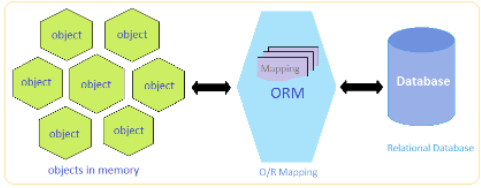
[Bài đọc] ORM

ORM (Object-Relational Mapping) là kỹ thuật liên kết giữa các đối tượng trong lập trình với các đối tượng trong CSDL. Khi sử dụng lập trình hướng đối tượng và CSDL quan hệ, ORM sẽ giúp ánh xạ các lớp đến các bảng và các đối tượng đến các bản ghi trong các bảng.

**Lợi ích của ORM**

* Cho phép truy xuất dễ dàng đến dữ liệu thông qua các đối tượng lập trình
* Giúp lập trình viên tập trung thao tác với các đối tượng, không cần quá quan tâm đến CSDL thực tế đang dùng
* Tự động hoá rất nhiều thao tác như quản lý giao dịch, validate dữ liệu, cache



Khi chúng ta lưu trữ dữ liệu của một ứng dụng, đứng từ phía một người quản trị dữ liệu và một người lập trình viên thì các góc nhìn có thể khác nhau, và các công nghệ chúng ta sử dụng cũng có thể khác nhau. Chẳng hạn, để lưu trữ danh sách thành viên trong CSDL thì một người quản trị CSDL sẽ thực hiện các công việc sau bằng ngôn ngữ SQL:

CREATE TABLE persons (  
       id integer NOT NULL,  
       name varchar(50) NOT NULL,  
       salary float, PRIMARY KEY(id)  
);  
  
INSERT INTO persons (id, name) VALUES (1, 'John Doe');  
UPDATE persons SET salary=2000 WHERE id=1;

Còn đối với một lập trình viên Java, các dòng mã có thể là:

public class Person {  
     public String name;  
     public float salary;  
     public Person(String name) { ... }  
 }  
 Person p = new Person("John Doe");  
 PersistenceLayer.save(p);  
 p.setSalary(2000);  
 PersistenceLayer.update(p);

Chúng ta quan sát thấy rằng, về mặt nghiệp vụ thì cả 2 góc nhìn này đều thực hiện các công việc giống nhau, nhưng mỗi bên lại sử dụng các công cụ (ngôn ngữ, câu lệnh) khác nhau để thực hiện việc này.

Một lập trình viên có thể sẽ gặp khó khăn khi thao tác trực tiếp với CSDL sử dụng ngôn ngữ SQL. Do đó, ORM sẽ làm cầu nối giữa công cụ của lập trình viên (là ngôn ngữ lập trình) với công cụ của người quản trị CSDL (là ngôn ngữ SQL). Bằng cách sử dụng ORM, lập trình viên sẽ không cần quan tâm chi tiết đến các câu lệnh truy xuất trực tiếp đến CSDL, thay vào đó, lập trình viên sẽ làm việc với CSDL thông qua mã nguồn lập trình.

Ưu điểm của ORM:

* Quản lý dữ liệu tập trung trong code
* Các thao tác với dữ liệu được thực hiện tự động
* Tránh được các lỗi cú pháp SQL
* Hỗ trợ giao dịch (transaction)
* Có thể cache dữ liệu để truy xuất nhanh hơn

Nhược điểm của ORM

* Đối với các dự án lớn, hiệu năng truy xuất dữ liệu thường hạn chế
* Lập trình viên dễ bị rơi vào bẫy truy xuất dữ liệu quá nhiều (vì quá dễ để truy xuất), ảnh hưởng đến hiệu năng của hệ thống
* Đối với các thao tác phức tạp, có thể cần đến việc sử dụng SQL thuần

Một số ORM framework dành cho Java

* Apache Cayenne
* ActiveJDBC
* EclipseLink
* Ebean
* OpenJPA
* Hibernate
* JPA (Java Persistence API)

Last modified: Wednesday, 27 February 2019, 3:01 PM