HQL - Hibernate Query Language

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

Advantage of HQL

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

* database independent
* easy to learn for Java Programmer

TypedQuery Interface

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

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

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

Example of HQL to get all the records

* TypedQuery query=session.createQuery("from Emp");//here persistent class name is Emp
* List list=query.getResultList();

**SQL :- select \* from emp; -- emp is tablename**

**HQL :- from Employee; -- Employee is classname**

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");

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 retreiving all the records of table whose salary is greater than 50000 etc.

Examples of Hibernate Criteria Query Language

There are given a lot of examples of HCQL.

Example of HCQL to get all the records

* Crietria c=session.createCriteria(Emp.**class**);//passing Class class argument
* List list=c.list();

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

* Crietria 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

* Crietria 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

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