USE Storage

GO

CREATE PROCEDURE InsertFromBranche

AS

BEGIN

BEGIN

INSERT INTO Storage.dbo.Customer (CustomerID, CustomerName)

SELECT CustomerID, CustomerName

FROM Branch\_1.dbo.Customer

WHERE Branch\_1.dbo.Customer.DateAdded > (select MAX(DateSynchron) from Storage.dbo.Synchronizes)

UNION ALL

SELECT CustomerID, CustomerName

FROM Branch\_2.dbo.Customer

WHERE Branch\_2.dbo.Customer.DateAdded > (select MAX(DateSynchron) from Storage.dbo.Synchronizes)

END

BEGIN

INSERT INTO Storage.dbo.[Order](OrderSystemID, DatePurchase)

SELECT CheckID, Date

FROM Branch\_1.dbo.[Check]

WHERE Branch\_1.dbo.[Check].DateAdded > (select MAX(DateSynchron) from Storage.dbo.Synchronizes)

UNION ALL

SELECT CheckID, Date

FROM Branch\_2.dbo.[Check]

WHERE Branch\_2.dbo.[Check].DateAdded > (select MAX(DateSynchron) from Storage.dbo.Synchronizes)

END

BEGIN

INSERT INTO Storage.dbo.Product (ProductID, ProductName)

SELECT ProductID, ProductName

FROM Branch\_1.dbo.Product

WHERE Branch\_1.dbo.Product.DateAdded > (select MAX(DateSynchron) from Storage.dbo.Synchronizes)

UNION ALL

SELECT ProductID, ProductName

FROM Branch\_2.dbo.Product

WHERE Branch\_2.dbo.Product.DateAdded > (select MAX(DateSynchron) from Storage.dbo.Synchronizes)

END

BEGIN

INSERT INTO Storage.dbo.TimePurchase (Year, Month, Day, DatePurchase)

SELECT DISTINCT DATEPART(year, Date), DATEPART(month, Date), DATEPART(day, Date), Date

FROM Branch\_1.dbo.[Check]

WHERE Branch\_1.dbo.[Check].DateAdded > (select MAX(DateSynchron) from Storage.dbo.Synchronizes)

UNION ALL

SELECT DISTINCT DATEPART(year, Date), DATEPART(month, Date), DATEPART(day, Date), Date

FROM Branch\_2.dbo.[Check]

WHERE Branch\_2.dbo.[Check].DateAdded > (select MAX(DateSynchron) from Storage.dbo.Synchronizes)

END

INSERT INTO Synchronizes (DateSynchron) VALUES (getdate())

INSERT INTO [Storage].[dbo].[Storage] (

CustomerID, BranchID, ProductID, OrderDataWarehouseID, SynchronizeID, TimePurchaseID, Count, PriceFix, PricePurchase

)

SELECT

Cust.CustomerID,

1 as Branch,

Prod.ProductID,

(SELECT OrderDataWarehouseID FROM [Storage].[dbo].[Order] where DatePurchase=Ord.Date),

(SELECT MAX(SynchronizeID) FROM [Storage].[dbo].[Synchronizes])-1 as SynchronizeID,

(SELECT TimePurchaseID FROM [Storage].[dbo].[TimePurchase] where [Storage].[dbo].[TimePurchase].DatePurchase=Ord.Date),

Spec.Count,

Prod.Price,

Spec.Price

FROM

[Storage].[dbo].[TimePurchase] as TimePurchase,

[Storage].[dbo].[Synchronizes] as Syn,

[Branch\_1].[dbo].[Customer] as Cust

INNER JOIN [Branch\_1].[dbo].[Check] as Ord ON Cust.CustomerID=Ord.CustomerID

INNER JOIN [Branch\_1].[dbo].[Specific] as Spec ON Spec.[CheckID]=Ord.CheckID

INNER JOIN [Branch\_1].[dbo].[Product] as Prod ON Prod.ProductID=Spec.ProductID

WHERE

Ord.DateAdded > (select max(DateSynchron) from Synchronizes where SynchronizeID < (select MAX(SynchronizeID) from Synchronizes))

AND

Ord.CheckID IN (SELECT CheckID FROM [Branch\_1].[dbo].[Check])

INSERT INTO [Storage].[dbo].[Storage] (

CustomerID, BranchID, ProductID, OrderDataWarehouseID, SynchronizeID, TimePurchaseID, Count, PriceFix, PricePurchase

)

SELECT

Cust.CustomerID,

2 as Branch,

Prod.ProductID,

(SELECT OrderDataWarehouseID FROM [Storage].[dbo].[Order] where DatePurchase=Ord.Date),

(SELECT MAX(SynchronizeID) FROM [Storage].[dbo].[Synchronizes]) as SynchronizeID,

(SELECT TimePurchaseID FROM [Storage].[dbo].[TimePurchase] where [Storage].[dbo].[TimePurchase].DatePurchase=Ord.Date),

Spec.Count,

Prod.Price,

Spec.Price

FROM

[Storage].[dbo].[TimePurchase] as TimePurchase,

[Storage].[dbo].[Synchronizes] as Syn,

[Branch\_2].[dbo].[Customer] as Cust

INNER JOIN [Branch\_2].[dbo].[Check] as Ord ON Cust.CustomerID=Ord.CustomerID

INNER JOIN [Branch\_2].[dbo].[Specific] as Spec ON Spec.CheckID=Ord.CheckID

INNER JOIN [Branch\_2].[dbo].[Product] as Prod ON Prod.ProductID=Spec.ProductID

WHERE

Ord.DateAdded > (select max(DateSynchron) from Synchronizes where SynchronizeID < (select MAX(SynchronizeID) from Synchronizes))

AND Ord.CheckID IN (SELECT CheckID FROM [Branch\_2].[dbo].[Check])

END

GO

EXECUTE InsertFromBranche