

HQL

HQL is an object-oriented query language, similar to SQL, but instead of operating on tables and columns, HQL works with persistent objects and their properties.

Advantage of HQL

There are many advantages of HQL. They are as follows:

- database independent
- supports polymorphic queries
- easy to learn for Java Programmer

Query Interface

It is an object oriented representation of Hibernate Query. The object of Query can be obtained by calling the `createQuery()` method Session interface.

The query interface provides many methods. There is given commonly used methods:

1. **`public int executeUpdate()`** is used to execute the update or delete query.
2. **`public List list()`** returns the result of the relation as a list.
3. **`public Query setFirstResult(int rowno)`** specifies the row number from where record will be retrieved.
4. **`public Query setMaxResult(int rowno)`** specifies the no. of records to be retrieved from the relation (table).
5. **`public Query setParameter(int position, Object value)`** it sets the value to the JDBC style query parameter.
6. **`public Query setParameter(String name, Object value)`** it sets the value to a named query parameter.

Update:

```
Query query=session.createQuery("update Employee set age=:age where  
name=:name");  
query.setInteger("age", 32);  
query.setString("name", "Lokesh Gupta");
```

```
int modifications=query.executeUpdate();
```

Delete:

```
Query query=session.createQuery("delete from Account where acctStatus=:status");  
query.setString("acctStatus", "PURGED");  
int rowsDeleted=query.executeUpdate();
```

Fetch All records:

```
Query query=session.createQuery("from Emp");  
List list=query.list();
```

Named Parameters:

```
String hql = "from Product where price > :price";  
Query query = session.createQuery(hql);  
query.setDouble("price",25.0);  
List results = query.list();
```