

Hibernate Query Language, Using HQL Select Query

Product.java

```
1 package str;
2
3 public class Product implements java.io.Serializable{
4
5     private static final long serialVersionUID = 1L;
6
7     private int productId;
8     private String proName;
9     private double price;
10
11     public void setProductId(int productId)
12     {
13         this.productId = productId;
14     }
15     public int getProductId()
16     {
17         return productId;
18     }
19
20     public void setProName(String proName)
21     {
22         this.proName = proName;
23     }
24     public String getProName()
25     {
26         return proName;
27     }
28
29     public void setPrice(double price)
30     {
31         this.price = price;
32     }
33     public double getPrice()
34     {
35         return price;
36     }
37 }
```

hibernate.cfg.xml

```
~
6 <hibernate-configuration>
7 <session-factory>
8 <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver
9 </property>
10 <property
    name="connection.url">jdbc:oracle:thin:@localhost:1521:XE</property>
11 <property name="connection.username">oe</property>
12 <property name="connection.password">oe</property>
13
14 <property name="dialect">org.hibernate.dialect.OracleDialect</property>
15 <property name="show_sql">true</property>
16 <property name="hbm2ddl.auto">update</property>
17
18 <mapping resource="Product.hbm.xml"></mapping>
19 </session-factory>
20 </hibernate-configuration>
```

SM

Product.hbm.xml

```
1 <?xml version="1.0"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
5
6 <hibernate-mapping>
7 <class name="str.Product" table="products">
8
9 <id name="productId" column="pid" />
10 <property name="proName" column="pname" length="10"/>
11 <property name="price"/>
12
13 </class>
14 </hibernate-mapping>
```

ForOurLogic.java

```
1 package str;
2
3 import org.hibernate.*;
4 import org.hibernate.cfg.*;
5 import java.util.*;
6
7 public class ForOurLogic {
8
9     public static void main(String[] args)
10    {
11
12        Configuration cfg = new Configuration();
13        cfg.configure("hibernate.cfg.xml");
14
15        SessionFactory factory = cfg.buildSessionFactory();
16        Session session = factory.openSession();
17
18        /* Selecting all objects(records) start_____ */
19
20        /*
21
22        Query qry = session.createQuery("from Product p");
23
24        List l =qry.list();
25        System.out.println("Total Number Of Records : "+l.size());
26        Iterator it = l.iterator();
27
28        while(it.hasNext())
29        {
30            Object o = (Object)it.next();
31            Product p = (Product)o;
32            System.out.println("Product id : "+p.getProductid());
33            System.out.println("Product Name : "+p.getProName());
34            System.out.println("Product Price : "+p.getPrice());
35            System.out.println("-----");
36        }
37
38        */
39
40        /* Selecting all objects(records) end_____ */
41
42        /* Selecting partial objects(More than one object) start_____ */
43
44        /*
```

```
45
46 Query qry = session.createQuery("select p.productId,p.proName from Product p");
47
48 List l =qry.list();
49 System.out.println("Total Number Of Records : "+l.size());
50 Iterator it = l.iterator();
51
52 while(it.hasNext())
53 {
54     Object o[] = (Object[])it.next();
55
56     System.out.println("Product id : "+o[0]+ "Product Name : "+o[1]);
57
58     System.out.println("-----");
59 }
60
61 */
62
63 /* Selecting partial objects(More than one object)end_____ */
64
65 // Selecting single object start_____
```

```
66
67     Query qry = session.createQuery("select p.productId
68 from Product p");
69
70 List l =qry.list();
71 System.out.println("Total Number Of Records : "+l.size());
72 Iterator it = l.iterator();
73
74 while(it.hasNext())
75 {
76     Integer i = (Integer)it.next();
77     System.out.println("Product id : "+i.intValue());
78     System.out.println("-----");
79 }
80
81 // selecting single object end_____
82
83 session.close();
84 factory.close();
85 }
86
87 }
88
```

SM

Hibernate Query Language, Passing Runtime Values

Product.java

```
1 package str;
2
3 public class Product implements java.io.Serializable{
4
5     private static final long serialVersionUID = 1L;
6
7     private int productId;
8     private String proName;
9     private double price;
10
11     public void setProductId(int productId)
12     {
13         this.productId = productId;
14     }
15     public int getProductId()
16     {
17         return productId;
18     }
19
20     public void setProName(String proName)
21     {
22         this.proName = proName;
23     }
24     public String getProName()
25     {
26         return proName;
27     }
28
29     public void setPrice(double price)
30     {
31         this.price = price;
32     }
33     public double getPrice()
34     {
35         return price;
36     }
37 }
```

hibernate.cfg.xml

```
~
6 <hibernate-configuration>
7 <session-factory>
8 <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver
9 </property>
10 <property
    name="connection.url">jdbc:oracle:thin:@localhost:1521:XE</property>
11 <property name="connection.username">oe</property>
12 <property name="connection.password">oe</property>
13
14 <property name="dialect">org.hibernate.dialect.OracleDialect</property>
15 <property name="show_sql">true</property>
16 <property name="hbm2ddl.auto">update</property>
17
18 <mapping resource="Product.hbm.xml"></mapping>
19 </session-factory>
20 </hibernate-configuration>
```

SM

Product.hbm.xml

```
1 <?xml version="1.0"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
5
6 <hibernate-mapping>
7 <class name="str.Product" table="products">
8
9 <id name="productId" column="pid" />
10 <property name="proName" column="pname" length="10"/>
11 <property name="price"/>
12
13 </class>
14 </hibernate-mapping>
```

```
1 package str;
2
3 import org.hibernate.*;
4 import org.hibernate.cfg.*;
5 import java.util.*;
6
7 public class ForOurLogic {
8
9     public static void main(String[] args)
10    {
11
12        Configuration cfg = new Configuration();
13        cfg.configure("hibernate.cfg.xml");
14
15        SessionFactory factory = cfg.buildSessionFactory();
16        Session session = factory.openSession();
17
18        /* Using label.....
19
20        Query qry = session.createQuery("from Product p where p.productId=
21        :prdId");
22        qry.setParameter("prdId",105);
23
24        List l =qry.list();
25        System.out.println("Total Number Of Records : "+l.size());
26        Iterator it = l.iterator();
27
28        while(it.hasNext())
29        {
30            Object o = (Object) it.next();
31            Product p = (Product)o;
32            System.out.println("Product Name : "+p.getProName());
33            System.out.println("Product Price : "+p.getPrice());
34            System.out.println("-----");
35        }
36
37    */
38
39    /* Using Question Mark */
40
41    Query qry = session.createQuery("from Product p where p.productId= ?");
42    qry.setParameter(0,105);
43
44    List l =qry.list();
45    System.out.println("Total Number Of Records : "+l.size());
46    Iterator it = l.iterator();
47
48    while(it.hasNext())
49    {
50        Object o = (Object) it.next();
51        Product p = (Product)o;
52        System.out.println("Product Name : "+p.getProName());
53        System.out.println("Product Price : "+p.getPrice());
54        System.out.println("-----");
```

```
55  
56 }  
57  
58 session.close();  
59 factory.close();  
60 }  
61  
62 }  
63  
64  
65 |  
66
```



Hibernate Query Language, HQL Update,Delete Queries

Product.java

```
1 package str;
2
3 public class Product implements java.io.Serializable{
4
5     private static final long serialVersionUID = 1L;
6
7     private int productId;
8     private String proName;
9     private double price;
10
11     public void setProductId(int productId)
12     {
13         this.productId = productId;
14     }
15     public int getProductId()
16     {
17         return productId;
18     }
19
20     public void setProName(String proName)
21     {
22         this.proName = proName;
23     }
24     public String getProName()
25     {
26         return proName;
27     }
28
29     public void setPrice(double price)
30     {
31         this.price = price;
32     }
33     public double getPrice()
34     {
35         return price;
36     }
37 }
```

hibernate.cfg.xml

```
~
6 <hibernate-configuration>
7 <session-factory>
8 <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver
9 </property>
10 <property
    name="connection.url">jdbc:oracle:thin:@localhost:1521:XE</property>
11 <property name="connection.username">oe</property>
12 <property name="connection.password">oe</property>
13
14 <property name="dialect">org.hibernate.dialect.OracleDialect</property>
15 <property name="show_sql">>true</property>
16 <property name="hbm2ddl.auto">update</property>
17
18 <mapping resource="Product.hbm.xml"></mapping>
19 </session-factory>
20 </hibernate-configuration>
```

SM

Product.hbm.xml

```
1 <?xml version="1.0"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
5
6 <hibernate-mapping>
7 <class name="str.Product" table="products">
8
9 <id name="productId" column="pid" />
10 <property name="proName" column="pname" length="10"/>
11 <property name="price"/>
12
13 </class>
14 </hibernate-mapping>
```

```
1 package str;
2
3 import org.hibernate.*;
4 import org.hibernate.cfg.*;
5
6 public class ForOurLogic {
7
8     public static void main(String[] args)
9     {
10
11         Configuration cfg = new Configuration();
12         cfg.configure("hibernate.cfg.xml");
13
14         SessionFactory factory = cfg.buildSessionFactory();
15         Session session = factory.openSession();
16
17         Query qry = session.createQuery("delete from Product p
18 where p.productId=:prdId");
19         qry.setParameter("prdId",110);
20         int res = qry.executeUpdate();
21
22         System.out.println("Command successfully executed...");
23         System.out.println("Numer of records effected due to delete
24 query"+res);
25
26         session.close();
27         factory.close();
28     }
29 }
```