**RƯỜNG ĐẠI HỌC QUY NHƠN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO GIỮA KỲ**

**HỌC PHẦN: CÔNG NGHỆ JAVA**

**Đề tài:Hibernate**

|  |  |
| --- | --- |
| Họ và tên | Trần Quốc Việt (Nhóm trưởng)  Ngô Dức Tường  Huỳnh Vạn Toàn  Nguyễn Văn Hải |
|  |
| Lớp: | Công nghệ phần mềm K41 |
| Giảng viên hướng dẫn: | Vũ Sơn Lâm |

Mục lục

[1. **Advanced Mapping**. 3](#_Toc89187080)

[**1.1.** **One-to-one** 3](#_Toc89187081)

[**1.2.** **One-to-Many và Many-to-one.** 4](#_Toc89187082)

[**1.3.** **Many-to-Many.** 4](#_Toc89187083)

[**2.** **Cascade.** 5](#_Toc89187084)

[**3.** **One-to-one relationship.** 5](#_Toc89187085)

[**3.1. Thiết lập môi trường Hibernate Annotation** 5](#_Toc89187086)

[**3.2.** **Cơ sở dữ liệu.** 7](#_Toc89187087)

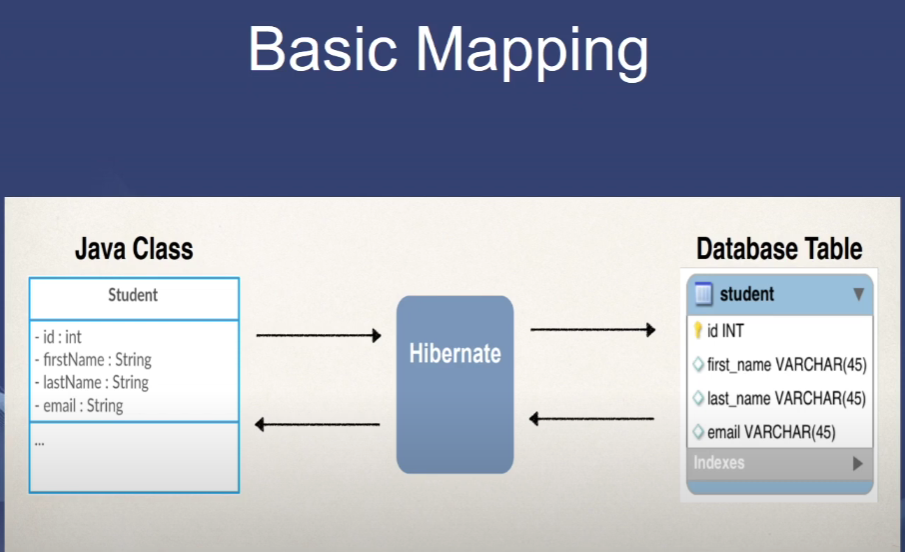
[**3.3.** **Tạo các lớp.** 9](#_Toc89187088)

[**3.4.** **Tạo truy vấn Create Student.** 10](#_Toc89187089)

[**4.** **Tài liệu tham khảo.** 11](#_Toc89187090)

1. Association Mappings.

Basic maping:



* 1 class java được hibernate sử dụng để map với 1 bảng trong cơ sở dữ liệu (Tương đương tên , thuộc tính của class java này với tên bảng , các trường của bảng tương ứng trong cơ sở dữ liệu)

Advanced Mapping:

Trong 1 database:

* Có nhiều bảng.
* Các bảng có quan hệ với nhau.

Mối quan hệ giữa các bảng này chính là Advadced Mapping.

* **Mapping Type:**
* **One-to-one**
* **One-to-many**
* **Many-to-one**
* **Many-to-many**
  1. **One-to-one**

Với 2 table A và B, chúng ta có thể hiểu là 1 record ở table A sẽ liên kết duy nhất một và chỉ một record ở table B mà thôi. Khi đó, ta sẽ nói A có quan hệ One-to-One với B hoặc B có quan hệ One-to\_one với A.

Thông thường, mối quan hệ One-to-One được thể hiện thông qua khoá foreign keys. Sử dụng các forein keys sẽ thực hiện join các table lại với nhau, có rất nhiều lựa chọn để đặt forein key, tức là A có foreign key đến B và B có foreign key đến A(quan hệ 2 chiều) hoặc A có foreign key đến B, B không có gì hoặc B có foreign key đến A, A không có gì.

* 1. **One-to-Many và Many-to-one.**

Một record trong table A có liên kết đến nhiều record ở table B, 1 record ở table B sẽ có 1 record ở table A, lúc này ta sẽ nói A có mối quan hệ một-nhiều với B, B có mối quan hệ nhiều-một với A.

Trong mối quan hệ One-to-Many và Many-to-one chúng ta sử dụng join table để store đến các mapping values, có nghĩa A và B sẽ không sử dụng foreign key mà chúng ta sẽ có một cái join table, trong join table này sẽ có các foreign key liên kết với các table A và B

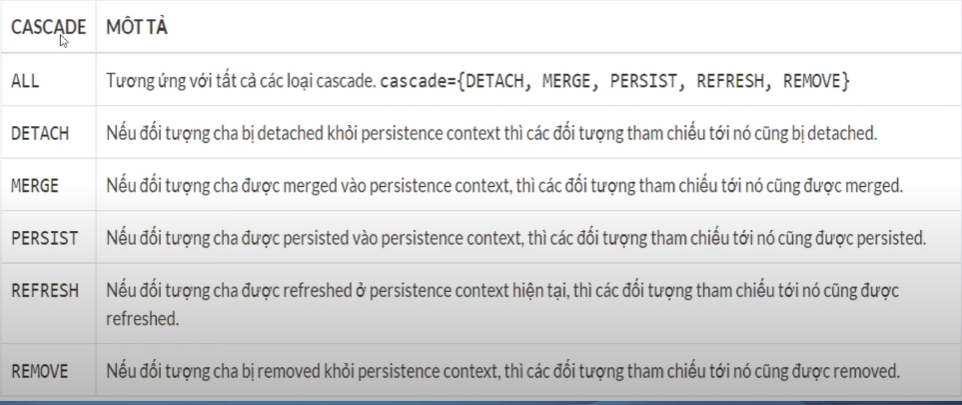
* 1. **Many-to-Many.**

Mối quan hệ Many-to-Many là mối quan hệ khi mà 1 record của table A có liên kết với nhiều record ở table B, ngược lại 1 record ở table B cũng có liên kết với nhiều record ở table A. Khi đó ta nói A, B có mối quan hệ Many-to-Many.

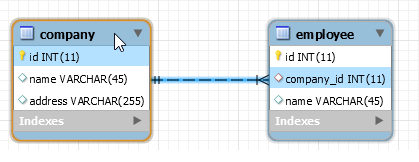
Trong mối quan hệ Many-to-Many chúng ta sử dụng join table để store đến các mapping values, có nghĩa A và B sẽ không sử dụng foreign key mà chúng ta sẽ có một cái join table, trong join table này sẽ có các foreign key liên kết với các table A và B

1. **Cascade.**

Cascade là một tính năng giúp quản lý trạng thái của các đối tượng trong một mối quan hệ một cách tự động.



Ví dụ ta có mối quan hệ giữa bảng company và bảng employee như sau:



Khi một bản ghi thay đổi thì nó sẽ tự động update các bản ghi đang tham chiếu tới nó(vd: khi xóa bảng company thì dữ liệu bảng emplyee cũng được xóa )

1. **One-to-one relationship.**

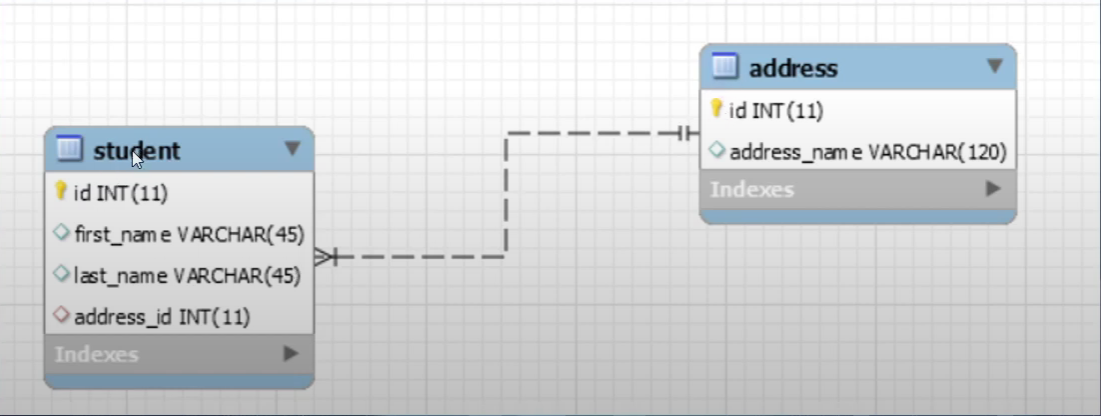
Mối quan hệ 1-1 giữa 02 đối tượng sử dụng Hibernate

* 1. **Thiết lập môi trường Hibernate Annotation**
* Trước hết, bạn phải đảm bảo rằng bạn đang sử dụng JDK 5.0 nếu không bạn cần phải nâng cấp JDK của mình lên JDK 5.0 hoặc bản mới hơn để được hỗ trợ sử dụng annotation.
* Thứ hai, bạn sẽ cần phải cài đặt gói Hibernate 3.x annotations, có sẵn từ sourceforge: (Download Hibernate Annotations) và sao chép hibernate-annotations.jar, lib/hibernate-comons-annotations.jar và lib/ejb3-persistence.jar từ file tải về đến CLASSPATH của bạn
* Hoặc bạn có thể add các dependency vào file pom.xml như sau:

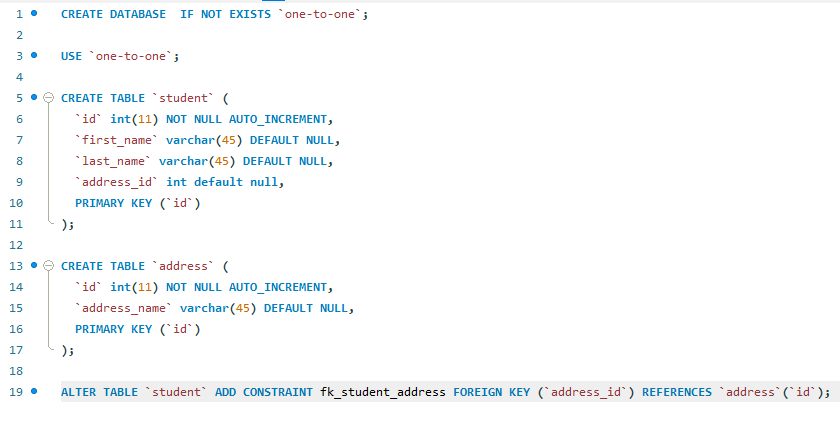


Demo.

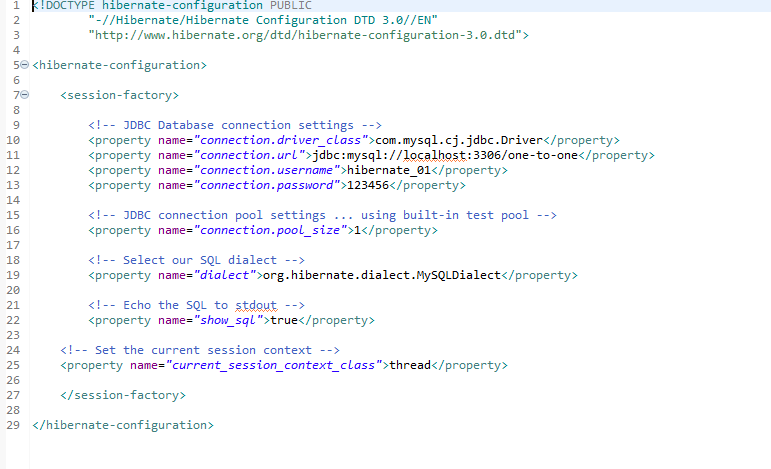
Ví dụ:



* 1 Student có 1 address và 1 address sẽ thuộc về 1 student.
  1. **Cơ sở dữ liệu.**

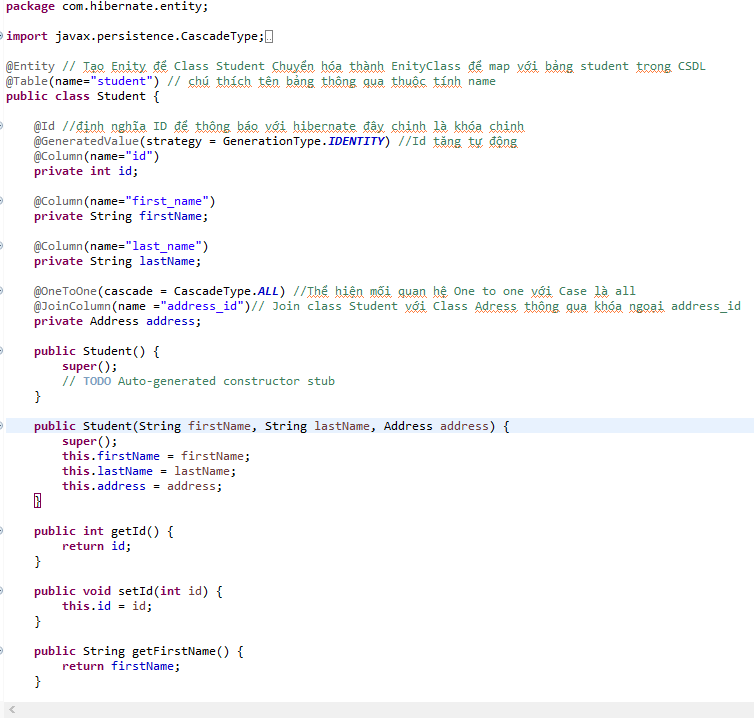


File cấu hình hibernate.cfg.xml.



* 1. **Tạo các lớp.**

Student.java



Thực hiện khai báo Quan hệ One to one:

Trên Thuộc tính Address khai báo:

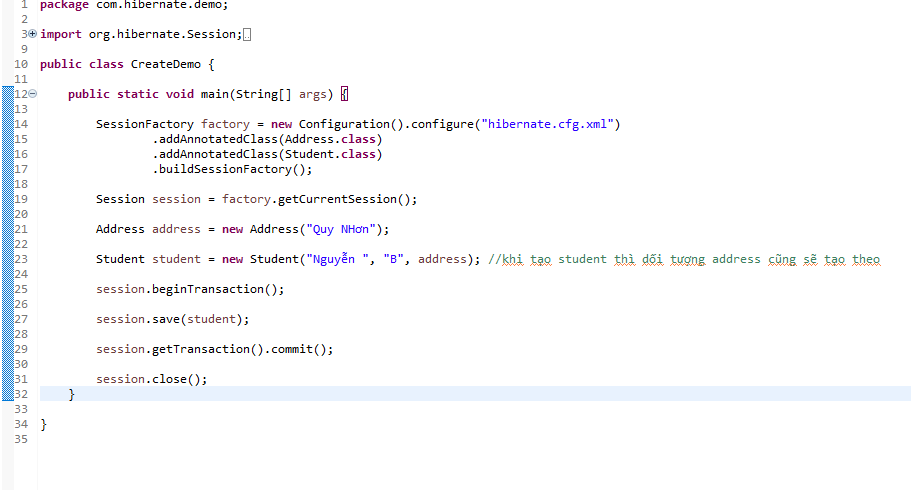
@OneToOne(cascade = CascadeType.***ALL***) //Thể hiện mối quan hệ One to one với Case là all

@JoinColumn(name ="address\_id")// Join class Student với Class Adress thông qua khóa ngoại address\_id

**private** Address address;  
Address.java



* 1. **Tạo truy vấn Create Student.**

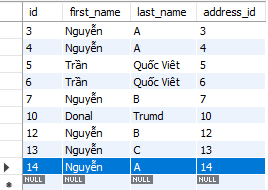


Tạo SessionFactory với 2 addAnnotatedClass là Adrress và Student.

Thực hiện thêm dữ liệu vào đối tượng Student với Tên Nguyễn B và địa chỉ là Quy Nhơn.

Kết quả :

Bảng Student: Bảng Address:



1. **Tài liệu tham khảo.**

<https://viettuts.vn/hibernate>

https://www.youtube.com/playlist?list=PLdDlHRNwGUQY1yDUXdxuQxNdnIpRALzAY