

# Cơ sở dữ liệu nâng cao

---

## Cơ sở dữ liệu hướng đối tượng

Phạm Nguyên Khang, Nguyễn Thanh Bình

Đỗ Thanh Nghị

{pnkhang|ntbinh|dtngghi}@cit.ctu.edu.vn

Cần Thơ

13-12-2010

# CSDL hướng đối tượng

---

## ■ Mục tiêu

- Trình bày HQTCSDL hướng đối tượng
- Giới thiệu các thành phần của HQTCSDL hướng đối tượng
- Tìm hiểu sâu về chuẩn HQTCSDL hướng đối tượng

# Tài liệu tham khảo

---

**[Connolly, 1998]**

Connolly T., Begg C., Strachan A. (1998)

Database Systems

A Practical Approach to Design, Implementation and Management

Addison Wesley

Chapitres 22

**[Cattell, 1997]**

Cattell R. et al. (1998)

Object Database Standard: ODMG 2.0

Morgan Kaufmann

# Tiếp cận HQTCSDL hướng đối tượng

---

- **sử dụng cách tiếp cận thuần đối tượng**
  - từ phân tích đến cài đặt, và lưu trữ suy nghĩ hướng đối tượng quản lý các đối tượng
  - từ giao diện người dùng đến ngôn ngữ lập trình
- **phát triển hệ thống mới**
  - « quên đi » các hệ CSDL quan hệ
  - thao tác, lưu trữ và quản lý các đối tượng
- **định nghĩa yêu cầu cho các hệ thống này**
  - Các thành phần của HQTCSDL hướng đối tượng
- **đề xuất chuẩn**
  - Chuẩn của HQTCSDL hướng đối tượng

# Định nghĩa

---

## ■ HQTCSDL hướng đối tượng

- **Hệ thống bao gồm:**
  - các chức năng của một HQTCSDL
  - tích hợp khái niệm đối tượng
- **nhằm mô hình hóa**
  - các đối tượng phức tạp
  - thiết kế thuần nhất: dữ liệu và chương trình
  - Khái niệm lớp và thừa kế
- **từ góc nhìn hệ thống**
  - lưu trữ hiệu quả tất cả các kiểu dữ liệu
  - tìm kiếm
  - quản lý giao dịch

# **Các thành phần của HQTCSDL HĐT**

# Tính năng bắt buộc

---

- đối tượng phức
- định danh đối tượng
- đóng gói
- kiểu hoặc lớp
- định nghĩa chồng và ràng buộc muộn
- quản lý cạnh tranh
- đầy đủ
- khả năng mở rộng
- thao tác dữ liệu
- lưu trữ lâu dài
- tin cậy
- thừa kế
- quản lý bộ nhớ ngoài

# Đối tượng phức

---

## ■ Định nghĩa

- Một đối tượng phức là đối tượng có một hai nhiều thuộc tính tham chiếu đến đối tượng khác hoặc đến một tập các giá trị.
- Một đối tượng có thể chứa nhiều đối tượng con

## ■ Ví dụ

- Một căn hộ bao gồm nhiều phòng

o451 : tuple ( No : 10,  
Address : "71 Pins",  
Nb-Room : 4;  
Rooms : set (o5, o6, o7, o8)  
Price : 35 000 )



# Định danh đối tượng

---

## ■ định nghĩa

- Tất cả các đối tượng phải có 1 định danh (OID) duy nhất cho phép tham chiếu đến nó. Định danh này bất biến trong suốt chu trình sống của đối tượng.

## ■ ví dụ

- Định danh của đối tượng "Apartment số 10" là: o451

## ■ khái niệm liên quan

- tính đồng nhất của đối tượng

hai đối tượng là một nếu chúng có cùng định danh

## ■ HQTCSDL HĐT phải hỗ trợ OID và tính đồng nhất của đối tượng

# Đóng gói

---

## ■ định nghĩa

- không được thao tác trực tiếp trên giá trị của đối tượng, chỉ có thể thao tác thông qua các phương thức được cho phép. Đây là nguyên lý của tính đóng gói.

## ■ ví dụ

- không được thao tác trực tiếp trên các thuộc tính của đối tượng
- Sử dụng các hàm để lấy (hoặc đặt lại) giá trị của thuộc tính

## ■ HQTCSDL HĐT phải hỗ trợ tính năng đóng gói

# Kiểu hoặc lớp

---

## ■ định nghĩa: Kiểu

- kiểu mô tả cách mà các giá trị được thao tác, cho phép kiểm tra tính hợp lệ của kiểu đối với phép toán nào đó

## ■ định nghĩa: Lớp

- lớp bao gồm tập các đối tượng có cùng các thuộc tính và hành vi chung. Lớp mô tả cấu trúc đối tượng và hành vi của nó và cho phép tạo ra đối tượng của lớp này.

## ■ HQTCSDL HĐT phải hỗ trợ Kiểu hoặc Lớp hoặc hỗ trợ cả 2

# Thừa kế

---

## ■ định nghĩa

- sự phân cấp kiểu hoặc lớp cho phép mô tả kiểu con hoặc lớp con thừa kế các đặc tính (cấu trúc và hành vi) của kiểu cha hoặc lớp cha.

## ■ Ví dụ

- Person (người), Prop (người chủ sở hữu), Locator (người thuê nhà)

## ■ HQTCSDL HĐT phải hỗ trợ thừa kế

# Định nghĩa chồng và ràng buộc muộn

---

- **overriding : định nghĩa lại**
  - phương thức được định nghĩa cho mỗi trường hợp sử dụng
- **overloading : nạp chồng**
  - sử dụng một tên chung (của phương thức) cho nhiều bản cài đặt khác nhau
- **late binding : ràng buộc muộn**
  - hệ thống xác định phương thức nào được sử dụng lúc thực thi (chứ không phải lúc biên dịch)
- **HQTCSDL HĐT phải cho phép định nghĩa lại các phương thức**

# Tính đầy đủ và khả năng mở rộng

---

## ■ Đầy đủ

- khả năng biểu diễn của ngôn ngữ lập trình
- hỗ trợ tất cả các kiểu thao tác

## ■ Khả năng mở rộng

- khả năng mở rộng tập kiểu cơ sở
- cho phép xây dựng kiểu mới

# Tin cậy và cạnh tranh

---

## ■ Tin cậy

- cơ chế hoạt động của hệ thống phải được định nghĩa để có thể đối mặt với sự cố

## ■ Cạnh tranh

- hỗ trợ chia sẻ dữ liệu giữa các người dùng khác nhau

## ■ Các cơ chế này phải thích nghi cho từng kiểu đối tượng khác nhau

# Quản lý bộ nhớ ngoài

---

- **cho phép lưu trữ và truy xuất dữ liệu**
  - đối tượng lớn
  - đối tượng phức
- **cung cấp cơ chế truy xuất hiệu quả**
  - chỉ mục
  - gom nhóm dữ liệu
  - quản lý vùng đệm
  - phương pháp truy xuất



# Lưu trữ lâu dài và thao tác dữ liệu

---

## ■ Lưu trữ lâu dài

- khả năng lưu trữ đối tượng càng trong suốt càng tốt
- lưu trữ lâu dài phải độc lập với kiểu đối tượng

## ■ Ngôn ngữ thao tác dữ liệu

- ngôn ngữ cấp cao
- độc lập với ứng dụng
- thích nghi cho việc tối ưu hóa câu truy vấn

# Các thành phần của HQTCSDL HĐT

---

## ■ Tính năng tùy chọn

- đa thừa kế
- kiểm tra và suy diễn kiểu
- phân tán
- giao dịch thiết kế
- quản lý phiên bản

## ■ Tính năng mở

- họ ngôn ngữ lập trình
- hệ thống biểu diễn
- hệ thống kiểu
- tính đồng nhất của hệ thống



# **Chuẩn HQTCSDL HĐT**

# Chuẩn HQTCSDL HĐT

---

## ■ Đại diện các công ty sau:

GemStone Systems,  
IBEX Computing SA,  
O2 Technology,  
Object Design,  
Objectivity,  
POET Software,  
UniSQL  
Versant Object Technology

# Chuẩn HQTCSDL HĐT

---

## ■ Chuẩn

- OM: mô hình dữ liệu đối tượng
- ODL: ngôn ngữ định nghĩa dữ liệu
- OQL: ngôn ngữ thao tác dữ liệu
- Bindings: quan hệ với ngôn ngữ lập trình

# Mô hình đối tượng

# Mô hình đối tượng

---

## ■ Mô hình dữ liệu

- dựa trên mô hình đối tượng của OMG
- mở rộng hỗ trợ
  - trình bày các liên kết
  - khái niệm giao dịch

## ■ Khái niệm mô hình đối tượng

- nguyên thủy (literal), đối tượng
- kiểu: đặc tả và cài đặt
- kiểu cơ bản và phương thức xây dựng
- tính chất : thuộc tính và liên kết
- phương thức

# Nguyên thủy (Literal)

---

## ■ Định nghĩa

- một nguyên thủy là cặp  
    <tên, giá trị>
- có thể là giá trị nguyên tử hay phức hợp  
    <age, 24>; <FirstNames, (Adrien, Jacques, Joseph)>

## ■ Tính chất

- không sở hữu định danh
- những giá trị không đổi
- được tích hợp vào trong một đối tượng



# Đối tượng

---

## ■ Đối tượng

- thể hiện của một lớp
- một đối tượng có một tham chiếu duy nhất (định danh đối tượng)

OID

- được định nghĩa như một cặp  
<OID, nguyên thủy>
- cài đặt những giao diện (interfaces)  
cách xử lý

# Đối tượng

---

## ■ Đối tượng khác nhau

- tồn lưu (persistent)  
lưu trong CSDL
- nhất thời (transient)  
hiện diện trong bộ nhớ trong

## ■ Tất cả các đối tượng

- chia sẻ tập các phép toán cơ bản  
tạo, chép, xóa, lưu trữ...
- xuất thân từ kiểu gốc  
đối tượng

# Kiểu

---

## ■ Định nghĩa

- một kiểu có một đặc tả và một hay nhiều cài đặt

## ■ Đặc tả

- định nghĩa những đặc tính và những phép toán trên các thể hiện của kiểu

## ■ Cài đặt

- định nghĩa

những cấu trúc dữ liệu

những ngoại lệ

phương thức trên cấu trúc dữ liệu

# Kiểu và lớp

---

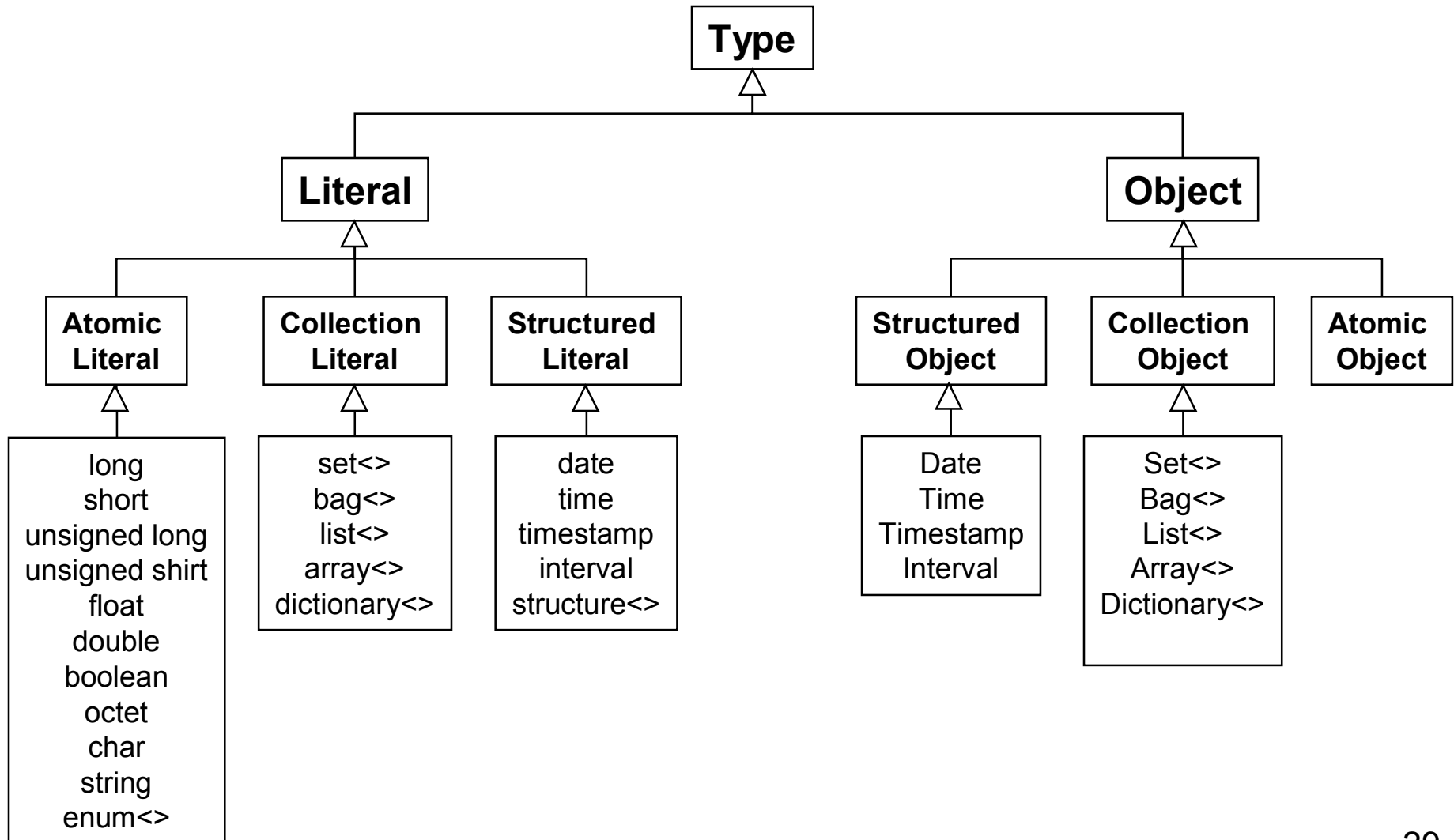
## ■ Lớp

- một lớp là sự kết hợp của đặc tả và cài đặt của kiểu
- lớp là khái niệm của việc cài đặt
- kiểu là khái niệm trừu tượng

## ■ Giao diện của một kiểu

- được tạo thành từ
  - tập hợp các kiểu trước đó (super-types)
  - mở rộng của kiểu (các thể hiện)
  - danh sách các khóa của kiểu

# Kiểu cơ bản và phương thức xây dựng



# Phương thức xây dựng

---

## ■ set

- tập hợp không có thứ tự, không có 2 phần tử giống nhau

## ■ bag

- tập hợp không thứ tự, cho phép 2 phần tử có thể giống nhau

## ■ list

- tập hợp có thứ tự, cho phép 2 phần tử có thể giống nhau

## ■ array

- mảng

## ■ dictionary

- dãy không thứ tự của các cặp <khóa, giá trị>, không có 2 phần tử có khóa giống nhau

# Các đặc tính

---

## ■ Thuộc tính

- cho phép ghi nhớ một giá trị hay một đối tượng (định danh đối tượng - OID)
- một thuộc tính không là một đối tượng, không có định danh

## ■ Liên kết (Association)

- cho phép định nghĩa những liên kết giữa các kiểu
- hiện chỉ có liên kết nhị phân 1:1, 1:N, M:N
- một liên kết không phải là một đối tượng
- có thể được thao tác nhờ vào đường dẫn traversal path

# Phương thức

---

- Một phương thức được định nghĩa bởi
  - tên
  - tên và kiểu của tham số đầu vào
  - tên và kiểu kết quả trả về
  - tên của các ngoại lệ



---

# **Ngôn ngữ định nghĩa đối tượng**

# Ngôn ngữ định nghĩa đối tượng

---

- Định nghĩa kiểu
- Định nghĩa lớp
- Định nghĩa phương thức
- Kế thừa
- Định nghĩa liên kết

# Định nghĩa kiểu

---

- **Mô tả cấu trúc và cách xử lý nhưng không có mở rộng (kế thừa)**

```
interface Person{ attribute string NAS;  
                  attribute string LastName;  
                  attribute string FirstName}
```

# Định nghĩa lớp

---

- **Mô tả cấu trúc, cách xử lý và có mở rộng (kế thừa)**

```
interface Prop:Person
  (extent Props;
  key NAS
  {relationship Set<Appartment> Posses inverse
  Appartment::Prop;
  short nbappart() raises (no_prop)}
```

---

# **Ngôn ngữ truy vấn đối tượng (OQL)**

# Ngôn ngữ truy vấn đối tượng (OQL)

---

- **OQL cho phép thao tác của những mở rộng của lớp**
  - đường dẫn truy xuất dữ liệu
  - thuộc tính, phương thức hay liên kết

# Ngôn ngữ truy vấn đối tượng

---

## ■ Ngôn ngữ thao tác dữ liệu

- xem lại cú pháp SQL
- viết câu truy vấn
  - kiểu xác định
  - kiểu duyệt, duyệt qua sự phân cấp của composition
- cho phép tạo ra kết quả

## ■ Tương thích với ngôn ngữ lập trình

- tích hợp ngôn ngữ truy vấn đối tượng OQL trong các ngôn ngữ lập trình
  - Smalltalk, C++, JAVA

# Biểu thức truy vấn

---

## ■ Truy vấn

SELECT DISTINCT	<expression>
FROM	<list of collections>
WHERE	<expression>
GROUP BY	<attributes> HAVING <condition>
ORDER BY	<expression>



# Biểu thức truy vấn

---

## ■ Danh sách tập dữ liệu (list of collections)

- định nghĩa biến trong câu truy vấn

FROM Props AS L

- định nghĩa biểu thức (clause IN)

những truy vấn hoàn thiện

cho phép truy vấn những quan hệ tạm thời

## ■ Cho phép thao tác

- mở rộng của một lớp
- có thể là một biểu thức mà kết quả cho ra một tập hợp

# Biểu thức truy vấn

---

## ■ Biểu thức có thể là

- **biểu thức cơ bản**

một lớp, một thuộc tính, một hàm...

- **biểu thức xây dựng**

hỗ trợ xây dựng struct, list, bag, set, array

biểu thức trên nhuwxntg tập hợp hay những đối tượng

- **những biểu thức nguyên tử**

nối chuỗi, trích chuỗi con...

## ■ Tính chất

- **số lượng lớn các biểu thức**

- **hoàn thiện về ngôn ngữ**

# Ngôn ngữ truy vấn đối tượng

---

## ■ Ví dụ 1 :

- Tìm kiếm NAS, LastName và Street của những chủ sở hữu ở Montréal

```
SELECT      L.NAS, L.LastName, L.Address.Str
FROM        Props L
WHERE       L.Address.City = "Montréal"
```

# Ngôn ngữ truy vấn đối tượng

---

## ■ Ví dụ 2:

- **Tìm số và địa chỉ của những căn hộ chung cư của chủ sở hữu Pierre Tremblay**

```
SELECT      A.No, A.Address
FROM        Props AS L, L.Posses AS A
WHERE       L.LastName="Tremblay"
AND         L.FirstName="Pierre"
```

- **phép kết nối được diễn đạt trong mệnh đề from nhờ vào định nghĩa tập hợp mới**

# Ngôn ngữ truy vấn đối tượng

---

- **Tìm kiếm đồng thời một đối tượng**
  - **Những HQTCSDL hướng đối tượng cho phép truy cập đồng thời một đối tượng**
    - duyet qua sự phân cấp
    - sử dụng định danh đối tượng (OID)

# Ngôn ngữ truy vấn đối tượng

---

## ■ Ví dụ

- truy xuất thông qua định danh đối tượng  
tìm giá của căn hộ chung cư có định danh đối tượng là o1

- o1->price

- truy xuất bằng duyệt từng phần tử của tập hợp các đối tượng của lớp apartment

Hiển thị giá của căn hộ chung cư

```
{  
    for (p in Apartment)  
        printf (p->price)  
}
```

# Ngôn ngữ truy vấn đối tượng

---

## ■ Duyệt qua sự phân cấp

- biểu thức nối kết không tường minh

tính diện tích phòng tắm của căn hộ chung cư tại địa chỉ  
31 Pins

```
SELECT      p.surface
FROM        a IN Apartments,
            p IN a.Rooms    !! jointure implicit !!
WHERE       a.address = "31 Pins"
AND         p.type = "Shower Room"
```

# Ngôn ngữ truy vấn đối tượng

---

## ■ Duyệt qua sự phân cấp

- biểu thức nối kết tường minh

Tìm chủ sở hữu mà họ ở cùng thành phố với nhau

```
SELECT      p1.lastname, p2.lastname, p1.city
FROM        p1 IN Props,
            p2 IN Props

WHERE       p1.nas <> p2.nas           !! jointure explicit !!
AND         p1.city = p2.city        !! jointure explicit !!
```



# Ví dụ

---

- **Tìm phòng diện tích nhỏ hơn 5m<sup>2</sup>**

```
SELECT    Rooms*  
WHERE     surface < 5
```

- **Tìm phòng diện tích nhỏ hơn 5m<sup>2</sup>, không phải phòng tắm**

```
SELECT    Rooms* difference ShowerRoom  
WHERE     surface < 5
```

# Ngôn ngữ truy vấn đối tượng

---

## ■ Sử dụng các phương thức

- có thể sử dụng phương thức của đối tượng trong truy vấn
- tiếp cận hàm, tiếp cận theo phương châm hàm hoàn toàn có thể được sử dụng
  - những thuộc tính được định nghĩa như những hàm
- một số phương thức có thể được định nghĩa từ lời của phương thức khác

## ■ Ví dụ

- tính diện tích căn hộ chung cư giá 30 000\$

```
O2          select  surface(a)
              from    a in apartment
              where    a.price = 30 000
```

# Tiến hóa của chuẩn ODMG

---

## ■ Nhiều phiên bản của chuẩn

**version 1:** ra đời 1994, bắt đầu làm việc 1991

**version 2:** ra đời 1995

**version 3:** ra đời 1997



Cám ơn !