Phase-2 Practice Project: Assisted Practice

4.Demonstrate Hibernate logging by Log4j.

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>

     <mapping class="com.ecommerce.EProduct" />
  </session-factory>
</hibernate-configuration>
```

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;
    }

}
```

HibernateQueryDemo.java

```
package com.simpli;

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

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

import org.hibernate.*;

import com.ecommerce.EProduct;

@WebServlet("/query-demo")
public class HibernateQueryDemo 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();


                // STE2 Create the session object

                Session session = factory.openSession();


                out.println("Hibernate Session opened.<br>");


                // STEP 3 Query the DB and get the data

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

out.println("<table border=1>");

for (EProduct prod : eproducts) {

        out.println("<tr>" + "<td>" + prod.getID() + "<td>" + prod.getName() + "<td>" +
prod.getPrice() + "<td>"

                        + prod.getDateAdded());

}

                out.println("</table>");


                session.close();
                out.println("Hibernate Session closed.<br>");


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


}
```

EPRODUCT.JAVA

```
package com.ecommerce;


import java.math.BigDecimal;
import java.util.Date;


import javax.persistence.*;


@Entity
@Table(name = "eproduct")
public class EProduct {


        @Id
        @GeneratedValue
        @Column(name = "ID")
        private long ID;


        @Column(name = "name")
```

```
private String name;

@Column(name = "price")
private BigDecimal price;

@Column(name = "date_added")
private Date dateAdded;

public EProduct() {

}

public EProduct(long id, String name, BigDecimal price, Date dateAdded) {
	this.ID = id;
	this.name = name;
	this.price = price;
	this.dateAdded = dateAdded;
}

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;

        }


}
```

Index.html

```
<title>Hibernate Demo  </title>

<br><br>

<h3>Hibernate Annotation Query Demo </h3>


<a href="query-demo">Query  Here and get the data</a><br>
```

Log4j.properties

```
# LOG4J configuration

log4j.rootLogger=DEBUG, Appender1,Appender2

log4j.appender.Appender1=org.apache.log4j.ConsoleAppender

log4j.appender.Appender1.layout=org.apache.log4j.PatternLayout

log4j.appender.Appender1.layout.ConversionPattern=%-7p %d [%t] %c %x - %m%n


log4j.appender.Appender2=org.apache.log4j.FileAppender

log4j.appender.Appender2.File=/home/jyotidevadigamp/Desktop/Mylog.txt

log4j.appender.Appender2.layout=org.apache.log4j.PatternLayout

log4j.appender.Appender2.layout.ConversionPattern=%-7p %d [%t] %c %x - %m%n
```

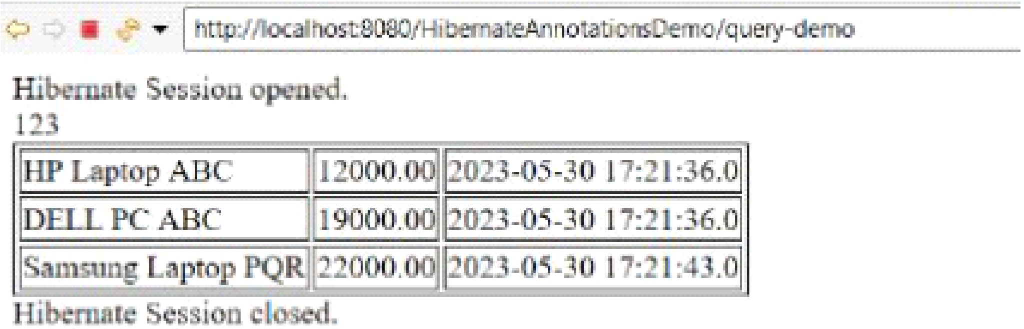
# Log everything. Good for troubleshooting

log4j.logger.org.hibernate=INFO

# Log all JDBC parameters

log4j.logger.org.hibernate.type=ALL

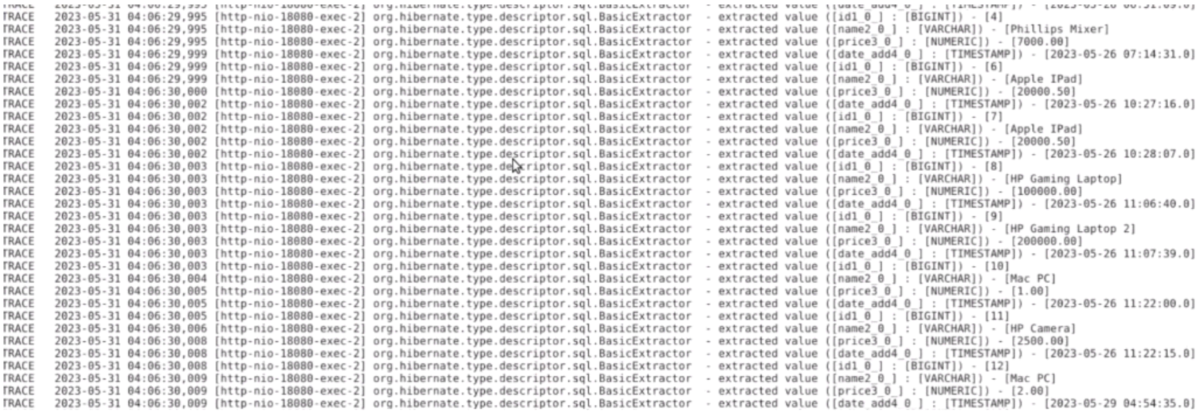
Output



Hibernate Session opened.
123

HP Laptop ABC	12000.00	2023-05-30 17:21:36.0
DELL PC ABC	19000.00	2023-05-30 17:21:36.0
Samsung Laptop PQR	22000.00	2023-05-30 17:21:43.0

Hibernate Session closed.



```
TRACE   2023-05-31 04:06:29,995 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([id1_0_] : [BIGINT]) - [4]
TRACE   2023-05-31 04:06:29,995 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([name2_0_] : [VARCHAR]) - [Phillips Mixer]
TRACE   2023-05-31 04:06:29,995 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([price3_0_] : [NUMERIC]) - [7000.00]
TRACE   2023-05-31 04:06:29,999 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([date_add4_0_] : [TIMESTAMP]) - [2023-05-26 07:14:31.0]
TRACE   2023-05-31 04:06:29,999 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([id1_0_] : [BIGINT]) - [6]
TRACE   2023-05-31 04:06:29,999 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([name2_0_] : [VARCHAR]) - [Apple IPad]
TRACE   2023-05-31 04:06:30,000 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([price3_0_] : [NUMERIC]) - [20000.50]
TRACE   2023-05-31 04:06:30,002 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([date_add4_0_] : [TIMESTAMP]) - [2023-05-26 10:27:16.0]
TRACE   2023-05-31 04:06:30,002 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([id1_0_] : [BIGINT]) - [7]
TRACE   2023-05-31 04:06:30,002 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([name2_0_] : [VARCHAR]) - [Apple IPad]
TRACE   2023-05-31 04:06:30,002 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([price3_0_] : [NUMERIC]) - [20000.50]
TRACE   2023-05-31 04:06:30,002 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([date_add4_0_] : [TIMESTAMP]) - [2023-05-26 10:28:07.0]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([id1_0_] : [BIGINT]) - [8]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([name2_0_] : [VARCHAR]) - [HP Gaming Laptop]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([price3_0_] : [NUMERIC]) - [100000.00]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([date_add4_0_] : [TIMESTAMP]) - [2023-05-26 11:06:40.0]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([id1_0_] : [BIGINT]) - [9]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([name2_0_] : [VARCHAR]) - [HP Gaming Laptop 2]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([price3_0_] : [NUMERIC]) - [200000.00]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([date_add4_0_] : [TIMESTAMP]) - [2023-05-26 11:07:39.0]
TRACE   2023-05-31 04:06:30,003 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([id1_0_] : [BIGINT]) - [10]
TRACE   2023-05-31 04:06:30,004 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([name2_0_] : [VARCHAR]) - [Mac PC]
TRACE   2023-05-31 04:06:30,005 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([price3_0_] : [NUMERIC]) - [1.00]
TRACE   2023-05-31 04:06:30,005 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([date_add4_0_] : [TIMESTAMP]) - [2023-05-26 11:22:00.0]
TRACE   2023-05-31 04:06:30,005 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([id1_0_] : [BIGINT]) - [11]
TRACE   2023-05-31 04:06:30,006 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([name2_0_] : [VARCHAR]) - [HP Camera]
TRACE   2023-05-31 04:06:30,008 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([price3_0_] : [NUMERIC]) - [2500.00]
TRACE   2023-05-31 04:06:30,008 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([date_add4_0_] : [TIMESTAMP]) - [2023-05-26 11:22:15.0]
TRACE   2023-05-31 04:06:30,008 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([id1_0_] : [BIGINT]) - [12]
TRACE   2023-05-31 04:06:30,009 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([name2_0_] : [VARCHAR]) - [Mac PC]
TRACE   2023-05-31 04:06:30,009 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([price3_0_] : [NUMERIC]) - [2.00]
TRACE   2023-05-31 04:06:30,009 [http-nio-18080-exec-2] org.hibernate.type.descriptor.sql.BasicExtractor  - extracted value ([date_add4_0_] : [TIMESTAMP]) - [2023-05-29 04:54:35.0]
```