

Les curseurs

Définition

- Lorsque l'on exécute un ordre SQL à partir de PL/SQL, Oracle alloue une zone de travail privée pour cet ordre.
- Cette zone de travail contient des informations relatives à l'ordre SQL ainsi que les statuts d'exécution de l'ordre.
- Les curseurs PL/SQL sont un mécanisme permettant de nommer cette zone de travail et de manipuler les données qu'elle contient.
- **Un curseur PL/SQL permet de récupérer et de traiter les données de la base dans un programme PL/SQL, ligne par ligne.**
- Il existe deux sortes de curseurs :
 - Les curseurs implicites.
 - Les curseurs explicites.
- Le langage PL/SQL crée de manière implicite un curseur pour chaque ordre SQL,

Curseur explicite

- Pour traiter une requête qui retourne plusieurs lignes, l'utilisateur doit définir un curseur qui lui permet d'extraire la totalité des lignes sélectionnées.
- L'utilisation d'un curseur pour traiter un ordre Select ramenant plusieurs lignes, nécessite 4 étapes :
 1. Déclaration du curseur
 2. Ouverture du curseur
 3. Traitement des lignes
 4. Fermeture du curseur.

Déclaration d'un Curseur explicite

- La déclaration du curseur permet de stocker l'ordre Select dans le curseur.

- Le curseur se définit dans la partie DECLARE d'un bloc PL/SQL :

**Cursor nomcurseur [(nomparam type [,nomparam type, ...])]
IS Requête SELECT ;**

- Exemple :

DECLARE

Cursor DEPT10 is

select nomE, salaire from employe where Numdep=10;

Ouverture et fermeture du Curseur

- L'étape d'ouverture permet d'effectuer :
 - l'allocation mémoire du curseur ;
 - l'analyse sémantique et syntaxique de l'ordre (parsing) ;
 - le positionnement de verrous éventuels (si select for update...)
- L'étape de fermeture permet d'effectuer : la libération de la place mémoire.
- **La syntaxe :**
 - Dans la partie traitement du bloc PL/SQL
 - Avoir préalablement déclaré le curseur pour l'ouvrir
 - Avoir préalablement ouvert le curseur pour le fermer

```
OPEN nomcurseur [(nomparam type [,nomparam type, ...])  
/* traitement des lignes */  
CLOSE nomcurseur
```

Ouverture et fermeture du Curseur

- **Exemple :**

DECLARE

Cursor **DEPT10** is

select nomE, salaire from employe where Numdep=10;

Begin

...

OPEN DEPT10

/* traitement des lignes */

CLOSE DEPT10

...

Traitement des lignes

- Il faut traiter les lignes une par une et renseigner les variables réceptrices définies dans la partie **Declare** du bloc.
- **Syntaxe :**
- Dans la partie traitement du bloc PL/Sql, avoir préalablement ouvert le curseur puis
**FETCH nomcurseur INTO { nomvariable [,nomvariable] ... |
nomrecord }**
- L'ordre **FETCH** ne ramène qu'une seule ligne à la fois.
- De ce fait il faut recommencer l'ordre pour traiter la ligne suivante.

Traitement des lignes

- **Exemple :**

DECLARE

Cursor **DEPT10** is

select nomE, salaire from employe where Numdep=10;

vnom varchar2(30);

vsal number;

Begin

OPEN DEPT10

Loop

FETCH DEPT10 into vnom, vsal;

/* traitement **d'une** ligne */

End Loop;

CLOSE DEPT10

Les attributs d'un curseur

- Les attributs d'un curseur nous fournissent des informations quant à l'exécution de l'ordre. Elles sont conservées par PL/Sql après l'exécution du curseur (explicite ou implicite).
- Ces attributs permettent de tester directement le résultat de l'exécution.

Curseur implicite	Curseur explicite :
Sql%Found	Nomcurseur%Found
Sql%Notfound	Nomcurseur%Notfound
Sql%Isopen	Nomcurseur%Isopen
Sql%Rowcount	Nomcurseur%Rowcount
	Nomcurseur%Rowtype

L'attribut %Found

- Cet attribut est de type booléen : soit vrai, soit faux.
- Le curseur implicite est vrai si les instructions insert, update, delete traitent au moins une ligne.
- Le curseur explicite est vrai si le Fetch ramène au moins une ligne.
- **Exemple :**

L'attribut %Found

```
DECLARE
Cursor DEPT10 is
    select nomE, salaire from employe where Numdep=10;
vnom varchar2(30);
vsal number;
Begin
    OPEN DEPT10
    Loop
        FETCH DEPT10 into vnom, vsal;
        if DEPT10%Found then
            if vsal > 1500 then
                insert into Cadre values (vnom,vsal);
            end if;
        else
            exit;
        end if;
    End Loop;
    CLOSE DEPT10
```

L'attribut %NotFound

- Cet attribut est de type booléen : soit vrai, soit faux.
- Le curseur implicite est vrai si les instructions insert, update, delete ne traitent aucune ligne.
- Le curseur explicite est vrai si le Fetch ne ramène plus de ligne.
- **Exemple :**

L'attribut %NotFound

DECLARE

Cursor **DEPT10** is

select nomE, salaire from employe where Numdep=10;

vnom varchar2(30);

vsal number;

Begin

OPEN DEPT10

Loop

FETCH DEPT10 into vnom, vsal;

Exit When DEPT10%NotFound;

if vsal > 1500 then

insert into Cadre values (vnom,vsal);

end if;

End Loop;

CLOSE DEPT10

L'attribut %Isopen

- Cet attribut est de type booléen : soit vrai, soit faux.
- Le curseur implicite est toujours faux car Oracle referme toujours les curseurs qu'il ouvre après chaque utilisation.
- Le curseur explicite est vrai si le curseur est ouvert.
- Exemple

```
BEGIN
```

```
    If not (Dept10%Isopen) then
```

```
        open Dept10 ;
```

```
    End if;
```

L'attribut %RowCount

- Cet attribut est de type numérique.
- Le curseur implicite indique le nombre de lignes traités par les ordres insert, update, delete.
- Le curseur explicite est :
 - incrémenté à chaque ordre fetch,
 - il traduit donc la nième ligne traitée.
- Exemple :

L'attribut %RowCount

DECLARE

Cursor **C1** is

select nomE, salaire from employe where Numdep=10;

vnom varchar2(30);

vsal number;

Begin

OPEN DEPT10

Loop

FETCH C1 into vnom, vsal;

Exit When C1%NotFound OR C1%RowCount>25;

if vsal > 1500 then

insert into Cadre values (vnom,vsal);

end if;

End Loop;

CLOSE DEPT10

L'attribut %Rowtype

- Cet attribut permet la déclaration implicite d'une structure dont les éléments sont d'un type identique aux colonnes ramenées par le curseur.

- **Syntaxe :**

- Dans la partie déclarative du bloc.

```
Cursor nomcurseur is ordre_select;  
nomrecord nomcurseur%Rowtype;
```

- Les éléments de la structure sont identifiés par :

```
nomrecord.nomcolonne
```

- La structure est renseignée par le Fetch :

```
Fetch nomcurseur into nomrecord;
```

- **Exemple**

L'attribut %Rowtype

DECLARE

Cursor **DEPT10** is

select nomE, salaire from employe where Numdep=10;

Empl DEPT10%Rowtype;

Begin

OPEN DEPT10

Loop

FETCH DEPT10 into Empl;

Exit When DEPT10%**NotFound;**

if Empl.salaire > 1500 then

insert into Cadre values (**Empl.nomE , Empl.salaire**);

end if;

End Loop;

CLOSE DEPT10

Les boucles et les curseurs

- L'objectif est de fournir au programmeur une structure simple et efficace pour utiliser les structures de boucle et les curseurs.

```
DECLARE
    Cursor nomcurseur is ordre_select;
BEGIN
    For nomrecord in nomcurseur Loop
        /* traitement */
    End Loop;
END;
```



```
DECLARE
    Cursor nomcurseur is ordre_select;
    nomrecord nomcurseur%rowtype;
BEGIN
    Open nomcurseur;
    Loop
        Fetch nomcurseur into nomrecord;
        Exit when nomcurseur%notfound
        /* traitement */
    End Loop;
    Close nomcurseur;
END;
```

Le curseur paramétré

- Il permet d'utiliser des variables dans le curseur. Principalement dans la clause where.
- Il faut pour cela spécifier les noms et les types des paramètres dans la déclaration du curseur.

**Cursor nomcurseur (param1 type, param2 type,...)
is select ordre_select ;**

- L'ordre select utilise les paramètres.
- Les types possibles sont : char, number, date, boolean sans spécifier la longueur.

Begin

Open nomcurseur (valeur1 , valeur2,) ;

- Exemple :

Le curseur paramétré

DECLARE

Cursor **C1(dep Number)** is

select nomE, salaire from employe where Numdep=**dep**;

Begin

For Empl in **C1(10)** loop

if Empl.salaire > 1500 then

insert into Cadre values (**Empl.nomE , Empl.salaire**);

end if;

End Loop;

End;

La clause "current of ..."

- Cette clause permet d'accéder directement à la ligne ramenée par l'ordre Fetch afin de la traiter (update, delete).
- Il faut se réserver la ligne lors de la déclaration du curseur par le positionnement d'un verrou d'intention :
(... **For update of nom_colonne**)
- Il faut spécifier que l'on veut traiter la ligne courante au Fetch par la clause
(... **Where current of nom_curseur**).
- Exemple :

La clause "current of ..."

DECLARE

Cursor **C1** is select nomE, salaire from employe **for update of salaire;**

Begin

For Empl in **C1 loop**

if Empl.salaire > 1500 then

insert into Cadre values (**Empl.nomE , Empl.salaire**);

update employe set salaire=salaire*1.3 where **current of C1;**

end if;

End Loop;

End;