

HCQL (Hibernate Criteria Query Language)

The Hibernate Criteria Query Language (HCQL) is used to fetch the records based on the specific criteria. The Criteria interface provides methods to apply criteria such as retrieving all the records of table whose salary is greater than 50000 etc.

Advantage of HCQL

The HCQL provides methods to add criteria, so it is **easy** for the java programmer to add criteria. The java programmer is able to add many criteria on a query.

Criteria Interface

The Criteria interface provides many methods to specify criteria. The object of Criteria can be obtained by calling the **createCriteria()** method of Session interface.

Syntax of createCriteria() method of Session interface

```
public Criteria createCriteria(Class c)
```

The commonly used methods of Criteria interface are as follows:

1. **public Criteria add(Criterion c)** is used to add restrictions.
2. **public Criteria addOrder(Order o)** specifies ordering.
3. **public Criteria setFirstResult(int firstResult)** specifies the first number of record to be retrieved.
4. **public Criteria setMaxResult(int totalResult)** specifies the total number of records to be retrieved.
5. **public List list()** returns list containing object.
6. **public Criteria setProjection(Projection projection)** specifies the projection.

Restrictions class

Restrictions class provides methods that can be used as Criterion. The commonly used methods of Restrictions class are as follows:

1. **public static SimpleExpression lt(String propertyName, Object value)** sets the **less than** constraint to the given property.
2. **public static SimpleExpression le(String propertyName, Object value)** sets the **less than or equal** constraint to the given property.
3. **public static SimpleExpression gt(String propertyName, Object value)** sets the **greater than** constraint to the given property.
4. **public static SimpleExpression ge(String propertyName, Object value)** sets the **greater than or equal** than constraint to the given property.
5. **public static SimpleExpression ne(String propertyName, Object value)** sets the **not equal** constraint to the given property.
6. **public static SimpleExpression eq(String propertyName, Object value)** sets the **equal** constraint to the given property.
7. **public static Criterion between(String propertyName, Object low, Object high)** sets the **between** constraint.
8. **public static SimpleExpression like(String propertyName, Object value)** sets the **like** constraint to the given property.

Order class

The Order class represents an order. The commonly used methods of Restrictions class are as follows:

1. **public static Order asc(String propertyName)** applies the ascending order on the basis of given property.
2. **public static Order desc(String propertyName)** applies the descending order on the basis of given property.

Examples of Hibernate Criteria Query Language

There are given a lot of examples of HCQL.

Example of HCQL to get all the records

```
Criteria c=session.createCriteria(Emp.class);//passing Class class argument
```

```
List list=c.list();
```

Example of HCQL to get the 10th to 20th record

```
Criteria c=session.createCriteria(Emp.class);  
c.setFirstResult(10);  
c.setMaxResult(20);  
List list=c.list();
```

Example of HCQL to get the records whose salary is greater than 10000

```
Criteria c=session.createCriteria(Emp.class);  
c.add(Restrictions.gt("salary",10000));//salary is the propertyname  
List list=c.list();
```

Example of HCQL to get the records in ascending order on the basis of salary

```
Criteria c=session.createCriteria(Emp.class);  
c.addOrder(Order.asc("salary"));  
List list=c.list();
```

HCQL with Projection

We can fetch data of a particular column by projection such as name etc. Let's see the simple example of projection that prints data of NAME column of the table only.

```
Criteria c=session.createCriteria(Emp.class);  
c.setProjection(Projections.property("name"));  
List list=c.list();
```

← prev

next →

Share 0