

# Hibernate Query Language (HQL)

Hibernate Query Language (HQL) is same as SQL (Structured Query Language) but it doesn't depends on the table of the database. Instead of table name, we use class name in HQL. So it is database independent query language.

## 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.

## Example of HQL to get all the records

```
Query query=session.createQuery("from Emp");//here persistent class name is Emp  
List list=query.list();
```

## Example of HQL to get records with pagination

```
Query query=session.createQuery("from Emp");  
query.setFirstResult(5);  
query.setMaxResult(10);  
List list=query.list();//will return the records from 5 to 10th number
```

## Example of HQL update query

```
Transaction tx=session.beginTransaction();  
Query q=session.createQuery("update User set name=:n where id=:i");  
q.setParameter("n","Udit Kumar");  
q.setParameter("i",111);  
  
int status=q.executeUpdate();  
System.out.println(status);  
tx.commit();
```

## Example of HQL delete query

```
Query query=session.createQuery("delete from Emp where id=100");  
//specifying class name (Emp) not tablename  
query.executeUpdate();
```

## HQL with Aggregate functions

You may call avg(), min(), max() etc. aggregate functions by HQL. Let's see some common examples:

## Example to get total salary of all the employees

```
Query q=session.createQuery("select sum(salary) from Emp");  
List<Integer> list=q.list();  
System.out.println(list.get(0));
```

## Example to get maximum salary of employee

```
Query q=session.createQuery("select max(salary) from Emp");
```

## Example to get minimum salary of employee

```
Query q=session.createQuery("select min(salary) from Emp");
```

## Example to count total number of employee ID

```
Query q=session.createQuery("select count(id) from Emp");
```

## Example to get average salary of each employees

```
Query q=session.createQuery("select avg(salary) from Emp");
```

[< prev](#)[next >](#)

## Please Share



## Learn Latest Tutorials



On CentOS



Framework7



On Mac



Phalcon