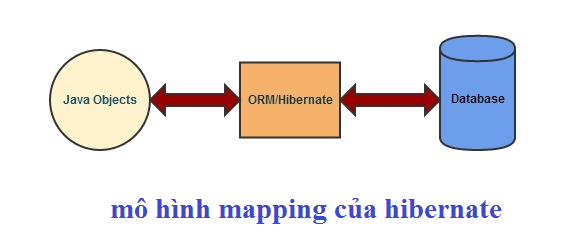
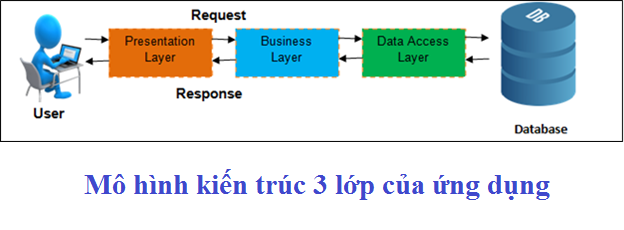
Tổng quan về Hibernate

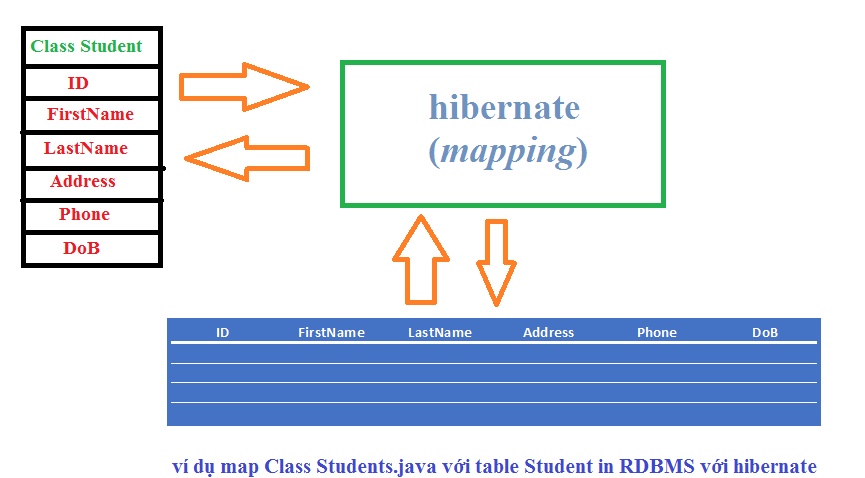


1. Giới thiệu về Hibernate



Mô hình lập trình hiện đại thường chia kiến trúc ứng dụng thành 3 lớp như sau: lớp giao diện người dùng (UI layer), lớp xử lý nghiệp vụ (business layer) và lớp chứa dữ liệu (data layer). Trong đó, business layer có thể bị chia nhỏ thành 2 layer con là business logic layer và persitence layer. Persistence layer chịu trách nhiệm giao tiếp với data. Persistence layer sẽ đảm nhiệm các nhiệm vụ mở kết nối, truy xuất và lưu trữ dữ liệu vào các RDBMS





Hibernate là một giải pháp ORM cho Java , một dự án open source chuyên nghiệp, một framework cho pesistence layer.

Hibernate ánh xạ các object Java với các table trong cơ sở dữ liệu và ánh xạ giữa các kiểu dữ liệu trong Java với các kiểu dữ liệu SQL. Hibernate giúp giảm thiểu các công việc liên quan đến nhiệm vụ xử lý dữ liệu thông thường trong phát triển ứng dụng.

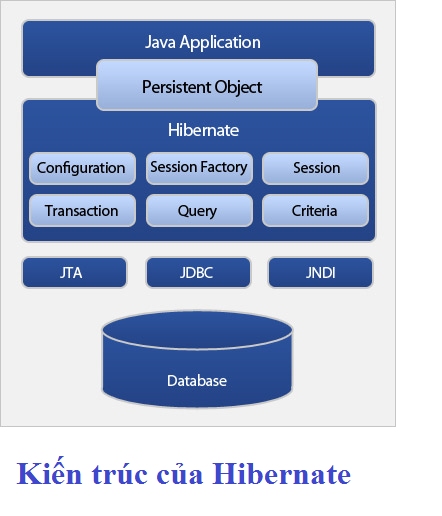
Hibernate nằm giữa các đối tượng Java truyền thống và cơ sở dữ liệu để giải quyết tất cả các công việc trong lớp persistence dựa trên mô hình kỹ thuật ORM.

**Ưu điểm của Hibernate**

* Hibernate chịu trách nhiệm ánh xạ giữa các lớp Java đến các bảng trong CSDL dùng các file XML
* Cung cấp các API đơn giản để lưu trữ, truy xuất trực tiếp các đối tượng Java và CSDL.
* Nếu có bất kỳ thay đổi nào trong CSDL thì chỉ cần thay đổi file XML.
* Cung cấp đầy đủ các tiện ích, tính năng truy vấn dữ liệu đơn giản, hiệu quả.
* Thao tác, xử lý được các quan hệ phức tạp của các đối tượng trong CSDL.
* Giảm thiểu tối đa sự truy cập đến CSDL nhờ có chiến lược tìm, nạp thông minh.
* Hỗ trợ hầu hết các RDBMS như: MySQL, PostgreSQL, Oracle,…
* Hỗ trợ một số công nghệ như: XDoclet Spring, J2EE, Eclipse plug-ins, Maven,…

1. Kiến trúc Hibernate

Hibernate sử dụng các file cấu hình để cung cấp các dịch vụ và đối tượng persistence cho ứng dụng



Hibernate dùng rất nhiều loại Java API có sẵn như JDBC, Java Transaction API (JTA) và Java Naming and Directory Interface (JNDI). JDBC cung cấp các chức năng làm việc với CSDL quan hệ ở mức thô sơ, trừu tượng, cho phép bất kỳ một CSDL nào dùng JDBC driver đều được hỗ trợ bởi Hibernate. JNDI và JTA cho phép hibernate được tích hợp với các server ứng dụng J2EE.

1. **Configuration Object**

Configuration object là file cấu hình hoặc file thuộc tính theo yêu cầu của hibernate. Configuration object cung cấp hai thành phần quan trọng sau:

* Database Connection: gồm các tập tin hibernate.properties và hibernate.cfg.xml
* Class Mapping Setup: Tạo sự kết nối giữa các lớp trong java và các bảng trong CSDL

1. **SessionFactory Object**

Configuration object được dùng để tạo ra các đối tượng SessionFactory, cho phép các đối tượng session được khởi tạo. SessionFactory thường được tạo ra từ lúc ứng dụng được khởi tạo và được giữ lại về sau.

1. **Session Object**

Một session được sử dụng để lấy kết nối vật lí với CSDL. Session được thiết kế để được khởi tạo cho mỗi lần tương tác với CSDL. Các đối tượng persistent được lưu trữ và truy cập thông qua đối tượng Session. Một session không nên mở trong thời gian dài bởi vì lý do không an toàn, mà nên được tạo và hủy chúng khi cần thiết.

1. **Transaction Object**

Một transaction đại diện cho một phiên giao dịch với CSDL. Transaction trong hibernate được thực thi bởi một trình quản lý giao dịch (từ JDBC hoặc JTA). Đây là một đối tượng tùy chọn và các ứng dụng hibernate có thể không chọn sử dụng nó, thay vào đó có thể quản lý transaction trong các đoạn code.

1. **Query Object**

Các đối tượng Query dùng SQL hoặc Hibernate Query Language (HQL) để truy xuất dữ liệu từ CSDL và tạo ra các đối tượng. Một query thường được sử dụng để ràng buộc các tham số truy vấn, giới hạn số lượng kết quả truy vấn và thực thi câu truy vấn.

1. **Criteria Object**

Criteria thường được sử dụng để tạo và thực thi các điều kiện truy vấn hướng đối tượng để truy xuất các đối tượng.

1. Tài liệu tham khảo
   * <https://en.wikipedia.org/wiki/Hibernate_(framework)>
   * <http://hibernate.org/orm/>