HIENLTH

HUTECH
University

HUTECH

Đại học Công nghệ Tp.HCM

Chủ đề 7: Thiết kế tầng dữ liệu

Thiết kế tầng quản lý dữ liệu

Thiết kế tầng quản lý dữ liệu (data management layer design) gồm 4 bước:

- Chọn dạng lưu trữ
- Ánh xạ các lớp đối tượng cần lưu trữ xuống dạng lưu trữ đã chọn
- Tối ưu hóa việc lưu trữ
- Thiết kế các lớp đối tượng phục vụ cho việc truy xuất và chỉnh sửa dữ liệu

Các dạng lưu trữ dữ liệu

- Lưu trữ dưới dạng file (truy xuất tuần tự hoặc truy xuất ngẫu nhiên).
- Lưu trữ bằng CSDL quan hệ (Relational Database).
- Lưu trữ bằng CSDL lai đối tượng – quan hệ (Object – Relational Database).
- Lưu trữ bằng CSDL hướng đối tượng (Object Oriented Database)

Lưu trữ dưới dạng file

- Có 2 cơ chế truy xuất: tuần tự và ngẫu nhiên.
- Thường được sử dụng cho các trường hợp sau:
  - Lưu trữ 1 đối tượng duy nhất
    - Ví dụ: đối tượng config của hệ thống
  - Dữ liệu nếu có insert thì chỉ cần insert vào cuối
    - Ví dụ: mailing list, history
  - Dữ liệu tĩnh
    - Ví dụ: mã các quốc gia
- Ưu điểm: thư viện truy xuất file thường được hỗ trợ sẵn trong hầu hết các môi trường lập trình, cách thức truy xuất đơn giản
  - C#: StreamReader, StreamWriter, FileStream
  - C++: ifstream, ofstream
  - C: FILE
- Khuyết điểm:
  - Không giải quyết vấn đề truy xuất đồng thời
  - Không đảm bảo toàn vẹn dữ liệu

Lưu trữ bằng CSDL quan hệ

- Được phát triển bởi E. F. Codd vào thập niên 70 và được phát triển rộng rãi từ đầu thập niên 80
- Ưu điểm:
  - Vấn đề quản lý, phân quyền, truy xuất đồng thời sẽ do hệ CSDL đảm nhận
  - Sử dụng ngôn ngữ chung SQL cho tất cả các CSDL quan hệ
- Khuyết điểm:
  - Mô hình thực thể kết hợp không đúng bằng sơ đồ lớp

Lưu trữ bằng CSDL đối tượng – quan hệ

- Các hệ CSDL đối tượng – quan hệ bản chất là hệ CSDL quan hệ có hỗ trợ thêm 1 số tính chất của đối tượng (ví dụ là user-defined type)
- Một số tính chất của hướng đối tượng vẫn chưa có hỗ trợ trong các hệ này
- Các hệ CSDL đối tượng – quan hệ như: DB2, Oracle, Informix,…
- Ưu điểm:
  - Có những ưu điểm của CSDL quan hệ
  - Giúp giải quyết 1 số bài toán hướng đối tượng đơn giản
- Khuyết điểm
  - Lược đồ CSDL vẫn chưa hoàn toàn đúng bằng sơ đồ lớp

Lưu trữ bằng CSDL hướng đối tượng

- Có 2 kiểu CSDL
  - Tích hợp vào trong ngôn ngữ hướng đối tượng
  - Hệ CSDL riêng biệt hỗ trợ ngôn ngữ truy vấn được chuẩn hóa:
    - ODL: Object Definition Language
    - OML: Object Manipulating Language
    - OQL: Object Query Language
- Ưu điểm:
  - Hỗ trợ tất cả các tính chất của hướng đối tượng
- Khuyết điểm:
  - Vẫn trong giai đoạn phát triển

Nội dung

- Chuyển đổi sơ đồ lớp sang mô hình dữ liệu quan hệ
  - ✓Chuyển đổi đối tượng
  - ✓Chuyển đổi mối quan hệ

Các luật chuyển đổi

- Luật 1: **Chuyển đổi tất cả các lớp đối tượng sang các bảng**. Nếu một lớp trong sơ đồ có nhiều lớp con, chuyển đổi các lớp (bao gồm lớp cha, lớp con) sang các bảng trong lược đồ CSDL quan hệ (luật 8).
- Luật 2: **Chuyển đổi các thuộc tính đơn trị thành các thuộc tính (cột) trong bảng dữ liệu**.
- Luật 3: **Chuyển đổi các phương thức thành các thủ tục** (store-procedure) hoặc các module chương trình.

Các luật chuyển đổi (tt)

- Luật 4: **Chuyển đổi các mối quan hệ đơn trị** (1-1) bằng cách lấy thuộc tính khóa của một lớp lưu trữ vào lớp còn lại (tương tự các thêm khóa ngoại cho một bảng dữ liệu).
- Luật 5: **Chuyển đổi các thuộc tính đa trị và các nhóm** lặp thành các bảng dữ liệu mới và tạo ra mối quan hệ 1-n từ bảng gốc đến bảng mới.
- Luật 6: **Chuyển đổi các mối kết hợp n-n** bằng cách tạo ra bảng dữ liệu mới liên quan đến 2 bảng dữ liệu gốc. Lưu trữ thuộc tính khóa chính của 2 bảng dữ liệu gốc vào bảng dữ liệu mới như là khóa ngoại.

Các luật chuyển đổi (tt)

- Luật 7: **Chuyển các mối quan hệ phối hợp 1-n** bằng cách lấy thuộc tính khóa của lớp có giá trị đơn (0..1, 1..1) lưu vào lớp có giá trị nhiều (1..*, 0..*) thành thuộc tính như khóa ngoại.

Các luật chuyển đổi (tt)

- Luật 8: **Chuyển các mối quan hệ kế thừa**.
  - 8a: Sử dụng cả lớp cha (superclass) và lớp con (subclass)
  - 8b: Sử dụng lớp con
  - 8c: Sử dụng lớp cha

Mối quan hệ kế thừa

- Trong mô hình CSDL quan hệ, hiện tại có 3 phương pháp để mô hình hóa mối quan hệ kế thừa
- Tuy nhiên mỗi phương pháp đều có nhược điểm riêng

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType

Cách 1

SECRETARY

SSN	TypingSpeed

TECHNICIAN

SSN	TGrade

ENGINEER

SSN	EngType

Cách 2

CAR

VehicleId	LicensePlateNo	Price	MaxSpeed	NoOfPassengers

TRUCK

VehicleId	LicensePlateNo	Price	NoOfAxles	Tonnage

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	EngType

Serving as the type attribute

Cách 3

Tối ưu việc lưu trữ

- Vấn đề về primary key & kiểu dữ liệu
- Vấn đề DeNormalization
- Vấn đề Clustering & Indexing
- Vấn đề Security & Backup
- Vấn đề Policy & Culture

Thiết kế các lớp đối tượng phục vụ cho việc truy xuất và chỉnh sửa dữ liệu

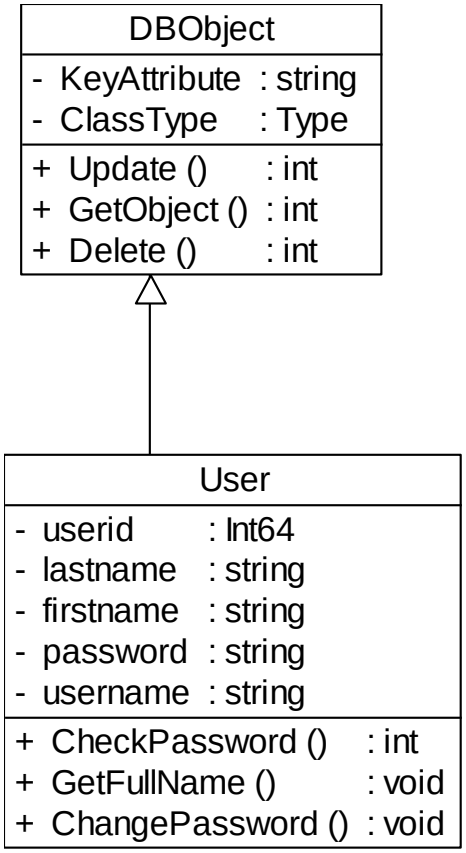
Double Duty Class

- Lớp đảm đương 2 nhiệm vụ: business & database operate
- Vi phạm tính cohesion
- Nếu đổi loại CSDL thì toàn bộ lớp phải viết lại

```
class User
{   private string firstname;
    private string lastname;
    private string password;
    public User(string userid)
    {   SqlConnection conn = new SqlConnection(".......");
        conn.Open();
        SqlCommand cmd =
              new SqlCommand(String.Format(
              "SELECT * FROM [user] WHERE userid = {0}", userid));
        SqlDataReader reader = cmd.ExecuteReader();
        firstname = ...;
        lastname = ....;
        conn.Close();
    }
    public void ChangePassword(string newpwd)
    {   password = newpwd;
        SqlConnection conn = new SqlConnection(".......");
        conn.Open();
        SqlCommand cmd =
              new SqlCommand(String.Format(
          "UPDATE * FROM [user] SET ..... WHERE userid = {0}", userid));
        int r = cmd.ExecuteNonQuery();
        conn.Close();
    }
}
```

Database Object

- Sử dụng 1 Database Object đảm nhận việc lưu trữ cho tất cả các object.
- Cần sử dụng cái khái niệm lập trình tổng quát như: delegate, reflection để xây dựng
- Cần giải quyết vấn đề Primary Key gồm nhiều thuộc tính
- Cần giải quyết mối quan hệ với khác lớp khác (khi lưu xuống database, nếu object này có mối quan hệ với object kia thì object kia có cần phải lưu hay không)
- Việc xây dựng DatabaseObject rất phức tạp. Do đó có thể sử dụng các thư viện có sẵn. Ví dụ: Enity Framework



Broker object

- Xây dựng các lớp trung gian đảm nhận việc lưu trữ cập nhật dữ liệu

```
class User
{
    private string firstname;
    private string lastname;
    private string password;
    private string username;
    private Int64 userid;
    public void ChangePassword(string newpwd)
    {
        password = newpwd;
        Users.UpdateUser(this);
    }
}
class Users
{
    public static User GetUser(string userid) { }
    public static User GetUser(string username, string password) { }
    public static void UpdateUser(User user) { }
}
```

Phương án giải quyết

- Nếu ứng dụng nhỏ có thể sử dụng các phương pháp double duty, broker
- Nếu ứng dụng cần phát triển nhanh có thể dùng các thư viện có sẵn như Entity Framework, JPA (Java Persistence API)
- Có thể phối hợp với phương pháp Broker Object

Tham khảo

- http://www.agiledata.org/essays/mappingObjects.html
- Slide bài giảng *Phân tích Thiết Kế Hướng đối tượng*, TS. Nguyễn Trần Minh Thư, ĐH KHTN TpHCM.

Câu hỏi và thảo luận

Thank you!!!