

# **SGBD - 2<sup>e</sup>**

## PL-SQL - Chapitre 3 - Les structures de contrôle

---

Anne Léonard

6 septembre 2023

Haute École de la Province de Liège

# Table des matières du chapitre i

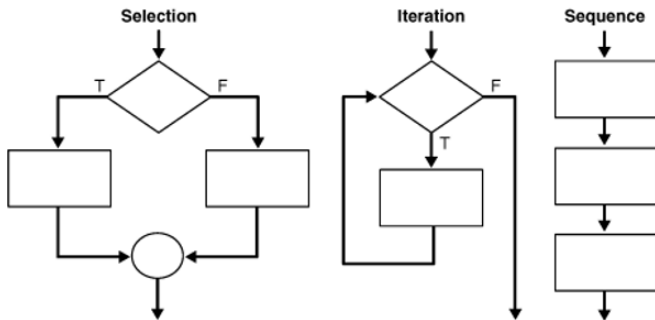
1. Introduction
2. Les structures conditionnelles
3. Les structures itératives

# Introduction

---

# Table des matières de la section : Introduction i

1. Introduction
2. Les structures conditionnelles
3. Les structures itératives



**Figure 1** – Structure de contrôle rappel

# **Les structures conditionnelles**

---

# Table des matières de la section : Les structures conditionnelles i

## 1. Introduction

## 2. Les structures conditionnelles

### 2.1 IF THEN

### 2.2 IF THEN ELSE

### 2.3 Case

## 3. Les structures itératives

# Les structures conditionnelles : IF THEN

- Les instructions comprises dans la branche **THEN** ne sont exécutées que si la condition est évaluée à **true**.
- Si la condition est évaluée à **false** ou **UNKNOWN**, le contrôle passe à l'instruction qui suit le **END IF**.



# Les structures conditionnelles : IF THEN

Une variable UNKNOWN

```
1  DECLARE
2    Vtest NUMBER;-- Vtest est NULL
3  BEGIN
4    IF Vtest >= 0
5      THEN DBMS_OUTPUT.PUT_LINE ('Vtest est positif');
6      ELSE DBMS_OUTPUT.PUT_LINE ('Vtest n''est pas positif
7        ');
8    END IF;
9    IF Vtest < 0
10     THEN DBMS_OUTPUT.PUT_LINE ('Vtest est négatif');
11     ELSE DBMS_OUTPUT.PUT_LINE ('Vtest n''est pas négatif
12       ');
13   END IF;
14 END;
15 -- Vtest N'est PAS positif
16 -- Vtest N'est PAS négatif
```

# Les structures conditionnelles : IF THEN

L'évaluation des conditions en PL/SQL suit le principe de l'évaluation rapide (short-circuit evaluation).

```
1 DECLARE
2     VContinue BOOLEAN := FALSE;
3
4 BEGIN
5     IF (NOT VContinue) OR ((41 / 0) > 0)
6     THEN
7         DBMS_OUTPUT.PUT_LINE('Pas d'erreur de division')
8         ;
9     END IF;
10 END;
--Pas d'erreur de division
```

# Les structures conditionnelles : IF THEN ELSE

Écriture douteuse

```
1 DECLARE
2   VTest BOOLEAN;          --Vtest est NULL
3
4 BEGIN
5   IF Vtest = TRUE
6   THEN
7     DBMS_OUTPUT.PUT_LINE('Vtest TRUE');
8   ELSE
9     DBMS_OUTPUT.PUT_LINE ('Vtest FALSE');
10  END IF;
11 END;
12 -- Vtest FALSE
13 -- Résultat compréhensible, mais non souhaitable.
```

# Les structures conditionnelles : IF THEN ELSE

Test valide

```
1 DECLARE
2   VTest BOOLEAN;          --Vtest est NULL
3
4 BEGIN
5   IF Vtest = TRUE
6   THEN
7     DBMS_OUTPUT.PUT_LINE('Vtest TRUE');
8   ELSIF Vtest = FALSE
9     THEN DBMS_OUTPUT.PUT_LINE ('Vtest FALSE');
10    ELSE DBMS_OUTPUT.PUT_LINE ('Vtest UNKNOWN');
11  END IF;
12 END;
13 -- Vtest UNKNOWN
```

# Les structures conditionnelles : Case

## Structure CASE - valeur

```
1  DECLARE
2      VTest BOOLEAN;
3  BEGIN
4      CASE VTest
5          WHEN TRUE THEN
6              DBMS_OUTPUT.PUT_LINE ('Vtest TRUE');
7          WHEN FALSE THEN
8              DBMS_OUTPUT.PUT_LINE ('Vtest FALSE');
9          ELSE DBMS_OUTPUT.PUT_LINE ('Vtest UNKNOWN');
10     END CASE;
11 END;
12 -- Vtest UNKNOWN
```

# Les structures conditionnelles : Case

## Structure CASE - expression

```
1  DECLARE
2    VGrade CHAR(3);
3  BEGIN
4    CASE
5      WHEN VGrade = 'PGD' -- ici une expression booléenne
6      THEN DBMS_OUTPUT.PUT_LINE ('Plus gd dis');
7      WHEN VGrade = 'GRD'
8      THEN DBMS_OUTPUT.PUT_LINE ('Gd dis');
9      WHEN VGrade = 'DIS'
10     THEN DBMS_OUTPUT.PUT_LINE ('Dis');
11     WHEN VGrade = 'SAT'
12     THEN DBMS_OUTPUT.PUT_LINE ('Satis');
13     ELSE DBMS_OUTPUT.PUT_LINE ('Echec');
14   END CASE;
15 END;
```

# **Les structures itératives**

---

# Table des matières de la section : Les structures itératives i

1. Introduction

2. Les structures conditionnelles

3. Les structures itératives

3.1 While

3.2 For



# Les structures itératives : While

Structure while

```
1  DECLARE
2      Vtest BOOLEAN := TRUE;
3  BEGIN
4      WHILE Vtest = TRUE
5          LOOP
6              Vtest := FALSE;
7          END LOOP;
8  END;
```

# Les structures itératives : For

- L'instruction **FOR** permet de définir une boucle.
- Le nombre d'itérations est défini entre 2 entiers.
- La séquence d'instructions est exécutée pour chaque entier compris dans l'intervalle défini.

## Important

Cet indice est défini implicitement dans la boucle !

# Les structures itératives : For

Structure for

```
1 BEGIN
2     FOR Viteration IN 1..3
3         LOOP
4             DBMS_OUTPUT.PUT_LINE
5                 ('Viteration = ' || Viteration);
6         END LOOP;
7 END;
8 -- Viteration = 1
9 -- Viteration = 2
10 -- Viteration = 3
```

## Important

La variable Viteration n'est pas déclarée!!

# Les structures itératives : For

Erreur d'affectation

```
1 BEGIN
2   FOR Viteration IN 1..3 LOOP
3     DBMS_OUTPUT.PUT_LINE
4       ('Viteration = ' || Viteration);
5     Viteration := Viteration + 1; -- erreur
6   END LOOP;
7 END;
8 -- ERREUR à la ligne 4 :
9 -- ORA-06550: Ligne 4, colonne 1:
10 -- PLS-00363: expression 'VITERATION' ne peut être
    utilisée comme cible d'affectation
```

# Les structures itératives : For

Portée

```
1 DECLARE
2   Viteration NUMBER ;
3 BEGIN
4   FOR Viteration IN 1..3 LOOP
5     DBMS_OUTPUT.PUT_LINE
6       ('Viteration = ' || Viteration);
7   END LOOP;
8   DBMS_OUTPUT.PUT_LINE (
9     'Valeur de Viteration après la boucle = '
10    || '<' || Viteration || '>');
11 END;
```

# Les structures itératives : For

- La variable **viteration** qui est explicitement déclarée est occultée dans la boucle par la variable compteur.
- On ne sait donc pas, dans la boucle, accéder à cette variable déclarée (sauf si on utilise un label)
- En dehors de la boucle, on ne peut accéder que à la variable Viteration déclarée, plus à la variable qui sert de compteur!!!

- Les présents diaporamas constituent un résumé du livre : DELMAL, P, *SQL2 - SQL3*, Bruxelles, De Boeck Université, 2004, 512 pages.
- Ce résumé a été étoffé d'éléments extraits de la documentation officielle d'Oracle : [Oracle Database Documentation](#), 12/02/2021.
- Ainsi que de l'ouvrage : SOUTOU, Ch, *SQL pour Oracle*, Paris, Editions Eyrolles, 2013, 642 pages.