

# CHUYÊN ĐỀ JAVA

## GIỚI THIỆU HIBERNATE

Nguyễn Hoàng Anh

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

[hoanganhis@gmail.com](mailto:hoanganhis@gmail.com)

ĐH KHTN, 2011

# Nội dung trình bày

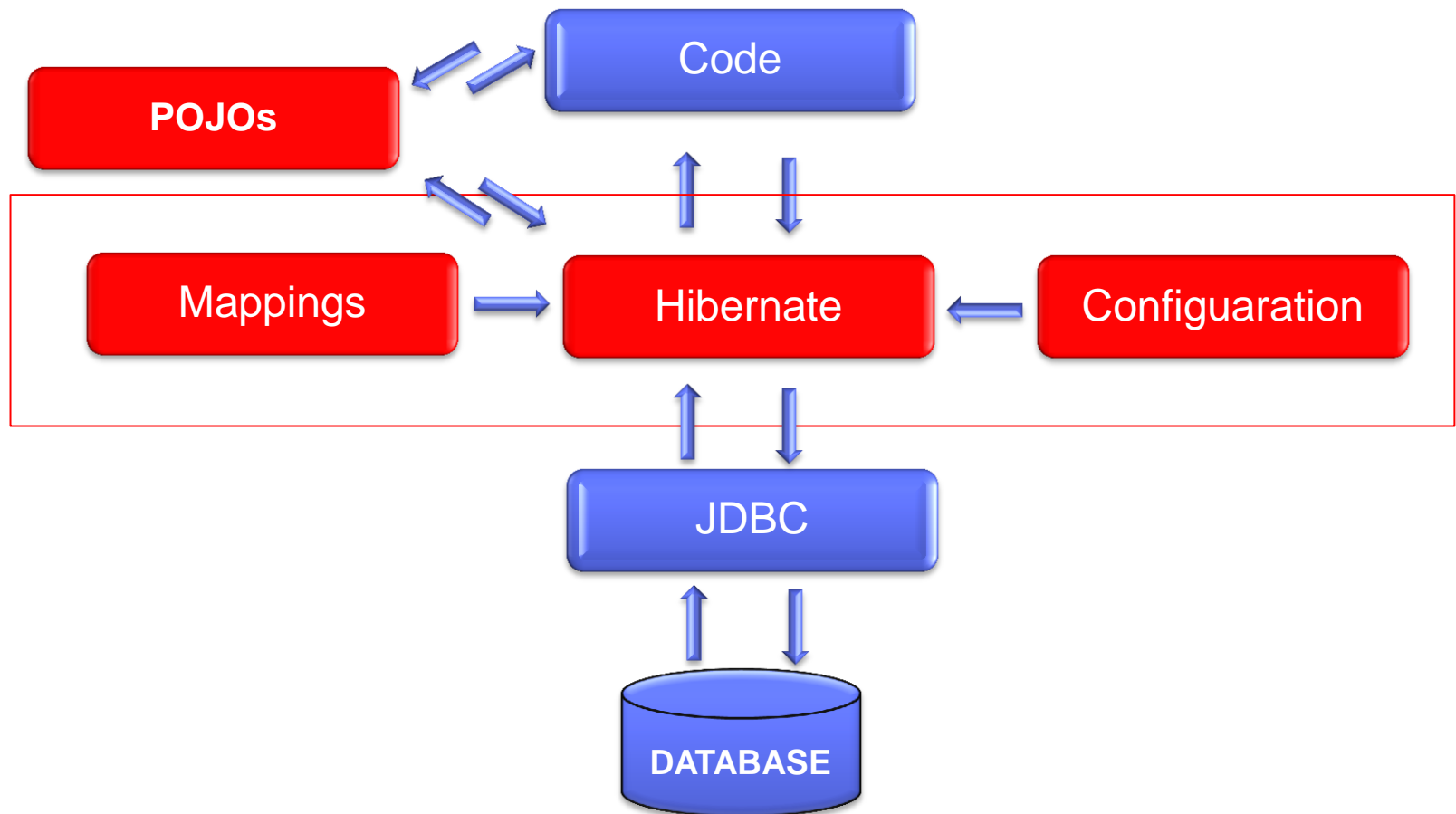
- Giới thiệu Hibernate
- Xây dựng ứng dụng quản lý sinh viên đơn giản với Hibernate
  - Lấy danh sách sinh viên
  - Lấy thông tin sinh viên
  - Thêm sinh viên
  - Cập nhật sinh viên
  - Xóa sinh viên

# Hibernate

- Cơ sở dữ liệu thường được thiết kế và lưu trữ theo hướng quan hệ.
- Tuy nhiên phần mềm thường được xây dựng theo hướng đối tượng.
- Đối với lập trình viên khi xây dựng phần mềm thường muốn làm việc với các đối tượng và không phải nhớ đến các dòng , các cột trong các bảng của cơ sở dữ liệu.

# Cài đặt

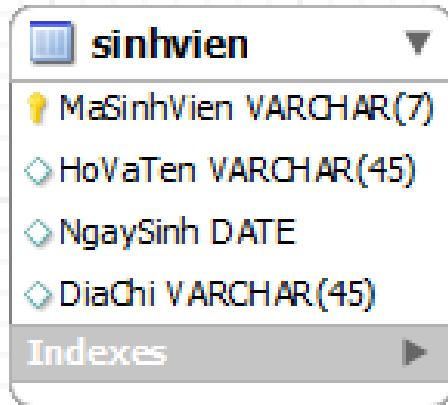
# Kiến trúc Hibernate



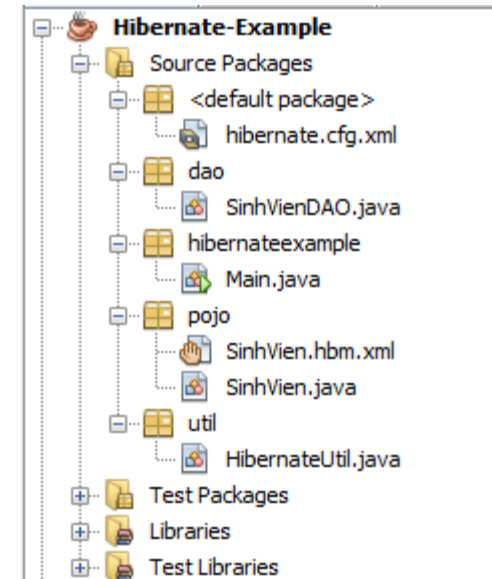
# Các bước sử dụng hibernate

- Bước 1: Tạo cơ sở dữ liệu
- Bước 2: Tạo các POJO
- Bước 3: Tạo file cấu hình hibernate.cfg.xml
- Bước 4: Tạo các file mapping <POJO>.hbm.xml
- Bước 5: Khai báo các file mapping vào hibernate.cfg.xml
- Bước 6: Xây dựng lớp HibernateUtil
- Bước 7: Xây dựng các DAO & Sử dụng

# Bước 1: Tạo cơ sở dữ liệu



**CSDL: MySQL  
QuanLySinhVien**



	MaSinhVien	HoVaTen	NgaySinh	DiaChi
▶	0312055	Cao Xuân Nam	1985-06-22	208 Vườn Lài Phú Thọ Hòa Tân Phú
	0312078	Hoàng Minh Châu	1985-05-13	Trảng Bom - Đồng Nai
	0312079	Hoàng Vũ Tuấn	1985-01-01	121 Phú Bình Q11
	0312137	Lê Hoài Tâm	1985-04-15	211 - Bình Minh - Vĩnh Long
	0312143	Nguyễn Hoàng Anh	1985-06-13	227 Nguyễn Văn Cừ Q5

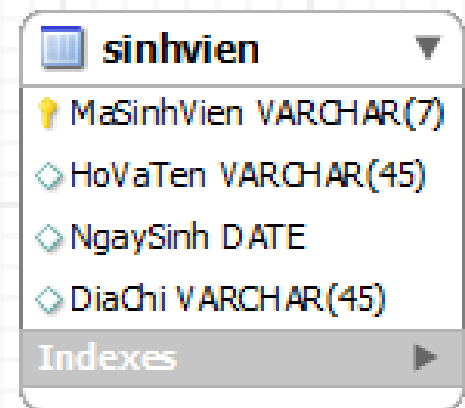
## Bước 2: Tạo POJO SinhVien

```
1 package pojo;
2 import java.util.Date;
3 public class SinhVien implements java.io.Serializable {
4     private String maSinhVien;
5     private String hoVaTen;
6     private Date ngaySinh;
7     private String diaChi;
8     public SinhVien() {
9     }
10    public SinhVien(String maSinhVien) {
11        this.maSinhVien = maSinhVien;
12    }
13    public SinhVien(String maSinhVien, String hoVaTen, Date
14 ngaySinh, String diaChi) {
15        this.maSinhVien = maSinhVien;
16        this.hoVaTen = hoVaTen;
17        this.ngaySinh = ngaySinh;
18        this.diaChi = diaChi;
19    }
```



## Bước 2: Tạo POJO SinhVien

```
1      public String getMaSinhVien() {
2          return this.maSinhVien;
3      } public void setMaSinhVien(String maSinhVien) {
4          this.maSinhVien = maSinhVien;
5      } public String getHoVaTen() {
6          return this.hoVaTen;
7      } public void setHoVaTen(String hoVaTen) {
8          this.hoVaTen = hoVaTen;
9      } public Date getNgaySinh() {
10         return this.ngaySinh;
11     } public void setNgaySinh(Date ngaySinh) {
12         this.ngaySinh = ngaySinh;
13     } public String getDiaChi() {
14         return this.diaChi;
15     } public void setDiaChi(String diaChi) {
16         this.diaChi = diaChi;
17     }
18 }
```



## Bước 3: Tạo file cấu hình hibernate.cfg.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//
3 //Hibernate/Hibernate Configuration DTD 3.0//EN"
4 "http://hibernate.sourceforge.net/hibernate-configuration-
5 3.0.dtd">
6 <hibernate-configuration>
7     <session-factory>
8         . . .
9     </session-factory>
10 </hibernate-configuration>
```

## Bước 3: Tạo file cấu hình hibernate.cfg.xml

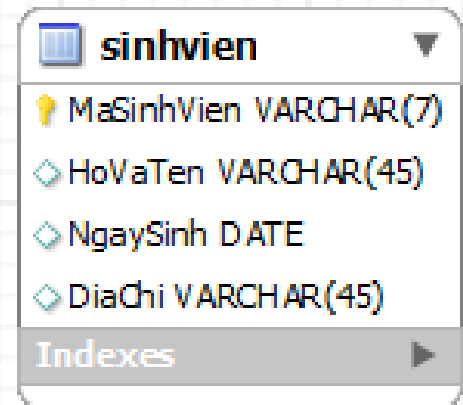
```
1 <session-factory>
2   <property
3     name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
4     <property name="hibernate.connection.driver_class">
5       com.mysql.jdbc.Driver
6     </property>
7     <property name="hibernate.connection.url">
8       jdbc:mysql://localhost:3306/QuanLySinhVien?autoReconnect&
9         useUnicode=true&characterEncoding=UTF-8
10    </property>
11    <property name="hibernate.connection.username">root</property>
12    <property name="hibernate.connection.password">root</property>
13    <mapping resource="pojo/SinhVien.hbm.xml"/>
14    <!--Mapping Resources-->
15  </session-factory>
```

## Bước 4: Tạo các file mapping SinhVien.hbm.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate
3 Mapping DTD 3.0//EN"
4 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
5 <hibernate-mapping>
6     ...
7 </hibernate-mapping>
```

## Bước 4: Tạo file mapping SinhVien.hbm.xml

```
1 <class catalog="quanlyhocsinh"
2     name="pojo.SinhVien" table="sinhvien">
3     <id name="maSinhVien" type="string">
4         <column length="7" name="MaSinhVien"/>
5         <generator class="assigned"/>
6     </id>
7     <property name="hoVaTen" type="string">
8         <column length="45" name="HoVaTen"/>
9     </property>
10    <property name="ngaySinh" type="date">
11        <column length="10" name="NgaySinh"/>
12    </property>
13    <property name="diaChi" type="string">
14        <column length="45" name="DiaChi"/>
15    </property>
16 </class>
```



sinhvien	
MaSinhVien	VARCHAR(7)
HoVaTen	VARCHAR(45)
NgaySinh	DATE
DiaChi	VARCHAR(45)
Indexes	

## Bước 5: Khai báo mapping vào hibernate.cfg.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE hibernate-configuration PUBLIC "-//
3  //Hibernate/Hibernate Configuration DTD 3.0//EN"
4  "http://hibernate.sourceforge.net/hibernate-configuration-
5  3.0.dtd">
6  <hibernate-configuration>
7      <session-factory>
8          <!--Property-->
9          ...
10         <mapping resource="pojo/SinhVien.hbm.xml"/>
11     </session-factory>
12 </hibernate-configuration>
```

## Bước 6: Xây dựng lớp HibernateUtil

```
1  import org.hibernate.cfg.AnnotationConfiguration;
2  import org.hibernate.SessionFactory;
3  public class HibernateUtil {
4      private static final SessionFactory sessionFactory;
5      static {
6          try {
7              sessionFactory = new
8  AnnotationConfiguration().configure().buildSessionFactory();
9          } catch (Throwable ex) {
10             System.err.println("Initial SessionFactory
11  creation failed." + ex);
12             throw new ExceptionInInitializerError(ex);
13         }
14     }
15     public static SessionFactory getSessionFactory() {
16         return sessionFactory;
17     }
18 }
```

## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

```
1  public class SinhVienDAO {
2      public static List<SinhVien> layDanhSachSinhVien() {
3          List<SinhVien> ds = null;
4          Session session = HibernateUtil.getSessionFactory()
5                          .openSession();
6          try {
7              String hql = "select sv from SinhVien sv";
8              Query query = session.createQuery(hql);
9              ds = query.list();
10         } catch (HibernateException ex) {
11             //Log the exception
12             System.err.println(ex);
13         } finally {
14             session.close();
15         }
16         return ds;
17     }
18     . . .
```



## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

```
1 public class Main {
2
3     public static void main(String[] args) {
4         //<editor-fold defaultstate="collapsed" desc="1. Lấy
5 danh sách sinh viên">
6         List<SinhVien> ds=SinhVienDAO.layDanhSachSinhVien();
7         for(int i=0; i<ds.size(); i++){
8             SinhVien sv=ds.get(i);
9             System.out.println("MSSV: "+sv.getMaSinhVien());
10            System.out.println("Họ và tên: "+sv.getHoVaTen());
11            System.out.println("Ngày sinh: " +
12 sv.getNgaySinh());
13            System.out.println("Địa chỉ: "+ sv.getDiaChi());
14        }
15        //</editor-fold>
```

## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

```
1  public class SinhVienDAO {
2      public static SinhVien layThongTinSinhVien(String
3  maSinhVien) {
4          SinhVien sv = null;
5          Session session = HibernateUtil.getSessionFactory()
6                                  .openSession();
7          try {
8              sv = (SinhVien) session.get(SinhVien.class,
9                                          maSinhVien);
10         } catch (HibernateException ex) {
11             //Log the exception
12             System.err.println(ex);
13         } finally {
14             session.close();
15         }
16         return sv;
17     }
18     . . .
```

## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

```
1 public class Main {
2     public static void main(String[] args) {
3
4         //<editor-fold defaultstate="collapsed" desc="2. Lấy
5 thông tin sinh viên">
6         SinhVien sv =
7 SinhVienDAO.layThongTinSinhVien("0312143");
8         if(sv!=null) {
9             System.out.println("MSSV: " + sv.getMaSinhVien());
10            System.out.println("Họ và tên: " +
11 sv.getHoVaTen());
12            System.out.println("Ngày sinh: " +
13 sv.getNgaySinh());
14            System.out.println("Địa chỉ: " + sv.getDiaChi());
15        } else {
16            System.out.println("Sinh viên 0312143 không tồn
17 tại");
18        }
19        //</editor-fold>
```

## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

```
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          Transaction transaction = null;
8          try {
9              transaction = session.beginTransaction();
10             session.save(sv);
11             transaction.commit();
12         } catch (HibernateException ex) {
13             //Log the exception
14             transaction.rollback();
15             System.err.println(ex);
16         } finally {
17             session.close();
18         }
19         return true;
20     }
```

## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

```
1 public class Main {
2     public static void main(String[] args) {
3         //<editor-fold defaultstate="collapsed" desc="3. Thêm
4 sinh viên">
5         SinhVien sv = new SinhVien();
6         sv.setMaSinhVien("0312171");
7         sv.setHoVaTen("Tạ Tấn Thêm");
8         Calendar calendar = Calendar.getInstance();
9         calendar.set(1985, 5, 18);
10        Date d = calendar.getTime();
11        sv.setNgàySinh(d);
12        sv.setDiaChi("Vũng Liêm - Vĩnh Long");
13        boolean kq = SinhVienDAO.themSinhVien(sv);
14        if (kq == true) {
15            System.out.println("Thêm thành công");
16        } else {
17            System.out.println("Thêm thất bại");
18        }
19        //</editor-fold>
```

## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

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

## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

```
1 public class Main {
2     public static void main(String[] args) {
3         //<editor-fold defaultstate="collapsed" desc="4. Cập nhật
4 sinh viên">
5         SinhVien sv = SinhVienDAO.layThongTinSinhVien("0312143");
6         if (sv != null) {
7             sv.setHoVaTen("Tạ Tấn Thêm");
8             Calendar calendar = Calendar.getInstance();
9             calendar.set(1985, 5, 18);
10            Date d = calendar.getTime();
11            sv.setNgàySinh(d);
12            sv.setDiaChi("Vũng Liêm - Vĩnh Long");
13            boolean kq = SinhVienDAO.capNhatThongTinSinhVien(sv);
14            if (kq == true) {
15                System.out.println("Cập nhật thành công");
16            } else {
17                System.out.println("Cập nhật thất bại");
18            }
19        }
20        //</editor-fold>
```

## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

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



## Bước 7 : Xây dựng lớp SinhVienDAO & sử dụng

```
1  public class Main {
2
3      public static void main(String[] args) {
4
5          //<editor-fold defaultstate="collapsed" desc="5. Xóa
6          sinh viên">
7
8          boolean kq = SinhVienDAO.xoaSinhVien("0312143");
9          if (kq == true) {
10             System.out.println("Xóa thành công");
11         } else {
12             System.out.println("Xóa thất bại");
13         }
14     }
15     //</editor-fold>
16 }
17 }
```

## Tài liệu tham khảo

- Nguyễn Hoàng Anh, Tập bài giảng và video môn chuyên đề Java, 2010
- Marty Hall, Tập bài giảng về Servlet, 2010
  - <http://courses.coreservlets.com/Course-Materials/csajsp2.html>



# HỎI VÀ ĐÁP