**Blackbaud SQL**

**Small Snippets:**

**Table Schema**

select '[' + CAST([COLUMN\_NAME] AS VARCHAR) + ']' + ' varchar(max),' from SOURCE\_SCHEMA.[INFORMATION\_SCHEMA].COLUMNS

where [TABLE\_NAME] = 'TableName'

**Dates**

* CASE when ISDATE([SRC\_DATE]) = 1 then Convert(varchar(10), cast([SRC\_DATE] as date), 101) END as [TRG\_DATE]

**Only add column if it doesn’t exist already**

If Not Exists (Select \* from <Src>.[Information\_Schema].[Columns]

Where Table\_Name = 'SRC\_TABLE' and column\_name = 'ROWNUM')

Begin

ALTER TABLE

[Src].[SRC\_TABLE]

ADD

ROWNUM INT IDENTITY

End

**Adding a max value for an ID to every entry for that ID**

SELECT [EMPLID]

,[ACAD\_PLAN\_DESCR]

,[STDNT\_DEGR]

,[DEGREE]

,[COMPLETION\_TERM]

, (Select max(cast(stdnt\_degr as int)) from src.DEGREES A where a.emplid = degrees.emplid) 'MaxStudDeg'

FROM [TRAINING\_RIC\_Src].[Src].[DEGREES]

ORDER BY [EMPLID]

Using that max to set something as true

SELECT [EMPLID]

,[ACAD\_PLAN\_DESCR]

,[STDNT\_DEGR]

,[DEGREE]

,[COMPLETION\_TERM]

, (Select MAX(CAST([STDNT\_DEGR] as INT)) FROM src.DEGREES dg WHERE dg.[EMPLID] = d.[EMPLID]) as [MaxStudDeg]

, **CASE WHEN (**Select MAX(CAST([STDNT\_DEGR] as INT)) FROM src.DEGREES dg WHERE dg.[EMPLID] = d.[EMPLID]**) = [STDNT\_DEGR]**

THEN 'True' ELSE 'False' END as [ESRPrimAlum]

FROM [TRAINING\_RIC\_Src].[Src].[DEGREES] d

ORDER BY [EMPLID]

Using an UPDATE statement to set a max value as true, when the max value is what distinguishes the ImpID for that record

UPDATE [TRAINING\_RIC\_Trg].[dbo].[Organization\_Relationship]

SET [ORIsprimary] = 'True'

-- SELECT [temp1].[ImportID], [temp1].[ORImpID], org.[ORIsprimary]

FROM (SELECT [ImportID], MAX([ORImpID]) AS [ORImpID]

FROM [TRAINING\_RIC\_Trg].[dbo].[Organization\_Relationship]

GROUP BY [ImportID]) AS [temp1]

INNER JOIN [TRAINING\_RIC\_Trg].[dbo].[Organization\_Relationship] org

ON [temp1].[ORImpID] = org.[ORImpID]

Or using an update statement when the ImpID does not have that max value appended to it

UPDATE [TRAINING\_RIC\_Trg].[dbo].[Organization\_Relationship]

SET [ORIsprimary] = 'True'

-- SELECT [temp1].[ImportID], [temp1].[SEQUENCE\_NBR], org.[ORImpID], org.[ORIsprimary]

FROM (SELECT [ImportID], MAX([NOTES]) AS [SEQUENCE\_NBR]

FROM [TRAINING\_RIC\_Trg].[dbo].[Organization\_Relationship]

GROUP BY [ImportID]) AS [temp1]

INNER JOIN [TRAINING\_RIC\_Trg].[dbo].[Organization\_Relationship] org

ON [temp1].[ImportID] = org.[ImportID]

AND [temp1].[SEQUENCE\_NBR] = org.[NOTES]

**Looping: Made by Peter for Derek**

-- create temp table for looping

-- need identity to iterate through the table

declare @list table

(

PKID int identity

-- column below is from Derek's data

, WRN\_KEY\_NUMBER varchar(20)

)

-- populate temp table to loop

insert into @list

(

-- data from Derek's table, no need to insert identity column, it self populates

-- this should be the identifier for each row you want to loop through

-- maybe obtained from a select distinct or some such

wrn\_key\_number

)

-- Derek's table/data

SELECT Distinct [WRN\_KEY\_NUMBER] FROM [mun\_48647\_src].[viking].[dvdbwrnfil]

-- declare looping variables

declare @Loop int

declare @max int

-- declare variable to hold data from the temporary table

-- can use this variable to help select rows from another table in your data

-- this might be used to hold an ImportID or other such data

to identify specific rows

declare @vid as varchar(20)

-- initialize the loop

select @loop = 0, @max = coalesce(max(pkid), 0)

from @list

-- set loop condition, so loop knows when to stop

while @Loop<@max

-- begin looping

begin

-- store your data that's in the temp table in a variable, move the loop (first round from 0 to 1)

-- where clause ensures you're looking at one row at a time (PKID is the identity column)

select @vid = wrn\_key\_number, @Loop = @loop+1 from @list where pkid = @Loop+1

-- declare variable to hold data from another table

-- store your desired data in the variable you just created

-- can use other variables like @vid to aid in the SELECT to ensure you're getting the right row

select @s = isNull(@s, '') + '| ' + isNull(rtrim(T.[WRN\_TYPE\_OF\_WARNING]), '')

from [mun\_48647\_src].[viking].[dvdbwrnfil] t1

Join [MUN\_48647\_Translation].[dbo].['TYPE OF WARNING-CODE VALUES$'] T

on t1.[WRN\_TYPE\_OF\_WARNING] = T.[WRN\_TYPE\_OF\_WARNING]

where WRN\_KEY\_NUMBER = @vid

-- now that you've selected data and stored in a variable, use that to do something

-- again, you can use other variables to aid in your doing of something

UPDATE C

SET C.[Annotation] = @S

FROM [MUN\_48647\_TRG].[dbo].[Constituent] C

--Join [MUN\_48647\_Translation].[dbo].['TYPE OF WARNING-CODE VALUES$'] T

-- on WRN.[WRN\_TYPE\_OF\_WARNING] = T.[WRN\_TYPE\_OF\_WARNING]

WHERE C.[ImportID] = @VID

-- stop the madness

end

**Change Table Names**

if coalesce(OBJECT\_ID('bannertables'), 0) > 0

begin

drop table dbo.bannertables

end

create table dbo.bannertables (tablename varchar(1000), tschema varchar(1000), id int identity)

insert into dbo.bannertables

--select distinct TABLE\_NAME, TABLE\_SCHEMA from INFORMATION\_SCHEMA.COLUMNS where TABLE\_NAME like 'JSCC\_%' or TABLE\_NAME like '%Empty%'

select distinct TABLE\_NAME, TABLE\_SCHEMA from INFORMATION\_SCHEMA.COLUMNS where TABLE\_NAME like 'ALUMNI\_%' or TABLE\_NAME like 'GENERAL\_%' or TABLE\_NAME like 'SATURN\_%'

declare @startid int

declare @endid int

declare @SQL varchar(4000)

declare @tablename varchar(1000)

--declare @columnname varchar(1000)

declare @tschema varchar(1000)

declare @newtablename varchar(1000)

set @startid = (select min(id) from dbo.bannertables)

set @endid = (select MAX(id) from dbo.bannertables)

while @startid<=@endid

begin

set @tablename = (select tablename from dbo.bannertables where id = @startid)

--set @columnname = (select column\_name from dbo.bannertables where id = @startid)

set @tschema = (select tschema from dbo.bannertables where id = @startid)

--set @newtablename = replace(REPLACE(@tablename,'JSCC\_',''),'- Empty','')

set @newtablename = replace(replace(REPLACE(@tablename,'ALUMNI\_',''),'GENERAL\_',''),'SATURN\_','')

set @SQL = 'exec sp\_rename '''+'['+@tschema+ '].'+@tablename+ ''','''+ @newtablename+''''

--set @SQL = 'exec sp\_rename '''+'['+@tschema+ '].'+@tablename+'.'+'['+@columnname+']'', '+ @newcolumnname+ ',''Column'''

print @sql

exec(@sql)

set @startid= @startid+1

end

**Remove Quotes**

if coalesce(OBJECT\_ID('bannercolumns'), 0) > 0

begin

drop table dbo.bannercolumns

end

create table dbo.bannercolumns (tablename varchar(1000), column\_name varchar(1000), tschema varchar(1000), id int identity)

insert into dbo.bannercolumns

select distinct TABLE\_NAME, COLUMN\_NAME, TABLE\_SCHEMA from INFORMATION\_SCHEMA.COLUMNS --where COLUMN\_NAME like '% %'

declare @startid int

declare @endid int

declare @SQL varchar(4000)

declare @tablename varchar(1000)

declare @columnname varchar(1000)

declare @tschema varchar(1000)

declare @newcolumnname varchar(1000)

set @startid = (select min(id) from dbo.bannercolumns)

set @endid = (select MAX(id) from dbo.bannercolumns)

while @startid<=@endid

begin

set @tablename = (select tablename from dbo.bannercolumns where id = @startid)

set @columnname = (select column\_name from dbo.bannercolumns where id = @startid)

set @tschema = (select tschema from dbo.bannercolumns where id = @startid)

set @newcolumnname = REPLACE(@columnname,' ','\_')

--set @SQL = 'exec sp\_rename '''+'['+@tschema+ '].'+@tablename+'.'+'['+@columnname+']'', '+ @newcolumnname+ ',''Column'''

set @sql = 'update Src.[' + @tableName + ']

set [' + @columnname + '] = replace([' + @columnname + '], ''"'', '''')

where charIndex(''"'', [' + @columnname + ']) > 0 '

print @sql

exec(@sql)

set @startid= @startid+1

end

**Change Table Schema**

if coalesce(OBJECT\_ID('temptables'), 0) > 0

begin

drop table dbo.temptables

end

create table dbo.temptables (tablename varchar(1000), tschema varchar(1000), id int identity)

insert into dbo.temptables

select distinct TABLE\_NAME, TABLE\_SCHEMA from INFORMATION\_SCHEMA.COLUMNS where TABLE\_SCHEMA = 'mpx\_dbo'

declare @startid int

declare @endid int

declare @SQL varchar(4000)

declare @tablename varchar(1000)

set @startid = (select min(id) from dbo.temptables)

set @endid = (select MAX(id) from dbo.temptables)

while @startid<=@endid

begin

set @tablename = (select tablename from dbo.temptables where id = @startid)

set @SQL = 'ALTER SCHEMA Src TRANSFER mpx\_dbo.' + @tablename

print @sql

exec(@sql)

set @startid= @startid+1

end

**Replace Null Text**

if exists

(select \* from MBU\_40671\_Src.INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'mbucolumns')

begin

drop table dbo.mbucolumns

end

create table dbo.mbucolumns (tablename varchar(1000), column\_name varchar(1000), tschema varchar(1000), id int identity)

insert into dbo.mbucolumns

select distinct TABLE\_NAME, COLUMN\_NAME, TABLE\_SCHEMA from INFORMATION\_SCHEMA.COLUMNS

declare @startid int

declare @endid int

declare @SQL varchar(4000)

declare @tablename varchar(1000)

declare @columnname varchar(1000)

declare @tschema varchar(1000)

declare @newcolumnname varchar(1000)

set @startid = (select min(id) from dbo.mbucolumns)

set @endid = (select MAX(id) from dbo.mbucolumns)

while @startid<=@endid

begin

set @tablename = (select tablename from dbo.mbucolumns where id = @startid)

set @columnname = (select column\_name from dbo.mbucolumns where id = @startid)

set @tschema = (select tschema from dbo.mbucolumns where id = @startid)

set @newcolumnname = REPLACE(@columnname,' ','\_')

--set @SQL = 'exec sp\_rename '''+'['+@tschema+ '].'+@tablename+'.'+'['+@columnname+']'', '+ @newcolumnname+ ',''Column'''

set @sql = 'update Src.[' + @tableName + ']

set [' + @columnname + '] = replace([' + @columnname + '], ''NULL'', NULL)

where charIndex(''NULL'', [' + @columnname + ']) > 0 '

print @sql

exec(@sql)

set @startid= @startid+1

end