

Exercices Transact SQL

Objet : les procédures stockées, les curseurs, les transactions

Vous travaillez pour la gestion de stocks d'une entreprise disposant de plusieurs entrepôts à travers la France. Il peut y avoir plusieurs entrepôts par départements, mais pas plus d'un dans une même ville.

1/ Creation d'une base de donnée et de ses tables sur SQL Serveur

PRODUIT(codeProduit, libelle, pu)

ENTREPOT(codeEntrepot, adresserue, ville, codep, codecom)

STOCK(#codeProduit, #codeEntrepot, quantitéStock, seuilAlerte, quantitéMaxi))

2/ Ecrire une procédure stockée permettant d'apprécier la répartition des produits dans les différents entrepôts. Pour cela, créer une table temporaire qui contiendra pour chaque produit : le codeProduit, le libelle, le nombre d'entrepôts où il n'est pas présent, le nombre d'entrepôt où il est en rupture de stock, le nombre d'entrepôt où il est en quantité suffisante

```
create PROCEDURE StatsProduit
AS
BEGIN
DECLARE @code int, @libelle varchar(20),@nbEntrepot int,@nbEntrepotNonAssigne int, @nbRupture
int, @nbSuffisant int
create table #StatProduit(code int, libelle varchar(50),NonPresent int,Rupture int, quantiteSuffisante int)
set @nbEntrepot = (select COUNT(distinct codeEntrepot) from Entrepot)
DECLARE leCurseur cursor
for
select codeProduit,libelle from Produit
open leCurseur
fetch leCurseur into @code,@libelle
while @@FETCH_STATUS = 0
begin
set @nbEntrepotNonAssigne = @nbEntrepot - (select COUNT(distinct codeEntrepot) from Stock where
codeProduit = @code)
set @nbRupture = (select COUNT(distinct codeEntrepot) from Stock where codeProduit = @code and
quantitéStock < seuilAlerte)
set @nbSuffisant = (select COUNT(distinct codeEntrepot) from Stock where codeProduit = @code and
quantitéStock > seuilAlerte)
insert into #StatProduit values(@code,@libelle,@nbEntrepotNonAssigne,@nbRupture,@nbSuffisant)
fetch leCurseur into @code, @libelle
end
close leCurseur
```

```
deallocate leCurseur
select * from #StatProduit
end
```

3/ Ecrire une procédure stockée permettant de créer une table temporaire contenant, par département, le code du département, le nombre d'entrepôts présent dans le département et le nom de la ville contenant l'entrepôt ayant le stock de plus grande valeur (somme des pu * quantité des produits de l'entrepôt):

```
create PROCEDURE depEntrpot
AS
BEGIN
DECLARE @codedep int, @nbEntrepotPresent int, @nomVilleEntrepot varchar(50), @ville varchar(50)
create table #depEntrpot(codedep int, nbEntrepotPresent int, nomVilleEntrepot varchar(50))

DECLARE leCurseur cursor
for
select codep from Entrepot
open leCurseur
fetch leCurseur into @codedep
while @@FETCH_STATUS = 0
begin
set @nbEntrepotPresent = (Select COUNT(distinct codeEntrepot) from Entrepot Where codep =
@codedep)
set @nomVilleEntrepot = (select ville from Produit,Stock,Entrepot where Produit.codeproduit =
Stock.codeProduit and Stock.codeEntrepot = Entrepot.codeEntrepot and codep = @codedep group by
ville having SUM(pu * quantitéStock) >= all (select SUM(pu * quantitéStock) from Produit,Stock,Entrepot
where Produit.codeproduit = Stock.codeProduit and Stock.codeEntrepot = Entrepot.codeEntrepot and
codep = @codedep))
insert into #depEntrpot values(@codedep, @nbEntrepotPresent, @nomVilleEntrepot)
fetch leCurseur into @codedep
end
close leCurseur
deallocate leCurseur
select * from #depEntrpot
end
```