



```

TennCareAnalysis.HipFracture_OutpatientRecipientsProvidersIntervalPrevWithDiagPrim_PopulateWorkspace.02.sql
use HCN3D;
go
--All units in days--
declare @IntervalName as varchar(255) = 'Hip Fracture Interval Days Prior'
declare @IntervalLength as smallint = 30
declare @Period as smallint = 365 --Period / Interval Length = Number of Intervals
declare @PostIntervalEnd as smallint
declare @PostIntervalBegin as smallint
--Purge workspace table--
truncate table TennCareAnalysis.HipFracture_OutpatientRecipientsProvidersIntervalPrevWithDiagPrim_Workspace;
--Initialize intervals to start loop--
set @PostIntervalEnd = 0;
set @PostIntervalBegin = @PostIntervalEnd - @IntervalLength;
--Enter 1st loop to insert rows for current interval where providers appear--
while (@PostIntervalEnd <= @Period)
begin
--Set interval for current iteration of loop--
update dbo.Intervals
set IntervalBegin = @PostIntervalBegin
IntervalEnd = @PostIntervalEnd
where IntervalName = @IntervalName;
--Insert rows for current interval where providers appear. Uses view that reference Begin and End Intervals set in previous command--
insert into TennCareAnalysis.HipFracture_OutpatientRecipientsProvidersIntervalPrevWithDiagPrim_Workspace
(
SAK_RECIP
PROV_KEY
CDE_DIAG_PRIM
CountOfDups
IntervalName
IntervalBegin
IntervalEnd
Appears
)
select SAK_RECIP
PROV_KEY
CDE_DIAG_PRIM
CountOfDups
IntervalName
IntervalBegin
IntervalEnd
Appears
from TennCare.HipFracture_OutpatientRecipientsProvidersIntervalWithInfoPrevWithDiagPrim;
--Increment variables for next iteration--
set @PostIntervalBegin = @PostIntervalEnd;
set @PostIntervalEnd = @PostIntervalEnd + @IntervalLength;
--Check for first interval when starting at day "0"--
if @PostIntervalEnd = 0 -- Check for first interval when starting at day "0"
--NOTE: Has to be performed in a second loop to include providers that appear in a later intervals/
as occurs with post event.
else
set @PostIntervalEnd = @PostIntervalEnd + @IntervalLength;
end;
--Reinitialize intervals to restart loop--
set @PostIntervalEnd = @PostIntervalEnd - @IntervalLength;
set @PostIntervalBegin = @PostIntervalEnd - @IntervalLength;
--Enter 2nd loop to insert rows for current interval where providers DO NOT appear--
--NOTE: Has to be performed in a second loop to include providers that appear in a later intervals/
while (@PostIntervalBegin <= @Period)
begin
--Set interval for current iteration of loop--
update dbo.Intervals
set IntervalBegin = @PostIntervalBegin
IntervalEnd = @PostIntervalEnd
where IntervalName = @IntervalName;
--Insert rows using MERGE with all Recipient/Provider pairs appearing across entire workspace as source--
print 'Inserting rows where providers DO NOT appear: Interval Begin/End ' + convert(varchar(255), @PostIntervalBegin) + ' / ' + convert(varchar(255), @PostIntervalEnd);
merge into HCN3D.TennCareAnalysis.HipFracture_OutpatientRecipientsProvidersIntervalPrevWithDiagPrim_Workspace as Destination
using
(
select SAK_RECIP
PROV_KEY
CDE_DIAG_PRIM
CountOfDups
IntervalName
IntervalBegin
IntervalEnd
Appears
) as Source
on (
Source.SAK_RECIP = Destination.SAK_RECIP
and Source.PROV_KEY = Destination.PROV_KEY
and Source.CDE_DIAG_PRIM = Destination.CDE_DIAG_PRIM
and Source.CountOfDups = Destination.CountOfDups
and Source.IntervalName = Destination.IntervalName
and Source.IntervalBegin = Destination.IntervalBegin
and Source.IntervalEnd = Destination.IntervalEnd
)
when not matched then
insert
(
SAK_RECIP
PROV_KEY
CDE_DIAG_PRIM
CountOfDups
IntervalName
IntervalBegin
IntervalEnd
Appears
)
values
(
Source.SAK_RECIP
Source.PROV_KEY
Source.CDE_DIAG_PRIM
Source.CountOfDups
Source.IntervalName
Source.IntervalBegin
Source.IntervalEnd
Source.Appears
);
--Increment variables for next iteration--
set @PostIntervalBegin = @PostIntervalBegin + @IntervalLength;
if @PostIntervalBegin = 0 -- Check for first interval when starting at day "0"
--NOTE: Has to be performed in a second loop to include providers that appear in a later intervals/
as occurs with post event.
else
set @PostIntervalBegin = @PostIntervalBegin + @IntervalLength;
end;

```

```

TennCareAnalysis.HipFracture_OutpatientRecipientsProvidersIntervalPostWithDiagPrim_PopulateWorkspace.02.sql
use HCN3D;
go
--All units in days--
declare @IntervalName as varchar(255) = 'Hip Fracture Interval Days Post'
declare @IntervalLength as smallint = 30
declare @Period as smallint = 365 --Period / Interval Length = Number of Intervals
declare @PostIntervalEnd as smallint
declare @PostIntervalBegin as smallint
--Purge workspace table--
truncate table TennCareAnalysis.HipFracture_OutpatientRecipientsProvidersIntervalPostWithDiagPrim_Workspace; --EDIT MADE: New table name.
--Initialize intervals to start loop--
set @PostIntervalEnd = @IntervalLength;
set @PostIntervalBegin = @PostIntervalEnd - @IntervalLength;
--Enter 1st loop to insert rows for current interval where providers appear--
while (@PostIntervalEnd <= @Period)
begin
--Set interval for current iteration of loop--
update dbo.Intervals
set IntervalBegin = @PostIntervalBegin
IntervalEnd = @PostIntervalEnd
where IntervalName = @IntervalName;
--Insert rows for current interval where providers appear. Uses view that reference Begin and End Intervals set in previous command--
print 'Inserting rows where providers appear: Interval Begin/End ' + convert(varchar(255), @PostIntervalBegin) + ' / ' + convert(varchar(255), @PostIntervalEnd);
insert into TennCareAnalysis.HipFracture_OutpatientRecipientsProvidersIntervalPostWithDiagPrim_Workspace --EDIT MADE: New table name.
(
SAK_RECIP
PROV_KEY
CDE_DIAG_PRIM
CountOfDups
IntervalName
IntervalBegin
IntervalEnd
Appears
)
select SAK_RECIP
PROV_KEY
CDE_DIAG_PRIM
CountOfDups
IntervalName
IntervalBegin
IntervalEnd
Appears
from TennCare.HipFracture_OutpatientRecipientsProvidersIntervalWithInfoPostWithDiagPrim; --EDIT: New table name.
--Increment variables for next iteration--
set @PostIntervalBegin = @PostIntervalBegin + @IntervalLength;
set @PostIntervalEnd = @PostIntervalEnd + @IntervalLength;
end;
--Reinitialize intervals to restart loop--
set @PostIntervalEnd = @PostIntervalEnd - @IntervalLength;
set @PostIntervalBegin = @PostIntervalEnd - @IntervalLength;
--Enter 2nd loop to insert rows for current interval where providers DO NOT appear--
--NOTE: Has to be performed in a second loop to include providers that appear in a later intervals/
while (@PostIntervalBegin <= @Period)
begin
--Set interval for current iteration of loop--
update dbo.Intervals
set IntervalBegin = @PostIntervalBegin
IntervalEnd = @PostIntervalEnd
where IntervalName = @IntervalName;
--Insert rows using MERGE with all Recipient/Provider pairs appearing across entire workspace as source--
print 'Inserting rows where providers DO NOT appear: Interval Begin/End ' + convert(varchar(255), @PostIntervalBegin) + ' / ' + convert(varchar(255), @PostIntervalEnd);
merge into HCN3D.TennCareAnalysis.HipFracture_OutpatientRecipientsProvidersIntervalPostWithDiagPrim_Workspace as Destination --EDIT MADE: New table name.
using
(
select SAK_RECIP
PROV_KEY
CDE_DIAG_PRIM
CountOfDups
IntervalName
IntervalBegin
IntervalEnd
Appears
) as Source
on (
Source.SAK_RECIP = Destination.SAK_RECIP
and Source.PROV_KEY = Destination.PROV_KEY
and Source.CDE_DIAG_PRIM = Destination.CDE_DIAG_PRIM
and Source.CountOfDups = Destination.CountOfDups
and Source.IntervalName = Destination.IntervalName
and Source.IntervalBegin = Destination.IntervalBegin
and Source.IntervalEnd = Destination.IntervalEnd
)
when not matched then
insert
(
SAK_RECIP
PROV_KEY
CDE_DIAG_PRIM
CountOfDups
IntervalName
IntervalBegin
IntervalEnd
Appears
)
values
(
Source.SAK_RECIP
Source.PROV_KEY
Source.CDE_DIAG_PRIM
Source.CountOfDups
Source.IntervalName
Source.IntervalBegin
Source.IntervalEnd
Source.Appears
);
--Increment variables for next iteration--
set @PostIntervalBegin = @PostIntervalBegin + @IntervalLength;
set @PostIntervalEnd = @PostIntervalEnd + @IntervalLength;
end;

```

