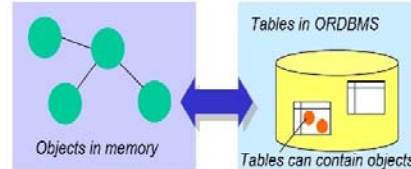


## Kiểu đối tượng và REF



### Mục tiêu:

- Khái quát được kiểu đối tượng trong CSDL quan hệ - đối tượng
- Biết cách xây dựng kiểu đối tượng Oracle
- Áp dụng được kiểu đối tượng vào các trường hợp thực tiễn



Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

1

1

## Nội dung



- Đặt vấn đề
- Kiểu đối tượng
- Sử dụng và thao tác với kiểu đối tượng
- Luyện tập
- Tóm tắt

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

2

2

## Đặt vấn đề



- Kiến trúc nền của mô hình quan hệ - đối tượng là **bảng + một số đặc trưng của mô hình hướng đối tượng**
  - Sử dụng bảng
  - Kiểu dữ liệu mới: kiểu đối tượng, kiểu REF, VARRAY, NESTED TABLE
  - Kế thừa
  - ...
- => Có khả năng biểu diễn dữ liệu phức tạp và hỗ trợ đa truy cập, tìm kiếm thuận lợi
- Ngôn ngữ SQL2003

**ORACLE®**  
DATABASE

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

3

3

## Kiểu đối tượng



- Kiểu đối tượng = Kiểu người dùng định nghĩa
- UDT (user-defined types) = (ADT) abstract data types
- Kiểu đối tượng xây dựng từ thuộc tính, phương thức

Object Type <i>person_typ</i>	
Attributes	Methods
idno	get_idno
first_name	display_details
last_name	
email	
phone	

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

4

4

## Kiểu đối tượng



### Cú pháp SQL3

```
CREATE TYPE <Tên_kiểu_UDT> AS (
<liệt kê các cặp thuộc tính-kiểu dữ liệu,
>);
```

Cú pháp Oracle khi định nghĩa kiểu thêm từ OBJECT sau AS

so với SQL3

```
CREATE [OR REPLACE] TYPE <tên_kiểu_UDT> AS
OBJECT(
<liệt kê các thuộc tính kiểu dữ liệu, >
[<phương thức>]);/
```

Chú ý dấu / để thực thi nhanh hơn (sử dụng vùng đệm)

## Kiểu đối tượng – ví dụ



```
CREATE TYPE person_typ AS OBJECT (
    idno NUMBER,
    first_name VARCHAR2(20),
    last_name VARCHAR2(25),
    email VARCHAR2(25),
    phone VARCHAR2(20),
    MAP MEMBER FUNCTION get_idno RETURN NUMBER;
/
CREATE TYPE BODY person_typ AS
MAP MEMBER FUNCTION get_idno RETURN NUMBER IS
BEGIN
    RETURN idno;
END;
END;
/
```

Object Type *person\_typ*

Attributes	Methods
idno first_name last_name email phone	get_idno display_details

## Sử dụng kiểu đối tượng – ví dụ



- Bảng đối tượng: Bảng chỉ chứa một kiểu đối tượng

```
CREATE TABLE person_obj_table OF person_typ;
```

- Kiểu đối tượng là kiểu dữ liệu của các cột trong bảng

```
CREATE TABLE contacts ( contact person_typ, contact_date DATE );
```

- Tham chiếu đối tượng (kiểu REF)

```
CREATE TYPE emp_person_typ AS OBJECT (
    name VARCHAR2(30),
    manager REF emp_person_typ);/
```

- Trong PL/SQL: Biến đối tượng

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

7

7

## CRUD với bảng đối tượng



- Tạo bảng đối tượng

```
CREATE TABLE <tên_bảng> OF <tên_kiểu_UDT>;
```

– Ví dụ trong oracle:

```
CREATE OR REPLACE TYPE svType AS OBJECT (
    masv CHAR(20),
    ht CHAR(20),
    ns NUMBER);
```

– Ta định nghĩa 1 bảng các đối tượng svType:

```
CREATE TABLE tblSV OF svType;
```

- Nhập dữ liệu vào bảng đối tượng

– Sử dụng INSERT INTO tương tự CSDL quan hệ

```
INSERT INTO tblSV VALUES ('1', 'PHAN', NULL);
```

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

8

8

## CRUD với bảng đối tượng



### ▪ Tạo bảng đối tượng

```
CREATE OR REPLACE TYPE svType AS OBJECT (
    masv CHAR(20),
    ht CHAR(20),
    ns NUMBER);
CREATE TABLE tblSV OF svType;

INSERT INTO tblSV VALUES ('1', 'PHAN', 1999);
INSERT INTO tblSV VALUES ('2', 'NGUYEN', 2001)
```

## CRUD với bảng đối tượng



### ▪ Truy cập bảng đối tượng

- Sử dụng ký pháp dấu chấm

```
SELECT tblSV.ht
FROM tblSV
```

- Sẽ là một thói quen tốt khi làm việc với Oracle, khi sử dụng bí danh cho mọi quan hệ trong CSDL quan hệ đối tượng

```
SELECT s.ht
FROM tblSV s
```

## CRUD với bảng đối tượng



### ▪ Value(bí danh bảng)

- **Value(Bí\_danh\_bảng)** nhận đối là bí danh của một bảng được kết hợp với một dòng của bảng đối tượng và trả về một thể hiện đối tượng được lưu trong bảng này.
- Kiểu của thể hiện đối tượng cùng kiểu với kiểu của bảng đối tượng

```
SELECT s.ht
FROM tblSV s

SELECT s.ht ho_ten
FROM tblSV s
WHERE value(s).ns=2001
```

## CRUD với bảng đối tượng



### ▪ Truy cập bảng đối tượng

- Sử dụng ký pháp dấu chấm

```
SELECT tblSV.ht
FROM tblSV
```

- Sẽ là một thói quen tốt khi làm việc với Oracle, khi sử dụng bí danh cho mọi quan hệ trong CSDL quan hệ đối tượng

```
SELECT s.ht
FROM tblSV s
```

### ▪ Cập nhật và xóa dữ liệu trong bảng đối tượng

```
Update <tên_bảng>
Set ???
```

## UDT là kiểu của cột trong bảng



- UDT có thể là kiểu dữ liệu của thuộc tính, hay cột trong bảng, khi đó nó được sử dụng trong lệnh CREATE TABLE tương tự CSDL quan hệ
- Ví dụ:

```
CREATE OR REPLACE TYPE diachiType AS OBJECT
(xa CHAR(30), huyen CHAR(20), tinh CHAR(30)); /
```

Tạo bảng (quan hệ) tblNV sử dụng diachiType làm kiểu dữ liệu cho một cột:

```
CREATE TABLE tblNV(
msnv int, hoten CHAR(30),
diachi diachiType); /
```

## CRUD với bảng có UDT là kiểu của cột



- Nhập dữ liệu
  - Dùng INSERT INTO ...VALUES (...) nhưng với thuộc tính có kiểu UDT thì phải mô tả tên\_kiểu(các giá trị)
  - Ví dụ sai:
 

```
INSERT INTO TBLNV VALUES
(1, 'TRAN', 'XA A', 'HUYEN B', 'TINH C') ;
```
  - Ví dụ đúng:
 

```
INSERT INTO TBLNV VALUES
(1, 'TRAN', DiachiType('XA A', 'HUYEN B', 'TINH C')) ;
```
- Tìm kiếm
  - Sử dụng bí danh và ký pháp dấu chấm
 

```
SELECT <bí_danh>.<thuộc_tính>
FROM <tên_bảng> <bí_danh>
WHERE <Điều_kiện>
```

## CRUD với bảng có UDT là kiểu của cột



- Cập nhật (chỉnh sửa/thay đổi) dữ liệu một dòng nhiều dòng

```
UPDATE <TÊN_BẢNG> <bí_danh>
SET <bí_danh>.<TÊN_CỘT>=<Biểu_thức>
WHERE <Điều_kiện>
```

- Xóa một dòng nhiều dòng

```
DELETE
FROM <TÊN_BẢNG> <bí_danh>
WHERE <Điều_kiện>
```

## CRUD với bảng có UDT là kiểu của cột



- Truy cập giá trị trong bảng có cột là kiểu UDT
  - Sử dụng biểu thức đường dẫn kết hợp với BÍ DANH.

Yêu cầu BÍ DANH LÀ BẮT BUỘC

– Ví dụ sai :

```
SELECT msnv, hoten,diachi.XA FROM TBLNV
```

– Ví dụ sai:

```
SELECT
    TBLNV.msnv,TBLNV.hoten,TBLNV.diachi.XA
FROM TBLNV
```

– Ví dụ đúng:

```
SELECT NV.msnv,NV.hoten,NV.diachi.XA
FROM TBLNV NV
```



## CRUD với bảng có UDT là kiểu của cột



```
CREATE OR REPLACE TYPE diachiType AS OBJECT
(xa CHAR(30),huyen CHAR(20),tinh CHAR(30));/

CREATE TABLE tblNV( msnv int primary key, hoten CHAR(30),
diachi diachiType);/

INSERT INTO TBLNV VALUES (1, 'TRAN', DiachiType('XAA', 'HUYEN B', 'TINH
C'));

INSERT INTO TBLNV VALUES (2, 'DINH', DiachiType('HB', 'Tp Vinh', 'Nghe An'));

SELECT NV.msnv, NV.hoten, NV.diachi.XA
FROM TBLNV NV

SELECT Nv.*, nv.diachi.xa FROM tblnv nv

UPDATE tblnv nv
SET nv.diachi.xa = 'New xa'
WHERE nv.msnv = 1
```

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

17

17

## Luyện tập



▪ Cho cơ sở dữ liệu quan hệ - đối tượng như sau:

```
CREATE OR REPLACE TYPE diachiType AS OBJECT
(xa CHAR(30),huyen CHAR(20),tinh CHAR(30));/
CREATE OR REPLACE TYPE gvType AS OBJECT (
    masv CHAR(20),
    ht CHAR(20),
    ns NUMBER,
    dc diachiType);
CREATE TABLE tblGV OF gvType;
```

- Viết lệnh SQL2003 để thực hiện CRUD thông tin của các giáo viên

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

18

18

## Bảng đối tượng lồng đối tượng



- Làm việc với bảng đối tượng của đối tượng
- CRUD với bảng đối tượng, trong đó kiểu đối tượng chứa một kiểu đối tượng khác thì được ứng xử như bảng có cột là kiểu đối tượng
- Tức là, sử dụng bí danh và ký pháp "dấu chấm"  

```
CREATE OR REPLACE TYPE gvType AS OBJECT (
    masv CHAR(20),
    ht CHAR(20),
    ns NUMBER,
    dc diachiType);

CREATE TABLE tblGV OF gvType;

INSERT INTO tblGV VALUES('1','GIAP',1999, diachiType('Ben Thuy','Vinh', 'Nghe An'));

INSERT INTO tblGV VALUES('2021','DINH',2003, diachiType('Xuan Lam','Nghie Xuan', 'Ha Tinh'));

SELECT GV.HT, gv.dc.xa FROM TBLGV GV

UPDATE TBLGV GV
SET gv.dc.xa = 'Trung Do'
WHERE GV.HT='PHAN'
```

## Bảng đối tượng lồng đối tượng



- Bài tập: Đưa ra một ví dụ minh họa để tạo lập một bảng đối tượng, trong đó kiểu đối tượng được xây dựng từ một kiểu đối tượng khác
- Ý nghĩa:  
 Sử dụng đối tượng trong bảng = Áp dụng kiểu bộ (tuple)
  - Class: (thuộc tính + phương thức + đóng gói+...)
  - Thuộc tính phức hợp
  - Mối liên kết 1-1

## Kiểu tham chiếu đối tượng REF



- REF (Bí\_danh):
- REF nhận đối là bí\_danh của một bảng đối tượng và trả về thể hiện của một đối tượng mà nó tham chiếu tới
- Deref() là một hàm của SQL2003 trả về thể hiện của một đối tượng được tham chiếu bởi REF. Như vậy đối của hàm Deref() là cột có kiểu dữ liệu REF

## Kiểu tham chiếu đối tượng (REF)



- Nếu  $T$  là một kiểu dữ liệu thì REF  $T$  là kiểu tham chiếu tới  $T$ , tức là, trở tới một đối tượng của kiểu  $T$ .
- REF  $T$  sẽ gọi OID của một đối tượng trong  $T$
- Khác với OID của đối tượng, REF có thể được hiển thị mặc dù nó không có ngữ nghĩa



## Kiểu tham chiếu đối tượng

- Kiểu tham chiếu đối tượng (REF) dùng để mô hình hóa *mối liên kết* giữa các đối tượng (bảng ) thay vì dùng khóa ngoài
- Kiểu tham chiếu đối tượng được định nghĩa bằng từ khóa *REF Kiểu\_dữ\_liệu*
- Trong ORACLE, tùy chọn SCOPE IS dùng để hạn chế sự tham chiếu, nó chỉ trỏ tới các đối tượng trong một bảng cụ thể (ràng buộc tham chiếu)



## Kiểu tham chiếu đối tượng REF

```
CREATE TYPE new_nv_type AS OBJECT (
  id number,
  name  VARCHAR2(30),
  manager REF new_nv_type);/

CREATE TABLE tbl_obj_new_nv OF new_nv_type;

INSERT INTO tbl_obj_new_nv VALUES (new_nv_type(2021, 'Ty',
NULL));

INSERT INTO tbl_obj_new_nv
SELECT new_nv_type(1,'Ngo', REF(nv))
FROM tbl_obj_new_nv nv
WHERE nv.id= 2021;
```

## Kiểu tham chiếu đối tượng REF



```
SELECT value(nv).name,  
DEREF(value(nv).manager  
)name as nguoi_quan_ly  
FROM tbl_obj_new_nv nv;/
```

```
UPDATE tbl_obj_new_nv  
SET manager = (select  
ref(nv) from  
tbl_obj_new_nv nv where  
nv.id=2021)
```

Cho biết các  
truy vấn bên  
làm gì?

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

25

25

## Kiểu tham chiếu đối tượng REF



- Kiểu tham chiếu đối tượng (REF) dùng để mô hình hóa mối liên kết 1-1 hoặc 1\_N giữa các đối tượng trong kiểu (bảng) đối tượng
- Kiểu tham chiếu đối tượng được định nghĩa bằng từ khóa *REF Kiểu\_đối\_tượng*

REF được  
sử dụng ở  
đâu?

Cơ sở dữ liệu quan hệ - đối tượng – TS. Phan Anh Phong

26

26