Primero debemos crear un lote de facturaciÃ³n para que se realicen los comprobantes de CS con los precios actualizados.

Una vez creado el lote (en esta faq lo llamaremos 'lote\_refacturado'):

========================================================================================================================================================================

1) Comparar el lote de facturaciÃ³n previamente realizado (lo llamaremos 'lote\_original') con el nuevo lote de los comprobantes refacturados:

select i.id\_trans, i.item, c.id\_cliente, i.id\_cliente as Id\_cliente\_I, c.destinatario, c.id\_tipo\_cbte, (i2.precio\_grav-i.precio\_grav) as Diferencia

from cbtes\_items i, cbtes c, cbtes\_items i2, cbtes c2

where c.id\_trans = i.id\_trans

and i2.id\_trans = c2.id\_trans

and c.lote = 'lote\_original'

and c2.lote = 'lote\_refacturado'

and i.id\_producto = i2.id\_producto

and i.id\_cliente = i2.id\_cliente

and i.id\_producto = 'CS'

and c.id\_estado\_cbte <> 3

and c2.id\_estado\_cbte <> 3

========================================================================================================================================================================

2) Insertamos en una tabla temporal:

select i.id\_trans, i.item, c.id\_cliente, i.id\_cliente as Id\_cliente\_I, c.destinatario, c.id\_tipo\_cbte, (i2.precio\_grav-i.precio\_grav) as Diferencia

into #CbtesDiferencia

from cbtes\_items i, cbtes c, cbtes\_items i2, cbtes c2

where c.id\_trans = i.id\_trans

and i2.id\_trans = c2.id\_trans

and c.lote = 'lote\_original'

and c2.lote = 'lote\_refacturado'

and i.id\_producto = i2.id\_producto

and i.id\_cliente = i2.id\_cliente

and i.id\_producto = 'CS'

and c.id\_estado\_cbte <> 3

and c2.id\_estado\_cbte <> 3

select \* from #CbtesDiferencia

========================================================================================================================================================================

3) Creamos las nuevas transacciones e insertamos los valores en transacciones:

DECLARE @Id\_trans INT

DECLARE curNombre CURSOR FOR

select id\_trans + 150000 from #CbtesDiferencia

OPEN curNombre

FETCH NEXT FROM curNombre INTO @Id\_trans

WHILE @@FETCH\_STATUS = 0

BEGIN

INSERT INTO Transacciones(Id\_Trans, Descripcion) VALUES (@Id\_trans, 'lote\_refacturado')

FETCH NEXT FROM curNombre INTO @Id\_trans

END

CLOSE curNombre

DEALLOCATE curNombre

========================================================================================================================================================================

5) Insertamos los valores en cbtes\_items:

DECLARE @Id\_trans INT

DECLARE @id\_tipo\_Cbte CHAR(4)

DECLARE @Id\_Cliente INT

DECLARE @Imp\_Gravado DECIMAL(12,2)

DECLARE @Item smallint

DECLARE curNombre CURSOR FOR

SELECT Id\_trans + 150000, Item, Id\_Cliente, Diferencia FROM #CbtesDiferencia

OPEN curNombre

FETCH NEXT FROM curNombre INTO @Id\_trans,@Item,@Id\_Cliente,@Imp\_Gravado

WHILE @@FETCH\_STATUS = 0

BEGIN

insert into cbtes\_items (Id\_trans,Item,Id\_estado\_Cbte,Id\_Cliente,Fecha\_QA,Imp\_Gravado,Precio\_Grav,Imp\_Final,Lote)

values(@Id\_trans,@Item,1,@Id\_Cliente,'20211001',@Imp\_Gravado,@Imp\_Gravado,@Imp\_Gravado,'lote\_refacturado')

FETCH NEXT FROM curNombre INTO @Id\_trans,@Item,@Id\_Cliente,@Imp\_Gravado

END

CLOSE curNombre

DEALLOCATE curNombre

========================================================================================================================================================================

6) Insertamos Transacciones en Cbtes y sumamos el total del cbte:

DECLARE @Id\_trans INT

DECLARE @SUM DECIMAL (12,2)

DECLARE curNombre CURSOR FOR

select id\_trans from transacciones where descripcion = 'lote\_refacturado'

OPEN curNombre

FETCH NEXT FROM curNombre INTO @Id\_trans

WHILE @@FETCH\_STATUS = 0

BEGIN

insert into cbtes (Id\_trans,lote) Values (@Id\_trans,'lote\_refacturado')

SET @SUM = (select SUM(IMP\_FINAL) from cbtes\_ITEMS where LOTE='lote\_refacturado' AND ID\_TRANS=@Id\_trans)

Update CBTES set IMP\_TOTAL= @SUM, IMP\_EXENTO=@SUM WHERE ID\_TRANS=@Id\_trans

FETCH NEXT FROM curNombre INTO @Id\_trans

END

CLOSE curNombre

DEALLOCATE curNombre

========================================================================================================================================================================

7) Anulamos los de importe en negativo (estos luego deben ser chequeados):

SELECT IMP\_TOTAL,SALDO,\* FROM CBTES WHERE LOTE='lote\_refacturado' AND IMP\_TOTAL <0

UPDATE CBTES SET ID\_ESTADO\_CBTE=3 WHERE LOTE='lote\_refacturado' AND IMP\_TOTAL <0

========================================================================================================================================================================

8) Renumeramos los cbtes:

DECLARE @Id\_trans INT

DECLARE @SUM DECIMAL (12,2)

DECLARE curNombre CURSOR FOR

select id\_trans, id\_cliente from transacciones where cbtes\_items = 'lote\_refacturado'

OPEN curNombre

FETCH NEXT FROM curNombre INTO @Id\_trans, @Id\_Cliente

WHILE @@FETCH\_STATUS = 0

BEGIN

EXEC dbo.CPJ\_Cbtes\_Fac\_Fin '{"FlagUsarTrans":"0","ModoDebug":"1","Id\_Trans":"' + @Id\_trans + '","Flag\_Renumerar":"1"}'

FETCH NEXT FROM curNombre INTO @Id\_trans

END

CLOSE curNombre

DEALLOCATE curNombre

========================================================================================================================================================================

9) Insertamos en Clientes\_CtaCte:

DECLARE @Id\_trans INT

DECLARE @Imp\_total DECIMAL (12,2)

DECLARE @Id\_Cliente INT

DECLARE curNombre CURSOR FOR

select id\_trans, id\_cliente, imp\_total from cbtes where cbtes = 'lote\_refacturado'

OPEN curNombre

FETCH NEXT FROM curNombre INTO @Id\_trans, @Id\_Cliente, @ImpTotal

WHILE @@FETCH\_STATUS = 0

BEGIN

insert into Clientes\_CtaCte (Id\_trans, Nro\_Cuota, Id\_Empresa, Id\_Cliente, Fecha, Fecha\_Vence,

Importe, Id\_estado\_Cbte, Observacion, Id\_Trans\_origen,

Nro\_Cuota\_Origen, Id\_Moneda, Saldo, Nro\_FacCre)

Values (@Id\_trans,0,0,@Id\_Cliente,'20211001','20211001',@ImpTotal, 1, 'lote\_refacturado',0,0,'PES',@ImpTotal,0)

FETCH NEXT FROM curNombre INTO @Id\_trans

END

CLOSE curNombre

DEALLOCATE curNombre

========================================================================================================================================================================

10) Modificamos en cbtes:

DECLARE @Id\_trans INT

DECLARE @Id\_Cliente INT

DECLARE @Imp\_Total DECIMAL (12,2)

DECLARE curNombre\_\_ CURSOR FOR

SELECT Id\_trans, id\_cliente, imp\_total from cbtes where lote = 'lote\_refacturado'

OPEN curNombre\_\_

FETCH NEXT FROM curNombre\_\_ INTO @Id\_trans,@Id\_Cliente,@Imp\_Total

WHILE @@FETCH\_STATUS = 0

BEGIN

Update Cbtes set imp\_neto = @Imp\_Total,

 saldo = @Imp\_Total,

 id\_moneda = 'PES',

 destinatario = (select razon\_social from clientes where id\_cliente = @Id\_Cliente),

 cant\_cuotas = 1,

 fecha\_venc = '20211001',

 imp\_exento = @Imp\_Total,

 periodo\_iva = '20211001',

 id\_cond\_pago = 10,

 id\_lista = (select id\_lista from cbtes where lote = 'lote\_original' and id\_cliente = @Id\_Cliente),

 cotizacion = 1,

 id\_iva = 'CF'

 WHERE ID\_TRANS=@Id\_trans

FETCH NEXT FROM curNombre\_\_ INTO @Id\_trans,@Id\_Cliente,@Imp\_Total

END

CLOSE curNombre\_\_

DEALLOCATE curNombre\_\_

========================================================================================================================================================================

11) Por ultimo modificamos los asientos:

DECLARE @Id\_trans INT

DECLARE @Id\_Cliente INT

DECLARE @Imp\_Total DECIMAL (12,2)

DECLARE @json VARCHAR(8000)

DECLARE curNombre\_ CURSOR FOR

SELECT Id\_trans, id\_cliente, imp\_total from cbtes where lote = 'lote\_refacturado'

OPEN curNombre\_

FETCH NEXT FROM curNombre\_ INTO @Id\_trans,@Id\_Cliente,@Imp\_Total

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @json = '{'

 SET @json = @json + '"Id\_Trans":"' + CONVERT(VARCHAR(10), @Id\_Trans)+ '"'

 SET @json = @json + ',"FlagRegenerar":"1"'

 SET @json = @json + '}'

EXEC CPJ\_Cbtes\_Ctas\_Contables\_Generar @json

FETCH NEXT FROM curNombre\_ INTO @Id\_trans,@Id\_Cliente,@Imp\_Total

END

CLOSE curNombre\_

DEALLOCATE curNombre\_

========================================================================================================================================================================

Habiendo modificado todo, debemos realizar un SELECT final de todas las tablas que se modificaron:

Transacciones

Cbtes

Cbtes\_Items

Clientes\_CtaCte

Cbtes\_Ctas\_Contables

DECLARE @Id\_contrato INT

DECLARE @Id\_cliente INT

DECLARE @Id\_Trans INT

DECLARE curNombre CURSOR FOR

select c.id\_cliente, c.id\_trans from cbtes c

where c.lote = 'Diferencia\_Octubre\_O'

OPEN curNombre

FETCH NEXT FROM curNombre INTO @Id\_cliente, @Id\_Trans

WHILE @@FETCH\_STATUS = 0

BEGIN

update Cbtes set id\_contrato = (select id\_contrato from contratos where id\_Cliente = @Id\_cliente and id\_tipo\_con = 1 and activo = 1)

where lote = 'Diferencia\_Octubre\_O' and id\_trans = @Id\_Trans and id\_cliente = @Id\_cliente

FETCH NEXT FROM curNombre INTO @Id\_cliente, @Id\_Trans

END

CLOSE curNombre

DEALLOCATE curNombre