

Phase-2 Practice Project: Assisted Practice

6.Demonstrate lazy collection in Hibernate.

Create these tables

```
use ECOMMERCE;
```

```
DROP TABLE IF EXISTS `eproduct`;
```

```
CREATE TABLE `eproduct` (
```

```
  `ID` bigint(20) NOT NULL AUTO_INCREMENT,
```

```
  `name` varchar(100) DEFAULT NULL,
```

```
  `price` decimal(10,2) DEFAULT NULL,
```

```
  `date_added` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

```
  `parts_hdd` varchar(10) DEFAULT NULL,
```

```
  `parts_cpu` varchar(10) DEFAULT NULL,
```

```
  `parts_ram` varchar(10) DEFAULT NULL,
```

```
  PRIMARY KEY (`ID`)
```

```
);
```

```
INSERT INTO `eproduct` VALUES (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57','2 Gb  
HDD','AMD Phenom','4 Gb'),(2,'Acer Laptop ABC',23300.00,'2019-06-04 07:19:07','500 Gb  
HDD','Core-i7','4 Gb'),(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19','1 Tb HDD','Core-  
i7','8 Gb');
```

Color.java

```
package com.ecommerce;
```

```
public class Color {
```

```
        private long COLORID;

private String name;


public Color() {
}

public Color(String name) {
    this.COLORID = 0;
    this.name = name;
}

public long getCOLORID() {return this.COLORID; }
public String getName() { return this.name;}
public void setCOLORID(long id) { this.COLORID = id;}
public void setName(String name) { this.name = name;}
}
```

Finance.java

```
package com.ecommerce;
```

```
public class Finance {

    private long FINANCEID;

    private String name;

    private String ftype;
```

```
    public Finance() {
```

```

    }

    public Finance(String name, String ftype) {
        this.FINANCEID = 0;
        this.name = name;
        this.ftype = ftype;
    }


    public long getFINANCEID() {return this.FINANCEID; }
    public String getName() { return this.name;}
    public String getFtype() { return this.ftype;}
    public void setFINANCEID(long id) { this.FINANCEID = id;}
    public void setName(String name) { this.name = name;}
    public void setFtype(String ftype) { this.ftype= ftype;}

}

```

OS.java

```

package com.ecommerce;

public class OS {
    private long OSID;
    private String name;
    public OS() {
    }
    public OS(String name) {
        this.OSID = 0;
        this.name = name;
    }
}

```

```
}  
  
public long getOSID() {return this.OSID; }  
  
public String getName() { return this.name;}  
  
public void setOSID(long oSID) {  
    OSID = oSID;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
}
```

ScreenSizes.java

```
package com.ecommerce;  
  
public class ScreenSizes {  
    private long SCREENID;  
    private String size;  
  
    public ScreenSizes() {  
  
    }  
  
    public ScreenSizes(String size) {  
        this.SCREENID = 0;  
        this.size = size;  
    }  
}
```

```
public long getSCREENID() {return this.SCREENID; }  
public String getSize() { return this.size;}  
public void setSCREENID(long id) { this.SCREENID = id;}  
public void setSize(String size) { this.size = size;}  
  
}
```

HibernateUtil.java

```
package com.simpli;  
  
import org.hibernate.SessionFactory;  
import org.hibernate.boot.*;  
import org.hibernate.boot.registry.*;  
  
public class HibernateUtil {  
  
    private static final SessionFactory sessionFactory;  
  
    static {  
        try {  
            StandardServiceRegistry standardRegistry = new  
StandardServiceRegistryBuilder()  
                .configure("hibernate.cfg.xml").build();  
  
            Metadata metaData = new  
MetadataSources(standardRegistry).getMetadataBuilder().build();  
        }  
    }  
}
```

```

        sessionFactory = metaData.getSessionFactoryBuilder().build();

    } catch (Throwable th) {

        System.out.println(th);

        throw new ExceptionInInitializerError(th);

    }

}

public static SessionFactory getSessionFactory() {

    return sessionFactory;

}

}

```

Color.hbm.xml

```

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd

```

</hibernate-mapping>

Finance.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
  <class name="Finance" table="fi nance" >
    <id name="FINANCEID" type="long" column="ID">
      <generator class="identity"/>
    </id>
    <property name="name" type="string" column="NAME"/>
    <property name="ftype" type="string" column="FTYPE"/>
  </class>
</hibernate-mapping>
```

EProduct.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
  <class name="EProduct" table="eproduct">
    <id name="ID" type="long" column="ID">
      <generator class="identity"/>
    </id>
  </class>
</hibernate-mapping>
```

</id>

<property name="name" type="string" column="NAME"/>

<property name="price" type="big_decimal" column="PRICE"/>

<property name="dateAdded" type="timestamp" column="DATE_ADDED"/>

<list name="colors" cascade="all" lazy="true">

<key column="product_id" />

<list-index column="idx" />

<one-to-many class="com.ecommerce.Color" />

</list>

<bag name="screensizes" cascade="all" lazy="true">

<key column="product_id"></key>

<one-to-many class="com.ecommerce.ScreenSizes"/>

</bag>

<set name = "os" cascade="all" lazy="true">

<key column = "product_id"/>

<one-to-many class="OS"/>

</set>

<map name = "finance" cascade="all">

<key column = "product_id"/>

<index column = "ftype" type = "string"/>

<one-to-many class="com.ecommerce.Finance"/>

</map>


```
</class>
</hibernate-mapping>

Os.hbm.xml

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
  <class name="OS" table="os">
    <id name="OSID" type="long" column="ID">
      <generator class="identity"/>
    </id>
    <property name="name" type="string" column="NAME"/>
  </class>
</hibernate-mapping>
```

```
ScreenSizes.hbm.xml

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
  <class name="ScreenSizes" table="screensizes">
    <id name="SCREENID" type="long" column="ID">
      <generator class="identity"/>
    </id>
```

```
        <property name="size" type="string" column="SIZE"/>
    </class>
</hibernate-mapping>
```

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>
        <!-- Database connection settings -->
        <property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="connection.url">jdbc:mysql://localhost:3306/ecommerce</property>
        <property name="connection.username">root</property>
        <property name="connection.password">Simplilearn</property>
        <property name="dialect">org.hibernate.dialect.MySQL8Dialect</property>

        <mapping resource="com/ecommerce/Color.hbm.xml"/>
        <mapping resource="com/ecommerce/OS.hbm.xml"/>
        <mapping resource="com/ecommerce/ScreenSizes.hbm.xml"/>
        <mapping resource="com/ecommerce/Finance.hbm.xml"/>
        <mapping resource="com/ecommerce/EProduct.hbm.xml"/>

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

index.html

<title>Hibernate Demo </title>

<h3>Hibernate Configuration Example </h3>

Initialize Hibernate

<h3>Hibernate Query Demo </h3>

Query Here and get the data

<h3>Hibernate Collections Mapping Demo </h3>

Click here

<h3>Hibernate Component Mapping Demo </h3>

Click here

ProductDetails.java

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.transaction.*;
import javax.xml.bind.*;
```

```
import java.io.Serializable;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collection;
import java.util.List;
import java.util.Map;
import java.util.Set;
```

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

```
import com.ecommerce.Color;
import com.ecommerce.EProduct;
import com.ecommerce.Finance;
import com.ecommerce.HibernateUtil;
import com.ecommerce.OS;
```

```
import com.ecommerce.ScreenSizes;
```

```
/**
```

```
 * Servlet implementation class ProductDetails
```

```
 */
```

```
@WebServlet("/ProductDetails")
```

```
public class ProductDetails extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
/**
```

```
 * @see HttpServlet#HttpServlet()
```

```
 */
```

```
    public ProductDetails() {
```

```
        super();
```

```
        // TODO Auto-generated constructor stub
```

```
    }
```

```
/**
```

```
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
```

```
 */
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {
```

```
        // TODO Auto-generated method stub
```

```
        try {
```

```
            SessionFactory factory = HibernateUtil.getSessionFactory();
```

```
Session session = factory.openSession();
```

```
List<EProduct> list = session.createQuery("from EProduct").list();
```

```
PrintWriter out = response.getWriter();
```

```
out.println("<html><body>");
```

```
out.println("<b>Product Listing</b><br>");
```

```
for(EProduct p: list) {
```

```
    out.println("ID: " + String.valueOf(p.getID()) + ", Name: " + p.getName() +
```

```
        ", Price: " + String.valueOf(p.getPrice()) + ", Date Added: " +  
p.getDateAdded().toString());
```

```
List<Color> colors = p.getColors();
```

```
out.println("Colors: ");
```

```
for(Color c: colors) {
```

```
    out.print(c.getName() + "&nbsp;");
```

```
}
```

```
Collection<ScreenSizes> sizes= p.getScreensizes();
```

```
out.println(", Screen Sizes: ");
```

```
for(ScreenSizes s: sizes) {
```

```
    out.print(s.getSize() + "&nbsp;");
```

```
}
```

```
Set<OS> os= p.getOs();
```

```
out.println(", OS : ");  
for(OS o: os) {  
    out.print(o.getName() + "&nbsp;");  
}
```

```
Map finances = p.getFinance();  
out.println(", Finance Options: ");  
if (finances.get("CREDITCARD") != null) {  
    Finance f = (Finance) finances.get("CREDITCARD");  
    out.println(f.getName() + " &nbsp;");  
}  
if (finances.get("BANK") != null) {  
    Finance f = (Finance) finances.get("BANK");  
    out.println(f.getName() + " &nbsp;");  
}
```

```
out.println("<hr>");  
}
```

```
session.close();
```

```
out.println("</body></html>");
```

```
} catch (Exception ex) {  
    throw ex;
```

```
    }

}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}

}
```

Output

http://localhost:18080/HibernateDemo/product-details							
ID	NAME	PRICE	DATE	COLORS	OS	SCREENSIZES	FINANCE
1	HP Laptop ABC	12000.00	2023-05-26 06:51:09.0	Red Silver	Windows 10	12 in	
2	DELL PC ABC	19000.00	2023-05-26 06:51:09.0	Gray White	Windows 10 RedHat Linux FreeDOS	14.5 in 14.9 in	
3	Samsung Laptop PQR	22000.00	2023-05-26 06:51:09.0	Maroon	Windows 10	15.5 in	
4	Phillips Mixer	7000.00	2023-05-26 07:14:31.0				
6	Apple iPad	20000.50	2023-05-26 10:27:16.0				
7	Apple iPad	20000.50	2023-05-26 10:28:07.0				
8	HP Gaming Laptop	100000.00	2023-05-26 11:06:40.0				
9	HP Gaming Laptop 2	200000.00	2023-05-26 11:07:39.0				
10	Mac PC	1.00	2023-05-26 11:22:00.0				
11	HP Camera	2500.00	2023-05-26 11:22:15.0				
12	Mac PC	2.00	2023-05-29 04:54:35.0				

uct-details