**pom.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.hibernate</groupId>

<artifactId>Day\_1\_HibernateUsingXML\_PracticeExercise</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>PracticeExerciseXML</name>

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.5.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.20</version>

</dependency>

</dependencies>

</project>

**product.hbm.xml**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

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

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

<hibernate-mapping>

<class name=*"com.mphasis.domain.Product"* table=*"products"*>

<id name=*"productId"* type=*"java.lang.String"*>

<column name=*"pdct\_Id"*/>

<generator class=*"assigned"*/>

</id>

<property name=*"productName"* type=*"java.lang.String"*>

<column name=*"pdct\_Name"*/>

</property>

<property name=*"productPrice"* type=*"double"*>

<column name=*"pdct\_Price"*/>

</property>

</class>

</hibernate-mapping>

**Product.java**

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE hibernate-mapping PUBLIC

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

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

<hibernate-mapping>

<**class** name="com.mphasis.domain.Product" table="products">

<id name="productId" type="java.lang.String">

<column name="pdct\_Id"/>

<generator **class**="assigned"/>

</id>

<property name="productName" type="java.lang.String">

<column name="pdct\_Name"/>

</property>

<property name="productPrice" type="double">

<column name="pdct\_Price"/>

</property>

</**class**>

</hibernate-mapping>

**Product.java**

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE hibernate-mapping PUBLIC

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

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

<hibernate-mapping>

<**class** name="com.mphasis.domain.Product" table="products">

<id name="productId" type="java.lang.String">

<column name="pdct\_Id"/>

<generator **class**="assigned"/>

</id>

<property name="productName" type="java.lang.String">

<column name="pdct\_Name"/>

</property>

<property name="productPrice" type="double">

<column name="pdct\_Price"/>

</property>

</**class**>

</hibernate-mapping>

**HibernateUtil.java**

**package** com.mphasis.dbutil;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**import** com.mphasis.domain.Product;

**public** **class** HibernateUtil {

**private** **static** SessionFactory *factory*;

**public** **static** SessionFactory getSessionFactory() {

**if**(*factory* == **null**) {

*factory* = **new** Configuration()

.configure("hibernate.cfg.xml")

.addAnnotatedClass(Product.**class**)

.buildSessionFactory();

}

**return** *factory*;

} }

**productManagementDAO.java**

**package** com.mphasis.dao;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** com.mphasis.dbutil.HibernateUtil;

**import** com.mphasis.domain.Product;

**public** **class** ProductManagementDAO {

SessionFactory factory = HibernateUtil.getSessionFactory();

**public** **void** getAllRecords() {

Session session = factory.openSession();

session.beginTransaction();

List<Product> results = session.createQuery("FROM Product").getResultList();

**if**(results.size() > 0) {

**for**(Product p: results) {

System.out.println("Product Id: " + p.getProductId());

System.out.println("Product Name: " + p.getProductName());

System.out.println("Product Price: " + p.getProductPrice());

System.out.println();

}

}

**else** {

System.out.println("No records found insert the products");

}

session.getTransaction().commit();

session.close();

}

**public** **boolean** insertRecord(Product p) {

**try** {

Session session = factory.openSession();

session.beginTransaction();

session.save(p);

session.getTransaction().commit();

session.close();

**return** **true**;

}

**catch**(Exception e) {

System.out.println("Product is not added");

}

**return** **false**;

}

**public** **boolean** updateRecord(String prdId, String prdName, **double** prdPrice) {

**try** {

Session session = factory.openSession();

session.beginTransaction();

Product p = session.get(Product.**class**, prdId);

p.setProductName(prdName);

p.setProductPrice(prdPrice);

session.saveOrUpdate(p);

session.getTransaction().commit();

session.close();

**return** **true**;

}

**catch**(Exception e) {

System.***out***.println("Product is not found to update");

}

**return** **false**;

}

**public** **boolean** deleteRecord(String prdId) {

**try** {

Session session = factory.openSession();

session.beginTransaction();

Product p = session.get(Product.**class**, prdId);

session.delete(p);

session.getTransaction().commit();

session.close();

**return** **true**;

}

**catch**(Exception e) {

System.***out***.println("Product not found to delete");

}

**return** **false**;

}

**public** **void** searchRecord(String prdId) {

**try** {

Session session = factory.openSession();

session.beginTransaction();

Product p = session.get(Product.**class**, prdId);

System.***out***.println("Product Id: " + p.getProductId());

System.***out***.println("Product Name: " +p.getProductName());

System.***out***.println("Product Price: " + p.getProductPrice());

session.close();

}

**catch**(Exception e) {

System.***out***.println("Product is not found");

} } }

**ProductManagementApp.java**

**package** com.mphasis.dao;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** com.mphasis.dbutil.HibernateUtil;

**import** com.mphasis.domain.Product;

**public** **class** ProductManagementDAO {

SessionFactory factory = HibernateUtil.getSessionFactory();

**public** **void** getAllRecords() {

Session session = factory.openSession();

session.beginTransaction();

List<Product> results = session.createQuery("FROM Product").getResultList();

**if**(results.size() > 0) {

**for**(Product p: results) {

System.out.println("Product Id: " + p.getProductId());

System.out.println("Product Name: " + p.getProductName());

System.out.println("Product Price: " + p.getProductPrice());

System.out.println();

}

}

**else** {

System.out.println("No records found insert products");

}

session.getTransaction().commit();

session.close();

}

**public** **boolean** insertRecord(Product p) {

**try** {

Session session = factory.openSession();

session.beginTransaction();

session.save(p);

session.getTransaction().commit();

session.close();

**return** **true**;

}

**catch**(Exception e) {

System.out.println("Product is not added");

}

**return** **false**;

}

**public** **boolean** updateRecord(String prdId, String prdName, **double** prdPrice) {

**try** {

Session session = factory.openSession();

session.beginTransaction();

Product p = session.get(Product.**class**, prdId);

p.setProductName(prdName);

p.setProductPrice(prdPrice);

session.saveOrUpdate(p);

session.getTransaction().commit();

session.close();

**return** **true**;

}

**catch**(Exception e) {

System.out.println("Product is not found to update");

}

**return** **false**;

}

**public** **boolean** deleteRecord(String prdId) {

**try** {

Session session = factory.openSession();

session.beginTransaction();

Product p = session.get(Product.**class**, prdId);

session.delete(p);

session.getTransaction().commit();

session.close();

**return** **true**;

}

**catch**(Exception e) {

System.out.println("Product is not found to delete");

}

**return** **false**;

}

**public** **void** searchRecord(String prdId) {

**try** {

Session session = factory.openSession();

session.beginTransaction();

Product p = session.get(Product.**class**, prdId);

System.out.println("Product Id: " + p.getProductId());

System.out.println("Product Name: " +p.getProductName());

System.out.println("Product Price: " + p.getProductPrice());

session.close();

}

**catch**(Exception e) {

System.out.println("Product not found");

}

}

}