

# Bases de Données

## C7- SQL3 (fin)

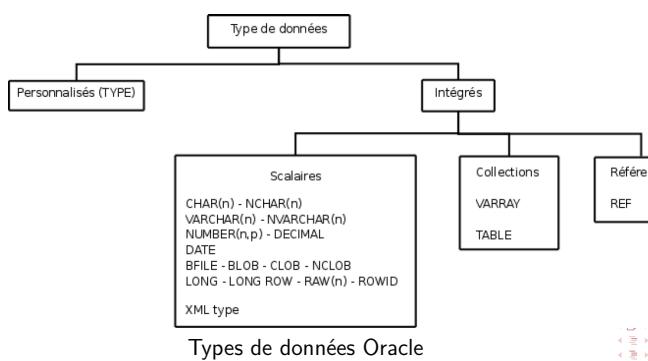
Lina Soualmia

Université de Rouen  
LITIS - Équipe TIBS-CISMef  
lina.soualmia@chu-rouen.fr

21 octobre 2015

## Rappels C6

Lina Soualmia	Bases de Données	SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés
1 / 91					



Lina Soualmia	Bases de Données	SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	Lina Soualmia	Bases de Données	SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés
3 / 91				4 / 91			

- Définition de chaque objet à partir d'un type décrivant
  - ▶ une structure de données se positionnant dans une hiérarchie d'héritage
  - ▶ des méthodes
- Utilisation d'un type :
  - ▶ Construire d'autres types
  - ▶ Définir une ou plusieurs tables objet
  - ▶ Définir une colonne d'une table relationnelle
  - ▶ Construire des vues objet

### Sous Oracle, 3 catégories d'objets :

- Objets colonne (column objects) :
  - ▶ stockés en tant que colonne structurée dans une table relationnelle
- Objets ligne (row objects) :
  - ▶ stockés en tant que ligne d'une table objet
  - ▶ possèdent un identificateur unique appelé OID (Object Identifier)
  - ▶ peuvent être indexés et partitionnés
- Objets non persistants : non stockés
  - ▶ ni dans une colonne d'une table relationnelle
  - ▶ ni dans une ligne d'une table objet
  - ▶ ces objets n'existent que durant l'exécution d'un programme PL/SQL

### Création d'un type

```
CREATE [OR REPLACE] TYPE Schéma.NomType
[AS OBJECT | UNDER] Schéma.NomSurType
```

définition de la structure

(Colonne1 type1, Colonne2 type2 ....)

définition du comportement

méthode1(paramètres1), méthode2(paramètres2)....  
[[NOT] INSTANTIABLE]

positionnement dans le graphe d'héritage

[[NOT] FINAL]

Lina Soualmia	Bases de Données	SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	Lina Soualmia	Bases de Données	SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés
5 / 91				6 / 91			

- Directives FINAL et NOT FINAL : positionnement d'un type dans le graphe d'héritage
- Directive NOT FINAL : à appliquer aux types génériques
- Par défaut, tout type est FINAL
- Un type FINAL ne peut servir à définir des sous-types

```
create type adresseT as object
(NRue NUMBER(3), Rue VARCHAR(40), Ville VARCHAR(30));

create type personnelT as object
(Nom VARCHAR(10), Prenom VARCHAR(10), Adresse
adresseT)
NOT FINAL;

create type enseignantT as object under personnelT
(Echelon NUMBER, Indice NUMBER)
FINAL;
```

Lina Soualmia	Bases de Données	SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	Lina Soualmia	Bases de Données	SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés
7 / 91				8 / 91			

- Directives **INSTANTIABLE** et **NOT INSTANTIABLE**
  - capacité d'instanciation d'un type
  - tous les types créés sont par défaut **INSTANTIABLE**
- NOT INSTANTIABLE** : similaire à la notion de classe abstraite
- Chaque type possède
  - un constructeur permettant de créer des objets (persistants ou non) à l'aide de la commande **NEW** ou au sein d'une commande **INSERT**
  - un constructeur (par défaut) et plusieurs dans le cas de surcharge
- Un type **NOT INSTANTIABLE** ne peut pas être **FINAL**
- Un sous-type **NOT INSTANTIABLE** peut hériter d'un type **INSTANTIABLE**

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

9 / 91

```
create type personnelT as object
(Nom VARCHAR(10), Prenom VARCHAR(10), Adresse
adresseT)
NOT INSTANTIABLE NOT FINAL;

create type enseignantT as object under personnelT
(Echelon NUMBER, Indice NUMBER)
INSTANTIABLE FINAL;
```

10 / 91

## Suppression d'un type

**DROP TYPE** NomType [**FORCE|VALIDATE**];

Directives :

- FORCE** : suppression du type même s'il y a des objets de ce type dans une base  
Oracle marque les colonnes dépendant de ce type par **UNUSED** et elles deviennent inaccessibles (non recommandé)
- VALIDATE** : vérification si les instances du type à supprimer peuvent être substitués par un sur-type

**DROP TYPE** PersonnelT **FORCE**;

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

11 / 91

## Extraction de la description d'un type

Définition de nouvelles vues du Dictionnaire des Données pour prendre en compte les types

```
create type emp_type as object
(ninsee VARCHAR2(13), age NUMBER, nom VARCHAR2(30));
```

Description de la structure au premier niveau d'un type :

SQL> DESC emp\_type

12 / 91

## Extraction de la description d'un type

Exemples de vues : (**USER\_...**, **DBA\_...**, **ALL\_...**)

Description :

- des collections : **USER\_COLL\_TYPES**
- des index sur les types : **USER\_INDEX\_TYPES**
- des types d'une manière générale : **USER\_TYPES**
- des attributs des types : **USER\_TYPE\_ATTRS**
- des méthodes des types : **USER\_TYPE\_METHODS**
- des versions des types : **USER\_TYPE VERSIONS**

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

13 / 91

Tables relationnelles					
— Table : MAGASINS2 SQL2	create table MAGASINS2	( NUMMAG INTEGER .	— Table : CLIENTS2 SQL2	create table CLIENTS2	( NUMCLI INTEGER .
NUMMAG NOMMAG	NUMMAG	NOMMAG CHAR(30) .	NUMCLI NOMCLI	NOMCLI CHAR(20) .	TELCLI CHAR(15) .
TELMAG	TELMAG	CHAR(15) .	ADRNUMCLI	ADRNUMCLI	VARCAR2(10) .
ADRNUMMAG	ADRNUMMAG	VARCAR2(10) .	ADRREUCLI	ADRREUCLI	VARCAR2(50) .
ADRREUEMAG	ADRREUEMAG	VARCAR2(50) .	ADRCPLCI	ADRCPLCI	VARCAR2(10) .
ADRCPMAG	ADRCPMAG	VARCAR2(10) .	ADRVILLECLI	ADRVILLECLI	VARCAR2(50) .
ADRVILLEMAG	ADRVILLEMAG	VARCAR2(50) .	ADRPAYSCLI	ADRPAYSCLI	VARCAR2(50) .
ADRPAYSMAG	ADRPAYSMAG	VARCAR2(50) .	constraint PK_CLIENTS2	constraint PK_CLIENTS2	primary key (NUMCLI);
constraint PK_MAGASINS2	constraint PK_MAGASINS2	primary key (NUMMAG) );	primary key (NUMCLI);	primary key (NUMCLI);	
insert into MAGASINS2 values (1, 'FB', '0145454545', '13', 'Avenue de la paix', '75015', 'Paris', 'France');					
1 1 FB 0145454545 13 Avenue de la paix 75015 Paris France					
2 2 FB 0155555555 20 Avenue de la liberté 06100 Nice France					
3 3 FB 0155555555 10 Avenue des Amis 6050 Bruxelles Belgique					
4 4 FB 71226002 10 Avenue du soleil 1001 Tunis Tunisie					
NUMCLI NOMCLI	NUMCLI NOMCLI	TELCLI ADRNU ADRREUCLI	ADRCP ADRVILLECL ADRPAYSCL	ADRCP ADRVILLECL ADRPAYSCL	
1 1 TRAIFOR 0645454545 13 Avenue de la paix 75015 Paris France					
2 2 CLEMENT 067080910 17 Avenue de la paix 75015 Paris France					
3 3 SOUCY 98980307 77 Route de la corniche 4001 Sousse Tunisie					

14 / 91

## Création d'un type :

Première extension du modèle relationnel : Types Abstraits de Données (TAD)

- TAD (contexte BD) :
  - Nouveau type d'attribut défini par l'utilisateur
  - Enrichissement de la collection existante de types disponibles par défaut  
(number,date,char,varchar...)
  - Structure de données partagée
    - Utilisation du type dans une ou plusieurs tables
    - Participation à la composition d'un ou plusieurs autres types
- Remarques :
  - Un TAD inclut des méthodes qui sont des procédures ou des fonctions
  - Elles permettent de manipuler les objets du type abstrait

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

15 / 91

```
create type ADRESSE_TYPE as object
( ADRNUM VARCHAR2(10),
ADRRLUE VARCHAR2(50),
ADRCPLCI VARCHAR2(10),
ADRVILLE VARCHAR2(50),
ADRPAYS VARCHAR2(50) )
/
create type MAG_TYPE as object
( NUMMAG INTEGER ,
NOMMAG CHAR(30),
TELMAG CHAR(15),
ADRMAG ADRESSE_TYPE )
/
create type CLI_TYPE as object
( NUMCLI INTEGER ,
NOMCLI CHAR(30),
TELCLI CHAR(15),
ADRCLI ADRESSE_TYPE )
```

16 / 91

```
create table MAGASINS3 OF MAGTYPE
(constraint PKMagasins3 primary key (NUMMAG));
create table CLIENTS3 OF CLITYPE
(constraint PKClients3 primary key (NUMCLI));
```

### Remarques :

- Un type ne peut pas contenir de contraintes :
  - ▶ NOT NULL
  - ▶ CHECK
  - ▶ UNIQUE
  - ▶ DEFAULT
  - ▶ PRIMARY KEY, FOREIGN...
- Les contraintes doivent être déclarées au niveau de la table objet
- Accès à la description des types à partir du Dictionnaire de Données :

Lina Soualmia		Bases de Données	17 / 91
SQL3		Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

SQL > select table\_name , object\_id\_type , table\_type\_own table\_type from user\_object\_tables;

Lina Soualmia		Bases de Données	18 / 91
SQL3		Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

### Création/description d'une table Exemples

```
SQL> desc clients2
Nom          NULL ?    Type
-----        -----   -----
NUMCLI        NOT NULL NUMBER(38)
NOMCLI
TELCLI
ADRNUMCLI
ADRUECLI
ADRCPLCI
ADRVILLECLI
ADRPAYSCLI

SQL> desc clients3
Nom          NULL ?    Type
-----        -----   -----
NUMCLI        NOT NULL NUMBER(38)
NOMCLI
TELCLI
ADRCLCI
```

- Si les Object identier (OID) sont basés sur la clé primaire : utilisation de l'option primary key

```
create table CLIENTS3 OF CLITYPE
(constraint PKClients3 primary key (NUMCLI))
object identifier is primary key;
```

- Index sur OID :

```
create table CLIENTS3 OF CLITYPE
(constraint PKClients3 primary key (NUMCLI));
object identifier is system generated OIDINDEX
ndxclients3;
```

Lina Soualmia		Bases de Données	20 / 91
SQL3		Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

### Instanciation : insertion

```
insert into MAGASINS3 values (MAGTYPE(1, 'FB', '0145454545',
ADRESSE_TYPE('13', 'Avenue de la paix', '75015', 'Paris', 'France')
insert into MAGASINS3 values (MAGTYPE(2, 'FB', '0155555555',
ADRESSE_TYPE('20', 'Avenue de la liberté', '06100', 'Nice', 'Fran
insert into MAGASINS3 values (MAGTYPE(3, 'FB', '0155555555',
ADRESSE_TYPE('10', 'Avenue des Amis', '6050', 'Bruxelles', 'Belgic
insert into MAGASINS3 values (MAGTYPE(4, 'FB', '71226002',
ADRESSE_TYPE('10', 'Avenue du soleil', '1001', 'Tunis', 'Tunisie'
```

```
SQL> select * from magasins3 ;
NUMMAG  NOMMAG  TELMAG  ADRCL(ADRNUM, ADRRUE, ADRCPL, ADRVILLE, ADRPAYS)
1       FB      0145454545  ADRESSE_TYPE('13', 'Avenue de la paix', '75015',
2       FB      0155555555  ADRESSE_TYPE('20', 'Avenue de la liberté', '06100',
3       FB      0155555555  ADRESSE_TYPE('10', 'Avenue des Amis', '6050',
4       FB      71226002  ADRESSE_TYPE('10', 'Avenue du soleil', '1001',
```

```
insert into CLIENTS3 values (CLITYPE(1, 'TRAIFOR', '0645454545',
ADRESSE_TYPE('13', 'Avenue de la paix', '75015', 'Paris', 'France')
insert into CLIENTS3 values (CLITYPE(2, 'CLEMENT', '0607080910',
ADRESSE_TYPE('17', 'Avenue de la paix', '75015', 'Paris', 'France')
insert into CLIENTS3 values (CLITYPE(3, 'SOUCY', '98980307',
ADRESSE_TYPE('77', 'Route de la corniche', '4001', 'Sousse', 'Tunis
```

```
SQL> Select * from clients3 ;
NUMCLI  NOMCLI  TELCLI  ADRCL(ADRNUM, ADRRUE, ADRCPL, ADRVILLE, ADRPAYS)
1       TRAIFOR  0645454545  ADRESSE_TYPE('13', 'Avenue de la paix', '75015',
2       CLEMENT  0607080910  ADRESSE_TYPE('17', 'Avenue de la paix', '75015',
3       SOUCY    98980307  ADRESSE_TYPE('77', 'Route de la corniche', '4001',
```

Lina Soualmia		Bases de Données	21 / 91
SQL3		Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

### Mises à jour Modifications/Suppressions de lignes ou d'objets

- Mise à jour d'une colonne standard
 

```
update CLIENTS3
set NOMCLI='CBON' where NUMCLI=2;
```
- Modification d'une colonne appartenant à un type imbriqué
 

```
update CLIENTS3 c
set c.ADRCLI.ADR.VILLE='MAVILLE' where
c.NUMCLI=2;
```
- Suppression d'objet
 

```
delete from CLIENTS3
where NUMCLI=3;
delete from CLIENTS3 c
where upper(c.ADRCLI.ADRPAYS)='FRANCE';
```

### Instanciation Table objet-relationnelle :

- Table dépendante d'un type
- Enregistrements (lignes) dans cette table considérés comme des objets car ils possèdent tous un OID unique

```
SQL> SELECT * FROM clients3 ;
NUMCLI  NOMCLI  TELCLI  ADRCL(ADRNUM, ADRRUE, ADRCPL, ADRVILLE, ADRPAYS)
1       TRAIFOR  0645454545  ADRESSE_TYPE('13', 'Avenue de la paix', '75015',
2       CLEMENT  0607080910  ADRESSE_TYPE('17', 'Avenue de la paix', '75015',
3       SOUCY    98980307  ADRESSE_TYPE('77', 'Route de la corniche', '4001',
```

Lina Soualmia		Bases de Données	24 / 91
SQL3		Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

## Utilisation de colonnes standards

```
select numcli, nomcli from clients3;
  NUMCLI NOMCLI
```

1 TRAIFOR
2 CLEMENT
3 SOUCY

## Utilisation d'une colonne appartenant à un type imbriqué

```
select numcli, nomcli, c.ADRCLI.ADRPAYS
from clients3 c;
  NUMCLI NOMCLI ADRCLI.ADRPAYS
```

1 TRAIFOR	France
2 CLEMENT	France
3 SOUCY	Tunisie

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

## Interrogations avec formatage

```
col nom format A10
col loc format A15
select numcli as cli, nomcli as nom,
       c.ADRCLI.ADRVILLE || ' ' || c.ADRCLI.ADRPAYS as loc
from clients3 c;
  CLI NOM LOC
```

1 TRAIFOR	Paris	France
2 CLEMENT	Paris	France
3 SOUCY	Sousse	Tunisie

## Interrogations avec contraintes

```
SQL> col c.ADRCLI.ADRPAYS format A10
SQL> col c.ADRCLI.ADRVILLE format A10
SQL> select numcli, nomcli, c.ADRCLI.ADRPAYS,
      2 c.ADRCLI.ADRVILLE from clients3 c
      3 WHERE upper(c.ADRCLI.ADRVILLE) like 'P%';
```

NUMCLI	NOMCLI	ADRCLI.ADRPAYS	ADRCLI.ADRVILLE
1	TRAIFOR	France	Paris
2	CLEMENT	France	Paris

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

## Tables imbriquées

Table imbriquée ([NESTED TABLE](#) : collection non ordonnée et non limitée d'éléments de [même type](#))

Exemple : table Département qui contient plusieurs employés

1 table contenant une colonne (table) : Association du type 1-N

NumDep	Budget	Employés		
		NInsee	Nom	Age

27 / 91

## Création

```
create type EmpType as object
(ninsee VARCHAR2(13), age NUMBER, nom VARCHAR2(30));
```

Création du type table

```
create type EmpsType as table of EmpType;
```

```
create type DepartementType as object
(NumDep VARCHAR(11), Budget NUMBER) employees EmpsType;
```

Création de la table maître

```
create table Departement of DepartementType
(primary key(NumDep))
nested table employes store as tabemp;
```

- clause [NESTED TABLE](#) : définition d'une table imbriquée
- clause [STORE AS](#) : nommage de la structure interne qui stocke les enregistrements de cette table imbriquée

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

```
SQL> desc departement
Nom
```

NULL	?	Type
NOT NULL	VARCHAR2(11)	
	NUMBER	
	EMPS_TYPE	

```
SQL> desc emps_type
emps_type TABLE OF EMP_TYPE
```

Nom	NULL	?	Type
NINSEE	VARCHAR2(13)		
AGE	NUMBER		
NOM	VARCHAR2(30)		

## Insertion des données dans une table imbriquée

```
insert into departement values ('D1', 100000, emps_type())
insert into departement values ('D2', 200000, emps_type())
```

```
SQL> select * from departement ;
  NUMDEP          BUDGET EMPLOYES(NINSEE, AGE, NOM)
```

D1	100000	EMPS_TYPE()
D2	200000	EMPS_TYPE()

31 / 91

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

32 / 91

Attention : dans l'exemple suivant, la table vide est non initialisée

```
insert into departement (numdep, budget)
values ('D3', 300000);

SQL> select * from departement ;
NUMDEP          BUDGET EMPLOYES(NINSEE, AGE, NOM)
-----  -----
D1              100000  EMPS_TYPE()
D2              200000  EMPS_TYPE()
D3              300000
```

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés



33 / 91

```
SQL> select * from departement ;
NUMDEP          BUDGET EMPLOYES(NINSEE, AGE, NOM)
-----  -----
D1              100000  EMPS_TYPE()
D2              200000  EMPS_TYPE()
D3              300000
D4              400000  EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'),
                           EMP_TYPE('N6', 26, 'Cici'),
                           EMP_TYPE('N7', 27, 'Didi'),
                           EMP_TYPE('N8', 28, 'Fifi'))
```

Remarque : La commande **INSERT** avec les constructeurs des types de la **NESTED TABLE** :

- stocke un objet dans la table
- initialise la table imbriquée associée avec des enregistrements

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés



35 / 91

### Insertion des données dans une table imbriquée

```
insert into departement values ('D4', 400000,
                                emps_type(emp_type('N5', 25, 'Bibi'),
                                           emp_type('N6', 26, 'Cici'),
                                           emp_type('N7', 27, 'Didi'),
                                           emp_type('N8', 28, 'Fifi')));
```

```
SQL> select * from departement ;
NUMDEP          BUDGET EMPLOYES(NINSEE, AGE, NOM)
```

D1	100000  EMPS_TYPE()
D2	200000  EMPS_TYPE()
D3	300000
D4	400000  EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'),                            EMP_TYPE('N6', 26, 'Cici'),                            EMP_TYPE('N7', 27, 'Didi'),                            EMP_TYPE('N8', 28, 'Fifi'))
D5	400000  EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'),                            EMP_TYPE('N6', 26, 'Cici'),                            EMP_TYPE('N7', 27, 'Didi'),                            EMP_TYPE('N8', 28, 'Fifi'))



Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

SQL> select * from departement ; NUMDEP          BUDGET EMPLOYES(NINSEE, AGE, NOM)
D1              100000  EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT')) D2              200000  EMPS_TYPE(EMP_TYPE('N2', 22, 'CLEMENTIN')) D3              300000 D4              400000  EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'),                            EMP_TYPE('N6', 26, 'Cici'),                            EMP_TYPE('N7', 27, 'Didi'),                            EMP_TYPE('N8', 28, 'Fifi'))

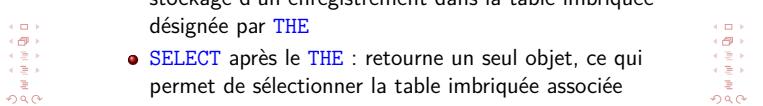
Remarques :

- Commande **INSERT INTO THE (SELECT ...)** : stockage d'un enregistrement dans la table imbriquée désignée par **THE**
- **SELECT** après le **THE** : retourne un seul objet, ce qui permet de sélectionner la table imbriquée associée

Insert avec **THE** dans une table imbriquée (D1 et D2 étaient initialisés à vide)

```
insert into THE (select d.employes from departement d
                 where d.numdep = 'D1')
                 values ('N1', 21, 'CLEMENT');
insert into THE (select d.employes from departement d
                 where d.numdep = 'D2')
                 values ('N2', 22, 'CLEMENTINE');
```

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés



37 / 91

Insert avec **THE** dans une table imbriquée (D3 n'était pas initialisé à vide)

Insert d'un employé dans le département D3 alors que celui-ci n'a pas été initialisé

```
SQL> insert into the (select d.employes from departement d
2 where d.numdep = 'D3') values ('N3', 23, 'NEMARCHEPAS'
insert into the (select d.employes from departement d
where d.numdep = 'D3')
```

\*  
ERREUR à la ligne 1 :  
ORA-22908: référence à une valeur de table NULL

Explications :

- ① Le département D3 est bien un objet de la table Departement
- ② mais il ne possède pas de table imbriquée car celle-ci n'a pas été créée lors de l'insertion.
- ③ ⇒ Il faut détruire l'objet D3 puis le recréer

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés



39 / 91

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés



40 / 91

## Mise à jour de la table principale

```
update departement d
set d.budget = d.budget * 1.5
where d.budget <= 200000 ;
```

SQL> select * from departement ;	NUMDEP	BUDGET	EMPLOYES(NINSEE, AGE, NOM)
D1	150000	EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT'))	
D2	300000	EMPS_TYPE(EMP_TYPE('N2', 22, 'CLEMENTINI'))	
D3	300000		
D4	400000	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N6', 26, 'Cici'), EMP_TYPE('N7', 27, 'Didi'), EMP_TYPE('N8', 28, 'Fifi'))	
D5	400000	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N8', 28, 'Fifi'))	

## Mise à jour de la table principale selon un prédictat dans la table imbriquée

```
update departement d set d.budget = d.budget + 777
where exists (select * from
the ( select dt.employes from departement dt
where dt.numdep =
d.numdep ) nt
where nt.age < 25 ) ;
```

- Description : requête qui retourne les employés de chaque département
- `select dt.employes from departement dt where dt.numdep=d.numdep;`
- Condition sur un attribut de la table imbriquée :
- `where nt.age < 25`
- Alias de la table imbriquée : nt

Lina Soualmia Bases de Données

SQL3 Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

Lina Soualmia Bases de Données

SQL3 Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

```
SQL> select * from departement ;
NUMDEP BUDGET EMPLOYES(NINSEE, AGE, NOM)
```

SQL> select * from departement ;	NUMDEP	BUDGET	EMPLOYES(NINSEE, AGE, NOM)
D1	150777	EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT'))	
D2	300777	EMPS_TYPE(EMP_TYPE('N2', 22, 'CLEMENTINE'))	
D3	300000		
D4	400000	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N6', 26, 'Cici'), EMP_TYPE('N7', 27, 'Didi'), EMP_TYPE('N8', 28, 'Fifi'))	
D5	400000	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N8', 28, 'Fifi'))	

## Mise à jour de la table principale selon un prédictat dans la table imbriquée

```
update departement d set d.budget = d.budget + 999
where exists
(select * from the
(select dt.employes from departement dt
where dt.numdep = d.numdep ) nt
where nt.age > 25 ) ;
```

Lina Soualmia Bases de Données

SQL3 Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

Lina Soualmia Bases de Données

SQL3 Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

```
SQL> select * from departement ;
NUMDEP BUDGET EMPLOYES(NINSEE, AGE, NOM)
```

SQL> select * from departement ;	NUMDEP	BUDGET	EMPLOYES(NINSEE, AGE, NOM)
D1	150777	EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT'))	
D2	300777	EMPS_TYPE(EMP_TYPE('N2', 22, 'CLEMENTINE'))	
D3	300000		
D4	400999	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N6', 26, 'Cici'), EMP_TYPE('N7', 27, 'Didi'), EMP_TYPE('N8', 28, 'Fifi'))	
D5	400999	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N8', 28, 'Fifi'))	

## Mise à jour dans la table imbriquée

```
update
the (select d.employes from departement d
where d.numdep = 'D2' ) nt
set nt.age = 44
where nt.ninsee = 'N2' ;
```

Lina Soualmia Bases de Données

SQL3 Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

Lina Soualmia Bases de Données

SQL3 Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

```
SQL> select * from departement ;
NUMDEP BUDGET EMPLOYES(NINSEE, AGE, NOM)
```

SQL> select * from departement ;	NUMDEP	BUDGET	EMPLOYES(NINSEE, AGE, NOM)
D1	150777	EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT'))	
D2	300777	EMPS_TYPE(EMP_TYPE('N2', 44, 'CLEMENTINE'))	
D3	300000		
D4	400999	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N6', 26, 'Cici'), EMP_TYPE('N7', 27, 'Didi'), EMP_TYPE('N8', 28, 'Fifi'))	
D5	400999	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N8', 28, 'Fifi'))	

## Suppression dans la table principale

```
delete from departement where numdep='D3'
```

```
SQL> select * from departement ;
NUMDEP BUDGET EMPLOYES(NINSEE, AGE, NOM)
```

SQL> select * from departement ;	NUMDEP	BUDGET	EMPLOYES(NINSEE, AGE, NOM)
D1	150777	EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT'))	
D2	300777	EMPS_TYPE(EMP_TYPE('N2', 44, 'CLEMENTINE'))	
D4	400999	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N6', 26, 'Cici'), EMP_TYPE('N7', 27, 'Didi'), EMP_TYPE('N8', 28, 'Fifi'))	
D5	400999	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N8', 28, 'Fifi'))	

Il est impossible de modifier plusieurs enregistrements de différentes tables imbriquées avec une seule commande UPDATE

Suppression à partir d'une valeur de la table imbriquée :  
élimination des départements qui emploient une personne dont  
le nom est FIFI

```
delete from departement d
  where exists (select * from
    the (select dt.employees from departement dt
          where dt.numdep = d.numdep) nt
  where upper(nt.nom) like '%FIFI%' );
```

```
SQL> select * from departement ;
NUMDEP      BUDGET EMPLOYES(NINSEE, AGE, NOM)
-----  -----
D1          150777 EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT'))
D2          300777 EMPS_TYPE(EMP_TYPE('N2', 44, 'CLEMENTINE'))
```

Lina Soualmia	Bases de Données	49 / 91
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

Lina Soualmia	Bases de Données	50 / 91
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

**Suppression d'une table imbriquée** : élimination des  
départements qui emploient une personne dont le nom est  
CLEMENT

```
delete the (select dt.employees
  from departement dt where dt.numdep = 'D1')
  where nt.nom = 'CLEMENT';

SQL> select * from departement ;
NUMDEP      BUDGET EMPLOYES(NINSEE, AGE, NOM)
```

```
D2          300777 EMPS_TYPE(EMP_TYPE('N2', 44, 'CLEMENTINE'))
```

Lina Soualmia	Bases de Données	51 / 91
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

Lina Soualmia	Bases de Données	52 / 91
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

```
select nt.ninsee , nt.nom
  from the (select dt.employees from departement dt
    where dt.numdep = 'D4') nt;
NINSEE      NOM
-----  -----
N5          Bibi
N6          Cici
N7          Didi
N8          Fifi
```

Lina Soualmia	Bases de Données	53 / 91
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

Lina Soualmia	Bases de Données	54 / 91
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

```
select nt.ninsee , nt.nom
  from the (select dt.employees from departement
    where dt.numdep = 'D4') nt where nt.age < 26;
NINSEE      NOM
-----  -----
N5          Bibi
```

Lina Soualmia	Bases de Données	55 / 91
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

```
SQL> select * from departement ;
NUMDEP      BUDGET EMPLOYES(NINSEE, AGE, NOM)
-----  -----
D1          150777 EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT'))
D2          300777 EMPS_TYPE(EMP_TYPE('N2', 44, 'CLEMENTINE'))
D4          400999 EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi')),
          EMP_TYPE('N6', 26, 'Cici'),
          EMP_TYPE('N7', 27, 'Didi'),
          EMP_TYPE('N8', 28, 'Fifi'))
D5          400999 EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi')),
          EMP_TYPE('N8', 28, 'Fifi'))
```

Lina Soualmia	Bases de Données	56 / 91
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés	

Requête : Quel est le nombre d'employés du département D4 ?

```
select COUNT(*) "Nombre d'employés"
      from the (select dt.employees from departement dt
                 where dt.numdep = 'D4') nt ;
```

Nombre d'employés

4

SQL> select \* from departement;

NUMDEP	BUDGET	EMPLOYES(NINSEE, AGE, NOM)
D1	150777	EMPS_TYPE(EMP_TYPE('N1', 21, 'CLEMENT'))
D2	300777	EMPS_TYPE(EMP_TYPE('N2', 44, 'CLEMENTINE'))
D4	400999	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N6', 26, 'Cici'), EMP_TYPE('N7', 27, 'Didi'), EMP_TYPE('N8', 28, 'Fifi'))
D5	400999	EMPS_TYPE(EMP_TYPE('N5', 25, 'Bibi'), EMP_TYPE('N8', 28, 'Fifi'))

Quels sont les numéros et les noms des employés des départements D1 et D2 ?

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

```
select select nt.ninsee , nt.nom from the
      (select dt.employees from departement dt
                 where dt.numdep = 'D1') nt
union
select nt.ninsee , nt.nom from the
      (select dt.employees from departement dt
                 where dt.numdep = 'D2') nt;
```

NINSEE

NOM

N1	CLEMENT
N2	CLEMENTINE

57 / 91

### Plusieurs tables imbriquées : Création

Regroupement des tables imbriquées Professeurs et Formations dans la table Cours

NumC	Titre	Professeurs	Formations
		Nom	Spécialité
		Filière	Horaire

```
create type prof_type as object
  (nom varchar2(30), specialite varchar2(30))
/
create type profs_type as table of prof_type
/
create type formation_type as object
  (filiere varchar2(30), horaire number(5))
/
create type formations_type as table of formation_type
```

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Insertion d'un objet dans la table Cours, sans le lier à des professeurs ou à des formations

```
insert into cours values ('BD', 'Bases de Données',
                           profs_type(), formations_type());
```

select \* from cours;

NUMC	TITRE	PROFESSEURS(NOM, SPECIALITE)	FORMATIONS(FILIERE, HORAI
BD	Bases de Données	PROFS_TYPE()	FORMATIONS_TYPE()

```
create type prof_type as object
  (nom varchar2(30), specialite varchar2(30))
/
create type profs_type as table of prof_type
/
create type formation_type as object
  (filiere varchar2(30), horaire number(5))
/
create type formations_type as table of formation_type
/
create type cours_type as object
  (numc varchar2(5), titre varchar2(15),
   professeurs profs_type, formations formations_type)
/
create table cours of cours_type
  (constraint pk_cours primary key (numc))
  nested table professeurs store as tabprofs,
  nested table formations store as tabformations ;
```

61 / 91

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Insertion, avec **VALUES** dans 2 tables imbriquées :

```
insert into cours values ('DW', 'Data WareHouse',
                           profs_type(prof_type('Clémence', 'BD'),
                                       prof_type('Adam', 'BD')),
                           formations_type(formation_type('Master1', 100),
                                           formation_type('DESS EID', 200),
                                           formation_type('DEA AIOC', 200)));
```

NUMC	TITRE	PROFESSEURS(NOM, SPECIALITE)	FORMATIONS(FILIERE, HORAI
BD	Bases de Données	PROFS_TYPE()	FORMATIONS_TYPE()
DW	Data WareHouse	PROFS_TYPE()                            PROF_TYPE('Clémence', 'BD'),                            PROF_TYPE('Adam', 'BD'))	FORMATIONS_TYPE()                            FORMATION_TYPE('Master1'),                            FORMATION_TYPE('DESS EID'),                            FORMATION_TYPE('DEA AIOC'))

Lina Soualmia Bases de Données

Lina Soualmia Bases de Données

63 / 91

64 / 91

### Insertion, avec **VALUES** dans 2 tables imbriquées

```
insert into cours values ('BDA',
    'Bases de Données Avancées',
    profs_type(prof_type('Clémence', 'BD'),
    prof_type('Traifor', 'BD'),
    prof_type('Le Bon', 'BD')),

formations_type(
formation_type('Master 2P', 200),
formation_type('Master 2R', 200)));
```

Lina Soualmia | Bases de Données

SQL3 | Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

65 / 91

### L'affichage SQLPlus est très mauvais

NumC	Titre	Professeurs		Formations	
		Nom	Spécialité	Filière	Horaire
BD	Bases de données	Clémence	BD	Master 1	100
		Adam	BD	Master 2P	200
		Clémence	BD	Master 2R	200
		Traifor	BD	Master 2P	
DW	DataWareHouse	Le Bon	BD	Master 2R	
BDA	Bases de données avancés				



66 / 91

### Insertion, avec **THE** et **VALUES** dans 2 tables imbriquées

Enregistrement des données : les professeurs Traifor et Parisi enseignent les BD

```
insert into the (select c.professeurs from cours c
    where numc = 'BD') values ('Traifor', 'SI');
insert into the (select c.professeurs from cours c
    where numc = 'BD') values ('Parisi', 'DM');
```

Lina Soualmia | Bases de Données

SQL3 | Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

67 / 91

```
select * from cours;
```

NumC	Titre	Professeurs		Formations	
		Nom	Spécialité	Filière	Horaire
BD	Bases de données	Traifor	SI		
		Parisi	DM		
		Clémence	BD	Master 1	100
		Adam	BD	Master 2P	200
DW	DataWareHouse				
BDA	Bases de données avancés	Clémence	BD	Master 2P	
		Traifor	BD	Master 2R	
		Le Bon	BD		



68 / 91

### Insertion, avec **THE** et **VALUES** dans 2 tables imbriquées

Le cours BD :

- appartient au cursus INFO1
- requiert un volume horaire de 70 heures

```
insert into the (select c.formations from cours c
    where numc = 'BD') values ('INFO1', 70);
```

Lina Soualmia | Bases de Données

SQL3 | Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

69 / 91

```
select * from cours;
```

NumC	Titre	Professeurs		Formations	
		Nom	Spécialité	Filière	Horaire
BD	Bases de données	Traifor	SI	INFO1	70
		Parisi	DM		
		Clémence	BD	Master 1	100
		Adam	BD	Master 2P	200
DW	DataWareHouse				
BDA	Bases de données avancés	Clémence	BD	Master 2P	
		Traifor	BD	Master 2R	
		Le Bon	BD		



70 / 91

### Insertion, avec **THE** et **SELECT**, dans 2 tables imbriquées

Le cours BD doit être enseigné désormais dans toutes les filières concernées par la matière DW à condition que celles-ci aient un volume de moins de 150 heures

```
insert into the (select c.formations from cours c
    where c.numc = 'BD')
select nesteddf.filiere, nesteddf.horaire
    from the (select c.formations from cours c
        where c.numc = 'DW') nesteddf
    where nesteddf.horaire < 150;
```

Lina Soualmia | Bases de Données

SQL3 | Types de données  
Tables imbriquées  
Plusieurs tables imbriquées  
Tableaux pré-dimensionnés

69 / 91

```
select * from cours;
```

NumC	Titre	Professeurs		Formations	
		Nom	Spécialité	Filière	Horaire
BD	Bases de données	Traifor	SI		70
		Parisi	DM	Master 1	
		Clemence	BD	Master 1	100
		Adam	BD	Master 2P	200
DW	DataWareHouse				
BDA	Bases de données avancés	Clemence	BD	Master 2P	
		Traifor	BD	Master 2R	
		Le Bon	BD		



72 / 91

Dans la matière Data Warehouse, le professeur Adam est remplacé par le professeur Saitout et les horaires pour le Master 2P augmentent de 30%

```
update the (select c.professeurs from cours c
            where c.titre = 'Data Warehouse') nestedprf
set nestedprf.nom = 'Saitout' where nestedprf.nom = 'Adam';

update the (select c.formations from cours c
            where c.titre = 'Data Warehouse') nestedfrm
set nestedfrm.horaire = horaire * 1.3
            where nestedfrm.filiere like 'Master 2P%';
```

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

### Explications :

- Modification à l'aide de la commande **UPDATE** d'un ou plusieurs attributs dans une des deux tables imbriquées de la table Cours
- Modification d'un professeur et une formation dans le cadre d'une matière donnée : 2 requêtes **UPDATE** distinctes (car les 2 tables imbriquées sont concernées)
- Nécessité de recourir à un alias pour identifier l'objet dans la table imbriquée

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

Pour la matière DW, remplacement de la filière Master1 par la filière MASTER 2 et enregistrement d'un volume horaire de 150 heures

```
update the (select c.formations from cours c
            where c.titre = 'Data Warehouse') nestedfrm
set nestedfrm.horaire = 150
            where nestedfrm.filiere = 'Master1';

update the (select c.formations from cours c
            where c.titre = 'Data Warehouse') nesteddf
set nestedfrm.filiere = 'MASTER 2'
            where nestedfrm.filiere = 'Master1';
```

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

### Suppression dans 2 tables imbriquées :

Le professeur Parisi n'enseigne plus la matière BD.  
Enregistrement de cette information

```
delete the (select c.professeurs
from cours c where c.numc = 'BD') nt
where nt.nom = 'Parisi';
```

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

### Suppression dans 2 tables imbriquées

La filière Master1 n'inclut plus la matière BD dans son cursus.  
Enregistrement de cette information

```
delete the (select c.formations
from cours c where c.numc = 'BD') nt
where nt.filiere = 'Master1';
```

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

NumC	Titre	Professeurs	Formations
BD	Bases de données	Traifor	SI
		Parisi	DM
DW	DataWarehouse	Clemence	BD
		Saitou	BD
		Clemence	MASTER 2
		Saitou	Master 2P
BDA	Bases de données avancés	Clemence	Master 2R
		Traifor	BD
		Le Bon	BD

NumC	Titre	Professeurs		Formations	
		Nom	Spécialité	Filière	Horaire
BD	Bases de données	Trafor	SI	INFO1	70
		Clémence	BD	MASTER 2	100
		Saitou	BD	Master 2P	290
DW	DataWareHouse			Master 2R	200
		Clémence	BD	Master 2P	
		Trafor	BD	Master 2R	
BDA	Bases de données avancés	Le Bon	BD		

Plusieurs niveaux d'imbrication :

NumC	Titre	Professeurs		Formations	
		Nom	Spécialité	Filière	Dates
BD	Bases de données	Trafor	SI	INFO1	70
		Clemence	BD	MASTER 2	100
		Saitou	BD	Master 2P	290
DW	DataWareHouse			Master 2R	200
		Clémence	BD	Master 2P	
		Trafor	BD	Master 2R	
BDA	Bases de données avancés	Le Bon	BD		

ORACLE 10g n'offre que 2 types de collections :

- table imbriquée **NESTED TABLE** : collection ordonnée et non limitée en nombre d'éléments
- tableau prédimensionné **VARRAY** : collection d'éléments de même type, ordonnée et limitée en taille

pas de possibilité d'implanter plusieurs niveaux d'imbrication

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

## Tableaux pré-dimensionnés

- VARRAY** (Varrying ARRAY) : collection ordonnée et limitée d'éléments de même type
- Si le nombre d'éléments maximum contenus dans une table imbriquée est connu à priori :
  - possibilité d'utiliser un tableau de type **VARRAY** à la place d'une table imbriquée

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Exemple : stockage de 3 numéros de téléphone maximum par professeur

Professeurs :

NumP	NomP	Adresse				Téléphones
		AdrNum	AdrRue	AdrVille	AdrCP	

## Création :

```
create type AAdresse_type as object
(AdrNum varchar2(10), AdrNom varchar2(30),
AdrVille varchar2(20), AdrCP varchar2(5))
/
create type tel_type as object (NumTel varchar2(20))
/
create type tels_type as varray(3) of tel_type
/
create type professeur_type as object
(numP varchar2(5), nomP varchar2(20),
Adresse AAdresse_type, Telephones tels_type)
/
create table professeurs of professeur_type
(constraint pk_professeurs primary key (numP))
```

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

## Insertion : **INSERT** avec **VALUES**

Stockage de 3 objets de type Professeur avec respectivement aucun, trois et deux numéros de téléphone (enregistrements du **VARRAY**)

```
insert into professeurs values ('P1', 'Clémence',
AAdresse_type(7, 'Avenue de la Paix', 'Paris', '75009'),
tels_type());
insert into professeurs values ('P2', 'Adam',
AAdresse_type(77, 'Rue de la liberté', 'Paris', '75015'),
tels_type(tel_type('01 53 80 07 99'),
tel_type('06 14 56 07 06'),
tel_type('01 49 40 07 40')));
insert into professeurs values ('P3', 'Saitou',
AAdresse_type(1, 'Rue de la liberté', 'Paris', '75015'),
tels_type(tel_type('01 53 80 53 80'),
tel_type('06 14 56 14 77'), NULL));
```

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Insertion :

## **INSERT** dans un **VARRAY** avec PL/SQL

Avec les tableaux **VARRAY** l'opérateur **THE** n'est pas opérationnel

Pour manipuler les tableaux, il est nécessaire d'utiliser un programme PL/SQL

```
DECLARE
new_tels tels_type :=tels_type(tel_type('01 55 55 55 55'),
tel_type('06 06 98 98 98'),
tel_type('01 40 40 40 40'))
BEGIN
update professeurs
set telephones = new_tels
where numP = 'P1';
END;
/
```

NumP	NomP	Adresse				Téléphones
		AdrNum	AdrRue	AdrVille	AdrCP	
P1	Clémence	77	Avenue de la paix	Paris	75009	NULL
						NULL
						NULL
P2	Adam	7	Rue de la liberté	Paris	75015	01 53 80 07 99
						06 14 56 07 06
						01 49 40 07 40
P3	Saitou	1	Rue de la liberté	Paris	75015	01 53 80 53 80
						06 14 56 14 77
						NULL

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

Lina Soualmia	Bases de Données
SQL3	Types de données Tables imbriquées Plusieurs tables imbriquées Tableaux pré-dimensionnés

NumP	NomP	Adresse				Téléphones	
		AdrNum	AdrRue	AdrVille	AdrCP	NumTel	
P1	Clémence	77	Avenue de la paix	Paris	75009	01 55 55 55 55 06 06 98 98 98 01 40 40 40 40	
P2	Adam	7	Rue de la liberté	Paris	75015	01 53 80 07 99 06 14 56 07 06 01 49 40 07 40	
P3	Saitou	1	Rue de la liberté	Paris	75015	01 53 80 53 80 06 14 56 14 77 NULL	

Insérer un seul numéro de téléphone pour le professeur P1 et le placer au 2ème rang dans le tableau téléphones

Rédaction ci-dessous de l'instruction d'affectation :

```
new_tels tels_type := tels_type(NULL,
                                  tel_type('06 06 98 98 98'), NULL);
```

## Conclusion

Comparaison entre **NESTED TABLE** et **VARRAY** :

- A vérifier selon les versions d'Oracle
- Possibilité de définir un index dans un **NESTED TABLE**
  - ▶ Le nombre d'éléments n'est pas limité dans une table imbriquée
- Pas de possibilité de définir d'index dans un **VARRAY**
  - ▶ Le nombre d'éléments est limité dans un tableau pré-dimensionné
- Possibilité d'accéder directement aux enregistrements stockés dans les deux 2 structures de données (fonctions : **EXISTS,FIRST,LAST ...**)
- Performances ? : **NESTED TABLE > VARRAY**