

Principes d'utilisation des systèmes de gestion de bases de données

PL/SQL : curseurs

M1 Informatique
Emmanuel Waller, LRI, Orsay

Curseurs : le problème

Select a,b into x,y from t where clé = 123

Récupère l'unique ligne du résultat de la requête

La place dans le couple de variables (x,y)

But : récupérer un résultat de requête faisant plusieurs lignes

Curseur

= zone mémoire

Nommée

À laquelle est associée une requête

Peut contenir 0, 1 ou plusieurs lignes

Taille réglée à l'exécution

Sert à contenir l'ensemble des lignes résultat de
cette requête

Fonctionnement

Déclaration du curseur

Remplissage en une seule fois par exécution de la requête

Récupération des lignes une par une

(parcours séquentiel du curseur par un pointeur logique)

Libération de la zone : elle devient inaccessible

Exemple 1

Table article(refart, nom, prixht)

Afficher à l'écran la référence du premier article par ordre alphabétique

On suppose qu'il y en a au moins un

Select into impossible car il peut y en avoir plusieurs

Declare

Exemple

```
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);
```

```
    -- éventuels autres non parcourus
```

```
    close c1;
```

```
end;
```

Exemple 2

Afficher à l'écran la référence du premier article par ordre alphabétique, et du deuxième s'il existe

On suppose qu'il y en a au moins un

Attributs : indicateurs sur l'état du curseur

Ex : c2 curseur

c2%found : booléen, à true ssi le dernier fetch a ramené une ligne

c2%notfound

C2%isopen

Declare

Exemple

```
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 du curseur

Analogue à parcours de fichier séquentiel

Notion de pointeur logique

Après le open :

pointeur logique positionné avant la 1ère ligne

c⁰%found est à vrai

Fetch c into a :

Avance le pointeur

Lit la ligne pointée par le pointeur

Si pas de ligne pointée (l'avant-dernier fetch avait lu la dernière) : c⁰%found devient faux

Exemple 3

Afficher à l'écran la référence de tous les articles

On ne suppose rien

Declare

Exemple

```
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;
```

Exemple 4

Afficher la référence et le prix de tous les articles

Declare

Exemple

```
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;
```

Curseurs et boucles : mieux

```
cursor c;
```

```
  for x in c loop
```

```
    print(x.a, x.b);
```

```
  end loop;
```

```
loop
```

```
  fetch c into x;
```

```
  exit when c%notfound;
```

```
  dbms_output.put_line(x.a, x.b);
```

```
end loop;
```

démonstration