420-KBA-LG, programmation de bases de données

Saliha Yacoub

Les curseurs

- Retour sur les dernières séances:
  - Point de vue des enseignants:
    - Lisez les consignes
    - Lisez les notes de cours (pas uniquement le Power Point)
    - Lisez les exemples et les solutionnaires lorsqu'ils sont disponibles
    - On respire ….
  - Point de vue de l'étudiant
- Les curseurs
  - Définition
  - Syntaxe de déclaration
  - Exemples

Les curseurs

- Définition: Les curseurs sont des zones mémoire utilisées par les SGBDs pour récupérer un ensemble de résultats issu d'une requête SELECT.
- Pour MS SQL Server, les curseurs sont explicites, ce qui veut dire qu'ils sont associés à une requête SELECT bien précise.
- Les données sont disponibles dans le curseur au moment de sa déclaration.

Les curseurs

- Définition: Les curseurs sont des zones mémoire utilisées par les SGBDs pour récupérer un ensemble de résultats issu d'une requête SELECT.
- Pour MS SQL Server, les curseurs sont explicites, ce qui veut dire qu'ils sont associés à une requête SELECTE bien précise.
- Les données sont disponibles dans le curseur au moment de sa déclaration.

```
DECLARE nomCurseur   CURSOR FOR SELECT ...  FROM ...
```

Un curseur a la même structure qu'une table.

Un curseur a besoin de son propre DECLARE et n'a pas @ devant son nom.

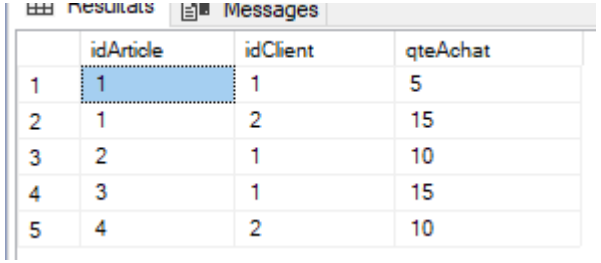
Les curseurs

- Lecture du contenu d'un curseur:
  - Pour lire un curseur, il faut l'ouvrir avec la fonction OPEN. (comme un fichier)
  - La commande **FETCH NEXT FROM.....INTO** (un peu comme READ) permet de lire l'enregistrement suivant. (une seule ligne à la fois).
  - On utilise une boucle WHILE pour parcourir l'ensemble des enregistrements du curseur. On arrêter lorsque le curseur est vide.
  - On ferme le curseur(très important)
  - On libère les ressources occupées par le curseur. (très important)

Les curseurs

› Exemple:

Voici le contenu de la table panierAchat qui représente un panier d'achat dans votre BD stockClg.


Résultats Messages

	idArticle	idClient	qteAchat
1	1	1	5
2	1	2	15
3	2	1	10
4	3	1	15
5	4	2	10

Lorsque, je déclare le curseur curPanier

declare curPanier cursor for select idArticle, qteAchat from PanierAchat where idClient =1;

idArticle	qteAchat
1	5
2	10
3	15

Les curseurs

Comment lire le contenu du curseur ? En transférant son contenu dans des variables et on lit les variables) . On procède par étape.

Étape 0: On ouvre le curseur avec OPEN.

Étape 1: on se déplace au premier enregistrement et on met le contenu dans des variables déjà déclarées.

```
fetch next from curPanier into @idArticle,@qtAchat;
```

idArticle	qteAchat
1	5
2	10
3	15

Étape 3: On utilise une boucle WHILE pour parcourir tout le curseur. À chaque itération, le pointeur (rouge) va à l'enregistrement suivant. On arrête lorsqu'il n' y a plus de ligne à lire.

La fonction @@FETCH_STATUS : Renvoie l'état de la dernière instruction FETCH effectuée sur un curseur. Elle renvoie 0 si tout s'est bien passé.

```
WHILE@@FETCH_STATUS=0
```

idArticle	qteAchat
1	5
2	10
3	15

Étape 4: On ferme le curseur avec CLOSE

Étape5: On libère les ressources occupées par le curseur: DEALLOCATE

Les curseurs.

- Par défaut, les curseur sont Forwrad ONLY, avec un parcours en avant seulement.
- Lorsque vous récupérer le contenu du curseur dans des variables, ces variables sont utilisées comme des variables ordinaires.
- Dans l'exemple suivant, lorsque le panier d'un client est supprimé alors la quantité en stock des articles qui étaient dans le panier est mise à jour.

Les curseurs.

```
create procedure supprimerPanier(@idClient smallint) as
begin
   Declare @qtAchat int, @idArticle int;
   declare curPanier cursor for select idArticle, qteAchat from PanierAchat   where idClient   =@idClient;
      open curPanier;
      fetch next from curPanier into @idArticle,@qtAchat;
       while @@FETCH_STATUS=0
                  begin
                  update Articles   set quantite_stock = quantite_stock + @qtAchat where idArticle =@idArticle;
                  fetch next from curPanier into @idArticle,@qtAchat;
                  end;
       delete from PanierAchat where idClient =@idClient;
       close curPanier;
      DEALLOCATE curPanier;
end;
```

Les curseurs.

Curseur scrollable:

- Par défaut, les curseurs sont Forward ONLY : ils ne sont pas scrollables.
- Lorsqu'un curseur est déclaré avec l'attribut SCROLL alors on peut accéder au contenu du curseur par d'autres options de la fonction FETCH.
- Nous pouvons avoir accès à la première ligne, la dernière ligne, une position absolue, exemple la ligne 3. Position relative à partir d'une position prédéfinie.
- L'accès au contenu du curseur se fait directement à l'endroit souhaité (accès direct).

Les curseurs.

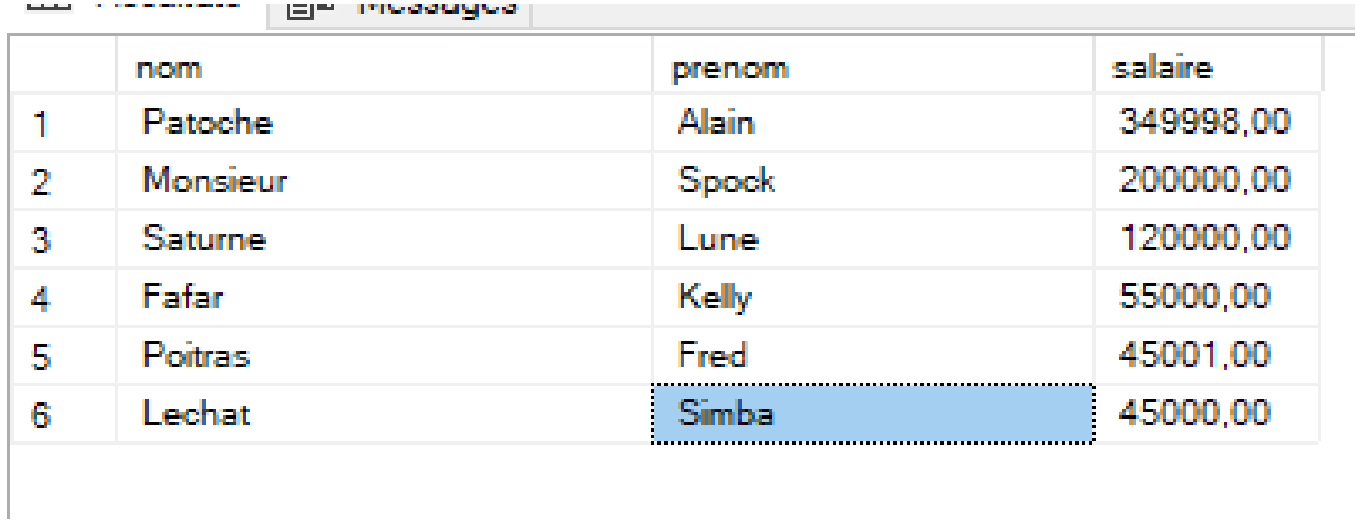
Quelques fonctions du curseur scrollable:

- FETCH FIRST : va à la première ligne du curseur
- FETCH LAST : va au dernier enregistrement
- FETCH ABSOLUTE n: va à la nième position dans le curseur
- FETCH RELATIVE n :pas à la nieme ligne après la position courante (n peut-être positif ou négatif).
- FETCH PRIOR :va à la ligne immédiatement avant la position courante

Les curseurs.

declare curempPermanent scroll cursor for select nom, prenom, salaire from employesClg C inner join EmpPermanent P on c.empno =P.empno order by salaire desc;

Le contenu du curseur est le suivant:



	nom	prenom	salaire
1	Patoche	Alain	349998,00
2	Monsieur	Spock	200000,00
3	Saturne	Lune	120000,00
4	Fafar	Kelly	55000,00
5	Poitras	Fred	45001,00
6	Lechat	Simba	45000,00

Le code de la page suivante, va me permettre d'accéder directement aux lignes du curseur.

Les curseurs.

```
begin
declare @nom varchar(30),
        @prenom varchar(30),
         @salaire money;
declare curempPermanent scroll cursor for select nom, prenom, salaire
from employesClg C inner join EmpPermanent P on c.empno =P.empno order by salaire desc;

open curempPermanent;
            print(' la premiere ligne');
            fetch first from curempPermanent into @nom,@prenom, @salaire;
            print concat (@nom,'----', @prenom,'----', @salaire);
 close curempPermanent;
deallocate curempPermanent;
end;
```

Les curseurs.

Le résultat sera: Patoche----Alain----349998.

Avec le même code, mais on met:

```
fetch absolute 3 from curempPermanent into @nom,@prenom, @salaire;
```

Le résultat sera: Saturne----Lune----120000.00

Nous somme à la 3eme ligne, puis on fait relative de 2

```
fetch absolute   3 from curempPermanent into @nom,@prenom, @salaire;
fetch relative   2 from curempPermanent into @nom,@prenom, @salaire;
```

Le résultat sera: (on avance de 2 par rapport à Saturne.

Poitras----Fred----45001.00

```
fetch absolute   3 from curempPermanent into @nom,@prenom, @salaire;
fetch relative   -1 from curempPermanent into @nom,@prenom, @salaire;
```

Le résultat sera: On recule de 1, par rapport à saturne

Monsieur----Spock----200000.00

Les curseurs.

Conclusion.

- Les curseurs sont des variables qui se déclarent avec la commande SELECT: **DECLARE nomCurseur CURSOR FOR SELECT**
- Pour lire un curseur, il faut d'abord l'ouvrir. Tout curseur ouvert doit-être obligatoirement fermé. Les curseurs ouverts non fermés provoquent des erreurs graves.
- La méthode FETCH ..NEXT permet de passer à l'enregistrement suivant. Dans ce cas, on parle d'accès **séquentiel.**
- Par défaut, les curseurs ne sont pas scrollables.
- Si le curseur est scrollable alors il est possible d'accéder directement aux lignes de celui-ci. On parle d'accès **direct**.
- N'utilisez pas des curseurs pour faire un simple SELECT. Ce n'est pas utile et c'est déconseillé.
  - Ça occupe des ressources. Si vous oubliez le deallocate→ problème
  - Si vous oubliez de fermer le curseur → problème.

CONCLUSION

QUESTIONS ??