

Hibernate One to Many relationship program

Name : Kshitij Shelke

Ref. No. SCS/CG/2021/024

1) User.class

```
package com.app.model;
import java.util.List;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToMany;
import javax.persistence.OrderColumn;
import javax.persistence.Table;
@Entity
@Table(name = "user_table")
public class User {
    @Override
    public String toString() {
        return "User [UserId=" + UserId + ", fName=" + fName + ", lName=" + lName + "];"
    }
    @Id
    @Column(name = "User_id")
    private int UserId;
    @Column(name = "First_name")
    private String fName;
    @Column(name = "Last_name")
    @OneToMany(targetEntity = PhoneNumber.class, cascade = CascadeType.ALL,
        orphanRemoval = true)
    @JoinColumn(name = "unid", referencedColumnName = "user_id")
    @OrderColumn(name = "list_index")
    private List<PhoneNumber> phoneNumbers;
    public int getUserId() {
        return UserId;
    }
    public void setUserId(int userId) {
        UserId = userId;
    }

    public String getfName() {
        return fName;
    }
```

```

}

public void setfName(String fName) {
    this.fName = fName;
}

public String getlName() {
    return lName;
}

public void setlName(String lName) {
    this.lName = lName;
}

private String lName;

public List<PhoneNumber> getPhoneNumbers() {
    return phoneNumbers;
}

public void setPhoneNumbers(List<PhoneNumber> phoneNumbers) {
    this.phoneNumbers = phoneNumbers;
}
}

```

2) PhoneNumber.class

```

package com.app.model;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
@Entity
@Table(name = "phoneNumber_table")
public class PhoneNumber {
    @Id
    private long phone;
    @Column(name = "number_type")
    private String numberType;
    public long getPhone() {
        return phone;
    }
}

```

```

    }
    public void setPhone(long phone) {
        this.phone = phone;
    }
    public String getNumberType() {
        return numberType;
    }
    public void setNumberType(String numberType) {
        this.numberType = numberType;
    }
    @Override
    public String toString() {
        return "PhoneNumber [phone=" + phone + ", numberType=" + numberType + "];"
    }
}

```

3) One to many Dao.class

```

package com.app.dao;
public interface OneToManyDao {
void insertData();
}

```

4) One to many Dao Impl.class

```

package com.app.dao.impl;
import java.util.ArrayList;
import java.util.List;

import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

import com.app.dao.OneToManyDao;
import com.app.model.PhoneNumber;
import com.app.model.User;
import com.app.util.UtilityClass;

public class OneToManyDaoimpl implements OneToManyDao {

```

```

public void insertData() {
// TODO Auto-generated method stub
Session session=UtilityClass.getSession();
Transaction tx=session.beginTransaction();
PhoneNumber phoneNumber=new PhoneNumber();
phoneNumber.setPhone(976882325);
phoneNumber.setNumberType("office");

PhoneNumber phoneNumber1=new PhoneNumber();
phoneNumber1.setPhone(830847223);
phoneNumber1.setNumberType("home");

List<PhoneNumber>list= new ArrayList<PhoneNumber>();
list.add(phoneNumber1);
list.add(phoneNumber);

User user=new User();
user.setfName("sangharsh");
user.setlName("Narwade");
user.setUserId(101);
user.setPhoneNumbers(list);
session.save(user);
tx.commit();
UtilityClass.closeSession();

}

}

```

5)One to many Dao Factory.class

```

package com.app.factory;
import com.app.dao.OneToManyDao;
import com.app.dao.impl.OneToManyDaoimpl;
public class OneToManyFactory {
public static OneToManyDao getInstance() {
return new OneToManyDaoimpl();
}
}

```

6)Utility.class

```
package com.app.util;

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

public class UtilityClass {
    private static SessionFactory factory;
    static {
        try {
            factory = new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }
    static ThreadLocal<Session> threadLocal = new ThreadLocal<Session>();
    static Session session = null;

    public static Session getSession() {
        try {
            if(threadLocal.get()==null) {
                session=factory.openSession();
                return session;
            }
            else {
                return threadLocal.get();
            }
        }catch (Exception e) {
            // TODO: handle exception
            return null;
        }
    }

    public static void closeSession() {
```

```

try {
    session.close();

    } catch (Exception e) {
    // TODO: handle exception
    e.printStackTrace();
    }
    }
    }

```

7) hibernate.cfg.xml

```

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

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

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

<property name = "hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>

<!-- Assume test is the database name -->

<property name = "hibernate.connection.url">jdbc:mysql://localhost:3306/hb1</property>
<property name = "hibernate.connection.username">root</property>

<property name = "hibernate.connection.password"></property>
<property name="show_sql">True</property>

<!-- <property name="show_sql">true</property>
<property name="format_sql">true</property> -->
<!-- List of XML mapping files -->

<mapping class="com.app.model.User"/>
<mapping class="com.app.model.PhoneNumber"/>

```

```
</session-factory>
</hibernate-configuration>
```

8)Test.class

```
package com.app.client;

import com.app.dao.OneToManyDao;
import com.app.factory.OneToManyFactory;

public class Test {
    public static void main(String[] args) {
        OneToManyDao dao = OneToManyFactory.getInstance();
        dao.insertData();
        System.out.println("success");
    }
}
```

Output:

```
Hibernate: alter table phoneNumber_table drop foreign key FKr4axh47mjgkgqs4d7msylwwvi
Hibernate: drop table if exists phoneNumber_table
Hibernate: drop table if exists user_table
Hibernate: create table phoneNumber_table (phone bigint not null, number_type
varchar(255), unid integer, list_index integer, primary key (phone)) engine=InnoDB
Hibernate: create table user_table (User_id integer not null, First_name
varchar(255), lName varchar(255), primary key (User_id)) engine=InnoDB
Hibernate: alter table phoneNumber_table add constraint FKr4axh47mjgkgqs4d7msylwwvi
foreign key (unid) references user_table (User_id)
Hibernate: select p1_0.phone, p1_0.number_type from phoneNumber_table as p1_0 where
p1_0.phone = ?
Hibernate: select p1_0.phone, p1_0.number_type from phoneNumber_table as p1_0 where
p1_0.phone = ?
Hibernate: insert into user_table (First_name, lName, User_id) values (?, ?, ?)
Hibernate: insert into phoneNumber_table (number_type, phone) values (?, ?)
Hibernate: insert into phoneNumber_table (number_type, phone) values (?, ?)
Hibernate: update phoneNumber_table set unid=?, list_index=? where phone=?
Hibernate: update phoneNumber_table set unid=?, list_index=? where phone=?
success
```