ORAE505

Conception et administration de B.D.

5. Le langage PL/SQLb. 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

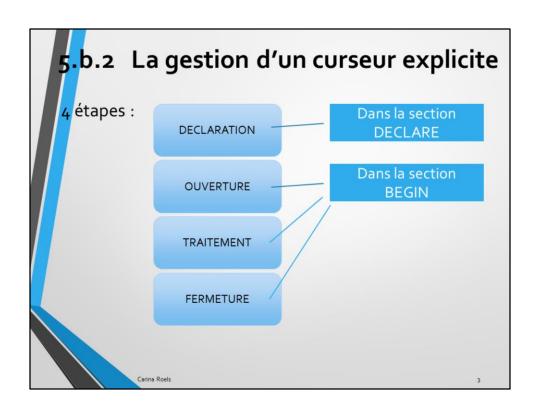
Généré et géré par le noyau ORACLE pour chaque instruction SQL (SELECT, INSERT, UPDATE, DELETE).

Le curseur EXPLICITE

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

Carina Roels

2

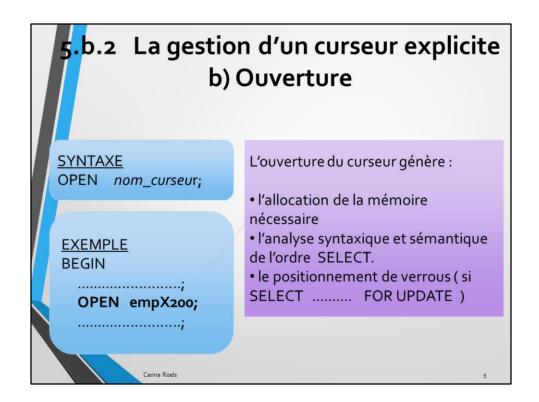


```
5.b.2 La gestion d'un curseur explicite
                 a) Déclaration
Déclarer une instruction SELECT en lui donnant un nom.
SYNTAXE
CURSOR nom_curseur IS ordre_select;
 EXEMPLE
 DECLARE
   CURSOR empX200 IS
                  codemp, nomemp, preemp,
       SELECT
 salemp
      FROM employes
                  coddep = 'X200'
       WHERE
       ORDER BY salemp;
           Carina Roels
```

DECLARE

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

BEGIN
...
END;
/
```



DECLARE

```
CURSOR empX200 IS

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

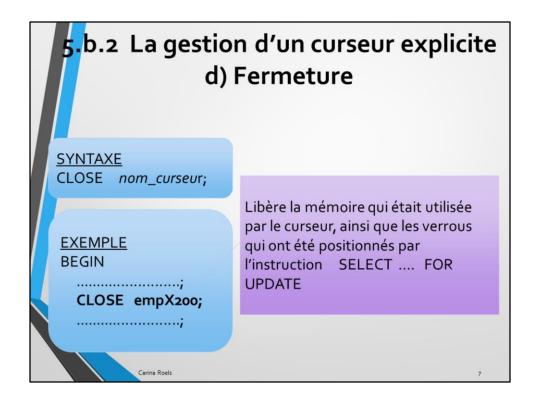
BEGIN
OPEN empX200;
...
END;
/
```

```
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.
```

```
DECLARE
 code
             employes.codemp%TYPE;
             employes.nomemp%TYPE;
 nom
             employes.preemp%TYPE;
 prenom
             employes.salemp%TYPE;
 salaire
 CURSOR
             empX200 IS
 SELECT
             codemp, nomemp, preemp, salemp
             employes
 FROM
             coddep = 'X200'
 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;
END;
```



```
DECLARE
 code
             employes.codemp%TYPE;
             employes.nomemp%TYPE;
 nom
 prenom
             employes.preemp%TYPE;
             employes.salemp%TYPE;
 salaire
 CURSOR
             empX200 IS
             codemp, nomemp, preemp, salemp
 SELECT
 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;
 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 : Curseur EXPLICITE : nomcurseur%FOUND Valeur = TRUE pour : NSERT, UPDATE, DELETE, SELECT qui traite au moins 1 ligne Curseur EXPLICITE : nomcurseur%FOUND

```
DECLARE
             EMPX200
 CURSOR
                           IS
             codemp, nomemp, preemp, salemp
 SELECT
 FROM
                           EMPLOYES
 WHERE
             coddep = 'X200'
 ORDER
             BY SALEMP;
             employes.codemp%TYPE;
 Vcode
 Vnom
             employes.nomemp%TYPE;
 Vprenom
             employes.preemp%TYPE;
             employes.salemp%TYPE;
 Vsalair
BEGIN
 OPEN
             empX200;
 FETCH
             empX200
                           INTO Vcode, vnom, vprenom, vsalair;
             empX200%FOUND
 WHILE
 LOOP
    lf
             vsalair < 6700 then
             INSERT INTO RESULTATN VALUES (Vnom, Vprenom, Vsalair);
    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 : Curseur EXPLICITE : nomcurseur%NOTFOUND Valeur = TRUE pour : Dernier FETCH exécuté qui n'a pas ramené de ligne Valeur = TRUE pour : Dernier FETCH exécuté qui n'a pas ramené de ligne

```
DECLARE
 CURSOR
             EMPX200
                          IS
             codemp, nomemp, preemp, salemp
 SELECT
 FROM
             EMPLOYES
 WHERE
             coddep = 'X200'
 ORDER
             BY SALEMP;
 Vcode
             employes.codemp%TYPE;
 Vnom
             employes.nomemp%TYPE;
 Vprenom
             employes.preemp%TYPE;
             employes.salemp%TYPE;
 Vsalair
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;
```

Carina Roels

```
5.b.3 Les attributs de curseur
c) %ISOPEN

Curseur IMPLICITE : Curseur EXPLICITE : nomcurseur%ISOPEN

Valeur = toujours FALSE Valeur = TRUE

ORACLE ferme le curseur après utilisation.

Si le curseur est ouvert.
```

```
DECLARE
 CURSOR
              EMPX200
 SELECT
              codemp, nomemp, preemp, salemp
 FROM
                             EMPLOYES
              coddep = 'X200'
 WHERE
 ORDER
              BY SALEMP;
              employes.codemp%TYPE;
 Vcode
              employes.nomemp%TYPE;
 Vnom
 Vprenom
              employes.preemp%TYPE;
 Vsalair
              employes.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 : Curseur EXPLICITE : nomcurseur%ROWCOUNT Pour INSERT, UPDATE, DELETE, SELECT = nombre de lignes traitées = le numéro de la ligne ramenée

```
DECLARE
 CURSOR
               EMPX200
               codemp, nomemp, preemp, salemp
 SELECT
 FROM
                              EMPLOYES
               coddep = 'X200'
 WHERE
               BY SALEMP;
 ORDER
 Vcode
               employes.codemp%TYPE;
               employes.nomemp%TYPE;
 Vnom
 Vprenom
               employes.preemp%TYPE;
 Vsalair
               employes.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, Vsalair);
   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;
                            CURSOR nom_curseur IS
                           ordre select;
                           BEGIN
 END LOOP;
                            FOR nom_rec IN nom_curseur
 CLOSE nom_curseur;
                            LOOP
END;
                            END LOOP;
                           END;
         Carina Roels
```

```
DECLARE
 CURSOR
            empX200
                        IS
 SELECT
            codemp, nomemp, preemp, salemp
 FROM
                        EMPLOYES
 WHERE
            coddep = 'X200'
 ORDER
            BY salemp;
BEGIN
 FOR
                        IN
                                    empX200
            emp
 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;
                      DECLARE
END;
                      BEGIN
                        FOR nom_rec IN ( ordre_select )
                        LOOP
                        END LOOP;
                       END;
          Carina Roels
```

```
DECLARE
BEGIN
 FOR emp IN
 ( SELECT codemp, nomemp, preemp, salemp
  FROM
           EMPLOYES
   WHERE coddep = 'X200'
           BY salemp )
   ORDER
 LOOP
           If emp.salemp < 6700 then
              INSERT INTO RESULTATN
              VALUES (emp.nomemp || emp.preemp, emp.salemp );
            End if;
 END LOOP;
END;
select * from resultatn;
```

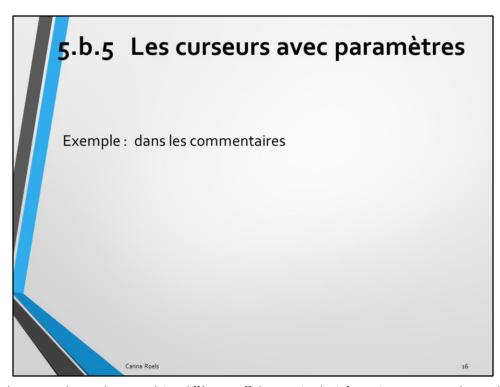
Utilisation d'un même curseur avec des valeurs différentes dans la clause WHERE du SELECT. DECLARE CURSOR nom_curseur(par1 type, par2 type, ...) IS ordre_select; BEGIN OPEN nom_curseur(val1, val2, ...); BEGIN FOR nom_struct IN nom_curseur(val1, val2, ...);

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.



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
                                                 IS
                grossal
                DISTINCT salemp FROM EMPLOYES
 SELECT
 ORDER BY salemp DESC;
 CURSOR
                C1(psal NUMBER) IS
 SELECT
                nomemp, preemp, salemp
 FROM
                EMPLOYES
 WHERE
                salemp = psal;
BEGIN
 For vgrossal IN grossal
   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

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