



MAPPING MANY TO ONE

Sinh viên: Sử Bá Thuần – Hồ Văn Tấn

GVHD: Thầy Nguyễn Hoàng Anh

15/10/2011

Các vấn đề trình bày

1

- **Many to one**

2

- **Lazy Initialization**

3

- **Fetch**

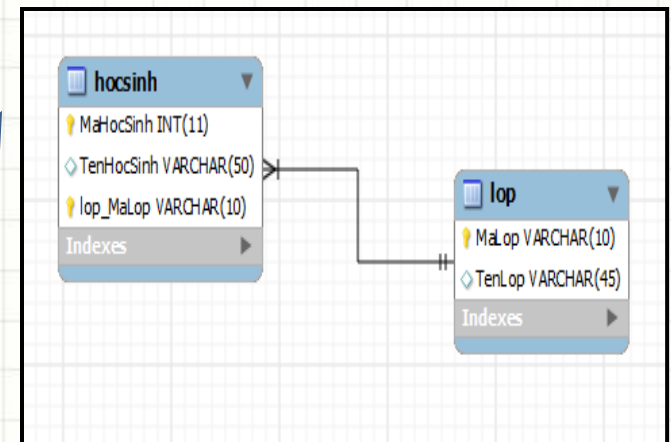
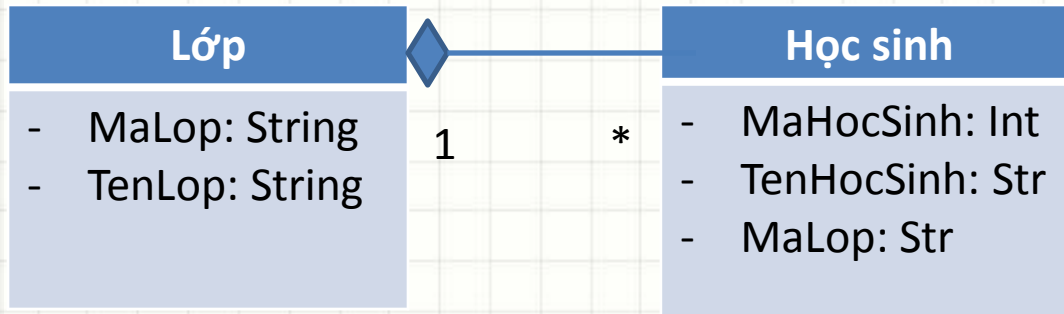
4

- **Cascade**



GIỚI THIỆU

Mapping Many To One



- Một **học sinh** thuộc về 1 **lớp**.
- Một **lớp** có nhiều **học sinh**.

Many to one: LopPOJO

```
1 package pojo;
2
3 public class LopPojo implements java.io.Serializable {
4     private String maLop;
5     private String tenLop;
6 }
//Các phương thức set, get, constructor
```

Many to one: Lop.hbm.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
5 <hibernate-mapping>
6     <class name="pojo.LopPojo" table="lop">
7         <id name="maLop" type="string">
8             <column name="MaLop" length="10"/>
9             <generator class="assigned"/>
10        </id>
11        <property name="tenLop" type="string">
12            <column name="TenLop" length="45" />
13        </property>
14    </class>
15 </hibernate-mapping>
```

Mapping giống như bình thường.

Many to one: HocSinhPOJO

```
1 package pojo;
2
3 public class HocSinhPojo implements java.io.Serializable {
4     private int maHocSinh;
5     private String tenHocSinh;
6     private LopPojo lop;
7
8     //Các phương thức get, set, constructor.
9
10 }
```

Many to one: HocSinh.hbm.xml

```
1 <hibernate-mapping>
2     <class name="pojo.HocSinhPojo" table="hocsinh">
3         <id name="maHocSinh" column="MaHocSinh" type="integer">
4             <generator class="assigned"/>
5         </id>
6         <property name="tenHocSinh" column="TenHocSinh"
7 type="string"/>
8         <many-to-one name="lop" class="pojo.LopPojo" >
9             <column name="MaLop" />
10        </many-to-one>
11    </class>
12 </hibernate-mapping>
```

<many-to-one

name="lop" → tên thuộc tính cần mapping

class="pojo.LopPojo" > → Tên lớp cần mapping tới

<column name="MaLop" /> → Tên cột trong table HocSinh

</many-to-one>

Lấy thông tin học sinh

```
1 public class Main {
2     public static void main(String[] args) {
3         HocSinhPojo hs = null;
4         SessionFactory ssFac = MyHibernateUtil.getSessionFactory();
5         Session ss = ssFac.openSession();
6         ss.getTransaction().begin();
7         try {
8             hs = (HocSinhPojo)ss.get(HocSinhPojo.class, 1);
9             System.out.println("Tên học sinh: " + hs.getTenHocSinh());
10            System.out.println("Mã lớp: " + hs.getLop().getMaLop());
11            System.out.println("Tên lớp: " + hs.getLop().getTenLop());
12        } catch (HibernateException ex ) {
13            System.out.println(ex.getMessage());
14        }
15        finally
16        {
17            ss.close();
18        }
19    }
20 }
```

Lấy thông tin học sinh khi còn mở Session

Thành công

```
Tên học sinh: HỒ VĂN TẤN
Mã lớp: 10A
Tên lớp: Lớp 10 ban A
BUILD SUCCESSFUL (total time: 2 seconds)
```

Lấy thông tin học sinh

```
1 public class Main {
2     public static void main(String[] args) {
3         HocSinhPojo hs = null;
4         SessionFactory ssFac = MyHibernateUtil.getSessionFactory();
5         Session ss = ssFac.openSession();
6         ss.getTransaction().begin();
7         try {
8             hs = (HocSinhPojo)ss.get(HocSinhPojo.class, 1);
9         } catch (HibernateException ex ) {
10             System.out.println(ex.getMessage());
11         }
12         finally
13         {
14             ss.close();
15         }
16         System.out.println("Tên học sinh: " + hs.getTenHocSinh());
17         System.out.println("Mã lớp: " + hs.getLop().getMaLop());
18         System.out.println("Tên lớp: " + hs.getLop().getTenLop());
19     }
20 }
```

Lấy thông tin học sinh sau khi đóng Session

→ chỉ lấy được tên và mã học sinh, không lấy được tên lớp.

Lỗi

Lấy thông tin học sinh

```
Tên học sinh: HỒ VĂN TẤN  
Mã lớp: 10A  
Exception in thread "main" org.hibernate.LazyInitializationException: could not initialize proxy - no Session  
    at org.hibernate.proxy.AbstractLazyInitializer.initialize(AbstractLazyInitializer.java:149)  
    at org.hibernate.proxy.AbstractLazyInitializer.getImplementation(AbstractLazyInitializer.java:195)  
    at org.hibernate.proxy.pojo.javassist.JavassistLazyInitializer.invoke(JavassistLazyInitializer.java:180)  
    at pojo.LopPojo_$$javassist_2.getTenLop(LopPojo_$$javassist_2.java)  
    at qlhs.Main.main(Main.java:31)  
Java Result: 1  
BUILD SUCCESSFUL (total time: 2 seconds)
```

Lấy thông tin học sinh sau khi đóng Session

→ chỉ lấy được tên và mã học sinh, không lấy được tên lớp.

Lỗi

Lấy thông tin học sinh

```
1 public class Main {
2     public static void main(String[] args) {
3         HocSinhPojo hs = null;
4         SessionFactory ssFac = MyHibernateUtil.getSessionFactory();
5         Session ss = ssFac.openSession();
6         ss.getTransaction().begin();
7         try {
8             hs = (HocSinhPojo)ss.get(HocSinhPojo.class, 1);
9             System.out.println("Tên lớp: " + hs.getLop().getTenLop());
10        } catch (HibernateException ex ) {
11            System.out.println(ex.getMessage());
12        }
13        finally
14        {
15            ss.close();
16        }
17        System.out.println("Tên học sinh: " + hs.getTenHocSinh());
18        System.out.println("Mã lớp: " + hs.getLop().getMaLop());
19    }
20 }
```

```
Tên lớp: Lớp 10 ban A
Tên học sinh: Hồ Văn Tấn
Mã lớp: 10A
BUILD SUCCESSFUL (total time: 2 seconds)
```

Thành công

Lấy thông tin học sinh

- Nguyên nhân lỗi:
 - Cơ chế **Lazy Initialization** đang được bật (= true)
 - Truy vấn đối tượng **HocSinh** sẽ không kèm theo truy vấn đối tượng **Lop**. (chỉ có thể truy vấn được mã lớp mà không truy vấn được tên lớp).
 - Truy vấn đối tượng cha sẽ không kèm theo truy vấn đối tượng con.

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`

Cách 1

- Sau khi có mã lớp, ta dùng làm lấy thông tin lớp theo mã lớp

```
LopDAO.layThongTinLop(int maLop);
```

Cách 2 – Khai báo **lazy = false** trong Hocsinh.hbm.xml

```
1 <hibernate-mapping>
2   <class name="pojo.HocSinhPojo" table="hocsinh">
3     <id name="maHocSinh" column="MaHocSinh" type="integer">
4       <generator class="assigned"/>
5     </id>
6     <property name="tenHocSinh" column="TenHocSinh" type="string"/>
7     <many-to-one name="lop" class="pojo.LopPojo" lazy="false" >
8       <column name="MaLop" />
9     </many-to-one>
10   </class>
11 </hibernate-mapping>
```

Cơ chế fetch

- 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 select để truy vấn cả lớp cha và con, cách này không hiệu quả vì phải truy xuất tới cơ sở dữ liệu 2 lần.
 - Fetch = “**join**” sử dụng phép kết để gộp truy vấn lớp cha và lớp con trong 1 truy vấn. → hiệu suất cao hơn, sử dụng 1 câu truy vấn.

Cơ chế fetch – sử dụng **select**

Hocsinh.hbm.xml

```
1 <hibernate-mapping>
2   <class name="pojo.HocSinhPojo" table="hocsinh">
3     <id name="maHocSinh" column="MaHocSinh" type="integer">
4       <generator class="assigned"/>
5     </id>
6     <property name="tenHocSinh" column="TenHocSinh" type="string"/>
7     <many-to-one name="lop" class="pojo.LopPojo" lazy="false" fetch="select">
8       <column name="MaLop" />
9     </many-to-one>
10   </class>
11 </hibernate-mapping>
```

Chú ý: mỗi khi sửa lại file cấu hình xml (cấu hình hibernate, cấu hình mapping, ...) Phải **clean and built** lại project thì thay đổi mới có hiệu lực.

Cơ chế fetch – sử dụng **select**

Hocsinh.hbm.xml

```
1 <hibernate-mapping>
2   <class name="pojo.HocSinhPojo" table="hocsinh">
3     <id name="maHocSinh" column="MaHocSinh" type="integer">
4       <generator class="assigned"/>
5     </id>
6     <property name="tenHocSinh" column="TenHocSinh" type="string"/>
7     <many-to-one name="lop" class="pojo.LopPojo" lazy="false" fetch="select">
8       <column name="MaLop" />
9     </many-to-one>
10   </class>
11 </hibernate-mapping>
```

Chú ý: mỗi khi sửa lại file cấu hình xml (cấu hình hibernate, cấu hình mapping, ...) Phải **clean and built** lại project thì thay đổi mới có hiệu lực.

Cơ chế fetch – sử dụng **select**

Bản chất, các câu truy vấn HQL đều được chuyển về SQL, như hình dưới có 2 câu select được gọi. ⇔ **truy xuất CSDL 2 lần**

```
Output
Hibernate-Mapping-Many-To-One-QLHS (dean.jar) x  Hibernate-Mapping-Many-To-One-QLHS (run) x
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Oct 16, 2011 1:00:26 PM org.hibernate.engine.jdbc.internal.LobCreatorBuild
INFO: HHH000423: Disabling contextual LOB creation as JDBC driver reporte
Oct 16, 2011 1:00:26 PM org.hibernate.engine.transaction.internal.Transac
INFO: HHH000399: Using default transaction strategy (direct JDBC transac
Oct 16, 2011 1:00:26 PM org.hibernate.hql.internal.ast.ASTQueryTranslator
INFO: HHH000397: Using ASTQueryTranslatorFactory
Hibernate:
select
    hocsinhpoj0_.MaHocSinh as MaHocSinh1_0_,
    hocsinhpoj0_.TenHocSinh as TenHocSinh1_0_,
    hocsinhpoj0_.MaLop as MaLop1_0_
from
    hocsinh hocsinhpoj0_
where
    hocsinhpoj0_.MaHocSinh=?
Hibernate:
select
    loppoj0_.MaLop as MaLop0_0_,
    loppoj0_.TenLop as TenLop0_0_
from
    lop loppoj0_
where
    loppoj0_.MaLop=?
Tên học sinh: HỒ VĂN TẤN
Mã danh mục: 10A
Tên danh mục: Lớp 10 ban A
BUILD SUCCESSFUL (total time: 1 second)
```

2 câu truy vấn
select được
gọi.

Cơ chế fetch – sử dụng join

Hocsinh.hbm.xml

```
1 <hibernate-mapping>
2   <class name="pojo.HocSinhPojo" table="hocsinh">
3     <id name="maHocSinh" column="MaHocSinh" type="integer">
4       <generator class="assigned"/>
5     </id>
6     <property name="tenHocSinh" column="TenHocSinh" type="string"/>
7     <many-to-one name="lop" class="pojo.LopPojo" lazy="false" fetch="join">
8       <column name="MaLop" />
9     </many-to-one>
10   </class>
11 </hibernate-mapping>
```

Chú ý: mỗi khi sửa lại file cấu hình xml (cấu hình hibernate, cấu hình mapping, ...) Phải **clean and built** lại project thì thay đổi mới có hiệu lực.

Cơ chế fetch – sử dụng join

```
1 public class Main {
2     public static void main(String[] args) {
3         HocSinhPojo hs = null;
4         SessionFactory ssFac = MyHibernateUtil.getSessionFactory();
5         Session ss = ssFac.openSession();
6         ss.getTransaction().begin();
7         try {
8             String hql="select hs
9                             from HocSinhPojo hs left join fetch hs.lop
10                             where hs.maHocSinh=:maHocSinh";
11             Query query=ss.createQuery(hql);
12             query.setInteger("maHocSinh", 1);
13             hs = (HocSinhPojo) query.uniqueResult();
14         } catch (HibernateException ex) {
15             System.out.println(ex.getMessage());
16         } finally {
17             ss.close();
18         }
19         System.out.println("Tên họcsinh: " + hs.getTenHocSinh());
20         System.out.println("Mã danh mục: " + hs.getLop().getMaLop());
21         System.out.println("Tên danh mục: " + hs.getLop().getTenLop());
22     }
23 }
```

Cơ chế fetch – sử dụng join

Có 1 câu select được gọi, có sử dụng phép Join ⇔ truy xuất CSDL 2 lần

```
Output - Hibernate-Mapping-Many-To-One-QLHS (run)
INFO: HHH000397: Using ASTQueryTranslatorFactory
Hibernate:
  select
    hocsinhpoj0_.MaHocSinh as MaHocSinh1_0_,
    loppoj01_.MaLop as MaLop0_1_,
    hocsinhpoj0_.TenHocSinh as TenHocSinh1_0_,
    hocsinhpoj0_.MaLop as MaLop1_0_,
    loppoj01_.TenLop as TenLop0_1_
  from
    hocsinh hocsinhpoj0_
  left outer join
    lop loppoj01_
      on hocsinhpoj0_.MaLop=loppoj01_.MaLop
  where
    hocsinhpoj0_.MaHocSinh=?
Tên học sinh: Hồ Văn Tấn
Mã danh mục: 10A
Tên danh mục: Lớp 10 ban A
BUILD SUCCESSFUL (total time: 2 seconds)
```

1 câu truy vấn
select được
gọi.

Cascade

- Save – update
- Delete

Cascade – None cascade

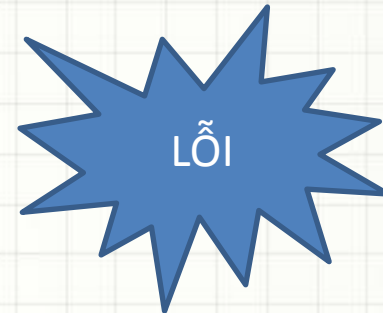
```
1 <hibernate-mapping>
2   <class name="pojo.HocSinhPojo" table="hocsinh">
3     <id name="maHocSinh" column="MaHocSinh" type="integer">
4       <generator class="assigned"/>
5     </id>
6     <property name="tenHocSinh" column="TenHocSinh" type="string"/>
7     <many-to-one name="lop" class="pojo.LopPojo"
8       lazy="false" fetch="join" cascade="none">
9       <column name="MaLop" />
10    </many-to-one>
11  </class>
</hibernate-mapping>
```

Mặc định nếu không khai báo thì **cascade=none**

Cascade – không dùng update-save

```
1 public static void main(String[] args) {
2     HocSinhPojo hs = new HocSinhPojo(15, "Trần Văn Đạt", null);
3     LopPojo lop = new LopPojo("12E", "Lớp 12 chuyên Hóa");
4
5     hs.setLop(lop);
6     if(HocSinhDAO.themHocSinh(hs))
7     {
8         System.out.println("Thêm thành công!");
9     }
10    else
11        System.out.println("Thêm thất bại!");
12 }
```

```
Output - Hibernate-Mapping-Many-To-One-QLHS (run)
Oct 17, 2011 12:17:48 AM org.hibernate.internal.CoreMessageLogger_$logger warn
WARN: SQL Error: 1452, SQLState: 23000
Cannot add or update a child row: a foreign key constraint fails (`quanlyhocsinh`
Oct 17, 2011 12:17:48 AM org.hibernate.internal.CoreMessageLogger_$logger error
Thêm thất bại!
ERROR: Cannot add or update a child row: a foreign key constraint fails (`quanlyh
BUILD SUCCESSFUL (total time: 2 seconds)
```



Lớp 12E không tồn tại trong cơ sở dữ liệu.

Cascade – sử dụng update-save

HocSinh.hbm.xml

```
1 <hibernate-mapping>
2   <class name="pojo.HocSinhPojo" table="hocsinh">
3     <id name="maHocSinh" column="MaHocSinh" type="integer">
4       <generator class="assigned"/>
5     </id>
6     <property name="tenHocSinh" column="TenHocSinh" type="string"/>
7     <many-to-one name="lop" class="pojo.LopPojo"
8       lazy="false" fetch="join" cascade="save-update">
9       <column name="MaLop" />
10    </many-to-one>
11  </class>
</hibernate-mapping>
```

Cascade – sử dụng update-save

```
1 public static void main(String[] args) {  
2     HocSinhPojo hs = new HocSinhPojo(15, "Trần Văn Đạt", null);  
3     LopPojo lop = new LopPojo("12E", "Lớp 12 chuyên Hóa");  
4  
5     hs.setLop(lop);  
6     if(HocSinhDAO.themHocSinh(hs))  
7     {  
8         System.out.println("Thêm thành công!");  
9     }  
10    else  
11        System.out.println("Thêm thất bại!");  
    }
```

MaLop	TenLop
10A	Lớp 10 ban A
10B	Lớp 10 ban B
10C	Lớp 10 ban C
10D	Lớp 10 ban D
11A	Lớp 11 ban A
11B	Lớp 11 ban B
12A	Lớp 12 ban A
12B	Lớp 12 ban B
12D	Lớp 12 ban D
12E	Lớp 12 chuyên Hóa
NULL	NULL



	MaHocSinh	TenHocSinh	MaLop
1		Hồ Văn Tấn	10A
2		Sử Bá Thuần	10B
3		Văn Kinh Luân	10A
4		Lương Quang Hà	10C
5		Lý Thị Loan	10D
▶	15	Trần Văn Đạt	12E
*	NULL	NULL	NULL

Cascade – không dùng delete

```
1 public class Main {  
2     public static void main(String[] args) {  
3         if(LopDAO.xoaLop("12E"))  
4             System.out.println("Xóa thành công!");  
5         else  
6             System.out.println("Xóa thất bại!");  
7     }  
8 }
```

Output - Hibernate-Mapping-Many-To-One-QLHS (run)

```
ERROR: Cannot delete or update a parent row: a foreign key constraint fails (`quanlyhoc`  
Oct 17, 2011 12:54:22 AM org.hibernate.engine.jdbc.batch.internal.AbstractBatchImpl  
org.hibernate.exception.ConstraintViolationException: Cannot delete or update a parent  
INFO: HHH000010: On release of batch it still contained JDBC statements  
BUILD SUCCESSFUL (total time: 2 seconds)
```

LỖI

Không thể xóa lớp 12E → lỗi tham chiếu khóa ngoại

Cascade – sử dụng delete

CHÚ Ý

The screenshot shows the 'hocsinh' database management tool interface. The 'Foreign Keys' tab is selected, displaying the configuration for the 'FK_Lop' foreign key. The 'Referenced Table' is 'quanlyhocsinh'. 'lop'. The 'Column' list includes 'MaHocSinh', 'TenHocSinh', and 'MaLop'. The 'Referenced Column' is 'MaLop'. The 'Foreign Key Options' section shows 'On Update' and 'On Delete' both set to 'CASCADE'. The 'Skip in SQL generation' checkbox is unchecked. The 'Foreign Key Comment' field is empty. The bottom status bar shows 'DBMS feedback messages will go here upon applying changes.' and buttons for 'Apply', 'Revert', and 'Close'.

Foreign Key Name	Referenced Table
FK_Lop	'quanlyhocsinh'. 'lop'

Column	Referenced Column
<input type="checkbox"/> MaHocSinh	
<input type="checkbox"/> TenHocSinh	
<input checked="" type="checkbox"/> MaLop	MaLop

Foreign Key Options

On Update: CASCADE

On Delete: CASCADE

☐ Skip in SQL generation

- Foreign Key Comment -

Table Columns Indexes Foreign Keys Triggers Partitioning Options

DBMS feedback messages will go here upon applying changes.

Apply Revert Close

Để dùng Cascade trong hibernate phải cho phép sử dụng cascade trong CSDL

Cascade – sử dụng delete

```
1 <hibernate-mapping>
2   <class name="pojo.HocSinhPojo" table="hocsinh">
3     <id name="maHocSinh" column="MaHocSinh" type="integer">
4       <generator class="assigned"/>
5     </id>
6     <property name="tenHocSinh" column="TenHocSinh" type="string"/>
7     <many-to-one name="lop" class="pojo.LopPojo"
8       lazy="false" fetch="join" cascade="save-update, delete">
9       <column name="MaLop" />
10    </many-to-one>
11  </class>
</hibernate-mapping>
```

Cascade – sử dụng delete

```
1 public class Main {  
2     public static void main(String[] args) {  
3         if(LopDAO.xoaLop("12E"))  
4             System.out.println("Xóa thành công!");  
5         else  
6             System.out.println("Xóa thất bại!");  
7     }  
8 }
```

	MaLop	TenLop
▶	10A	Lớp 10 ban A
	10B	Lớp 10 ban B
	10C	Lớp 10 ban C
	10D	Lớp 10 ban D
	11A	Lớp 11 ban A
	11B	Lớp 11 ban B
	12A	Lớp 12 ban A
	12B	Lớp 12 ban B
	12D	Lớp 12 ban D
*	NULL	NULL



Xóa thành công!
BUILD SUCCESSFUL (total time: 1 second)

	MaLop	TenLop
▶	10A	Lớp 10 ban A
	10B	Lớp 10 ban B
	10C	Lớp 10 ban C
	10D	Lớp 10 ban D
	11A	Lớp 11 ban A
	11B	Lớp 11 ban B
	12A	Lớp 12 ban A
	12B	Lớp 12 ban B
	12D	Lớp 12 ban D
*	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?