Phase – 2 Practice Project: Assisted Practice

26. Demonstrate lazy collection in Hibernate.

- **Code**

✓ **Database Code**

```
create database ecommerce1;


use ecommerce1;


CREATE TABLE eproduct (
        ID bigint primary key auto_increment,
   name varchar(100), price decimal(10,2),
   date_added timestamp default now()
   )


INSERT INTO eproduct(name, price) VALUES ('HP Laptop ABC', 12000);
INSERT INTO eproduct(name, price) VALUES ('Acer Laptop ABC', 14000);
INSERT INTO eproduct(name, price) VALUES ('Lenovo Laptop ABC', 12000);
INSERT INTO eproduct(name, price) VALUES ('Apple Laptop ABC', 18000);


CREATE TABLE `colors` (
`ID` bigint(20) NOT NULL AUTO_INCREMENT,
`color_name` varchar(40) DEFAULT NULL,
`idx` int(11) DEFAULT NULL,
`product_id` bigint(20) DEFAULT NULL,
PRIMARY KEY (`ID`)
);


INSERT INTO `colors` VALUES
(1,'Red',0,1),(2,'Silver',1,1),(3,'Gray',0,2),(4,'White',1,2),(5,'Maroon',0,3);
```

```
CREATE TABLE `screensizes` (
`ID` bigint(20) NOT NULL AUTO_INCREMENT,
`size` varchar(10) DEFAULT NULL,
`product_id` bigint(20) DEFAULT NULL,
PRIMARY KEY (`ID`)
) ;

INSERT INTO `screensizes` VALUES (1,'12 in',1),(2,'14.5 in',2),(3,'14.9 in',2),(4,'15.5 in',3);

CREATE TABLE `os` (
`ID` bigint(20) NOT NULL AUTO_INCREMENT,
`name` varchar(30) DEFAULT NULL,
`product_id` bigint(20) DEFAULT NULL,
PRIMARY KEY (`ID`)
);

INSERT INTO `os` VALUES (1,'Windows 10',1),(2,'Windows 10',2),
(3,'FreeDOS',2),(4,'RedHat Linux',2),(5,'Windows 10',3);

CREATE TABLE `finance` (
`ID` bigint(20) NOT NULL AUTO_INCREMENT,
`ftype` varchar(10) DEFAULT NULL,
`name` varchar(30) DEFAULT NULL,
`product_id` bigint(20) DEFAULT NULL,
PRIMARY KEY (`ID`)
) ;

INSERT INTO `finance` VALUES (1,'CREDITCARD','EMI on Citibank
Card',1),(3,'BANK','40% finance from SBI',2), (4,'BANK','60% finance from ICICI',3),
(5,'BANK','20% finance from ICICI',1);
```

✓ **Color.java**

```
package com.ecomerce;

public class Color {

	private long COLORID;
	private String name;

	public Color() {

	}

	public long getCOLORID() {
		return COLORID;
	}
```

```
	public void setCOLORID(long cOLORID) {
		COLORID = cOLORID;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

}
```

✓ EProduct.java

```
package com.ecomerce;

import java.math.BigDecimal;
import java.util.*;

public class EProduct {

	private long ID;
	private String name;
	private BigDecimal price;
	private Date dateAdded;

	private List<Color> colors;

	private Set<OS> os;

	private Collection<ScreenSizes> screenSizes;

	private Map finance;

	public EProduct() {

	}

	public long getID() {
		return ID;
	}

	public void setID(long iD) {
		ID = iD;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}
```

```
    public BigDecimal getPrice() {
        return price;
    }

    public void setPrice(BigDecimal price) {
        this.price = price;
    }

    public Date getDateAdded() {
        return dateAdded;
    }

    public void setDateAdded(Date dateAdded) {
        this.dateAdded = dateAdded;
    }

    public List<Color> getColors() {
        return colors;
    }

    public void setColors(List<Color> colors) {
        this.colors = colors;
    }

    public Set<OS> getOs() {
        return os;
    }

    public void setOs(Set<OS> os) {
        this.os = os;
    }

    public Collection<ScreenSizes> getScreenSizes() {
        return screenSizes;
    }

    public void setScreenSizes(Collection<ScreenSizes> screenSizes) {
        this.screenSizes = screenSizes;
    }

    public Map getFinance() {
        return finance;
    }

    public void setFinance(Map finance) {
        this.finance = finance;
    }

}
```

✓ **Finanace.java**

```
package com.ecomerce;

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.ecomerce;

public class OS {

     private long OSID;
     private String name;

     public OS() {

     }

     public OS(long oSID, String name) {
          super();
          OSID = oSID;
          this.name = name;
     }

     public long getOSID() {
          return OSID;
     }

     public void setOSID(long oSID) {
          OSID = oSID;
     }

     public String getName() {
```

```
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

}
```

✓ **ScreenSizes.java**

```
package com.ecomerce;

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) {
                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 package="com.ecomerce">
      <class name="Color" table="colors">
            <id name="COLORID" type="long" column="ID">
                  <generator class="identity" />
            </id>
            <property name="name" type="string" column="COLOR_NAME" />
      </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.ecomerce">
      <class name="EProduct" table="eproduct">
            <id name="ID" column="ID">
                  <generator class="increment" />
            </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.ecomerce.Color" />
            </list>

            <set name="os" cascade="all" lazy="true">
                  <key column="product_id" />
                  <one-to-many class="OS" />
            </set>
```

```
            <bag name="screenSizes" cascade="all" lazy="true">
                 <key column="product_id"></key>
                 <one-to-many class="com.ecomerce.ScreenSizes" />
            </bag>

            <map name="finance" cascade="all">
                 <key column="product_id" />
                 <index column="ftype" type="string" />
                 <one-to-many class="com.ecomerce.Finance" />
            </map>

      </class>
</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.ecomerce">
    <class name="Finance" table="finance" >
        <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>
```

✓ 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.ecomerce">
     <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.ecomerce">
    <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/ecommerce1</property>
    <property name="connection.username">root</property>
    <property name="connection.password">Arun121212</property>

     <mapping resource="com/ecomerce/EProduct.hbm.xml"></mapping>

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

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

✓ index.html

```
<br> <h3> Hibernate Mapping  Demo</h3>
<a href="product-details"> Hibernate Mapping Demo</a><br>
```

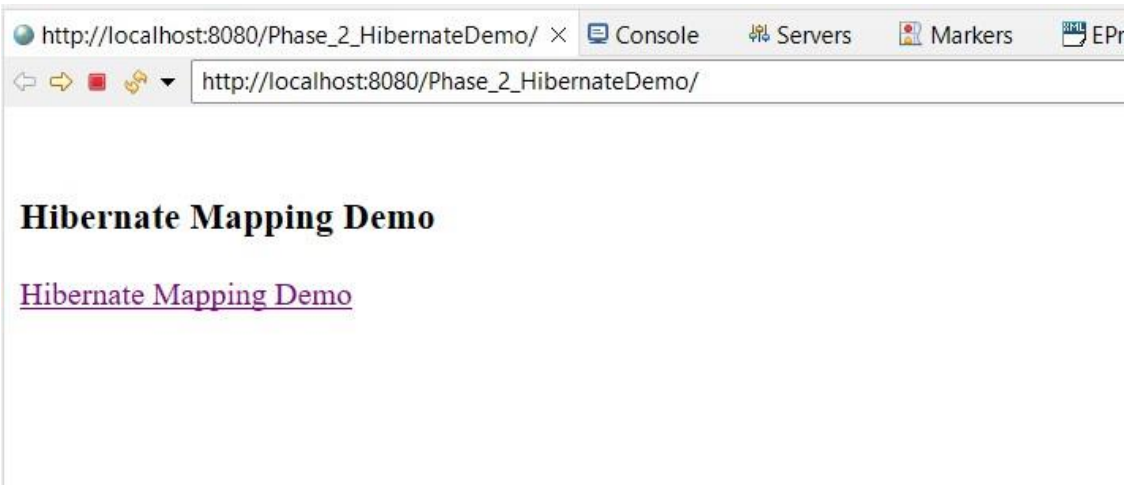
✓ ProductDetailsServlet.java

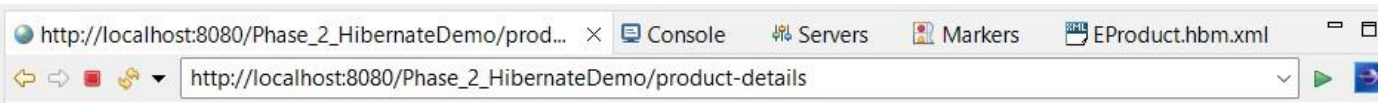
```
package com.simpli;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.*;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

import org.hibernate.*;
```

```
import com.ecomerce.EProduct;
import com.ecomerce.*;

@WebServlet("/product-details")
public class ProductDetailsServlet extends HttpServlet {

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

          PrintWriter out = response.getWriter();
          out.println("<html><body>");

          // STEP 1: Get a Session (connection) from the Session Factory
class
          SessionFactory factory = HibernateUtil.getSessionFactory();
          // STEP 2 Session
          Session session = factory.openSession();

          // STEP 32 execute the HQL commands
          // for now we will only test if the connection is establised
with MySQL server.
          List<EProduct> eproducts = session.createQuery("from
EProduct").list();

          out.println("<br> Data from the eproduct table<table
border=1>");
          out.println("<th> ID <th> NAME <th> PRICE <th> DATE ADDED <th>
COLORS <th> Screen Sizes <th> OS  <th> FINANCE OPTIONS </th>    ");
          for (EProduct prod : eproducts) {

               // Display Core properties/details
               out.println("<tr><td>" + prod.getID() + "<td>" +
prod.getName() + "<td>" + prod.getPrice() + "<td>"
                         + prod.getDateAdded());

               // Display the available colors
               List<Color> colors = prod.getColors();
               out.println("<td> ");
               for(Color color: colors)
                    out.println(color.getName() + "  ");

               // Display the available screensizes
               Collection<ScreenSizes> screenSizes =
prod.getScreenSizes();
               out.println("<td> " );
               for(ScreenSizes sSize: screenSizes)
                    out.println(sSize.getSize() + "  ");

               // Display the available OSes
               Set<OS> OSes = prod.getOs();
               out.println("<td>   ");
               for(OS os: OSes)
                    out.println(os.getName() + "  ");

               // Display the available finance options
```

```
                Map finances = prod.getFinance();
            out.println("<td>  ");
            if (finances .get("CREDITCARD") != null) {
                    Finance f = (Finance) finances .get("CREDITCARD");
                    out.println(f.getName() + "  ");
            }
            if (finances .get("BANK") != null) {
                    Finance f = (Finance) finances .get("BANK");
                    out.println(f.getName() + "  ");
            }



        }

        session.close();

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

    }

}
```

- **Output**





Data from the eproduct table

ID	NAME	PRICE	DATE ADDED	COLORS	Screen Sizes	OS	FINANCE OPTIONS
1	HP Laptop ABC	12000.00	2023-05-30 13:05:39.0	Red Silver	12 in	Windows 10	EMI on Citibank Card 20% finance from ICICI
2	Acer Laptop ABC	14000.00	2023-05-30 13:05:39.0	Gray White	14.5 in 14.9 in	FreeDOS RedHat Linux Windows 10	40% finance from SBI
3	Lenovo Laptop ABC	12000.00	2023-05-30 13:05:39.0	Maroon	15.5 in	Windows 10	60% finance from ICICI
4	Apple Laptop ABC	18000.00	2023-05-30 17:33:06.0				