



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**
Membre de **HONORIS UNITED UNIVERSITIES**

Compte rendu

Série N°2 : Curseurs T-SQL + Transactions



Professeur :

-Mme. EL HAIBA Maria

Réalise par :

- EL assili kaoutar

-Boulakssili Dhoha

Exercice 1 : Série 2 TP des curseurs

1. Ecrire un programme qui affiche la liste des articles sous la forme

♣ L'article Numéro portant la désignationcoûte

```
SQLQuery1.sql - D:\P-RO0H3PR\pc (60))* -> X
declare @n int, @d VARCHAR(40), @p DECIMAL;
declare cur CURSOR FOR select numArt, DesArt, PUArt from Article;

open cur;

fetch next from cur into @n, @d, @p;

while @@FETCH_STATUS = 0
BEGIN
print CONCAT('L'article Numéro ',@n, ' portant la designation ',@d, ' coute ', @p);
fetch next from cur into @n, @d, @p;
end;
close cur;
DEALLOCATE cur;
```

100 %

Messages

```
L'article Numero 1 portant la designation Biscuit coute 12
L'article Numero 2 portant la designation Jus coute 20
L'article Numero 3 portant la designation Yougurt coute 5
L'article Numero 4 portant la designation Ordinateur coute 7000
L'article Numero 5 portant la designation Table coute 500
L'article Numero 6 portant la designation Imprimante Lazer HP coute 700
L'article Numero 7 portant la designation Routeurs sans fils coute 200
L'article Numero 8 portant la designation Disques durs 500Go coute 175
L'article Numero 9 portant la designation Smartphone coute 6500
L'article Numero 10 portant la designation TV coute 10000

Heure de fin : 2023-05-19T21:13:24.1179898+02:00
```

2. Ecrire un programme qui pour chaque commande :

♣ Affiche le numéro et la date de commande sous la forme : Commande N° : Effectuée le : ...

♣ Le montant de cette commande, Son montant est : ...

```

SQLQuery1.sql - D:\P-RO0H3PR\pc (60)) *  X
BEGIN
declare @var1 int, @var2 Date, @var3 DECIMAL;
declare cur3 CURSOR FOR select c.numCom, DatCom, 'Montant' = Sum(QteCommandee * PUArt) from Commande c
inner join Com_Art ca on c.NumCom = ca.NumArt inner join Article a on ca.NumArt = a.NumArt group by c.NumCom, DatCom;
BEGIN
open cur3
fetch next from cur3 into @var1, @var2, @var3;
while @@FETCH_STATUS = 0
BEGIN
print CONCAT('Commande No : ', @var1, ' Effectuee le : ', @var2);
print CONCAT('Le montant de cette commande, Son montant est : ', @var3);
fetch next from cur3 into @var1, @var2, @var3;
end;
close cur3
DEALLOCATE cur3
end
end

```

100 %

Messages

```

Commande No : 1 Effectuee le : 2020-03-12
Le montant de cette commande, Son montant est : 120
Commande No : 2 Effectuee le : 2020-05-13
Le montant de cette commande, Son montant est : 500
Commande No : 3 Effectuee le : 2020-08-15
Le montant de cette commande, Son montant est : 20
Commande No : 4 Effectuee le : 2021-01-10
Le montant de cette commande, Son montant est : 14000

Heure de fin : 2023-05-19T21:19:21.8500675+02:00

```

3. Ecrire un programme qui pour chaque commande vérifie si cette commande a au moins un article.

♣ Si c'est le cas : affiche la liste de ses articles (Numéro, Désignation et Prix) et sa quantité commandée,

♣ Sinon : affiche un message d'erreur : « Aucun article pour la commande N°.... Elle sera supprimée », Et supprime cette commande.

```

SQLQuery1.sql - D:\P-RO0H3PR\pc (60)) *  X
BEGIN
declare @numcom int, @numart int, @cout int;
declare cur4 CURSOR for SELECT numcom from Commande;
open cur4;
fetch next from cur4 into @numcom;
while @@FETCH_STATUS = 0
BEGIN
if (select count(*) from Com_Art where NumCom = @numcom) >= 1
BEGIN
print CONCAT('Commande No ', @numcom)
select ca.numcom, ca.numart, desart, PUArt, qteCommandee from Commande c inner join Com_Art ca on c.NumCom = ca.NumArt
inner join Article a on ca.NumArt = a.NumArt where ca.NumCom = @numcom;
end;
else
BEGIN
print CONCAT('Aucun article pour la commande No : ', @numcom, ' Elle sera supprimee ');
delete from Commande where NumCom = @numcom;
end;
fetch next from cur4 into @numcom;
end;
close cur4;
DEALLOCATE cur4;
END;
select * from commande;
select * from commande where numcom not in (select numcom from Com_Art);
insert into Commande values( 20, '2020-02-12', 1);

```

100 %

Résultats

Messages

| | numcom | numart | desart | PUArt | qteCommandee |
|---|--------|--------|---------|-------|--------------|
| 1 | 1 | 1 | Biscuit | 12 | 10 |
| 2 | 1 | 2 | Jus | 20 | 5 |
| 3 | 1 | 3 | Yogurt | 5 | 4 |

| | numcom | numart | desart | PUArt | qteCommandee |
|---|--------|--------|------------|-------|--------------|
| 1 | 2 | 4 | Ordinateur | 7000 | 2 |

| | numcom | numart | desart | PUArt | qteCommandee |
|---|--------|--------|--------|-------|--------------|
| 1 | 3 | 2 | Jus | 20 | 20 |

| | numcom | numart | desart | PUArt | qteCommandee |
|--|--------|--------|--------|-------|--------------|
|--|--------|--------|--------|-------|--------------|

| | NumCom | DatCom | NumCl |
|---|--------|------------|-------|
| 1 | 1 | 2020-03-12 | 1 |
| 2 | 2 | 2020-05-13 | 2 |
| 3 | 3 | 2020-08-15 | 3 |
| 4 | 4 | 2021-01-10 | 4 |

| | NumCom | DatCom | NumCl |
|--|--------|--------|-------|
|--|--------|--------|-------|

4. Ecrire un programme permettant d'accéder directement à la première ligne du curseur relatif aux clients (Nom, Prénom) avec le montant de leurs commandes par ordre décroissant.

- ♣ Ensuite, afficher l'enregistrement se trouvant à la ligne N°3 du curseur,
- ♣ Afficher le contenu de la deuxième ligne après la position courante,
- ♣ Afficher l'enregistrement se trouvant immédiatement avant la position courante

```
SQLQuery1.sql - D:\P-RO0H3PR\pc (60)) * -> X
SELECT c.Nom, c.Prenom, SUM(ca.QteCommandee * a.PUArt) AS MontantCommandes
FROM Client c
JOIN Commande com ON c.NumCl = com.NumCl
JOIN Com_Art ca ON com.NumCom = ca.NumCom
JOIN Article a ON ca.NumArt = a.NumArt
GROUP BY c.NumCl, c.Nom, c.Prenom
ORDER BY MontantCommandes DESC

SELECT *
FROM (
    SELECT c.Nom, c.Prenom, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS RowNum
    FROM Client c
    JOIN Commande com ON c.NumCl = com.NumCl
    JOIN Com_Art ca ON com.NumCom = ca.NumCom
    JOIN Article a ON ca.NumArt = a.NumArt
    GROUP BY c.NumCl, c.Nom, c.Prenom
) AS subquery
WHERE RowNum = 3;

SELECT *
FROM (
    SELECT c.Nom, c.Prenom, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS RowNum
    FROM Client c
    JOIN Commande com ON c.NumCl = com.NumCl
    JOIN Com_Art ca ON com.NumCom = ca.NumCom
    JOIN Article a ON ca.NumArt = a.NumArt
    GROUP BY c.NumCl, c.Nom, c.Prenom
) AS subquery
WHERE RowNum = (SELECT RowNum + 2 FROM subquery WHERE Nom = 'Position courante' AND Prenom = 'Position courante');

SELECT *
FROM (
    SELECT c.Nom, c.Prenom, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS RowNum
    FROM Client c
```

SQLQuery1.sql - D:\P-RO0H3PR\pc (60)* -> X

```

) AS subquery
WHERE RowNum = 3;
SELECT *
FROM (
    SELECT c.Nom, c.Prenom, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS RowNum
    FROM Client c
    JOIN Commande com ON c.NumCl = com.NumCl
    JOIN Com_Art ca ON com.NumCom = ca.NumCom
    JOIN Article a ON ca.NumArt = a.NumArt
    GROUP BY c.NumCl, c.Nom, c.Prenom
) AS subquery
WHERE RowNum = (SELECT RowNum + 2 FROM subquery WHERE Nom = 'Position courante' AND Prenom = 'Position courante');
SELECT *
FROM (
    SELECT c.Nom, c.Prenom, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS RowNum
    FROM Client c
    JOIN Commande com ON c.NumCl = com.NumCl
    JOIN Com_Art ca ON com.NumCom = ca.NumCom
    JOIN Article a ON ca.NumArt = a.NumArt
    GROUP BY c.NumCl, c.Nom, c.Prenom
) AS subquery
WHERE RowNum = (SELECT RowNum - 1 FROM subquery WHERE Nom = 'Position courante' AND Prenom = 'Position courante');

```

Résultats Messages

| | Nom | Prenom | MontantCommandes |
|---|---------|----------|------------------|
| 1 | Oussama | Fihri | 14000 |
| 2 | Joseph | Antonio | 4875 |
| 3 | Touria | Karam | 400 |
| 4 | Karim | Yahyaoui | 240 |

| | Nom | Prenom | RowNum |
|---|--------|--------|--------|
| 1 | Touria | Karam | 3 |

Exercice 2 : Les transactions

```

-- si vous avez deja cree la colomn
ALTER TABLE Com_Art
DROP COLUMN Montant;
-- Modifier la table
Alter table Com_Art ADD Montant float;
-- remplir la table
UPDATE Com_Art set Montant = PUArt* QteCommandee from
Com_art c join Article a
ON c.NumArt = a.NumArt;
Declare @Qte int =1 , @prix float;--quantité commandée et prix
select @prix = PUArt from Article where NumArt = 6;

BEGIN TRANSACTION; -- Début de la transaction

BEGIN TRY
    -- a) Création d'une nouvelle commande pour le client numéro 7
    INSERT INTO Commande (NumCom, DatCom, NumCl)
    VALUES (6, GETDATE(), 7);

    -- b) Insertion de la commande dans la table Com_Art
    INSERT INTO Com_Art (NumCom, NumArt, QteCommandee)
    VALUES (6, 6, 10);

    -- ou bien INSERT INTO Com_Art VALUES (6, 6, @Qte,NULL);

    -- c) Vérification de la quantité disponible dans la table Article
    DECLARE @QuantiteEnStock INT;
    SELECT @QuantiteEnStock = QteEnStock
    FROM Article
    WHERE NumArt = 6;

```

```

SQLQuery2.sql - D:\P-RO0H3PR\pc (53)) * - P X
WHERE NumArt = 6;

IF @QuantiteEnStock < @Qte
BEGIN
    -- Quantité insuffisante, annulation de la transaction
    ROLLBACK TRANSACTION;
    PRINT 'La quantité en stock est insuffisante. La transaction a été annulée.';
END
ELSE
BEGIN
    -- d) Mise à jour de la quantité dans la table Article
    UPDATE Article
    SET QteEnStock = QteEnStock - @Qte
    WHERE NumArt = 6;

    -- Mise à jour du montant de la commande pour cet article dans la table Com_Art
    UPDATE Com_Art
    SET Montant = @prix * @Qte
    WHERE NumCom = 6 AND NumArt = 6; --Numero d'article = 6 et Numero de la commande est 6

    -- Officialisation de la transaction
    COMMIT TRANSACTION;
    PRINT 'La transaction a été effectuée avec succès.';
END
END TRY
BEGIN CATCH
    -- En cas d'erreur, annulation de la transaction et affichage du message d'erreur
    ROLLBACK TRANSACTION;
    PRINT 'Une erreur s'est produite lors de la transaction : ' + ERROR_MESSAGE();
END CATCH;

```



Messages

(1 ligne affectée)

(1 ligne affectée)

(1 ligne affectée)

(1 ligne affectée)

La transaction a été effectuée avec succès.

Heure de fin : 2023-05-20T18:18:41.7341335+02:00