

LE MODELE RELATIONNEL ETENDU
LES COLLECTIONS: LES TABLEAUX IMBRIQUES
VARRAY

- Ensemble d'éléments ordonnés de même type.
Chaque élément occupe une place unique
- Utilisation: implémenter une relation n-aire avec une cardinalité fixée:
- Exemple 1: une personne de sexe masculin peut avoir au maximum 4 conjointes
- Exemple 2: un module peut être composé au maximum de
- 5 cours.

Exemple1:

Un tableau d'un type classique

```
create type liste_prenoms as varray(5) of varchar2(20);
```

Exemple2:

Un tableau d'un type objet

```
create type T_adresse as object (  
    rue varchar2(20),  
    ville varchar2(20));
```

```
create type liste_adresses_privees as varray(5) of T_adresse;
```

Exemple3:

Un tableau d'un type de référence vers des objets

```
create type T_adresse as object (  
    rue varchar2(20),  
    ville varchar2(20));
```

```
create type liste_adresses_privees as varray(5) of REF T_adresse;
```

Exemple d'utilisation:

```
create table table_employees (  
  nom                varchar2(20),  
  prenom             liste_prenoms,  
  adr_privees        liste_adresses_privees,  
  adr_publiques      liste_adresses_publiques  
);
```

Tab_Employes

nom	prenoms				Adr_privees				Adr_publicues		
	1	2	...	max	1	2	...	max	1	...	max
	string	string	...	string	Tadr	Tadr	...	Tadr	REF Tadr	...	REF Tadr
Toto										...	

Tab_Adresses

rue	ville
rue1	Ville1
rue2	Ville2

Exemple d'insertion d'un VARRAY:

```
insert into table_employes values (  
'nom1',  
liste_prenoms('prenom11', 'prenom12', 'prenom13'),  
liste_adresses_privees(T_adresse('r', 'v'), T_adresse('x', 'y')),  
liste_adresses_publicues((select ref(d) from ....), (select ...))  
);
```

Remarque: même principe que pour les objets et les références

Exemple de sélection dans un VARRAY: **aplatisseur TABLE**

```
SQL> --- sélection des rues de la personnes
```

```
SQL> ---
```

```
SQL>
```

```
SQL> select e.rue from table_employees p, table(p.adr_privees) e;
```

```
RUE
```

```
-----
```

```
rue11
```

```
rue12
```

Exemple de modification dans un VARRAY: ACCES GLOBAL

```
SQL> update table_employes
```

```
2  set prenom=liste_prenoms('XXXXX')
```

```
3  where nom='nom1';
```

1 ligne mise à jour.

```
SQL> select prenom from table_employes;
```

PRENOMS

LISTE_PRENOMS('XXXXX')

Exemple de modification d'un prénom dans une case donnée dans un VARRAY:

```
SQL> declare
2  new_prenom liste_prenoms;
3  begin
4  select prenom INTO new_prenom
5  from table_employes where nom='nom1';
6  new_prenom(1):='ZZZZZZ';
7  update table_employes
8  set prenom=new_prenom
9  where nom='nom1';
10 end;
11 /
```

Procédure PL/SQL terminée avec succès.

SQL>

SQL>

SQL> select prenom from table_employees;

PRENOMS

LISTE_PRENOMS('ZZZZZZ')

Exemple d'utilisation de EXTEND: allocation d'une nouvelle place si la limite n'est pas atteinte

```
SQL> declare
  2 new_prenom liste_prenoms;
  3 begin
  4 select prenom into new_prenom
  5 from table_employes where nom='nom1';
  6 new_prenom.EXTEND;
  7 update table_employes set prenom=new_prenom
  8 where nom='nom1';
  9 end;
10 /
```

Exemple d'utilisation de EXTEND:

SQL>

SQL> ---verification de l'extension

SQL>

SQL> select prenom from table_employees;

PRENOMS

LISTE_PRENOMS('ZZZZZZ', **NULL**)

Les méthodes associées au type VARRAY

Fonction	description
EXISTS(X)	Retourne TRUE si le Xime élément de la collection existe
COUNT	Retourne le nombre d'éléments dans la collection
LIMIT	Retourne le nombre maximum d'éléments dans un VARRAY
FIRST	Retourne le premier indice de l'élément de la collection
LAST	Retourne le dernier élément de la collection
PRIOR(X)	Retourne l'élément avant le Xime élément de la collection
NEXT(X)	Retourne l'élément après le Xime élément de la collection
TRIM(X)	Supprime X éléments à partir de la fin de la collection
EXTEND EXTEND(X) EXTEND(X,Y)	Ajoute une ou plusieurs copies à partir du Yime élément de la collection. X est le nombre d'élément à copier

Exemple d'utilisation d'une fonction de VARRAY

SQL> declare

2 new_prenom liste_prenoms;

3 begin

4 select prenom into new_prenom

5 from table_employees where nom='nom3';

6 if new_prenom.EXISTS(2) then

7 insert into table_employees (nom) values ('nom4');

8 else insert into table_employees (nom) values ('nom5');

9 end if;

10 end;

11 /

Procédure PL/SQL terminée avec succès.

TRAVAUX PRATIQUES

TAB_COURS

NUMCOU RS			LISTE_ENS

TAB_PERS

CIN	NOM	*****
K345234	toto	

TABLE AU IMBRIQUEE DE REF
DES PERSONNES