**1.HQL:-**

HQL stands for Hibernate Query language.

HQL is provided by hibernate in order to perform the bulk records manipulations.

It is object oriented Query language. It is able to support the object oriented features like polymorphism,abstraction,encapsulation…etc.

HQL is data base independent query language.But SQL is Database dependent language. HQL query syntax is almost all same as SQL query syntax with small changes.

In SQL, Developer prepated SQL query using table name and table provided columns directly where as In HQL, Developer propers SQL query using domain/persistence class name and persitence class provided properties names.

Before Hibernate 3.x , hibernated was used only for retrieving[multiple records] purpose only. Hibernate 3.x onwards, hibernate is being used for insertion operations, deletion operations , updation operations and select operations.

In HQL , Retrieved records are available in the form of Collections. By default collection is **“Serializable”. So** Query result can be carried over the network.

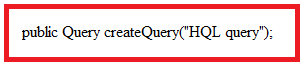
In hibernate application, The programmer defines HQL query. Hibernate will translate HQL query to underlying database dependent native SQL query.

**DisAdvantage Of HQL:-**

1. HQL is useful only for doing DML Operations. But It is not useful for doing DDL operations.
2. HQL is not supporting to invoking stored procedures and functions from the Hibernate Application.

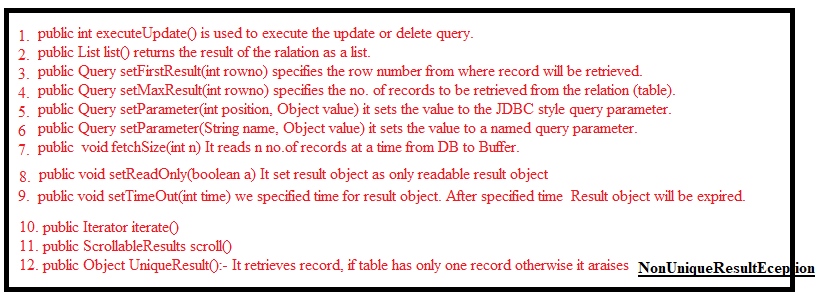
**2.HOW TO USE HQL IN HIBERNATE APPLICATION:**

a. Query is an interface.prepare HQL Query in the form of org.hibernate.Query object. The object of Query implementation class can be obtained by calling the createQuery() method Session interface.



b.Set values in custom property names over query.

The following are Query interface methods. They are used to set values in custom properties names.



1. Execute the HQL query using executeUpdate() method.

**3.Parameters:-**  we can provide the parameters to HQL query. There are two types of parameters in HQL. Parameter index starts from “0”.

**3.1.Positional parameters(?)**

The following method is used to value in positional parameter.

Query object.setParameter(index,value);

**3.2.Named Parameters(:parameterName)**

The following method is used to value in positional parameter.

Query object.setParameter(“name of parameter”,value);