**The process for updating the Athena Data View in the PP\_Athena\_ODS database. The update process involves three main scenarios:**

**New View (Scenario: 1):**Adding new tables to the Data View**.**

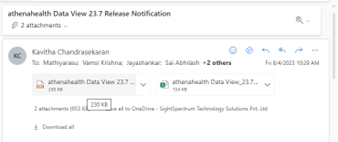
**New Columns to be Added (Scenario: 2):**Updating tables with new columns**.**

**Data Update (Scenario: 3):**Refreshing data in existing tables**.**

**Step 1: Login to Snowflake**

* **URL:** [Snowflake Console](https://xaa19981.snowflakecomputing.com/console#/internal/worksheet)
* **Credentials:**
  + **Username (UN):** edevaraj
  + **Password (PW):** Prim@ry01!

**Step 2: Get the List of All Data View Tables Changes**



**Scenario 1: New View**

**Check if all tables are present in Source and if the Same Table is present in a Different Schema**

* Execute the following query to check if all tables from the list of new tables to be added exist in the source
* Mark missing tables and capture them in a sheet. Remove missing tables from the list
* Run SQL to check if tables are present and mark missing tables.
* Execute a union query to identify tables present in different schemas.
* Add missing tables to the New View scenario.

1. **List of New Tables to be ADDED:**
   * **Run SQL to check if tables are present and mark missing tables.**

select  TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES

Where TABLE\_NAME in (< **List of Tables to be Data Refreshed** >)

\*\*\***Mark Missing tables capture them in a sheet and remove it from the List of Tables to be Data Refreshed**

* + **Execute a union query to identify tables present in different schemas.**

select TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES

Where TABLE\_NAME in (< List of Tables to be Data Refreshed >)

group by TABLE\_NAME

having count(TABLE\_SCHEMA) > 1

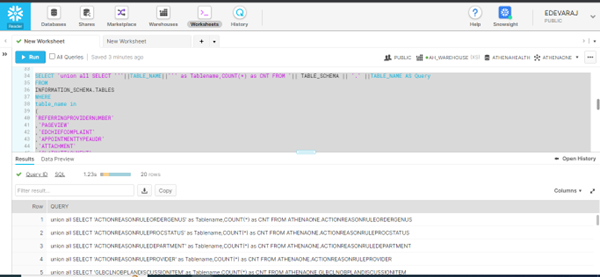
* + **Add missing tables to New View scenario.**

SELECT 'union all SELECT '''||TABLE\_NAME||''' as Tablename , COUNT(\*) as CNT FROM '|| TABLE\_SCHEMA || '.' ||TABLE\_NAME AS Query

FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME in

( < List of Tables to be Data Refreshed > )

1. **Execute Snowflake Query to Identify New Tables:**
2. **Capture Missing Tables:**
   * If any tables are missing, add them to the New View scenario.
3. **Execute Query in Snowflake:**
   * Execute the Snowflake query to identify new tables and copy the result.



1. **Paste Copied Query:**
   * Paste the copied query into a subquery, removing unnecessary text.

select listagg(''''||Tablename||'''',',')    within group(order by Tablename) as CSVColumnlist

 ----- select Tablename

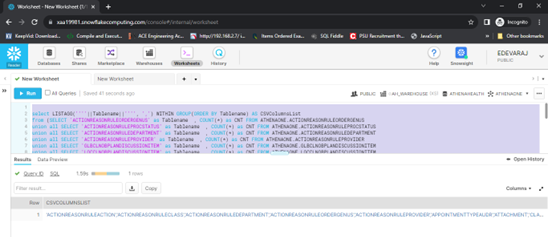
 from (<**Copied Query**>) abc where CNT > 0

select Tablename

from (**<Copied Query>**) abc where CNT = 0

1. **Final DataView Tables to be ADDED:**
   * The result is the final list of tables to be added in ODS.

select \* from INFORMATION\_SCHEMA.TABLES where TABLE\_NAME in (< **List of New Tables to be ADDED** >)



**Add Tables List and Run on Snowflake:**

* + Run the query on Snowflake to add the new tables.

SELECT 'union all select '||''''||table\_schema||''''||' as table\_schema,'

||''''||table\_name||''''||' as table\_name,'||''''||Column\_name||''''||' as Column\_name, ' ||

'max(len('||Column\_name||')) MaxLen from '||table\_schema||'.'||table\_name  as Query

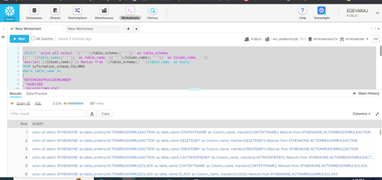
FROM information\_schema.COLUMNS

where table\_name in

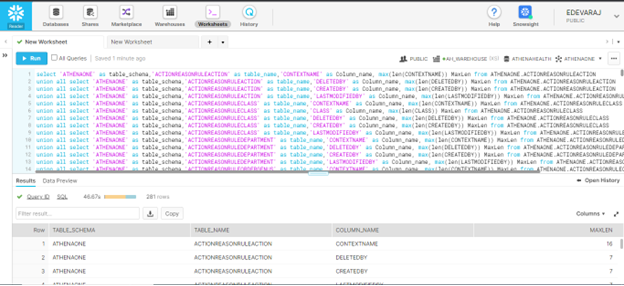
(**< List of New Tables to be ADDED >** )

and DATA\_TYPE='TEXT' and CHARACTER\_MAXIMUM\_LENGTH='16777216'

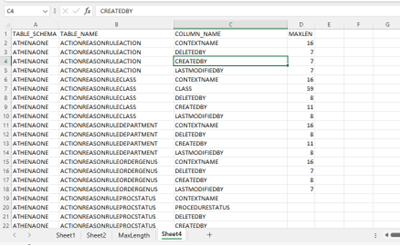
order by table\_name,ordinal\_position



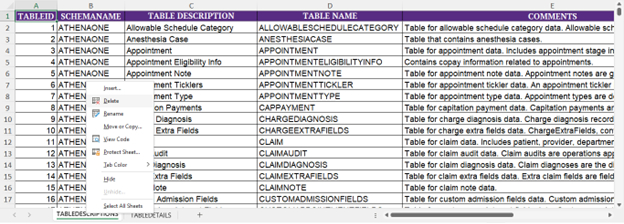
1. **Execute Query for Varchar Columns Max Length:**
   * Execute a query to determine varchar columns max length and copy the result.

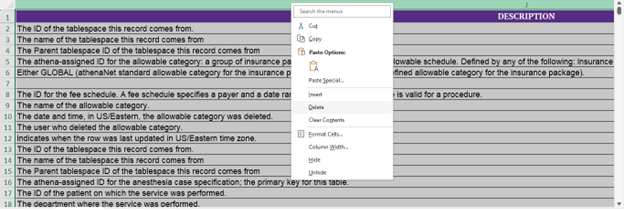


1. **Paste in Notepad and Excel:**
   * Paste the result into Notepad and then into Excel for Athena Team sharing.



1. **Open Athena Data View Excel:**
   * Remove unnecessary sheets and columns.
   * Add a Serial Number column to the TableDetails sheet. Add Value 1 in A2 Cell and Place =A2+1 in A3 Cell
   * Drag the formula to end of sheet .
   * Remove columns (I) Descriptions and (J) ForeignKey ,Save the sheet



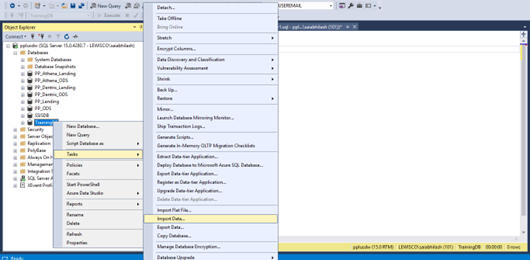


1. **Connect to Database (PP\_Athena\_Landing):**
   * Import Excel data into PP\_Athena\_Landing.

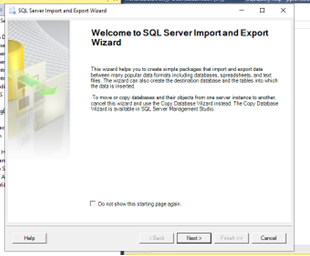
Truncate table [dbo].[AthenaDataView]

1. **Truncate AthenaDataView Table:**
   * Truncate the AthenaDataView table in PP\_Athena\_Landing.
2. **Import Athena Data View Excel Data:**
   * Import data from Athena Data View Excel into PP\_Athena\_Landing.
3. (Here for Example I have taken Training DB)

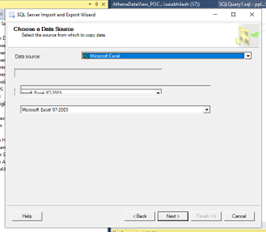
TraningDB 🡪 Tasks 🡪 Select ‘’import data’’



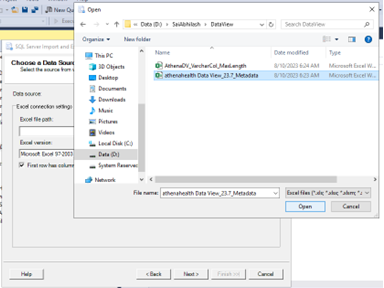
The SQL Server Import and Export dialogue box open and **Click Next>**

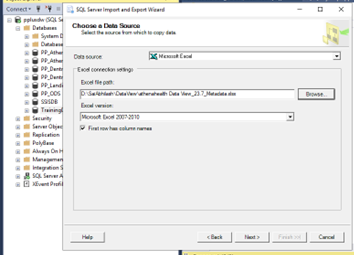


Select the Source Path and **Click Next>**

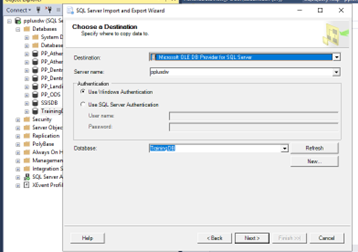


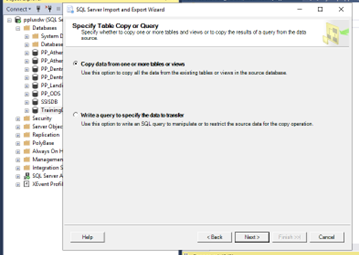
 Open the file which should already copied from source and **Click Next>**

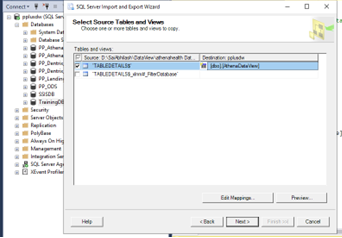




Select the Destination (Database) or server authentication and **Click Next>**



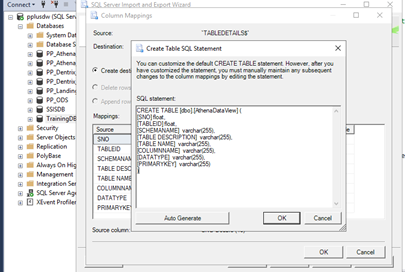


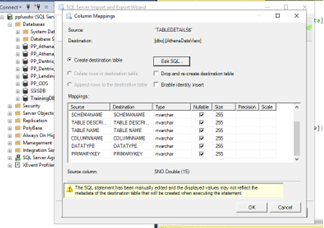


Replace Destination table name as **[dbo].[AthenaDataView]**

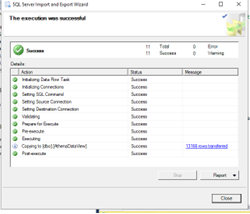
Select Edit Mapping à Edit SQL

Replace nvarchar with varchar in column datatypes





Ok à Next àNext àFinish



select \* FROM  [dbo].[AthenaDataView]

**-----Alter table and add new column**

alter table [dbo].[AthenaDataView]

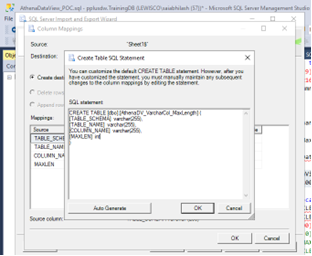
add VarcharDataType varchar(100)

alter table [dbo].[AthenaDataView]

add TableType varchar(100)

**Import AthenaDV\_VarcharCol\_MaxLength.xls sheet into**

Table : **AthenaDV\_VarcharCol\_MaxLength**



select \* from  AthenaDV\_VarcharCol\_MaxLength

**USE PP\_Athena\_Landing**

----------Delete records other than list of tables

delete a

--select \*

from [dbo].[AthenaDataView] a

where [TABLE NAME] not in

(

(<**List of New Tables to be ADDED** >)

**---------Delete duplicate tables who are present in two schemas**

delete a

--select \*

from [dbo].[AthenaDataView] a

where [TABLE NAME]  in

('APPOINTMENT') and SCHEMANAME='SCHEDULING'

-----Mark Insert tables

update a set TableType='Insert'

--select \*

from

[dbo].[AthenaDataView] a

where [TABLE NAME] in

(<**List of New Tables to be ADDED** >)

--------The count should be equal to List of tables count

select distinct [TABLE NAME]

from [dbo].[AthenaDataView] a where TableType='Insert'

**--------Count should be same as total tables list (all tables should have**

**Last updated date column)**

select distinct [TABLE NAME]

from

[dbo].[AthenaDataView] where COLUMNNAME='LASTUPDATED'

And TableType='Insert'

**-------Assign Varchar(Max) Columns Varchar length**

update a set VarcharDataType=case when isnull(l.MAXLEN,0) < 20 then 'varchar(200)'

            when isnull(l.MAXLEN,0) < 100 then 'varchar(500)'

when isnull(l.MAXLEN,0) < 500 then 'varchar(2000)'

else 'varchar(8000)' end

--select l.\*,isnull(l.MAXLEN,0),case when isnull(l.MAXLEN,0) < 20 then 'varchar(200)'

--            when isnull(l.MAXLEN,0) < 100 then 'varchar(500)'

-- when isnull(l.MAXLEN,0) < 500 then 'varchar(2000)'

-- else 'varchar(8000)' end

from

[dbo].[AthenaDataView] a

join [dbo].AthenaDV\_VarcharCol\_MaxLength l

on a.SCHEMANAME =l.TABLE\_SCHEMA and a.[TABLE NAME]=l.TABLE\_NAME

and a.COLUMNNAME=l.COLUMN\_NAME

where a.DATATYPE='VARCHAR(16777216)'

-----**No records should come in all 3 queries,If any record comes stop the process**

select \* from [dbo].[AthenaDataView] a

where a.DATATYPE='VARCHAR(16777216)'

and VarcharDataType is null

SELECT \*

FROM  [dbo].[AthenaDataView]

WHERE

(LEN([table name]) != DATALENGTH([table name]))

or

(LEN(COLUMNNAME) != DATALENGTH(COLUMNNAME))

or

(LEN(DATATYPE) != DATALENGTH(DATATYPE))

or

(LEN(PRIMARYKEY) != DATALENGTH(PRIMARYKEY))

SELECT distinct DATATYPE

FROM  [dbo].[AthenaDataView]

where DATATYPE not in ('BOOLEAN','TIMESTAMP\_NTZ(9)','VARCHAR(16777216)','DATE')

and DATATYPE not like '%NUMBER%'

order by DATATYPE

**------check for duplicate column details**

with cte as

(

select \*,

ROW\_NUMBER() over(partition by SCHEMANAME,[TABLE NAME],COLUMNNAME order by sno asc)  Rnk

from [dbo].[AthenaDataView]

)

----delete

--select \*

from cte

where Rnk > 1

**/\*Create table script for PP\_Athena\_Landing\*/**

**Use PP\_Athena\_Landing**

select

'GO'+char(10)+' Create table '+SCHEMANAME+'.'+[TABLE NAME]+' ( '+ColumnList+',Record\_Created\_Date datetime '+' ) '

from

(

select SCHEMANAME,[TABLE NAME]

,STRING\_AGG(DerivedColumnName,',')  WITHIN GROUP ( ORDER BY sno ASC) ColumnList

FROM

(

select \*  ,  COLUMNNAME+'   '+

case when DATATYPE='BOOLEAN' then 'bit'

when DATATYPE like '%NUMBER%' then replace(DATATYPE,'NUMBER','numeric')

when DATATYPE='TIMESTAMP\_NTZ(9)' then '[datetime2](7)'

when DATATYPE='VARCHAR(16777216)' then 'varchar(max)'

else DATATYPE end DerivedColumnName

FROM  [dbo].[AthenaDataView]

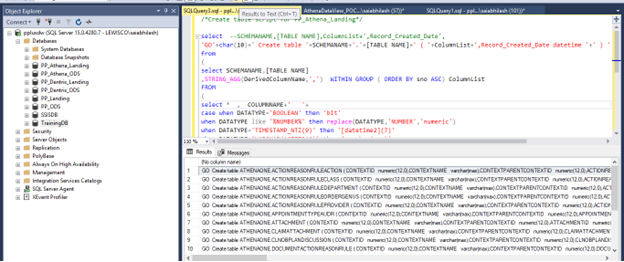
where TableType='Insert'

)a

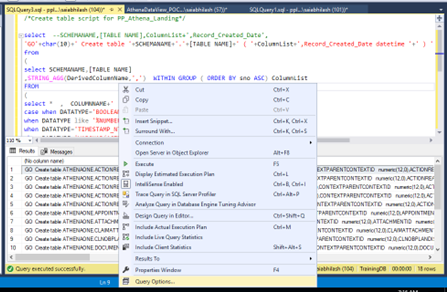
group by SCHEMANAME,[TABLE NAME]

) res

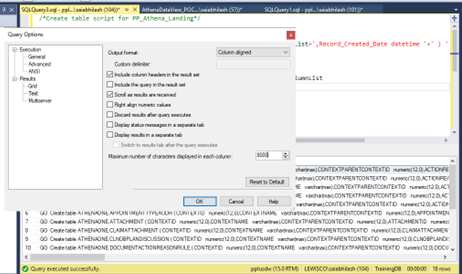
Result to Text

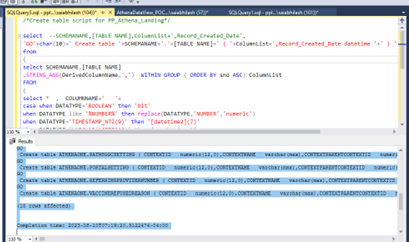
Right Click Query Options



Change results à text à max no. of characters to 8000



Execute    
Copy Result



Paste in new tab and executes (Execute them in training db for testing first)

Follow same procedure followed for Athena Landing tables creation

/\*Create table script for PP\_Athena\_ODS\*/

**Use PP\_Athena\_ODS**

**go**

select

'GO'+char(10)+' Create table '+SCHEMANAME+'.'+[TABLE NAME]+' ( '+ColumnList+

',Record\_Created\_Date datetime , Record\_Updated\_Date datetime '+' ) '

from

(

select SCHEMANAME,[TABLE NAME]

,STRING\_AGG(DerivedColumnName,',')  WITHIN GROUP ( ORDER BY sno ASC) ColumnList

FROM

(

select \*  ,  COLUMNNAME+'   '+

case when DATATYPE='BOOLEAN' then 'bit'

when DATATYPE like '%NUMBER%' then replace(DATATYPE,'NUMBER','numeric')

when DATATYPE='TIMESTAMP\_NTZ(9)' then '[datetime2](7)'

when DATATYPE='VARCHAR(16777216)' then VarcharDataType

else DATATYPE end DerivedColumnName

FROM **[PP\_Athena\_Landing]**..[AthenaDataView]

where TableType='Insert'

)a

group by SCHEMANAME,[TABLE NAME]

) res

-----**Insert into SSIS config table PP\_Athena\_Landing**

**USE PP\_Athena\_Landing**

INSERT INTO SSISConfig (

SourceTable,TargetTable,SourceColumn,TargetColumn,

Record\_Created\_Date,Datefield,IsLoadedToLanding,IsLoadedToODS,IsActive,Schema\_Name,Group\_No

)

select

SourceTable,TargetTable,SourceColumn,TargetColumn,

Record\_Created\_Date,Datefield,IsLoadedToLanding,IsLoadedToODS,IsActive,SCHEMANAME,

NTILE (7) OVER (

ORDER BY SourceTable

)Group\_No

from

(

select [TABLE NAME] as SourceTable ,[TABLE NAME] as TargetTable

,STRING\_AGG(COLUMNNAME,',')  WITHIN GROUP ( ORDER BY sno ASC) as SourceColumn

,STRING\_AGG(COLUMNNAME,',')  WITHIN GROUP ( ORDER BY sno ASC) as TargetColumn

,GETDATE() as Record\_Created\_Date,'LASTUPDATED' as Datefield,

0 as IsLoadedToLanding,

0 as IsLoadedToODS,

1 as IsActive,

SCHEMANAME

FROM [PP\_Athena\_Landing]..[AthenaDataView]

where TableType='Insert'

group by SCHEMANAME,[TABLE NAME] )res

**----------Insert into GetMerge table (PP\_Athena\_ODS)**

**USE PP\_Athena\_ODS**

insert into GetMergeTables

(SourceDBName,TargetDBName,SourceTable,TargetTable,PKColumnlist,IsActive)

select 'PP\_Athena\_Landing','PP\_Athena\_ODS',[TABLE NAME],[TABLE NAME],'['+COLUMNNAME+']',1

from

PP\_Athena\_Landing..[AthenaDataView] a

where PRIMARYKEY ='Primary Key' and TableType='insert'

order by a.[TABLE NAME],SNO

**------Create new indexes**

**Use PP\_Athena\_ODS**

select

'CREATE CLUSTERED INDEX IX\_'+[TABLE NAME]+' ON '+SCHEMANAME+'.'+[TABLE NAME]+' ( '+PKColumnList+' )'

+char(10)+' GO '

from

(

select SCHEMANAME,[TABLE NAME],STRING\_AGG(COLUMNNAME,',')  WITHIN GROUP ( ORDER BY sno desc) PKColumnList

from

**PP\_Athena\_Landing**..[AthenaDataView] a

where PRIMARYKEY ='Primary Key' and TableType='insert'

group by SCHEMANAME,[TABLE NAME]  )abc

1. **Additional SQL Operations on AthenaDataView:**
   * Perform additional SQL operations as needed.

**Scenario 2: New Columns to be Added (Update Table Scenario)**

1. **List of Tables Whose Columns to be Updated:**
   * Check for missing tables and execute queries similar to Scenario 1.

select  TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES

Where TABLE\_NAME in (< **List of Tables whose columns to be updated** >)

select TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES

Where TABLE\_NAME in (< **List of Tables whose columns to be updated** >)

group by TABLE\_NAME

having count(TABLE\_SCHEMA) > 1

* + Mark tables to be updated and execute queries.

1. **Execute Snowflake Query to Identify Tables for Column Update:**

SELECT 'union all SELECT '''||TABLE\_NAME||''' as Tablename , COUNT(\*) as CNT FROM '|| TABLE\_SCHEMA || '.' ||TABLE\_NAME AS Query

FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME in

( < List of Tables whose columns to be updated > )

1. **Capture Missing Tables:**
   * If any tables are missing, add them to the Update Table scenario.
2. **Execute Query in Snowflake:**
   * Execute the Snowflake query to identify tables for column update and copy the result.

5. **Mark Zero Count tables in Separate Sheet and Remove from List of tables**

select Tablename

from (**<Copied Query>**) abc where CNT = 0

6.**Check whether Table list from Point 4 O/p Exists in PP\_Athena\_ODS and Landing**

select \* from INFORMATION\_SCHEMA.TABLES where TABLE\_NAME in (< **List of Tables whose columns to be updated** >)

**7.Check if any of these tables are important tables used in Reports**

select \* from SPandItsTables

where TableName in (< **List of Tables whose columns to be updated** >)

If these tables are important tables then we should plan carefully else can proceed with below

**\*\*\***Add the missed table to New view scenario and remove from List of Tables whose columns to be updated

**7. Execute Point 4 Query in Snowflake DB and Copy the Tables list**

**This** CSVCOLUMNSLIST **Result is the FINAL DataView Tables to be ADDED in ODS**

**8.Add Tables list and run this on snowflakes**

SELECT 'union all select '||''''||table\_schema||''''||' as table\_schema,'

||''''||table\_name||''''||' as table\_name,'||''''||Column\_name||''''||' as Column\_name, ' ||

'max(len('||Column\_name||')) MaxLen from '||table\_schema||'.'||table\_name  as Query

FROM information\_schema.COLUMNS

where table\_name in

(**<** **List of Tables whose columns to be updated >** )

and DATA\_TYPE='TEXT' and CHARACTER\_MAXIMUM\_LENGTH='16777216'

order by table\_name,ordinal\_position

**9. Select COPY button and paste it in the same window and Remove QUERY Union all in first line and Execute**

**10.** Copy Result and Paste in Notepad  And Copy Notepad content and paste in Excel and save it in

Excel :(AthenaDV\_VarcharCol\_MaxLength)

**11.Open Athena Data View Excel shared by Athena Team**

i) Remove **Table Description** Sheet

ii) Add Column **SNO** at the starting of **TableDetails** Sheet .Add Value 1 in A2 Cell and Place =A2+1 in A3 Cell    
iii) Drag the formula to end of sheet .

iv )Remove columns (I) Descriptions and (J) ForeignKey ,Save the sheet

12 . **Connect to Database : PP\_Athena\_Landing  and Import Excel Data**(**Initially you can use training DB for testing the queries and then do same activity in athena landing** )

Drop table [dbo].[AthenaDataView]    
Drop table **AthenaDV\_VarcharCol\_MaxLength**

**CREATE TABLE [dbo].[AthenaDataView] (**

**SNO int ,**

**[TABLEID] int,**

**[SCHEMANAME]  varchar(255),**

**[TABLE DESCRIPTION]  varchar(255),**

**[TABLE NAME]  varchar(255),**

**[COLUMNNAME]  varchar(255),**

**[DATATYPE]  varchar(255),**

**[PRIMARYKEY]  varchar(255)**

**)**

**CREATE TABLE [dbo].[AthenaDV\_VarcharCol\_MaxLength] (**

**[TABLE\_SCHEMA] varchar(255),**

**[TABLE\_NAME] varchar(255),**

**[COLUMN\_NAME] varchar(255),**

**[MAXLEN] float**

**)**

1. **Import of Athena Data View** Excel **Data to PP\_Athena\_Landing**

select \* FROM  [dbo].[AthenaDataView]

**-----Alter table and add new column**

**alter table [dbo].[AthenaDataView]**

**add VarcharDataType varchar(100)**

alter table [dbo].[AthenaDataView]

add TableType varchar(100)

**II) Import AthenaDV\_VarcharCol\_MaxLength.xls sheet into**

Table : **AthenaDV\_VarcharCol\_MaxLength**

select \* from  AthenaDV\_VarcharCol\_MaxLength

**USE PP\_Athena\_Landing**

----------Delete records other than list of tables

delete a

--select \*

from [dbo].[AthenaDataView] a

where [TABLE NAME] not in

**(< List of Tables whose columns to be updated >)**

---------Delete duplicate tables who are present in two schemas

delete a

--select \*

from [dbo].[AthenaDataView] a

where [TABLE NAME]  in

('APPOINTMENT') and SCHEMANAME='SCHEDULING'

------------Mark Update tables

 update a set TableType='update'

 --select \*

 from

 [dbo].[AthenaDataView] a

 where [TABLE NAME] in

 (**<** **List of Tables whose columns to be updated >**)

**--------The count should be equal to List of tables count**

select distinct [TABLE NAME]

from [dbo].[AthenaDataView] a where TableType='update'

**--------Count should be same as total tables list (all tables should have**

**Last updated date column)**

select distinct [TABLE NAME]

from

[dbo].[AthenaDataView] where COLUMNNAME='LASTUPDATED'

And TableType='update'

-------**Assign Varchar(Max) Columns Varchar length**

update a set VarcharDataType=case when isnull(l.MAXLEN,0) < 20 then 'varchar(200)'

            when isnull(l.MAXLEN,0) < 100 then 'varchar(500)'

when isnull(l.MAXLEN,0) < 500 then 'varchar(2000)'

else 'varchar(8000)' end

--select l.\*,isnull(l.MAXLEN,0),case when isnull(l.MAXLEN,0) < 20 then 'varchar(200)'

--            when isnull(l.MAXLEN,0) < 100 then 'varchar(500)'

-- when isnull(l.MAXLEN,0) < 500 then 'varchar(2000)'

-- else 'varchar(8000)' end

from

[dbo].[AthenaDataView] a

join [dbo].AthenaDV\_VarcharCol\_MaxLength l

on a.SCHEMANAME =l.TABLE\_SCHEMA and a.[TABLE NAME]=l.TABLE\_NAME

and a.COLUMNNAME=l.COLUMN\_NAME

where a.DATATYPE='VARCHAR(16777216)'

-----**No records should come in all 3 queries,If any record comes stop the process**

select \* from [dbo].[AthenaDataView] a

where a.DATATYPE='VARCHAR(16777216)'

and VarcharDataType is null

SELECT \*

FROM  [dbo].[AthenaDataView]

WHERE

(LEN([table name]) != DATALENGTH([table name]))

or

(LEN(COLUMNNAME) != DATALENGTH(COLUMNNAME))

or

(LEN(DATATYPE) != DATALENGTH(DATATYPE))

or

(LEN(PRIMARYKEY) != DATALENGTH(PRIMARYKEY))

SELECT distinct DATATYPE

FROM  [dbo].[AthenaDataView]

where DATATYPE not in ('BOOLEAN','TIMESTAMP\_NTZ(9)','VARCHAR(16777216)','DATE')

and DATATYPE not like '%NUMBER%'

order by DATATYPE

**If Varchar(n) comes in above query its ok**

**------check for duplicate column details**

with cte as

(

select \*,

ROW\_NUMBER() over(partition by SCHEMANAME,[TABLE NAME],COLUMNNAME order by sno asc)  Rnk

from [dbo].[AthenaDataView]

)

----delete

--select \*

from cte  where Rnk > 1

**USE PP\_Athena\_Landing**

-----Check for any duplicates

select s.TargetTable,[Schema\_Name]

from SSISConfig s

join

(

select distinct SCHEMANAME,[TABLE NAME]

from

[AthenaDataView]

where TableType='Update'

) a on a.SCHEMANAME=s.[Schema\_Name] and a.[TABLE NAME]=s.TargetTable

group by s.TargetTable,[Schema\_Name]

having count(1)> 1

**----Drop tables for update columns tables**

**Use PP\_Athena\_Landing**

select

distinct 'GO'+char(10)+' drop table '+SCHEMANAME+'.'+[TABLE NAME]

from

[dbo].[AthenaDataView] a

where TableType='Update'

**/\*Create table script for PP\_Athena\_Landing\*/**

**Use PP\_Athena\_Landing**

select

'GO'+char(10)+' Create table '+SCHEMANAME+'.'+[TABLE NAME]+' ( '+ColumnList+',Record\_Created\_Date datetime '+' ) '

from

(

select SCHEMANAME,[TABLE NAME]

,STRING\_AGG(DerivedColumnName,',')  WITHIN GROUP ( ORDER BY sno ASC) ColumnList

FROM

(

select \*  ,  COLUMNNAME+'   '+

case when DATATYPE='BOOLEAN' then 'bit'

when DATATYPE like '%NUMBER%' then replace(DATATYPE,'NUMBER','numeric')

when DATATYPE='TIMESTAMP\_NTZ(9)' then '[datetime2](7)'

when DATATYPE='VARCHAR(16777216)' then 'varchar(max)'

else DATATYPE end DerivedColumnName

FROM  [dbo].[AthenaDataView]

where TableType='update'

)a

group by SCHEMANAME,[TABLE NAME]

) res

**----Drop tables for update columns tables**

**Use PP\_Athena\_ODS**

select

distinct 'GO'+char(10)+' drop table '+SCHEMANAME+'.'+[TABLE NAME]

from

PP\_Athena\_Landing..[AthenaDataView]  a

where TableType='Update'

/\*Create table script for PP\_Athena\_ODS\*/

**Use PP\_Athena\_ODS**

**go**

select

'GO'+char(10)+' Create table '+SCHEMANAME+'.'+[TABLE NAME]+' ( '+ColumnList+

',Record\_Created\_Date datetime , Record\_Updated\_Date datetime '+' ) '

from

(

select SCHEMANAME,[TABLE NAME]

,STRING\_AGG(DerivedColumnName,',')  WITHIN GROUP ( ORDER BY sno ASC) ColumnList

FROM

(

select \*  ,  COLUMNNAME+'   '+

case when DATATYPE='BOOLEAN' then 'bit'

when DATATYPE like '%NUMBER%' then replace(DATATYPE,'NUMBER','numeric')

when DATATYPE='TIMESTAMP\_NTZ(9)' then '[datetime2](7)'

when DATATYPE='VARCHAR(16777216)' then VarcharDataType

else DATATYPE end DerivedColumnName

FROM **[PP\_Athena\_Landing]**..[AthenaDataView]

where TableType='update'

)a

group by SCHEMANAME,[TABLE NAME]

) res

**USE PP\_Athena\_Landing**

**------------Update SSIS Config table for Updated columns tables :**

update s set SourceColumn=a.SourceColumn ,TargetColumn=a.TargetColumn,Record\_Created\_Date=a.Record\_Created\_Date

---select s.TargetTable,s.[Schema\_Name],

---s.SourceColumn,s.TargetColumn,a.SourceColumn,a.TargetColumn

from SSISConfig s

join

(

select [TABLE NAME] as SourceTable

,STRING\_AGG(COLUMNNAME,',')  WITHIN GROUP ( ORDER BY sno ASC) as SourceColumn

,STRING\_AGG(COLUMNNAME,',')  WITHIN GROUP ( ORDER BY sno ASC) as TargetColumn

,GETDATE() as Record\_Created\_Date

,SCHEMANAME

FROM [PP\_Athena\_Landing]..[AthenaDataView]

where TableType='Update'

group by SCHEMANAME,[TABLE NAME]

) a on a.SCHEMANAME=s.[Schema\_Name] and a.SourceTable=s.SourceTable

**--------Insert into GetMerge table (PP\_Athena\_ODS) FOR New table only or if primary key change ,be aware of tables like [transaction] which has [] in table name.**

**USE PP\_Athena\_ODS**

insert into GetMergeTables

(SourceDBName,TargetDBName,SourceTable,TargetTable,PKColumnlist,IsActive)

select 'PP\_Athena\_Landing','PP\_Athena\_ODS',[TABLE NAME],[TABLE NAME],'['+COLUMNNAME+']',1 from

**PP\_Athena\_Landing..**[AthenaDataView] a

left join GetMergeTables g on a.[TABLE NAME]=g.SourceTable and g.PKColumnlist= '['+a.COLUMNNAME+']'

where a.PRIMARYKEY ='Primary Key' and a.TableType='update'

and g.PKColumnlist is null

order by a.[TABLE NAME],SNO

**-------Delete from GetMerge table**

select 'PP\_Athena\_Landing','PP\_Athena\_ODS',[TABLE NAME],[TABLE NAME],'['+COLUMNNAME+']',1 from

GetMergeTables g

left join

**PP\_Athena\_Landing..**[AthenaDataView] a

on a.[TABLE NAME]=g.SourceTable and g.PKColumnlist= '['+a.COLUMNNAME+']'

where a.PRIMARYKEY ='Primary Key' and a.TableType='update'

and a.COLUMNNAME is null

order by a.[TABLE NAME],SNO

**------Create new indexes**

**Use PP\_Athena\_ODS**

select

'CREATE CLUSTERED INDEX IX\_'+[TABLE NAME]+' ON '+SCHEMANAME+'.'+[TABLE NAME]+' ( '+PKColumnList+' )'

+char(10)+' GO '

from

(

select SCHEMANAME,[TABLE NAME],STRING\_AGG(COLUMNNAME,',')  WITHIN GROUP ( ORDER BY sno desc) PKColumnList

from

**PP\_Athena\_Landing**..[AthenaDataView] a

where PRIMARYKEY ='Primary Key' and TableType='update'

group by SCHEMANAME,[TABLE NAME]  )abc

**----------Validations**

**USE PP\_Athena\_Landing**

select  TABLE\_SCHEMA,TABLE\_NAME,COLUMN\_NAME FROM INFORMATION\_SCHEMA.COLUMNS

Where TABLE\_NAME in(

<List of tables >

) and COLUMN\_NAME='Record\_Created\_Date'

**Use PP\_Athena\_ODS**

select  TABLE\_SCHEMA,TABLE\_NAME,COLUMN\_NAME FROM INFORMATION\_SCHEMA.COLUMNS

Where TABLE\_NAME in(

<List of tables >

) and COLUMN\_NAME='Record\_Created\_Date'

select  TABLE\_SCHEMA,TABLE\_NAME,COLUMN\_NAME FROM INFORMATION\_SCHEMA.COLUMNS

Where TABLE\_NAME in(

<List of tables >

) and COLUMN\_NAME='Record\_Updated\_Date'

**After Testing in Training DB :**

**Pp\_athena\_landing**

drop table [AthenaDataView]

drop table AthenaDV\_VarcharCol\_MaxLength

select \* into AthenaDataView from TrainingDB..[AthenaDataView]

select \* into AthenaDV\_VarcharCol\_MaxLength from TrainingDB..AthenaDV\_VarcharCol\_MaxLength

INSERT INTO SSISConfig (

SourceTable,TargetTable,SourceColumn,TargetColumn,

Record\_Created\_Date,Datefield,IsLoadedToLanding,IsLoadedToODS,IsActive,Schema\_Name,Group\_No

)

select

'QUICKSTART\_SQL' SourceTable,'QUICKSTART\_SQL' TargetTable,

'ID,VERSION,NAME,VERSION\_DESCRIPTION,SQL\_QUERY,DESCRIPTION,TIME\_CREATED,TIME\_MODIFIED,STATUS' SourceColumn,

'ID,VERSION,NAME,VERSION\_DESCRIPTION,SQL\_QUERY,DESCRIPTION,TIME\_CREATED,TIME\_MODIFIED,STATUS' TargetColumn,

getdate() Record\_Created\_Date,

'TIME\_MODIFIED' Datefield,

'0' IsLoadedToLanding,'0' IsLoadedToODS,'1' IsActive,'TEMPLATES' SCHEMANAME,

1 Group\_No

insert into GetMergeTables

(SourceDBName,TargetDBName,SourceTable,TargetTable,PKColumnlist,IsActive)

select 'PP\_Athena\_Landing','PP\_Athena\_ODS','QUICKSTART\_SQL'[TABLE NAME],

'QUICKSTART\_SQL'[TABLE NAME],'[ID]' PKColumnlist ,1  IsActive

CREATE CLUSTERED INDEX IX\_QUICKSTART\_SQL ON TEMPLATES.QUICKSTART\_SQL(ID)

DROP INDEX IX\_QUICKSTART\_SQL

ON TEMPLATES.QUICKSTART\_SQL

1. **Validations:**
   * Perform validations on the updated data.

**Conclusion**

This detailed documentation outlines the steps for managing data in Snowflake, covering scenarios related to new views, new columns, and data updates. Each step is explained to ensure clarity and understanding of the process.