` PROCEDURE `BasedOnIdName`(in id int, in pname varchar(255))

BEGIN

select\*from sep2024.person where pid = id and pname = pname;

END

3)getPersoninfo

CREATE DEFINER=`root`@`localhost` PROCEDURE `getPersoninfo`()

BEGIN

select\*from sep2024.person;

END

4)conditioncheck

CREATE DEFINER=`root`@`localhost` PROCEDURE `conditioncheck`(in pid int ,in pname varchar(255), in page int, in pqual varchar(255))

BEGIN

if page >= 21 THEN

insert into sep2024.person values(pid, pname, page, pqual);

else

set page = 25;

insert into sep2024.person values(pid, pname, page, pqual);

end if;

END

5)checkwhile

CREATE DEFINER=`root`@`localhost` PROCEDURE `checkwhile`(in maxIterations int)

BEGIN

declare count int default 1;

while count <= maxIterations

Do

insert into sep2024.person values(count, 'siva', 32, 'mca');

select 'Employee inserted successfully' As DebugMessage;

select 'count value===' +count As DebugMessage;

set count = count+1;

end while;

END

-----------------15/02/2024------------Thursday---------Class-24---------------

Topic:

Morning

Part -1:

STORED PROCEDURES IN MYSQL

Out procedures

1. GETEmployeeDetails

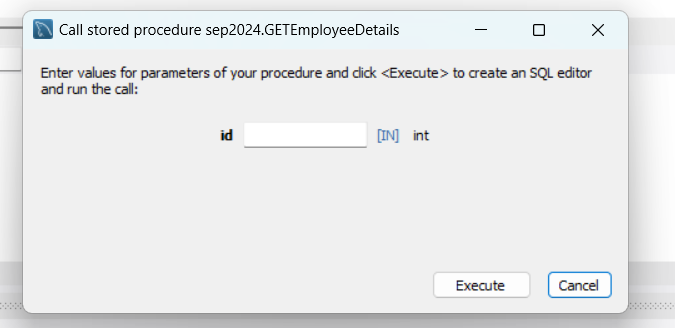
CREATE DEFINER=`root`@`localhost` PROCEDURE `GETEmployeeDetails`( in id int, out pnames varchar(255), out pages int)

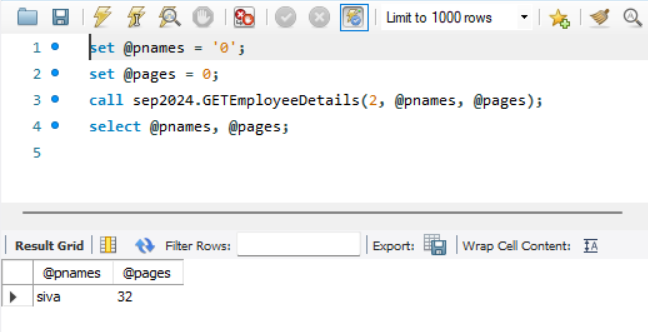
BEGIN

select pname, page into pnames, pages from sep2024.person where pid = id;

END

**Output**





set @pnames = '0';

set @pages = 0;

call sep2024.GETEmployeeDetails(1, @pnames, @pages);

select @pnames, @pages;

Part-2:

How to call Stored procedures By Hibernate

Afternoon no class

------------16/02/2024------Friday-----------------------Class-25---------------

**Topic:**

**Named Parameters**

**----------------------17/02/2024----------Saturday-------------Noclass----------**

**----------------19/02/2024-----------Monday-----------Class-26-----------------**