

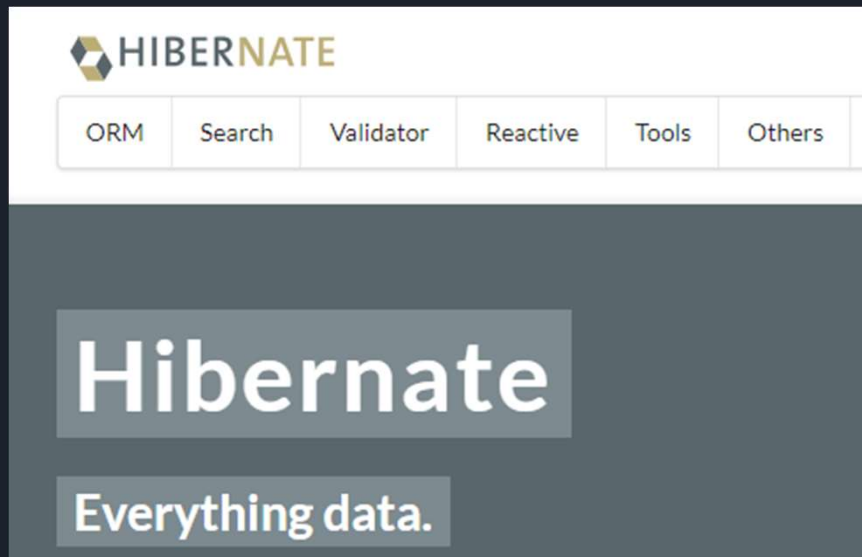


GIỚI THIỆU VỀ HIBERNATE VÀ JPA

Giới thiệu về Hibernate

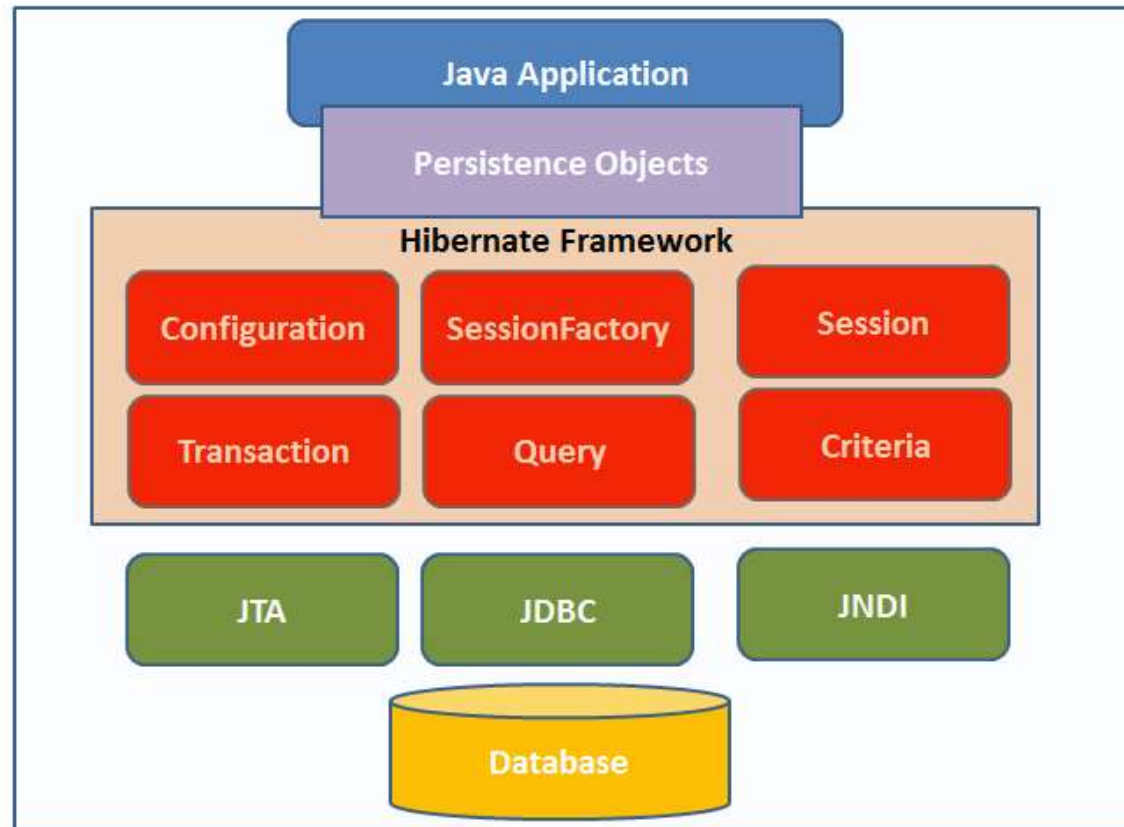
Hibernate là một framework phát triển ứng dụng trong lĩnh vực lập trình Java.

Được sử dụng để giải quyết vấn đề liên quan đến lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu.



<https://hibernate.org/>

Hibernate Architecture



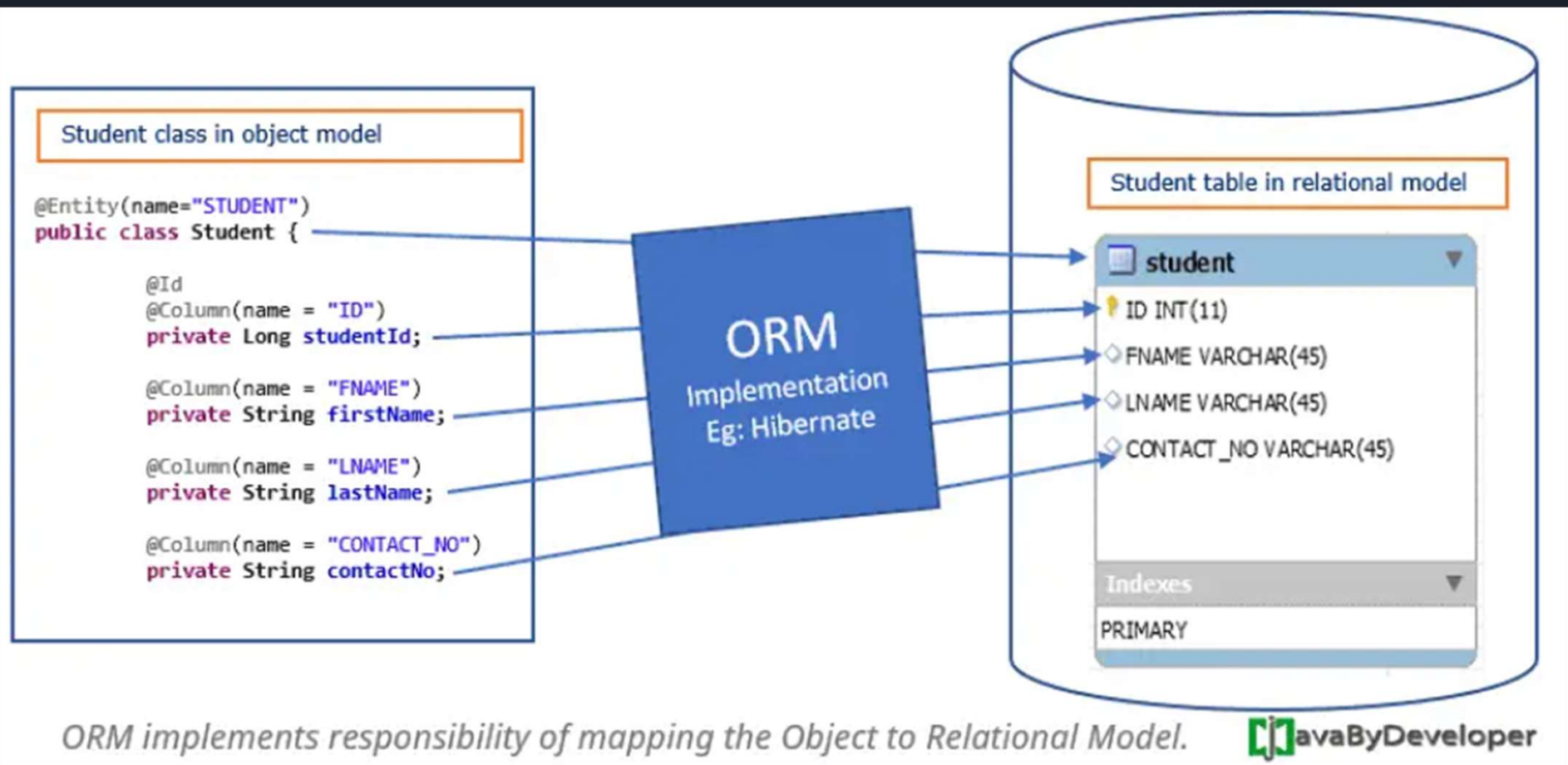
<https://topdev.vn/blog/hibernate-la-gi-sao-phai-dung-no-thay-jdbc/>



Vì sao nên dùng Hibernate?

- Ánh xạ đối tượng quan hệ - Object Relational Mapping (ORM)
- Quản lý mối quan hệ - Relationship Management
- Tối ưu hóa truy vấn - Query Optimization

Object Relational Mapping (ORM)



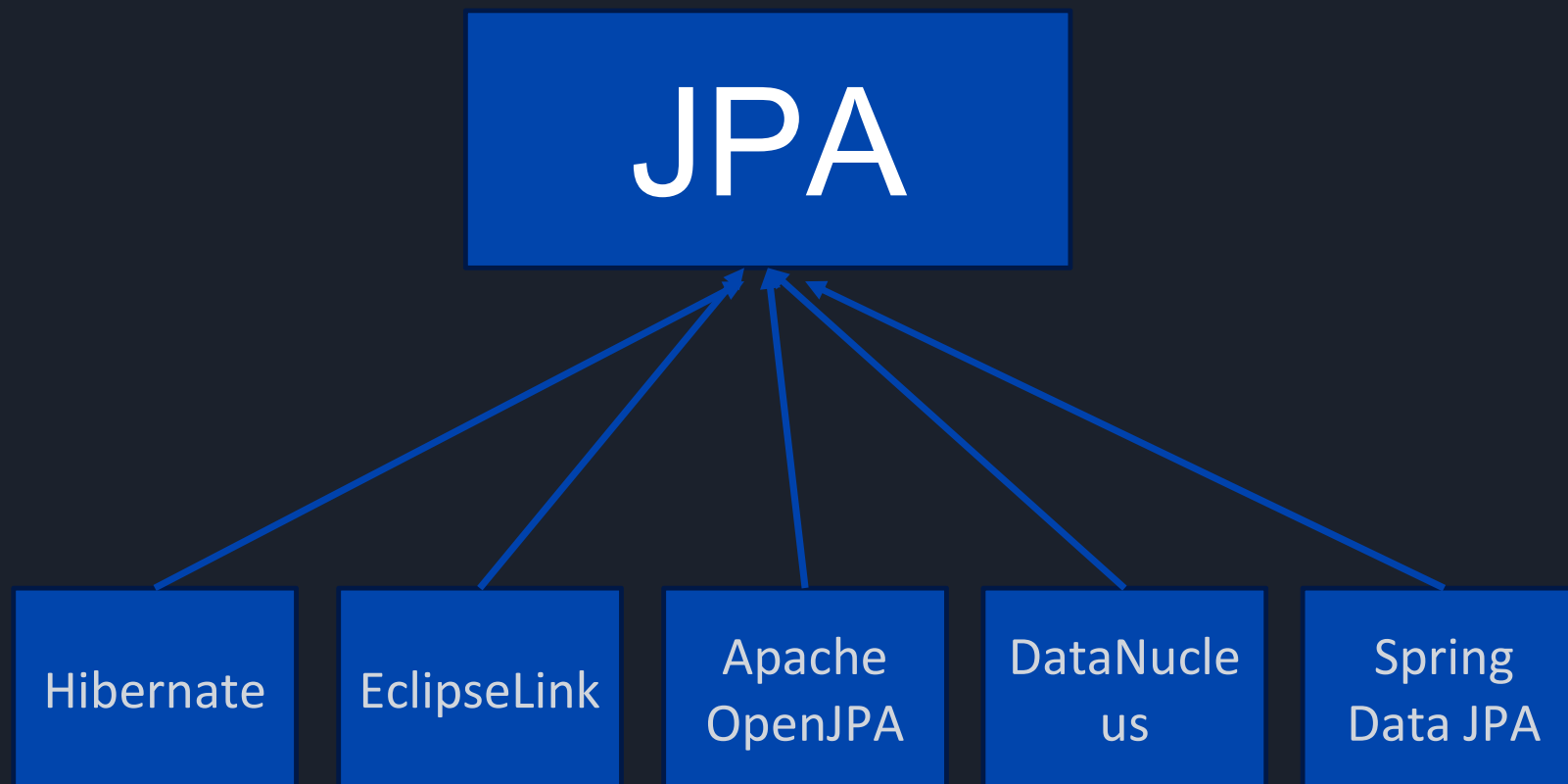


JPA là gì?

Jakarta Persistence API (JPA) (bắt đầu từ Java EE9)
(trước đây được hiểu là Java Persistence API)

Là một API tiêu chuẩn cho việc phát triển các framework ORM.

Nó bao gồm các Interface và các phương thức yêu cầu hiện thực để sử dụng.



LNT0

đá

Le Nhat Tung, 2023-07-08T21:02:27.248



Những lợi ích của JPA

- Không bị phụ thuộc vào 1 nhà cung cấp cụ thể.
- Dễ bảo trì, code linh động.
- Nếu có một nhà cung cấp không còn tồn tại thì sao?

```
Student student = new Student("Ramesh", "Fadatare", "rameshfadatare@javaguides.com");
Student student1 = new Student("John", "Cena", "john@javaguides.com");
Transaction transaction = null;
try (Session session = HibernateUtil.getSessionFactory().openSession()) {
    // start a transaction
    transaction = session.beginTransaction();

    // save the student objects
    session.save(student);
    session.save(student1);

    // commit transaction
    transaction.commit();
} catch (Exception e) {
    if (transaction != null) {
        transaction.rollback();
    }
    e.printStackTrace();
}
```

JPA


[Hibernate - Create, Read, Update and Delete \(CRUD\) Operations Example \(javaguides.net\)](http://javaguides.net)

```
Transaction transaction = null;
try (Session session = HibernateUtil.getSessionFactory().openSession()) {
    // start a transaction
    transaction = session.beginTransaction();

    // get Student entity using get() method
    Student student = session.get(Student.class, id);
    System.out.println(student.getFirstName());
    System.out.println(student.getEmail());

    // commit transaction
    transaction.commit();
} catch (Exception e) {
    if (transaction != null) {
        transaction.rollback();
    }
    e.printStackTrace();
}
```

[Hibernate - Create, Read, Update and Delete \(CRUD\) Operations Example \(javaguides.net\)](http://javaguides.net)



```
try (Session session = HibernateUtil.getSessionFactory().openSession()) {  
    List < Student > students = session.createQuery("from Student", Student.class).list();  
    students.forEach(s -> System.out.println(s.getFirstName()));  
} catch (Exception e) {  
    if (transaction != null) {  
        transaction.rollback();  
    }  
    e.printStackTrace();  
}
```



CRUD là gì?

Create

Read

Update

Delete

Cái nào làm thay đổi Database?

Hibernate/JPA có liên quan gì JDBC?

