UIT VNUHCM University of Information Technology

CHƯƠNG 5:
MỘT SỐ MÔ HÌNH CSDL TIÊN TIẾN:
CSDL HƯỚNG ĐỐI TƯỢNG

Khoa Khoa học và kỹ thuật thông tin
Bộ môn Thiết bị di động và Công nghệ Web

Nội dung

1. Đặt vấn đề.
2. Các tính chất của CSDL hướng đối tượng.
3. Chuyển đổi CSDL quan hệ sang CSDL Hướng đối tượng.

ĐẶT VẤN ĐỀ

Ưu/nhược điểm của CSDL quan hệ

- Nền tảng toán học vững.
- Nhiều hệ QTCSDL hỗ trợ.
- Mô hình dữ liệu, khái niệm đơn giản, dễ hiểu.
- Phù hợp khá nhiều các ứng dụng.
- Ngôn ngữ truy vấn SQL dễ hiểu, đáp ứng các nhu cầu người dùng.
- Có thể phục hồi dữ liệu khi gặp sự cố.
- Dựa trên nền tảng dạng chuẩn.

- Ngôn ngữ truy vấn thiếu: rẻ nhánh, vòng lặp.
- Không hỗ trợ các kiểu dữ liệu phức tạp: cấu trúc, tập hợp, ảnh, tri thức. . .
- Biểu diễn dữ liệu trái với sự biểu diễn của thực tiễn.

Giải quyết

- Hướng giải quyết
  - + Mở rộng CSDLQH:
    - Mô hình ERD
      - Hình thành khái niệm: TQH và CBH.
  - + Đề xuất mô hình mới:
    - Bỏ dạng chuẩn 1.
    - Biểu diễn các đối tượng phức tạp.
- Gồm 2 trường phái:
  - + NNLTHDT tạo tiền đề để hiệu chỉnh, xây dựng CSDL HDT.
  - + CSDLHDT tạo tiền đề để xây dựng NNLTHDT .
  - ➔ TP2 có vẻ thắng thế.

ĐẶC ĐIỂM CỦA CSDL HƯỚNG ĐỐI TƯỢNG

CÁI KHÁI NIỆM

- Đối tượng (object): là sự kết hợp giữa dữ liệu và các hành vi mô tả cho một thực thể
- Tính chất (Property): đặc trưng của một đối tượng được chỉ định bằng một tên có thể ứng với một thuộc tính, một hàm hay một đối tượng con thành phần.
- Ví dụ:
  + Thuộc tính đơn: tên của một người, ...
  + Hàm: Hàm tuổi (của một người), ...
  + Thuộc tính kép: các con của một người, ...

Đối tượng và phương thức

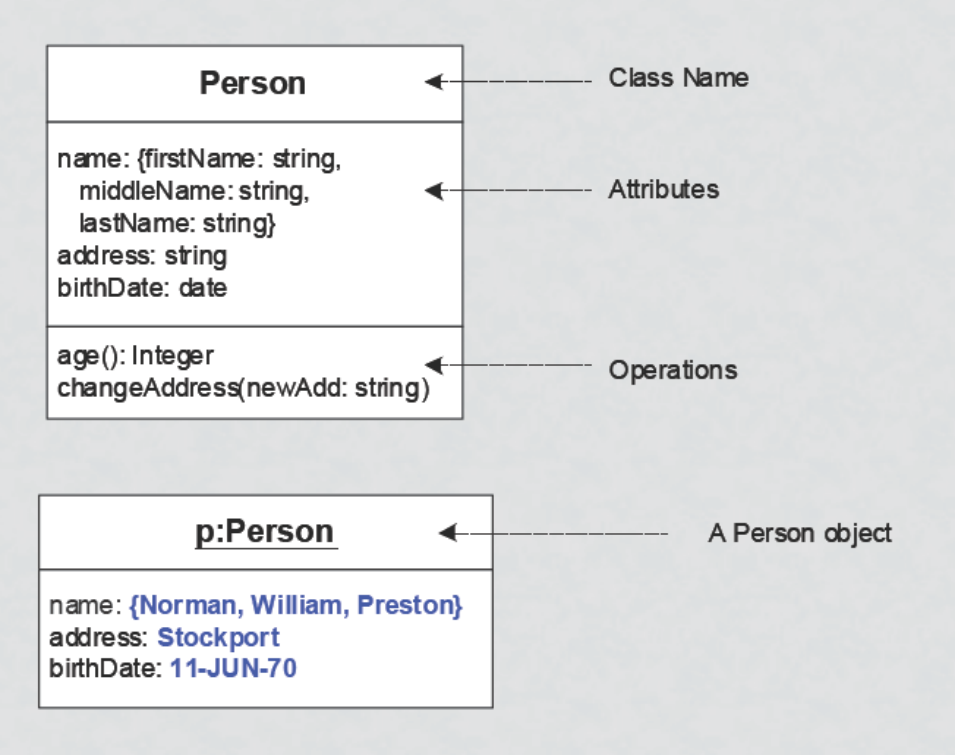
Đối tượng (object)

- Các đối tượng có cùng tính chất, được đặc trưng bởi một cấu trúc và tập các phép toán tác dụng lên các đối tượng của lớp bằng cách che dấu cấu trúc.
- Việc đặc tả tiến triển của các lớp đối tượng làm thành một CSDL hướng đối tượng, cho phép mô hình hoá hành vi chung của các đối tượng một cách đơn thể và mở rộng được.

Phương thức (method)

- Thao tác liên kết với một lớp, xử lý hay đưa trả lại trạng thái của một đối tượng hay một phần của đối tượng thuộc lớp.
- Một đối tượng được thao tác bởi phương thức của lớp và được thấy qua các phương pháp: nguyên lý bao gói. Phương thức có thể áp dụng được cho nhiều đối tượng thuộc các lớp khác nhau: đa lớp → dùng để mô hình hoá các mối liên kết giữa các lớp.

Đối tượng và phương thức



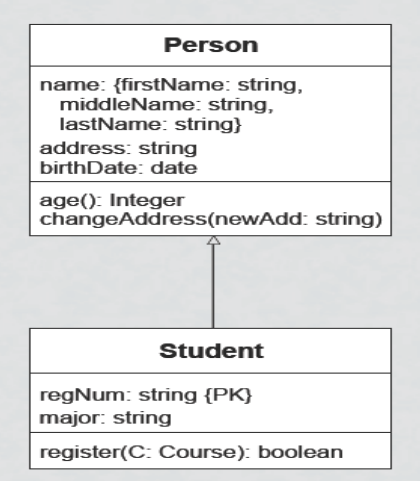
Các tính chất của hướng đối tượng

- Khái quát hóa:
  - + Liên kết phân cấp giữa hai lớp xác định rằng các đối tượng của lớp trên tổng quát hơn các đối tượng của lớp dưới, các đối tượng của lớp dưới có các tính chất đầy đủ và tinh tế hơn.
- Tính kế thừa:
  - + Sự truyền tính chất của một lớp cha tới lớp con của nó. Mọi phần tử của lớp con kế thừa các tính chất của lớp trên. Một số tính chất của lớp con có thể được làm tinh tế hơn (định nghĩa lại).
- Tính kế thừa bội:
  - + Cho phép một lớp có nhiều lớp trên trực tiếp. Lớp con kế thừa các tính chất và phương pháp của các lớp trên. Có thể xảy ra và cần được giải quyết những xung đột về tên các tính chất hay phương pháp

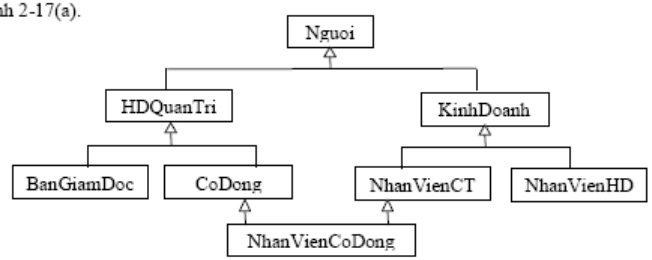
Các loại thừa kế

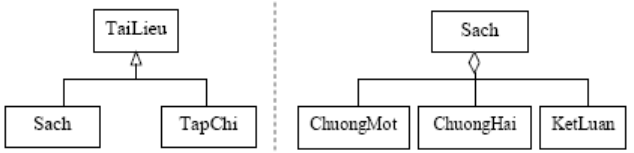
- Có 2 loại thừa kế
  - + Thừa kế ISA: kiểu con thừa kế từ Interface => chỉ thừa kế hành vi
  - + Thừa kế EXTENDS : kiểu con thừa kế từ Class => cho thừa kế cả hành vi và đặc trưng

Ví dụ về thừa kế



hình 2-17(a).





Hình 2-16 Quan hệ tổng quát hoá ngược lại với quan hệ kết tập

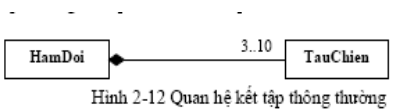
Class diagram

– Sơ đồ lớp

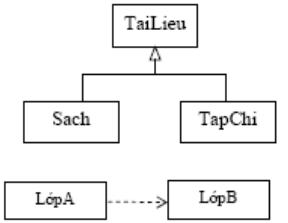
+ Các mô hình đối tượng thường phân biệt các tính chất được phân chia bởi nhiều lớp và hợp chúng trong những các mối liên kết.

+ Các dạng của mối liên kết

- Tương tự các LK trong ERD
- Các liên kết khác



Hình 2-12 Quan hệ kết tập thông thường



Các kiểu dữ liệu mới

- Bộ (tuple): cho phép gộp các thuộc tính (tích Đề các).
- Tập hợp (set): cho phép định nghĩa các đối tượng không sắp thứ tự, không chứa các phần tử giống nhau.
- Túi (bag): các tập không sắp thứ tự, có các phần tủ giống nhau.
- Danh sách (list): cho phép định nghĩa các đối tượng có thứ tự, được phép có các phần tử giống nhau.
- Bảng (table): các thực thể có thứ tự và có chỉ số.

Các ứng dụng của CSDL hướng đối tượng

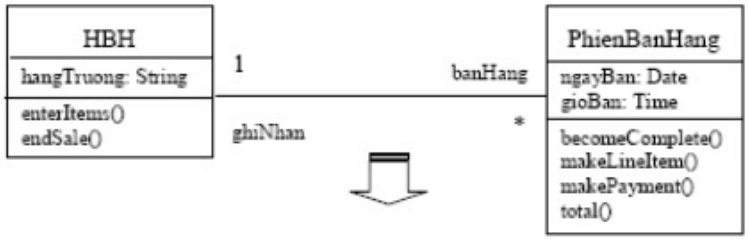
- Những ứng dụng thiết kế công nghệ. VD: CAD (*CAD*: *Computer-Aided Design*), CAM (*CAM: Computer-Aided Manufacturing*), CIM (*CIM: Computer-Integrated Manufacturing*).
- Các ứng dụng đa phương tiện (Multimedia).
- Các cơ sở tri thức.
- Những ứng dụng đòi hỏi xử lý phân tán và tương tranh.
- Các phần mềm nhúng.

CHUYỂN ĐỔI TỪ CSDL QUAN HỆ SANG CSDL HƯỚNG ĐỐI TƯỢNG

VÍ DỤ VỀ CSDL HĐT (1)

```
Class CuaHang
{
  attribute String(30)
  tenGoi;
  attribute struct diaChi{
      char(3) soPho,
      char(20) tenPho,
      char(15) tinhThanh
  };
  void addSale(){…}
};
```

VÍ DỤ VỀ CSDL HĐT (2)

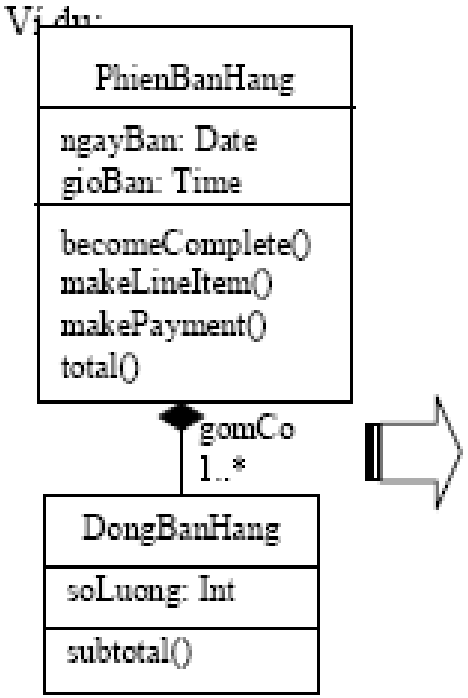



```
Class PhienBanHang
{
        attribute Date ngayBan;
        attribute Time gioBan;
        relationship HBH
        banHang
        inverse ghiNhan::HBH;
        Boolean becomeComplete(){…}
        void makeLineItem()
        makePayment(){…}
        Number total(){…}
};
```

```
Class HBH
{
        attribute String(25) hangTruong;
        attribute String(15) tenTruong;
        relationship set<PhienBanHang>
        ghiNhan
        inverse banHang::PhienBanHang;
        void HBH();
        void enterItems(){…}
        void endSale(){…}
};
```

VÍ DỤ VỀ CSDL HĐT (3)

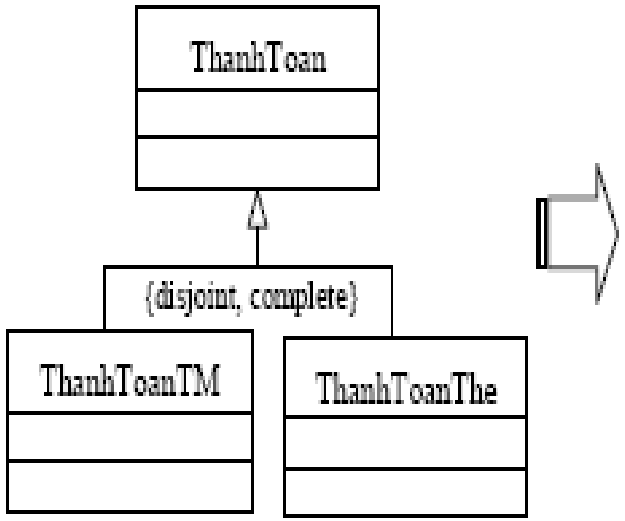
```
Class PhienBanHang
{
        attribute Date ngayBan;
        attribute Time gioBan;
        attribute List<DongBanHang> gomCo;
        Boolean becomeComplete(){…}
        void makeLineItem()
        void makePayment(){…}
        Number total(){…}
};
Class DongBanHang
{
        attribute Integer soLuong;
        Number subtotal(){…}
};
```



VÍ DỤ VỀ CSDL HĐT (4)

```
Class ThanhToanTM extends ThanhToan
(extent ThanhToanTM )
{
        …
};

Class ThanhToanThe extends ThanhToan
(extent ThanhToanThe)
{
        …
};
```



So sánh hai hệ QT

Table 2. Comparing OODBMS with RDBMS considering their objectives

OODBMS	RDBMS
• Main objectives: data encapsulation and independence.	• Main objective: ensuring data independence from application programs.
• Independence of classes: classes can be reorganized without affecting the mode of using them.	• Data independence: Data can be reorganized and modified without affecting the mode of using them.
• OODBMS store data and methods.	• RDBMS store only data.
• Encapsulation: the data can be used only through their classes' methods.	• Data partitioning: data can be partitioned depending on the requirements of the users and on the specific users applications.
• Active objects: the objects active. Requests cause objects to execute their methods.	• Passive data: the data are passive. Certain operations, which are limited, can be automatically brought into use when the data are used.
• Complexity: the structure of data may be complex, involving different types of data.	• Simplicity: users perceive data as columns, rows/tuples and tables.
• Chained data: data can be chained so that the methods of classes may bring about increased performance. Structured data such as BLOBS (binary large objects) are used for sound, image, video etc.	• Separate Tables: each relation/table is separate. The Join Operator refers data from separate tables.
• Non-redundancy of methods: data and methods non-redundancy is achieved through encapsulation and inheritance. Inheritance helps to reduce the redundancy of methods.	• Data non-redundancy: data normalization aims at eliminating or reducing data redundancy. It is used in the stage of designing the database and not in the stage of developing the applications.
• Optimizing classes: the data for an object can be interrelated and stored together, so that they may all be accessed by the access mechanism.	• RDBMS performance is related to the level of complexity of the data structure.
• Consistent conceptual model: the models used for analysis, designing, programming and accessing the database are similar. The classes of objects directly represent the concepts of applications.	• Different conceptual model: the model of data structure and data access represented by tables and JOINS is different from the model of analysis, designing and programming. The project must be converted in relational and access tables in accordance with SQL.

Chuyển từ mô hình quan hệ sang hướng đối tượng

- Bước 1: Xác định đối tượng (thực thể) từ mô hình quan hệ.
- Bước 2: Xác định quan hệ giữa các đối tượng.
- Bước 3: Xây dựng mô hình hướng đối tượng:
  - + Để biểu diễn cho 1 đối tượng, ta dùng kiểu tuple (bộ).
  - + Để biểu diễn cho nhiều đối tượng, ta dùng list (danh sách) hoặc set (tập hợp):
    - list: có thứ tự, trùng nhau.
    - set: không sắp thứ tự, không trùng nhau.

VÍ DỤ: CHUYỂN CSDL SAU SANG HƯỚNG ĐỐI TƯỢNG

Tacgia (#mstg,	tentg,	sdt,	email)	
TG01	Nguyen A	098731	a@gmail.com	
TG02	Nguyen B	098731	b@gmail.com	
Sach (#mssach,	tensach,	sotrang,	sotien,	msnxb)
S01	ABC	6	100000	DHQG
Nxb (#msnxb,	tennxb,	sdt-xb,	email-xb)	
NXB01	NXB-DHQG	0844643	nxb@gmail.com	
Tg-Sach(#mstg,	#mssach,	nam-xb)		
TG01S01	2019			
TG02S01	2020			

CHUYỂN TỪ QUAN HỆ SANG HĐT MÔ HÌNH

QUAN HỆ

Tacgia (#mstg, tentg, sdt, email)

Sach (#mssach, tensach, sotrang, sotien, msnxb)

Nxb (#msnxb, tennxb, sdt-xb, email-xb)

Tg-Sach(#mstg, #mssach, nam-xb)

Số thực thể là 3. Gồm: tác giả, sách, Nxb:

sách-nxb: quan hệ n-1.

tác giả-sách: quan hệ: n-n.

HƯỚNG ĐỐI TƯỢNG

TacgiaObj(mstg, tentg, sdt, email, **set(SachObj, nam-xb)**)

SachObj(mssach, tensach, sotrang, sotien,

tuple(NxbObj), **set(TacgiaObj, nam-xb)**)

NxbObj(msnxb, tennxb, sdt-xb, email-xb, ***set(SachObj)***)

CHUYỂN TỪ QUAN HỆ SANG HDT
MÔ HÌNH + Dữ liệu

TacgiaObj(mstg, tentg, sdt, email, **set(SachObj), nam-xb**)

TG01	Nguyen A	098731	a@gmail.com	{(S01, ABC, 6, 100000, DHQG), 2019}
TG02	Nguyen B	098731	b@gmail.com	{(S01, ABC, 6, 100000, DHQG), 2020}

SachObj(mssach, tensach, sotrang, sotien, tuple(NxbObj)), **set(TacgiaObj), nam-xb**)

S01	ABC	6	100000	(NXB01, NXB-DHQG, 0844643, nxb@gmail.com)	[{(TG01, Nguyen A, 098731, a@gmail.com), 2019}, {(TG02, Nguyen B, 098731, b@gmail.com), 2020}

NxbObj(msnxb, tennxb, sdt-xb, email-xb, set(SachObj))

NXB01	NXB-DHQG	0844643	nxb@gmail.com	{ (S01, ABC, 6, 100000) }

TÀI LIỆU THAM KHẢO

1. Nguyễn Gia Tuấn Anh, Trương Châu Long, *Bài tập và bài giải SQL Server*, NXB Thanh niên (2005).
2. Đỗ Phúc, Nguyễn Đăng Tỵ, *Cơ sở dữ liệu*, NXB Đại học quốc gia TPHCM (2010).
3. *Nguyễn Gia Tuấn Anh, Mai Văn Cường, Bùi Danh Hường, Cơ sở dữ liệu nâng cao*, NXB Đại học quốc gia TPHCM (2019).
4. Itzik Ben-Gan, *Microsoft SQL Server 2012- TSQL Fundamentals*.


Q
A

BÀI TẬP

Bài 1. Chuyển mô hình sau sang hướng đối tượng

Khach(#msk, tenk, sdt, email)

Nhanvien(#msnv, tennv, sdt-nv, email-nv)

Mathang(#msmh, tenmh)

Hoadon(#mshd, ngayhiod, tongtien, mskh, msnv)

CTHD(#mshd, #msmh, SL, DG).

Bài 2. Thực nghiệm so sánh các thao tác: insert, update, delete, select và tính thời gian thực thi cho bài 1, 2.

Bài 3. Tại sao CSDLĐPT thích hợp với CSDLHDT, cho ví dụ minh họa cho câu trả lời.