

Déclarer des variables

Objectifs

A la fin de ce chapitre, vous pourrez :

- **reconnaître un bloc PL/SQL de base et ses différentes sections**
- **décrire la signification des variables en PL/SQL**
- **déclarer des variables PL/SQL**
- **exécuter un bloc PL/SQL**

Structure d'un bloc PL/SQL

DECLARE (facultatif)

Variables, curseurs, exceptions définies par l'utilisateur

BEGIN (obligatoire)

- Instructions SQL
- Instructions PL/SQL

EXCEPTION (facultatif)

Actions à réaliser lorsque des erreurs se produisent

END ; (obligatoire)

DECLARE

• • •

BEGIN

• • •

EXCEPTION

• • •

END ;

Exécuter des instructions et des blocs PL/SQL

```
DECLARE
    v_variable  VARCHAR2(5);
BEGIN
    SELECT column_name
    INTO v_variable
    FROM table_name;
EXCEPTION
    WHEN exception_name THEN
        ...
END;
```

DECLARE

...

BEGIN

...

EXCEPTION

...

END;

Types de bloc

Anonyme

```
[DECLARE]

BEGIN
    --statements

[EXCEPTION]

END ;
```

Procédure

```
PROCEDURE name
IS

BEGIN
    --statements

[EXCEPTION]

END ;
```

Fonction

```
FUNCTION name
RETURN datatype
IS
BEGIN
    --statements
    RETURN value;
[EXCEPTION]

END ;
```

Structures de programme

DECLARE

• • •

BEGIN

• • •

EXCEPTION

• • •

END ;

Structures des outils

Blocs anonymes

Procédures ou fonctions
applicatives

Packages applicatifs

Déclencheurs applicatifs

Types d'objet

Structures du serveur de bases de données

Blocs anonymes

Procédures ou fonctions
stockées

Packages stockés

Déclencheurs de base de
données

Types d'objet

Utiliser des variables

Les variables peuvent être utilisées pour les raisons suivantes :

- **stockage temporaire de données**
- **manipulation de valeurs stockées**
- **possibilité de réutilisation**
- **facilité de maintenance**

Traiter les variables en PL/SQL

- **Déclarer et initialiser les variables dans la section déclarative**
- **Affecter de nouvelles valeurs aux variables dans la section exécutable**
- **Transmettre des valeurs aux blocs PL/SQL via des paramètres**
- **Afficher les résultats via des variables de sortie**

Types de variable

- **Variables PL/SQL :**
 - scalaires
 - composites
 - référencées
 - LOB (Large Object)
- **Variables non PL/SQL : variables attachées et variables hôte**

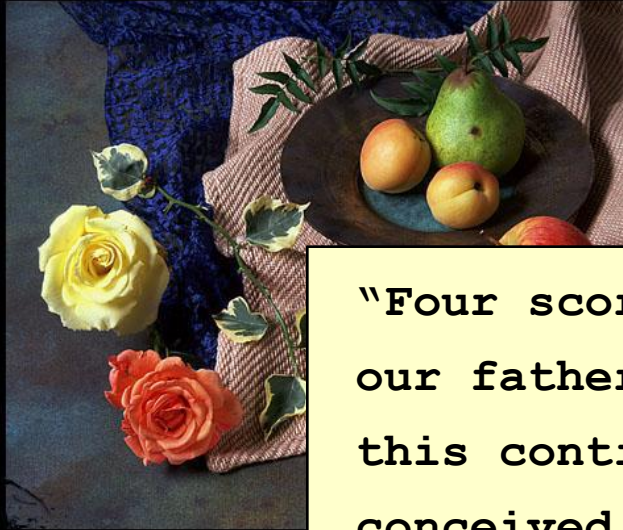
Utiliser des variables *iSQL*Plus* dans des blocs PL/SQL

- Le langage PL/SQL ne permet pas les entrées/sorties
- Vous pouvez référencer des variables de substitution dans un bloc PL/SQL en les faisant précéder d'un signe esperluette
- Les variables attachées ou variables hôte *iSQL*Plus* peuvent être utilisées pour transmettre des valeurs d'exécution du bloc PL/SQL vers l'environnement *iSQL*Plus*

Types de variable

25-JAN-01

TRUE



256120.08

"Four score and seven years ago
our fathers brought forth upon
this continent, a new nation,
conceived in LIBERTY, and dedicated
to the proposition that all men
are created equal."



Atlanta

Déclarer des variables PL/SQL

Syntaxe :

```
identifier [CONSTANT] datatype [NOT NULL]  
[:= | DEFAULT expr];
```

Exemples :

```
DECLARE  
  v_hiredate      DATE;  
  v_deptno        NUMBER(2) NOT NULL := 10;  
  v_location      VARCHAR2(13) := 'Atlanta';  
  c_comm          CONSTANT NUMBER := 1400;
```

Règles relatives à la déclaration de variables PL/SQL

- Suivre les conventions d'appellation
- Initialiser les variables NOT NULL et CONSTANT
- Déclarer un identificateur par ligne
- Initialiser les identificateurs en utilisant l'opérateur d'affectation (:=) ou le mot réservé DEFAULT

```
identifieur := expr;
```

Règles d'appellation

- Deux variables peuvent porter le même nom si elles sont dans des blocs distincts
- Le nom de la variable (identificateur) doit être différent du nom des colonnes de table utilisées dans le bloc.

```
DECLARE
    employee_id NUMBER(6);
BEGIN
    SELECT    employee_id
    INTO      employee_id
    FROM      employees
    WHERE     last_name = 'Kochhar';
END;
/
```

**Adopter une
convention
d'appellation pour
les identificateurs
PL/SQL :
par exemple,
v_employee_id**

Initialisation de variables et mots-clés

- Opérateur d'affectation (**:=**)
- Mot-clé **DEFAULT**
- Contrainte **NOT NULL**

Syntaxe :

```
identifiant := expr;
```

Exemples :

```
v_hiredate := '01-JAN-2001';
```

```
v_ename := 'Maduro';
```

Types de données scalaires

- Ils stockent une seule valeur
- Ils n'ont pas de composant interne

25-OCT-99

256120.08

"Four score and seven years
ago our fathers brought
forth upon this continent, a
new nation, conceived in
LIBERTY, and dedicated to
the proposition that all men
are created equal"

TRUE

Atlanta

Types de données scalaires de base

- CHAR [(*maximum_length*)]
- VARCHAR2 (*maximum_length*)
- LONG
- LONG RAW
- NUMBER [(*precision*, *scale*)]
- BINARY_INTEGER
- PLS_INTEGER
- BOOLEAN

Types de données scalaires de base

- DATE
- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- TIMESTAMP WITH LOCAL TIME ZONE
- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND

Déclaration de variables scalaires

Exemples :

```
DECLARE
  v_job          VARCHAR2 (9) ;
  v_count        BINARY_INTEGER := 0;
  v_total_sal    NUMBER(9,2) := 0;
  v_orderdate    DATE := SYSDATE + 7;
  c_tax_rate     CONSTANT NUMBER(3,2) := 8.25;
  v_valid        BOOLEAN NOT NULL := TRUE;
  ...
```

Attribut %TYPE

- **Déclarer une variable d'après :**
 - une définition de colonne de base de données
 - une autre variable précédemment déclarée
- **Faire précéder %TYPE :**
 - de la table et de la colonne de base de données
 - du nom de la variable précédemment déclarée

Déclarer des variables avec l'attribut %TYPE

Syntaxe :

```
identifier        Table.column_name%TYPE;
```

Exemples :

```
...  
  v_name            employees.last_name%TYPE;  
  v_balance        NUMBER(7,2);  
  v_min_balance    v_balance%TYPE := 10;  
...
```

Déclarer des variables booléennes

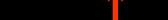
- Seules les valeurs **TRUE**, **FALSE** et **NULL** peuvent être affectées à une variable booléenne
- Les variables sont comparées en utilisant les opérateurs logiques **AND**, **OR** et **NOT**
- Les variables renvoient toujours **TRUE**, **FALSE** ou **NULL**
- Des expressions de type arithmétique, caractère ou date peuvent être utilisées pour renvoyer une valeur booléenne.

Types de données composites

TRUE	23-DEC-98	ATLANTA	
------	-----------	---------	---

Structure d'une table PL/SQ

1	SMITH
2	JONES
3	NANCY
4	TIM



BINARY_INTEGER

VARCHAR2

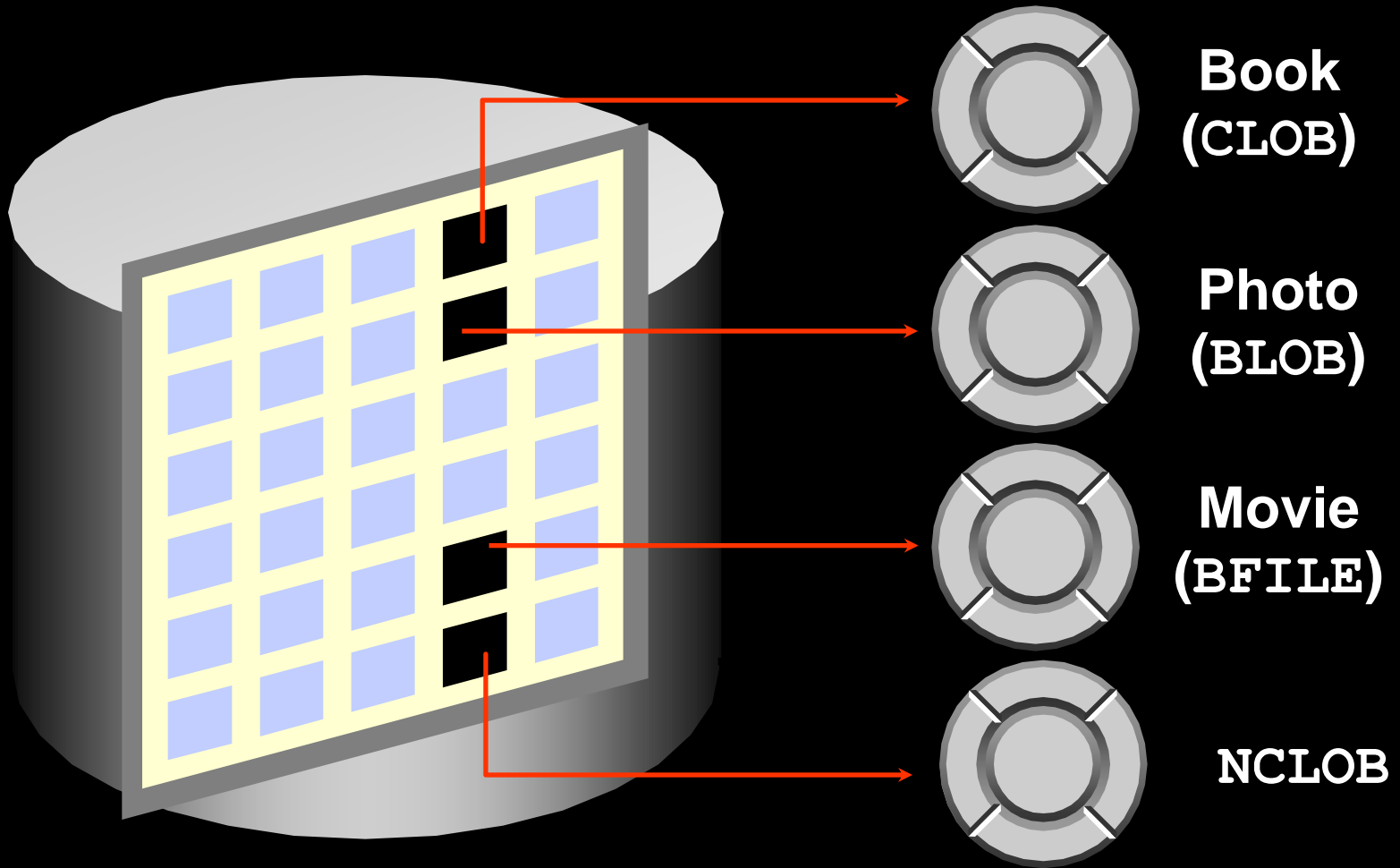
Structure d'une table PL/SQL

1	5000
2	2345
3	12
4	3456

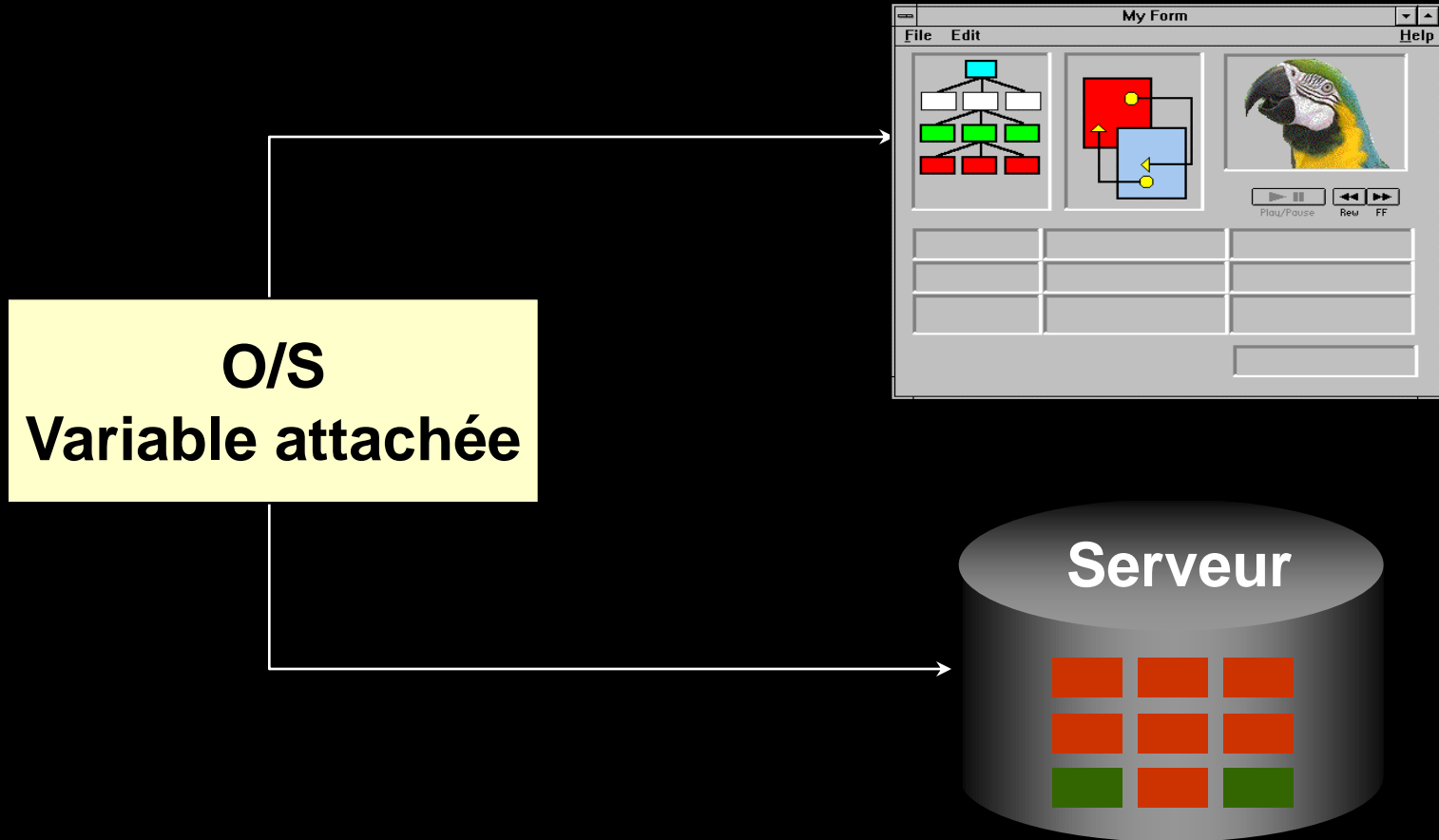
NUMBER

BINARY INTEGER

Variables de type LOB



Variables attachées



Utiliser des variables attachées

En PL/SQL, pour référencer une variable attachée, il faut faire précéder son nom du signe deux-points (:)

Exemple :

```
VARIABLE      g_salary NUMBER
BEGIN
  SELECT      salary
  INTO        :g_salary
  FROM        employees
  WHERE       employee_id = 178;
END;
/
PRINT g_salary
```

Référencer des variables non PL/SQL

Stocker le salaire annuel dans une variable hôte
*iSQL*Plus*.

```
:g_monthly_sal := v_sal / 12;
```

- Référencer des variables non PL/SQL en tant que variables hôte
- Faire précéder les références à l'aide d'un signe deux-points (:)

DBMS_OUTPUT.PUT_LINE

- Procédure de package fournie par Oracle
- Autre méthode d'affichage des données d'un bloc PL/SQL
- Doit être activé dans *iSQL*Plus* via
SET SERVEROUTPUT ON

```
SET SERVEROUTPUT ON
```

```
DEFINE p_annual_sal = 60000
```

```
DECLARE
```

```
    v_sal NUMBER(9,2) := &p_annual_sal;
```

```
BEGIN
```

```
    v_sal := v_sal/12;
```

```
    DBMS_OUTPUT.PUT_LINE ('The monthly salary is ' ||  
                           TO_CHAR(v_sal));
```

```
END;
```

```
/
```

Synthèse

Ce chapitre vous a appris les règles suivantes :

- **Les blocs PL/SQL sont composés des sections suivantes :**
 - **déclarative (facultative)**
 - **exécutable (obligatoire)**
 - **traitement des exceptions (facultative)**
- **Un bloc PL/SQL peut être un bloc, une procédure ou une fonction anonyme.**

DECLARE

• • •

BEGIN

• • •

EXCEPTION

• • •

END ;

Synthèse

Ce chapitre vous a appris les règles suivantes :

- **Les identificateurs PL/SQL :**
 - **sont définis dans la section déclarative**
 - **peuvent être de type scalaire, composite, référencé ou LOB**
 - **peuvent être basés sur la structure d'une autre variable ou d'un autre objet de base de données**
 - **peuvent être initialisés**
- **Les variables déclarées dans un environnement externe tel qu'*iSQL**Plus sont appelées variables hôte**
- **DBMS_OUTPUT.PUT_LINE permet d'afficher les données d'un bloc PL/SQL.**

Présentation de l'exercice 1

Dans cet exercice, vous allez :

- **déterminer la validité de quelques déclarations**
- **déclarer un bloc PL/SQL simple**
- **exécuter un bloc PL/SQL simple**