



ORAE505

Conception et administration de B.D.

5. Le langage PL/SQL

b. Les curseurs

Carina Roels

5.b.1 Qu'est-ce un curseur?

Un curseur est une zone de mémoire utilisée par le noyau ORACLE pour analyser / interpréter des instructions SQL.

Le curseur IMPLICITE

Généré et géré par le noyau ORACLE pour chaque instruction SQL (SELECT, INSERT, UPDATE, DELETE).

Le curseur EXPLICITE

Généré et géré par l'utilisateur pour traiter un ordre SELECT qui ramène plusieurs lignes.

5.b.2 La gestion d'un curseur explicite

4 étapes :

DECLARATION

Dans la section
DECLARE

OUVERTURE

Dans la section
BEGIN

TRAITEMENT

FERMETURE

5.b.2 La gestion d'un curseur explicite

a) Déclaration

Déclarer une instruction SELECT en lui donnant un nom.

SYNTAXE

```
CURSOR nom_curseur IS ordre_select;
```

EXEMPLE

DECLARE

```
    CURSOR empX200 IS
        SELECT      codemp, nomemp, preemp,
        salemp
        FROM employes
        WHERE        coddep = 'X200'
        ORDER BY    salemp;
```

Carina Roels

4

Exemple :

DECLARE

```
CURSOR    empX200 IS
SELECT    codemp, nomemp, preemp, salemp
FROM      employes
WHERE     coddep = 'X200'
ORDER BY  salemp;
```

BEGIN

...

END;

/

5.b.2 La gestion d'un curseur explicite

b) Ouverture

SYNTAXE

OPEN *nom_curseur*;

EXEMPLE

BEGIN

.....;

OPEN empX200;

.....;

L'ouverture du curseur génère :

- l'allocation de la mémoire nécessaire
- l'analyse syntaxique et sémantique de l'ordre **SELECT**.
- le positionnement de verrous (si **SELECT** **FOR UPDATE**)

Carina Roels

5

Exemple :

DECLARE

```
CURSOR    empX200 IS
SELECT    codemp, nomemp, preemp, salemp
FROM      employes
WHERE     coddep = 'X200'
ORDER BY  salemp;
```

BEGIN

OPEN empX200;

...

END;

/

5.b.2 La gestion d'un curseur explicite

c) Traitement

SYNTAXE

FETCH *nom_curseur* INTO *var1, var2, var3, ...* ;

EXEMPLE

LOOP

FETCH empX200 INTO code, nom, prenom, salaire ;

...

EXIT WHEN
END LOOP;

Les variables doivent être déclarées
dans la section DECLARE.

Carina Roels

6

Exemple :

```
DECLARE
  code      employes.codemp%TYPE;
  nom       employes.nomemp%TYPE;
  prenom    employes.preemp%TYPE;
  salaire   employes.salemp%TYPE;

CURSOR empX200 IS
SELECT   codemp, nomemp, preemp, salemp
FROM     employes
WHERE    coddep = 'X200'
ORDER BY salemp;
BEGIN
  OPEN empX200;
  LOOP
    FETCH empX200 INTO code, nom, prenom, salaire ;

    IF salaire < 6700
    THEN   INSERT INTO RESULTAT( chaine, num )
          VALUES ( nom || prenom, salaire );
    END IF;
    EXIT WHEN salaire > 6700;
  END LOOP;
END;
/
```

5.b.2 La gestion d'un curseur explicite

d) Fermeture

SYNTAXE

CLOSE *nom_curseur*;

EXEMPLE

BEGIN

.....;

CLOSE empX200;

.....;

Libère la mémoire qui était utilisée par le curseur, ainsi que les verrous qui ont été positionnés par l'instruction **SELECT FOR UPDATE**

Carina Roels

7

Exemple :

```

DECLARE
code          employes.codemp%TYPE;
nom           employes.nomemp%TYPE;
prenom       employes.preemp%TYPE;
salaire       employes.salemp%TYPE;

CURSOR empX200 IS
SELECT
FROM
WHERE
ORDER BY
FROM
WHERE
ORDER BY
salemp;
BEGIN
OPEN          empX200;
LOOP
    FETCH      empX200 INTO code, nom, prenom, salaire ;

    IF salaire < 6700
    THEN
        INSERT INTO RESULTAT( chaine, num )
        VALUES ( nom || prenom , salaire );
    END IF;
    EXIT WHEN salaire > 6700;
END LOOP;
CLOSE empX200;
END;
/

```

5.b.3 Les attributs de curseur

Un attribut d'un curseur est un indicateur sur l'état du curseur.

Les attributs existent pour les CURSEURS IMPLICITES et EXPLICITES.

%FOUND

%NOTFOUND

%ISOPEN

%ROWCOUNT

5.b.3 Les attributs de curseur

a) %FOUND

Curseur IMPLICITE :
SQL%FOUND

Valeur = TRUE pour :
INSERT, UPDATE, DELETE,
SELECT
qui traite au moins 1 ligne

Curseur EXPLICITE :
***nomcurseur*%FOUND**

Valeur = TRUE pour :
Dernier FETCH exécuté
qui a ramené 1 ligne

Carina Roels

9

```

DECLARE
CURSOR      EMPX200      IS
SELECT      codemp, nomemp, preemp, salemp
FROM        EMPLOYES
WHERE       coddep = 'X200'
ORDER BY    SALEMP ;

Vcode       employes.codemp%TYPE ;
Vnom        employes.nomemp%TYPE ;
Vprenom     employes.preemp%TYPE ;
Vsalair     employes.salemp%TYPE ;

BEGIN
OPEN        empX200 ;
FETCH      empX200      INTO Vcode, vnom, vprenom, vsalair ;

WHILE      empX200%FOUND
LOOP
  If        vsalair < 6700 then
    INSERT INTO RESULTATN VALUES( Vnom, Vprenom, Vsalaire ) ;
  End if ;

  FETCH      empX200      INTO Vcode, vnom, vprenom, vsalair ;
END LOOP ;
CLOSE      empX200 ;
END ;
/

select * from resultatn ;

```

5.b.3 Les attributs de curseur

b) %NOTFOUND

Curseur IMPLICITE :
SQL%NOTFOUND

Curseur EXPLICITE :
nomcurseur%NOTFOUND

Valeur = TRUE pour :
INSERT, UPDATE, DELETE,
SELECT
qui traite n'a pas traité de ligne

Valeur = TRUE pour :
Dernier FETCH exécuté
qui n'a pas ramené de ligne

Carina Roels

10

```

DECLARE
CURSOR EMPX200 IS
SELECT      codemp, nomemp, preemp, salemp
FROM        EMPLOYES
WHERE       coddep = 'X200'
ORDER BY    SALEMP ;

Vcode      employes.codemp%TYPE ;
Vnom        employes.nomemp%TYPE ;
Vprenom     employes.preemp%TYPE ;
Vsalair     employes.salemp%TYPE ;

BEGIN
OPEN        empX200 ;
LOOP
    FETCH    empX200 INTO Vcode, vnom, vprenom, vsalair ;
    EXIT WHEN empX200%NOTFOUND ;

    If       vsalair < 6700 then
        INSERT INTO RESULTATN VALUES( Vnom, Vprenom, Vsalair ) ;
    End if ;

END LOOP ;
CLOSE      empX200 ;
END ;
/
select * from resultatn ;

```

5.b.3 Les attributs de curseur

c) %ISOPEN

Curseur IMPLICITE :
SQL%ISOPEN

Valeur = toujours FALSE

ORACLE ferme le curseur après utilisation.

Curseur EXPLICITE :
nomcurseur%ISOPEN

Valeur = TRUE

si le curseur est ouvert.

Carina Roels

11

```

DECLARE
CURSOR      EMPX200      IS
SELECT      codemp, nomemp, preemp, salemp
FROM        EMPLOYES
WHERE       coddep = 'X200'
ORDER BY    SALEMP ;

Vcode       employees.codemp%TYPE ;
Vnom        employees.nomemp%TYPE ;
Vprenom     employees.preemp%TYPE ;
Vsalair     employees.salemp%TYPE ;

BEGIN
  If NOT( empX200%ISOPEN ) then
    OPEN      empX200 ;
  End if ;

  LOOP
    FETCH      empX200      INTO  Vcode, vnom, vprenom, vsalair ;
    EXIT WHEN  empX200%NOTFOUND ;

    If  vsalair < 6700 then
      INSERT INTO RESULTATN  VALUES( Vnom, Vprenom, Vsalair ) ;
    End if ;

  END LOOP ;
  CLOSE      empX200 ;
END ;
/
select * from resultatn ;

```

5.b.3 Les attributs de curseur

d) %ROWCOUNT

Curseur IMPLICITE :
SQL%ROWCOUNT

Pour INSERT, UPDATE,
DELETE, SELECT
= nombre de lignes traitées

Curseur EXPLICITE :
nomcurseur%ROWCOUNT

Pour le dernier FETCH exécuté
= le numéro de la ligne ramenée

Carina Roels

12

```

DECLARE
CURSOR      EMPX200      IS
SELECT      codemp, nomemp, preemp, salemp
FROM        EMPLOYES
WHERE       coddep = 'X200'
ORDER BY    SALEMP ;

Vcode       employees.codemp%TYPE ;
Vnom        employees.nomemp%TYPE ;
Vprenom     employees.preemp%TYPE ;
Vsalair     employees.salemp%TYPE ;

BEGIN
  If NOT( empX200%ISOPEN ) then
    OPEN empX200 ;
  End if ;

  LOOP
    FETCH      empX200      INTO Vcode, vnom, vprenom, vsalair ;
    EXIT WHEN  empX200%NOTFOUND OR empX200%ROWCOUNT > 15 ;

    If vsalair < 6700 then
      INSERT INTO RESULTATN VALUES( Vnom, Vprenom, Vsalaire ) ;
    End if ;
  END LOOP ;

  CLOSE      empX200 ;
END ;
/

select * from resultatn ;

```


5.b.4 Les simplifications d'écriture

a) Traitement de curseur (1)

```

DECLARE
  CURSOR nom_curseur IS ordre_select;
  Nom_rec nom_curseur%ROWTYPE;
BEGIN
  OPEN nom_curseur;
  LOOP
    FETCH nom_curseur INTO nom_rec;
    EXIT WHEN
      nom_curseur%NOTFOUND;
    .....
  END LOOP;
  CLOSE nom_curseur;
END;

```



```

DECLARE
  CURSOR nom_curseur IS
    ordre_select;
BEGIN
  FOR nom_rec IN nom_curseur
  LOOP
    .....
  END LOOP;
END;

```

Carina Roels

13

Exemple :

```

DECLARE
  CURSOR      empX200      IS
  SELECT      codemp, nomemp, preemp, salemp
  FROM        EMPLOYEES
  WHERE      coddep = 'X200'
  ORDER      BY salemp;

BEGIN
  FOR      emp      IN      empX200
  LOOP
    If emp.salemp < 6700 then
      INSERT INTO RESULTATN
      VALUES ( emp.nomemp || emp.preemp, emp.salemp );
    End if;

  END LOOP;
END;
/


select * from resultatn ;

```

5.b.4 Les simplifications d'écriture

b) Traitement de curseur (2)

```
DECLARE
  CURSOR nom_curseur IS ordre_select ;
BEGIN
  FOR nom_rec IN nom_curseur
  LOOP
    .....
  END LOOP ;
END ;
```



```
DECLARE
BEGIN
  FOR nom_rec IN ( ordre_select )
  LOOP
    .....
  END LOOP ;
END ;
```

Carina Roels

14

Exemple :

DECLARE

BEGIN

FOR emp IN

(SELECT codemp, nomemp, preemp, salemp

FROM EMPLOYES

WHERE coddep = 'X200'

ORDER BY salemp)

LOOP

If emp.salemp < 6700 then

INSERT INTO RESULTATN

VALUES (emp.nomemp || emp.preemp, emp.salemp);

End if ;

END LOOP ;

END ;

/

select * from resultatn ;

5.b.5 Les curseurs avec paramètres

Utilisation d'un même curseur avec des valeurs différentes dans la clause WHERE du SELECT.

```
DECLARE
  CURSOR nom_curseur( par1 type, par2 type, ... )
  IS ordre_select ;
BEGIN
  OPEN      nom_curseur( val1, val2, ... ) ;

  BEGIN
    FOR nom_struct IN  nom_curseur( val1, val2, ... ) ;
```

Carina Roels

15

Dans la déclaration de curseur, l'ordre SELECT doit utiliser les paramètres nommés.

Les types de paramètres sont :

CHAR, VARCHAR, DATE, NUMBER, BOOLEAN sans spécification de longueur.

Le passage de paramètres se fait à l'ouverture du curseur.

5.b.5 Les curseurs avec paramètres

Exemple : dans les commentaires

Carina Roels

16

Exemple : Trouver les N plus gros salaires différents. Afficher ensuite des informations concernant les employés qui ont ces salaires.

Prompt Nombre de plus gros salaires à chercher ?
Accept nombre

```
DECLARE
  CURSOR          grossal                               IS
  SELECT          DISTINCT salemp FROM EMPLOYES
  ORDER BY salemp DESC ;

  CURSOR          C1( psal NUMBER ) IS
  SELECT          nomemp, preemp, salemp
  FROM            EMPLOYES
  WHERE           salemp = psal ;

BEGIN
  For vgrossal IN grossal
  LOOP
    EXIT WHEN grossal%ROWCOUNT > &nombre ;
    FOR vemploye In C1( vgrossal.salemp )
    LOOP
      INSERT INTO RESULTATN VALUES
        ( vemploye.nomemp || vemploye.preemp, vemploye.salemp ) ;
    END LOOP ;
  END LOOP ;
END ;
/
select chaine EMPLOYE, num SALAIRE from resultatn ;
rollback ;
```


5.b.6 La mise à jour par curseur

La clause CURRENT OF
Permet de modifier ou de supprimer directement la ligne
ramenée par un FETCH.

Lors de la déclaration du curseur,
il faut réserver les lignes par :

FOR UPDATE

FOR UPDATE OF nom_col1, nom_col2, ...

Carina Roels

17

Exemple :

```
DECLARE
  CURSOR C1 IS
  SELECT nomemp, salemp FROM EMPLOYES
  FOR UPDATE OF salemp ;

BEGIN
  FOR rec_empl In C1
  LOOP
    If rec_empl.salemp < 7000 then
      INSERT INTO RESULTATN VALUES
        ( rec_empl.nomemp || rec_empl.preemp , rec_empl.salemp ) ;
      UPDATE EMPLOYES SET salemp = salemp * 1.2
      WHERE CURRENT OF C1 ;
    End if ;
  END LOOP ;
END ;
/

select * from resultatn ;
```