

Bases de données
PL/SQL – curseurs

Silviu Maniu

2021/2022

Polytech App3

Introduction

PL/SQL : Programming Language for SQL = SQL + langage procédural (impératif)

Rappel :

- **declaratif** (SQL) : dire **quoi**
- **procedural** : dire **comment**

PL/SQL : l'union entre SQL et un langage de programmation

Le probleme :

```
select a,b into x,y from t where cle = 123
```

- récupère l'unique ligne du résultat d'une requête et place dans les variables x et y
- et si plusieurs lignes?

Curseur

- zone mémoire
- nommée
- associée à une requête
- peut contenir 0, 1 ou plusieurs lignes
- taille déterminée à l'exécution
- sert à contenir et traiter l'ensemble des lignes résultat de cette requête

Étapes :

1. **déclaration** du curseur
2. **remplissage** en une seule fois par exécution de la requête
3. **recupération** des lignes une par une (parcours séquentiel, pointeur logique)
4. **libération** de la zone

Curseur PL/SQL – Exemple (première ligne)

Table `article(refart,nom,prixht)` – récupérer première ligne par ordre alphabétique

- `select into` impossible car plusieurs lignes possibles

```
declare
  cursor c1 is
    select refart from article order by nom;
  art article.refart%type;
begin
  open c1;
  fetch c1 into art; -- affectation de variable (fetch into)
  dbms_output.putline(art);
  close c1;
end;
```

Curseur PL/SQL – Exemple (première et deuxième ligne)

Table `article(refart,nom,prixht)` – récupérer première ligne par ordre alphabétique et la deuxième si elle existe

- **indicateurs curseur** : `c%found` (true si dernier fetch a trouvé une ligne), `c%notfound`, `c%isopen`

```
declare
```

```
  cursor c2 is select refart from article order by nom;  
  art article.refart%type;
```

```
begin
```

```
  open c2; fetch c2 into art; dbms_output.putline(art);  
  fetch c2 into art;  
  if c2%found  
  then dbms_output.putline(art);  
  close c2;
```

```
end;
```


Parcours : analogue à un parcours de fichier séquentiel

- **pointeur logique**

Après `open` :

- pointeur positionné **avant** la première ligne
- `c%found` est `true`

`fetch c into ... :`

- avance le pointeur et lit la ligne pointée
- si pas de ligne, `c%found` devient `false`

Curseur PL/SQL – Exemple (tout) 1/2

Table article(refart,nom,prixht) – afficher toutes les références

```
declare
  cursor c3 is select refart from article order by nom;
  art article.refart%type;
begin
  open c3; fetch c3 into art;
  while c3%found loop
    dbms_output.putline(art);
    fetch c3 into art;
  end loop;
  close c3;
end;
```

Curseur PL/SQL – Exemple (tout) 2/2

Table `article(refart,nom,prixht)` – afficher toutes les références et le prix

```
declare
  cursor c4 is select refart, prixht from article;
  art c4%rowtype; -- type dérivé (curseur)
begin
  open c4; fetch c4 into art;
  while c4%found loop
    dbms_output.putline(art.refart || ' ' || art.prixht);
    fetch c4 into art;
  end loop;
  close c4;
end;
```

Curseur PL/SQL – Utilisation boucles

Boucle `for` dans un curseur `c`, variable `x` de type `c%rowtype` :

```
for x in c loop
    dbms_output.put_line(x.a || ' ' || x.b);
end loop;
```

Boucle `loop...exit when` :

```
loop
    fetch c into x;
    exit when c%notfound;
    dbms_output.put_line(x.a || ' ' || x.b);
end loop;
```

Remerciements

Le contenu de ce cours est grandement inspiré des cours d'Emmanuel Waller

- <https://www.lri.fr/~waller/>