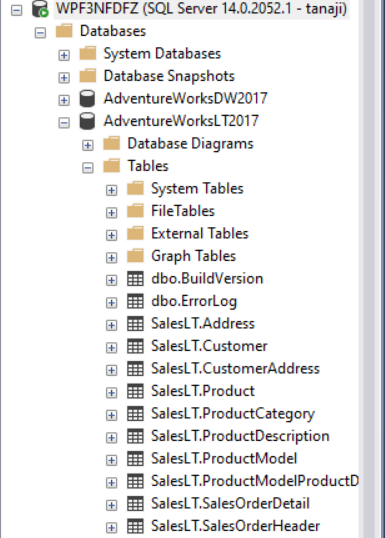
**End to End Data Engineering project (gen 2).**

* **Architecture Diagram**

A diagram of a server

Description automatically generated

1. We have **AdventureWorksLT2017** on premise database.



Note: - If you need files as CSV format in **datawarehouse** layer/container, we need to do some changes in Source i.e. Adventure works database.

As per the below table - columns are not supported whenever we are loading the data in DW because in this column having long value so that

|  |  |
| --- | --- |
| **TableName** | **ColumnName** |
| [SalesLT].[Product] | ThumbNailPhoto |
| [SalesLT].[ProductDescription] | Description |
| [SalesLT].[ProductModel] | CatalogDescription |

we need to create duplicate tables by using this tables. Please find the below SQL statements

[SalesLT].[Product]= [SalesLT].[Product\_Old]

* **[SalesLT].[Product]**

Select \* into SalesLT].[Product\_Old] from SalesLT].[Product] 🡪 Create new table

Alter table SalesLT].[Product\_Old] Drop Column ThumbNailPhoto 🡪 Drop Column from New table

* **[SalesLT].[ProductModel]**

Select \* into SalesLT].[Product\_Old] from SalesLT].[Product] 🡪 Create new table

Alter table SalesLT].[Product\_Old] Drop Column ThumbNailPhoto 🡪 Drop Column from New table

* **[SalesLT].[ProductDescription]**

SELECT [ProductDescriptionID]

,TRIM(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(Left([Description],250),'?',''),'.',''),'-',''),'"',''),',',''),';','')) as [Description]

,[rowguid]

,[ModifiedDate] into [AdventureWorksLT2017].[SalesLT].[ProductDescription\_Old]

FROM [AdventureWorksLT2017].[SalesLT].[ProductDescription] 🡪 Create new table

**Azure Resources: -**

A screenshot of a computer

Description automatically generated

1. **Azure Data Factory: -** Create Azure Data factory for Ingesting the data from on premise SQL server and load in Staging Layer in Azure Blob storage gen 2.
   1. **IR: -** We need to create IR (self-hosted) for fetching the data from on premise SQL server.
   2. **Linked Service: -**
      1. Create Link service for SQL server (We need to pass SQL server password though key vault.)
      2. We have already created key vault resource in resource group so now need to create linked service for Key Vault which needs to be pass when you need to pass secret in SQL server linked service (Please find the below snap).

A screenshot of a computer

Description automatically generated

* + 1. Create Key vault secret and pass this secret in linked service.

A screenshot of a computer

Description automatically generated

* 1. **Create Dataset for lookup Activity: -** Create dataset for on premise SQL server database by using SQL Server linked service.
  2. **Lookup activity: -** Using this activity we got all the schema name and table names from this on-premises database. (Please find the below query)

select S.Name as Schemaname,

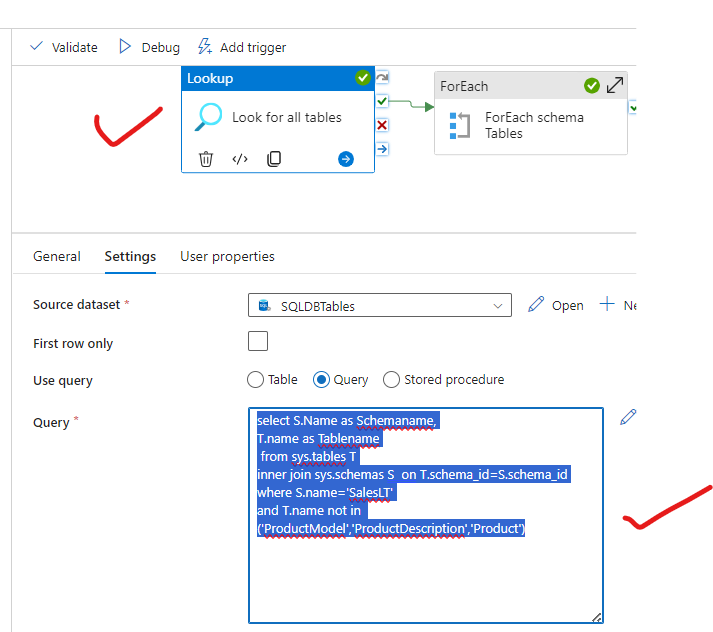
T.name as Tablename

from sys.tables T

inner join sys.schemas S on T.schema\_id=S.schema\_id

where S.name='SalesLT'

and T.name not in ('ProductModel','ProductDescription','Product')



* 1. **Foreach activity: -** In this activity we can read loop-wise all the table names from lookup activity and by using copy activity we can fetch data from SQL server and load into Blob storage container gen2

A screenshot of a computer

Description automatically generated

* 1. **Copy activity: -** Copy activity is used for data collect from source and load into sink location.
     1. **Source:**- @{concat('SELECT \* FROM ', item().SchemaName,'.', item().tableName)}

A screenshot of a computer

Description automatically generated

* + 1. **Sink: -** 1)We need to create dataset for dynamically store the files in csv format in Azure blob storage gen 2 Staging container with proper directory.

2) Create Dataset parameters as per the below snap

**A screenshot of a computer

Description automatically generated**

3) in the file path Please use the below content. (In Dataset Connection Option)

* + - 1. staging
      2. @{concat(dataset().schemaname, '/',dataset().tablename)}
      3. @{concat(dataset().tablename,'.csv')}

A screenshot of a computer

Description automatically generated

4) Go to Copy activity sink and in Dataset Properties we need to pass those items which is available in the dataset parameter. (please find the below details).

**A screenshot of a computer

Description automatically generated**

1. **Azure Data Bricks**: - Using Azure data bricks notebook, we can transform the data which is available in blob storage gen 2,
   * 1. Create Compute cluster in Azure data bricks.

**A screenshot of a computer

Description automatically generated**

* + 1. Go to the shared workspace 🡪 and create notebook 🡪 We need to mount the blob storage in Azure data bricks using Blob storage Access key. (Please find the below code)
* **Mount Storage account gen 2 container**

# Databricks notebook source

# MAGIC %md

# MAGIC ##Mount blob storages (Staging, Load and Datawarehouse) in Azure data bricks

# COMMAND ----------

dbutils.fs.mount(

source = 'wasbs://staging@satandatalakegen2.blob.core.windows.net',

mount\_point = '/mnt/staging',

extra\_configs = {'fs.azure.account.key.satandatalakegen2.blob.core.windows.net':'aUiNca/KjvQ1xEGMd2RpFEDBM8kGmDnqvbyjskMIwAdLxgCNOYNb1DbEKlKG4r5JL3I3ydLznYqJ+AStG/lMhQ=='})

# COMMAND ----------

dbutils.fs.ls("/mnt/staging/SalesLT")

# COMMAND ----------

dbutils.fs.mount(

source = 'wasbs://load@satandatalakegen2.blob.core.windows.net',

mount\_point = '/mnt/load',

extra\_configs = {'fs.azure.account.key.satandatalakegen2.blob.core.windows.net':'aUiNca/KjvQ1xEGMd2RpFEDBM8kGmDnqvbyjskMIwAdLxgCNOYNb1DbEKlKG4r5JL3I3ydLznYqJ+AStG/lMhQ=='})

# COMMAND ----------

dbutils.fs.ls("/mnt/load")

# COMMAND ----------

dbutils.fs.mount(

source = 'wasbs://datawarehouse@satandatalakegen2.blob.core.windows.net',

mount\_point = '/mnt/datawarehouse',

extra\_configs = {'fs.azure.account.key.satandatalakegen2.blob.core.windows.net':'aUiNca/KjvQ1xEGMd2RpFEDBM8kGmDnqvbyjskMIwAdLxgCNOYNb1DbEKlKG4r5JL3I3ydLznYqJ+AStG/lMhQ=='})

# COMMAND ----------

dbutils.fs.ls("/mnt/datawarehouse")

* + 1. We need to do 1st level of transformation for changing Date fields data types in all the tables as a “**YYYY-MM-dd**” format and load in **Staging to Load layer** with csv format.

**Change the data type of date columns and load the file in Staging to Load Layer (Storage Account gen2)**

# Databricks notebook source

# MAGIC %md

# MAGIC ## Change datatype Datetime to Date Transformation for all tables

# COMMAND ----------

table\_name =[]

for i in dbutils.fs.ls('mnt/staging/SalesLT/'):

table\_name.append(i.name.split('/')[0])

# COMMAND ----------

table\_name

# COMMAND ----------

from pyspark.sql.functions import from\_utc\_timestamp, date\_format

from pyspark.sql.types import TimestampType

for i in table\_name:

path='/mnt/staging/SalesLT/'+ i + '/' + i +'.csv'

df=spark.read.format('csv').option('header','true').load(path)

column = df.columns

for col in column:

if "Date" in col or "date" in col:

df = df.withColumn(col, date\_format(from\_utc\_timestamp(df[col].cast(TimestampType()), "UTC"),"yyyy-MM-dd"))

output\_path = '/mnt/load/SalesLT/'+ i

df.write.format('csv').option('header','true').mode("overwrite").save(output\_path)

# COMMAND ----------

display(df)

* + 1. We need to set the proper column names and move all the files in **Datawarehouse** layer.

**Set proper column names like as ColumnName to Column\_Name and load files Load to Datawarehouse Layer (Storage Account gen2)**

# Databricks notebook source

# MAGIC %md

# MAGIC ## Change "ColumnName" to "Column\_Name" for all tables

# COMMAND ----------

table\_name = []

for i in dbutils.fs.ls('mnt/load/SalesLT/'):

table\_name.append(i.name.split('/')[0])

# COMMAND ----------

table\_name

# COMMAND ----------

for name in table\_name:

path='/mnt/load/SalesLT/'+ name

print(path)

df=spark.read.format("csv").option('header','true').load(path)

#get the list of column\_names

column\_names=df.columns

for old\_col\_name in column\_names:

# convert column name from column to column\_name format

new\_col\_name="".join(["\_"+ char if char.isupper() and not old\_col\_name[i-1].isupper() else char for i, char in enumerate(old\_col\_name)]).lstrip("\_")

# Change the column name using withcolumnRenamed and regexp\_replace

df=df.withColumnRenamed(old\_col\_name,new\_col\_name)

output\_path='/mnt/datawarehouse/SalesLT/' + name + '/'

df.write.format('csv').option('header','true').mode("overwrite").save(output\_path)

# COMMAND ----------

display(df)

* + 1. In Azure data bricks we have created notebook for transformation of the data 🡪 Now we need to add Notebook Activity in Azure data factory.
       1. We need to create Access token in Azure data bricks because we need to provide access Azure data bricks to Azure data factory 🡪 Go to Profile 🡪 Setting 🡪 User 🡪 Developer 🡪 generate new token🡪 one key is generated (This key is only one time visible)🡪 we need to Copy this key and create one secret in the key vault and save it.
    2. Go to Azure data factory and create one linked service for Azure Data bricks 🡪 In Authentication type choose Access token 🡪 select linked service for key vault and choose secret name in this secret we have already stored Azure data bricks token key. (**point no v**)

**A screenshot of a computer

Description automatically generated**

* + 1. In ADF add Azure data bricks Notebook activity, select Linked service🡪 then go to setting and select actual path of the note which is available in azure data bricks.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. **Azure Synapse Analytics**: - Create Azure Synapse Analytics resource(*whenever you creating it in Primary ADLS Gen2 file system = Staging*) 🡪 Open It Synapse workspace🡪 Data Pane 🡪 Linked tab 🡪 Azure Data Lake storage gen2(Blob storage Gen 2 auto linked with Synapse) 🡪 Expand 🡪 Check the layers which is available in blob container 🡪 you can access the file which are available in container 🡪 Whenever you are right click and select Top 1000 rows 🡪 run the below script 🡪 you got the error 🡪

**A screenshot of a computer

Description automatically generated**

-- This is auto-generated code

SELECT

    TOP 100 \*

FROM

    OPENROWSET(

        BULK 'https://satandatalakegen2.dfs.core.windows.net/datawarehouse/SalesLT/Address/\*\*',

        FORMAT = 'CSV',

        PARSER\_VERSION = '2.0'

    ) AS [result]

Q) How to overcome this error Directory not listed?

A computer code with red line

Description automatically generated

🡪We need to provide the access in Storage account for this **ADF, Azure Synapse Analytics** and **our user** as **Storage Blob Data Contributor** 🡪 and then check scripts 🡪 then you can access the data.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

As per the above image select text format🡪 Apply 🡪 change the script as per the requirement 🡪 Please find the below snap and run it.

A screenshot of a computer

Description automatically generated

**Server less SQL server DB create: -**🡪Create SQL database 🡪 Workspace 🡪 + 🡪 SQL database 🡪 Serverless 🡪 (serverless: - we have small data in this files) 🡪 provide database name (Name: - database\_AW) 🡪 create the database.

A screenshot of a computer

Description automatically generated

We need to create a view for each file and load into Azure Synapse Workspace SQL server.

**Note: -** This Query only used for one single View created in Azure Synapse Workspace SQL server.

CREATE View Address as

SELECT

    \*

FROM

    OPENROWSET(

        BULK 'https://satandatalakegen2.dfs.core.windows.net/datawarehouse/SalesLT/Address/\*\*',

        FORMAT = 'CSV',

        FIRSTROW = 1,

        PARSER\_VERSION = '2.0'

HEADER\_ROW = TRUE

    ) AS [result]

As per the above script its static way for creating view but we need to create each folder’s view by using Dynamic way because if new view folder is inserted in Datawarehouse layer then view will auto create.

We can create Stored procedure for this.

**For Storage Account gen 2: -**

USE database\_AW

GO

CREATE OR ALTER PROCEDURE createSQLserverlessView\_datawarehouse @ViewName NVARCHAR(100)

AS

BEGIN

DECLARE @Statement VARCHAR(MAX)

    SET @Statement = N'CREATE OR ALTER VIEW ' + @ViewName + ' AS

        SELECT \*

        FROM

            OPENROWSET(

            BULK ''https://satandatalakegen2.dfs.core.windows.net/datawarehouse/SalesLT/'+ @ViewName +'/'',

            FORMAT = ''CSV'',

            FIRSTROW = 1,

            PARSER\_VERSION = ''2.0'',

HEADER\_ROW = TRUE

        ) AS [result]

    '

EXEC (@Statement)

END

GO

A screenshot of a computer

Description automatically generated

* Publish above code in Azure Synapse

Now we can create the Data factory (Pipeline) in Azure Synapse Analytics using above Stored procedure.

**Linked Service for serverless SQL: -** Create Linked service for Azure SQL Database in synapse 🡪 Go to manage 🡪 New 🡪 Change the Name as ServerlessSQLDB🡪 Click on Enter Manually 🡪 Fully qualified domain name = Go to Synapse properties and copy serverless SQL end point 🡪 find the below snap

A screenshot of a computer

Description automatically generated

and copy the database name as well from synapse workspace (**database\_dw**) 🡪 Find the below snap.

A screenshot of a computer

Description automatically generated

If you get the error while test connection, 🡪 you need to allow this permission. 🡪 Wait for some time and again test 🡪 then connection will be tested 🡪 (find the below snap)

A screenshot of a computer

Description automatically generated

**Synapse Data factory Create: -** Go to Integrate tab 🡪 create new pipeline🡪

* **Get Metadata Activity: -** fetch all the metadata names in blob storage gen2
  + In this metadata activity in setting tab 🡪 connect data set 🡪 New 🡪 Azure Blob storage gen 2🡪Continue🡪Binary🡪 select workspace default storage linked service and browse path (check below snap)

A screenshot of a computer

Description automatically generated

* + Then choose Child Items as field List (PF snap)

A screenshot of a computer

Description automatically generated

* **Foreach loop Activity: -** use foreach loop activity 🡪 in setting tab🡪 Items 🡪

@activity('Get tablenames').output.childItems

A screenshot of a computer

Description automatically generated

* + **Stored Procedure Activity: -** As we already created stored procedure in serverless SQL pool, we need to use this stored procedure in this pipeline by using Stored Procedure Activity. (Please find the below snap for configuration)

A screenshot of a computer

Description automatically generated

Change the pipeline name as a “**create view Pipeline**” and then publish it🡪 Run OR Debug the pipeline and check in Data 🡪 database 🡪 check views 🡪 views are auto created using SQL stored procedure in synapse Data factory.

1. **Power BI Report :** -
   1. Connect data source from Azure synapse DB to power BI.

A screenshot of a computer

Description automatically generated

**Server =** Go to Synapse resource 🡪 Overview 🡪 Serverless SQL endpoint 🡪 Copy this URL and paste in server in Power BI

**Database =** copy database from synapse 🡪 Go to Data Pane 🡪Workspace 🡪 SQL database 🡪 Copy the name and paste it

Authenticate and connect it.

A screenshot of a computer

Description automatically generated

Select All the tables and load the table into Power BI.

* 1. **Model creation: -** Create the model as per the below diagram.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

**Report Snap: -**

A screenshot of a computer

Description automatically generated