

# Chapitre 4

## PL/SQL Blocs et curseurs

**Durée : 3 heures 55**

### Mots clés

DECLARE, BEGIN, END, LOOP, FOR, WHILE, IF, CASE, CURSOR, FORALL, EXCEPTION, SUBTYPE, DBMS\_OUTPUT

### Objectif

Ce chapitre est consacré à la mise en pratique des premiers concepts du langage PL/SQL, c'est-à-dire à bien comprendre la structure de bloc PL/SQL, le fonctionnement des curseurs, la déclaration de variable, les tableaux et les exceptions.

Une mise en pratique du package DBMS\_OUTPUT est également présentée. Ce package est particulièrement utile pour suivre la progression des traitements dans un bloc PL/SQL.

### Pré-requis

*Pour valider les pré-requis nécessaires, avant d'aborder le TP, répondez aux questions ci-dessous :*

1. Quels sont les quatre mots clés utilisés pour structurer un bloc PL/SQL ?

---

---

---

2. Quelles instructions sont possibles dans un bloc PL/SQL ?

- a. SELECT
- b. INSERT, UPDATE, DELETE
- c. COMMIT, ROLLBACK
- d. CREATE, ALTER, DROP

3. Quel est l'avantage de passer par un bloc PL/SQL plutôt que de choisir la solution purement SQL lors de l'exécution de plusieurs instructions du DML (INSERT, UPDATE et DELETE) ?
- a. L'exécution est plus rapide.
  - b. C'est la même chose.
  - c. Les exceptions peuvent être traitées.
  - d. Il n'y a alors pas besoin de gérer les transactions.
4. Quelles déclarations de variables ne sont pas valides ?
- a. `c varchar2(10);`
  - b. `12 int;`
  - c. `entier number(2);`
  - d. `tableau entier;`
5. Quelle déclaration permet de définir une variable structurée comme la table des clients ?
- a. `Client clients%type;`
  - b. `Client clients%rowtype;`
  - c. `Client clients;`
6. La déclaration suivante est-elle possible ? Justifiez votre réponse.  
`phrase varchar2(5000)`
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
7. Comment est-il possible de définir notre type entier comme étant un `number(10)` ?
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
8. Comment est-il possible de définir une variable structurée composée des champs `numero number(10)` et `nom varchar(40)` ?
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

9. Le langage PL/SQL est-il sensible à la casse ?

---



---



---

10. Quelle structure de boucle n'est pas valide ?

- a. `LOOP ... END LOOP;`
- b. `FOR i IN 1..10 LOOP ... END LOOP;`
- c. `WHILE (i<10) LOOP ... END LOOP;`
- d. `LOOP ... WHILE(i<10);`

11. Quel attribut du curseur est utilisé pour savoir si la dernière exécution de l'instruction `FETCH` a permis de ramener une ligne d'information ?

- a. `%FOUND`
- b. `%NOTFOUND`
- c. `%ISOPEN`
- d. `%ROWCOUNT`

12. Comment est-il possible de déclarer un curseur qui accepte des paramètres ?

- a. Ce n'est pas possible.
- b. Il faut faire précéder le nom de la variable par le caractère `&`.
- c. Il faut déclarer les paramètres avant le curseur en tant que `CURSOR PARAMETER`.
- d. Il faut déclarer le curseur de la façon suivante :  
`CURSOR leCurseur(param1 type,...) IS...`

13. Quelle plage de numéros est réservée aux erreurs définies par le programmeur ?

---



---



---

14. Quelle est l'instruction qui permet de lever une exception ?

- a. `raise`
- b. `raise_error`
- c. `raise_exception`
- d. `exception`

15. Comment est-il possible de reprendre le déroulement classique du bloc après qu'une exception a été levée et traitée ?

---

---

---

Corrigé p. 137

## Énoncé 4.1 Mise à jour conditionnelle

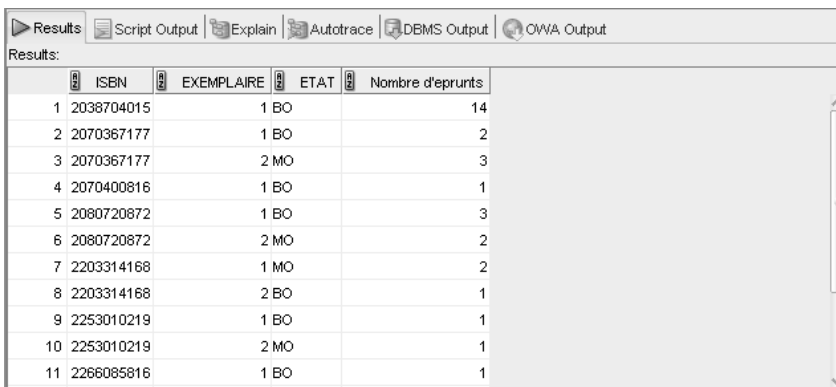
**Durée estimative** : 25 minutes

Tous les exemplaires ont été enregistrés avec l'état neuf. Cependant, au fur et à mesure des emprunts, leur état s'est dégradé. Il est donc nécessaire d'ajuster la valeur de cet état en prenant en considération le nombre de fois où l'exemplaire a été emprunté. C'est en effet le nombre d'emprunteurs qui a plus d'incidence sur l'état général de l'exemplaire que la durée effective des emprunts.

La mise à jour des informations est donc effectuée à l'aide du tableau suivant :

Nombre d'emprunts	État
Moins de 10	neuf
De 11 à 25	bon
De 26 à 40	moyen
Plus de 41	mauvais

Pour la réalisation de ce premier script, nous considérons que tous les exemplaires sont achetés à l'état neuf et que l'ensemble des emprunts de tous les exemplaires sont conservés en ligne. Exemple de l'état des exemplaires avant l'exécution du bloc PL/SQL :



	ISBN	EXEMPLAIRE	ETAT	Nombre d'emprunts
1	2038704015	1	BO	14
2	2070367177	1	BO	2
3	2070367177	2	MO	3
4	2070400816	1	BO	1
5	2080720872	1	BO	3
6	2080720872	2	MO	2
7	2203314168	1	MO	2
8	2203314168	2	BO	1
9	2253010219	1	BO	1
10	2253010219	2	MO	1
11	2266085816	1	BO	1

À la suite de l'exécution du bloc, la même requête permet d'obtenir le résultat suivant :

	ISBN	EXEMPLAIRE	ETAT	Nombre d'emprunts
1	2038704015	1	BO	14
2	2070367177	1	NE	2
3	2070367177	2	NE	3
4	2070400816	1	NE	1
5	2080720872	1	NE	3
6	2080720872	2	NE	2
7	2203314168	1	NE	2
8	2203314168	2	NE	1
9	2253010219	1	NE	1
10	2253010219	2	NE	1
11	2266085816	1	NE	1

Corrigé p. 139

## Énoncé 4.2 Suppression conditionnelle

**Durée estimative** : 30 minutes

Écrivez un bloc PL/SQL qui permet de supprimer les membres dont l'adhésion a expiré depuis plus de 2 ans.

Si des fiches d'emprunts existent et si tous les exemplaires empruntés ont été rendus, alors il faut mettre à **null** la valeur présente dans la colonne **membre**.

S'il reste des livres empruntés et non rendus, alors il ne faut pas supprimer le membre.

Par exemple, la liste des membres dont l'adhésion a expiré depuis plus de 2 ans donne le résultat suivant :

NUMERO	NOM	PRENOM	ADRESSE	ADHESION	DUREE	MOBILE
1	21 LOMOBO	Laurent	31 rue des Lilas	02/03/06	1 (null)	

Après l'exécution du bloc PL/SQL, la même requête ne doit ramener aucune ligne.  
Pour les besoins de l'exercice, ajoutez un membre non actif depuis plus de 2 ans par l'intermédiaire de l'instruction suivante :

```
INSERT INTO membres (numero, nom, prenom, adresse, adhesion, duree)  
VALUES (seq_membre.NEXTVAL, 'LOMOBO', 'Laurent', '31 rue des  
lilas', sysdate-1000, 1);
```

## Indice

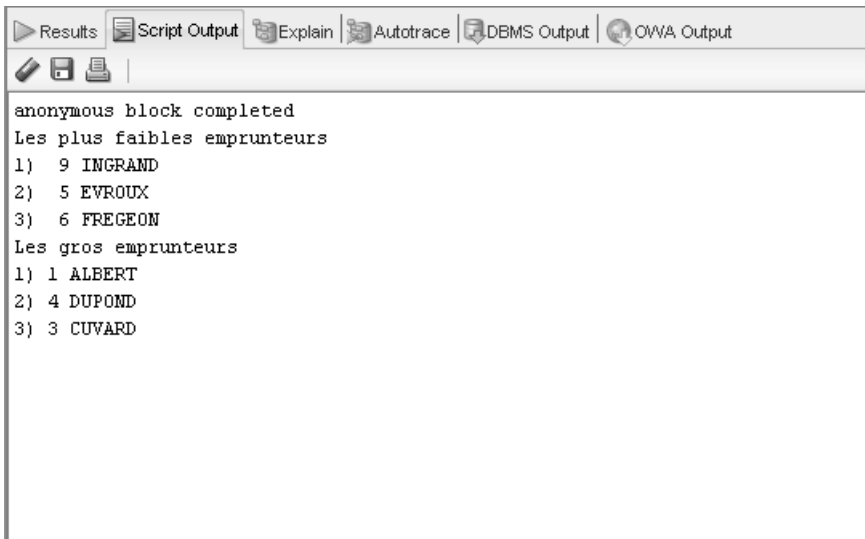
*Apportez les modifications de structure nécessaires avant de réaliser le script.*

Corrigé p.141

## Énoncé 4.3 Affichage d'informations pendant l'exécution d'un bloc

**Durée estimative :** 30 minutes

Écrivez un bloc PL/SQL qui permet d'afficher la liste des trois membres qui ont emprunté le plus d'ouvrages au cours des dix derniers mois et établissez également la liste des trois membres qui en ont emprunté le moins.



```
anonymous block completed  
Les plus faibles emprunteurs  
1) 9 INGRAND  
2) 5 EVROUX  
3) 6 FREGEON  
Les gros emprunteurs  
1) 1 ALBERT  
2) 4 DUPOND  
3) 3 CUVARD
```

Pour ce type de calcul, l'exemplaire d'un ouvrage est considéré comme emprunté par l'un des membres à partir du moment où il est inscrit sur la fiche. C'est donc la date figurant dans la table des emprunts qui fait référence pour l'ensemble des calculs.