



MAPPING ONE TO MANY

Sinh viên: Sử Bá Thuần – Hồ Văn Tấn

GVHD: Thầy Nguyễn Hoàng Anh

15/10/2011

Các vấn đề trình bày

1

- One To Many

2

- Lazy Initialization & Fetch

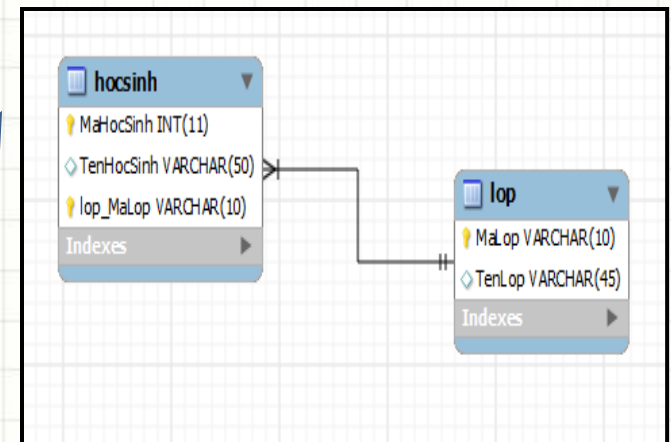
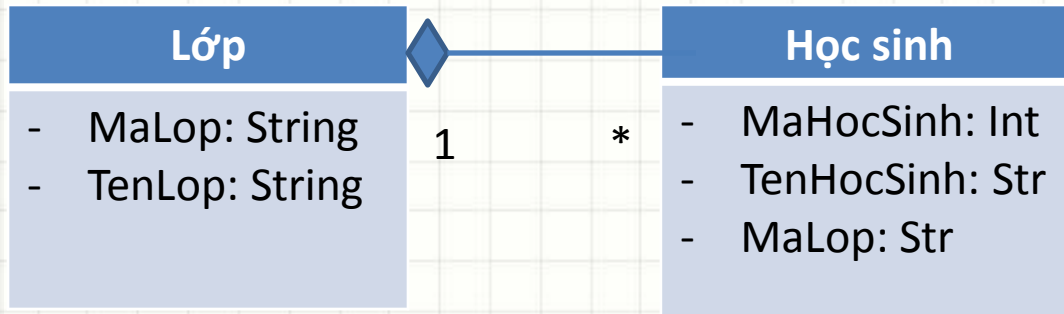
4

- Inverser



MAPPING ONE TO MANY

Mapping One To Many



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

LopPOJO

```
1 package pojo;
2
3 import java.util.HashSet;
4 import java.util.Set;
5
6 public class LopPojo implements java.io.Serializable {
7     private String maLop;
8     private String tenLop;
9     private Set<HocSinhPojo> danhSachHocSinh = new HashSet<HocSinhPojo>(0);
10
11     // Các phương thức get, set, construction
12
13 }
```

Lop.hbm.xml

```
1 <hibernate-mapping>
2   <class name="pojo.LopPojo" table="lop">
3     <id name="maLop" type="string">
4       <column name="MaLop" length="10"/>
5       <generator class="assigned"/>
6     </id>
7     <property name="tenLop" type="string">
8       <column name="TenLop" length="45" />
9     </property>
10    <set name="danhSachHocSinh" lazy="false" fetch="select">
11      <key>
12        <column name="MaLop"/>
13      </key>
14      <one-to-many class="pojo.HocSinhPojo" />
15    </set>
  </class>
</hibernate-mapping>
```


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 }
```

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>
```


DAO: Lấy thông tin lớp học

```
1  public static LopPojo layThongTinLop(String maLop) {  
2      LopPojo lop = null;  
3      SessionFactory ssFac = MyHibernateUtil.getSessionFactory();  
4      Session ss = ssFac.getCurrentSession();  
5      Transaction trans = ss.getTransaction();  
6      trans.begin();  
7      try {  
8          lop = (LopPojo)ss.get(LopPojo.class, maLop);  
9          trans.commit();  
10     } catch (HibernateException ex ) {  
11         System.out.println(ex.getMessage());  
12     }  
13     return lop;  
14 }
```

Lấy thông tin lớp học

```
1 public class Main {
2     public static void main(String[] args) {
3         LopPojo lop = LopDAO.layThongTinLop("10A");
4         System.out.println("Mã lớp: " + lop.getMaLop());
5         System.out.println("Tên lớp: " + lop.getTenLop());
6         System.out.println("-----");
7         Iterator<HocSinhPojo> dsHocSinh =
8             lop.getDanhsachHocSinh().iterator();
9
10        while(dsHocSinh.hasNext())
11        {
12            HocSinhPojo hs = dsHocSinh.next();
13            System.out.println("Mã học sinh: " + hs.getMaHocSinh());
14            System.out.println("Tên học sinh: " +
15                               hs.getTenHocSinh());
16            System.out.println("_____");
17        }
18    }
19 }
20
```

Lấy thông tin lớp học

Kết quả

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



Mã lớp: 10A



Tên lớp: Lớp 10 ban A



Mã học sinh: 1



Tên học sinh: HỒ VĂN TẤN

Mã học sinh: 4

Tên học sinh: Lương Quang Hà

Mã học sinh: 3

Tên học sinh: Văn Kinh Luân

BUILD SUCCESSFUL (total time: 2 seconds)



INVERSER

Khi **tạo mới** hay **thay đổi** thông tin “một lớp học” thì có cần phải **cập nhật** lại “mã lớp” của học sinh hay không?

Inverser - False

Lưu ý: để xuất câu truy vấn SQL ra console cho tiện việc xem xét như các slide bên dưới ta phải thêm **<property name="hibernate.format_sql">true</property>** vào file cấu hình hibernate.cfg.xml.

```
1 <hibernate-configuration>
2   <session-factory>
3     <property
4       name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
5     <property
6       name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
7     <property
8       name="hibernate.connection.url">jdbc:mysql://localhost:3306/quanlysach?use
9       Unicode=true&characterEncoding=UTF-8</property>
10    <property name="hibernate.connection.username">root</property>
11    <property name="hibernate.connection.password">password</property>
12    <property
13      name="hibernate.current_session_context_class">thread</property>
14    <property name="hibernate.format_sql">true</property>
15    <mapping resource="entity/Khachhang.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

Inverser - False

Khi **tạo mới** hay **thay đổi** thông tin “một lớp học” thì có cần phải **cập nhật** lại “mã lớp” của học sinh hay không?

- **Nếu inverser = false (Mặc định)**

```
1 <set name="danhSachHocSinh" lazy="false" fetch="select"  
2 inverse="false">  
3     <key>  
4         <column name="MaLop"/>  
5     </key>  
6     <one-to-many class="pojo.HocSinhPojo" />  
7 </set>
```

Inverser = false đồng nghĩa với **LopHoc** đóng vai trò chủ đạo trong mối quan hệ **LopHoc-HocSinh**, vì vậy khi tạo mới hay cập nhật thông tin một lớp học sẽ phải cập nhật lại mã lớp của học sinh.

Inverser - FALSE

```
1 public class Main {
2     public static void main(String[] args) {
3
4         Session session=MyHibernateUtil.getSessionFactory().getCurrentSession();
5         session.beginTransaction();
6
7         LopPojo lop = new LopPojo("12F", "Lớp 12 chuyên văn");
8
9         HocSinhPojo hs = new HocSinhPojo(6, "Mạc Thị Bưởi", lop);
10
11         lop.getDanhSachHocSinh().add(hs);
12
13         session.save(lop);
14         session.save(hs);
15         session.getTransaction().commit();
16     }
17 }
```

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

Hibernate:

insert

into

lop

(TenLop, MaLop)

values

(?, ?)

Hibernate:

insert

into

hocsinh

(TenHocSinh, MaLop, MaHocSinh)

values

(?, ?, ?)

Hibernate:

update

hocsinh

set

MaLop=?

where

MaHocSinh=?

BUILD SUCCESSFUL (total time: 1 second)

Inverser - TRUE

Hibernate:

```
insert  
into  
    lop  
    (TenLop, MaLop)  
values  
    (?, ?)
```

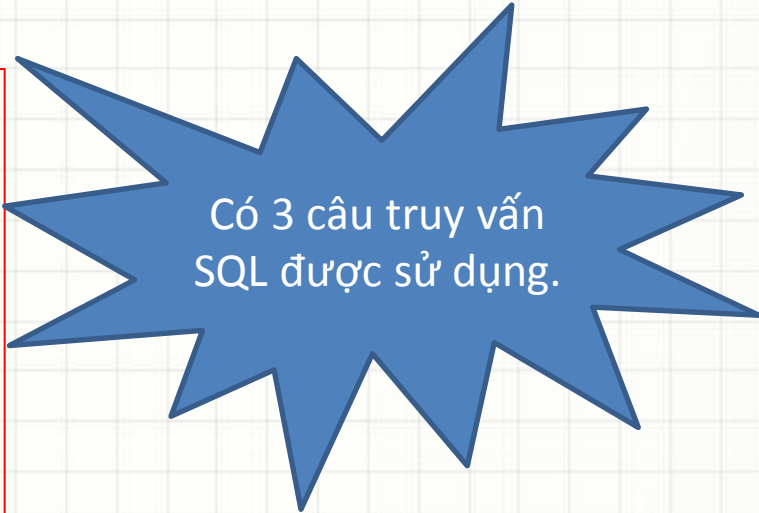
Hibernate:

```
insert  
into  
    hocsinh  
    (TenHocSinh, MaLop, MaHocSinh)  
values  
    (?, ?, ?)
```

Hibernate:

```
update  
    hocsinh  
set  
    MaLop=?  
where  
    MaHocSinh=?
```

BUILD SUCCESSFUL (total time: 1 second)



Có 3 câu truy vấn SQL được sử dụng.

- Có 2 câu insert dữ liệu
- Có 1 câu update dùng để Update lại mã lớp của học sinh.
- Trong trường hợp này câu update không cần thiết.

Inverser

Khi **tạo mới** hay **thay đổi** thông tin “một lớp học” thì có cần phải **cập nhật** lại “mã lớp” của học sinh hay không?

- **Nếu inverser = true**

```
1 <set name="danhSachHocSinh" lazy="false" fetch="select"  
2 inverse="true">  
3     <key>  
4         <column name="MaLop"/>  
5     </key>  
6     <one-to-many class="pojo.HocSinhPojo" />  
7 </set>
```

Inverser = true đồng nghĩa với HocSinh đóng vai trò chủ đạo trong mối quan hệ này, vì vậy khi tạo mới hay cập nhật thông tin một lớp học sẽ không cần phải cập nhật lại mã lớp của học sinh.

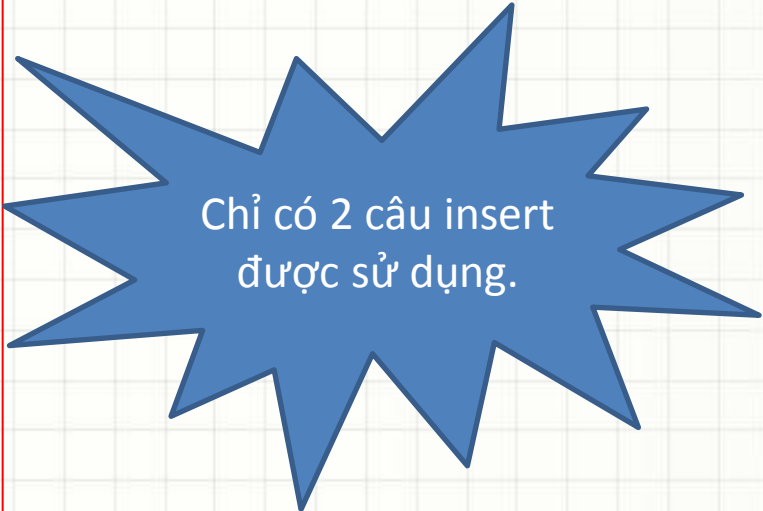
Inverser - TRUE

```
1 public class Main {  
2     public static void main(String[] args) {  
3  
4         Session session=MyHibernateUtil.getSessionFactory().getCurrentSession();  
5         session.beginTransaction();  
6  
7         LopPojo lop = new LopPojo("12F", "Lớp 12 chuyên văn");  
8  
9         HocSinhPojo hs = new HocSinhPojo(6, "Mạc Thị Bưởi", lop);  
10  
11         lop.getDanhSachHocSinh().add(hs);  
12  
13         session.save(lop);  
14         session.save(hs);  
15         session.getTransaction().commit();  
16     }  
17 }
```

```
: Output - Hibernate-Mapping-Many-To-One-QLHS (run)  
Hibernate:  
insert  
into  
    lop  
    (TenLop, MaLop)  
values  
    (?, ?)  
Hibernate:  
insert  
into  
    hocsinh  
    (TenHocSinh, MaLop, MaHocSinh)  
values  
    (?, ?, ?)  
BUILD SUCCESSFUL (total time: 1 second)
```

Inverser - TRUE

```
Hibernate:
  insert
  into
    lop
    (TenLop, MaLop)
  values
    (?, ?)
Hibernate:
  insert
  into
    hocsinh
    (TenHocSinh, MaLop, MaHocSinh)
  values
    (?, ?, ?)
BUILD SUCCESSFUL (total time: 1 second)
```



Chỉ có 2 câu insert
được sử dụng.

Bớt được một truy vấn đến CSDL
→ tăng hiệu năng đáng kể.

Tài liệu tham khảo

- Hibernate Core Reference Manual

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



QUESTIONS?