



MAPPING MANY TO MANY

GVHD: Thầy Nguyễn Hoàng Anh

Sinh viên: 0812463 – Hồ Văn Tấn
0812505 – Sử Bá Thuận

Các vấn đề trình bày

1

- **Many to many**

2

- **Lazy Initialization**

3

- **Fetch**

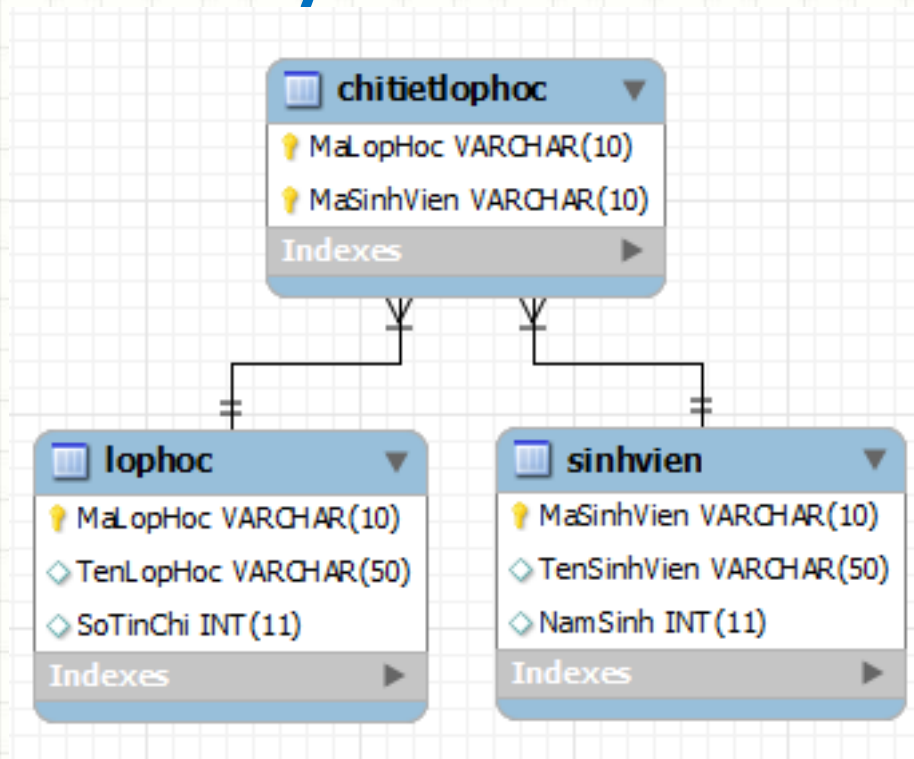
4

- **Cascade**



GIỚI THIỆU

Many to many



- Quan hệ Many – Many:
 - Một **sinhvien** có thể học nhiều **lophoc**.
 - Một **lophoc** có nhiều **sinhvien**.

LopHocPOJO & SinhVienPOJO

```
1 public class LopHocPOJO implements java.io.Serializable {  
2     private String maLopHoc;  
3     private String tenLopHoc;  
4     private int soTinChi;  
5     private Set<SinhVienPOJO> dsSinhVien = new HashSet<SinhVienPOJO>();  
6     //Các phương thức set, get, constructor  
}
```

```
1 public class SinhVienPOJO implements java.io.Serializable {  
2     private String maSinhVien;  
3     private String tenSinhVien;  
4     private int namSinh;  
5     private Set<LopHocPOJO> dsLopHoc = new HashSet<LopHocPOJO>();  
6     //Các phương thức set, get, constructor  
}
```

Mapping: SinhVien.hbm.xml

```
1 <hibernate-mapping>
2   <class name="pojo.SinhVienPOJO" table="sinhvien">
3     <id name="maSinhVien" column="MaSinhVien" type="string" length="10"/>
4     <property name="tenSinhVien" type="string" column="TenSinhVien" length="50" />
5     <property name="namSinh" type="integer" column="NamSinh" />
6     <set name="dsLopHoc" table="chitietlophoc">
7       <key column="MaSinhVien"/>
8       <many-to-many column="MaLopHoc" class="pojo.LopHocPOJO"></many-to-many>
9     </set>
10   </class>
11 </hibernate-mapping>
```


Mapping: LopHoc.hbm.xml

```
1 <hibernate-mapping>
2   <class name="pojo.LopHocPOJO" table="lophoc">
3     <id name="maLopHoc" type="string" column="MaLopHoc" length="10" />
4     <property name="tenLopHoc" type="string" column="TenLopHoc" length="50" />
5     <property name="soTinChi" type="integer" column="SoTinChi" />
6     <set name="dsSinhVien" table="chitietlophoc">
7       <key column="MaLopHoc"/>
8       <many-to-many column="MaSinhVien" class="pojo.SinhVienPOJO"></many-to-many>
9     </set>
10   </class>
11 </hibernate-mapping>
```

SinhVienDAO

```
1 public static List<SinhVienPOJO> layDsSinhVien() {  
2     List<SinhVienPOJO> list = null;  
3     SessionFactory sf = MyHibernateUtil.getSessionFactory();  
4     Session ss = sf.getCurrentSession();  
5     Transaction trans = ss.beginTransaction();  
6     try {  
7         trans.begin();  
8         list = ss.createQuery("from pojo.SinhVienPOJO").list();  
9         trans.commit();  
10    } catch (Exception ex) {  
11        System.out.append(ex.getMessage());  
12    }  
13    return list;  
14 }
```


App

```
1 List<SinhVienPOJO> list = dao.SinhVienDAO.layDsSinhVien();
2     for (int i = 0; i < list.size(); ++i) {
3         System.out.println(list.get(i).getMaSinhVien() + "-"
4             + list.get(i).getTenSinhVien() + "-"
5             + list.get(i).getNamSinh());
6     }
```

	MaSinhVien	TenSinhVien	NamSinh
▶	SV01	Hồ Văn Tấn	1990
	SV02	Sử Bá Thuận	1990
	SV03	Đào Quang Duy	1989
*	NULL	NULL	NULL

```
SV01-Hồ Văn Tấn-1990
SV02-Sử Bá Thuận-1990
SV03-Đào Quang Duy-1989
```

App

```
1 List<SinhVienPOJO> list = dao.SinhVienDAO.layDsSinhVien();
2   for (int i = 0; i < list.size(); ++i) {
3       System.out.println(list.get(i).getMaSinhVien() + "-"
4           + list.get(i).getTenSinhVien() + "-"
5           + list.get(i).getNamSinh());
6       Iterator<LopHocPOJO> dsLop = list.get(i).getDsLopHoc().iterator();
7       while (dsLop.hasNext()) {
8           LopHocPOJO lh = dsLop.next();
9           System.out.println(lh.getMaLopHoc() +
10               "-" + lh.getTenLopHoc() +
11               "-" + lh.getSoTinChi());
12       }
13   }
```

???

failed to lazily initialize a collection of role:.pojo.SinhVienPOJO.dsLopHoc, no session or session was closed

App

- Cách giải quyết:
 - Điều chỉnh thuộc tính lazy trong file mapping.
 - Mặc định lazy = “true”

```
1 <set name="dsLopHoc" table="chitietlophoc" lazy="false" fetch="join">
2     <key column="MaSinhVien"/>
3     <many-to-many column="MaLopHoc" class="pojo.LopHocPOJO"/>
4 </set>
```

Lazy Initialization & fetch

- Lazy = “**true**” truy vấn lớp cha **sẽ không kèm theo** truy vấn lớp con
- Lazy = “**false**” truy vấn lớp cha **kèm theo** truy vấn lớp con.
 - Fetch = “**select**” sử dụng select để truy vấn lớp con. ➔ sử dụng 2 câu truy vấn, làm chậm tốc độ.
 - Fetch = “**join**” sử dụng phép kết để gộp truy vấn lớp cha và lớp con trong 1 truy vấn.

Lazy Initialization & fetch

- Trong Hibernate, Lazy Initialization giúp
 - Tránh các câu truy vấn cơ sở dữ liệu không cần thiết
 - Gia tăng hiệu suất thực thi
 - Lazy mặc định có giá trị là `true`

App

```
1 List<SinhVienPOJO> list = dao.SinhVienDAO.layDsSinhVien();
2     for (int i = 0; i < list.size(); ++i) {
3         System.out.println(list.get(i).getMaSinhVien() + "-"
4             + list.get(i).getTenSinhVien() + "-"
5             + list.get(i).getNamSinh());
6         Iterator<LopHocPOJO> dsLop = list.get(i).getDsLopHoc().iterator();
7         while (dsLop.hasNext()) {
8             LopHocPOJO lh = dsLop.next();
9             System.out.println(lh.getMaLopHoc() +
10                 "-" + lh.getTenLopHoc() +
11                 "-" + lh.getSoTinChi());
12         }
13     }
```

Kết quả sau khi chạy lại đoạn code. ➡

```
SV01-Hồ Văn Tân-1990
CTT02-Tin học cơ sở-4
CTT01-Nhập môn lập trình-4
SV02-Sử Bá Thuần-1990
CTT02-Tin học cơ sở-4
SV03-Đào Quang Duy-1989
```


SinhVienDAO

```
1 public static boolean dangKyLop(SinhVienPOJO info) {  
2     boolean kq = true;  
3     SessionFactory sf = MyHibernateUtil.getSessionFactory();  
4     Session ss = sf.getCurrentSession();  
5     Transaction trans = ss.beginTransaction();  
6     try {  
7         trans.begin();  
8         ss.saveOrUpdate(info);  
9         trans.commit();  
10    } catch (Exception ex) {  
11        kq = false;  
12        System.out.append(ex.getMessage());  
13    }  
14    return kq;  
15 }
```

App

```
1 SinhVienPOJO sv = dao.SinhVienDAO.layThongTinSinhVien("SV03");  
2   sv.getDsLopHoc().add(new LopHocPOJO("CTT03", "Java phân tán", 4));  
3   dao.SinhVienDAO.dangKyLop(sv);
```

```
SEVERE: Cannot add or update a child row: a foreign key constraint fails (`quanlysinhvien`.`chitietlophoc`, CONSTRAINT `FK_Lop`  
Oct 16, 2011 4:32:40 PM org.hibernate.event.def.AbstractFlushingEventListener performExecutions  
SEVERE: Could not synchronize database state with session  
org.hibernate.exception.ConstraintViolationException: Could not execute JDBC batch update  
Could not execute JDBC batch update      at org.hibernate.exception.SQLStateConverter.convert(SQLStateConverter.java:71)
```

Trong CSDL không tồn
tại sẵn lophoc có mã
là CTT03 như lúc
thêm

Cascade

- Cách giải quyết:
 - Điều chỉnh thuộc tính cascade trong file mapping.
 - Cascade có 2 giá trị: save-update / delete
 - Nếu dùng cả 2 giá trị thì cách nhau bằng dấu ,

```
1 <set name="dsLopHoc" table="chitietlophoc" lazy="false" fetch="join"  
2 cascade="save-update">  
3     <key column="MaSinhVien"/>  
4     <many-to-many column="MaLopHoc" class="pojo.LopHocPOJO"/>  
    </set>
```

App

```
1 SSinhVienPOJO sv = dao.SinhVienDAO.layThongTinSinhVien("SV03");
2     sv.getDsLopHoc().add(new LopHocPOJO("CTT03", "Java phân tán", 4));
3     boolean kq = dao.SinhVienDAO.dangKyLop(sv);
4     if (kq)
5         System.out.println("Thêm thành công");
6     else
7         System.out.println("Thêm thất bại");
```

	MaLopHoc	TenLopHoc	SoTinChi
▶	CTT01	Nhập môn lập trình	4
	CTT02	Tin học cơ sở	4
	CTT03	Java phân tán	4
*	NULL	NULL	NULL

```
Thêm thành công
BUILD SUCCESSFUL (total time: 2 seconds)
```

	MaLopHoc	MaSinhVien
▶	CTT01	SV01
	CTT02	SV01
	CTT02	SV02
	CTT03	SV03
*	NULL	NULL

Tài liệu tham khảo

- Hibernate Core Reference Manual

http://docs.jboss.org/hibernate/core/3.6/reference/en-US/pdf/hibernate_reference.pdf



QUESTIONS?