# BỘ MÔN CNPM - KHOA CNTT - ĐH KHTN J2EE PROGRAMMING TRAINING COURSE 2012 - 2.0.1

# HIBERNATE QUERY LANGUAGE

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

# 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

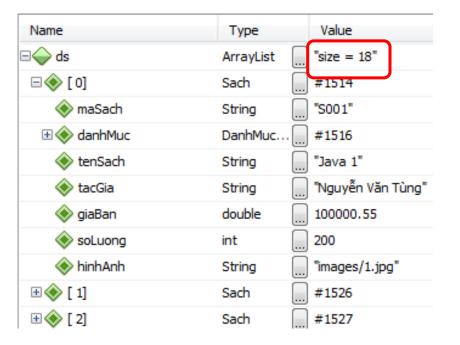
- HQL không phân biệt thường hoa ngoại trừ
  - Tên các lớp đối tượng
  - Các thuộc tính trong lớp đối tượng
- Ví dụ 2 câu truy vấn giống nhau
  - select s from Sach s <-> SELECT s FROM Sach s
  - Select s From Sach s <-> SELECT s FROM Sach s
- Ví dụ 2 câu truy vấn khác nhau
  - select s from sach s 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

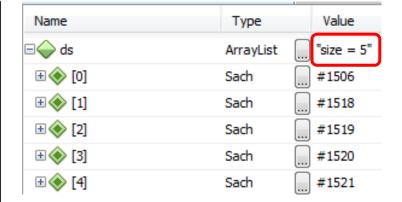
```
String hql = "from Sach";
Query query = session.createQuery(hql);
List<Sach> ds=query.list();
```



# HQL - Mệnh đề from - phân trang

```
String hql = "from Sach";
Query query = session.createQuery(hql);
query.setFirstResult(3);
query.setMaxResults(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



# 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

# 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

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

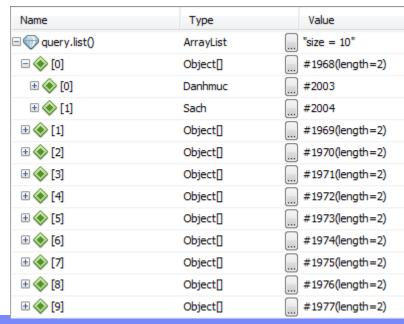
⊡� ds	ArrayList	"size = 90"
□ <b>◆</b> [0]	Object[]	#1514(length=2)
⊕♦ [0]	Sach	<u>#1616</u>
<b>⊕ (</b> 1]	DanhMuc	#1617
⊡� [1]	Object[]	#1527(length=2)
⊕ 🍲 [0]	Sach	#1616
⊕ 🏈 [1]	DanhMuc	#1618
⊕ 🍑 [2]	Object[]	#1528(length=2)
⊕ 🏈 [3]	Object[]	#1529(length=2)
⊕ 🏈 [4]	Object[]	#1530(length=2)

# HQL - inner join

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

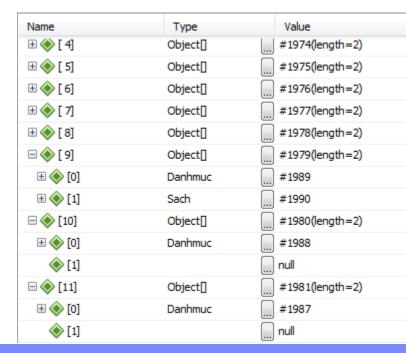
inner join dm.sachs

- Kết quả
  - -Số lượng: 10



# 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



# 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

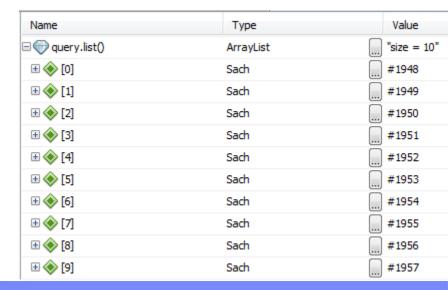
# HQL - left join fetch

- 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



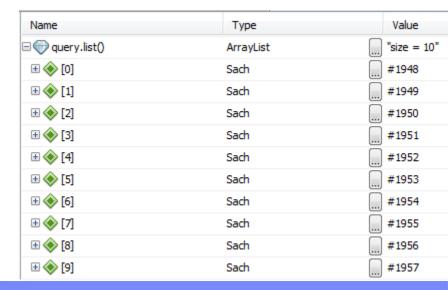
# HQL - right join fetch

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

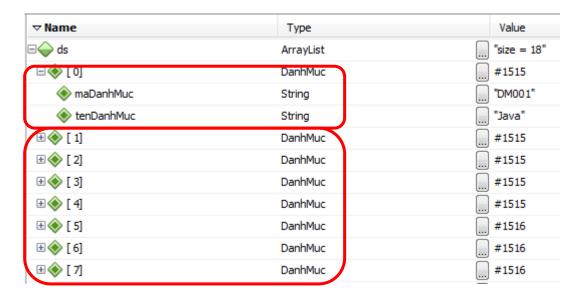
from Sach s right join fetch

s.danhMuc

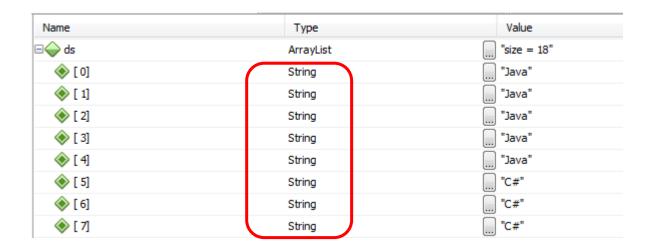
- Kết quả
  - -Số lượng: 10



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



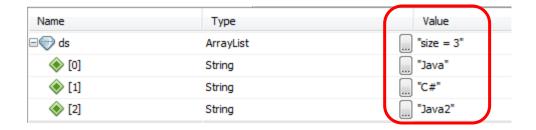
```
String hql = "select s.danhMuc.tenDanhMuc from Sach s";
Query query = session.createQuery(hql);
List<String> ds=query.list();
```



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

Name	Туре	Value
∃ <b>⇔</b> ds	ArrayList	"size = 18"
⊡� [0]	Object[]	#1510(length=3)
(o)	String	"S001"
[1]	String	"Java 1"
± <b>♦</b> [2]	DanhMuc	#1541
<b>⊕</b> 🍲 [ 1]	Object[]	#1522(length=3)
± <b>♦</b> [2]	Object[]	#1523(length=3)

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



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

```
String hql =
 1
           "select new pojo.MyClass (s.maSach, s.tenSach, s.danhMuc)
 3
            from Sach s";
     Query query = session.createQuery(hql);
 4
 5
     List<MyClass> ds=query.list();
     for(int i=0; i<ds.size(); i++){</pre>
 6
 7
          MyClass my=ds.get(i);
          String maSach=my.getMaSach();
 8
           String tenSach=my.getTenSach();
          DanhMuc dm=my.getDanhMuc();
                                       Value
Name
                 Type
∃ 🧼 ds2
                 List<MyClass>
                                       "size = 18"
⊡ ◈ [0]
                 MyClass
                                       #1521
                                       "S001"
    maSach
                 String
                                       "Java 1"
                 String
 DanhMuc$$EnhancerByCGLIB$$b53f...
                                       #1541
⊕ 🌑 [1]
                 MyClass
                                       #1522
⊕ ◆ [2]
                 MyClass
                                       #1523
⊕ ◆ [3]
                 MyClass
                                       #1524
```

# **HQL** - Aggregate functions

- avg
- min
- max
- count
- sum

#### **HQL** - Aggregate functions

```
" select avg(s.giaBan) as GiaTrungBinh,"
     String hql =
1
                            + " count(*) as SoLuongDauSach,"
                            + " min(s.giaBan) as GiaBanThapNhat,"
                            + " max(s.giaBan) as GiaBanCaoNhat, "
4
                            + " sum(s.soLuong) as TongSoLuongSach
6
                            + " from Sach s";
7
     Query query = session.createQuery(hql);
     Object[] objs = (Object[]) query.uniqueResult();
     double giaTrungBinh=(Double)objs[0];
10
     long soLuongDauSach=(Long)objs[1];
     double giaBanThapNhat=(Double)objs[2];
11
                                                      Name
                                                               Type
12
     double giaBanCaoNhat=(Double)objs[3];
                                                     ⊟ 🍛 obis
                                                                    #1487(length=5)
                                                              Object[]
13
     long tongSoLuongSach=(Long)objs[4];
                                                      □ ◆ [0]
                                                              Double
                                                                     #1493
                                                        value
                                                              double
                                                                    204111.141666667
                                                      ⊕ 🌑 [1]
                                                                    #1494
                                                              Lona
                                                      ⊕ ♠ [2]
                                                              Double
                                                                    #1495

⊕ 
♠ [3]

                                                              Double
                                                                    #1496

⊕ ◆ [4]

                                                                    #1497
                                                              Long
```

#### HQL - where

```
String tenSach="Java";
String hql = "from Sach s where s.tenSach like :tenSach";
Query query = session.createQuery(hql);
query.setString("tenSach", "%"+tenSach+"%");
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

```
String tenSach="Java";

String hql = " from Sach s";

hql=hql +" where s.tenSach like :tenSach";

hql=hql +" order by s.tenSach desc";

Query query = session.createQuery(hql);
query.setString("tenSach", "%"+tenSach+"%");
ArrayList<Sach> ds = query.list();
```

## HQL - group by & having

```
String hql = " select s.danhMuc, sum(s.soLuong)";
hql = hql + " from Sach s";
hql = hql + " group by s.danhMuc";
hql = hql + " having sum(s.soLuong)>100";

Query query = session.createQuery(hql);
ArrayList<Object[]> ds = query.list();
```

#### 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();
```

# Tài liệu tham khảo

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



# HỞI VÀ ĐÁP