



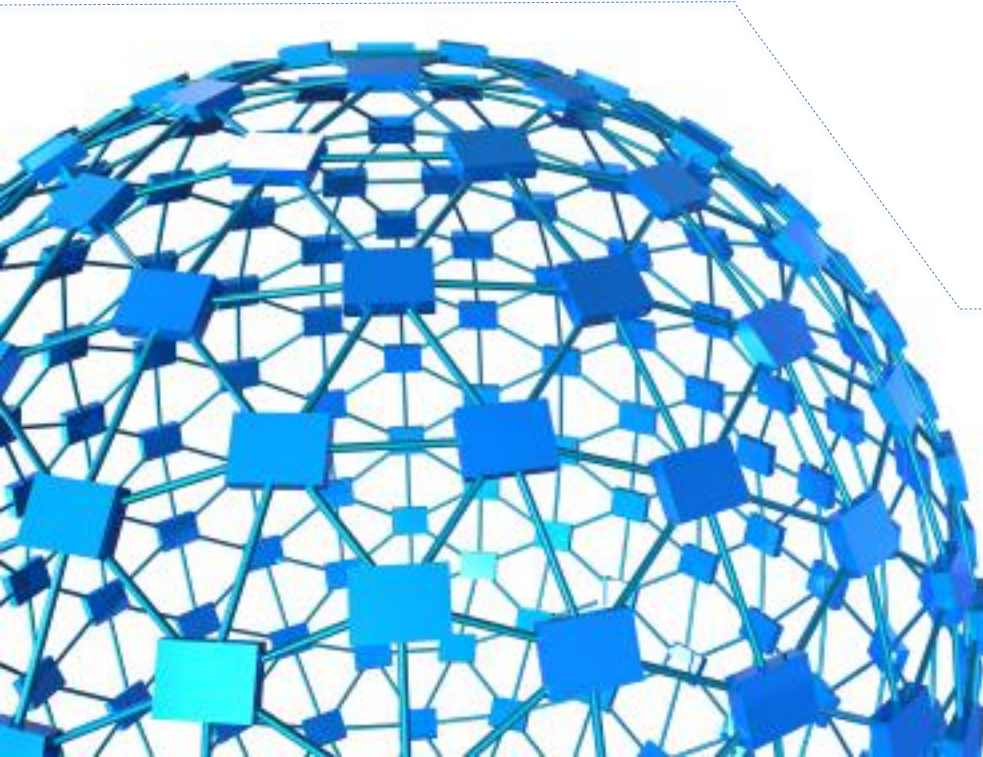
Write Once, Run Anywhere

SERVLET/JSP



BASIC HIBERNATE

ThS. Nguyễn Nghiệm
0913.745.789
nnghiem@yahoo.com





Nội dung-Cơ bản

☆ Giới Thiệu

- ✎ Ưu Điểm Của Hibernate
- ✎ Kiến Trúc Hibernate

☆ Chuẩn Bị Môi Trường và CSDL

- ✎ Cài Đặt Môi Trường
- ✎ Cấu Hình Nguồn Dữ Liệu
- ✎ Tập tin cấu hình
- ✎ Ánh Xạ
 - ✓ Lớp thực thể
 - ✓ Tập tin ánh xạ
 - ✓ Kiểu dữ liệu ánh xạ

☆ Lập Trình Hibernate

- ✎ Truy vấn
 - ✓ Truy vấn tất cả các thực thể
 - ✓ Truy vấn một thực thể theo khóa chính
- ✎ Thao tác dữ liệu
 - ✓ Thêm mới
 - ✓ Thêm mới hoặc cập nhật
 - ✓ Cập nhật
 - ✓ Xóa
 - ✓ Tải lại (làm tươi) một thực thể

- ★ Hibernate là một giải pháp ánh xạ đối tượng quan hệ trong Java và nó được xem như một framework quản lý lưu trữ mở.
- ★ Hibernate ánh xạ các lớp Java vào các bảng của CSDL quan hệ và kiểu dữ liệu Java với kiểu dữ liệu SQL.
- ★ Hibernate đứng ở giữa các đối tượng Java truyền thống và server CSDL để điều hành các công việc quản lý lưu trữ trạng thái của các đối tượng đó dựa trên cơ chế ánh xạ thích hợp.





Ưu điểm của Hibernate

- ★ Hibernate tập trung vào việc ánh xạ các lớp Java vào các bảng trong CSDL bằng XML hoặc annotation mà không cần phải viết mã Java.
- ★ Hibernate cung cấp API đơn giản để lưu trữ và truy xuất các đối tượng Java trực tiếp từ cơ sở dữ liệu.
- ★ Nếu có sự thay đổi trong cơ sở dữ liệu hoặc trong bất kỳ bảng nào, chúng ta chỉ cần thay đổi các thuộc tính trong tập tin XML hoặc annotation.
- ★ Trong suốt với SQL: chúng ta chỉ làm việc với các đối tượng Java quen thuộc.
- ★ Thao tác các đối tượng kết hợp một cách dễ dàng thông qua mối quan hệ giữa các thực thể.
- ★ Giảm thiểu sự truy cập cơ sở dữ liệu nhờ cơ chế xử lý thông minh.
- ★ Cung cấp kỹ thuật truy vấn dữ liệu đơn giản.
- ★ Công việc này sẽ giảm bớt 95% nhiệm vụ của người lập trình lưu trữ dữ liệu.

★ Hibernate hỗ trợ hầu hết các hệ quản trị CSDL quan hệ chính hiện tại.

✗ HSQL Database Engine

✗ DB2/NT

✗ MySQL

✗ PostgreSQL

✗ FrontBase

✗ Oracle

✗ **Microsoft SQL Server Database**

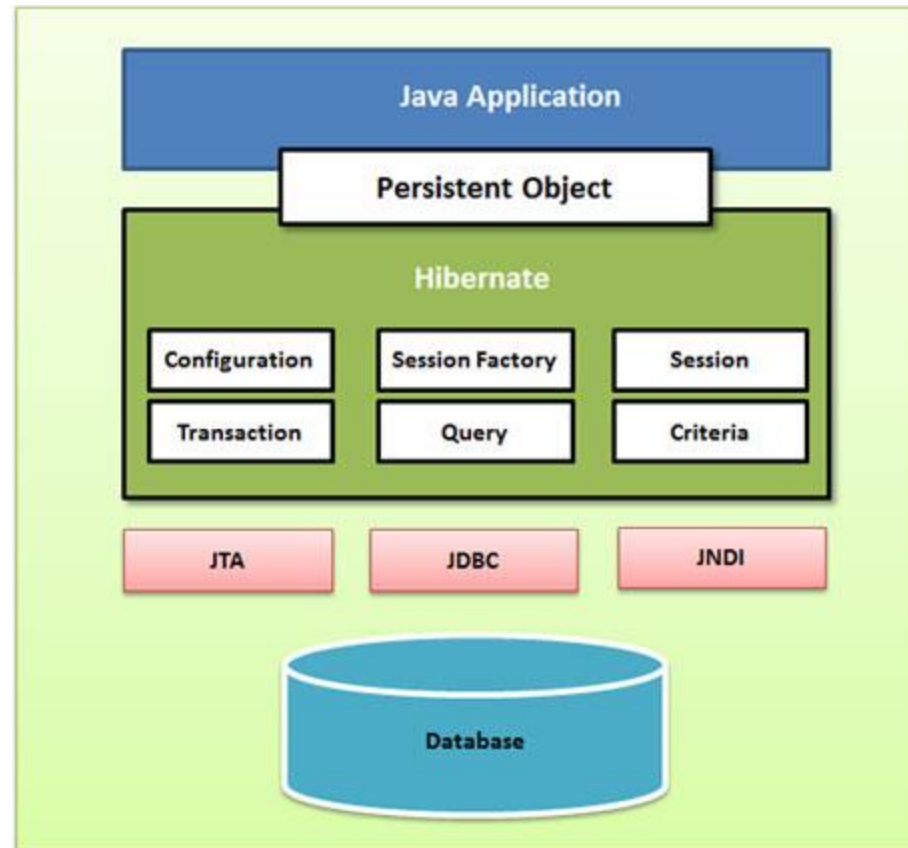
✗ Sybase SQL Server

✗ Informix Dynamic Server



Kiến trúc Hibernate

- ★ Kiến trúc Hibernate được phân tầng để bạn không cần phải quan tâm về API phía dưới. Hibernate tạo ra CSDL và dữ liệu cấu hình để cung cấp cho các dịch vụ quản lý lưu trữ trạng thái đối với ứng dụng.





Các thành phần trong Hibernate

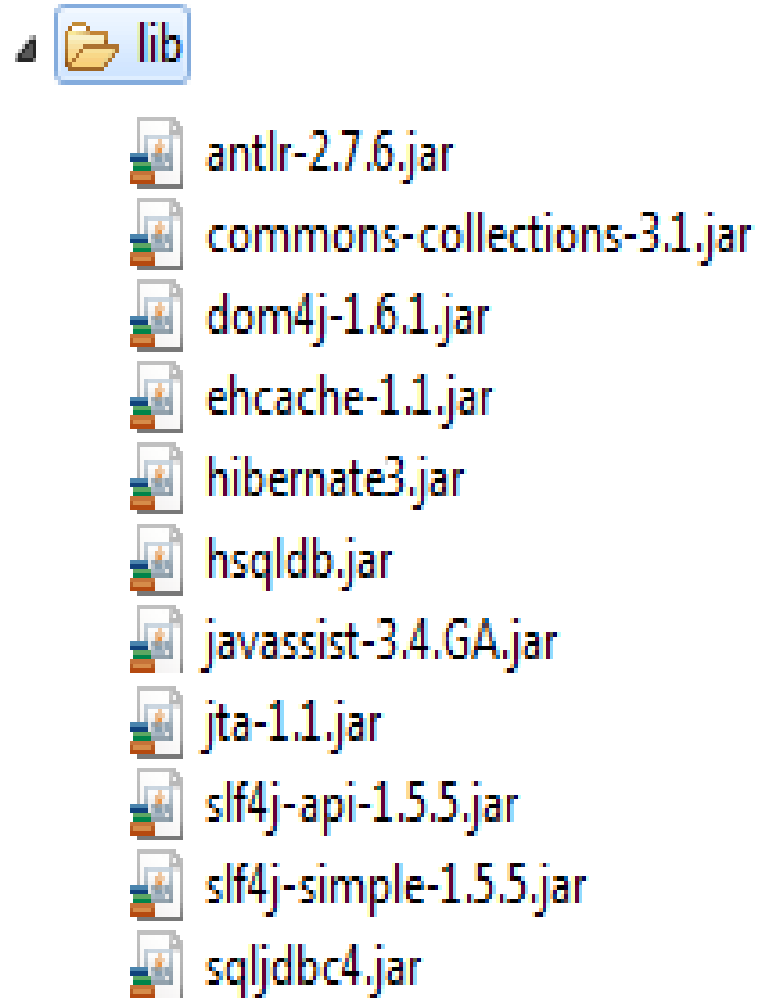
- ★ **Configuration**: được sử dụng để quản lý thông tin cấu hình kết nối đến CSDL và ánh xạ thực thể vào CSDL.
- ★ **SessionFactory**: đối tượng này cho phép sản sinh ra nhiều phiên làm (session) việc khác nhau.
- ★ **Session**: sử dụng để tạo một phiên làm việc với CSDL thông qua Hibernate.
- ★ **Transaction**: sử dụng để điều khiển transaction (none or all)
- ★ **Query**: Đối tượng này được sử dụng để thực hiện truy vấn dữ liệu
- ★ **Criteria**: Đối tượng này được sử dụng để xây dựng câu lệnh truy vấn bằng lập trình thay cho truy vấn bằng câu lệnh HQL hay SQL.



Cài đặt môi trường

★ Trong số các thư viện trên, **sqljdbc4.jar** được sử dụng để làm việc với SQL Server 2000/2005/2008 của Microsoft.

★ Nếu bạn muốn làm việc với một CSDL cụ thể nào đó (Oracle, MySQL, DB2...) thì phải bổ sung thư viện JDBC tương ứng với CSDL đó vào ứng dụng của bạn.



★ CSDL mẫu được sử dụng để viết ứng dụng với Hibernate có cấu trúc sau và được đặt tên là **Java**

Categories					
	Column Name	Condensed Type	Nullable	Identity	Description
🔑	Id	char(3)	No	<input type="checkbox"/>	Mã loại
	Name	nvarchar(50)	No	<input type="checkbox"/>	Tên loại
	Description	nvarchar(255)	Yes	<input type="checkbox"/>	Mô tả
				<input type="checkbox"/>	

Products					
	Column Name	Condensed Type	Nullable	Identity	Description
🔑	Id	int	No	<input checked="" type="checkbox"/>	Mã hàng hóa
	Name	nvarchar(50)	No	<input type="checkbox"/>	Tên hàng hóa
	UnitPrice	float	No	<input type="checkbox"/>	Đơn giá
	ProductDate	datetime	No	<input type="checkbox"/>	Ngày sản xuất
	Image	nvarchar(50)	No	<input type="checkbox"/>	Hình ảnh
	Description	nvarchar(4000)	Yes	<input type="checkbox"/>	Mô tả
	CategoryId	char(3)	No	<input type="checkbox"/>	Mã loại
	Available	bit	No	<input type="checkbox"/>	Vẫn còn hàng
				<input type="checkbox"/>	

Courses					
	Column Name	Condensed Type	Nullable	Identity	Description
🔑	Id	int	No	<input checked="" type="checkbox"/>	Mã khóa học
	Name	nvarchar(50)	No	<input type="checkbox"/>	Tên khóa học
	Schoolfee	float	No	<input type="checkbox"/>	Học phí
	NoOfLearners	int	No	<input type="checkbox"/>	Số học viên
	StartDate	datetime	No	<input type="checkbox"/>	Ngày khai giảng
	Finished	bit	No	<input type="checkbox"/>	Kết thúc?
				<input type="checkbox"/>	



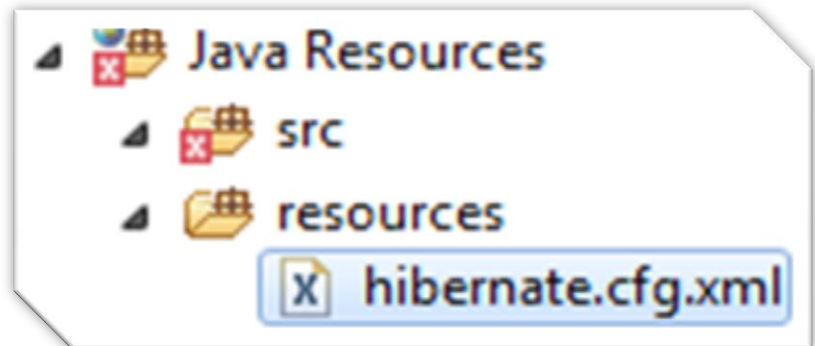
Cấu hình DataSource

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
<session-factory>
    <property name="hibernate.connection.driver_class">
        com.microsoft.sqlserver.jdbc.SQLServerDriver</property>
    <property name="hibernate.connection.url">
        jdbc:sqlserver://localhost:1433;DatabaseName=Java;</property>
    <property name="hibernate.connection.username">sa</property>
    <property name="connection.password">songlong</property>
    <property name="connection.pool_size">100</property>
    <property name="show_sql">>false</property>
    <property name="hbm2ddl.auto">none</property>
    <property name="hibernate.dialect">
        org.hibernate.dialect.SQLServerDialect</property>

    <!-- Mapping files -->
    <mapping resource="Course.hbm.xml"/>
    <mapping resource="Category.hbm.xml"/>
    <mapping resource="Product.hbm.xml"/>
</session-factory>

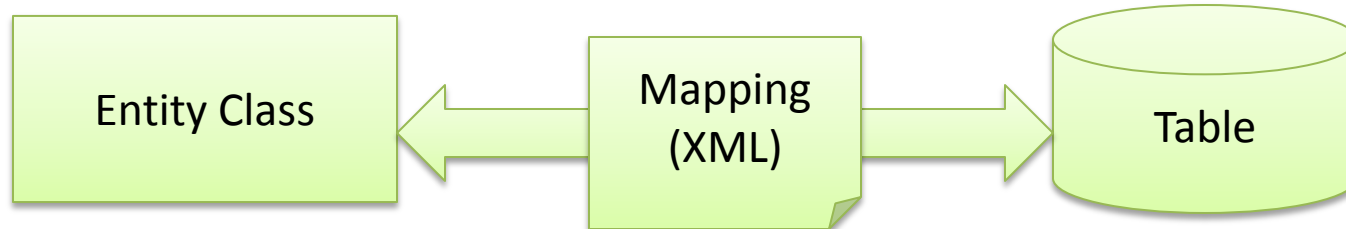
</hibernate-configuration>
```





Thuộc tính cấu hình

@name	Mô tả
hibernate.connection.driver_class	JDBC driver
hibernate.connection.url	Chuỗi kết nối đến CSDL
hibernate.connection.username	Mã đăng nhập CSDL
connection.password	Mật khẩu đăng nhập CSDL
connection.pool_size	Số kết nối tối đa trong hồ
show_sql	Xuất câu lệnh SQL khi truy vấn hoặc thao tác dữ liệu
hbm2ddl.auto	Chỉ ra cơ chế tự định nghĩa CSDL, thường dùng <ul style="list-style-type: none">✓ create-drop: tự tạo CSDL✓ none: làm việc với CSDL đã tồn tại
hibernate.dialect	Chỉ ra phương pháp làm việc với CSDL của Hibernate (xem bảng bên dưới để khai báo cho đúng)



★ Entity Class: lớp mô phỏng thực thể dùng để ánh xạ vào các cột của một bảng trong CSDL. Lớp phải thỏa qui ước JavaBean

~~✗~~ **Constructor mặc định**

~~✗~~ **getXyz()/setXyz()**

★ Mapping File: tập tin XML dùng để ánh xạ giữa lớp thực thể với một bảng trong CSDL. Tên của tập tin ánh xạ tuân theo: **<tên thực thể>.hbm.xml**.

★ Ví dụ:

~~✗~~ **Course.hbm.xml**

~~✗~~ Category.hbm.xml

~~✗~~ Product.hbm.xml



Entity Class

```
package nnghiem.entity;

import java.util.Date;

public class Course {

    protected int id;
    protected String name;
    protected double schoolfee;
    protected int noOfLearners;
    protected Date startDate;
    protected boolean finished;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```



Mapping file

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
  <class name="nnghiem.entity.Course" table="Courses">
    <id name="id" type="int" column="Id">
      <generator class="native" />
    </id>
    <property name="name" type="string" column="Name" />
    <property name="schoolfee" type="double" column="Schoolfee" />
    <property name="noOfLearners" type="int" column="NoOfLearners" />
    <property name="startDate" type="date" column="StartDate" />
    <property name="finished" type="boolean" column="Finished" />
  </class>
</hibernate-mapping>
```



Các thẻ ánh xạ

- ★ **<hibernate-mapping>**: là thẻ gốc, chứa nhiều thẻ <class>
- ★ **<class>**: ánh xạ một lớp thực thể và một bảng
 - ✎ **@name**: chỉ ra tên lớp thực thể
 - ✎ **@table**: chỉ ra tên bảng
- ★ **<id>**: ánh xạ thuộc tính của lớp thực thể với cột khóa chính
 - ✎ **@name**: tên thuộc tính
 - ✎ **@type**: kiểu của thuộc tính
 - ✎ **@column**: cột được ánh xạ trong bảng
 - ✎ **<generator>**: chỉ ra cách sinh giá trị tự tăng
- ★ **<property>**: ánh xạ thuộc tính của lớp thực thể với cột thông thường
 - ✎ **@name**: tên thuộc tính
 - ✎ **@type**: kiểu của thuộc tính
 - ✎ **@column**: cột được ánh xạ trong bảng





Kiểu ánh xạ

Mapping type	Java type	ANSI SQL Type
integer	int or java.lang.Integer	INTEGER
long	long or java.lang.Long	BIGINT
short	short or java.lang.Short	SMALLINT
float	float or java.lang.Float	FLOAT
double	double or java.lang.Double	DOUBLE
big_decimal	java.math.BigDecimal	NUMERIC
character	java.lang.String	CHAR(1)
string	java.lang.String	VARCHAR
byte	byte or java.lang.Byte	TINYINT
boolean	boolean or java.lang.Boolean	BIT
yes/no	boolean or java.lang.Boolean	CHAR(1) ('Y' or 'N')
true/false	boolean or java.lang.Boolean	CHAR(1) ('T' or 'F')
date	java.util.Date or java.sql.Date	DATE
time	java.util.Date or java.sql.Time	TIME
timestamp	java.util.Date or java.sql.Timestamp	TIMESTAMP
calendar	java.util.Calendar	TIMESTAMP
calendar_date	java.util.Calendar	DATE








Lập trình Hibernate

★ Truy vấn đơn giản

-  Truy vấn tất cả các thực thể
-  Truy vấn 1 thực thể

★ Thao tác

-  Thêm mới thực thể
-  Thêm mới hoặc cập nhật
-  Cập nhật thực thể
-  Xóa thực thể
-  Làm tươi (tải lại) thực thể



Truy vấn tất cả các thực thể

```
Configuration config = new Configuration().configure();  
SessionFactory factory = config.buildSessionFactory();  
Session session = factory.openSession();
```

```
String hql = "FROM Course";  
Query query = session.createQuery(hql);  
List<Course> courses = query.list();
```

```
for (Course course : courses) {  
    String name = course.getName();  
    double fee = course.getSchoolfee();  
    System.out.println("+ Name of course: " + name);  
    System.out.println("+ Schoolfee of course: " + fee);  
}
```



Truy vấn một thực thể theo khóa chính

```
Configuration config = new Configuration().configure();  
SessionFactory factory = config.buildSessionFactory();  
Session session = factory.openSession();
```

```
Course course = (Course) session.load(Course.class, 1);  
String name = course.getName();  
double fee = course.getSchoolfee();  
System.out.println("+ Name of course: " + name);  
System.out.println("+ Schoolfee of course: " + fee);
```



Thao tác thực thể (TTT)

```
Configuration config = new Configuration().configure();  
SessionFactory factory = config.buildSessionFactory();  
Session session = factory.openSession();
```

```
Transaction transaction = session.beginTransaction();
```

```
try{
```

```
    ...mã thao tác thực thể...
```

```
    /*
```

```
     * Chấp nhận kết quả
```

```
    */
```

```
    transaction.commit();
```

```
}
```

```
catch (Exception e) {
```

```
    /*
```

```
     * Hủy bỏ kết quả
```

```
    */
```

```
    transaction.rollback();
```

```
}
```



TTT-Thêm mới thực thể

```
Configuration config = new Configuration().configure();  
SessionFactory factory = config.buildSessionFactory();  
Session session = factory.openSession();
```



```
Transaction transaction = session.beginTransaction();
```

```
try{
```

```
    Course entity = new Course();  
    entity.setName("Clocks 2012");  
    entity.setSchoolfee(123.9);  
    entity.setNoOfLearners(25);  
    entity.setStartDate(new Date());  
    entity.setFinished(false);
```

```
    session.save(entity);
```

```
    transaction.commit();
```

```
}
```

```
catch (Exception e) {
```

```
    transaction.rollback();
```

```
}
```



TTT-Thêm mới hoặc cập nhật

```
Configuration config = new Configuration().configure();
SessionFactory factory = config.buildSessionFactory();
Session session = factory.openSession();

Transaction transaction = session.beginTransaction();
try{
    Course entity = new Course();
    entity.setId(1);
    entity.setName("Clocks 2012");
    entity.setSchoolfee(123.9);
    entity.setNoOfLearners(25);
    entity.setStartDate(new Date());
    entity.setFinished(false);
    session.saveOrUpdate(entity);
    transaction.commit();
}
catch (Exception e) {
    transaction.rollback();
}
```



TTT-Cập nhật thực thể

```
Configuration config = new Configuration().configure();
SessionFactory factory = config.buildSessionFactory();
Session session = factory.openSession();

Transaction transaction = session.beginTransaction();
try{
    Course entity = (Course) session.load(Course.class, 1);
    entity.setNoOfLearners(38);
    entity.setStartDate(new Date());
    session.update(entity);
    transaction.commit();
}
catch (Exception e) {
    transaction.rollback();
}
```



TTT-Xóa thực thể

```
Configuration config = new Configuration().configure();
SessionFactory factory = config.buildSessionFactory();
Session session = factory.openSession();

Transaction transaction = session.beginTransaction();
try{
    Course entity = new Course();
    entity.setId(1);
    session.delete(entity);
    transaction.commit();
}
catch (Exception e) {
    transaction.rollback();
}
```




TTT-Làm tươi thực thể

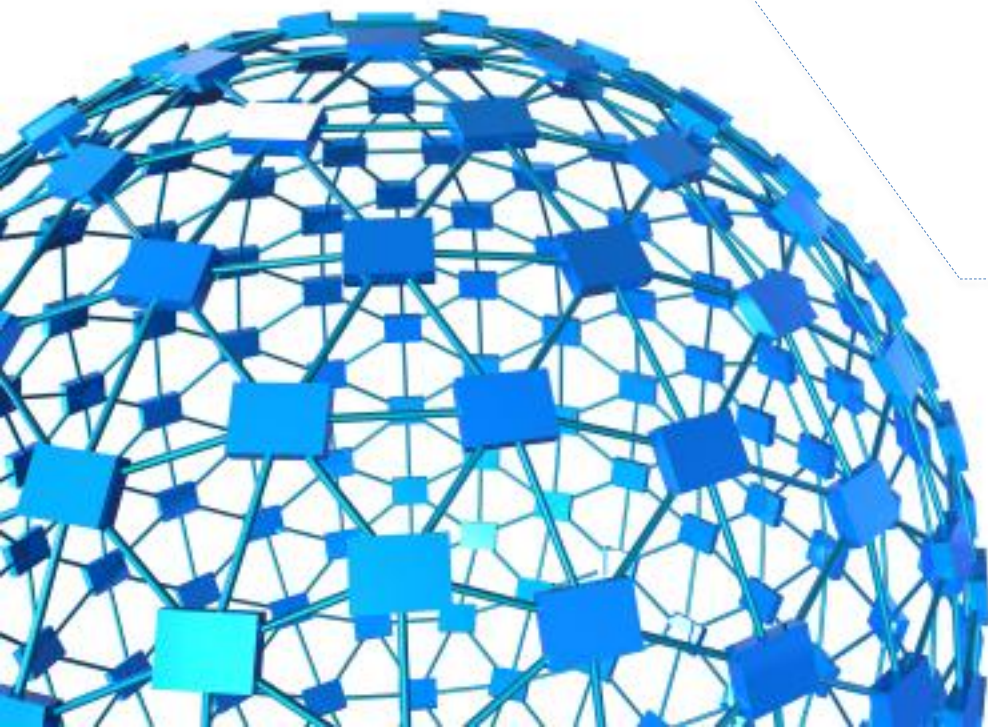
```
Configuration config = new Configuration().configure();
SessionFactory factory = config.buildSessionFactory();
Session session = factory.openSession();

Transaction transaction = session.beginTransaction();
try{
    Course entity = new Course();
    entity.setId(1);
    session.refresh(entity);
    transaction.commit();
}
catch (Exception e) {
    transaction.rollback();
}
```



Write Once, Run Anywhere

SERVLET/JSP



ADVANCED HIBERNATE

ThS. Nguyễn Nghiệm
0913.745.789
nnghiem@yahoo.com



Nội dung - Nâng cao

★ Truy vấn nâng cao

- ☒ Truy Vấn Nâng Cao
- ☒ Truy vấn một số thuộc tính
- ☒ Truy vấn một giá trị đơn
- ☒ Tham số
- ☒ Phân trang
- ☒ Truy vấn đặc thù

★ Thực Thể Kết Hợp

- ☒ Many2One
- ☒ One2Many

★ Ánh Xạ Bằng Annotations

- ☒ Thư viện bổ sung
- ☒ Các JPA annotations
- ☒ Thực thể ánh xạ

★ HQL - Ngôn Ngữ Truy Vấn Đối Tượng



Truy vấn một số thuộc tính

```
Configuration config = new Configuration().configure();
SessionFactory factory = config.buildSessionFactory();
Session session = factory.openSession();

/*
 * Truy vấn một số thuộc tính của thực thể
 */
String hql = "SELECT name, schoolfee FROM Course";
Query query = session.createQuery(hql);
List<Object[]> arrays = query.list();

for(Object[] array : arrays){
    String name = (String)array[0];
    double fee = (Double)array[1];
    System.out.println("+ Name of course: " + name);
    System.out.println("+ Schoolfee of course: " + fee);
}
```



Truy vấn một giá trị đơn

```
Configuration config = new Configuration().configure();  
SessionFactory factory = config.buildSessionFactory();  
Session session = factory.openSession();
```

```
/*
```

```
 * Truy vấn một giá trị đơn
```

```
*/
```

```
String hql = "SELECT AVG(schoolfee) FROM Course";
```

```
Query query = session.createQuery(hql);
```

```
double average = (Double)query.uniqueResult();
```

```
System.out.println("+ Average schoolfee of course: " + average);
```



Truy vấn có tham số

```
Configuration config = new Configuration().configure();
SessionFactory factory = config.buildSessionFactory();
Session session = factory.openSession();
/*
 * Truy vấn với tham số
 */
String hql = "FROM Course WHERE name LIKE :name "
            + " AND schoolfee BETWEEN :min AND :max";
Query query = session.createQuery(hql);
query.setParameter("name", "%asp%");
query.setParameter("min", 5.5);
query.setParameter("max", 200);
List<Course> courses = query.list();

for(Course course : courses){
    String name = course.getName();
    double fee = course.getSchoolfee();
    System.out.println("+ Name of course: " + name);
    System.out.println("+ Schoolfee of course: " + fee);
}
```



Truy vấn phân trang

```
Configuration config = new Configuration().configure();
SessionFactory factory = config.buildSessionFactory();
Session session = factory.openSession();

/*
 * Truy vấn có phân trang dữ liệu
 */
String hql = "FROM Course";
Query query = session.createQuery(hql);
query.setFirstResult(5); // vị trí bắt đầu
query.setMaxResults(10); // số lượng thực thể
List<Course> courses = query.list();

for (Course course : courses) {
    String name = course.getName();
    double fee = course.getSchoolfee();
    System.out.println("+ Name of course: " + name);
    System.out.println("+ Schoolfee of course: " + fee);
}
```



Truy vấn với SQL đặc thù

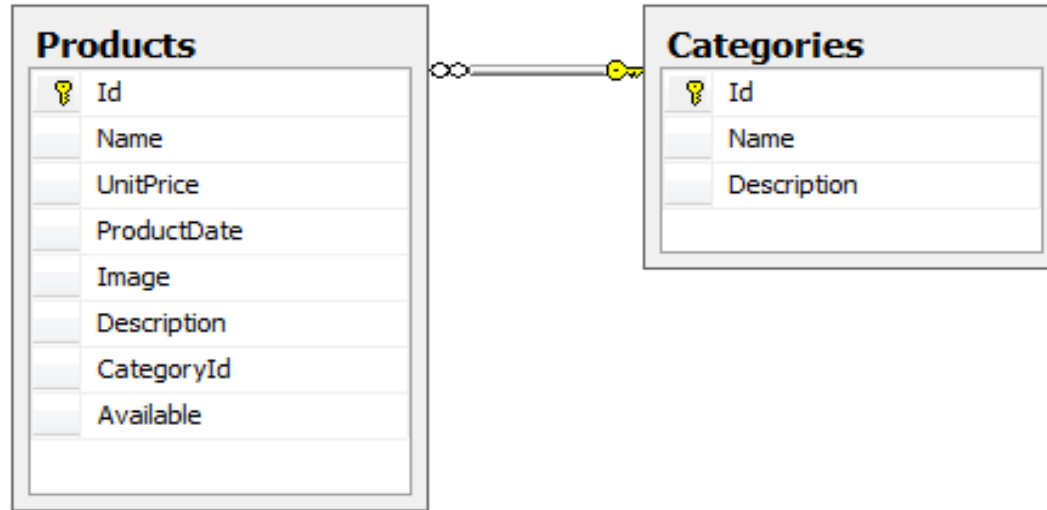
```
Configuration config = new Configuration().configure();
SessionFactory factory = config.buildSessionFactory();
Session session = factory.openSession();

/*
 * Truy vấn theo đặc thù ngôn ngữ SQL
 */
String sql = "SELECT * FROM Courses ORDER BY NewID()";
Query query = session.createQuery(SQLQuery(sql));
List<Course> courses = query.list();

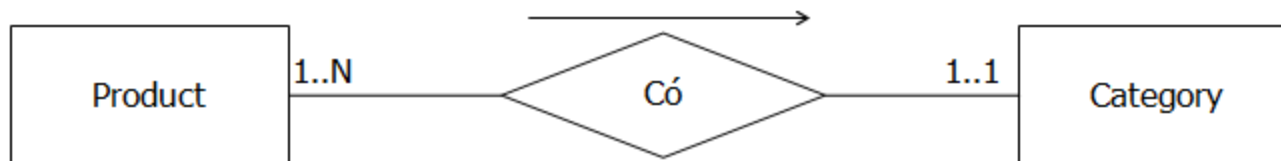
for(Course course : courses){
    String name = course.getName();
    double fee = course.getSchoolfee();
    System.out.println("+ Name of course: " + name);
    System.out.println("+ Schoolfee of course: " + fee);
}
```



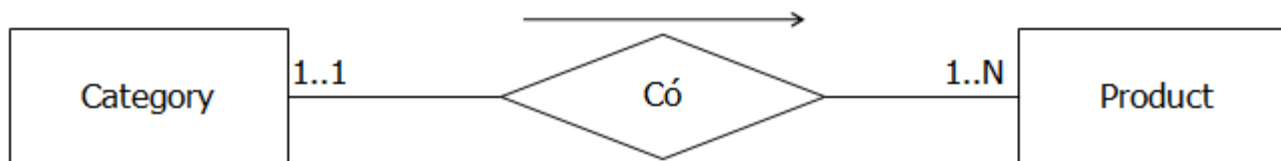

Thực thể kết hợp



Many-To-One



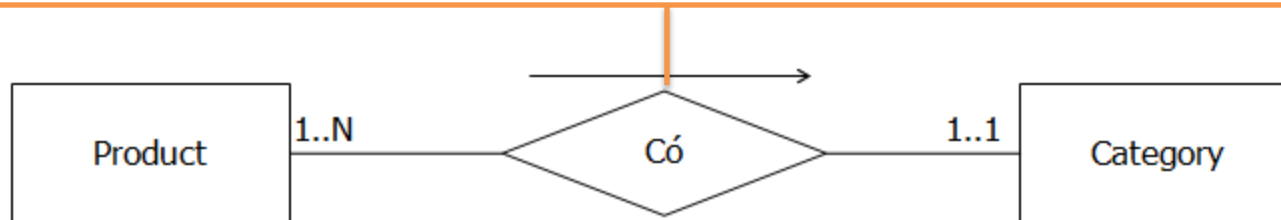
One-To-Many





Many-To-One

```
<many-to-one name="category" column="CategoryId"  
    class="nnghiem.entity.Category" cascade="all" not-null="true"/>
```



```
Category category; // thực thể kết hợp
```

```
public Category getCategory() {  
    return category;  
}
```

```
public void setCategory(Category category) {  
    this.category = category;  
}
```



Entity Class

```
package nnghiem.entity;

import java.util.Date;

public class Product {
    int id;
    String name, image, description;
    double unitPrice;
    Date productDate;
    boolean available;
    Category category; // thực thể kết hợp

    public Category getCategory() {
        return category;
    }
    public void setCategory(Category category) {
        this.category = category;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getImage() {
```



Mapping file

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
  <class name="nnghiem.entity.Product" table="Products">
    <id name="id" type="int" column="id">
      <generator class="native" />
    </id>
    <property name="name" type="string" column="Name" />
    <property name="unitPrice" type="double" column="UnitPrice" />
    <property name="productDate" type="date" column="ProductDate" />
    <property name="image" type="string" column="Image" />
    <property name="description" type="string" column="Description" />
    <property name="available" type="boolean" column="Available" />
    <many-to-one name="category" column="CategoryId"
      class="nnghiem.entity.Category" cascade="all" not-null="true"/>
  </class>
</hibernate-mapping>
```



Associated Entity Class - Category

```
package nnghiem.entity;

public class Category {
    String id, name, description;
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```



Tập tin cấu hình hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
<session-factory>
    <property name="hibernate.connection.driver_class">
        com.microsoft.sqlserver.jdbc.SQLServerDriver</property>
    <property name="hibernate.connection.url">
        jdbc:sqlserver://localhost:1433;DatabaseName=Java;</property>
    <property name="hibernate.connection.username">sa</property>
    <property name="connection.password">songlong</property>
    <property name="connection.pool_size">100</property>
    <property name="show_sql">>false</property>
    <property name="hbm2ddl.auto">none</property>
    <property name="hibernate.dialect">
        org.hibernate.dialect.SQLServerDialect</property>

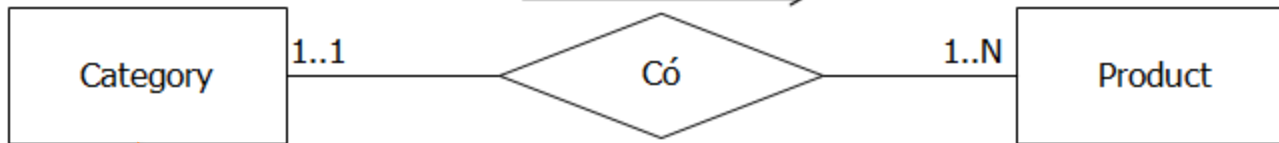
    <!-- Mapping files -->
    <mapping resource="Course.hbm.xml"/>
    <mapping resource="Category.hbm.xml"/>
    <mapping resource="Product.hbm.xml"/>
</session-factory>

</hibernate-configuration>
```



One-To-Many

```
<set name="products" cascade="all">  
  <key column="CategoryId"/>  
  <one-to-many class="nnghiem.entity.Product"/>  
</set>
```



```
Collection<Product> products; // các thực thể kết hợp  
  
public Collection<Product> getProducts() {  
    return products;  
}  
  
public void setProducts(Collection<Product> products) {  
    this.products = products;  
}
```



Entity Class

```
package nnghiem.entity;
```

```
public class Category {  
    String id, name, description;
```

```
    Collection<Product> products; // các thực thể kết hợp
```

```
    public Collection<Product> getProducts() {  
        return products;  
    }
```

```
    public void setProducts(Collection<Product> products) {  
        this.products = products;  
    }
```

```
    public String getId() {  
        return id;  
    }
```

```
    public void setId(String id) {  
        this.id = id;  
    }
```




Mapping file

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
    <class name="nnghiem.entity.Category" table="Categories">
        <id name="id" type="string" column="id"/>
        <property name="name" type="string" column="Name" />
        <property name="description" type="string" column="Description" />
        <set name="products" cascade="all">
            <key column="CategoryId"/>
            <one-to-many class="nnghiem.entity.Product"/>
        </set>
    </class>
</hibernate-mapping>
```



Associated Entity Class

```
package nnghiem.entity;

import java.util.Date;

public class Product {
    int id;
    String name, image, description;
    double unitPrice;
    Date productDate;
    boolean available;
    Category category; // thực thể kết hợp

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```



Ánh xạ bằng annotations

- ★ Sử dụng annotation để ánh xạ thực thể thay cho việc ánh xạ bằng XML.
- ★ Để sử dụng được với Annotations bạn cần bổ sung các thư viện sau đây
 - ~~✗~~ **hibernate-annotations.jar**
 - ~~✗~~ **hibernate-comons-annotations.jar**
 - ~~✗~~ **ejb3-persistence.jar**
- ★ Download phiên bản mới nhất ở đây
 - ~~✗~~ <http://sourceforge.net/projects/hibernate/files/latest/download?source=files>



```
@Entity
@Table(name="Courses")
public class Course {

    @Id
    @GeneratedValue
    @Column(name="Id")
    protected int id;

    @Column(name="Name")
    protected String name;

    @Column(name="Schoolfee")
    protected double schoolfee;

    @Column(name="NoOfLearners")
    protected int noOfLearners;

    @Temporal(TemporalType.DATE)
    @Column(name="StartDate")
    protected Date startDate;

    @Column(name="Finished")
    protected boolean finished;

    public int getId() {
        return id;
    }
    public void setId(int id) {
```



Annotations

Annotation	Mô tả	Ví dụ
@Entity	Chỉ ra class là thực thể	@Entity @Table(name="Courses") public class Course{...}
@Table	Ánh xạ lớp thực thể với bảng	
@Id	Chỉ ra cột khóa chính	@Id @GeneratedValue int id;
@GeneratedValue	Chỉ ra cột tự tăng	
@Column	Ánh xạ thuộc tính với cột	@Column(name = "Name") String name
@Temporal	Chỉ ra kiểu dữ liệu chuyển đổi	@Temporal(TemporalType.DATE) Date startDate
@ManyToOne	Ánh xạ quan hệ nhiều-1	@ManyToOne @JoinColumn(name="CategoryId") Category category;
@JoinColumn	Chỉ ra cột kết nối	
@OneToMany	Ánh xạ quan hệ 1-nhiều	@OneToMany(mappedBy="category") Collection<Product> products;



Associated Entity

```
@ManyToOne
```

```
@JoinColumn(name="CategoryId")
```

```
Category category;
```

```
@OneToMany(mappedBy="category")
```

```
Collection<Product> products;
```



Tập tin cấu hình hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
    <property name="hibernate.connection.driver_class">
        com.microsoft.sqlserver.jdbc.SQLServerDriver</property>
    <property name="hibernate.connection.url">
        jdbc:sqlserver://localhost:1433;DatabaseName=Java;</property>
    <property name="hibernate.connection.username">sa</property>
    <property name="connection.password">songlong</property>
    <property name="connection.pool_size">100</property>
    <property name="show_sql">>false</property>
    <property name="hbm2ddl.auto">none</property>
    <property name="hibernate.dialect">
        org.hibernate.dialect.SQLServerDialect</property>

    <!-- Mapping class -->
    <mapping class = "nnghiem.entity.Course"/>
    <mapping class = "nnghiem.entity.Category"/>
    <mapping class = "nnghiem.entity.Product"/>
</session-factory>
</hibernate-configuration>
```



Tải tập tin cấu hình Annotation

```
Configuration config = new AnnotationConfiguration().configure();  
SessionFactory factory = config.buildSessionFactory();  
Session session = factory.openSession();
```




★ FROM

- ✗ FROM Course
- ✗ FROM Course as c
- ✗ FROM Course c

★ SELECT

- ✗ SELECT name, schoolfee FROM Course
- ✗ SELECT c.name, c.schoolfee FROM Course c

★ WHERE

- ✗ FROM Course WHERE name LIKE 'Nguyễn %'
- ✗ FROM Course WHERE schoolfee BETWEEN 100 AND 250
- ✗ FROM Product WHERE description IS NOT NULL
- ✗ FROM Product p WHERE p.category.id IN (1, 3, 5, 7)

★ ORDER BY

- ✗ FROM Course ORDER BY startDate DESC, schoolfee

★ GROUP BY

- ✗ SELECT AVG(unitPrice), MAX(discount) FROM Product p GROUP BY p.category
- ✗ SELECT AVG(unitPrice), MAX(discount) FROM Product p GROUP BY p.category HAVING AVG(unitPrice) > 100
- ✗ SELECT AVG(unitPrice), MAX(discount) FROM Product p GROUP BY p.category HAVING p.category.name LIKE '%Nokia%'



HQL-Hàm và toán tử

★ Tổng hợp số liệu

✎ avg(...), sum(...), min(...), max(...)

✎ count(*)

✎ count(...), count(distinct ...), count(all...)

★ Toán tử

✎ Số học: +, -, *, /

✎ So sánh: =, >=, <=, <>, !=

✎ Logic: AND, OR, NOT

✎ Đặc biệt: [NOT] IN, [NOT] BETWEEN, IS [NOT] NULL, [NOT] LIKE, IS [NOT] EMPTY, [NOT] MEMBER OF



Các hàm khác

- ★ `concat(...,...)`: ghép chuỗi
- ★ `current_date()`: lấy ngày, tháng năm
- ★ `current_time()`: lấy giờ, phút và giây
- ★ `current_timestamp()`: lấy ngày giờ
- ★ `second(...)`: lấy giây
- ★ `minute(...)`: lấy phút
- ★ `hour(...)`: lấy giờ trong ngày
- ★ `day(...)`: lấy ngày trong tháng
- ★ `month(...)`: lấy tháng
- ★ `year(...)`: lấy năm
- ★ `substring()`: lấy chuỗi con
- ★ `trim()`: cắt bỏ ký tự trắng 2 đầu chuỗi
- ★ `lower()`: chuyển in thường
- ★ `upper()`: chuyển in hoa
- ★ `length()`: lấy độ dài chuỗi
- ★ `locate()`: tìm vị trí chuỗi con
- ★ `abs()`: lấy giá trị tuyệt đối
- ★ `sqrt()`: tính căn bậc 2
- ★ `str()`: chuyển số/ngày sang chuỗi
- ★ `cast(... as ...)`: ép kiểu