

Chapitre 2: Les curseurs

Objectifs

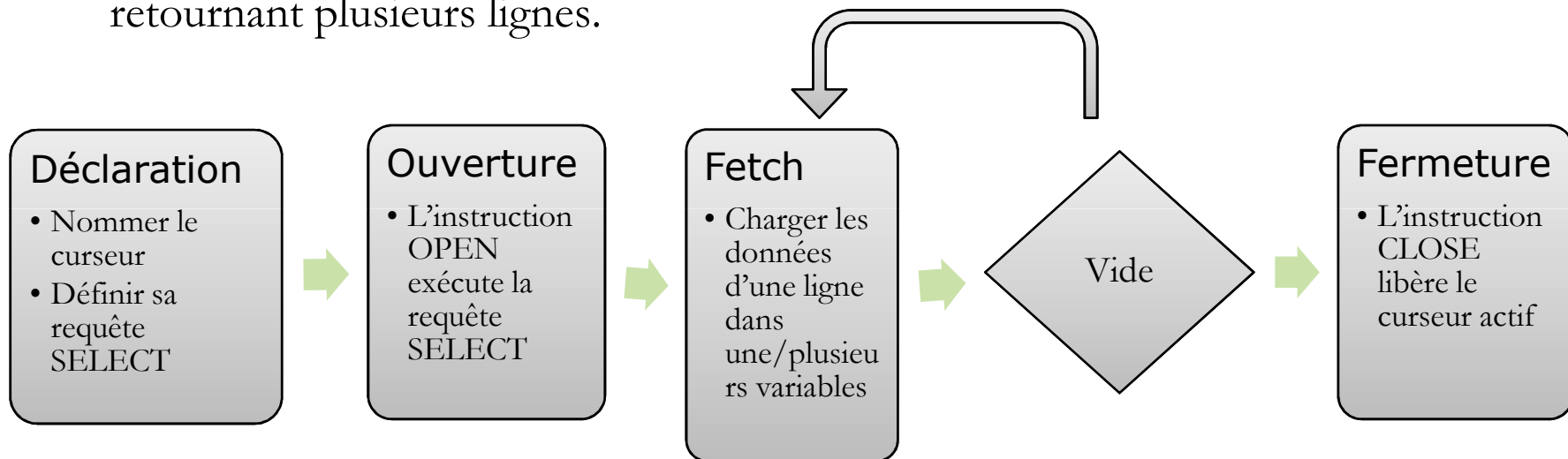
- Connaître le mécanisme de fonctionnement des curseurs
- Etre capable de traiter les curseurs :
 - Déclaration,
 - Ouverture,
 - Exécution
 - Fermeture
- Utiliser les curseurs simples et les curseurs paramétrés

Définition

- ❑ Un curseur est un mécanisme permettant de rechercher un nombre arbitraire de lignes avec une instruction SELECT.
- ❑ Types de curseurs:
 - **Le curseur implicite:** est créé pour toute opération de LMD, et toute requête SQL retournant une seule ligne, dont le résultat sera affecté à des variables.
 - **Le curseur explicite:** généré par l'utilisateur pour traiter un ordre SELECT qui ramène plusieurs lignes. Utilisation:
 - Déclaration
 - Ouverture du curseur
 - Traitement des lignes
 - Fermeture du curseur

Curseurs Explicites

- ❑ Il est créé par le programmeur pour gérer le résultat d'une requête SELECT retournant plusieurs lignes.



- **OPEN** : exécute la requête, identifie le résultat et place le curseur sur la première ligne
- **FETCH** : extrait la ligne courante et avance ensuite le curseur à la ligne suivante
- **CLOSE** : fermeture du curseur quand la dernière ligne est traitée. Le curseur est désactivé

Déclaration d'un curseur explicite

□ Syntaxe

CURSOR** **nom_curseur** **IS** **ordre_select

□ Exemple

Declare

```
CURSOR dept_10 IS select last_name, salary From  
employees where department_id = 10  
order by salary;
```

Begin

...

End;

Ouverture d'un curseur

- L'ouverture déclanche:
 - Allocation de mémoire pour le lignes du curseur
 - L'analyse syntaxique et sémantique du select
 - Le positionnement de verrous éventuels

Syntaxe:

OPEN nom_curseur;

Exemple

Declare

cursor dept_10 is select last_name, salary From employees
where department_id = 10 order by salary;

Begin

...;
open dept_10;

End;

Traitement de lignes

- ❑ Les lignes ramenées par le curseur seront traitées une par une, la valeur de chaque colonne doit être stockée dans une variable réceptrice.
- ❑ Syntaxe:
FETCH nom_curseur INTO liste_variables;
- ❑ Le Fetch ramène une ligne à la fois.

Exemple

Declare

```
CURSOR dept_10 is select last_name, salary From employees      where  
department_id = 10 order by salary;  
v_nom employees.last_name%TYPE;  
v_salaire employees.salary%TYPE;
```

Begin

```
Open dept_10;  
Loop  
    Fetch dept_10 into v_nom, v_salaire;  
    dbms_output.put_line (v_nom || ' ' || v_salaire);  
exit when dept_10%NOTFOUND;  
end loop;
```

End;

Fermeture d'un curseur

- ❑ Action: libère la place de mémoire
- ❑ Syntaxe:

***CLOSE** nom_curseur;*

DECLARE

 cursor dept_10 is select last_name, salary From employees
 where department_id = 10 order by salary;

BEGIN

 OPEN dept_10 ;

 CLOSE dept_10 ;

END;

/

Les attributs d'un curseur

- ❑ Définissent des indicateurs sur l'état d'un curseur.
 - %FOUND : nom_curseur%FOUND
 - TRUE: le dernier FETCH a ramené une ligne
 - FALSE: plus de ligne
 - %NOTFOUND: nom_curseur%NOTFOUND
 - TRUE: le FETCH n'a pas ramené de ligne
 - %ISOPEN: nom_curseur%ISOPEN
 - TRUE: le curseur est ouvert
 - %ROWCOUNT: nom_curseur%ROWCOUNT
 - Nombre de lignes ramenées par le FETCH

Exemple - %FOUND

Declare

```
cursor dept_10 is
    select last_name, salary From employees
    where department_id = 10 order by salary;
v_nom employees.last_name%TYPE;
v_salaire employees.salary%TYPE;
```

Begin

```
Open dept_10;
Fetch dept_10 into v_nom, v_salaire;
While dept_10%FOUND
Loop
    dbms_output.put_line ('le nom est : ' || v_nom);
    dbms_output.put_line ('le salaire est : ' || v_salaire);
    Fetch dept_10 into v_nom, v_salaire;
end loop;
close dept_10;
```

End;

Exemple - %NOTFOUND

Declare

```
cursor dept_10 is
    select last_name, salary From employees
    where department_id = 10 order by salary;
v_nom employees.last_name%TYPE;
v_salaire employees.salary%TYPE;
```

Begin

```
Open dept_10;
Loop
    Fetch dept_10 into v_nom, v_salaire;
    dbms_output.put_line ('le nom est : ' || v_nom);
    dbms_output.put_line ('le salaire est : ' || v_salaire);
Exit when dept_10%NOTFOUND ;
end loop;
close dept_10;
```

End;

Exemple - %ISOPEN

Declare

```
cursor dept_10 is
    select last_name, salary From employees
    where department_id = 10 order by salary;
v_nom employees.last_name%TYPE;
v_salaire employees.salary%TYPE;
```

Begin

```
If not(dept_10%ISOPEN) then Open dept_10; end if;
Loop
    Fetch dept_10 into v_nom, v_salaire;
    dbms_output.put_line ('le nom est : ' || v_nom);
    dbms_output.put_line ('le salaire est : ' || v_salaire);
Exit when dept_10%NOTFOUND;
end loop;
close dept_10;
```

End;

Exemple - %ROWCOUNT

Declare

```
cursor dept_10 is
    select last_name, salary From employees
    where department_id = 10 order by salary;
v_nom employees.last_name%TYPE;
v_salaire employees.salary%TYPE;
```

Begin

```
Open dept_10;
Loop
    Fetch dept_10 into v_nom, v_salaire;
    dbms_output.put_line ('le nom est : ' || v_nom);
    dbms_output.put_line ('le salaire est : ' || v_salaire);
Exit when dept_10%NOTFOUND or dept_10%ROWCOUNT > 15;
end loop;
close dept_10;
```

End;

Utilisation simplifiée des curseurs

- ❑ Déclarer une structure dont les éléments sont de même type que ceux du curseur

Declare

```
CURSOR cur_emp is select employee_id,last_name from employees;
```

```
V_cur cur_emp%ROWTYPE;
```

Begin

```
Open cur_emp;
```

```
Loop
```

```
    Fetch cur_emp into v_cur;
```

```
    Dbms_output.put_line('Identifiant : ' || v_cur.employee_id);
```

```
    Dbms_output.put_line('Nom : ' || v_cur.last_name);
```

```
Exit when cur_emp%NOTFOUND;
```

```
End loop;
```

```
Close cur_emp;
```

```
End;
```

Utilisation simplifiée des curseurs For .. IN

- ❑ La syntaxe FOR IN est encore plus concise, elle évite de déclarer le curseur dans la partie declare.

Begin

```
For emp_record in (select employee_id, last_name from employees)
loop
    --ouverture et fetch implicites
    Insert into emp_tmp values (emp_record.employee_id,
    emp_record.last_name);
end loop;
--sortie de la boucle (après fin du parcours du curseur) et fermeture
--du curseur implicites
commit;
End;
```

Les curseurs paramétrés

- ❑ Il s'agit de définir des paramètres au curseur lors de sa déclaration
- ❑ Instanciation de ses paramètres lors de l'ouverture du curseur (et donc lors de l'exécution de la requête du curseur)
- ❑ **Utilité :** ouvrir le curseur avec des valeurs (de paramètres) différentes

Syntaxe :

```
CURSOR cursor_name[(param1 datatype , param2 datatype,...)]  
IS req_select;
```

Ouverture :

```
OPEN cursor_name(par_val1, par_val2,...);
```


Les curseurs paramétrés

DECLARE

```
CURSOR cur_emp(v_dept number,v_sal employees.salary%TYPE) IS  
SELECT employee_id, last_name,salary FROM employees where  
department_id=v_dept and salary>v_sal ;  
rec_emp cur_emp%ROWTYPE ;
```

BEGIN

```
OPEN cur_emp(80,6000) ;  
LOOP  
FETCH cur_emp INTO rec_emp ;  
EXIT WHEN cur_emp%NOTFOUND ;  
dbms_output.put_line('Employee no : ' || rec_emp.employee_id || '  
Nom : ' || rec_emp.last_name || ' Salaire : ' || rec_emp.salary) ;  
END LOOP ;  
CLOSE cur_emp ;
```

END ;