

Hibernate Assignment by

Name : Prashant Maurya

EmpID: 12417

Kiit Roll: 1828259

Questions

1. Make a Hibernate Project which will connect this table is in DB a.
Table one "Student" with the following column
 - i. student_id INTEGER PRIMARY KEY (This has to be autogenerated by Hibernate Framework, we have not discussed how to do this, in this document/session, This is not very difficult to implement just do a small research on the net. Hint: generator)
 - ii. roll_num INTEGER NOT NULL
 - iii. name VARCHAR(45) NOT NULL
 - iv. age INTEGER NOT NULL
 - v. gender VARCHAR(1)
2. Save 8 entries in the table using .save() method
3. Display following using HQL (without criteria object)
 - a. Students with age >12
 - b. Female Students with age <20
4. Display following using HQL (with criteria object)
 - a. Male Students with age >18 and name starting "A"
 - b. Students with gender is empty

Xml file-

Hibernate.cfg.xml

```
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
```

```

<hibernate-configuration>

    <session-factory>

        <!-- JDBC Database connection settings -->
        <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/kt_schema</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">root</property>
        <property name="hbm2ddl.auto">update</property>

        <!-- Echo the SQL to stdout -->
        <property name="show_sql">true</property>

        <mapping resource="com/work/cfg/highako.hbm.xml" />

    </session-factory>

</hibernate-configuration>

```

Highako.hbm.xml

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 5.3//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-5.3.dtd">

<hibernate-mapping>
    <class name="com.work.model.PojoModel" table="student">
        <id name="student_id" type="java.lang.Integer" column="student_id" />
        <property name="roll_num" type="java.lang.Integer" column="roll_num" />
        <property name="name" type="string" column="name" />
        <property name="age" type="java.lang.Integer" column="age" />
        <property name="gender" type="string" column="gender" />
    </class>

</hibernate-mapping>

```

Pojo Class-

```

package com.work.model;

public class PojoModel {

```

```
int student_id;
int roll_num;
String name;
int age;
String gender;

public PojoModel() {

}

public PojoModel(int sid,int roll , String name , int age,String g) {

    setAge(age);
    setGender(g);
    setName(name);
    setRoll_num(roll);
    setStudent_id(sid);

}

public int getStudent_id() {
    return student_id;
}
public void setStudent_id(int student_id) {
    this.student_id = student_id;
}
public int getRoll_num() {
    return roll_num;
}
public void setRoll_num(int roll_num) {
    this.roll_num = roll_num;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public int getAge() {
    return age;
}
public void setAge(int age) {
    this.age = age;
}
public String getGender() {
    return gender;
}
public void setGender(String g) {
```

```

        this.gender = g;
    }

    @Override
    public String toString() {
        return "PojoModel [student_id=" + student_id + ", roll_num=" + roll_num + ", name=" +
name + ", age=" + age
                                + ", gender=" + gender + "]\n";
    }
}

```

Controller classes

MainClass

```

package com.work.controller;

import java.util.ArrayList;
import java.util.List;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

import com.work.model.PojoModel;

public class AppendModel {

    public static void main(String[] args) {

        List<PojoModel> list = new ArrayList<>();

        PojoModel obj1 = new PojoModel(1,20,"ASD",20,"M");
        PojoModel obj2 = new PojoModel(2,21,"ZXC",22,"F");
        PojoModel obj3 = new PojoModel(3,22,"AWE",20,"M");
        PojoModel obj4 = new PojoModel(4,23,"AXC",21,"F");
        PojoModel obj5 = new PojoModel(5,24,"ZSE",19,"M");
        PojoModel obj6 = new PojoModel(6,25,"QXC",25,"F");
        PojoModel obj7 = new PojoModel(7,26,"XAD",23,"M");
        PojoModel obj8 = new PojoModel(8,27,"CVB",21,"F");

        list.add(obj1);
        list.add(obj2);
    }
}

```

```
list.add(obj3);
list.add(obj4);
list.add(obj5);
list.add(obj6);
list.add(obj7);
list.add(obj8);
```

```
Configuration config = new Configuration().configure("com/work/cfg/hibernate.cfg.xml");
```

```
SessionFactory factory = config.buildSessionFactory();
```

```
for(PojoModel obj : list)
{
    Session ses = factory.openSession();
    Transaction txn = null;
    try {
        txn = ses.beginTransaction();
        ses.save(obj);
        txn.commit();
    }
    catch (Exception e) {
        if(txn!=null)
        {
            txn.rollback();
        }
        e.printStackTrace();
    }
    finally {
        ses.close();
    }
}
```

```
}
```

```
}
```

```
}
```

```
=====
```

3. Display following using HQL (without criteria object)

- a. Students with age <22
- b. Female Students

Read data (**without** Criteria)

```
package com.work.controller;

import java.util.List;

import org.hibernate.Criteria;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Restrictions;

import com.work.model.PojoModel;

public class ReadStudent {

    public static void main(String[] args) {

        Configuration config = new Configuration().configure("com/work/cfg/hibernate.cfg.xml");

        SessionFactory factory = config.buildSessionFactory();

        Session session = factory.openSession();

        Query theQuery = session.createQuery("FROM PojoModel obj WHERE obj.gender = :gender and obj.age>20");

        theQuery.setParameter("gender","F");

        List<PojoModel> pojoList = theQuery.list();

        for(PojoModel obj:pojoList)
        {
            System.out.println(obj.toString());
        }

    }

}
```

Read data (**with** Criteria)

```
package com.work.controller;
```

```

import java.util.List;

import org.hibernate.Criteria;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Restrictions;

import com.work.model.PojoModel;

public class ReadStudent {

    public static void main(String[] args) {

        Configuration config = new Configuration().configure("com/work/cfg/hibernate.cfg.xml");

        SessionFactory factory = config.buildSessionFactory();

        Session session = factory.openSession();

        // USING CRITERIA

        Criteria c = session.createCriteria(PojoModel.class);

        c.add(Restrictions.eq("gender", "M"));
        c.add(Restrictions.mt("age", 20));
        c.add(Restrictions.like("name", "A%"));

        List<PojoModel> pojoList = c.list();

        for(PojoModel obj:pojoList)
        {
            System.out.println(obj.toString());
        }

    }

}

```

