

**+ Hibernate Tuts**

- [⊕ ORM là gì?](#)
- [⊕ Hibernate là gì?](#)
- [⊕ Hibernate - Kiến trúc](#)
- [⊕ Hibernate - Môi trường](#)
- [⊕ Hibernate - File cấu hình XML](#)
- [⊕ Hibernate - File Mapping](#)
- [⊕ Hibernate - Thuộc tính type](#)
- [⊕ Hibernate - Lớp persistent](#)
- [⊕ Hibernate - Sessions](#)
- [⊕ Hibernate - example](#)
- [⊕ Hibernate - O/R mapping](#)
- [⊕ Hibernate - Annotations](#)

**+ Hibernate - Query Language**

- [⊕ Hibernate - Criteria Queries](#)
- [⊕ Hibernate - Native SQL](#)
- [⊕ Hibernate - Bộ nhớ cache](#)
- [⊕ Hibernate - Batch processing](#)
- [⊕ Hibernate - Interceptors](#)

**+ Hibernate với các ví dụ**

- [⊕ Hibernate - many-to-one relationship với MySQL trên Eclipse](#)
- [⊕ Hibernate - one-to-one relationship với MySQL trên Eclipse](#)
- [⊕ Hibernate - one-to-many relationship với MySQL trên Eclipse](#)
- [⊕ Hibernate - many-to-many relationship với MySQL trên Eclipse](#)

**+ Hibernate Tool**

- [⊕ Cài đặt Hibernate/Jboss Tools trong Eclipse IDE](#)
- [⊕ Tạo Hibernate Configuration \(hibernate.cfg.xml\) với Hibernate Tools](#)
- [⊕ Sử dụng Hibernate Tools tạo các file mapping và annotation](#)

**+ Hibernate 5 Examples**

- [⊕ Ví dụ Hibernate 5 XML Mapping - MySQL](#)



## Hibernate – Query Language (HQL)

**⊖ Hibernate – Annotations****⊖ Hibernate – Criteria Queries**

Hibernate Query Language (HQL) là một ngôn ngữ truy vấn hướng đối tượng, tương tự như SQL nhưng thay vì làm việc trên các bảng và cột, HQL làm việc với các đối tượng persistent và các thuộc tính của chúng. Các truy vấn HQL được dịch bởi Hibernate thành các truy vấn SQL thông thường, lần lượt thực hiện các công việc trên cơ sở dữ liệu.

Mặc dù bạn có thể sử dụng các câu lệnh SQL trực tiếp với Hibernate bằng cách sử dụng [Native SQL](#). Nhưng chúng tôi khuyên bạn nên sử dụng HQL bất cứ khi nào có thể để tránh sự phức tạp về tính linh hoạt của cơ sở dữ liệu và tận dụng chiến lược generation và caching của Hibernate.

Các từ khoá như SELECT, FROM và WHERE vv không phân biệt chữ hoa và chữ thường nhưng các thuộc tính như tên bảng và cột có phân biệt chữ hoa chữ thường trong HQL.

**Nội dung chính:**

- [Mệnh đề FROM](#)
- [Mệnh đề AS](#)
- [Mệnh đề SELECT](#)
- [Mệnh đề WHERE](#)
- [Mệnh đề ORDER BY](#)
- [Mệnh đề GROUP BY](#)
- [Sử dụng các tham số đặt tên](#)
- [Mệnh đề UPDATE](#)
- [Mệnh đề DELETE](#)
- [Mệnh đề INSERT](#)
- [Các phương thức tổng hợp](#)
- [Phân trang bằng việc sử dụng Query](#)

### Mệnh đề FROM

Bạn sẽ sử dụng mệnh đề **FROM** nếu bạn muốn tải các đối tượng persistent vào bộ nhớ. Sau đây là cú pháp đơn giản sử dụng mệnh đề FROM:

```
String hql = "FROM Employee";
Query query = session.createQuery(hql);
List results = query.list();
```

### Recent Updates

- [⊕ Xử lý ngoại lệ trong Python](#)
- [⊕ File I/O trong Python](#)
- [⊕ Mô-đun trong Python](#)
- [⊕ Hàm trong Python](#)
- [⊕ Date và Time trong Python](#)
- [⊕ Miagi Solution tuyển dụng Java, .NET, Ruby, PHP,...làm việc tại Hà Nội](#)
- [⊕ Mitrais tuyển dụng 16 Java Web Developer làm việc tại Hà Nội](#)
- [⊕ Tuyển Dụng Công Nghệ Thông Tin](#)
- [⊕ List câu hỏi phỏng vấn Java Collection](#)
- [⊕ 201 câu hỏi phỏng vấn java core \(phần 1\)](#)
- [⊕ Dictionary trong Python](#)
- [⊕ Tuple trong Python](#)



26&27 JULY  
TRANSCORP HILTON, ABUJA

#TEForum2019

### VietTuts on facebook



1 người bạn thích nội dung này



- [Ví dụ Hibernate 5 XML  
Mapping – SQLServer 2017](#)
- [Ví dụ Hibernate 5 Annotation  
Mapping – MySQL](#)
- [Ví dụ Hibernate 5 Annotation  
Mapping – SQLServer 2017](#)
- + Hibernate 4 Examples**
- [Ví dụ Hibernate 4 XML  
Mapping – MySQL](#)
- [Ví dụ Hibernate 4 XML  
Mapping – SQLServer 2017](#)
- [Ví dụ Hibernate 4 Annotation  
Mapping – MySQL](#)
- [Ví dụ Hibernate 4 Annotation  
Mapping – SQLServer 2017](#)
- + Hibernate 3 Examples**
- [Ví dụ Hibernate 3 XML  
Mapping – MySQL](#)
- [Ví dụ Hibernate 3 XML  
Mapping – SQLServer 2017](#)
- [Ví dụ Hibernate 3 Annotation  
Mapping – MySQL](#)
- [Ví dụ Hibernate 3 Annotation  
Mapping – SQLServer 2017](#)
- + Phòng vấn Hibernate**
- [List câu hỏi phòng vấn  
Hibernate](#)



111.300 ₫



256.000 ₫



116.000 ₫



257.500 ₫

Nếu bạn cần sử dụng tên lớp đầy đủ trong HQL, chỉ cần chỉ định gói và tên lớp như sau:

```
String hql = "FROM com.hibernatebook.criteria.Employee";
Query query = session.createQuery(hql);
List results = query.list();
```

## Mệnh đề AS

Mệnh đề **AS** có thể được sử dụng để gán các bí danh cho các lớp trong các truy vấn HQL, đặc biệt khi bạn có các truy vấn dài. Ví dụ:

```
String hql = "FROM Employee AS E";
Query query = session.createQuery(hql);
List results = query.list();
```

Tùy khóa **AS** là tùy chọn và bạn cũng có thể chỉ định bí danh trực tiếp sau tên lớp, như sau:

```
String hql = "FROM Employee E";
Query query = session.createQuery(hql);
List results = query.list();
```

## Mệnh đề SELECT

Mệnh đề SELECT cung cấp kiểm soát nhiều hơn các tập kết quả hơn mệnh đề from. Nếu bạn muốn lấy ra những thuộc tính cần thiết của các đối tượng thay vì đối tượng hoàn chỉnh, hãy sử dụng mệnh đề SELECT. Sau đây là cú pháp đơn giản của việc sử dụng mệnh đề SELECT để chỉ lấy ra trường first\_name của đối tượng Employee:

```
String hql = "SELECT E.firstName FROM Employee E";
Query query = session.createQuery(hql);
List results = query.list();
```

**Chú ý:** `Employee.firstName` là thuộc tính của đối tượng Employee chứ không phải là một trường của bảng EMPLOYEE...

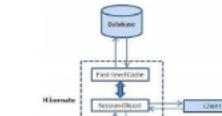
## Mệnh đề WHERE

Nếu bạn muốn thu hẹp các đối tượng cụ thể được trả về từ cơ sở dữ liệu, bạn sử dụng mệnh đề WHERE. Sau đây là cú pháp đơn giản sử dụng mệnh đề WHERE:

```
String hql = "FROM Employee E WHERE E.id = 10";
Query query = session.createQuery(hql);
List results = query.list();
```

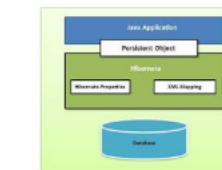
Best SQL Backup Software - To Azure, S3, FTP, SFTP, etc.

QC cloudbacko.com



## Bộ nhớ cache trong...

viettuts.vn



## Kiến trúc Hibernate

viettuts.vn



## Tuyển Dụng Công...

viettuts.vn

## Mệnh đề ORDER BY

Để sắp xếp kết quả truy vấn HQL của bạn, bạn sẽ cần sử dụng mệnh đề **ORDER BY**. Bạn có thể sắp xếp các kết quả theo bất kỳ thuộc tính nào trên các đối tượng trong tập kết quả hoặc tăng dần (ASC) hoặc giảm dần (DESC). Sau đây là cú pháp đơn giản của việc sử dụng mệnh đề ORDER BY:

```
String hql = "FROM Employee E WHERE E.id > 10 ORDER BY E.salary DESC";
Query query = session.createQuery(hql);
List results = query.list();
```

Nếu bạn muốn sắp xếp theo nhiều thuộc tính, bạn chỉ cần thêm các thuộc tính bổ sung vào cuối của mệnh đề ORDER BY, được phân cách bằng dấu phẩy như sau:

```
String hql = "FROM Employee E WHERE E.id > 10 " +
    "ORDER BY E.firstName DESC, E.salary DESC ";
Query query = session.createQuery(hql);
List results = query.list();
```

## Mệnh đề GROUP BY

Mệnh đề này cho phép Hibernate lấy thông tin từ cơ sở dữ liệu và phân nhóm chúng dựa trên giá trị của một thuộc tính. Sau đây là cú pháp đơn giản của việc sử dụng mệnh đề GROUP BY:

```
String hql = "SELECT SUM(E.salary), E.firstName FROM Employee E " +
    "GROUP BY E.firstName";
Query query = session.createQuery(hql);
List results = query.list();
```

## Sử dụng các tham số đặt tên

Hibernate hỗ trợ các tham số được đặt tên trong các truy vấn HQL của nó. Điều này làm cho việc viết các truy vấn HQL chấp nhận đầu vào từ người dùng dễ dàng và bạn không phải phòng chống các cuộc tấn công SQL injection. Sau đây là cú pháp đơn giản sử dụng các tham số được đặt tên:

```
String hql = "FROM Employee E WHERE E.id = :employee_id";
Query query = session.createQuery(hql);
query.setParameter("employee_id", 10);
List results = query.list();
```

## Mệnh đề UPDATE

Cập nhật hàng loạt là kiểu mới đổi với HQL trong Hibernate 3. Và xóa khác nhau trong Hibernate 3 so với Hibernate 2. Giao tiếp Query giờ đây chứa một phương thức gọi là `executeUpdate()` để thực hiện các câu lệnh HQL UPDATE hoặc DELETE.

Mệnh đề **UPDATE** có thể được sử dụng để cập nhật một hoặc nhiều thuộc tính của một hoặc nhiều đối tượng. Sau đây là cú pháp đơn giản sử dụng mệnh đề UPDATE:

```
String hql = "UPDATE Employee set salary = :salary " +  
           "WHERE id = :employee_id";  
Query query = session.createQuery(hql);  
query.setParameter("salary", 1000);  
query.setParameter("employee_id", 10);  
int result = query.executeUpdate();  
System.out.println("Rows affected: " + result);
```

## Mệnh đề DELETE

Mệnh đề **DELETE** được sử dụng để xóa một hoặc nhiều đối tượng. Sau đây là cú pháp đơn giản sử dụng mệnh đề DELETE:

```
String hql = "DELETE FROM Employee " +  
           "WHERE id = :employee_id";  
Query query = session.createQuery(hql);  
query.setParameter("employee_id", 10);  
int result = query.executeUpdate();  
System.out.println("Rows affected: " + result);
```

## Mệnh đề INSERT

HQL chỉ hỗ trợ mệnh đề INSERT INTO nơi mà các bản ghi có thể được insert từ một đối tượng này sang một đối tượng khác. Sau đây là cú pháp đơn giản sử dụng mệnh đề INSERT INTO:

```
String hql = "INSERT INTO Employee(firstName, lastName, salary)" +  
           "SELECT firstName, lastName, salary FROM old_employee";  
Query query = session.createQuery(hql);  
int result = query.executeUpdate();  
System.out.println("Rows affected: " + result);
```

## Các phương thức tổng hợp

HQL hỗ trợ một loạt các phương thức tổng hợp tương tự như SQL. Chúng hoạt động theo cùng một cách trong HQL như trong SQL và sau đây là danh sách các chức năng có sẵn:

No.	Các hàm	Mô tả
1	avg(property name)	Trung bình của các giá trị của một thuộc tính.
2	count(property name or *)	Số lần một thuộc tính xảy ra trong kết quả.
3	max(property name)	Giá trị lớn nhất trong các giá trị của một trường.
4	min(property name)	The minimum value of the property values
5	sum(property name)	Tính tổng các giá trị của một trường

Từ khóa **distinct** chỉ tính các giá trị duy nhất trong tập hàng. Truy vấn sau đây sẽ chỉ trả về số lần duy nhất:

```
String hql = "SELECT count(distinct E.firstName) FROM Employee E";
Query query = session.createQuery(hql);
List results = query.list();
```

## Phân trang bằng việc sử dụng Query

Có hai phương thức của giao tiếp Query cho việc phân trang.

No.	Phương thức & Mô tả
1	<b>Query setFirstResult(int startPosition)</b> Phương thức này lấy một số nguyên đại diện cho hàng đầu tiên trong tập kết quả, bắt đầu với hàng 0.
2	<b>Query setMaxResults(int maxResult)</b> Phương thức này cho Hibernate tìm kiếm một số <b>maxResults</b> của các đối tượng.

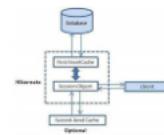
Sử dụng hai phương thức trên, chúng ta có thể xây dựng một thành phần phân trang trong ứng dụng Web hoặc Swing. Sau đây là ví dụ bạn có thể mở rộng để lấy 10 hàng cùng một lúc:

```
String hql = "FROM Employee";
Query query = session.createQuery(hql);
query.setFirstResult(1);
query.setMaxResults(10);
List results = query.list();
```

Bài tiếp theo: [Hibernate – Criteria Queries](#)

## ⊕ Hibernate – Annotations

Hibernate – Criteria Queries



## SQL Server Cloud /Local Bộ nhớ cache trong Backup - Azure, SFTP, S Hibernate

## Đào tạo Nhân sự lập trình Java - Kỹ năng, Chuyên môn trên Dự Án

QC cloudbacko.com

viettuts.vn

Bạn cần 1 website để kinh doanh



QC imic.edu.vn

viettuts.vn





Tạo Website Chuyên  
Nghiệp

Tuyển Dụng Công Nghệ  
Thông Tin

Ví dụ về Hibernate  
Mapping - MySQL

QC Haravan

viettuts.vn

viettuts.vn

viettuts.vn

0 bình luận

Sắp xếp theo Cũ nhất



Thêm bình luận...

Plugin bình luận trên Facebook

## Danh sách bài học

[Học java](#)  
[Học servlet](#)  
[Học jsp](#)  
[Học Hibernate](#)  
[Học Struts2](#)  
[Học Spring](#)  
[Học SQL](#)

## Câu hỏi phỏng vấn

[201 câu hỏi phỏng vấn java](#)  
[26 câu hỏi phỏng vấn servlet](#)  
[75 câu hỏi phỏng vấn jsp](#)  
[52 câu hỏi phỏng vấn Hibernate](#)  
[70 câu hỏi phỏng vấn Spring](#)  
[57 câu hỏi phỏng vấn SQL](#)

## About VietTuts.Vn

Hệ thống bài học trên VietTuts.Vn bao gồm các bài lý thuyết và thực hành về các công nghệ java và công nghệ web. Các bài lý thuyết trên hệ thống VietTuts.Vn được tham khảo và tổng hợp từ các trang <http://javatpoint.com>, <http://www.tutorialspoint.com>, <http://docs.oracle.com/en> ...