

BỘ MÔN CNPM - KHOA CNTT - ĐH KHTN

J2EE PROGRAMMING TRAINING COURSE 2012 - 2.0.1

HIBERNATE QUERY LANGUAGE

cuu duong than cong . com

Nguyễn Hoàng Anh
nhanh@fit.hcmus.edu.vn

cuu duong than cong . com

Nội dung trình bày

- Hibernate Query Language (HQL)
- HQL – from
- HQL – select
- HQL – aggregate function
- HQL – where
- HQL – Expression
- HQL – order by
- HQL – group by & having
- HQL – sub query

JDBC - SQL

- JDBC sử dụng các câu lệnh SQL để thực hiện truy vấn dữ liệu để truy vấn dữ liệu và các thao tác cập nhật như thêm, xóa, sửa trên bảng dữ liệu.
- Để thao tác tốt các câu lệnh SQL cần quan tâm đến các bảng, các dòng, các cột và mối quan hệ giữa các bảng và đặc biệt là hệ quản trị cơ sở dữ liệu đang làm việc.
- Kết quả trả về của câu lệnh truy vấn là danh sách các dòng dữ liệu.

Hibernate - HQL

- **Hibernate** cung cấp các **API** cho phép thực hiện tác thao tác cập nhật như **thêm, xóa, sửa**.
- **Hibernate** cung cấp ngôn ngữ truy vấn rất mạnh được gọi là **Hibernate Query Language (HQL)**.
- **HQL độc lập hệ quản trị cơ sở dữ liệu** và được **Hibernate** thông dịch sang **SQL** tương ứng trong quá trình thực thi.
- **HQL là ngôn ngữ truy vấn theo hướng đối tượng**. Kết quả truy vấn là đối tượng
- **Hibernate sử dụng các lớp đối tượng và các thuộc tính** thay cho các **bảng và các cột**.

HQL – Phân biệt hoa thường

- HQL không phân biệt thường hoa ngoại trừ
 - Tên các lớp đối tượng
 - Các thuộc tính trong lớp đối tượng
- Ví dụ 2 câu truy vấn giống nhau
 - **select** s **from** Sach s <-> **SELECT** s **FROM** Sach s
 - **Select** s **From** Sach s <-> **SELECT** s **FROM** Sach s
- Ví dụ 2 câu truy vấn khác nhau
 - **select** s **from** **sach** s <-> **SELECT** s **FROM** **Sach** s
 - **Select** s **From** **SACH** s <-> **SELECT** s **FROM** **Sach** s

HQL – Mệnh đề **from**

- Lấy tất cả cá đối tượng danh mục
 - **from** DanhMuc
 - **select** dm **from** DanhMuc dm
 - **select** dm **from** DanhMuc as dm
- Lấy tất cả đối tượng sách
 - **from** Sach
 - **select** s **from** Sach s
 - **select** s **from** Sach as s

HQL – Mệnh đề from – lấy tất cả đối tượng

```
1 String hql = "from Sach";  
2 Query query = session.createQuery(hql);  
3 List<Sach> ds=query.list();
```

Name	Type	Value
ds	ArrayList	size = 18
[0]	Sach	#1514
maSach	String	"S001"
danhMuc	DanhMuc...	#1516
tenSach	String	"Java 1"
tacGia	String	"Nguyễn Văn Tùng"
giaBan	double	100000.55
soLuong	int	200
hinhAnh	String	"images/1.jpg"
[1]	Sach	#1526
[2]	Sach	#1527

HQL – Mệnh đề **from** – phân trang

```
1 String hql = "from Sach";
2 Query query = session.createQuery(hql);
3 query.setFirstResult(3);
4 query.setMaxResults(5);
5 List<Sach> ds=query.list();
```

- Lấy từ vị trí thứ n (tính từ 0)
setFirstResult (int n)
- lấy tối đa m đối tượng
setMaxResults (int m)
- Tương tự trong MYSQL
LIMIT N, M

Name	Type	Value
ds	ArrayList	"size = 5"
[0]	Sach	#1506
[1]	Sach	#1518
[2]	Sach	#1519
[3]	Sach	#1520
[4]	Sach	#1521

HQL - join

- cross join
- inner join hoặc join
- left join hoặc left outer join
- right join hoặc right outer join
- left join fetch hoặc left outer join fetch
- right join fetch hoặc right outer join fetch

cuduongthanhcong.com

HQL – cross join

- Thực hiện tích cartesian (còn gọi là cross join)
 - from Sach, DanhMuc
 - from Sach s, DanhMuc dm
 - select s, dm from Sach s, DanhMuc dm
- Kết quả trả về là một **ArrayList<Object []>**
- **Số lượng phần tử = số lượng danh mục * số lượng sách**

HQL – cross join

```
1 String hql = "from Sach s, DanhMuc dm";
2 Query query = session.createQuery(hql);
3 ArrayList<Object[]> ds = query.list();
4 for(int i=0; i<ds.size(); i++){
5     Object[] objs=ds.get(i);
6     Sach s=(Sach)objs[0];
7     DanhMuc dm=(DanhMuc)objs[1];
8 }
```

ds	ArrayList	"size = 90"
[0]	Object[]	#1514(length=2)
[0]	Sach	#1616
[1]	DanhMuc	#1617
[1]	Object[]	#1527(length=2)
[0]	Sach	#1616
[1]	DanhMuc	#1618
[2]	Object[]	#1528(length=2)
[3]	Object[]	#1529(length=2)
[4]	Object[]	#1530(length=2)

HQL – inner join

- Số lượng danh mục: 4
- Số lượng sách: 10 (đều thuộc danh mục)
- inner join

– from DanhMuc dm

inner join dm.sachs

- Kết quả

– Số lượng: 10

Name	Type	Value
query.list()	ArrayList	"size = 10"
[0]	Object[]	#1968(length=2)
[0]	Danhmuc	#2003
[1]	Sach	#2004
[1]	Object[]	#1969(length=2)
[2]	Object[]	#1970(length=2)
[3]	Object[]	#1971(length=2)
[4]	Object[]	#1972(length=2)
[5]	Object[]	#1973(length=2)
[6]	Object[]	#1974(length=2)
[7]	Object[]	#1975(length=2)
[8]	Object[]	#1976(length=2)
[9]	Object[]	#1977(length=2)

HQL – left join

- Số lượng danh mục: 4 (2 danh mục không có sách)
- Số lượng sách: 10 (đều thuộc danh mục)
- Left join

– from DanhMuc dm

left join dm.sachs

- Kết quả

– Số lượng: 12

– 2 danh mục không có
sách nào thuộc về

Name	Type	Value
+ [4]	Object[]	#1974(length=2)
+ [5]	Object[]	#1975(length=2)
+ [6]	Object[]	#1976(length=2)
+ [7]	Object[]	#1977(length=2)
+ [8]	Object[]	#1978(length=2)
- [9]	Object[]	#1979(length=2)
+ [0]	Danhmuc	#1989
+ [1]	Sach	#1990
- [10]	Object[]	#1980(length=2)
+ [0]	Danhmuc	#1988
+ [1]		null
- [11]	Object[]	#1981(length=2)
+ [0]	Danhmuc	#1987
+ [1]		null

HQL – right join

- Số lượng danh mục: 4
- Số lượng sách: 10 (đều thuộc danh mục)
- Right join
 - from DanhMuc dm
 - right join dm.sachs
- Kết quả
 - Số lượng: 10

HQL – left join fetch

- Số lượng danh mục: 4 (có)
- Số lượng sách: 10 (đều thuộc danh mục)
- Left join fetch

from Sach s left join fetch

s.danhMuc

- Kết quả

– Số lượng: 10

Name	Type	Value
query.list()	ArrayList	"size = 10"
[0]	Sach	#1948
[1]	Sach	#1949
[2]	Sach	#1950
[3]	Sach	#1951
[4]	Sach	#1952
[5]	Sach	#1953
[6]	Sach	#1954
[7]	Sach	#1955
[8]	Sach	#1956
[9]	Sach	#1957

HQL – right join fetch

- Số lượng danh mục: 4 (có)
- Số lượng sách: 10 (đều thuộc danh mục)
- right join fetch

from Sach s right join fetch

s.danhMuc

- Kết quả

– Số lượng: 10

Name	Type	Value
query.list()	ArrayList	"size = 10"
[0]	Sach	#1948
[1]	Sach	#1949
[2]	Sach	#1950
[3]	Sach	#1951
[4]	Sach	#1952
[5]	Sach	#1953
[6]	Sach	#1954
[7]	Sach	#1955
[8]	Sach	#1956
[9]	Sach	#1957

HQL - select

```
1 String hql = "select s.danhMuc from Sach s";  
2 Query query = session.createQuery(hql);  
3 List<DanhMuc> ds=query.list();
```

Name	Type	Value
ds	ArrayList	size = 18
[0]	DanhMuc	#1515
maDanhMuc	String	"DM001"
tenDanhMuc	String	"Java"
[1]	DanhMuc	#1515
[2]	DanhMuc	#1515
[3]	DanhMuc	#1515
[4]	DanhMuc	#1515
[5]	DanhMuc	#1516
[6]	DanhMuc	#1516
[7]	DanhMuc	#1516

HQL - select

```
1 String hql = "select s.danhMuc.tenDanhMuc from Sach s";  
2 Query query = session.createQuery(hql);  
3 List<String> ds=query.list();
```

Name	Type	Value
ds	ArrayList	size = 18
[0]	String	"Java"
[1]	String	"Java"
[2]	String	"Java"
[3]	String	"Java"
[4]	String	"Java"
[5]	String	"C#"
[6]	String	"C#"
[7]	String	"C#"

HQL - select

```
1 String h ="select s.maSach, s.tenSach, s.danhMuc from Sach s";
2 Query query = session.createQuery(h);
3 List<Object[]> ds=query.list();
4 for(int i=0; i<ds.size(); i++){
5     Object[] objs=ds.get(i);
6     String maSach=(String)objs[0];
7     String tenSach=(String)objs[1];
8     DanhMuc dm=(DanhMuc)objs[2];
9 }
```

Name	Type	Value
ds	ArrayList	size = 18
[0]	Object[]	#1510(length=3)
[0]	String	"S001"
[1]	String	"Java 1"
[2]	DanhMuc	#1541
[1]	Object[]	#1522(length=3)
[2]	Object[]	#1523(length=3)

HQL - select

```
1 String hql ="select distinct s.danhMuc.tenDanhMuc from
2 Sach s";
3 Query query = session.createQuery(hql);
4 List<String> ds=query.list();
```

cuu duong than cong . com

Name	Type	Value
ds	ArrayList	"size = 3"
[0]	String	"Java"
[1]	String	"C#"
[2]	String	"Java2"

cuu duong than cong . com

HQL - select

```
1 package pojo;
2
3 public class MyClass {
4     private String maSach;
5     private String tenSach;
6     private DanhMuc danhMuc;
7
8     public MyClass() {
9     }
10
11     public MyClass(String maSach, String tenSach,
12                     DanhMuc danhMuc) {
13         this.maSach = maSach;
14         this.tenSach = tenSach;
15         this.danhMuc = danhMuc;
16     }
17     //Getters & Setters
18 }
```

HQL - select

```
1 String hql =
2     "select new pojo.MyClass (s.maSach, s.tenSach, s.danhMuc)
3     from Sach s";
4 Query query = session.createQuery(hql);
5 List<MyClass> ds=query.list();
6 for(int i=0; i<ds.size(); i++){
7     MyClass my=ds.get(i);
8     String maSach=my.getMaSach();
9     String tenSach=my.getTenSach();
10    DanhMuc dm=my.getDanhMuc();
11 }
```

Name	Type	Value
ds2	List<MyClass>	"size = 18"
[0]	MyClass	#1521
maSach	String	"S001"
tenSach	String	"Java 1"
danhMuc	DanhMuc\$\$EnhancerByCGLIB\$\$b53f...	#1541
[1]	MyClass	#1522
[2]	MyClass	#1523
[3]	MyClass	#1524

HQL - Aggregate functions

- avg
- min
- max
- count
- sum

cuu duong than cong . com

cuu duong than cong . com

HQL - Aggregate functions

```
1 String hql = " select avg(s.giaBan) as GiaTrungBinh,"
2             + " count(*) as SoLuongDauSach,"
3             + " min(s.giaBan) as GiaBanThapNhat,"
4             + " max(s.giaBan) as GiaBanCaoNhat, "
5             + " sum(s.soLuong) as TongSoLuongSach "
6             + " from Sach s";
7 Query query = session.createQuery(hql);
8 Object[] objs = (Object[]) query.uniqueResult();
9 double giaTrungBinh=(Double)objs[0];
10 long soLuongDauSach=(Long)objs[1];
11 double giaBanThapNhat=(Double)objs[2];
12 double giaBanCaoNhat=(Double)objs[3];
13 long tongSoLuongSach=(Long)objs[4];
```

Name	Type	Value
objs	Object[]	#1487(length=5)
objs[0]	Double	#1493
value	double	204111.141666667
objs[1]	Long	#1494
objs[2]	Double	#1495
objs[3]	Double	#1496
objs[4]	Long	#1497

HQL - where

```
1 String tenSach="Java";
2 String hql = "from Sach s where s.tenSach like :tenSach";
3 Query query = session.createQuery(hql);
4 query.setString("tenSach", "%"+tenSach+"%");
5 ArrayList<Sach> ds=query.list();
```

```
1 String tenDanhMuc="Java";
2 String giaBan=20000;
3 String hql="from Sach s
4           where s.danhMuc.tenDanhMuc=:tenDanhMuc
5                  and s.giaBan>:giaBan"
6 Query query = session.createQuery(hql);
7 query.setString("tenDanhMuc", tenDanhMuc);
8 query.setString("giaBan", giaBan);
9 ArrayList<Sach> ds=query.list();
```

HQL - Expression

- +, -, *, /
- =, >=, <=, <>, !=, like
- and, or, not
- ()
- in, not in, between, is null, is not null, is empty, is not empty, member of, not member of
- case ... when ... then ... else ... end

HQL - Expression

- `concat(..., ...)`
- `current_date()`, `current_time()`,
`current_timestamp()`
- `second(...)`, `minute(...)`, `hour(...)`, `day(...)`,
`month(...)`, `year(...)`
- `substring()`, `trim()`, `lower()`, `upper()`,
`length()`, `locate()`, `abs()`, `sqrt()`,
`bit_length()`, `mod()`
- `coalesce()` , `nullif()`
- `str()`, ...

HQL – order by

```
1 String tenSach="Java";
2
3 String hql = " from Sach s";
4 hql=hql      +" where s.tenSach like :tenSach";
5 hql=hql      +" order by s.tenSach desc";
6
7 Query query = session.createQuery(hql);
8 query.setString("tenSach", "%"+tenSach+"%");
9 ArrayList<Sach> ds = query.list();
```

cuu duong than cong . com

HQL – group by & having

```
1 String hql = " select s.danhMuc, sum(s.soLuong)";
2 hql = hql + " from Sach s";
3 hql = hql + " group by s.danhMuc";
4 hql = hql + " having sum(s.soLuong)>100";
5
6 Query query = session.createQuery(hql);
7 ArrayList<Object[]> ds = query.list();
```

cuu duong than cong . com

HQL – subquery

```
1 String hql = " select s";
2 hql = hql + " from Sach s";
3 hql = hql + " where s.giaBan > ";
4 hql = hql + "      (";
5 hql = hql + "          select avg(s.giaBan)";
6 hql = hql + "          from Sach s";
7 hql = hql + "      )";
8 Query query = session.createQuery(hql);
9 ArrayList<Sach> ds = query.list();
```

cuu duong than cong . com

Tài liệu tham khảo

- Nguyễn Hoàng Anh, Tập bài giảng môn chuyên đề Java, 2011
- <http://docs.jboss.org/hibernate/core/3.5/reference/en/html/queryhql.html>

cuu duong than cong . com

cuu duong than cong . com



cuu duong than cong . com

HỎI VÀ ĐÁP