

CHUYÊN ĐỀ JAVA

# **HIBERNATE MAPPING**

## **MANY-TO-MANY**

Nguyễn Hoàng Anh

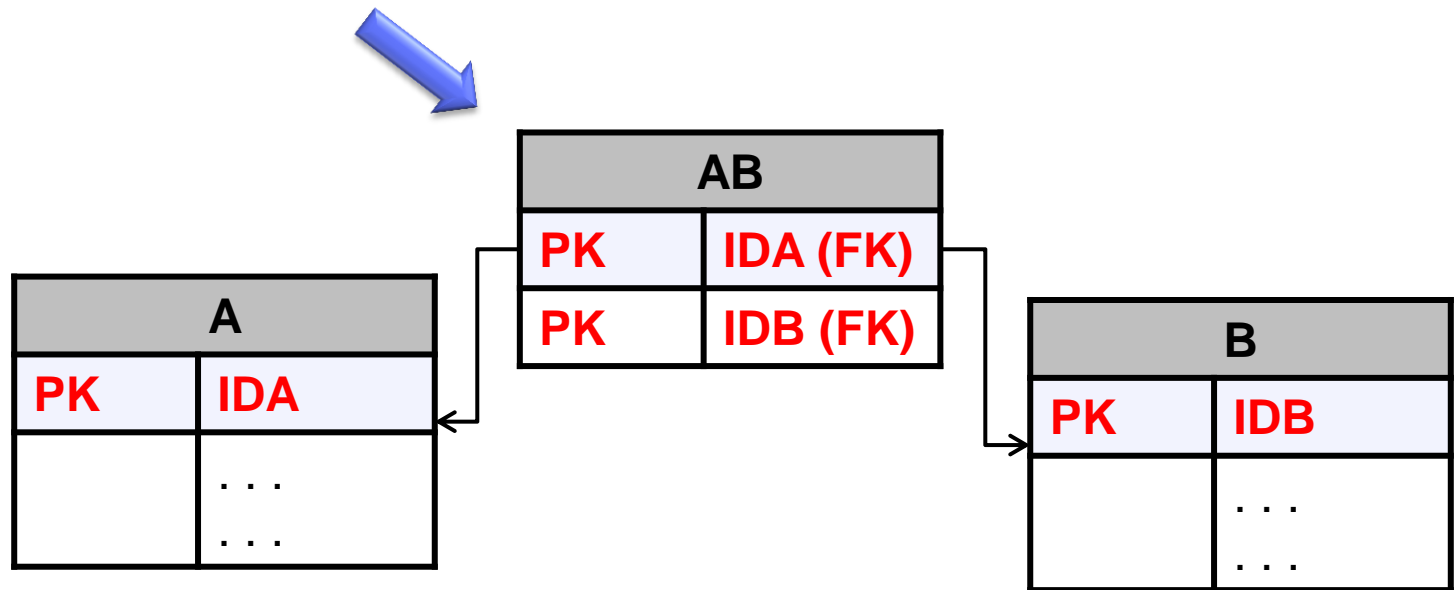
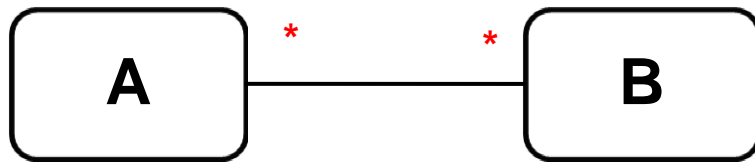
Email: [nhanh@fit.hcmus.edu.vn](mailto:nhanh@fit.hcmus.edu.vn)

ĐH KHTN, 2011

# Nội dung trình bày

- One to many
- Lazy Initialization
- Fetch
- Cascade

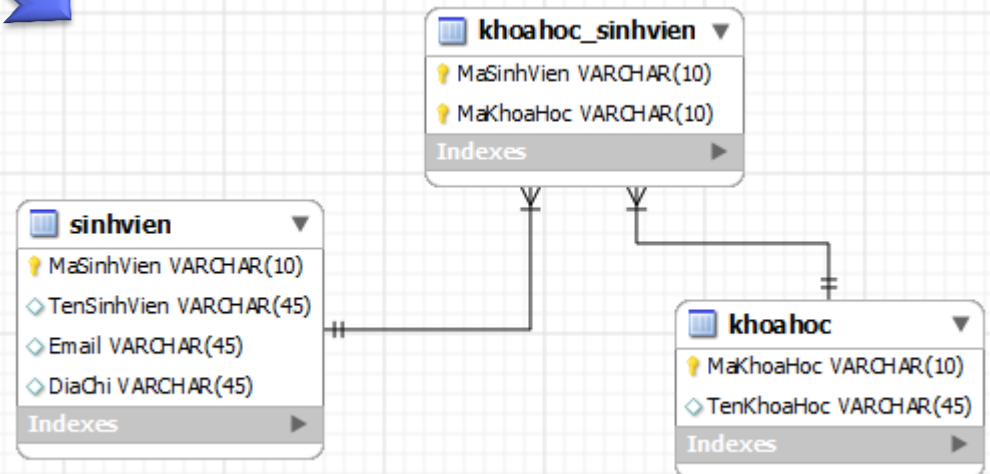
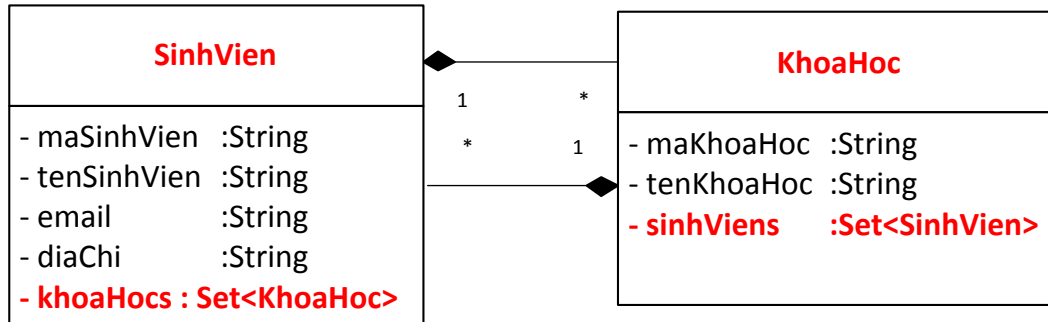
## Mối quan hệ nhiều-nhiều (many-to-many)



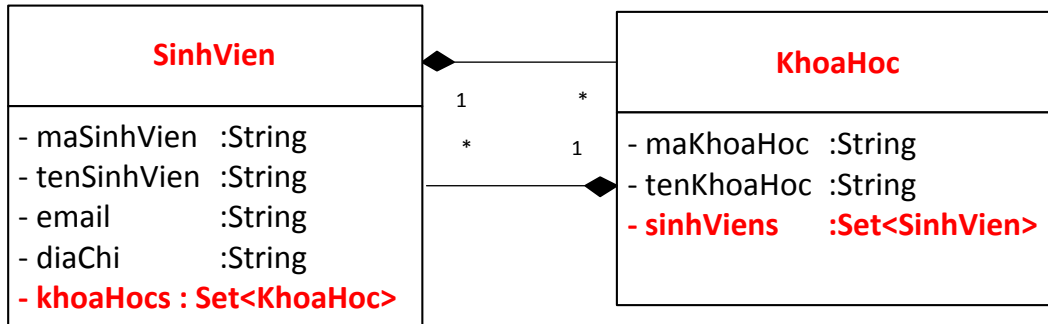
## Mối quan hệ **nhieu-nhiều (many-to-many)**

- Trong ứng dụng đăng ký học phần
  - Mỗi sinh viên có thể đăng ký nhiều khóa học
  - Mỗi khóa học có thể có nhiều sinh viên đăng ký học
- Mối quan hệ hướng từ sinh viên đến khóa học và ngược lại gọi là **mối quan hệ nhiều – nhiều(many-to-many)**

# Many to many



# Many to many

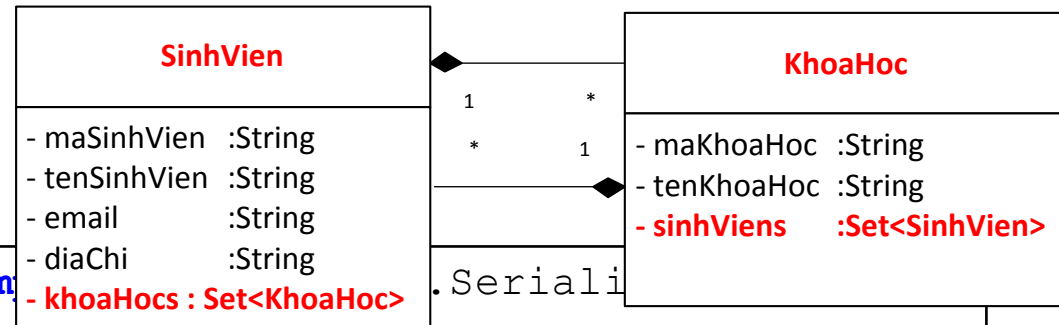


	MaSinhVien	MaKhoaHoc
▶	0312055	KHJAVA
	0312055	KHPHP
	0312079	KHJAVA
	0312079	KHNET
	0312143	KHJAVA
	0312143	KHNET

	MaSinhVien	TenSinhVien	Email	DiaChi
▶	0312055	Cao Xuân Nam	cxnam@gmail.com	228 Vườn Lài Tân Phú
	0312079	Hoàng Vũ Tuấn	hvtuan999@yahoo.com	213 Phú Bình Q11
	0312143	Nguyễn Hoàng Anh	hoanganhis@gmail.com	227 Nguyễn Văn Cừ Q5

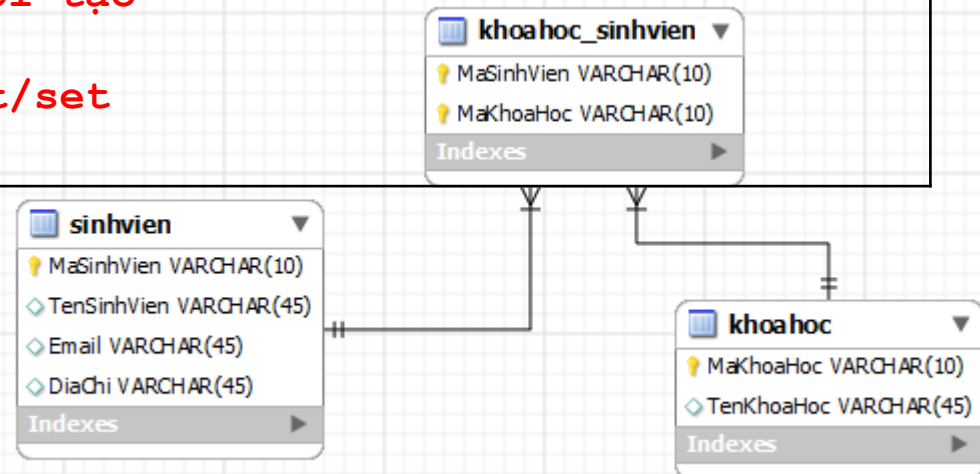
	MaKhoaHoc	TenKhoaHoc
▶	KHJAVA	J2EE
	KHNET	.NET 3.5
	KHPHP	PHP

# SinhVien POJO

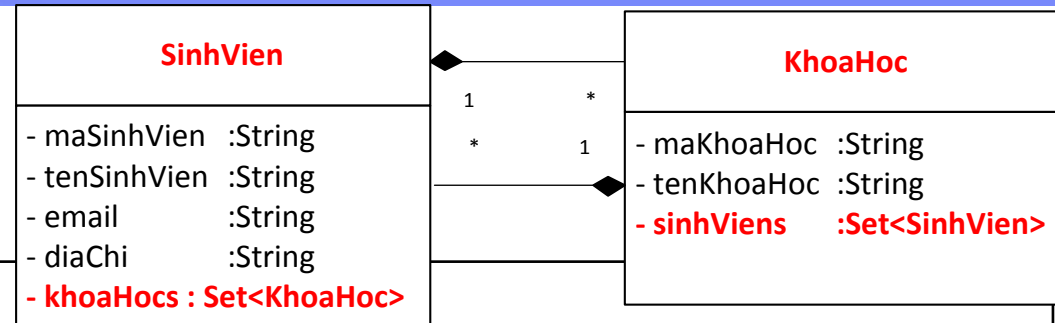


```

1  public class SinhVien implements Serializable {
2
3
4      private String maSinhVien;
5      private String tenSinhVien;
6      private String email;
7      private String diaChi;
8      private Set<KhoaHoc> khoaHocs = new HashSet<KhoaHoc>(0);
9
10     // Các phương thức khởi tạo
11     . . .
12     // Các phương thức get/set
13     . . .
    
```

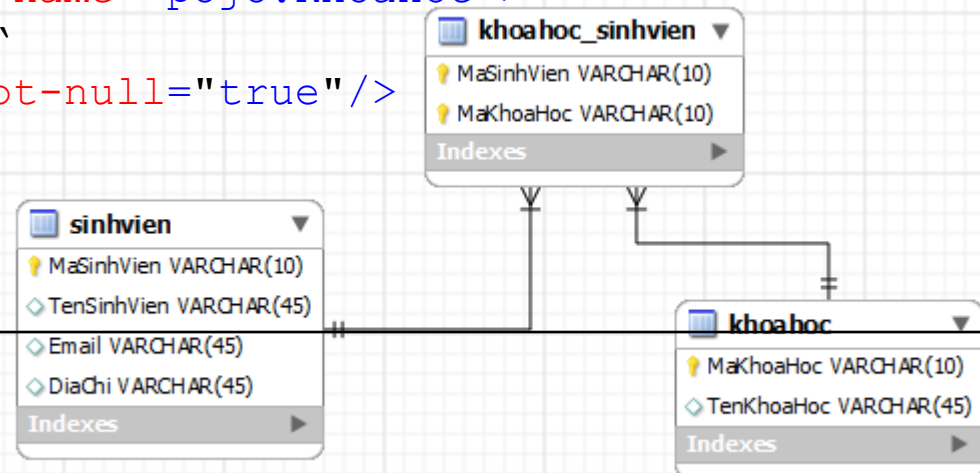


# SinhVien.hbm.xml



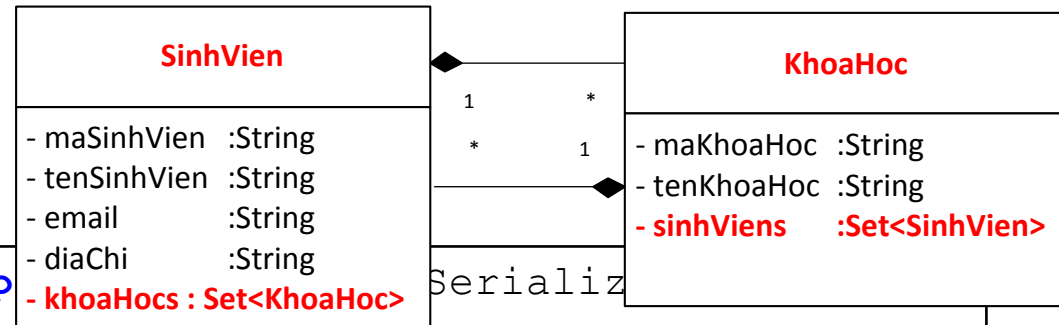
```

1 <hibernate-mapping>
2   <class catalog="quanlydangkyhocphan" name="pojo.SinhVien"
3   table="sinhvien">
4     .
5     .
6     .
7     <set inverse="false" name="khoahocs"
8     table="khoahoc_sinhvien">
9       <key>
10        <column length="10" name="MaSinhVien" not-null="true"/>
11      </key>
12      <many-to-many entity-name="pojo.KhoaHoc">
13        <column length="10"
14        name="MaKhoaHoc" not-null="true"/>
15      </many-to-many>
16    </set>
17  </class>
18 </hibernate-mapping>
  
```





# KhoaHoc POJO



```
1 public class KhoaHoc implements Serializable {
```

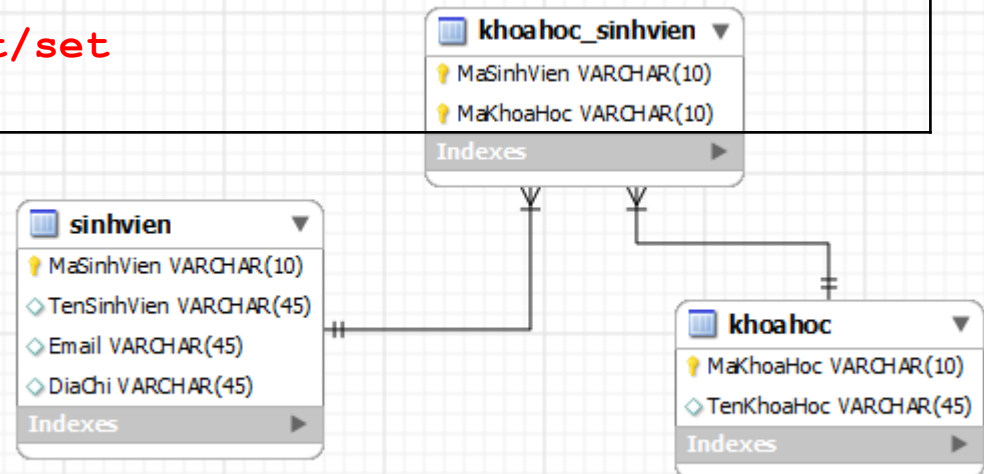
```
2
3
4     private String maKhoaHoc;
5     private String tenKhoaHoc;
6     private Set<SinhVien> sinhviens =
7         new HashSet<SinhVien>(0);
```

```
8
9     // Các phương thức khởi tạo
```

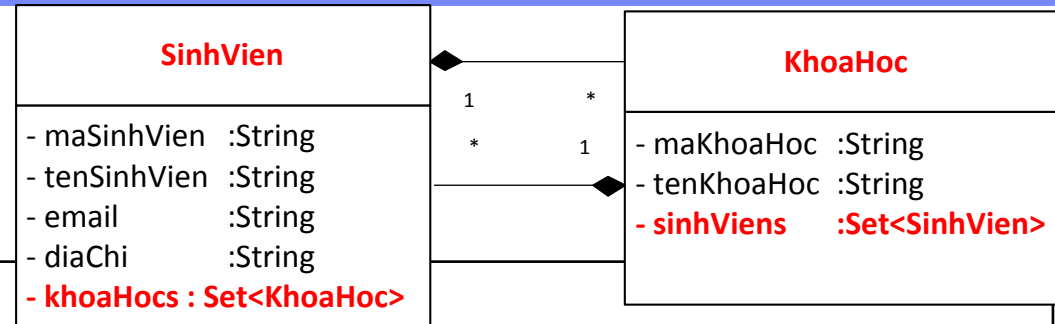
```
10     . . .
```

```
11     // Các phương thức get/set
```

```
12     . . .
```



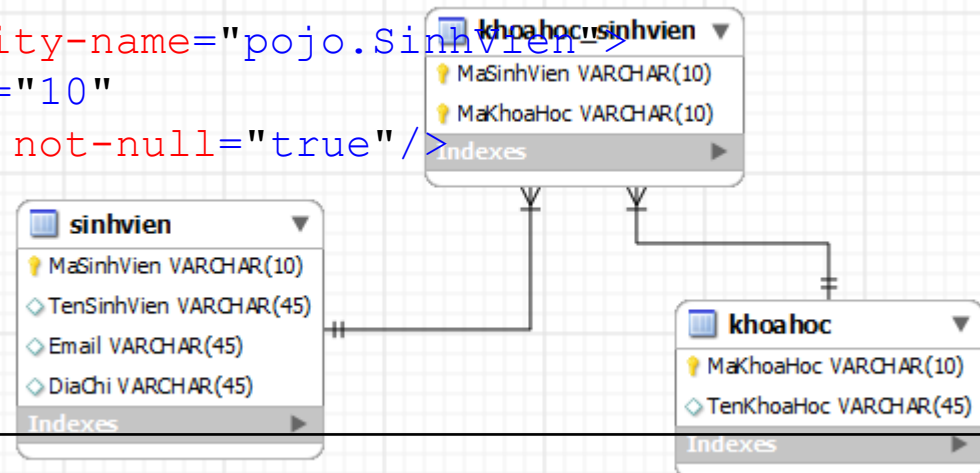
# SinhVien.hbm.xml



```

1 <hibernate-mapping>
2   <class catalog="quanlydangkyhocphan"
3     name="pojo.KhoaHoc" table="khoahoc">
4     . . .
5     <set inverse="true" name="sinhvien"
6       table="khoahoc_sinhvien">
7       <key>
8         <column length="10" name="MaKhoaHoc"
9           not-null="true"/>
10      </key>
11      <many-to-many entity-name="pojo.SinhVien">
12        <column length="10"
13          name="MaSinhVien" not-null="true"/>
14      </many-to-many>
15    </set>
16  </class>
17 </hibernate-mapping>

```



# SinhVienDAO – lấy thông tin sinh viên

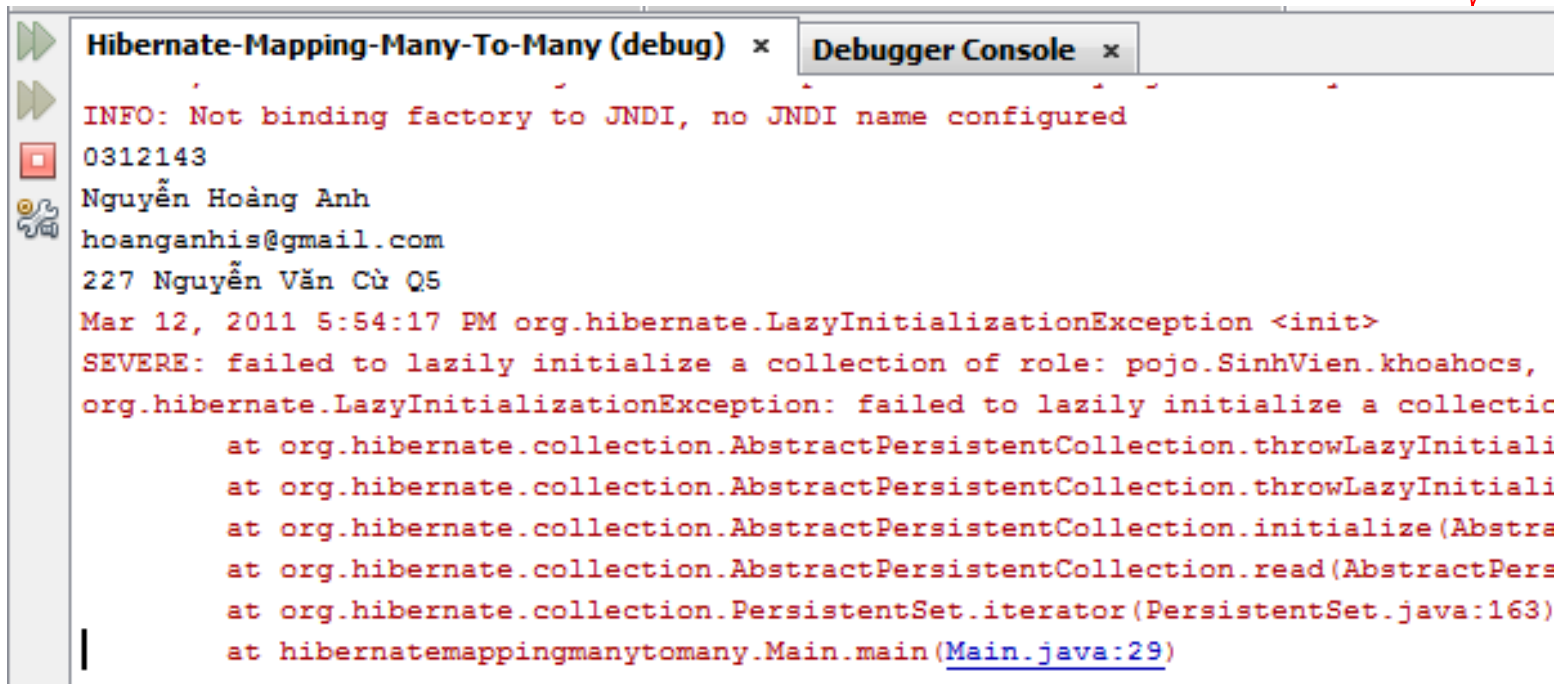
```
1 public class SinhVienDAO {
2     public static SinhVien layThongTinSinhVien(String maSinhVien)
3     {
4         SinhVien sv = null;
5         Session session = HibernateUtil.getSessionFactory()
6                                 .openSession();
7         try {
8             sv =(SinhVien) session.get(SinhVien.class, maSinhVien);
9         } catch (HibernateException ex) {
10             System.err.println(ex);
11         } finally {
12             session.close();
13         }
14         return sv;
15     }
```

# SinhVienDAO – lấy thông tin sinh viên

```
1  public class Main {
2      public static void main(String[] args) {
3          SinhVien sv=SinhVienDAO.layThongTinSinhVien("0312143");
4          System.out.println(sv.getMaSinhVien());
5          System.out.println(sv.getTenSinhVien());
6          System.out.println(sv.getEmail());
7          System.out.println(sv.getDiaChi());
8          Iterator<KhoaHoc> khoaHocs=sv.getKhoahocs().iterator();
9          while(khoaHocs.hasNext()){
10             KhoaHoc kh=khoaHocs.next();
11             System.out.println(kh.getMaKhoaHoc());
12             System.out.println(kh.getTenKhoaHoc());
13         }
14     }
15 }
```

# SinhVienDAO – lấy thông tin sinh viên

Lazy = true

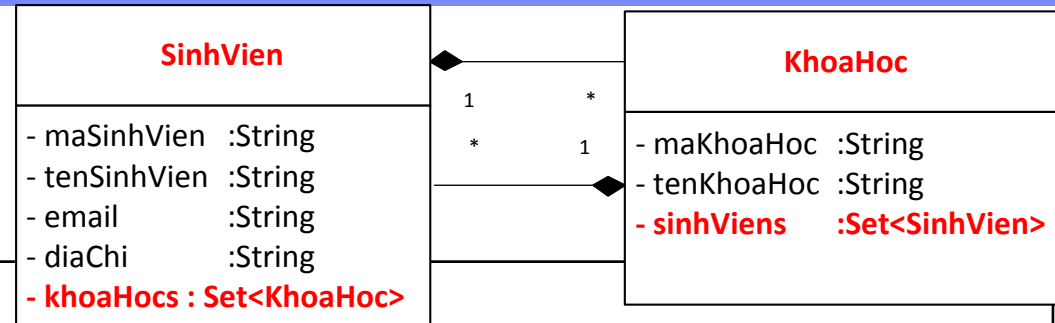


```
Hibernate-Mapping-Many-To-Many (debug) x Debugger Console x
INFO: Not binding factory to JNDI, no JNDI name configured
0312143
Nguyễn Hoàng Anh
hoanganhis@gmail.com
227 Nguyễn Văn Cừ Q5
Mar 12, 2011 5:54:17 PM org.hibernate.LazyInitializationException <init>
SEVERE: failed to lazily initialize a collection of role:.pojo.SinhVien.khoahocs,
org.hibernate.LazyInitializationException: failed to lazily initialize a collectic
    at org.hibernate.collection.AbstractPersistentCollection.throwLazyInitiali
    at org.hibernate.collection.AbstractPersistentCollection.throwLazyInitiali
    at org.hibernate.collection.AbstractPersistentCollection.initialize (Abstra
    at org.hibernate.collection.AbstractPersistentCollection.read (AbstractPers
    at org.hibernate.collection.PersistentSet.iterator (PersistentSet.java:163)
    at hibernatemappingmanytomany.Main.main (Main.java:29)
```

# Lazy Initialization

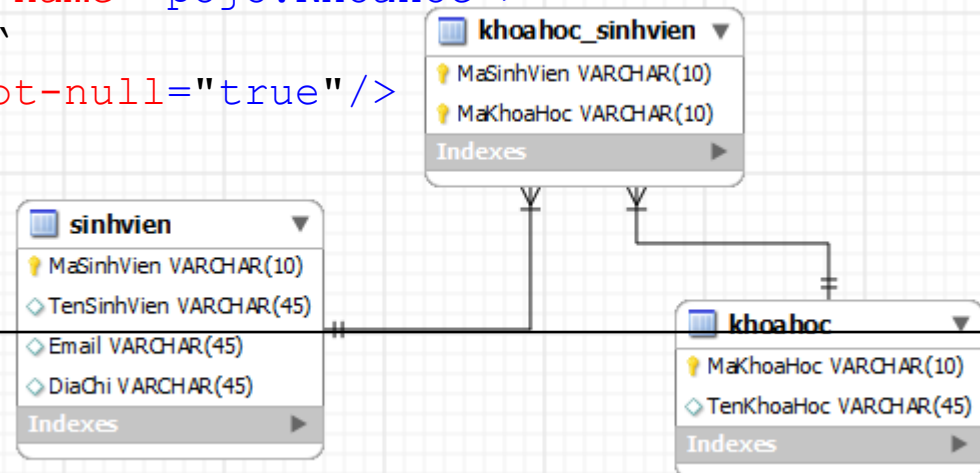
- Trong Hibernate, **Lazy Initialization** giúp
  - Tránh các câu truy vấn cơ sở dữ liệu không cần thiết
  - Gia tăng hiệu suất thực thi
  - Lazy mặc định có giá trị là **true**

# SinhVien.hbm.xml



```

1 <hibernate-mapping>
2   <class catalog="quanlydangkyhocphan" name="pojo.SinhVien"
3   table="sinhvien">
4     .
5     .
6     .
7     <set inverse="false" name="khoahocs" fetch="join"
8       lazy="false" table="khoahoc_sinhvien">
9       <key>
10        <column length="10" name="MaSinhVien" not-null="true"/>
11      </key>
12      <many-to-many entity-name="pojo.KhoaHoc">
13        <column length="10"
14          name="MaKhoaHoc" not-null="true"/>
15      </many-to-many>
16    </set>
17  </class>
18 </hibernate-mapping>
  
```



# SinhVienDAO – lấy thông tin sinh viên

```
1  public class Main {
2      public static void main(String[] args) {
3          SinhVien sv=SinhVienDAO.layThongTinSinhVien("0312143");
4          System.out.println(sv.getMaSinhVien());
5          System.out.println(sv.getTenSinhVien());
6          System.out.println(sv.getEmail());
7          System.out.println(sv.getDiaChi());
8          Iterator<KhoaHoc> khoaHocs=sv.getKhoahocs().iterator();
9          while(khoaHocs.hasNext()){
10             KhoaHoc kh=khoaHocs.next();
11             System.out.println(kh.getMaKhoaHoc());
12             System.out.println(kh.getTenKhoaHoc());
13         }
14     }
15 }
```



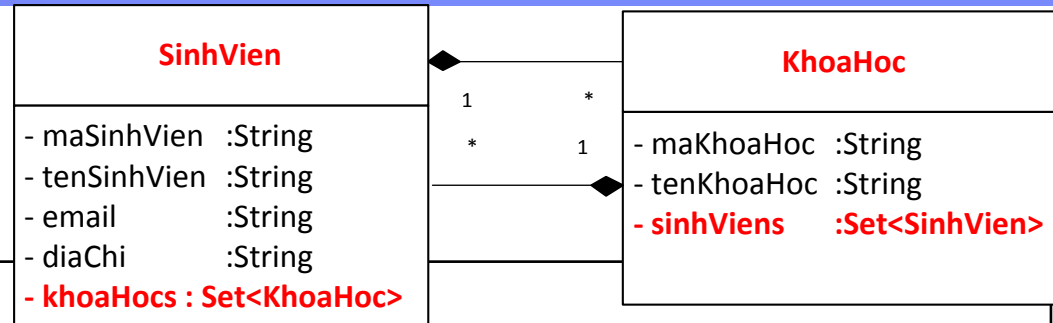
# SinhVienDAO – lấy thông tin sinh viên

Variables		Breakpoints	Output
Name	Type	Value	
sv	SinhVien	#1407	
maSinhVien	String	"0312143"	
tenSinhVien	String	"Nguyễn Hoàng Anh"	
email	String	"hoanganhis@gmail.com"	
diaChi	String	"227 Nguyễn Văn Cừ Q5"	
khoahocs	PersistentSet	"size = 2"	
[0]	KhoaHoc	#1430	
[1]	KhoaHoc	#1431	

**Lazy = false**

Variables	Breakpoints
INFO: Not binding factory to JNDI, no JNDI name configured	
0312143	
Nguyễn Hoàng Anh	
hoanganhis@gmail.com	
227 Nguyễn Văn Cừ Q5	
KHJAVA	
J2EE	
KHNET	
.NET 3.5	
BUILD SUCCESSFUL (total time: 2 seconds)	

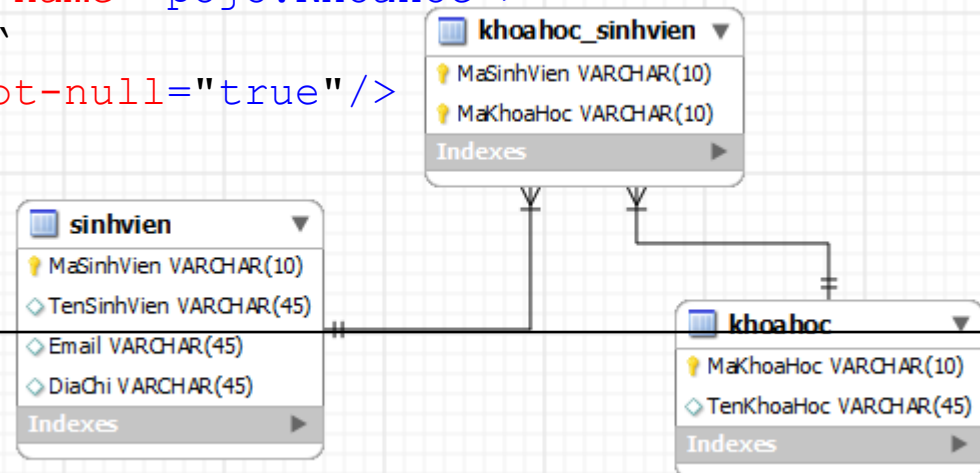
# SinhVien.hbm.xml



```

1 <hibernate-mapping>
2   <class catalog="quanlydangkyhocphan" name="pojo.SinhVien"
3   table="sinhvien">
4     .
5     .
6     .
7     <set inverse="false" name="khoahocs"
8     table="khoahoc_sinhvien">
9       <key>
10        <column length="10" name="MaSinhVien" not-null="true"/>
11      </key>
12      <many-to-many entity-name="pojo.KhoaHoc">
13        <column length="10"
14        name="MaKhoaHoc" not-null="true"/>
15      </many-to-many>
16    </set>
17  </class>
18 </hibernate-mapping>

```



# SinhVienDAO – lấy thông tin sinh viên

```
1  public class SinhVienDAO {
2      public static SinhVien layThongTinSinhVien(String maSinhVien)
3      {
4          SinhVien sv = null;
5          Session session = HibernateUtil.getSessionFactory()
6                                  .openSession();
7
8          try {
9              String hql= " select sv ";
10             hql += "from SinhVien sv left join fetch sv.khoahocs";
11             hql += " where sv.maSinhVien=:maSinhVien";
12             Query query = session.createQuery(hql);
13             query.setString("maSinhVien", maSinhVien);
14             sv = (SinhVien) query.uniqueResult();
15         } catch (HibernateException ex) {
16             System.err.println(ex);
17         } finally {
18             session.close();
19         }
20     }
21     return sv;
22 }
```

# SinhVienDAO – lấy thông tin sinh viên

```
1  public class Main {
2      public static void main(String[] args) {
3          SinhVien sv=SinhVienDAO.layThongTinSinhVien("0312143");
4          System.out.println(sv.getMaSinhVien());
5          System.out.println(sv.getTenSinhVien());
6          System.out.println(sv.getEmail());
7          System.out.println(sv.getDiaChi());
8          Iterator<KhoaHoc> khoaHocs=sv.getKhoahocs().iterator();
9          while(khoaHocs.hasNext()){
10             KhoaHoc kh=khoaHocs.next();
11             System.out.println(kh.getMaKhoaHoc());
12             System.out.println(kh.getTenKhoaHoc());
13         }
14     }
15 }
```

# SinhVienDAO – lấy thông tin sinh viên

Variables		Breakpoints	Output
Name	Type	Value	
sv	SinhVien	#1407	
maSinhVien	String	"0312143"	
tenSinhVien	String	"Nguyễn Hoàng Anh"	
email	String	"hoanganhis@gmail.com"	
diaChi	String	"227 Nguyễn Văn Cừ Q5"	
khoahocs	PersistentSet	"size = 2"	
[0]	KhoaHoc	#1430	
[1]	KhoaHoc	#1431	

HQL

Variables	Breakpoints
INFO: Not binding factory to JNDI, no JNDI name configured	
0312143	
Nguyễn Hoàng Anh	
hoanganhis@gmail.com	
227 Nguyễn Văn Cừ Q5	
KHJAVA	
J2EE	
KHNET	
.NET 3.5	
BUILD SUCCESSFUL (total time: 2 seconds)	

# SinhVienDAO – thêm sinh viên

```
1 public class SinhVienDAO {
2     public static boolean themSinhVien(SinhVien sv) {
3         Session session = HibernateUtil.getSessionFactory().openSession();
4         if (SinhVienDAO.layThongTinSinhVien(sv.getMaSinhVien()) != null) {
5             return false;
6         }
7         boolean kq = true;
8         Transaction transaction = null;
9         try {
10             transaction = session.beginTransaction();
11             session.saveOrUpdate(sv);
12             transaction.commit();
13         } catch (HibernateException ex) {
14             transaction.rollback();
15             System.err.println(ex);
16             kq = false;
17         } finally {
18             session.close();
19         }
20         return kq;
21     }
22 }
```

# SinhVienDAO – thêm sinh viên

```
1  public class Main {  
2  
3      public static void main(String[] args) {  
4          SinhVien sv=new SinhVien();  
5          sv.setMaSinhVien("0312078");  
6          sv.setTenSinhVien("Hoàng Minh Châu");  
7          sv.setDiaChi("231 Phan Đăng Lưu Bình Thạnh");  
8          sv.setEmail("hoangminh@gmail.com");  
9          KhoaHoc kh=new KhoaHoc();  
10         kh.setMaKhoaHoc("KHNET");  
11         kh.setTenKhoaHoc(".NET");  
12         sv.getKhoahocs().add(kh);  
13         SinhVienDAO.themSinhVien(sv);  
14     }  
15 }
```

# SinhVienDAO – thêm sinh viên

	MaSinhVien	TenSinhVien	Email	DiaChi
▶	0312055	Cao Xuân Nam	cxnam@gmail.com	228 Vườn Lài Tân Phú
	0312078	Hoàng Minh Châu	hoangminh@gmail.com	231 Phan Đăng Lưu - Bình Thạnh
	0312079	Hoàng Vũ Tuấn	hvtuan999@yahoo.com	213 Phú Bình Q11
	0312143	Nguyễn Hoàng Anh	hoanganhis@gmail.com	227 Nguyễn Văn Cừ Q5

	MaSinhVien	MaKhoaHoc
▶	0312055	KHJAVA
	0312055	KHPHP
	0312078	KHNET
	0312079	KHJAVA
	0312079	KHNET
	0312143	KHJAVA
	0312143	KHNET



# Cascade

- save-update
- delete

## Tài liệu tham khảo

- Nguyễn Hoàng Anh, Tập bài giảng và video môn chuyên đề Java, 2010
- Gary Mak, Tập hướng dẫn từng bước Hibernate, 2006



# HỎI VÀ ĐÁP