

[JAVA TUTORIAL](#)[#INDEX POSTS](#)[#INTERVIEW QUESTIONS](#)[RESOURCES](#)[HIRE ME](#)[DOWNLOAD ANDROID APP](#)[CONTRIBUTE](#)**Subscribe to Download Java Design Patterns eBook****DOWNLOAD NOW**[HOME](#) » [HIBERNATE](#) » [HQL – HIBERNATE QUERY LANGUAGE – EXAMPLE TUTORIAL](#)

HQL – Hibernate Query Language – Example Tutorial

APRIL 2, 2018 BY [PANKAJ](#) — [9 COMMENTS](#)

HQL or Hibernate Query Language is the object-oriented query language of Hibernate Framework. HQL is very similar to SQL except that we use Objects instead of table names, that makes it more close to object oriented programming.

Table of Contents [\[hide\]](#)

- [1 Hibernate Query Language – HQL](#)
 - [1.1 HQL Example Database Setup](#)
 - [1.2 Hibernate Maven Dependencies](#)
 - [1.3 Hibernate Configuration XML](#)
 - [1.4 Hibernate SessionFactory Utility class](#)
 - [1.5 Model Classes with Annotation based mapping](#)
 - [1.6 HQL Example Test Class](#)

Hibernate Query Language – HQL



HQL and Case Sensitivity: HQL is case-insensitive except for java class and variable names. So `SeLeCT` is the same as `sELeCT` is the same as `SELECT`, but `com.journaldev.model.Employee` is not same as `com.journaldev.model.EMPLOYEE`.

Some of the commonly supported clauses in HQL are:

1. **HQL From:** HQL From is same as select clause in SQL, `from Employee` is same as `select * from Employee`. We can also create alias such as `from Employee emp` or `from Employee as emp`.
2. **HQL Join :** HQL supports inner join, left outer join, right outer join and full join. For example, `select e.name, a.city from Employee e INNER JOIN e.address a`. In this query, Employee class should have a variable named address. We will look into it in the example code.
3. **Aggregate Functions:** HQL supports commonly used aggregate functions such as `count(*)`, `count(distinct x)`, `min()`, `max()`, `avg()` and `sum()`.
4. **Expressions:** HQL supports arithmetic expressions (+, -, *, /), binary comparison operators (=, >=, <=, <>, !=, like), logical operations (and, or, not) etc.
5. HQL also supports `ordr` by and `group by` clauses.
6. HQL also supports sub-queries just like SQL queries.
7. HQL supports DDL, DML and executing store procedures too.

Let's look at a simple example of using HQL in our program.

HQL Example Database Setup

I am using MySQL database for my example, below script will create two tables Employee and Address. They have one-to-one mapping and I am inserting some demo data for my example.

```
CREATE TABLE `Employee` (  
  `emp_id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `emp_name` varchar(20) NOT NULL,  
  `emp_salary` double(10,0) NOT NULL DEFAULT '0',
```

```

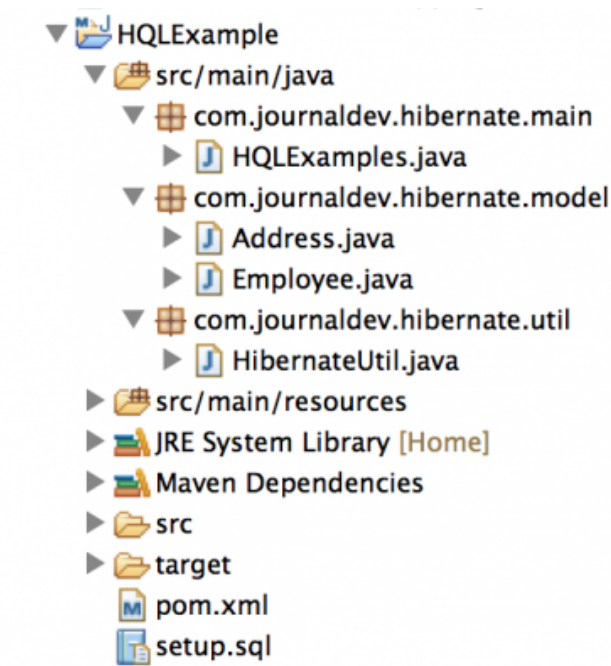
PRIMARY KEY (`emp_id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

CREATE TABLE `Address` (
  `emp_id` int(11) unsigned NOT NULL,
  `address_line1` varchar(50) NOT NULL DEFAULT '',
  `zipcode` varchar(10) DEFAULT NULL,
  `city` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`emp_id`),
  CONSTRAINT `emp_fk_1` FOREIGN KEY (`emp_id`) REFERENCES `Employee` (`emp_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `Employee` (`emp_id`, `emp_name`, `emp_salary`)
VALUES
  (1, 'Pankaj', 100),
  (2, 'David', 200),
  (3, 'Lisa', 300),

```

Create a maven project in Eclipse or the IDE you are using, our final project will look like below image.



Hibernate Maven Dependencies

Our final pom.xml contains dependencies for Hibernate and MySQL driver.

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

```

```

<modelVersion>4.0.0</modelVersion>
<groupId>com.journaldev.hibernate</groupId>
<artifactId>HQLExample</artifactId>
<version>0.0.1-SNAPSHOT</version>
<dependencies>
    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>4.3.5.Final</version>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>5.0.5</version>
    </dependency>
</dependencies>
</project>

```

Hibernate Configuration XML

Our hibernate configuration xml file contains database connection related properties and mapping classes. I will be using annotations for Hibernate mapping.

hibernate.cfg.xml code:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.password">pankaj123</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost/TestDB</property>
        <property name="hibernate.connection.username">pankaj</property>
        <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

```

```

<property name="hibernate.current_session_context_class">thread</property>
<property name="hibernate.show_sql">true</property>

<mapping class="com.journaldev.hibernate.model.Employee"/>
<mapping class="com.journaldev.hibernate.model.Address"/>
</session-factory>

```

Hibernate SessionFactory Utility class

We have a utility class to configure **hibernate SessionFactory**.

```

package com.journaldev.hibernate.util;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;

public class HibernateUtil {

    private static SessionFactory sessionFactory;

    private static SessionFactory buildSessionFactory() {
        try {
            // Create the SessionFactory from hibernate.cfg.xml
            Configuration configuration = new Configuration();
            configuration.configure("hibernate.cfg.xml");
            System.out.println("Hibernate Configuration loaded");

            ServiceRegistry serviceRegistry = new
StandardServiceRegistryBuilder().applySettings(configuration.getProperties()).build();

            System.out.println("Hibernate serviceRegistry created");

```

Model Classes with Annotation based mapping

Our model classes with JPA annotations looks like below.

```

package com.journaldev.hibernate.model;

import javax.persistence.Column;
import javax.persistence.Entity;

```

```

import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;

import org.hibernate.annotations.Cascade;

@Entity
@Table(name = "EMPLOYEE")
public class Employee {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "emp_id")
    private long id;

    // ...

package com.journaldev.hibernate.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.PrimaryKeyJoinColumn;
import javax.persistence.Table;

import org.hibernate.annotations.GenericGenerator;
import org.hibernate.annotations.Parameter;

@Entity
@Table(name = "ADDRESS")
public class Address {

    @Id
    @Column(name = "emp_id", unique = true, nullable = false)
    @GeneratedValue(generator = "gen")
    @GenericGenerator(name = "gen", strategy = "foreign",
        parameters = { @Parameter(name = "property", value =

```

HQL Example Test Class

Let's see how to use HQL in java programs.

```
package com.journaldev.hibernate.main;

import java.util.Arrays;
import java.util.List;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.journaldev.hibernate.model.Employee;
import com.journaldev.hibernate.util.HibernateUtil;

public class HQLExamples {

    @SuppressWarnings("unchecked")
    public static void main(String[] args) {

        //Prep work
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        Session session = sessionFactory.getCurrentSession();
```

Notice that I am using HQL for Select, Update and Delete operations. It also shows how to use HQL Join and HQL Aggregate functions.

When I run above hql example program, we get following output.

```
May 22, 2014 1:55:37 PM
org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {4.0.4.Final}
May 22, 2014 1:55:37 PM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate Core {4.3.5.Final}
May 22, 2014 1:55:37 PM org.hibernate.cfg.Environment <clinit>
INFO: HHH000206: hibernate.properties not found
May 22, 2014 1:55:37 PM org.hibernate.cfg.Environment buildBytecodeProvider
INFO: HHH000021: Bytecode provider name : javassist
May 22, 2014 1:55:37 PM org.hibernate.cfg.Configuration configure
INFO: HHH000043: Configuring from resource: hibernate.cfg.xml
May 22, 2014 1:55:37 PM org.hibernate.cfg.Configuration getConfigurationInputStream
INFO: HHH000040: Configuration resource: hibernate.cfg.xml
May 22, 2014 1:55:37 PM org.hibernate.cfg.Configuration doConfigure
INFO: HHH000041: Configured SessionFactory: null
```

```
Hibernate Configuration loaded
Hibernate serviceRegistry created
May 22, 2014 1:55:37 PM
org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl
configure
WARN: HHH000402: Using Hibernate built-in connection pool (not for production use!)
May 22, 2014 1:55:37 PM
```

Notice that once delete operation is performed, further operations are not showing that record data (sum of salary is 600). However I am rolling back the transaction, so the data in table will remain unchanged. Change the code to commit the transaction and it will be reflected in the database tables.

I don't like using HQL query a lot because as you can see that we need to take care of table mappings in our code. If we will use Session to delete the Employee object, it will delete the record from both the tables.

You can download the sample hql example project from below link and try more examples.

[Download Hibernate HQL Project](#)

« **PREVIOUS**

Hibernate Tools Eclipse Plugin

NEXT »

org.hibernate.HibernateException: Could not parse
configuration: hibernate.cfg.xml

About Pankaj

If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on [Google Plus](#), [Facebook](#) or [Twitter](#). I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on [Youtube](#).

FILED UNDER: [HIBERNATE](#)

Comments

hoang anh tien says

JULY 23, 2018 AT 8:50 AM

Thanks you verygood

[Reply](#)

Ananta says

FEBRUARY 12, 2018 AT 11:32 PM

Hey, what happens if we delete a row in the address table? It wouldn't affect the Employee table..right?

[Reply](#)

pavan kumar says

MARCH 6, 2017 AT 5:28 AM

can i know how to use mappings to store data by using form

[Reply](#)

Vijay Raj R says

FEBRUARY 17, 2017 AT 9:30 PM

For Pagination we must use setMaxResults(2) instead of setFetchSize(2).

[Reply](#)

Issak Madeena RahamaTullah shaik says

JANUARY 27, 2015 AT 11:01 PM

Hi

I have requirement i.e I need to join morethan two tables using criteria api, so please guide me in this regard

[Reply](#)**Shailesh says**

JUNE 30, 2014 AT 4:13 AM

Can you please add some tutorials about inheritance in hibernate.

Also what is 'discriminator' in hibernate??

It was asked in an interview.

[Reply](#)**Farhan says**

MAY 22, 2014 AT 3:43 AM

Hi Pankaj,

Your HQL example is good but I need more and more for example. all joins examples with one-to-many and many-to-many association for HQL. some example of HQL method that is used with collection properties.

HQL using Map, List, wrapper object in it select clause for example select new List(empId, empName, empSex) from employees etc other stuff like pagination and etc

[Reply](#)**Pankaj says**

MAY 22, 2014 AT 8:30 AM

Okay, I will add few more in the tutorial.

[Reply](#)**Pankaj says**

MAY 22, 2014 AT 1:57 PM

I have added some more examples of order by, group by, pagination, like etc.

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

☐

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

Search for tutorials...



DOWNLOAD ANDROID APP



HIBERNATE FRAMEWORK

Hibernate Tutorial

- > [Hibernate Example](#)
- > [Hibernate SessionFactory](#)
- > [Hibernate Session get load](#)
- > [Hibernate Session save](#)
- > [HQL Example](#)
- > [Hibernate Criteria](#)
- > [Hibernate SQL](#)
- > [Hibernate Named Query](#)
- > [Hibernate Log4J](#)
- > [Hibernate Validator](#)
- > [Hibernate Tomcat DataSource](#)

Hibernate Mapping

- > [Hibernate One to One Mapping](#)
- > [Hibernate One to Many Mapping](#)
- > [Hibernate Many to Many Join Tables](#)

Hibernate Caching

- > [Hibernate Cache](#)
- > [Hibernate EHCache](#)

Hibernate Integrations

- > [Hibernate Spring](#)
- > [Hibernate Spring MVC](#)
- > [Hibernate Struts 2](#)
- > [Hibernate Primefaces](#)
- > [Hibernate Primefaces Spring](#)
- > [Hibernate SpringRoo Primefaces](#)
- > [Hibernate JSF Spring](#)

Miscellaneous

- > [Hibernate Tools Eclipse Plugin](#)
- > [Hibernate Configuration Offline](#)
- > [\[Solved\] No identifier specified](#)
- > [Hibernate Program Not Terminating](#)
- > [Access to DialectResolutionInfo](#)
- > [get is not valid](#)
- > [No CurrentSessionContext configured](#)
- > [Hibernate Interview Questions](#)

RECOMMENDED TUTORIALS

Java Tutorials

- > [Java IO](#)
- > [Java Regular Expressions](#)
- > [Multithreading in Java](#)
- > [Java Logging](#)
- > [Java Annotations](#)
- > [Java XML](#)
- > [Collections in Java](#)
- > [Java Generics](#)
- > [Exception Handling in Java](#)
- > [Java Reflection](#)
- > [Java Design Patterns](#)
- > [JDBC Tutorial](#)

Java EE Tutorials

- > [Servlet JSP Tutorial](#)
- > [Struts2 Tutorial](#)
- > [Spring Tutorial](#)
- > [Hibernate Tutorial](#)
- > [Primefaces Tutorial](#)
- > [Apache Axis 2](#)
- > [JAX-RS](#)
- > [Memcached Tutorial](#)

Learn Java Programming

© 2018 · Privacy Policy · Don't copy, it's Bad Karma · P

Top Programming Jobs

**Computer Programming
Software**

Web Based Programming

Online Computer Programming

SQL Tutorial

SQL Language

Query Language

