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

Série N°3 : Procédures et Fonctions Stockées

Considérons la base de données **GestionC** suivante :

Article (NumArt, DesArt, PUArt, QteEnSTock, SeuilMinimum, SeuilMaximum)

Client (NumCl, Nom, Prenom, Ville)

Commande (NumCom, DatCom, #NumCl)

Com_Art (#NumCom, #NumArt, QteCommandee, MtArt)

Montant de l'achat pour l'article

Exercice 1 : Procédure stockée

Créer une procédure stockée nommée **AjouterCommande** qui permet de créer une commande pour un client. Elle reçoit un numéro de commande, un numéro de client, une date, un numéro d'article et la quantité commandée.

Cette procédure fait les opérations suivantes :

1) Si **l'article n'existe pas** ou si la **quantité demandée n'est pas disponible**, afficher un **message d'erreur**
2) Sinon, Si la **commande** introduite en paramètre **n'existe pas**, la **créer** *(à insérer dans la table Commande)*

- **Ajouter** ensuite la ligne de commande *(à insérer dans la table Com_Art)*
- **Mettre à jour** la quantité en stock après achat *(dans la table Article)*
- **Mettre à jour** le montant de l'achat pour l'article *(dans la table Com_Art)*

Exercice 2 : Fonctions stockée

Ecrire un programme qui permet de retourner le détail de **chaque** commande (Afficher la désignation, le prix, la quantité commandée et le montant par ligne) en utilisant **les fonctions** (Pour afficher le détail) et **les curseurs** (Pour parcourir les commandes).

Le résultat devrait apparaître comme suit :

Résultats

```
Le détail de la commande 1 est:
DesArt                                             PUArt                QteCommandee MtArt
-------------------------------------------------- -------------------- ------------ ------
Biscuit                                            12                   10           120
Jus                                                20                   5            100
Yougurt                                            5                    4            20

(3 lignes affectées)

Le détail de la commande 2 est:
DesArt                                             PUArt                QteCommandee MtArt
-------------------------------------------------- -------------------- ------------ ------
Ordinateur                                         7000                 2            14000
```

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

Correction Exercice 1 : Procédure

```
CREATE or alter PROCEDURE ajouterCommande(@numcom int, @numcl int,@date date, @numart int,
@quaCde int)
as
BEGIN
declare @prix float, @quantite int, @mtart float;
select @prix = PUArt from Article where NumArt=@numart;
select @quantite = QteEnStock from Article where NumArt=@numart;
begin transaction
    if not exists(select NumArt from article where NumArt=@numart)
       or (@quantite <= @quaCde)
         Begin
          Rollback transaction;
          Print 'Cet article n''existe pas ou le Stock est insuffisant';
         End
    else
         Begin
         if not exists(select NumCom from Commande where NumCom=@numcom)
         begin
          insert into Commande values(@numcom,@date,@numcl);
          insert into Com_Art(NumCom,NumArt,QteCommandee) values (@numcom,@numart,@quaCde);
          update Article set QteEnStock =QteEnStock-@quaCde where NumArt=@numart;
          update Com_Art set MtArt =@quaCde * @prix where NumArt=@numart and NumCom=@numcom;
          Commit transaction;
          Print 'Commande ajoutée avec succès';
         End;
         End;
END;
GO
```

Exécution de la Procédure

```
/*Exemple d'Exécution*/
--NumClient = 7, NumCommande = 6, NumArticle = 6 (Imprimante, Stock dispo = 25)
Declare @d Date
Set @d=getdate()
Execute ajouterCommande 6, 7,@d, 6, 30
```

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

Correction Exercice 2 : Fonction

```
create or alter function DetailCommande(@commande int) returns table
as
Return(
select  A.DesArt, A.PUArt, CA.QteCommandee, MtArt
From Article A inner join Com_Art CA on A.NumArt=CA.NumArt
            inner join Commande C on C.NumCom=CA.NumCom
               where C.NumCom=@commande
)
--Select * from DetailCommande (1) --Exécution de la fonction pour vérification
GO

--Curseur faisant appel à la fonction créée
declare @commande int
declare c cursor for select NumCom from Commande
open c
fetch next from c into @commande
while (@@FETCH_STATUS=0)
begin
print concat('Le détail de la commande ' , @commande ,' est: ')
select * from DetailCommande(@commande)
fetch next from c into @commande
end
close c
deallocate c
```