Université A/Mira Bejaïa
Faculté des Sciences Exactes
Département d'Informatique

Niveau : 1 re année Master
Module : BDD Avancées
Année Universitaire 2017/2018

BDDA (compléments du cours 2):Modèle de données RO, types et tables).

création de type

Syntaxe de création de type :

```
CREATE [OR REPLACE] TYPE schema.nomType [AS OBJECT | UNDER schema.nomSurType ]
        (colonne1 type1, ...
        methode1 (paramètres), ...)
        [[NOT] INSTANTIABLE]
        [[NOT] FINAL]
        /
```

Syntaxe de création de type.

L'exemple suivant décrit le type *pilote_type* composé de quatre colonnes au premier niveau. Les colonnes *etat_civil_t* et *adresse_t* sont appelées types composants.

Pilote _type

brevet	Etat_civil_t		Adresse_t			paye
	nom	datenais	nrue	rue	ville	

Figure 01. types composants

```
CREATE OR REPLACE TYPE etat_civil_type AS OBJECT
(nom VARCHAR(30), datenais DATE)
/
CREATE OR REPLACE TYPE adresse_type AS OBJECT
(nrue NUMBER(3), rue VARCHAR(40), ville VARCHAR(30))
/
CREATE OR REPLACE TYPE Pilote_type AS OBJECT
(brevet CHAR(6), etat_civil_t etat_civil_type, adresse_t adresse_type,
paye NUMBER(6,2)) /
```

Tables relationnelles :

Objets colonne

```
CREATE TABLE Vols
        (       numero NUMBER, dateVol DATE,
                nombrePax NUMBER(3), depart VARCHAR(30),
                arrivee VARCHAR(30),
                CDB_t   Pilote_type, COPI_t   Pilote_type,
                CONSTRAINT pk_Vols PRIMARY KEY(numero));
```

Instanciation

```
INSERT INTO Vols VALUES (1, SYSDATE, 120, 'Orly-Ouest',
'Blagnac', Pilote_type('PL-11', etat_civil_type('Peyrard','05-02-1970'),
adresse_type (1,'G. Brassens','Blagnac'),3500) ,
Pilote_type ('CPL-10',        etat_civil_type ('Mercier','05-12-1947'),
adresse_type(5,'Boeldieu','Toulouse'),5600));
```

Extraction

```
SELECT v.numero, v.datevol, v.nombrePax, v.depart, v.arrivee, v.cdb.etat_civil.nom FROM Vols v;

NUMERO DATEVOL NOMBREPAX DEPART ARRIVEE CBD_T.ETAT_CIVIL_T.NOM
--------------------------------------------------
1      24/02/03  120      Orly-ouest  Blagnac      Diffis
```

Valeurs par défaut :

```
CREATE TABLE Vols (numero NUMBER, dateVol
DATE,nombrePax NUMBER(3), depart VARCHAR(30), arrivee
VARCHAR(30),CDB_t Pilote_type DEFAULT Pilote_type('PL-
1',etat_civil_type('Sigaudes','10-06-1960'),
adresse_type(12,'Lasbordes','Balma'),3500),COPI_t Pilote_type
DEFAULT Pilote_type('PL-2',etat_civil_type('Soutou','05-02-
1965'),adresse_type(7,'Camparols','Blagnac'), 3000),
        CONSTRAINT pk_Vols PRIMARY KEY(numero));


INSERT INTO Vols (numero, dateVol, nombrePax, depart, arrivee)
        VALUES (2, SYSDATE, 200, 'Nice', 'Biarritz');
```

L'affichage de quelques colonnes de la table VOLS confirme l'insertion des valeurs par défaut.

```
SELECT v.numero, v.depart, v.arrivee, v.cdb.etat_civil.nom, v.copi.etat_civil.nom   FROM Vols v;

NUMERO DEPART    ARRIVEE CBD_T.ETAT_CIVIL_T.NOM COPI_T.ETAT_CIVIL_T.NOM
-------------------------------------------------------
2      Nice      Biarritz Sigaudes Soutou
```

Tables objet :

La directive OF dans l'instruction CREATE TABLE permet de spécifier le type qui décrira les objets stockés dans la table objet.

(CREAT TABLE *nomTableObjet* OF *nomType*) est une table objet. Chaque enregistrement est un objet ligne (*rowobject*). A ce titre, chaque ligne est munie d'un unique OID.

Contraintes

NB : Un type ne peut pas contenir de contraintes (NOT NULL, CHECK, etc.). Les contraintes doivent être déclarées au niveau de la table objet.

```
CREATE TABLE Pilote OF Pilote_type
        (CONSTRAINT pk_Pilote PRIMARY KEY(brevet),
         CONSTRAINT df_paye paye DEFAULT 3000,
         CONSTRAINT nn_paye CHECK (paye IS NOT
NULL),
         CONSTRAINT ck_paye CHECK (paye BETWEEN
2000 AND 5000),
        CONSTRAINT nn_nom  CHECK (etat_civil.nom IS
NOT NULL),
         CONSTRAINT un_nom  UNIQUE (etat_civil.nom)
        );
```

Instanciation

```
INSERT INTO Pilote
        VALUES (Pilote_type('PL-11',etat_civil_type('Peyrard','05-02-
1970'),
                adresse_type(1,'G. Brassens','Blagnac'),3500));

INSERT INTO Pilote
        VALUES ('CPL-2',etat_civil_type('Laroche','15-12-1963'),
                adresse_type(2,'Foch','Montauban'),2600);

INSERT INTO Pilote
        VALUES ('CPL-3',etat_civil_type('Labat','25-12-1965'),
                adresse_type(3,'Camparols','Pau'),3000);
```

OID basés sur la clé primaire

Pour préciser que les OID doivent être basés sur la clé primaire de la table, il faut opter pour l'option PRIMARY KEY comme le montre l'instruction suivante :

```
CREATE TABLE Pilote OF Pilote_type
(CONSTRAINT pk_Pilote PRIMARY
KEY(brevet),autrescontraintes..)
OBJECT IDENTIFIER IS PRIMARY KEY;
```

Manipulation des objets

```
UPDATE Pilote p
        SET  p.etat_civil.nom = 'Sigaudes', p.paye=3700
        WHERE brevet = 'PL-11';

DELETE FROM Pilote p
        WHERE NOT (p.adresse.ville = 'Blagnac');
```

Chargement d'un objet (VALUE)

```
SELECT VALUE(ap) FROM Pilote ap  WHERE ap.brevet= 'PL-11' ;

VALUE(AP)( BREVET,
ETAT_CIVIL_T(NOM,DATEDAIS),ADRESSE_T(NRUE, RUE,
VILLE),PAYE)
---------------------------------------------------
PILOTE_TYPE('PL-11', ETAT_CIVIL_TYPE('Sigaudes', '05/02/70'),
ADRESSE_TYPE(1, 'G. Brassens', 'Blagnac'),3700)


SELECT * FROM Pilote WHERE brevet = 'PL-11';

BREVET  ETAT_CIVIL_T(NOM,DATEDAIS)
ADRESSE_T(NRUE, RUE, VILLE)   (PAYE)
------  -------------------------  ----------------------
'PL-11' ETAT_CIVIL_TYPE('Sigaudes', ADRESSE_TYPE(1, 'G.
Brassens', 3700
      '05/02/70')              'Blagnac')
```

Le bloc PL/SQL suivant charge ces données dans un objet non persistant, l'objet non persistant doit ensuite être manipulé par la notation pointé

```
DECLARE
        var_pilote Pilote_type;
BEGIN
        SELECT VALUE(ap) INTO var_pilote FROM Pilote ap
                WHERE ap.brevet = 'PL-11';
        IF (var_pilote.etat_civil.nom = 'Sigaudes') THEN
                DBMS_OUTPUT.PUT_LINE('Sigaudes est trouvé! sa
paye : ' || TO_CHAR(var_pilote.paye));
        END IF;
END;
/
```

```
Sigaudes est trouvé ! sa paye : 3700
Rocedure PL/SQL terminée avec succés
```