



PL SQL

Séance 3

Pr. M'barek ELHALOUI

Révision/ Questions

- Les enregistrements
- Les tables et méthodes associées
- Affectation des variables
- Questions ?

Corrigé de l'exercice

```
DECLARE
salMax E_EMP.salary%TYPE;

BEGIN
  select max(salary) INTO salMax
    from E_EMP ;
  dbms_output.put_line('Salaire Maximum: ' || salMax);
END;
```

Les structures de contrôle

- *If.... Then....Else...End If ;*

```
IF condition THEN  
    instructions;  
END IF;
```

```
IF condition THEN  
    instructions;  
ELSE instructions;  
END IF;
```

```
IF condition THEN  
    instructions;  
  
ELSEIF condition THEN  
    instructions;  
  
ELSE instructions;  
  
END IF;
```

- **ELSEIF** permet de finaliser les tests de conditions

Les structures de contrôle

- *If.... Then....else...End If : Exemples*

```
IF condition THEN  
    instructions;  
END IF;
```

```
DECLARE  
x integer := 5;  
y integer := 10 ;  
BEGIN  
    IF x < Y  
    THEN  
        dbms_output.put_line(X || 'est inférieur à ' || y );  
    END IF;  
END ;
```

Les structures de contrôle

- *If.... Then....else...End If : Exemples*

```
IF condition THEN  
    instructions;  
ELSE instructions;  
  
END IF;
```

```
DECLARE  
x  integer := 5;  
y  integer := 10 ;  
BEGIN  
    x:= x+10;  
    IF x < Y  
        THEN  
        dbms_output.put_line(X ||'est inférieur à '|| y );  
    ELSE  
        dbms_output.put_line(X ||'est supérieur ou egale à '|| y );  
    END IF;  
END ;
```

Les structures de contrôle

- *If.... Then....else...End If : Exemples*

IF *condition* **THEN**
instructions;

ELSEIF *condition* **THEN**
instructions;

ELSE *instructions;*

END IF;

```
DECLARE
x  integer := 5;
y  integer := 10 ;
BEGIN
    x:= x+5;
    IF x < Y
    THEN
        dbms_output.put_line(X ||'est inférieur à '|| y );
    ELSEIF x > Y
        dbms_output.put_line(X ||'est supérieur à '|| y );
    ELSE
        dbms_output.put_line(X ||'est égale à '|| y );
    END IF;
END ;
```

Les structures de contrôle

- Boucle **While Loop... end Loop ;**

```
While condition
Loop      instruction 1;
           instruction 2;

           instruction n;

END Loop ;
```

- **Exemple :**

```
DECLARE
i  integer := 0;
J  integer := 10 ;

BEGIN
    While i<=j Loop
        dbms_output.put_line(i);
        i:=i+1;
    End loop;
END ;
```

- Un label peut être utilisé avec Loop (*boucle1 Loop.....End Loop boucle1 ;*) qui sert à identifier la boucle et la quitter (**EXIT**) notamment dans les boucles imbriquées.

Les structures de contrôle

- Boucle **While Loop... end Loop ;**

```
While condition
Loop      instruction 1;
           instruction 2;

           instruction n;

END Loop ;
```

- **Exemple :**

```
DECLARE
i  integer := 0;
J  integer := 10 ;

BEGIN
    While i<=j Loop
        dbms_output.put_line(i);
        i:=i+1;
    End loop;
END ;
```

- Un label peut être utilisé avec Loop (*boucle1 Loop.....End Loop boocle1 ;*) qui sert à identifier la boucle et la quitter (EXIT) notamment dans les boucles imbriquées.

Les structures de contrôle

- Boucle **LOOP... END LOOP ;**

```
<label> LOOP  
    instruction1;  
    instruction2;  
  
    instruction2;  
EXIT [label][WHEN condition1]  
END LOOP <label>;
```

- **EXIT** : force la sortie de la boucle sans conditions.
- **EXIT WHEN** : permet une sortie de boucle si la condition est vraie.
- **EXIT <label> WHEN** : permet une sortie d'une boucle nommée label si la condition est vraie.
- **EXIT <label>** : force une sortie de la boucle nommée label.

- Exemple :

```
DECLARE  
i integer := 0; J integer := 1 ;  
BEGIN  
    Loop  
    i:=i+1;  
    Test1 Loop  
        dbms_output.put_line(i*J);  
        j:=i+1;  
        Exit Test1 When J> 5;  
  
        End loop Test1;  
    Exit when x=3  
end loop;  
END ;
```

Les structures de contrôle

- Boucle **FORIN...Loop END Loop;**

```
FOR compteur IN [REVERSE]  
borne_inf..borne_sup LOOP
```

instruction1 ;

instruction2 ;

Instruction n ;

```
[EXIT WHEN condition];  
END LOOP;
```

- **Exemples :**

```
DECLARE  
i integer;  
  
BEGIN  
    FOR i in 1 .. 10 Loop  
        dbms_output.put_line(i);  
    End loop;  
END ;
```

```
DECLARE  
i integer;  
  
BEGIN  
    FOR i in reverse 1 .. 10 Loop  
        dbms_output.put_line(i);  
    End loop;  
END ;
```

Les structures de contrôle

- Choix **CASE...WHEN...THEN...ELSE...END CASE ;**

CASE expression

WHEN expr 1 **THEN** instructions 1;

WHEN expr 2 **THEN** instructions 2;

...

WHEN expr n **THEN** instructions n;

ELSE instructions Y ;

END CASE;

- *Exemple :*

```
DECLARE
A  integer := 10 ;

BEGIN
A := A*10 ;
  CASE A
    WHEN 1 THEN dbms_output.put_line('Année');
    WHEN 10 THEN dbms_output.put_line('Décennie');
    WHEN 100 THEN dbms_output.put_line('Siècle');
    WHEN 1000 THEN dbms_output.put_line('Millénaire');
  ELSE dbms_output.put_line('Autres');

  END CASE;

END ;
```

Exercices

- Ecrire un programme PL SQL qui permet de réaliser une rotation des éléments d'un tableau vers la droite.
- Ecrire un programme PL SQL qui permet le classement d'un tableau de 10 chiffres entiers du plus petit au plus grand.