

Hibernate

Nguyễn Anh Tuấn

KTECH
COLLEGE



Nội dung bài giảng

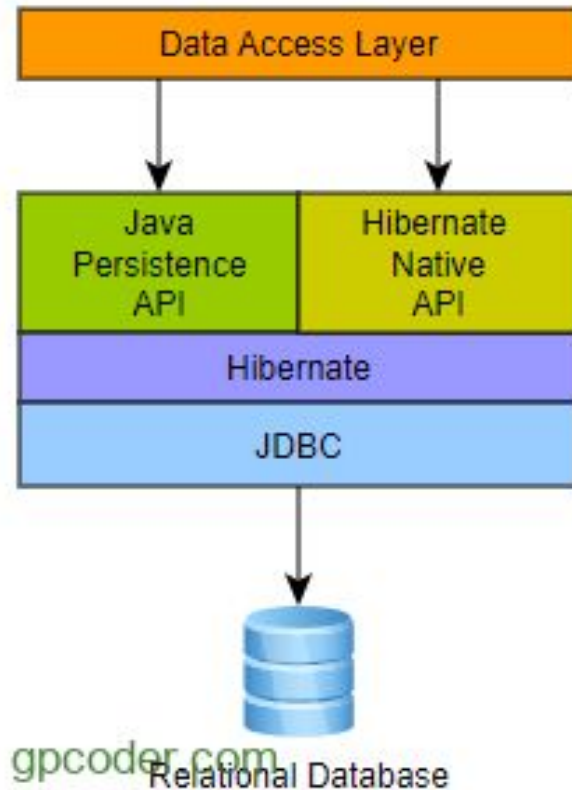
- 1 Hibernate là gì?
- 2 Kiến trúc Hibernate
- 3 Các tính chất của Hibernate

Hibernate là gì?

Hibernate là gì?

- **Định nghĩa:**

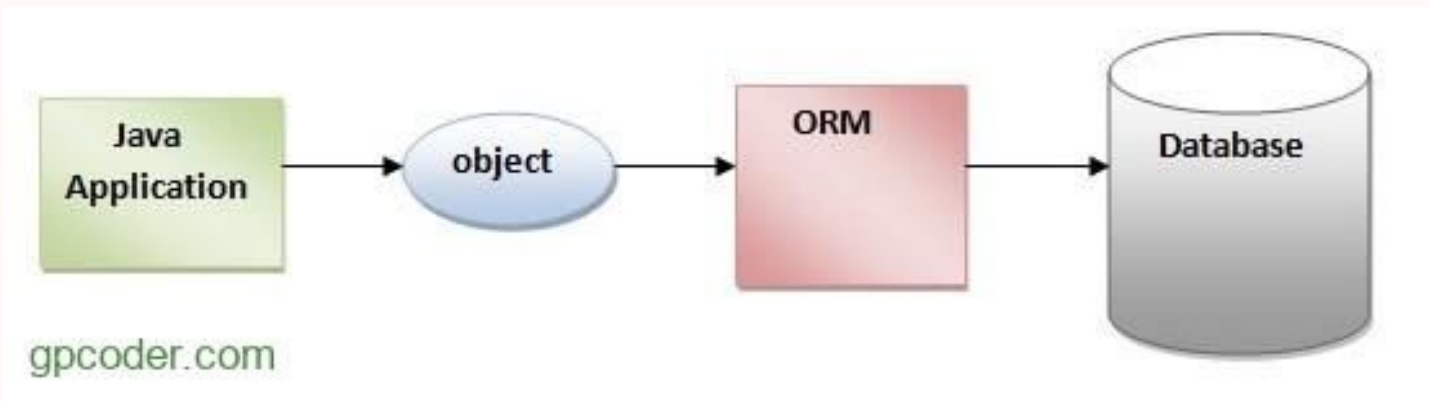
- **Hibernate** ra đời nhằm tối ưu hơn và giải quyết vấn đề tồn tại của JDBC.
- **Hibernate** là một thư viện **ORM (Object Relational Mapping)** mã nguồn mở, hỗ trợ viết ứng dụng Java có thể map các objects với hệ quản trị cơ sở dữ liệu quan hệ, và hỗ trợ thực hiện các khái niệm lập trình hướng đối tượng với cơ dữ liệu quan hệ.
- **Hibernate** cho phép truy vấn dữ liệu thông qua **JPA (Java Persistence API)** hoặc bằng ngôn ngữ SQL mở rộng của Hibernate (HQL) hoặc bằng SQL thuần (Native SQL).



ORM (Object Relational Mapping) là gì?

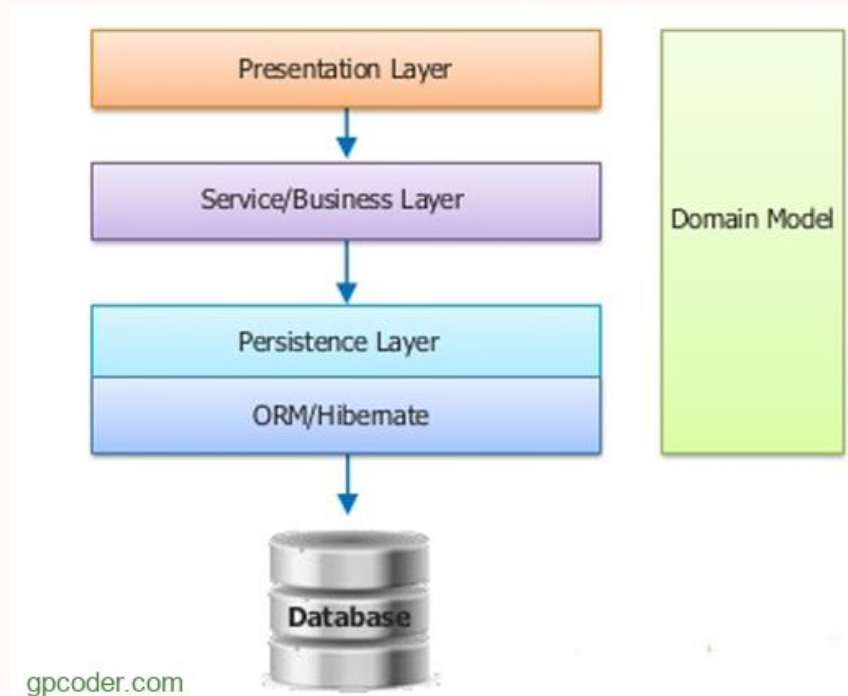
- **Định nghĩa:**

- **ORM (Object Relational Mapping) framework** là một cơ chế cho phép người lập trình thao tác với database một cách hoàn toàn tự nhiên thông qua các đối tượng. Lập trình viên hoàn toàn không quan tâm đến loại database sử dụng SQL Server, MySQL, PostgreSQL, ...



JPA (Java Persistence API) là gì?

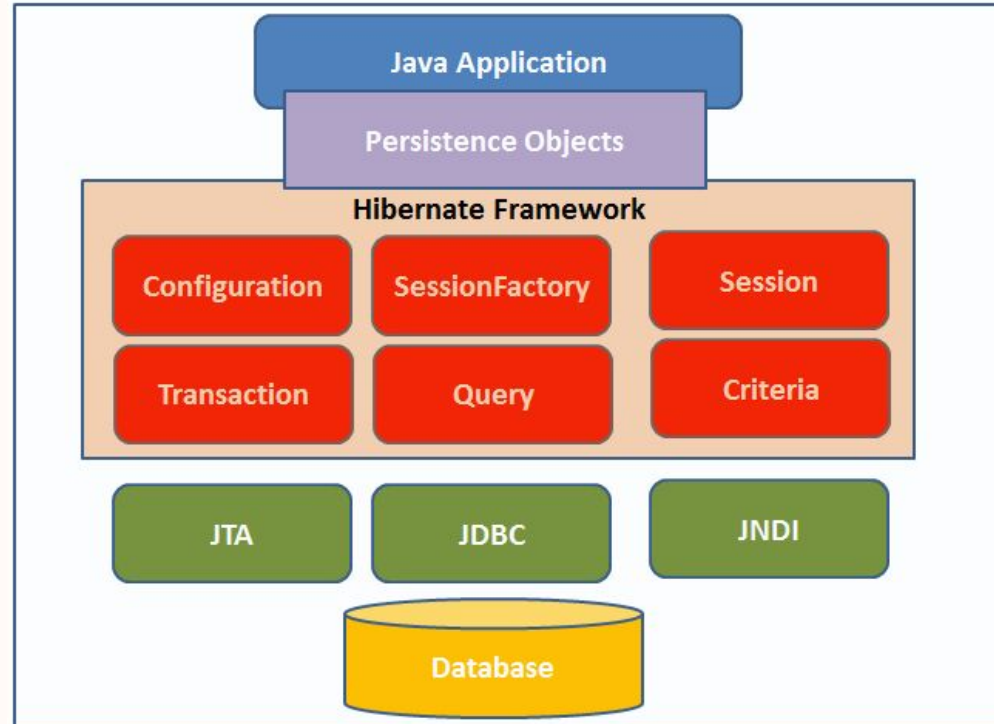
- Một ứng dụng có thể được chia làm 3 phần như sau:
 - Tầng giao diện người dùng (presentation layer)
 - Tầng xử lý nghiệp vụ (business layer):
 - Business logic layer: Bao gồm các tính toán logic, nghiệp vụ từ phía client.
 - Persistence layer: Giao tiếp với tầng dữ liệu (kết nối, truy xuất và lưu trữ dữ liệu).
 - Tầng dữ liệu (data layer).



Kiến trúc của Hibernate

Kiến trúc của Hibernate

- **Persistence object:**
 - Là các object map với các table tương ứng của cơ sở dữ liệu quan hệ.
 - Là những container chứa dữ liệu từ ứng dụng để lưu xuống database, hay chứa dữ liệu tải lên từ database.



Kiến trúc của Hibernate

- **Configuration:**

- Là đối tượng đại diện cho một tập tin cấu hình hoặc thuộc tính yêu cầu của Hibernate.
- Configuration cung cấp hai thành phần chính:
 - **Database Connection:** Thao tác này được xử lý thông qua một hoặc nhiều tệp cấu hình được Hibernate hỗ trợ. Các tệp này là hibernate.properties và hibernate.cfg.xml.
 - **Class Mapping Setup:** Thành phần này tạo ra kết nối giữa các lớp Java và các bảng cơ sở dữ liệu.

Kiến trúc của Hibernate

- **Session Factory:**

- Là một interface giúp tạo ra session kết nối đến database bằng cách đọc các cấu hình trong Hibernate configuration.
- SessionFactory là đối tượng nặng (heavy weight object) nên thường được tạo ra trong quá trình khởi động ứng dụng và lưu giữ để sử dụng sau này.
- Mỗi một database phải có một session factory. Vì vậy, nếu sử dụng nhiều cơ sở dữ liệu thì phải tạo nhiều đối tượng SessionFactory.

Kiến trúc của Hibernate

- **Hibernate Session**

- Một session được sử dụng để có được một kết nối vật lý với một cơ sở dữ liệu.
- Session là đối tượng nhẹ và được thiết kế để được tạo ra instance mỗi khi tương tác với cơ sở dữ liệu.
- Các đối tượng Session không nên được mở trong một thời gian dài bởi vì chúng thường không phải là luồng an toàn (thread-unsafe) và chúng cần được tạo ra và được đóng khi cần thiết.
- Mỗi một đối tượng session được Session factory tạo ra sẽ tạo một kết nối đến database.

Kiến trúc của Hibernate

- **Transaction**

- Một Transaction đại diện cho một đơn vị làm việc với cơ sở dữ liệu và hầu hết các RDBMS hỗ trợ chức năng transaction.
- Transaction đảm bảo tính toàn vẹn của phiên làm việc với cơ sở dữ liệu. Tức là nếu có một lỗi xảy ra trong transaction thì tất cả các tác vụ thực hiện sẽ thất bại.
- Transaction là một đối tượng tùy chọn và các ứng dụng Hibernate có thể chọn không sử dụng interface này, thay vào đó quản lý transaction trong code ứng dụng riêng.

Kiến trúc của Hibernate

- **Query**

- Các đối tượng Query sử dụng chuỗi truy vấn SQL (Native SQL) hoặc Hibernate Query Language (HQL) để lấy dữ liệu từ cơ sở dữ liệu và tạo các đối tượng.

- **Criteria**

- Đối tượng Criteria được sử dụng để tạo và thực hiện các tiêu chí truy vấn để lấy các đối tượng từ database.

Các tính chất của Hibernate

Các tính chất của Hibernate

- **Object Mapping**

- Hibernate hỗ trợ map các trường trong bảng với các thuộc tính của Java object một cách “tự động” thông qua các file cấu hình map XML hay sử dụng các annotation.

@Entity

@Table(name = "Students")

@Data

```
public class UserModel {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Integer id;
```

```
    @NotEmpty
```

```
    @Column(name = "fullname")
```

```
    private String fullname;
```

```
}
```

Các tính chất của Hibernate

- **HQL**
 - Hibernate cung cấp các câu lệnh truy vấn tương tự SQL.
 - HQL hỗ trợ các truy vấn đa hình có thể kể tới như: HQL “hiểu” được những khái niệm về kế thừa (inheritance), đa hình (polymorphism) và liên kết (association).

Các tính chất của Hibernate

- **Database Independent**

- Code sử dụng Hibernate là độc lập với hệ quản trị cơ sở dữ liệu, nghĩa là không cần thay đổi câu lệnh HQL khi ta chuyển từ hệ quản trị CSDL MySQL sang Oracle, hay các hệ quản trị CSDL khác... Do đó rất dễ để thay đổi CSDL quan hệ, đơn giản bằng cách thay đổi thông tin cấu hình hệ quản trị CSDL trong file cấu hình.

```
com.mysql.jdbc.Driver // Cấu hình MySQL  
oracle.jdbc.driver.OracleDriver // Cấu hình Oracle
```

```
// Câu lệnh MySQL
```

```
SELECT column_name FROM table_name ORDER BY column_name ASC;
```

```
// Câu lệnh Oracle
```

```
SELECT column_name FROM table_name ORDER BY column_name ASC;
```

Các tính chất của Hibernate

- **Minimize Code Changes**

- Khi thay đổi (thêm) cột vào bảng, với **JDBC**:
 - Thêm thuộc tính vào object class.
 - Thay đổi method chứa câu truy vấn “select”, “insert”, “update” để bổ sung cột mới.
- Với **Hibernate**:
 - Thêm thuộc tính vào Entity class.
 - Cập nhật Hibernate Annotation để map column – property.

Các tính chất của Hibernate

- **Lazy Loading**

- Với những ứng dụng Java làm việc với cơ sở dữ liệu lớn hàng trăm triệu bản ghi, việc có sử dụng Lazy loading trong truy xuất dữ liệu từ database mang lại lợi ích rất lớn.
- Ví dụ 1 user có nhiều document. Mỗi khi lấy thông tin user và document tương ứng từ database, để ứng dụng không bị chậm Lazy loading sẽ lấy user, còn document chỉ lấy khi được truy cập.



Các tính chất của Hibernate

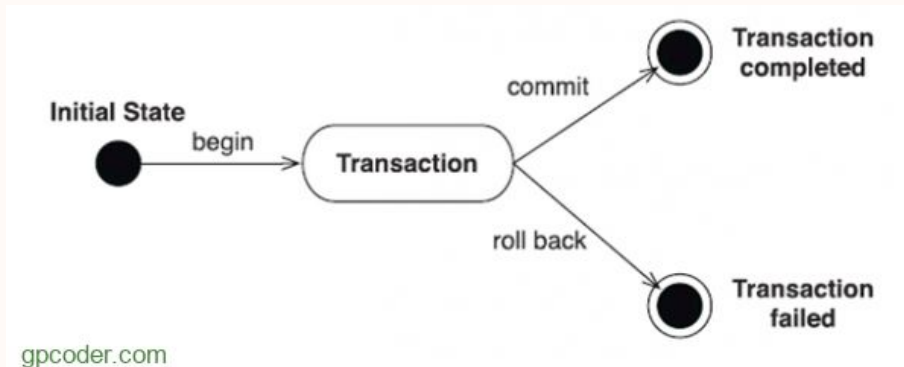
- **Loại bỏ Try-Catch Blocks**

- Sử dụng JDBC nếu lỗi xảy khi thao tác với database thì sẽ có exception SQLException ném ra. Bởi vậy phải sử dụng try-catch block để xử lý ngoại lệ.
- Hibernate override toàn bộ JDBC exception thành Unchecked Exception nên không cần try-catch những exception liên quan tới JDBC.

Các tính chất của Hibernate

- **Quản lý commit/rollback Transaction**

- Transaction là nhóm các hoạt động (với database) của một tác vụ. Nếu một hoạt động không thành công thì toàn bộ tác vụ không thành công.
- Với JDBC, cần phải chủ động thực hiện commit khi toàn bộ hoạt động của tác vụ thành công, hay phải rollback khi có một hoạt động không thành công để kết thúc tác vụ.
- Với Hibernate thì không cần quan tâm đến commit hay rollback, Hibernate quản lý cả auto-commit và auto-rollback.



Các tính chất của Hibernate

- **Hibernate Caching**

- Hibernate cung cấp một cơ chế bộ nhớ đệm, giúp giảm số lần truy cập vào database của ứng dụng càng nhiều càng tốt. Điều này sẽ có tác dụng tăng performance đáng kể.
- Hibernate lưu trữ các đối tượng trong session khi transaction được kích hoạt. Khi một query được thực hiện liên tục, giá trị lưu trữ trong session được sử dụng lại. Khi một transaction mới bắt đầu, dữ liệu được lấy lại từ database và được lưu trữ session.

Các tính chất của Hibernate

- **Associations**

- Liên kết giữa các bảng bằng Hibernate như quan hệ một-một, một-nhiều, nhiều-một và nhiều-nhiều trong Hibernate bằng cách sử dụng các Annotation để ánh xạ đối tượng của bảng:

- **@OneToOne**
- **@OneToMany**
- **@ManyToMany**
- **@ManyToOne**

Các tính chất của Hibernate

- **JPA Annotation Support**

- Hibernate implement đặc tả JPA, do đó có thể sử dụng các Annotation của JPA như @Entity, @Table, @Column, ... để chuyển đổi giữa các ORM Framework mà không cần phải sửa đổi code.

- **Connection Pooling**

- Connection Pooling giúp tăng performance nhờ vào sử dụng lại các kết nối khi có yêu cầu thay vì việc tạo kết nối mới.

ROAD TO KOREA

Nếu có bất kỳ thắc mắc nào, hãy đặt câu hỏi qua

mail@mail.com hoặc Zalo 0xxx xxx xxx