Subscription (sbsubs)

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[TBAI\_Sbsubs] Script Date: 2/8/2021 11:41:34 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

Create TRIGGER [dbo].[TBAI\_Subscription]

ON [dbo].[Subscription]

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

-- JDP 2018-03-21

UPDATE dbo.Subscription SET

CreatedDate = GetDate()

FROM dbo.Subscription o

INNER JOIN Inserted i ON o.id = i.id

SET NOCOUNT OFF;

END

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[tbd\_Sbsubs] Script Date: 2/8/2021 11:42:43 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

-- Create Delete Trigger

Create TRIGGER [dbo].[tbd\_Subscription] ON [dbo].[Subscription]

FOR DELETE AS

Begin

SET NOCOUNT ON

-- JDP 2018-03-21: insert into the DeleteLog

INSERT INTO [dbo].[DeleteLog]

([TableName]

,[TableId]

,[DeletedDate])

SELECT

'Subscription',

d.id,

GETDATE()

FROM deleted d

SET NOCOUNT OFF

END

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[TBD\_SBSUBS\_Callidus] Script Date: 2/8/2021 11:43:41 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER Trigger [dbo].[TBD\_Subscription\_Callidus] ON [dbo].[Subscription]

FOR DELETE AS

Begin

SET NOCOUNT ON

Declare @SubscriptionId int

Declare @Id int

declare @Temp table (id int identity, [key] int)

insert into @Temp

select d.id

from deleted d

While exists(

select 1

from @Temp) Begin

set @Id = (select top 1 ID

from @Temp)

select @SubscriptionId = [Key]

from @Temp

where ID = @ID

insert into SF.DeleteRecords

select 'CallidusSubscription', @SubscriptionId, null

delete

from @Temp

where ID = @ID

End

SET NOCOUNT OFF

End

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[TBD\_Sbsubs\_DataTransfer] Script Date: 2/8/2021 11:44:09 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create Trigger [dbo].[TBD\_Subscription\_DataTransfer] ON [dbo].[Subscription]

FOR DELETE AS

BEGIN

SET NOCOUNT ON

Delete from DataTransfer

Where

DataSourceName = 'Subscription' and

PrimaryId in (select id from deleted) and

TransferDate is null and

TransferSystem = 'Unica' and

IsInbound = 0

SET NOCOUNT OFF

END

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[TBI\_SBSUBS\_MaxSubnoCheck] Script Date: 2/8/2021 11:44:46 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create TRIGGER [dbo].[TBI\_Subscription\_MaxSubnoCheck] ON [dbo].[Subscription]

INSTEAD OF INSERT

AS

BEGIN

IF EXISTS (

SELECT 1 FROM inserted

WHERE convert(INT, SubscriptionNumber) - 20 > (SELECT max(convert(INT, SubscriptionNumber)) FROM Subscription

WHERE AccountId =inserted. AccountId)

)

BEGIN

DECLARE @msg VARCHAR(255) = 'Subscription Number is not in sequence.'

ROLLBACK

RAISERROR ('%s', 16, 1, @msg)

END

ELSE

BEGIN

-- Otherwise perform the insert

INSERT INTO dbo. Subscription

SELECT \* FROM inserted

END

END

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[TBIU\_SBSUBS\_Callidus] Script Date: 2/8/2021 11:45:14 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create Trigger [dbo].[TBIU\_Subscription\_Callidus] ON [dbo].[Subscription]

FOR INSERT, UPDATE AS

Begin

SET NOCOUNT ON

Declare @SubscriptionId int

Declare @Id int

declare @Temp table (id int identity, [key] int)

insert into @Temp

select i.id

from inserted i

While exists(

select 1

from @Temp) Begin

set @Id = (select top 1 ID

from @Temp)

select @SubscriptionId = [Key]

from @Temp

where ID = @ID

insert into SF.Integration

select 'CallidusSubscription', @SubscriptionId, null

delete

from @Temp

where ID = @ID

End

SET NOCOUNT OFF

End

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[TBIU\_Sbsubs\_SalesForce] Script Date: 2/8/2021 11:46:14 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create Trigger [dbo].[TBIU\_Subscription\_SalesForce] ON [dbo].[Subscription]

FOR INSERT, UPDATE AS

BEGIN

SET NOCOUNT ON

if UPDATE(ContactId) or

UPDATE(AccountId) or

UPDATE(ProductCode) or

UPDATE(OrderId) or

UPDATE(SubscriptionQuantity) or

UPDATE(SendRenewalNotices) or

UPDATE(WapNumber) or

UPDATE(OrderDetailId) or

UPDATE(StartDate) or

UPDATE([SubscriptionStatus]) or

UPDATE(SubscriptionNumber) or

UPDATE(id) or

UPDATE(Term) or

UPDATE(SubscriptionType) or

UPDATE(SendUpdatesOnly) or

UPDATE(TimesRenewed) or

UPDATE (SendUpdates) or

UPDATE (RenewalSubscriptionId)

Begin

Declare @SubscriptionID int

Declare @Id int

declare @Temp table (id int identity, [key] int)

insert into @Temp

select i.id

from inserted i

left outer join ARCUST arc (nolock)

on i. AccountId = arc. AccountId

left join SOMAST s (nolock)

on s.SOMKEY = i. OrderId

where arc.Code not in ('MGD')

--arc.Code not in ('I', 'DEL', 'MGD') ---removed condition account Code not in Inactive and Deleted --DE170338

--and i.startdate >= '2008-01-01' ---- SR11002027-Exclude OldSubscription

and

(

exists (select 1 from sf.OrderHeaderMapping (nolock) where PhoenixId = i.somkey)

or year(isnull(s.ORDATE,getdate())) >= year(getdate()) - 5

)

While exists(

select 1

from @Temp) Begin

set @Id = (select top 1 ID

from @Temp)

select @SubscriptionID = [Key]

from @Temp

where ID = @ID

insert into SF.Integration

select 'Subscription', @SubscriptionID, null

delete

from @Temp

where ID = @ID

End

End

SET NOCOUNT OFF

End

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[TBU\_SBSUBS] Script Date: 2/8/2021 11:46:40 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

-- Create ModifiedDate Trigger

ALTER Trigger [dbo].[TBU\_Subscription] ON [dbo].[Subscription]

FOR UPDATE AS

begin

SET NOCOUNT ON

IF NOT UPDATE(ModifiedDate)

BEGIN

UPDATE dbo.Subscription SET ModifiedDate = GetDate()

FROM dbo.Subscription o

INNER JOIN Inserted i ON o.id = i.id

END

IF UPDATE(ContactId)

BEGIN

INSERT INTO bb\_log ([user\_name], [change\_date], [table\_name], [column\_name], [old\_value], [key\_value])

SELECT Replace(System\_User, 'ms\', ''), GETDATE(),’Subscription’, ' ContactId ', d. ContactId, d.id FROM DELETED d inner join inserted i on d.id = i.id

WHERE isnull(i.ContactId,'') <> isnull(d.ContactId,'')

END

IF UPDATE(SendUpdatesOnly)

BEGIN

INSERT INTO bb\_log ([user\_name], [change\_date], [table\_name], [column\_name], [old\_value], [key\_value])

SELECT Replace(System\_User, 'ms\', ''), GETDATE(), ‘Subscription’, ' SendUpdatesOnly', d. SendUpdatesOnly', d.id FROM DELETED d inner join inserted i on d.id = i.id

WHERE isnull(i. SendUpdatesOnly','') <> isnull(d. SendUpdatesOnly','')

END

IF UPDATE(SendRenewalNotices)

BEGIN

INSERT INTO bb\_log ([user\_name], [change\_date], [table\_name], [column\_name], [old\_value], [key\_value])

SELECT Replace(System\_User, 'ms\', ''), GETDATE(), ' Subscription ', ' ‘SendRenewalNotices', d. SendRenewalNotices, d.id FROM DELETED d inner join inserted i on d.id = i.id

WHERE isnull(i. SendRenewalNotices,'') <> isnull(d. SendRenewalNotices,'')

END

IF UPDATE(SendUpdates)

BEGIN

INSERT INTO bb\_log ([user\_name], [change\_date], [table\_name], [column\_name], [old\_value], [key\_value])

SELECT Replace(System\_User, 'ms\', ''), GETDATE(), ' Subscription ', ' SendUpdates', d. SendUpdates, d.id FROM DELETED d inner join inserted i on d.id = i.id

WHERE isnull(i. SendUpdates,'') <> isnull(d. SendUpdates,'')

END

/\*

IF UPDATE(EUPDATE)

BEGIN

INSERT INTO bb\_log ([user\_name], [change\_date], [table\_name], [column\_name], [old\_value], [key\_value])

SELECT Replace(System\_User, 'ms\', ''), GETDATE(), 'SBSUBS', 'EUPDATE', d.EUPDATE, d.SUBKEY FROM DELETED d inner join inserted i on d.subkey = i.subkey

WHERE isnull(i.eupdate,'') <> isnull(d.eupdate,'')

END

\*/

IF UPDATE(SubscriptionStatus)

BEGIN

INSERT INTO bb\_log ([user\_name], [change\_date], [table\_name], [column\_name], [old\_value], [key\_value])

SELECT Replace(System\_User, 'ms\', ''), GETDATE(), ' Subscription ', ' SubscriptionStatus ', d. SubscriptionStatus, d.id FROM DELETED d inner join inserted i on d.id = i.id

WHERE isnull(i. SubscriptionStatus,'') <> isnull(d. SubscriptionStatus,'')

END

SET NOCOUNT OFF

end

USE [Phoenix]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[TBU\_SBSUBS\_CenveoOrderUpdate] Script Date: 2/8/2021 11:47:04 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

ALTER Trigger [dbo].[TBU\_Subscription\_CenveoOrderUpdate] ON [dbo].[Subscription]

FOR UPDATE AS

BEGIN

SET NOCOUNT ON

if update(SubscriptionStatus)

begin

delete PS

from Smartbatch2PreventSend PS

join inserted I

on PS.somkey = I. OrderId

join deleted D

on D.id = I.id

where

isnull(D. SubscriptionStatus, '') <> isnull(I. SubscriptionStatus, '')

end

SET NOCOUNT OFF

END

USE [Phoenix]

/\*\*\*\*\*\* Object: Trigger [dbo].[TBU\_SBSUBS\_DataTransfer] Script Date: 09/04/2014 11:20:25 \*\*\*\*\*\*/

ALTER Trigger [dbo].[TBU\_Subscription\_DataTransfer] ON [dbo].[SBSUBS]

FOR UPDATE AS

BEGIN

SET NOCOUNT ON

If update(ProductId) or

update(ContactId) or

update(AccountId) or

update(ProductCode) or

update(OrderId) or

update(SubscriptionQuantity) or

update(SendRenewalNotices) or

update(OrderDetailId) or

update(StartDate) or

update(SubscriptionStatus) or

update(SubscriptionNumber) or

update(id) or

update(Term) or

update(SubscriptionType) or

update(SendUpdatesOnly) or

update(TimesRenewed) or

update(RenewalSubscriptionId)

Begin

insert into DataTransfer (DataSourceName, PrimaryId, TransferDate, TransferSystem, IsInbound)

select 'Subscription', i.id, null, 'Unica', 0

from inserted i

inner join ARCUST arc (nolock) on i. AccountId = arc.ARCKEY

left join deleted d on d.id = i.id

inner join SOTRAN sot (nolock) on sot.subkey = i.id

where (arc.SLSVOL is null or arc.slsvol not in ('C', 'V', 'I'))

and arc.Code not in ('I', 'DEL', 'MGD')

and i.STARTDATE >= '2008-01-01'

and i.id not in

( select PrimaryId

from DataTransfer (nolock)

where DataSourceName = 'Subscription'

and PrimaryId = i.id

and TransferDate is null

and IsInbound = 0)

-- JDP 2013-11-13-13: If the status = 'V', make sure a Sales Order record has already been sent to Unica

---and (i.STATUS <> 'V' or i.SOMKEY in (select PrimaryId from DataTransfer where DataSourceName = 'SalesOrder'))

--Kvaralak 2015-10-20: Updated above to: If status = 'V', make sure a Subs record has already been sent to Unica

and (i.STATUS <> 'V' or i.id in (select PrimaryId from DataTransfer (nolock) where DataSourceName = 'Subscription'))

and

(

isnull(i.ProductId, 0) <> isnull(d.ProductId, 0) or

isnull(i.ContactId, 0) <> isnull(d.ContactId, 0) or

isnull(i.AccountId, 0) <> isnull(d.AccountId, 0) or

isnull(i.ProductCode, '') <> isnull(d.ProductCode, '') or

isnull(i.OrderId, 0) <> isnull(d.OrderId, 0) or

isnull(i.SubscriptionQuantity, 0) <> isnull(d.SubscriptionQuantity, 0) or

isnull(i.SendRenewalNotices, 0) <> isnull(d.SendRenewalNotices, 0) or

isnull(i.OrderDetailId, 0) <> isnull(d.OrderDetailId, 0) or

isnull(i.StartDate, '') <> isnull(d.StartDate, '') or

isnull(i.SubscriptionStatus, '') <> isnull(d.SubscriptionStatus, '') or

isnull(i.SubscriptionNumber, '') <> isnull(d.SubscriptionNumber, '') or

isnull(i.id, 0) <> isnull(d.id, 0) or

isnull(i.Term, 0) <> isnull(d.Term, 0) or

isnull(i.SubscriptionType, '') <> isnull(d.SubscriptionType, '') or

isnull(i.SendUpdatesOnly, 0) <> isnull(d.SendUpdatesOnly, 0) or

isnull(i.TimesRenewed, 0) <> isnull(d.TimesRenewed, 0) or

isnull(i.RenewalSubscriptionId, 0) <> isnull(d.RenewalSubscriptionId, 0)

)

End

SET NOCOUNT OFF

END

/\*\*\*\*\*\* Object: Trigger [dbo].[TBU\_SBSUBS\_WAP] Script Date: 04/20/2015 16:14:40 \*\*\*\*\*\*/

ALTER TRIGGER [dbo].[TBU\_Subscription\_WAP] ON [dbo].[Subscription]

FOR UPDATE

AS

BEGIN

SET NOCOUNT ON

/\*

DE168085 Phoenix web application returns

1. Added "or Update([SUBQTY]"

2, Trigger is updated to cancel subscription in wapsubstage table when updated sbsubs.status is C.

\*/

IF UPDATE ([SubscriptionStatus]) or Update([SubscriptionQuantity])

BEGIN

INSERT wapsubstage (

SUBKEY

,ARCKEY

,APPID

,SUBQTY

,STARTDATE

,STATUS

,ENDDATE

,SUBNO

,TYPE

,phnxstatus

,Source

,Misc

)

SELECT TOP 1 sbsubs.id

,CASE

WHEN wap\_reseller = 1

THEN sbsubs. ContactId \* - 1

ELSE sbsubs.AccountId

END AS ARCKEY

,wapsubstage.appid

,sbsubs. SubscriptionQuantity

,sbsubs.startdate

,1 AS STATUS

,wapsubstage.enddate

,sbsubs. SubscriptionNumber

,sbsubs. SubscriptionType

,sbsubs. SubscriptionStatus

,wapsubstage.source

,'' AS Misc

FROM Subscription

INNER JOIN arcust ON sbsubs. AccountId = arcust.arckey

INNER JOIN wapsubstage ON sbsubs.id = wapsubstage.subkey

AND wapsubstage.STATUS = 1

INNER JOIN inserted I ON sbsubs. AccountId = I.arckey

AND sbsubs.SubscriptionNumber = I.subno

AND sbsubs.SubscriptionStatus = 'A'

ORDER BY merged DESC

Insert into wapsubstage (Subkey, ARCKEY, APPID, SUBQTY, STARTDATE, STATUS, ENDDATE, SUBNO, TYPE, phnxstatus, Source, Misc, Somkey)

select top 1 Subkey, ARCKEY, APPID, SUBQTY, STARTDATE, STATUS = 0, ENDDATE, SUBNO, TYPE, 'C' as phnxstatus, Source, Misc, Somkey

from wapsubstage (nolock)

where wapsubstage.wsskey in

(

select max(wsskey)

from wapsubstage w (nolock)

where w.subkey in

(select sb.id

FROM Subscription sb (nolock)

INNER JOIN inserted I ON sb.id = I.id

WHERE I.[SubscriptionStatus] = 'C'

)

group by w.subkey

)

AND wapsubstage.STATUS = 1

END

SET NOCOUNT OFF

END