



PL SQL

Séance 2

Pr. M'barek ELHALOUI

Révision/ Questions

- Pourquoi PLSQL ? Caractéristiques, avantages
- Structure d'un programme PL SQL
- SET SERVEROUTPUT ON et DBMS_OUTPUT.PUT_LINE
- Les variables en PLSQL : Types, déclaration, ..
- Questions ?

Variables composées : Les enregistrements

- Type **RECORD (Enregistrement)** : Quand tous les attributs sont d'un type SQL, une variable de type record peut représenter une ligne d'une table relationnelle.
- Déclaration du type Record :
 - Soit par référence à une structure de table ou de curseur en utilisant **ROWTYPE** :
nom-variable nom-table%ROWTYPE ;
 - Soit par définition du type enregistrement. Se fait en deux étapes :
 - Déclaration du type enregistrement :
TYPE *nom-du-type-record* IS RECORD
(attribut1 type-attribut1,
attribut2 type-attribu2,
***...)* ;**
 - Déclaration de la variable de type enregistrement :
nom-variable nom-du-type-record ;

Variables composées : Les enregistrements

- Type **RECORD** : Exemple

```
TYPE T_Rec_Client IS RECORD
(NUM NUMBER (6,2)
NOM VARCHAR2(30),
PRENOM VARCHAR2(30),
);

Cli T_Rec_Client ;
```

Variables composées : Les enregistrements

- Type **RECORD** : Exemple 2

```

TYPE T_REC_EMP IS RECORD
(NUM E_EMP.EMPLOYEE_ID%TYPE,
NOM E_EMP.FIRST_NAME%TYPE,
PRE E_EMP.LAST_NAME%TYPE,
SALAIRE E_EMP.SALARY%TYPE,

) ;

V_EMPL T_REC_EMP ;

```

Accès

V_EMPL.num
V_EMPL.NOM
V_EMPL.Pre
V_EMPL.Salaire

Table E_EMP

Column Name	Data Type	Nullable	Data Default	COLUMN ID	Primary Key
EMPLOYEE_ID	NUMBER(6,0)	No	(null)	1	1
FIRST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
LAST_NAME	VARCHAR2(25 BYTE)	No	(null)	3	(null)
EMAIL	VARCHAR2(25 BYTE)	No	(null)	4	(null)
PHONE_NUMBER	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)
HIRE_DATE	DATE	No	(null)	6	(null)
JOB_ID	VARCHAR2(10 BYTE)	No	(null)	7	(null)
SALARY	NUMBER(8,2)	Yes	(null)	8	(null)
COMMISSION_PCT	NUMBER(2,2)	Yes	(null)	9	(null)
MANAGER_ID	NUMBER(6,0)	Yes	(null)	10	(null)
DEPARTMENT_ID	NUMBER(4,0)	Yes	(null)	11	(null)

Variables composées : Les Tables

- Type **Table** ou **Varry** (Variable-size array)
- Structure composée d'éléments d'un même type *scalaire*
- L'accès à un élément de la table s'effectue grâce à un indice, ou clé primaire
- Cet index est déclaré de type **BINARY_INTEGER** (valeurs entières signées)
- Déclaration du type TABLE :
TYPE *nom-du-type-table*
IS TABLE OF *type-argument*
INDEX BY BINARY_INTEGER;
nom-variable nom-du-type-table;

Variables composées : Les Tables

- Exemple : Déclaration de Table

```
DECLARE
TYPE TabNoms IS TABLE OF VARCHAR2(20)
INDEX BY BINARY_INTEGER;
TabEtudiants tabNoms;
i BINARY_INTEGER;
BEGIN
TabEtudiants(3) := 'BenOmar';
i := 7;
TabEtudiants(i) := 'BenKamal';
END;
/
```

Variables composées : Les Tables

- Exemple 2 : Tableau dynamique ou collection

```
DECLARE TYPE EnameTabTyp IS TABLE OF E_EMP.EMPLOYEE_ID%TYPE INDEX BY BINARY_INTEGER;
```

```
DECLARE TYPE SalTabTyp IS TABLE OF E_EMP.SALARY%TYPE NOT NULL INDEX BY BINARY_INTEGER;
```

```
DECLARE TYPE EmpTabTyp IS TABLE OF E_EMP%ROWTYPE INDEX BY BINARY_INTEGER;
```

```
DECLARE
```

```
TYPE TimeRecTyp IS RECORD ( hour SMALLINT := 0, minute SMALLINT := 0, second SMALLINT := 0);
```

```
TYPE TimeTabTyp IS TABLE OF TimeRecTyp INDEX BY BINARY_INTEGER;
```


Variables composées : Les Tables

Méthodes utilisées avec les tables PL/SQL :

- **EXISTS(n)** : TRUE si le nème élément existe
- **COUNT** : Nombre d'éléments
- **FIRST** ou **LAST** : Plus petit et plus grand valeur d'index
- **PRIOR(n)** : Valeur de l'index qui précède l'index n
- **NEXT(n)** : Valeur de l'index qui suit l'index n
- **DELETE** : Supprime tous les éléments du tableau
- **DELETE(n)** : Supprime le nème élément
- **DELETE(n,m)** : Supprime les éléments de n à m

Affectations des variables

- Donner une valeur à une variable se fait à travers :
 - `:=` (Affectation)
 - la directive **INTO** de la requête SELECT
- Exemples :

```
DECLARE
dateNaissance Date ;
dateNaissance := '15/01/2000';

BEGIN END;
```

```
BEGIN
select nomEmpl INTO nom
      from EMPLOYE
      where matricule = '509';
END;
```

Affectations des variables

- **SelectInto**

- Pour une sélection mono-tuple (un seul enregistrement) , le résultat est affecté avec la clause INTO à une variable PLSQL.
- Pour plusieurs tuples il faut une variable applé CURSOR.

- **Conflit de noms :**

- Si une variable porte le même nom qu'une colonne d'une table, c'est la colonne qui l'emporte;
- Pour éviter ça, le plus simple est de ne pas donner de nom de colonne à une variable !

Exercice

- Que permet de faire le programme suivant :

```
DECLARE
EmpRec E_EMP%ROWTYPE;
salMax E_EMP.salary%TYPE;

BEGIN
SELECT * INTO EmpRec
FROM E_EMP WHERE EMPLOYEE_ID='705';
dbms_output.putline( EmpRec.FIRST_NAME || ' ' || EmpRec.LAST_NAME || ' ' || EmpRec.Salary);
END;
```

- Ecris un programme qui permet de sélectionner le salaire max