

Cơ sở dữ liệu nâng cao

Cơ sở dữ liệu quan hệ - đối tượng

Đỗ Thanh Nghị
dtnghi@cit.ctu.edu.vn

Cần Thơ
11-10-2016

A decorative L-shaped line in a reddish-brown color, consisting of a vertical segment on the left and a horizontal segment extending to the right, framing the text.

Động lực

Động lực

- **Yêu cầu của ứng dụng mới**
 - đối tượng phức tạp
 - tích hợp dữ liệu và ứng xử
 - thao tác dữ liệu xuyên qua composition
- **Tích hợp với đối tượng hình thức**
 - đối tượng
 - phân tích, khái niệm, phát triển, CSDL
 - tồn lưu của dữ liệu trong môi trường đối tượng
- **Sự cần thiết**
 - mô hình dữ liệu cung cấp các khái niệm trên
 - hệ thống hỗ trợ cho các mô hình dữ liệu

Mô hình hóa dữ liệu

■ Mô hình dữ liệu

- cho phép trình bày thể giới thực
- tạo thành từ

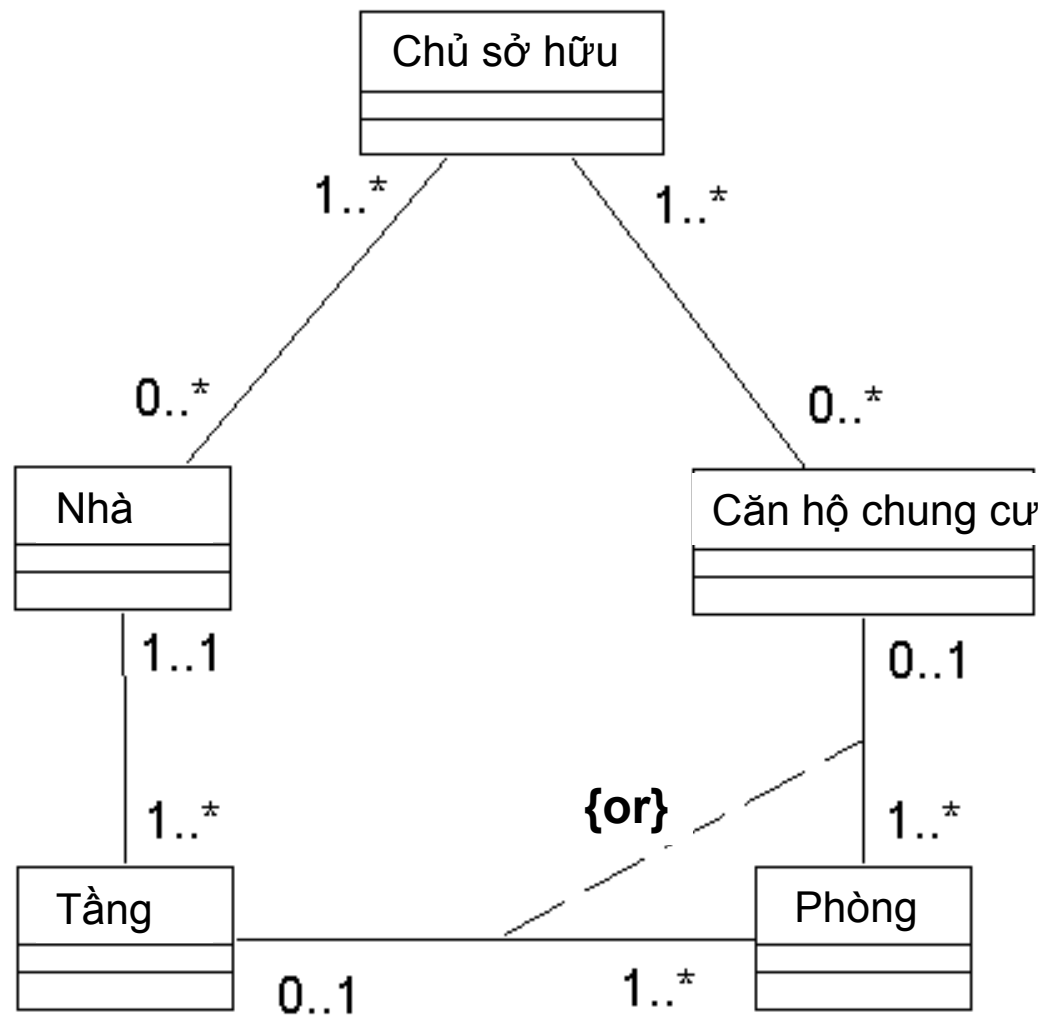
tập hợp các khái niệm

tập hợp các luật để quản lý khái niệm

■ Ví dụ về mô hình

- mô hình thực thể-liên kết
- mô hình ngữ nghĩa
- mô hình quan hệ
- mô hình đối tượng

Mô hình dữ liệu



Khái niệm mô hình thực thể-liên kết

■ Thực thể

- chủ sở hữu
- nhà
- tầng
- căn hộ chung cư
- phòng

■ Ràng buộc

- một căn hộ chung cư phải có ít nhất một phòng tắm
- một căn nhà có thể nhiều chủ sở hữu

Khái niệm mô hình thực thể-liên kết

■ Liên kết

- chủ sở hữu: nhà
- chủ sở hữu: căn hộ chung cư
- nhà: tầng
- tầng: phòng
- căn hộ chung cư: phòng

■ Kiểu khác nhau của liên kết

- composition
- tổng quát, chuyên biệt

Mô hình hóa dữ liệu

■ Mô hình quan hệ

- chỉ duy nhất khái niệm: quan hệ
- chỉ có dữ liệu được trình bày
- giới hạn: chuẩn 1 (1FN)
- bắt buộc: chuẩn hóa dữ liệu

Mô hình hóa dữ liệu

■ Ứng dụng mới

- trình bày dữ liệu và xử lý dữ liệu
- phân biệt thực thể và liên kết
- làm mịn liên kết

tổng quát, chuyên biệt
composition

■ Khả năng cải tiến

- mở rộng mô hình quan hệ
HQTCSDDL quan hệ - đối tượng
- giới thiệu mô hình mới
HQTCSDDL đối tượng

Mở rộng mô hình quan hệ

Mở rộng mô hình quan hệ

■ Mục tiêu

- đáp ứng yêu cầu ứng dụng mới
- tận dụng các mô hình quan hệ có sẵn
- hoàn thiện phần khiếm khuyết của mô hình quan hệ

■ Khiếm khuyết mô hình quan hệ

- nghèo nàn về kiểu dữ liệu
 - chuẩn 1: thuộc tính chỉ có giá trị nguyên tử
 - không thể định nghĩa phép toán mới
- => tập hợp đóng kiểu, và phép toán

Mở rộng mô hình quan hệ

■ Mở rộng cần thiết

- vượt qua chuẩn 1
- tích hợp khái niệm kiểu trừu tượng

■ Hệ thống mới

- HQTCSDL quan hệ - đối tượng

Vượt qua chuẩn 1

Chuẩn 1

■ Định nghĩa

- chuẩn 1 (1FN) :

Tất cả các thuộc tính có giá trị nguyên tử

■ Hệ quả

- một thuộc tính không thể
 - một quan hệ khác
 - danh sách giá trị
- những compositions phải xếp đặt phẳng

Giải pháp

■ giới thiệu mô hình chuẩn 2 (NF2)

- không theo chuẩn 1
- khả năng định nghĩa thuộc tính là danh sách giá trị của phép toán mới được giới thiệu

NEST và UNNEST

■ mở rộng ngôn ngữ định nghĩa dữ liệu

- định nghĩa kiểu cấu trúc
- tạo quan hệ xây dựng từ các kiểu dữ liệu này

■ thay đổi phép toán cơ bản

- xem lại chọn, chiếu, kết nối

Ví dụ

■ Căn hộ chung cư tạo thành từ danh sách các phòng

- Mô hình quan hệ

Phòng	NoAp	NoR	Tên	Diện tích
	1	1	Khách	30
	1	2	Phòng ăn	20
	1	3	Bếp	10
	1	4	Phòng ngủ	20
	1	5	Phòng tắm	6
	2	2	Phòng ngủ	15
	2	1	Phòng ngủ	40
	2	3	Phòng tắm	4

Sử dụng mô hình chuẩn 2

Phòng	NoAp	Phòng
	1	<1, khách, 30> <2, ăn, 20> <3, bếp, 10> <4, ngủ, 20> <5, tắm, 6>
	2	<2, ngủ, 15> <1, ngủ, 40> <3, tắm, 4>

Mô hình chuẩn 2

■ Phép toán mới: NEST và UNNEST

- **NEST: chuẩn 1 => chuẩn 2**

cho phép gom tập hợp các thành phần

- **UNNEST: chuẩn 2 => chuẩn 1**

phân rã đưa về chuẩn 1

Nest và Unnest

Phòng	NoAp	NoR	Tên	Diện tích
	1	1	Khách	30
	1	2	Phòng ăn	20
	1	3	Bếp	10
	1	4	Phòng ngủ	20
	1	5	Phòng tắm	6
	2	2	Phòng ngủ	15
	2	1	Phòng ngủ	40
	2	3	Phòng tắm	4

NEST

UNNEST

Phòng	NoAp	Phòng
	1	<1, khách, 30> <2, ăn, 20> <3, bếp, 10> <4, ngủ, 20> <5, tắm, 6>
	2	<2, ngủ, 15> <1, ngủ, 40> <3, tắm, 4>

Mô hình chuẩn 2

■ Mở rộng mô hình quan hệ

- thay đổi nhân cơ bản của HQTCSDL

hỗ trợ phép toán quan hệ trên đối tượng dữ liệu phức tạp
xử lý hiệu quả các thao tác trên đối tượng này

- cải tiến ngôn ngữ định nghĩa dữ liệu

định nghĩa kiểu mới

tạo mối liên quan được xác định từ các kiểu mới

■ Thay đổi quá nhiều

SQL3 và đối tượng

Mục tiêu

- **Cho phép định nghĩa kiểu dữ liệu**
 - đặc thù cho ứng dụng
 - tái sử dụng (kế thừa)
 - có thể mở rộng
- **Cho phép tích hợp hàm và thủ tục**
 - giới thiệu khái niệm kiểu trừu tượng
 - định nghĩa thủ tục, hàm xử lý kiểu dữ liệu này

Kiểu dữ liệu cơ bản trong SQL

■ SQL 2 hay SQL92

- mở rộng tập kiểu cơ bản

date, time, bit

- tạo miền giá trị

CREATE DOMAIN Name IS VARCHAR (30)

- giới thiệu ràng buộc toàn vẹn

ràng buộc miền

ràng buộc tham chiếu

Điểm mới trong SQL3

■ Kiểu mẫu tin

- sử dụng kiểu mẫu tin để tạo kiểu mới

■ Kiểu tham chiếu

- cho phép gắn định danh đến mẫu tin

■ Kiểu tập hợp

- mảng (ARRAY), tập hợp, danh sách, ...

■ Kiểu con, kiểu cha

- kế thừa

Điểm mới trong SQL3

■ Kiểu trừu tượng

- người sử dụng định nghĩa
- cho phép tích hợp thủ tục, hàm xử lý kiểu này

■ Lưu trữ thủ tục

- cho phép lưu trữ mã chương trình

■ những triggers

- thủ tục kích hoạt tự động

Kiểu mẫu tin

■ Mục tiêu

- định nghĩa kiểu mới bằng cách tập hợp các kiểu dữ liệu cơ bản có sẵn
- có thể được sử dụng trong định nghĩa thuộc tính

■ Cú pháp

- định nghĩa kiểu đơn
CREATE ROW TYPE TypeName
- định nghĩa đệ quy

Kiểu mẫu tin

■ Ví dụ 1:

```
CREATE ROW TYPE PropType (  
    NAS:  char(10),  
    LastName:  char(20),  
    FirstName:  char(20));
```

■ Ví dụ 2:

```
CREATE TABLE Apartment (  
    Nap:  integer,  
    Addr:  ROW  ( No:  integer,  
                  Str:  char(20),  
                  City: char(20)));
```

Kiểu tham chiếu

■ Mục tiêu

- cho phép định nghĩa tham chiếu trên mẫu tin
- sử dụng cho việc cài đặt định danh đối tượng
- tham chiếu có thể truy cập
- trong khi tạo bảng, người ta đặc tả rằng tham chiếu được sinh ra bởi hệ thống

■ Cú pháp

REF (TypeName)

Kiểu tham chiếu

■ Ví dụ 1:

```
CREATE ROW TYPE BuyType (  
    Prop:          REF(PropType),  
    Apartment:     REF(AppType),  
    BuyDate:       date,  
    Price:         float);
```

■ Ví dụ 2:

```
CREATE ROW TYPE PropType (  
    NAS:          char(10);  
    LastName:     char(20),  
    FirstName:    char(20),  
    PropId: REF(PropType));
```

Tạo bảng

■ Nhiều cách khác nhau

- sử dụng những kiểu để định nghĩa những lớp kiểu
- định nghĩa những thuộc tính khi tạo bảng

■ Ví dụ 1:

```
CREATE TABLE Prop OF TYPE PropType  
VALUES FOR PropId ARE SYSTEM GENERATED;
```

■ Ví dụ 2:

```
CREATE TABLE Prop AS (  
    NAS:          char(10);  
    LastName:     char(20),  
    FirstName:    char(20),  
    PropId: REF(PropType));
```

Những kiểu tập hợp

■ Mục tiêu

- định nghĩa tập hợp

■ Phép tạo

- tập hợp: không quan tâm đến thứ tự, không có 2 phần tử giống nhau

SET

- Giỏ: không quan tâm đến thứ tự, có thể có 2 phần tử giống nhau

MULTISET

- Danh sách: thứ tự, có thể có 2 phần tử giống nhau

LIST

- Mảng: thứ tự, có chỉ mục

ARRAY

Những kiểu tập hợp

■ Trong SQL3

- hiện nay chỉ có kiểu mảng
- các kiểu khác sẽ có trong tương lai (SQL4)

■ Ví dụ

- định nghĩa thuộc tính **FirstName** là mảng 3 phần tử

```
CREATE ROW TYPE PropType (  
    NAS:          char(10);  
    LastName:     char(20),  
    FirstName:    char(20) ARRAY(3),  
    PropId: REF(PropType));
```

Kiểu con, kiểu cha

■ Mục tiêu

- cho phép tái sử dụng định nghĩa kiểu
- cho phép chuyên biệt hóa kiểu

■ Cú pháp

```
CREATE TYPE sontype UNDER fathertype  
AS (Danh sách thuộc tính chuyên biệt)
```

■ Ví dụ

```
CREATE TYPE TChild UNDER TPerson  
AS (PrefGame:      char(20));
```

Bảng, kế thừa bảng

■ Mục tiêu

- cho phép định nghĩa kế thừa khi tạo bảng

■ Cú pháp

```
CREATE TABLE undertable UNDER table  
AS (Danh sách thuộc tính chuyên biệt)
```

■ Ví dụ

```
CREATE TABLE Child UNDER Person  
AS (PrefGame: char(20));
```

Kiểu người sử dụng định nghĩa

■ Định nghĩa

- kiểu được tạo từ các kiểu cơ bản và được tích hợp các phép toán xử lý nó
- tương tự kiểu trừu tượng

■ Bao gồm

- định nghĩa cấu trúc dữ liệu
- phép toán (thủ tục, hàm)

Kiểu người sử dụng định nghĩa

■ Cú pháp

```
CREATE TYPE TypeName  
AS (Danh sách thuộc tính chuyên biệt)  
[định nghĩa lại các phép toán = và <]  
[định nghĩa phương thức]
```

■ Ví dụ

```
CREATE TYPE      Point (x integer, y integer,  
FUNCTION EQUALS (:p Point, :q Point)  
RETURN BOOLEAN;  
BEGIN  
IF :p.x = :q.x AND :p.y = :q.y  
THEN RETURN TRUE ELSE RETURN FALSE;  
END;
```

Định nghĩa hàm, thủ tục

■ Sử dụng ngôn ngữ lập trình

- Oracle: PL/SQL
- Trong khai báo phương thức, ta có thể định nghĩa signature và mã phương thức

■ Định nghĩa hàm, thủ tục

CREATE FUNCTION

Ví dụ: định nghĩa dữ liệu

■ Định nghĩa kiểu

```
CREATE TYPE      BuyType (  
    Prop:        REF(PropType),  
    Appart:      REF(AppType),  
    BuyDate:     date,  
    Price:       float);
```

```
CREATE TYPE      PropType (  
    NAS:         char(10);  
    LastName:    char(20),  
    FirstName:   char(20) ARRAY(3),  
    PropId:      REF(PropType));
```

Ví dụ: định nghĩa dữ liệu

■ Tạo kiểu:

CREATE TYPE	RoomType (
Size:	integer,
Cat:	char(20));

Ví dụ: định nghĩa dữ liệu

CREATE TYPE	AppType (
Nap:	integer,
Addr:	ROW (
	No: integer,
	Str: char(20),
	City: char(20))
Rooms:	SET (RoomType)
FUNCTION	SurfaceTotal (A AppType)
	RETURNS INTEGER
RETURN /*	mã chtrình
	 */
END);	

Ví dụ: định nghĩa dữ liệu

■ Tạo bảng

```
CREATE TABLE Prop OF TYPE PropType  
VALUES FOR PropId ARE SYSTEM GENERATED;
```

```
CREATE TABLE Apartments OF TYPE AppType;
```

```
CREATE TABLE Buy OF TYPE BuyType;
```

A decorative L-shaped line in a reddish-brown color, consisting of a vertical segment on the left and a horizontal segment extending to the right, framing the title text.

Thao tác dữ liệu

Sử dụng kiểu dữ liệu định nghĩa

■ Thao tác trên các thuộc tính

- tìm kiếm số của những căn hộ chung cư trên đường Logan của thành phố Saint-Lambert

SELECT	Nap
FROM	Appartments
WHERE	Addr.Str = 'Logan'
AND	Addr.City = 'Saint-Lambert'

Sử dụng những tham chiếu

■ Phép duyệt

- tìm kiếm ngày mua liên tiếp của căn hộ chung cư tại 341 đường Green thành phố Montréal

```
SELECT      BuyDate
FROM        Buy
WHERE       Appart->Addr.Str = 'Green'
AND         Appart->Addr.City = 'Montréal'
AND         Appart->Addr.No = 341
```

Gọi hàm

■ Gọi hàm

- tính tổng diện tích của căn hộ chung cư tại 341 đường Victoria thành phố Québec

```
SELECT      SurfaceTotal
FROM        Apartment A
WHERE       Addr.Str = 'Victoria'
AND         Addr.City = 'Québec'
AND         Addr.No = 341
```

Duyệt qua phân cấp kế thừa

■ Chuyên biệt hóa trong câu truy vấn từ đồ thị kế thừa

- tìm kiếm họ của người có tên Adrien

SELECT	LastName
FROM	ONLY Person
WHERE	FirstName = 'Adrien'



Cám ơn !