

**BỘ MÔN CNPM - KHOA CNTT - ĐH KHTN**

# **HIBERNATE NOTES**

**Nguyễn Hoàng Anh - [nhanh@fit.hcmus.edu.vn](mailto:nhanh@fit.hcmus.edu.vn)**

# Nội dung trình bày

- Một số lưu ý trong Hibernate
- Tài liệu tham khảo

# Configuration

- Programmatic configuration

```
Configuration configuration = new Configuration()
.addResource("com/metaarchit/bookshop/Book.hbm.xml")
.setProperty("hibernate.dialect", "org.hibernate.dialect.DerbyDialect")
.setProperty("hibernate.connection.driver_class", "org.apache.derby.jdbc.EmbeddedDriver")
.setProperty("hibernate.connection.url", "jdbc:derby://localhost:1527/BookShopDB")
.setProperty("hibernate.connection.username", "book")
.setProperty("hibernate.connection.password", "book");
SessionFactory factory = configuration.buildSessionFactory();
```

# Configuration

- Programmatic configuration

```
Configuration configuration = new Configuration()  
.addJar(new File("mapping.jar"))  
.setProperty("hibernate.dialect", "org.hibernate.dialect.DerbyDialect")  
.setProperty("hibernate.connection.driver_class", "org.apache.derby.jdbc.EmbeddedDriver")  
.setProperty("hibernate.connection.url", "jdbc:derby://localhost:1527/BookShopDB")  
.setProperty("hibernate.connection.username", "book")  
.setProperty("hibernate.connection.password", "book");  
SessionFactory factory = configuration.buildSessionFactory();
```

Gop tat ca file .hbm.xml vao file .jar

# Configuration

- XML configuration

```
new Configuration().configure("/config/recipes.cfg.xml")
```

- Properties file configuration: hibernate.properties

# load() vs get()

- load()

```
Book book = (Book) session.load(Book.class, isbn);
```

- Nếu không tìm thấy, quăng Exception
- Trả về proxy, chưa gọi CSDL

- get()

```
Book book = (Book) session.get(Book.class, isbn);
```

- Nếu không tìm thấy, trả về null
- Gọi CSDL ngay

# Annotation

```
<mapping class="com.hibernaterecipes.annotations.domain.Book"/>
```

```
package com.hibernaterecipes.annotations.domain;

import java.util.Date;
import javax.persistence.Column;
import javax.persistence.*;
import javax.persistence.Entity;
import javax.persistence.Table;

/**
 * @author Guruzu
 *
 */
@Entity
@Table (name="BOOK")
```

# Annotation

```
public class Book {  
  
    @Column (name="isbn")  
    @Id  
    String isbn;  
  
    @Column (name="book_Name")  
    String bookName;  
  
    @Column (name="publisher_code")  
    String publisherCode;  
  
    @Column (name="publish_date")  
    Date publishDate;  
  
    @Column (name="price")  
    Long price;  
}
```



# Logging

- Simple Logging Façade for Java (SLF4J)
  - NOP
  - Simple
  - Log4j
  - JDK logging
  - ...

# Logging

- Thực hiện với log4j:
  - Add file slf4j-log4j12.jar
  - Cấu hình log4j (log4j.properties)

- Hiển thị câu truy vấn

```
<property name="show_sql">true</property>
```

- File hibernate.cfg.xml

# Generate DB schema

- Sử dụng Ant & task schemaexport

```
<project name="BookShop" default="schemaexport">
<property name="build.dir" value="bin" />
<property name="hibernate.home" value="c:/hibernate-3.1" />
<property name="derby.home" value="c:/derby" />
<path id="hibernate-Classpath">
<fileset dir="${hibernate.home}">
<include name="**/*.jar" />
</fileset>
<fileset dir="${derby.home}">
<include name="lib/*.jar" />
</fileset>
```

```
<pathelement path="${build.dir}" />
</path>
<!-- Defining Ant targets -->
</project>
```

```
<target name="schemaexport">
<taskdef name="schemaexport"
classname="org.hibernate.tool.hbm2ddl.SchemaExportTask"
Classpathref="hibernate-Classpath" />
<schemaexport config="${build.dir}/hibernate.cfg.xml"
output="BookShop.sql" />
</target>
```

# Generating ID

- Mỗi persistent class phải có một ID
- ID có thể gồm một hoặc nhiều thuộc tính (composite-id)
- ID có thể tự động phát sinh theo nhiều cơ chế khác nhau:
  - assigned
  - increment
  - hilo (hilo algorithm), sequence, guid
- Mỗi DB không nhất thiết hỗ trợ hết các kiểu phát sinh này

## saveOrUpdate()

- Nếu persistent object sử dụng ID tự phát sinh:
  - ID là empty → new object → generate ID → INSERT
  - ID không empty → existing object → UPDATE
- Qui định ID là empty
  - C1: Thuộc tính unsaved-value="0" trong tag id
  - C2: Sử dụng các kiểu dữ liệu chứa null (Long, Integer, ...)

# Dynamic SQL Generation

- App sử dụng Hibernate + start app → Hibernate phát sinh các câu lệnh CRUD cho mỗi persistent class
- Các câu lệnh này không được thực thi
- Các câu lệnh này được cache trong bộ nhớ
  - Longer start-up time + ảnh hưởng đến hiệu năng
  - Dynamic SQL Generation: tắt tính năng trên của Hibernate

```
<class name="BookCh2" table="BOOK" dynamic-insert="true" dynamic-update="true">
```

# Collection

- `<set>`
- `<bag>`
- `<list>`
- `<map>`
- ...

# Sorting collections

- sort="natural"

```
<set name="chapters" table="Book61_Chapter" schema="BOOK6" sort="natural" lazy="false">  
  <key column="BOOK_ID"></key>  
  <element type="string" column="CHAPTER_NAME" not-null="true"></element>  
</set>
```

- Tự viết class comparator
- Sắp xếp trong CSDL

```
<set name="chapters" table="Book61_Chapter" schema="BOOK6"  
  lazy="false" order-by=" CHAPTER_NAME">
```



# Binding in HQL

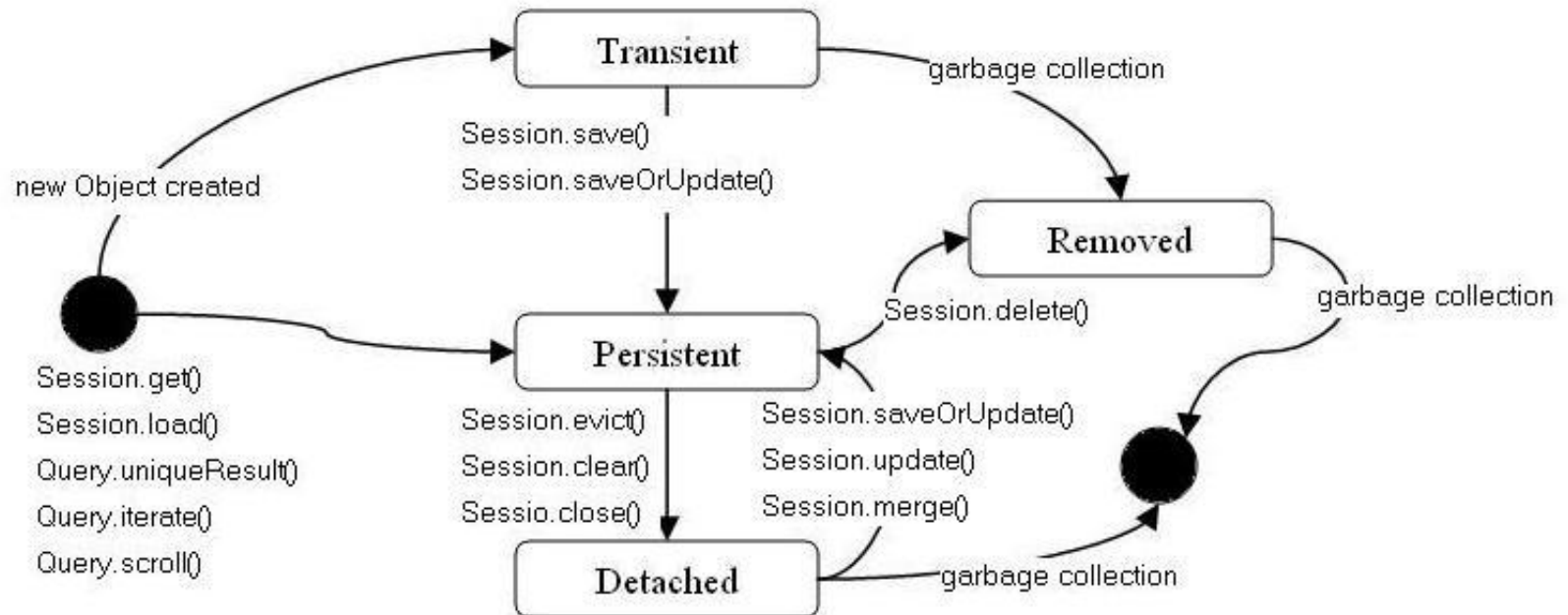
- Tại sao phải binding:
  - Câu query dạng cộng chuỗi → SQL Injection
  - Sử dụng cơ chế binding trong HQL sẽ an toàn hơn
- 2 cơ chế binding:
  - Named parameter binding

```
Query query = session.createQuery("from Book where isbn =:isbn");  
query.setLong("isbn", 520);  
Book bookCh8 = (Book) query.uniqueResult();  
System.out.println(bookCh8);
```

Positional parameter binding: bắt đầu từ 0

```
query = session.createQuery("from Book where isbn = ?");  
query.setLong(0, 520);  
Book bookCh8 = (Book) query.uniqueResult();
```

# Object states



## Tài liệu tham khảo

- Hồ Tuấn Thanh, Nguyễn Hoàng Anh
  - BMCNPM – KCNTT – ĐHKHTN
  - Hibernate Recipes: A Problem-Solution Approach, Gary Mak, Srinivas Guruzu
- Tất cả các sách Hibernate liên quan



# HỎI VÀ ĐÁP