LES COLLECTIONS

I. LES TABLES IMBRIQUÉES

Exemple

nom	prénom	liste des diplômes
…	…	
…	…	

Syntaxe

```
CREATE TYPE nomcollection_nt_type AS TABLE OF
nomelement_elt_nt_type [NOT NULL]
/
```

```
CREATE TABLE nomdetable [ OF nomType / (col1 type1, …,
collection_nt nomcollection_nt_type, …) ]
        NESTED TABLE collection_nt STORE AS
collection_tabnt;
```

-- type des éléments de la table imbriquée

```
CREATE TYPE employe _elt_nt_type AS OBJECT
        (       code_emp    VARCHAR(20),
                Nom_emp     VARCHAR(30),
                Age_emp     NUMBER) /
```

-- type pour une table imbriquée

-- employe_ nt_type *est le nom du type de la table imbriquée*
-- employe_ elt_nt_type *est le type des éléments de la table*

```
CREATE TYPE employe_ nt_type AS TABLE OF employe_
elt_nt_type
/
```

-- table relationnelle en non première forme normale :

- employes_tabnt est le nom physique de la table imbriquée (ne sert jamais dans les requêtes)

```
CREATE TABLE departements
        (       num_dep     NUMBER,
                budget      NUMBER,

                employes    employe_ nt_type

        ) NESTED TABLE employes STORE AS
employes_tabnt;
```

num_dep	budget	employes
1	100 000	
13	50 000	

<u>Création des tables objet relationnelles utilisant des tables imbriquées</u>

```
CREATE TYPE departement_type AS OBJECT
        (       num_dep     NUMBER,
                budget      NUMBER,
                employes    employe_ nt_type
        )
        /
```

-- table objet relationnelle

-- STORE AS table_name à préciser lors de la création de la table

```
CREATE TABLE departements OF department_type
                NESTED TABLE employes STORE AS
        employes_tabnt;
```

--Insertion dans une relation utilisant une table imbriquée

```
INSERT INTO departements VALUES ( 1, 200 000,
employe_nt_type (   employe _elt_nt_type (12345, 'toto', 25),
                    employe _elt_nt_type (2222, 'titi', 28))) ;
```

-- employe_ nt_type est le constructeur de type de la table employé
-- employe _elt_nt_type est le constructeur de type des éléments de la table imbriquée.
-- (12345, 'toto', 25) et (2222, 'titi', 28) sont les valeurs des attributs de l'élément

```
INSERT INTO departements VALUES (2, 100 000, employe_ nt_type
()) ;
```
--insertion d'une table imbriquée vide.

```
INSERT INTO departements (numdep, budget) VALUES (4, 100 000)
;
```

	num_dep	budget	employes
OID	1	200 000	
OID	2	100 000	
OID	4	100 000	null

Insertion dans une table imbriquée

Il existe une commande particulière pour insérer dans une table imbriquée : THE

```
INSERT INTO THE ( SELECT employes FROM departements
WHERE numdep = 1 )
VALUES (employe _elt_nt_type (789, 'tutu', 20) ) ;
```

-- (SELECT …) est la table imbriquée dans laquelle on insère l'élément
-- employes est l'attribut de la table imbriquée
-- numdep = 1 est la condition du n-uplet dont on veut modifier la table
-- employe _elt_nt_type (789, 'tutu', 20) est l'élément à inséser dans la table imbriquée

Résultat

	num_dep	budget	employes
OID	1	200 000	

Que se passe-t-il avec les requêtes suivantes ?

INSERT INTO THE (SELECT employes FROM departements WHERE numdep = 2)
VALUES (employe _elt_nt_type (5432, 'lulu', 27)) ;

-- on insère l'élément

INSERT INTO THE (SELECT employes FROM departements WHERE numdep = 4)
VALUES (employe _elt_nt_type (987, 'nono', 50)) ;

-- ça ne marche pas car la table imbriquée n'existe pas

Comment faire pour que ça marche ?

Il faut créer la table imbriquée au préalable.

On met à jour le n-uplet pour rendre l'insertion possible :

UPDATE departements SET employes = employe_ nt_type () WHERE numdep = 4 ;

Puis on exécute la requête d'insertion.

INSERT INTO THE (SELECT employes FROM departements WHERE numdep = 4)
VALUES (employe _elt_nt_type (987, 'nono', 50)) ;

2. On met à jour en insérant l'élément :

UPDATE departements SET employes = employe _elt_nt_type (t_employe(987, 'nono', 50)
WHERE numdep = 4 ;

<u>Interrogation des tables utilisant des tables imbriquées</u>

Pour la table objet-relationnelle

SELECT * FROM departements ;

On obtient les valeurs des attributs comme habituellement avec les tables imbriquées sous
Forme de constructeurs de type.

Pour les tables imbriquées, on utilise THE comme pour l'insertion

> **SELECT e.nom FROM THE (SELECT employes FROM departements WHERE numdep = 1) e ;**

-- on doit utiliser un alias de table (e)
-- e.nom est un attribut du type employe _elt_nt_type
-- (SELECT …) est une table imbriquée

Remarque

Ici aussi, il faut sélectionner une seule table imbriquée.

<u>Comment récupérer les informations de plusieurs tables imbriquées simultanément ?</u>

Utilisation de la commande CURSOR, par exemple :

SELECT d.numdep CURSOR (SELECT e.nom FROM TABLE (employes) e)
FROM departements d ;

On obtient la liste des départements avec, pour chaque département, la liste des noms des employés.

RQ :

L'opérateur TABLE n'existait pas dans la première version objet d'Oracle, il était nommé THE. Pour se conformer à SQL3, cette commande a été débaptisée (toutefois encore opérationnelle afin d'assurer la compatibilté).

Syntaxe de manipulation de tables imbriquées :

INSERT INTO TABLE
(SELECT Collection FROM nomtable WHERE ……)
VALUES constructeurelement (valeurs)

UPDATE TABLE
(SELECT Collection FROM nomtable WHERE ……) acoll
SET acoll.colonne=
WHERE…….

DELETE FROM TABLE
(SELECT Collection FROM nomtable WHERE ……) acoll
WHERE acoll.colonne=

II. LES VARRAY

Syntaxe :

CREATE TYPE *nomcollection_vry_type* IS VARRAY(*n*) OF *elt_vry_type* [NOT NULL]
/

Exemple :

Compagnie

comp	nomComp	{pilotes_vry}			{filiales_vry}	
		brevet	nom	age	codfil	pourcent
ALIT	Air-Littoral	PL-1	Lamothe	36	AF	78
		PL-2	Albaric	37	MAIF	22
		PL-3	Peyrard	33		
ABLA	Air-Blagnac				ALIT	90
ATAH	Tahiti-AL	PL-4	Bidal	38		
		PL-5	Soutou	38		

1.

```
CREATE TYPE filiales_elt_vry_type AS OBJECT
 (codefil VARCHAR(4), pourcent NUMBER(3))
/
CREATE TYPE filiales_vry_type AS VARRAY(5) OF
filiales_elt_vry_type
/
CREATE TYPE pilotes_elt_vry_type AS OBJECT
 (brevet CHAR(6), nom VARCHAR(20), age NUMBER(2))
/
CREATE TYPE pilotes_vry_type AS VARRAY(3) OF
pilotes_elt_vry_type
/
```

```
CREATE TYPE Compagnie_type AS OBJECT
 (comp VARCHAR(4), nomComp VARCHAR(15),
  pilotes_vry    pilotes_vry_type,
  filiales_vry   filiales_vry_type)
/

CREATE TABLE Compagnie OF Compagnie_type
 ( CONSTRAINT pk_Compagnie PRIMARY KEY(comp) );
```

Table R-Objet

```
CREATE TABLE Compagnie

 (comp VARCHAR(4), nomComp VARCHAR(15),
  pilotes_vry    pilotes_vry_type,
  filiales_vry   filiales_vry_type) ;
```

Table R-1FN²

2.

```
INSERT INTO Compagnie
     VALUES (Compagnie_type('ALIT','Air-Littoral',
     pilotes_vry_type(pilotes_elt_vry_type('PL-
3','Lamothe',36),
                      pilotes_elt_vry_type('CPL-
4','Albaric',37),
                      pilotes_elt_vry_type('PL-
5','Peyrard',33)),
          filiales_vry_type(
     filiales_elt_vry_type('AF',78),

     filiales_elt_vry_type('MAIF',22)
                           )
          ));
```

```
INSERT INTO Compagnie
     VALUES (Compagnie_type('ABLA','Air-Blagnac',
pilotes_vry_type(),

     filiales_vry_type(filiales_elt_vry_type('ALIT'
,90),

filiales_elt_vry_type(NULL,NULL),

filiales_elt_vry_type(NULL,NULL),

filiales_elt_vry_type(NULL,NULL),

filiales_elt_vry_type(NULL,NULL))
          ));
```

```
INSERT INTO Compagnie
     VALUES (Compagnie_type('ATAH','Tahiti-AL',
     pilotes_vry_type(pilotes_elt_vry_type('PL-
4','Bidal',38),
                      pilotes_elt_vry_type('PL-
5','Soutou',38),

     pilotes_elt_vry_type(NULL,NULL,NULL)),NULL
          ));
```

NB :

Pour manipuler un élément d'une collection varray, il faut programmer la mise à jour sous PL/SQL. Le principe est de charger en mémoire la collection, d'appliquer une des fonctions PL/SQL pour les collections, puis de mettre à jour l'objet colonne dans sa globalité. Le premier indice d'un tableau est 1.

Le bloc suivant ajoute deux filiales à la compagnie de code ‘ABLA’. La clause FOR UPDATE OF verouille préventivement la collection pour éviter un accès concurrent. Deux types d’écritures sont possibles pour modifier un élément (l’élément entier par l’intermédiaire du constructeur ou par colonne en spécifiant chaque valeur).

```
DECLARE
   tableau_filiales filiales_vry_type;
BEGIN
     SELECT filiales_vry INTO tableau_filiales
          FROM Compagnie WHERE comp = 'ABLA' For
UPDATE OF filiales_vry;
     tableau_filiales(2) :=
filiales_elt_vry_type('ATAH',5);
     tableau_filiales(3).codefil  := 'IUT';
     tableau_filiales(3).pourcent := 5;
     UPDATE Compagnie
          SET filiales_vry = tableau_filiales
          WHERE comp = 'ABLA';
END;
/
```

Vérifions ces ajouts en interrogeant la collection *varray* à l’aide de la directive TABLE.

```
SELECT * FROM   TABLE (SELECT filiales_vry FROM
Compagnie WHERE comp = 'ABLA') ;
```

Codefil	pourcent
ALIT	90
ATAH	5
IUT	5