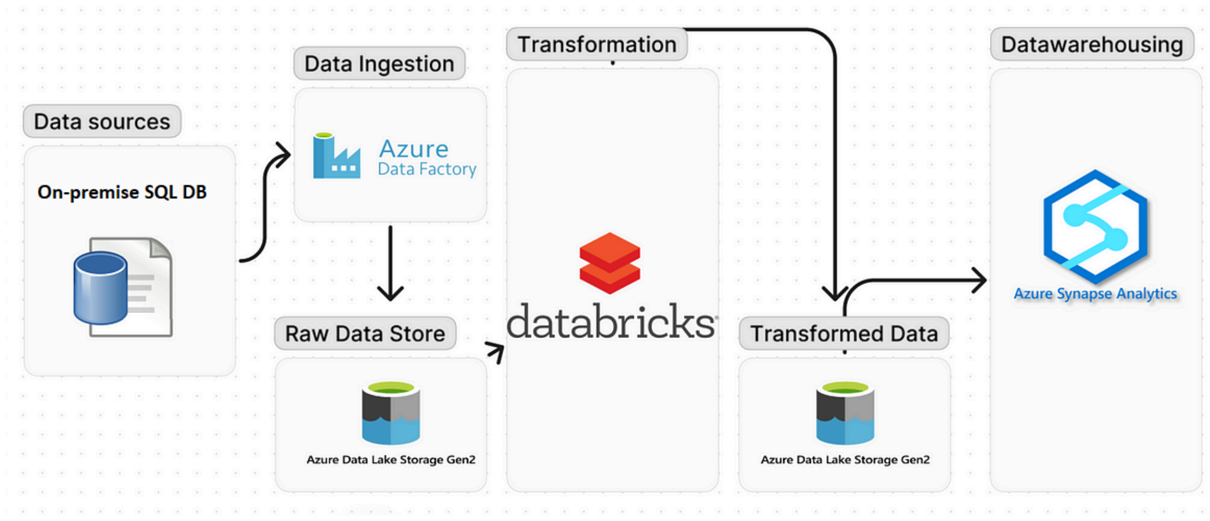
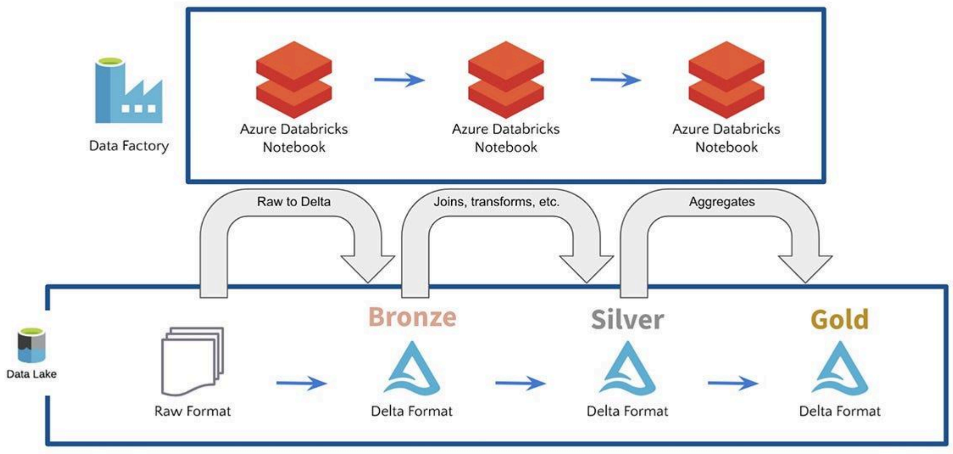
Mahesh Raut - Project - 31 May 2024

Project 1 and 2

Project Architecture:

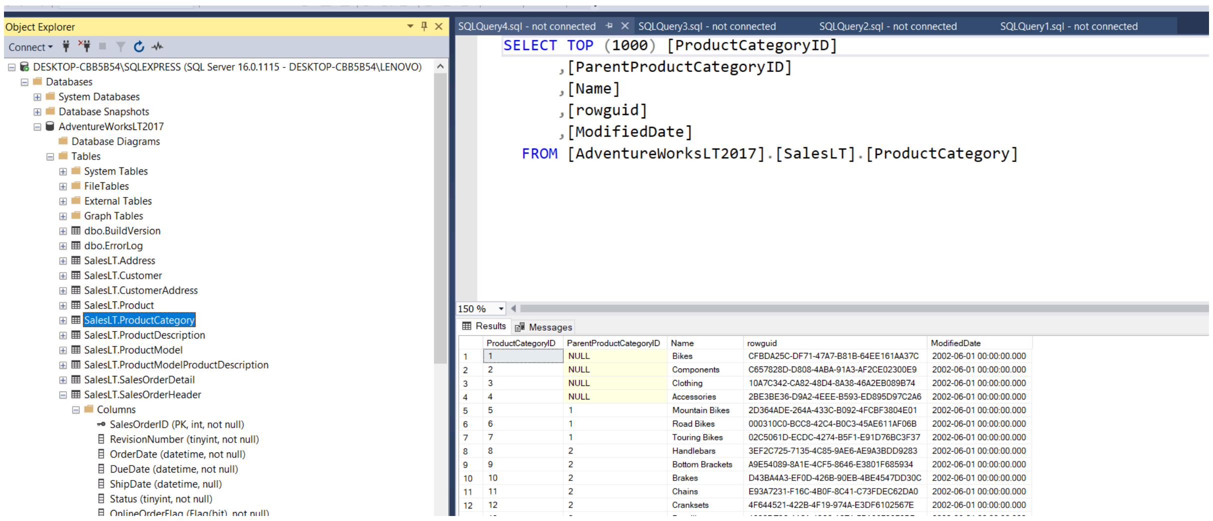




Part 1 - Dataset Overview:

AdventureWorks database supports standard online transaction processing scenarios for a fictitious bicycle manufacturer - Adventure Works Cycles. Scenarios include Manufacturing, Sales, Purchasing, Product Management, Contact Management, and Human Resources.

Mahesh Raut - Project - 31 May 2024

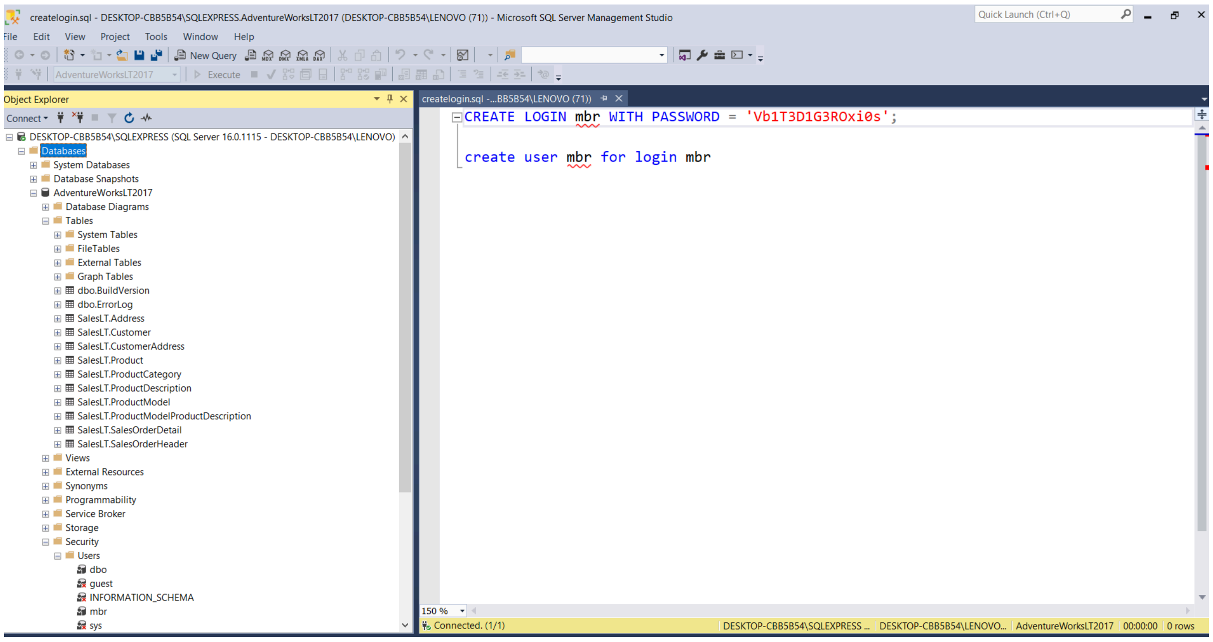

Object Explorer
Connect
DESKTOP-CBB5B54\SQLEXPRESS (SQL Server 16.0.1115 - DESKTOP-CBB5B54\LENOVO)
Databases
System Databases
Database Snapshots
AdventureWorksLT2017
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.BuildVersion
dbo.ErrorLog
SalesLT.Address
SalesLT.Customer
SalesLT.CustomerAddress
SalesLT.Product
SalesLT.ProductCategory
SalesLT.ProductDescription
SalesLT.ProductModel
SalesLT.ProductModelProductDescription
SalesLT.SalesOrderDetail
SalesLT.SalesOrderHeader
Columns
SalesOrderID (PK, int, not null)
RevisionNumber (tinyint, not null)
OrderDate (datetime, not null)
DueDate (datetime, not null)
ShipDate (datetime, null)
Status (tinyint, not null)
OnlineOrderFlag (Flag(bit), not null)
SQLQuery4.sql - not connected
SQLQuery3.sql - not connected
SQLQuery2.sql - not connected
SQLQuery1.sql - not connected
150 %
Results
Messages

```sql
SELECT TOP (1000) [ProductCategoryID]
      ,[ParentProductCategoryID]
      ,[Name]
      ,[rowguid]
      ,[ModifiedDate]
  FROM [AdventureWorksLT2017].[SalesLT].[ProductCategory]
```

	ProductCategoryID	ParentProductCategoryID	Name	rowguid	ModifiedDate
1	1	NULL	Bikes	CFBDA25C-DF71-47A7-B81B-64EE161AA37C	2002-06-01 00:00:00.000
2	2	NULL	Components	C657828D-D808-4ABA-91A3-AF2CE02300E9	2002-06-01 00:00:00.000
3	3	NULL	Clothing	10A7C342-CA82-48D4-8A38-46A2EB089B74	2002-06-01 00:00:00.000
4	4	NULL	Accessories	2BE3BE36-D9A2-4EEE-B593-ED895D97C2A6	2002-06-01 00:00:00.000
5	5	1	Mountain Bikes	2D364ADE-264A-433C-B092-4FCBF3804E01	2002-06-01 00:00:00.000
6	6	1	Road Bikes	000310C0-BCC8-42C4-B0C3-45AE611AF06B	2002-06-01 00:00:00.000
7	7	1	Touring Bikes	02C5061D-ECDC-4274-B5F1-E91D76BC3F37	2002-06-01 00:00:00.000
8	8	2	Handlebars	3EF2C725-7135-4C85-9AE6-AE9A3BDD9283	2002-06-01 00:00:00.000
9	9	2	Bottom Brackets	A9E54089-8A1E-4CF5-8646-E3801F685934	2002-06-01 00:00:00.000
10	10	2	Brakes	D43BA4A3-EF0D-426B-90EB-4BE4547DD30C	2002-06-01 00:00:00.000
11	11	2	Chains	E93A7231-F16C-4B0F-8C41-C73FDEC62DA0	2002-06-01 00:00:00.000
12	12	2	Cranksets	4F644521-422B-4F19-974A-E3DF6102567E	2002-06-01 00:00:00.000

Create Database login

```sql
CREATE LOGIN mbr WITH PASSWORD = 'Vb1T3D1G3ROxi0s';
create user mbr for login mbr
```

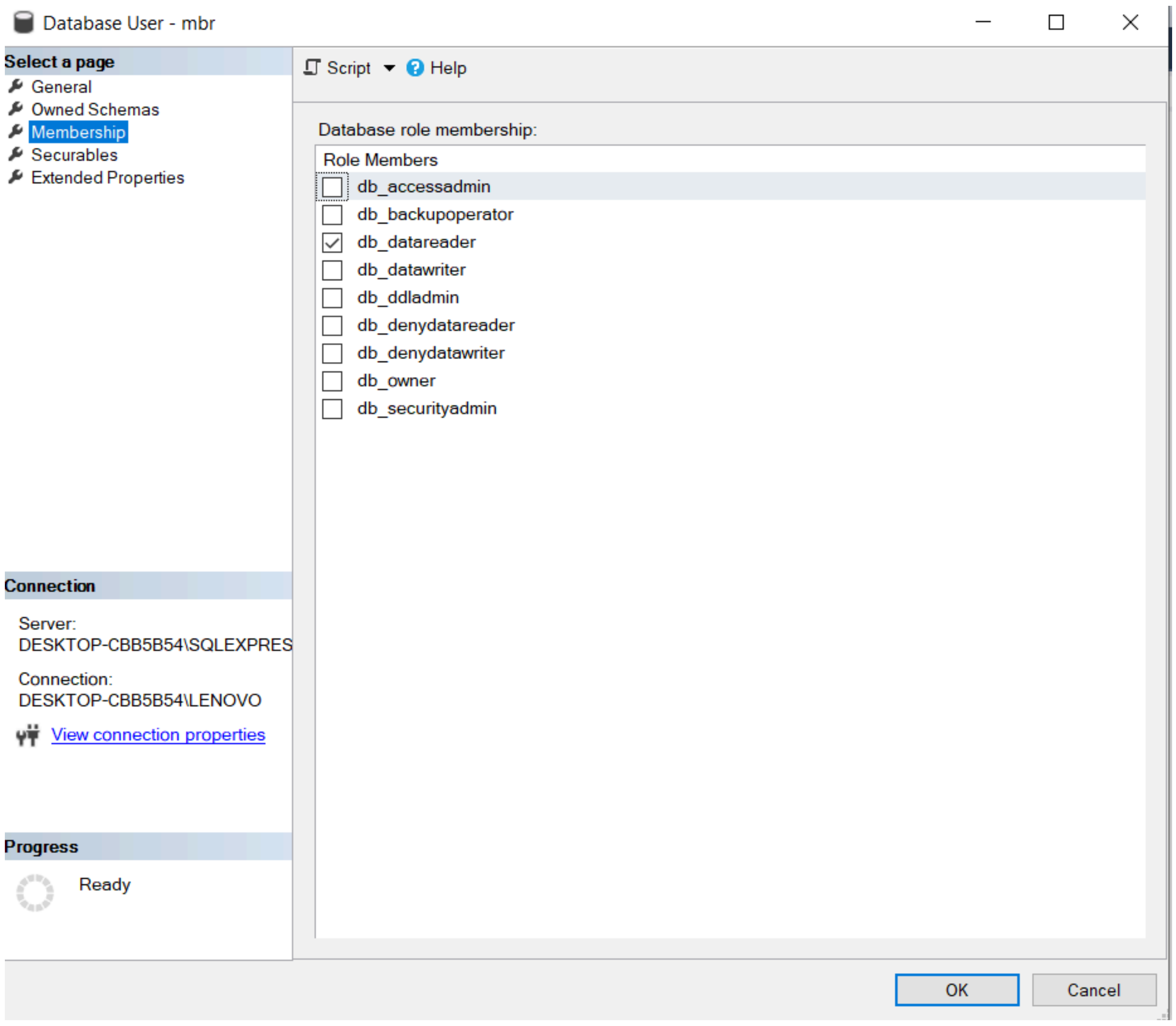


```sql
CREATE LOGIN mbr WITH PASSWORD = 'Vb1T3D1G3ROxi0s';

create user mbr for login mbr
```

```
Connected. (1/1)    DESKTOP-CBB5B54\SQLEXPRESS ...  DESKTOP-CBB5B54\LENOVO...  AdventureWorksLT2017  00:00:00  0 rows
```

Mahesh Raut - Project - 31 May 2024

Give User access:


Database User - mbr

Select a page
- General
- Owned Schemas
- Membership
- Securables
- Extended Properties

Script ▾ Help

Database role membership:

Role Members
☐ db_accessadmin
☐ db_backupoperator
☑ db_datareader
☐ db_datawriter
☐ db_ddladmin
☐ db_denydatareader
☐ db_denydatawriter
☐ db_owner
☐ db_securityadmin

Connection

Server:
DESKTOP-CBB5B54\SQLEXPRES

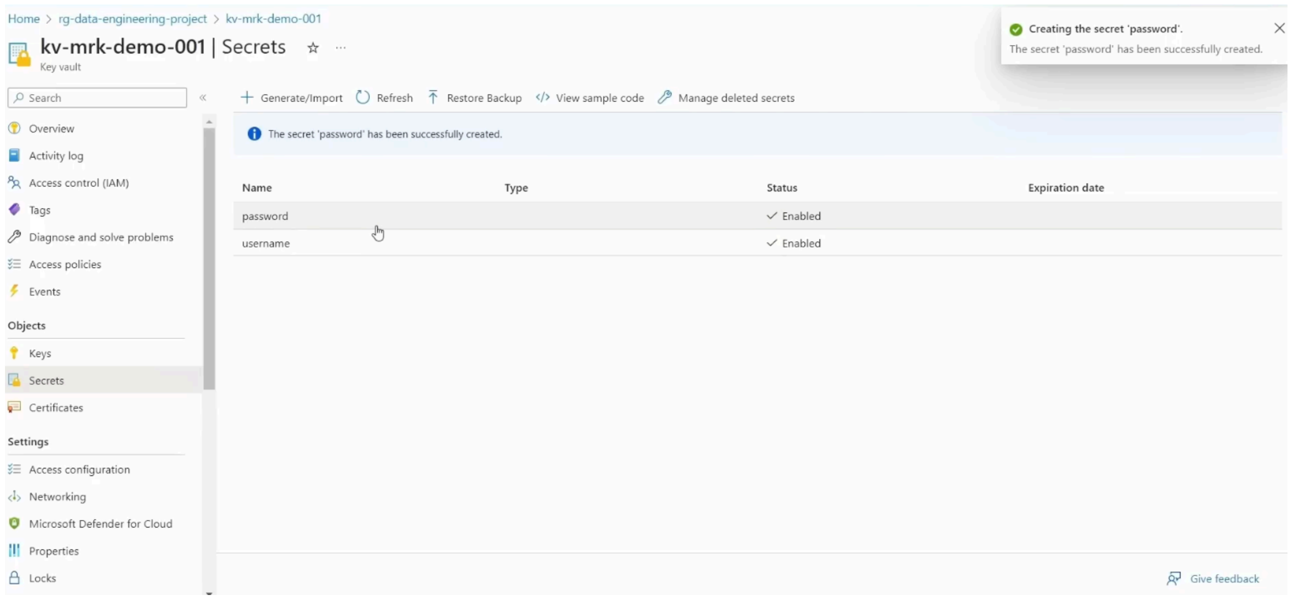
Connection:
DESKTOP-CBB5B54\LENOVO

View connection properties

Progress

Ready

Store secrets in Azure Key Vault:


Home > rg-data-engineering-project > kv-mrk-demo-001

kv-mrk-demo-001 | Secrets

Key vault

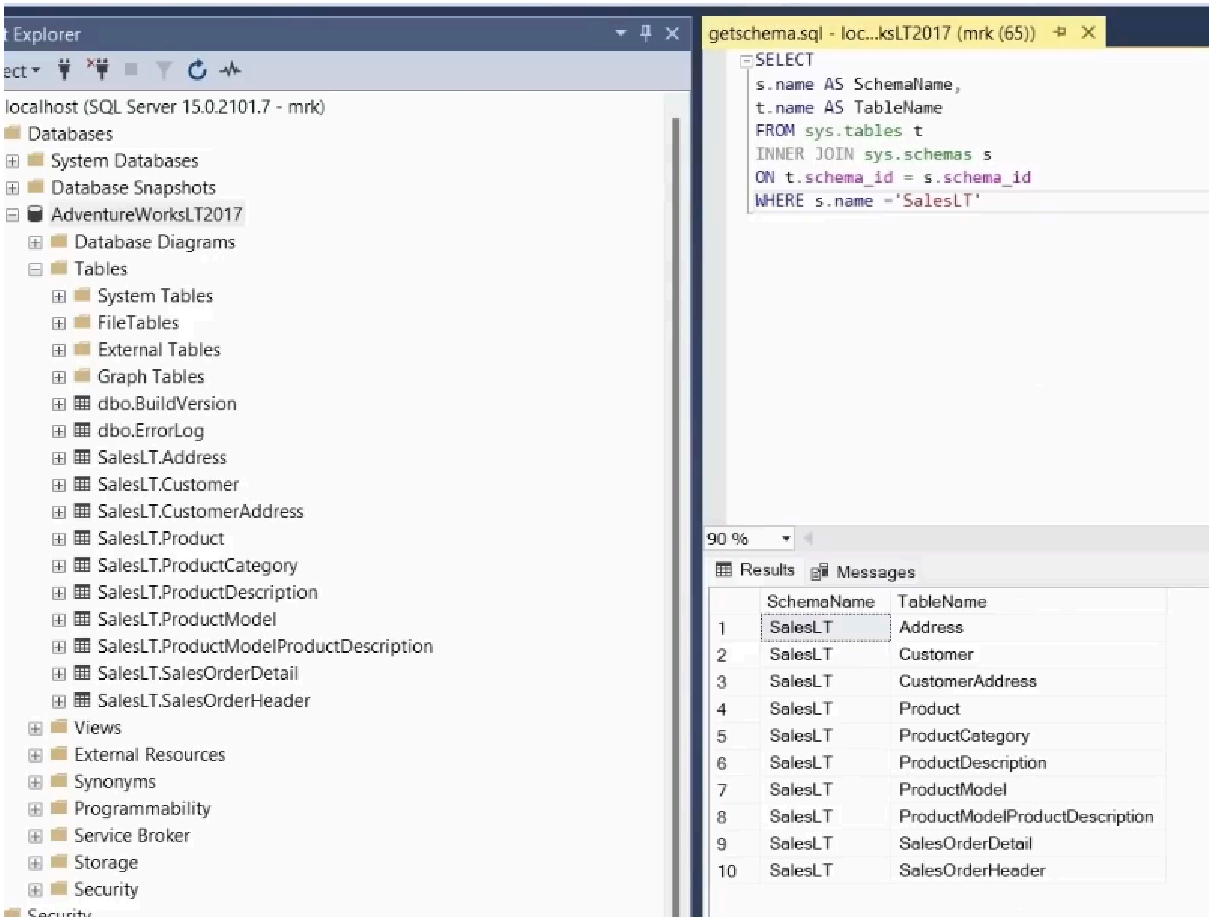
Creating the secret 'password'.
The secret 'password' has been successfully created.

The secret 'password' has been successfully created.

Name	Type	Status	Expiration date
password		✓ Enabled	
username		✓ Enabled	

Part 2 - Data Ingestion: From On-premise to Azure Data Lake Storage (Bronze container) (SQL DB to Parquet Format)

getschema.sql



```sql
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'
```

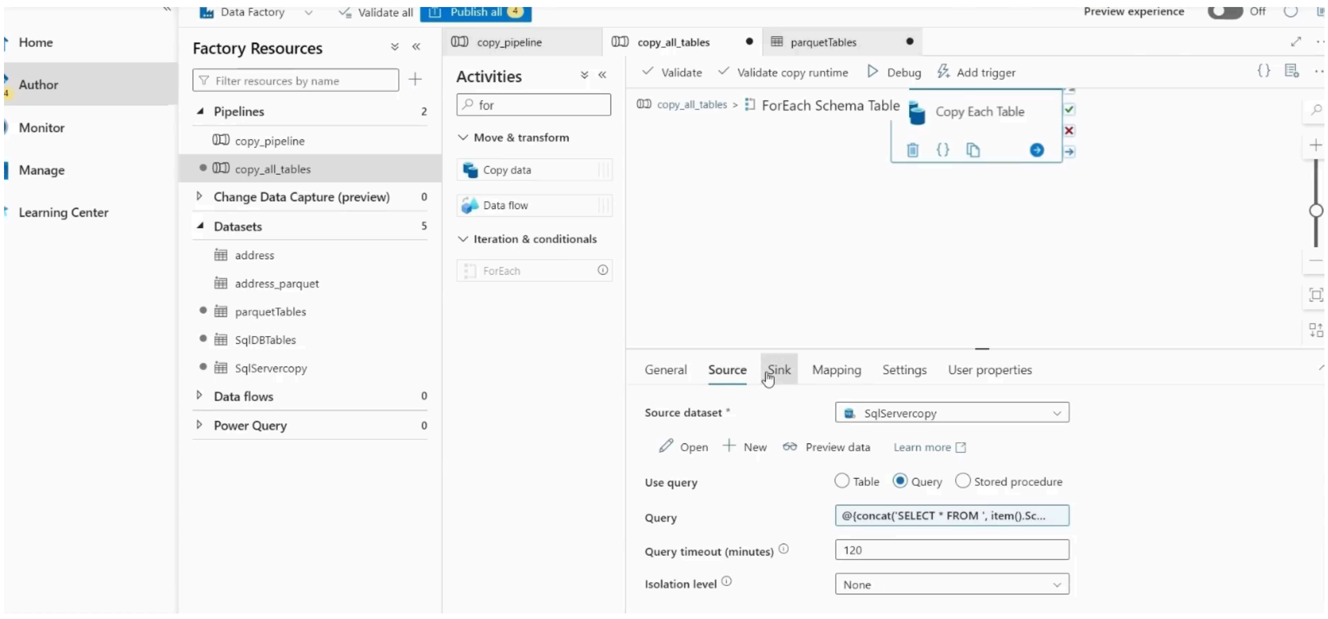
	SchemaName	TableName
1	SalesLT	Address
2	SalesLT	Customer
3	SalesLT	CustomerAddress
4	SalesLT	Product
5	SalesLT	ProductCategory
6	SalesLT	ProductDescription
7	SalesLT	ProductModel
8	SalesLT	ProductModelProductDescription
9	SalesLT	SalesOrderDetail
10	SalesLT	SalesOrderHeader

Folder Structure in Data Lake:

```
bronze/Schema/Tablename/Tablename.parquet
Ex. bronze/SalesLT/Address/Address.parquet
```

Mahesh Raut - Project - 31 May 2024

Source dataset - sql db


Source dataset * SqlServercopy

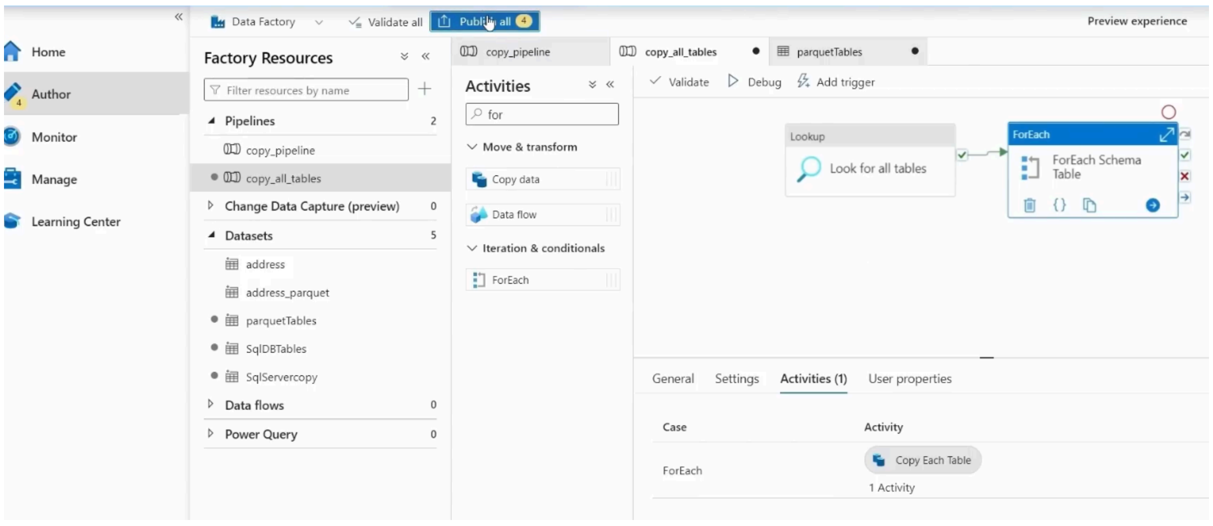
Use query: ( ) Table [✓ Query] ( ) Stored procedure

Query: @{concat('SELECT * FROM ', item().Sc...

Query timeout (minutes): 120

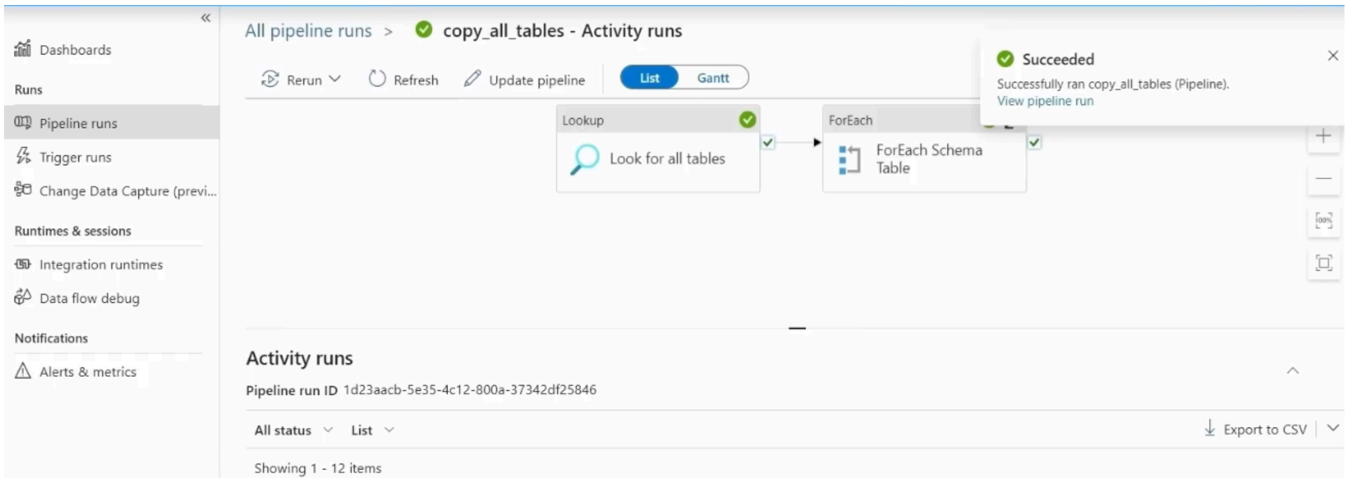
Isolation level: None

Pipeline build for looping each table in sql db to copy all tables in bronze container


Case	Activity
ForEach	Copy Each Table 1 Activity

Mahesh Raut - Project - 31 May 2024

Trigger pipeline once - copy all tables



All pipeline runs > copy_all_tables - Activity runs

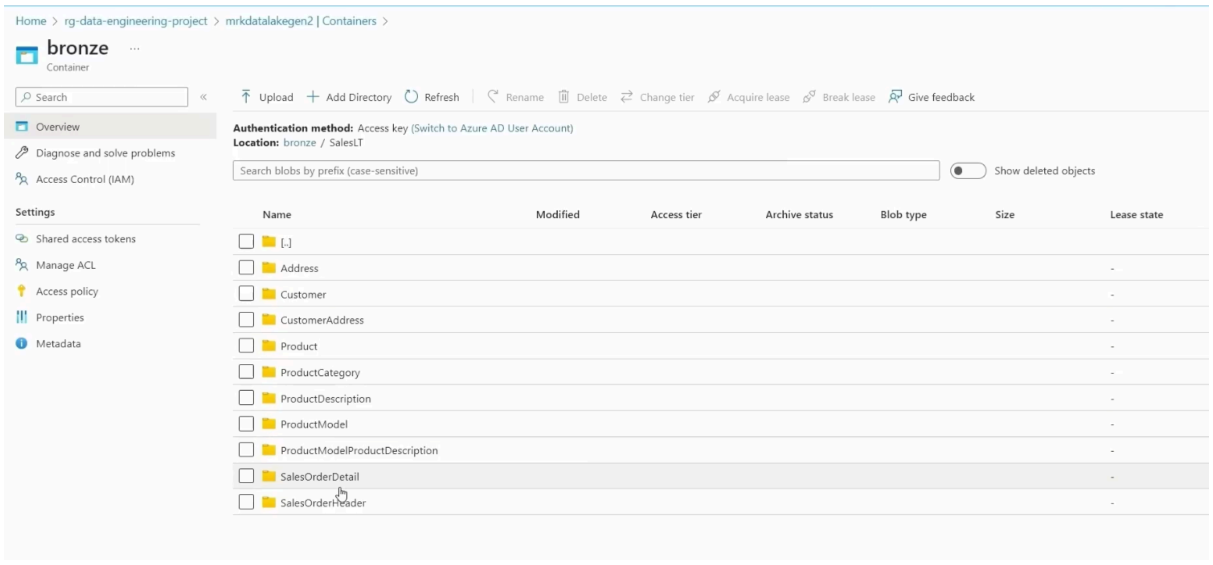
Succeeded
Successfully ran copy_all_tables (Pipeline).
View pipeline run

Activity runs
Pipeline run ID 1d23aacb-5e35-4c12-800a-37342df25846

All status · List · Export to CSV

Showing 1 - 12 items

Check bronze container

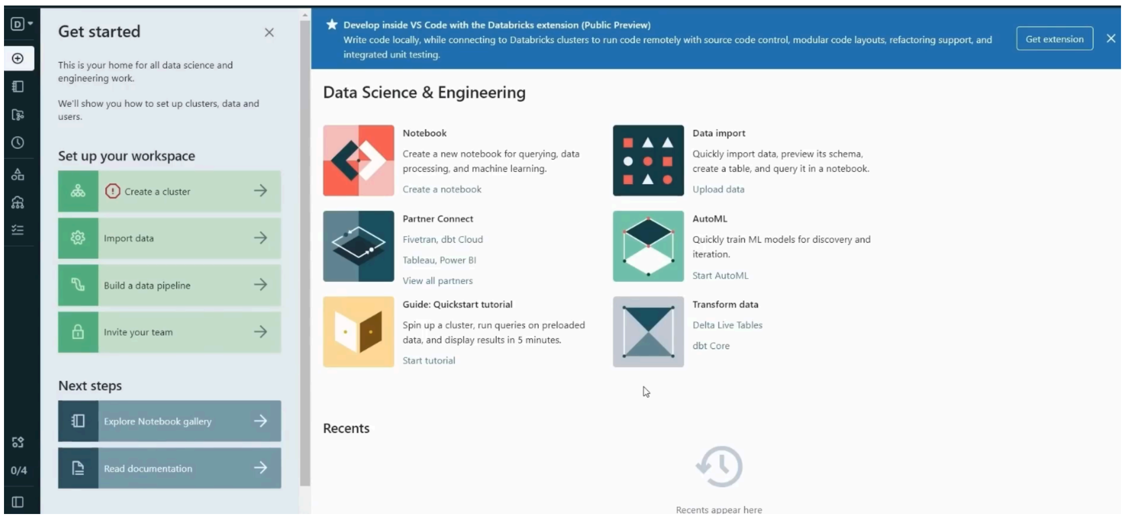


Authentication method: Access key (Switch to Azure AD User Account)
Location: bronze / SalesLT

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
Address						-
Customer						-
CustomerAddress						-
Product						-
ProductCategory						-
ProductDescription						-
ProductModel						-
ProductModelProductDescription						-
SalesOrderDetail						-
SalesOrderHeader						-

Part 3 - Data Transformation

Azure Databricks Overview



This is your home for all data science and engineering work.

We'll show you how to set up clusters, data and users.

Develop inside VS Code with the Databricks extension (Public Preview)
Write code locally, while connecting to Databricks clusters to run code remotely with source code control, modular code layouts, refactoring support, and integrated unit testing.

Notebook
Create a new notebook for querying, data processing, and machine learning.
Create a notebook

Data import
Quickly import data, preview its schema, create a table, and query it in a notebook.
Upload data

Partner Connect
Fivetran, dbt Cloud
Tableau, Power BI
View all partners

AutoML
Quickly train ML models for discovery and iteration.
Start AutoML

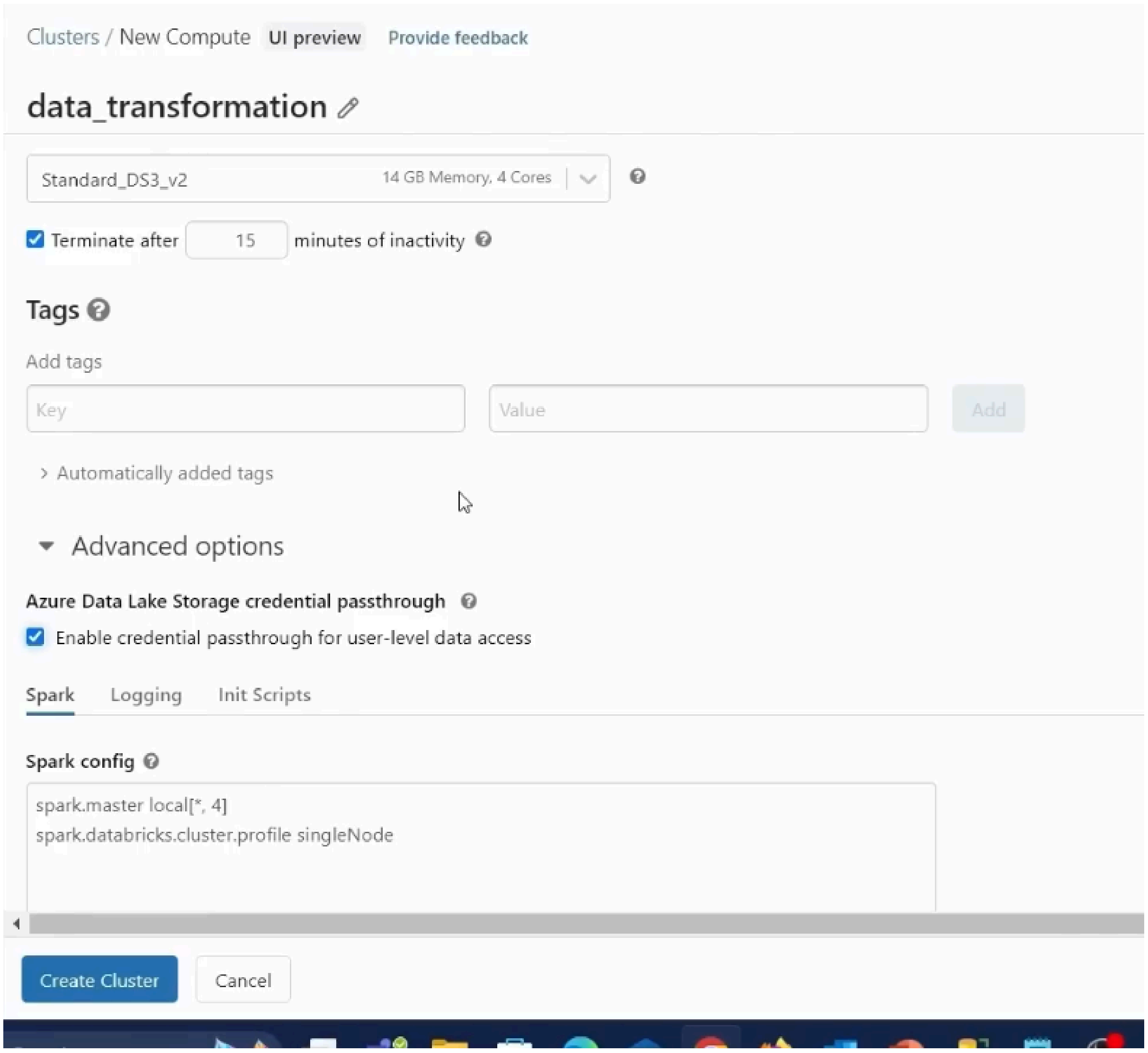
Guide: Quickstart tutorial
Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.
Start tutorial

Transform data
Delta Live Tables
dbt Core

Recents appear here

Mahesh Raut - Project - 31 May 2024

Create Compute Cluster


Clusters / New Compute UI preview Provide feedback

data_transformation

Standard_DS3_v2 14 GB Memory, 4 Cores

[✓] Terminate after 15 minutes of inactivity

Tags

Add tags

Key | Value | Add

› Automatically added tags

Advanced options

Azure Data Lake Storage credential passthrough

[✓] Enable credential passthrough for user-level data access

Spark | Logging | Init Scripts

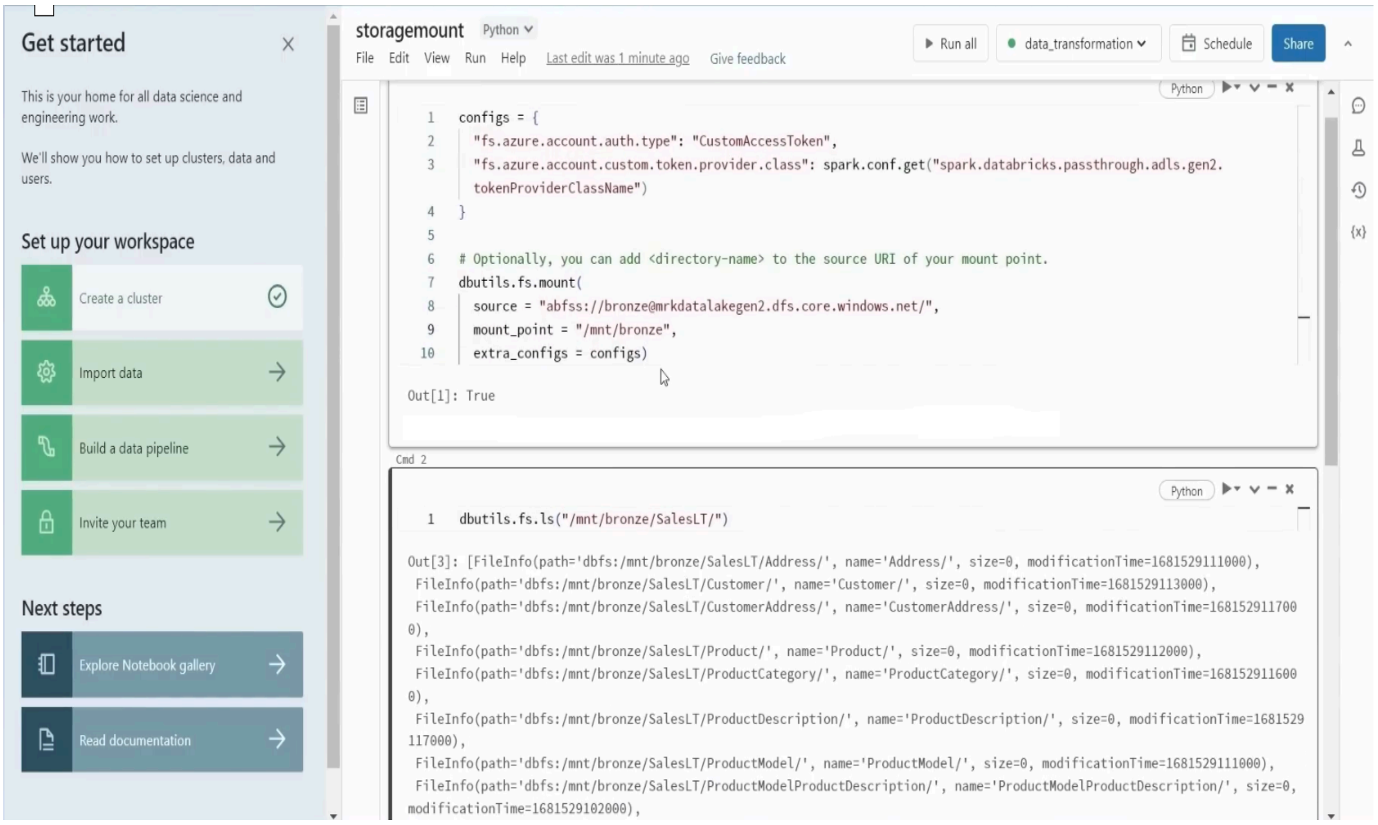
Spark config

```
spark.master local[*, 4]
spark.databricks.cluster.profile singleNode
```

Create Cluster | Cancel

Mahesh Raut - Project - 31 May 2024

Storagemount notebook - bronze, silver, gold



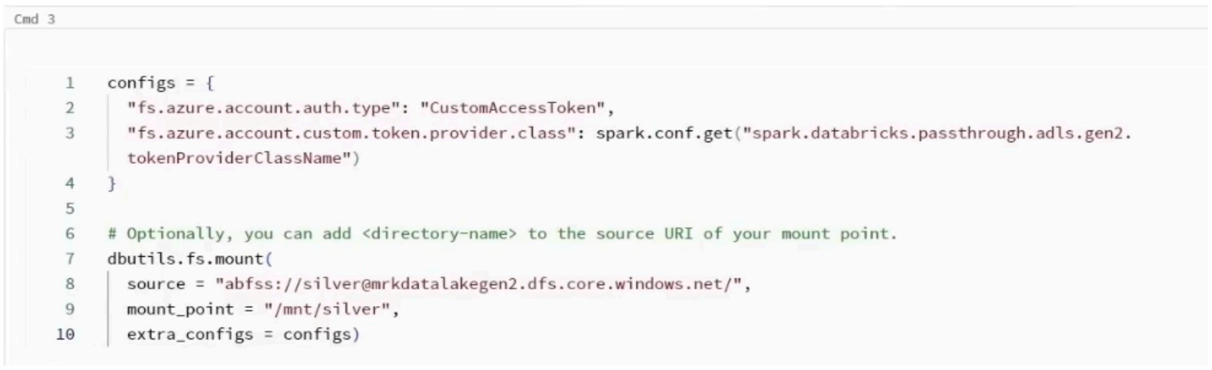
```
configs = {
  "fs.azure.account.auth.type": "CustomAccessToken",
  "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.
  tokenProviderClassName")
}

# Optionally, you can add <directory-name> to the source URI of your mount point.
dbutils.fs.mount(
  source = "abfss://bronze@mrkdatalakegen2.dfs.core.windows.net/",
  mount_point = "/mnt/bronze",
  extra_configs = configs)

Out[1]: True
```

Cmd 2

```
dbutils.fs.ls("/mnt/bronze/SalesLT/")

Out[3]: [FileInfo(path='dbfs:/mnt/bronze/SalesLT/Address/', name='Address/', size=0, modificationTime=1681529111000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/Customer/', name='Customer/', size=0, modificationTime=1681529113000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/CustomerAddress/', name='CustomerAddress/', size=0, modificationTime=168152911700
0),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/Product/', name='Product/', size=0, modificationTime=1681529112000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/ProductCategory/', name='ProductCategory/', size=0, modificationTime=168152911600
0),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/ProductDescription/', name='ProductDescription/', size=0, modificationTime=1681529
117000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/ProductModel/', name='ProductModel/', size=0, modificationTime=1681529111000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/ProductModelProductDescription/', name='ProductModelProductDescription/', size=0,
modificationTime=1681529102000),
```

```
configs = {
    "fs.azure.account.auth.type": "CustomAccessToken",
    "fs.azure.account.custom.token.provider.class":
spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClas
sName")
}

dbutils.fs.mount(
    source = "abfss://bronze@mrkdatakagen2.dfs.core.windows.net/",
    mount_point = "/mnt/bronze",
    extra_configs = configs
)

dbutils.fs.ls("/mnt/bronze/SalesLT/")
```

Silvermount

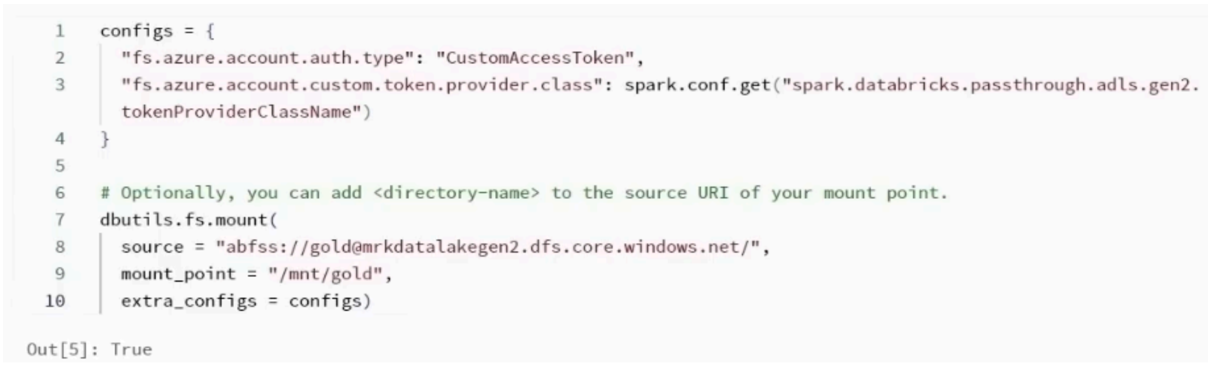


Cmd 3

```
configs = {
  "fs.azure.account.auth.type": "CustomAccessToken",
  "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.
  tokenProviderClassName")
}

# Optionally, you can add <directory-name> to the source URI of your mount point.
dbutils.fs.mount(
  source = "abfss://silver@mrkdatalakegen2.dfs.core.windows.net/",
  mount_point = "/mnt/silver",
  extra_configs = configs)
```

Mahesh Raut - Project - 31 May 2024

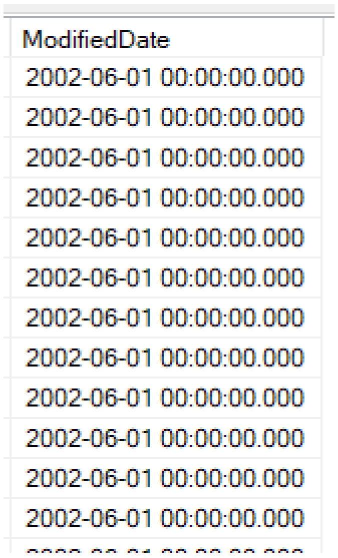
Goldmount



```
configs = {
  "fs.azure.account.auth.type": "CustomAccessToken",
  "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.
  tokenProviderClassName")
}

# Optionally, you can add <directory-name> to the source URI of your mount point.
dbutils.fs.mount(
  source = "abfss://gold@mrkdatalakegen2.dfs.core.windows.net/",
  mount_point = "/mnt/gold",
  extra_configs = configs)

Out[5]: True
```

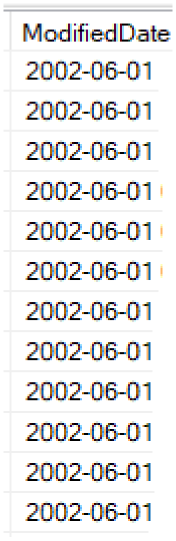
Bronze to silver notebook

Convert ModifiedDate column with Date Column



ModifiedDate
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000
2002-06-01 00:00:00.000

-------------- to —-------------



ModifiedDate
2002-06-01
2002-06-01
2002-06-01
2002-06-01
2002-06-01
2002-06-01
2002-06-01
2002-06-01
2002-06-01
2002-06-01
2002-06-01
2002-06-01

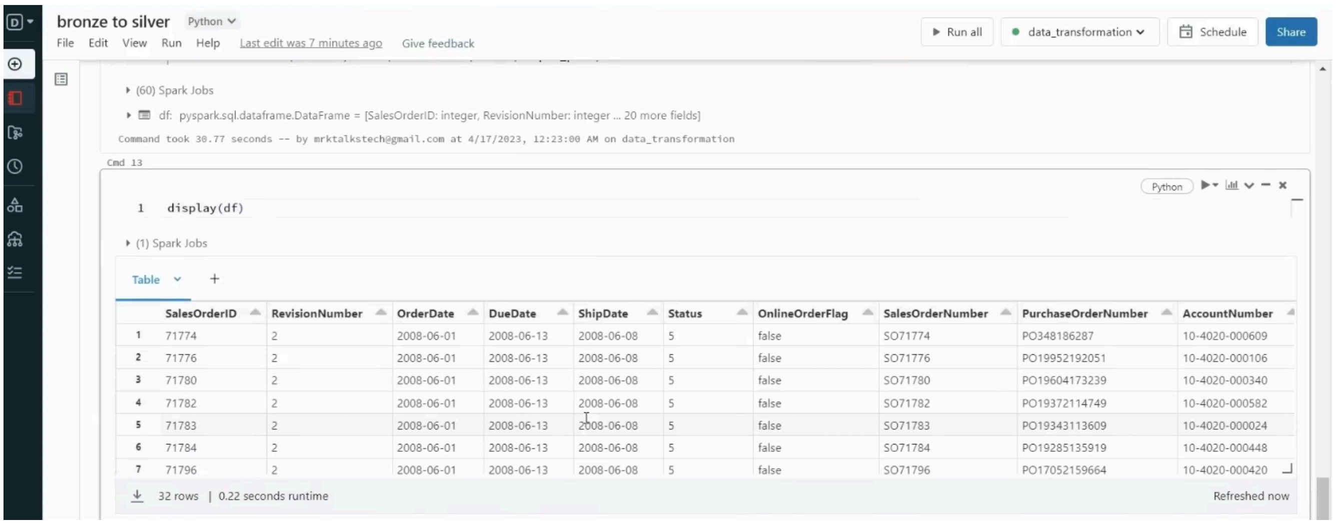
```
dbutils.fs.ls('mnt/bronze/SalesLT/')

dbutils.fs.ls('mnt/silver/')

input_path = '/mnt/bronze/SalesLT/Address/Address.parquet'

df = spark.read.format('parquet').load(input_path)

display(df)

from pyspark.sql.functions import from_utc_timestamp, date_format
from pyspark.sql.types import TimestampType

df = df.withColumn("ModifiedDate",
date_format(from_utc_timestamp(df["ModifiedDate"].cast(TimestampType()),
"UTC"), "yyyy-MM-dd"))
```

Doing transformation for all tables

```
table_name = []
for i in dbutils.fs.ls('mnt/bronze/SalesLT/'):
        table_name.append(i.name.split('/')[0])
```

```
from pyspark.sql.functions import from_utc_timestamp, date_format
from pyspark.sql.types import TimestampType

# Loop through table names
for i in table_name:
  # Construct path to the Parquet file
  path = '/mnt/bronze/SalesLT/' + i + '/' + i + '.parquet'

  # Read data from Parquet file
  df = spark.read.format('parquet').load(path)

  # Get list of columns in the DataFrame
  column = df.columns

  # Loop through each column
  for col in column:
    # Check if the column name contains "Date" or "date"
    if "Date" in col or "date" in col:
      # Convert the column to timestamp format (assuming it's not
already a timestamp)
      df = df.withColumn(col,
date_format(from_utc_timestamp(df[col].cast(TimestampType()), "UTC"),
"yyyy-MM-dd"))

  # Construct path to write the transformed data
  output_path = '/mnt/silver/SalesLT/' + i + '/'

  # Write the transformed DataFrame to Delta format in overwrite mode
  df.write.format('delta').mode("overwrite").save(output_path)

display(df)
```

Mahesh Raut - Project - 31 May 2024

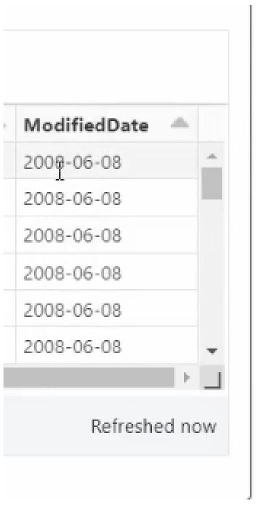


```
(60) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [SalesOrderID: integer, RevisionNumber: integer ... 20 more fields]
Command took 30.77 seconds -- by mrktalkstech@gmail.com at 4/17/2023, 12:23:00 AM on data_transformation
Cmd 13

display(df)

(1) Spark Jobs
```

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderNumber	PurchaseOrderNumber	AccountNumber
1	71774	2	2008-06-01	2008-06-13	2008-06-08	5	false	SO71774	PO348186287	10-4020-000609
2	71776	2	2008-06-01	2008-06-13	2008-06-08	5	false	SO71776	PO19952192051	10-4020-000106
3	71780	2	2008-06-01	2008-06-13	2008-06-08	5	false	SO71780	PO19604173239	10-4020-000340
4	71782	2	2008-06-01	2008-06-13	2008-06-08	5	false	SO71782	PO19372114749	10-4020-000582
5	71783	2	2008-06-01	2008-06-13	2008-06-08	5	false	SO71783	PO19343113609	10-4020-000024
6	71784	2	2008-06-01	2008-06-13	2008-06-08	5	false	SO71784	PO19285135919	10-4020-000448
7	71796	2	2008-06-01	2008-06-13	2008-06-08	5	false	SO71796	PO17052159664	10-4020-000420

32 rows | 0.22 seconds runtime   Refreshed now

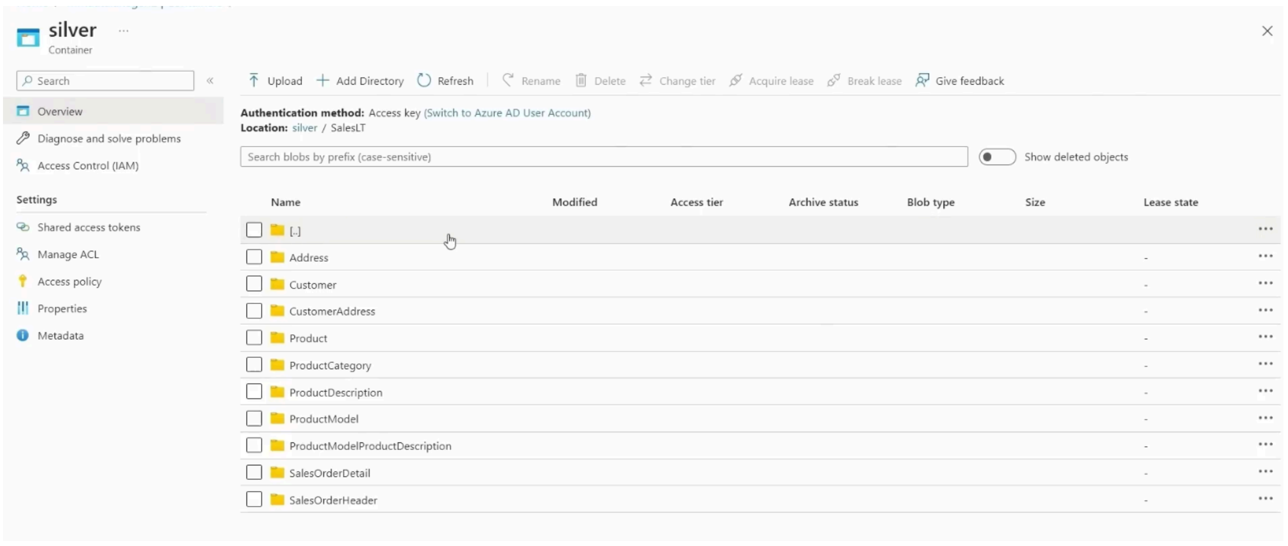
Updated column:



ModifiedDate
2008-06-08
2008-06-08
2008-06-08
2008-06-08
2008-06-08
2008-06-08

Refreshed now

Silver container

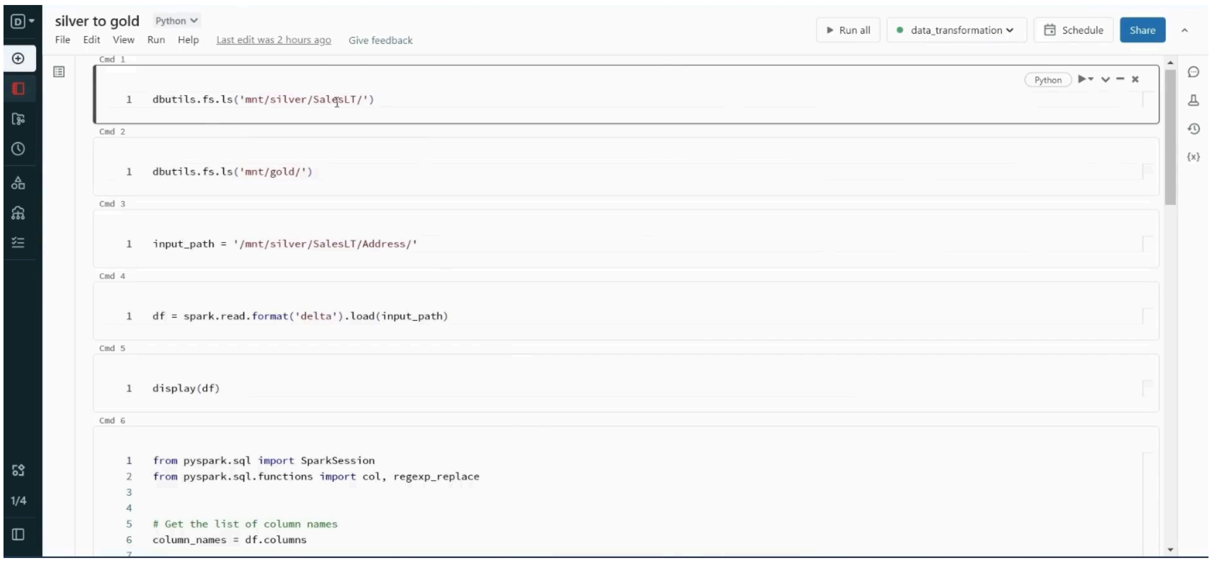


Authentication method: Access key (Switch to Azure AD User Account)
Location: silver / SalesLT

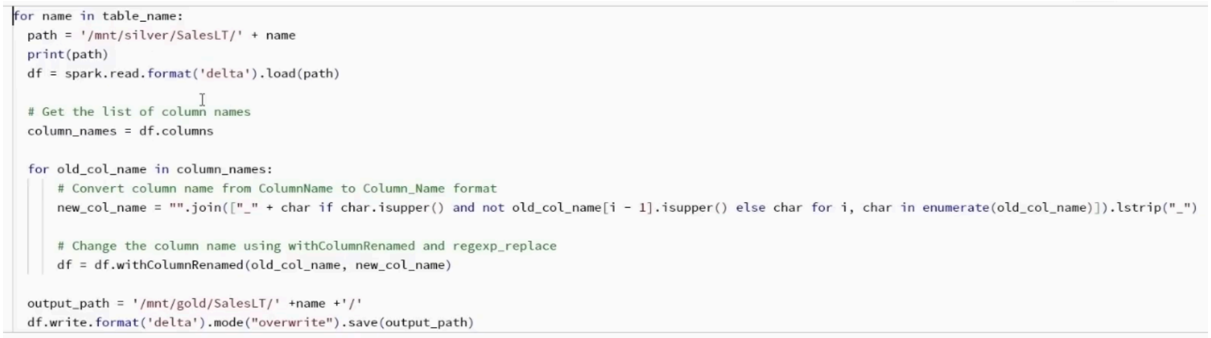
Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
Address						-
Customer						-
CustomerAddress						-
Product						-
ProductCategory						-
ProductDescription						-
ProductModel						-
ProductModelProductDescription						-
SalesOrderDetail						-
SalesOrderHeader						-

Mahesh Raut - Project - 31 May 2024

Silver to gold notebook
Transform the column names for all tables (ex. ProductId to Product_Id)



Cmd 1
```
dbutils.fs.ls('mnt/silver/SalesLT/')
```
Cmd 2
```
dbutils.fs.ls('mnt/gold/')
```
Cmd 3
```
input_path = '/mnt/silver/SalesLT/Address/'
```
Cmd 4
```
df = spark.read.format('delta').load(input_path)
```
Cmd 5
```
display(df)
```
Cmd 6
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, regexp_replace


# Get the list of column names
column_names = df.columns
```

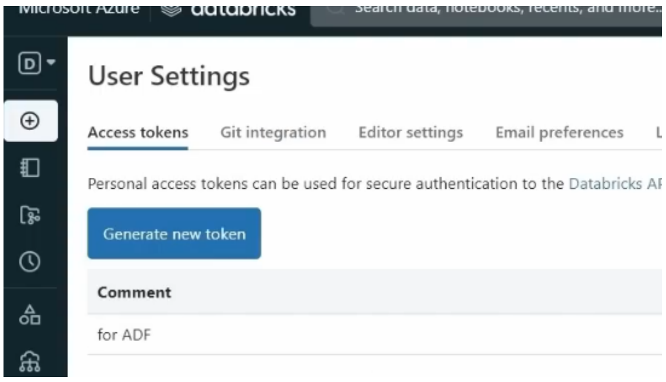


```
for name in table_name:
  path = '/mnt/silver/SalesLT/' + name
  print(path)
  df = spark.read.format('delta').load(path)

  # Get the list of column names
  column_names = df.columns

  for old_col_name in column_names:
      # Convert column name from ColumnName 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/gold/SalesLT/' +name +'/'
  df.write.format('delta').mode("overwrite").save(output_path)
```

```
table_name = []
for i in dbutils.fs.ls('mnt/silver/SalesLT/'):
  table_name.append(i.name.split('/')[0])
# Get list of column names in column_names
column_names = df.columns

# Loop through each column name
for old_col_name in column_names:
  # Convert column name from CamelCase to snake_case format
  new_col_name = "".join([word.lower() if word.isupper() and not i == 0
else word for i, word in enumerate(old_col_name)]).strip("_")

  # Change the column name using withColumnRenamed
  df = df.withColumnRenamed(old_col_name, new_col_name)

# Print transformed DataFrame
print(df)
```

Mahesh Raut - Project - 31 May 2024

Make connection with cluster in ADF

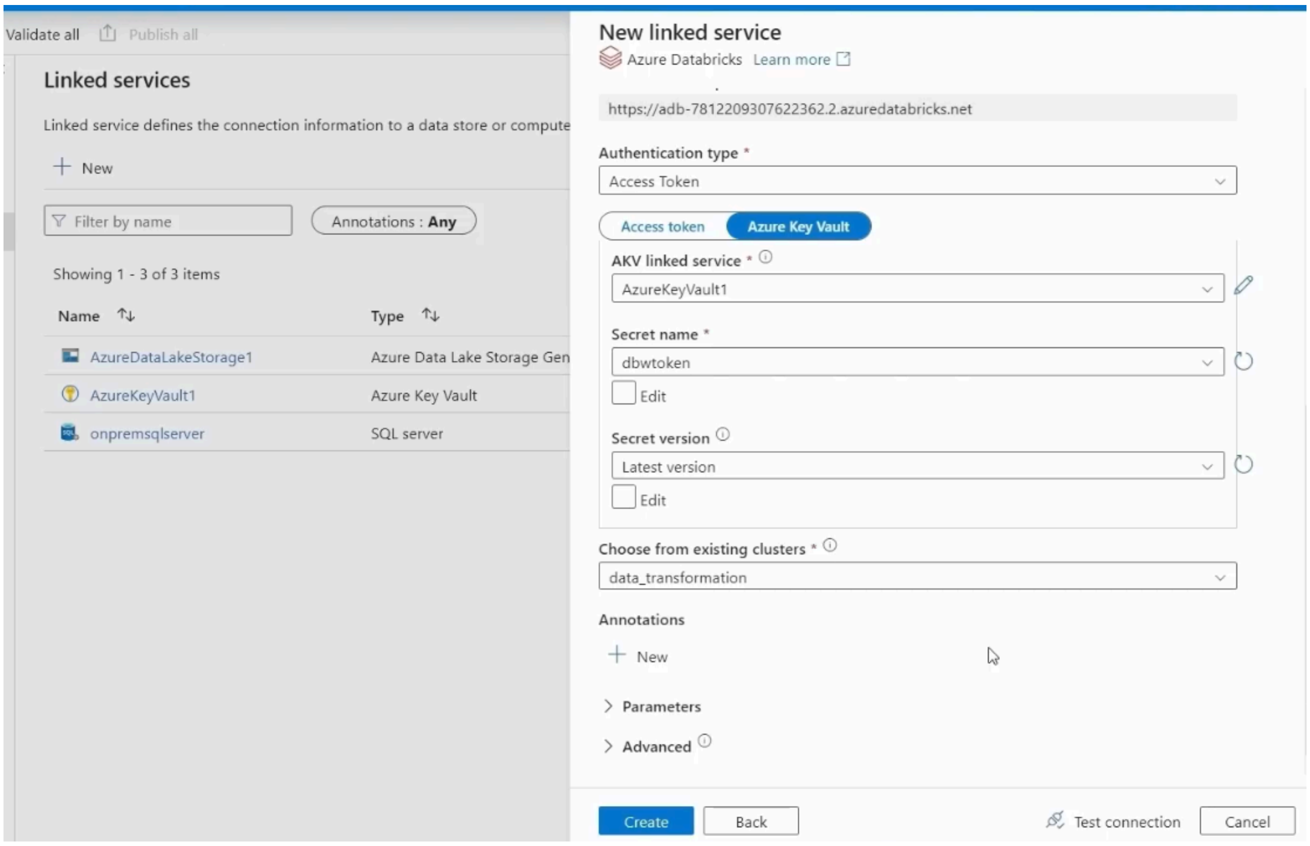

User Settings

Access tokens | Git integration | Editor settings | Email preferences

Personal access tokens can be used for secure authentication to the Databricks A

Generate new token

Comment
for ADF


Linked services

Linked service defines the connection information to a data store or compute

Showing 1 - 3 of 3 items

Name	Type
AzureDataLakeStorage1	Azure Data Lake Storage Gen
AzureKeyVault1	Azure Key Vault
onpremsqlserver	SQL server

New linked service

Azure Databricks Learn more

https://adb-7812209307622362.2.azuredatabricks.net

Authentication type *: Access Token

Access token [✓ Azure Key Vault]

AKV linked service *: AzureKeyVault1

Secret name *: dbwtoken

Secret version: Latest version

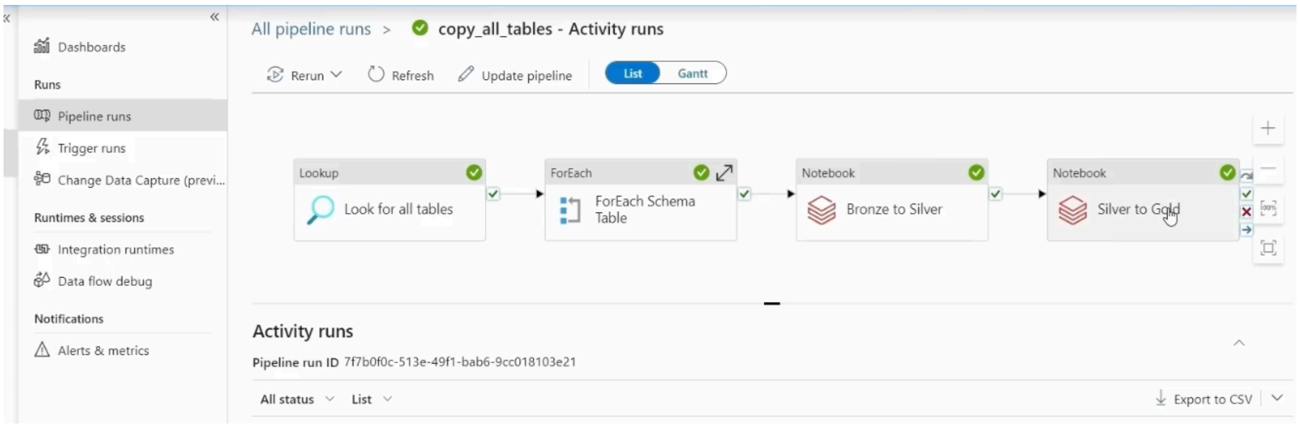
Choose from existing clusters *: data_transformation

Annotations

Parameters

Advanced

Create | Back | Test connection | Cancel


All pipeline runs > copy_all_tables - Activity runs

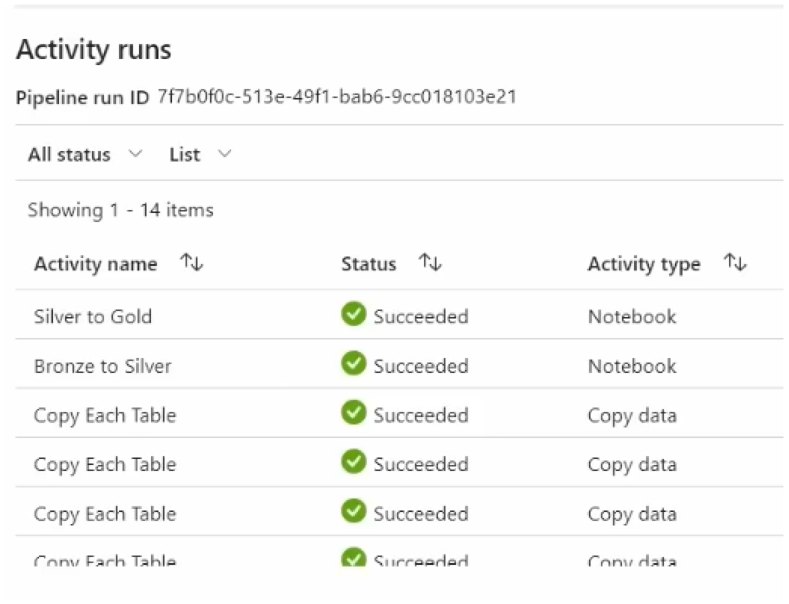
Rerun | Refresh | Update pipeline | List | Gantt

Lookup: Look for all tables → ForEach: ForEach Schema Table → Notebook: Bronze to Silver → Notebook: Silver to Gold

Activity runs

Pipeline run ID 7f7b0f0c-513e-49f1-bab6-9cc018103e21

Mahesh Raut - Project - 31 May 2024



Activity runs

Pipeline run ID 7f7b0f0c-513e-49f1-bab6-9cc018103e21

All status ∨ List ∨

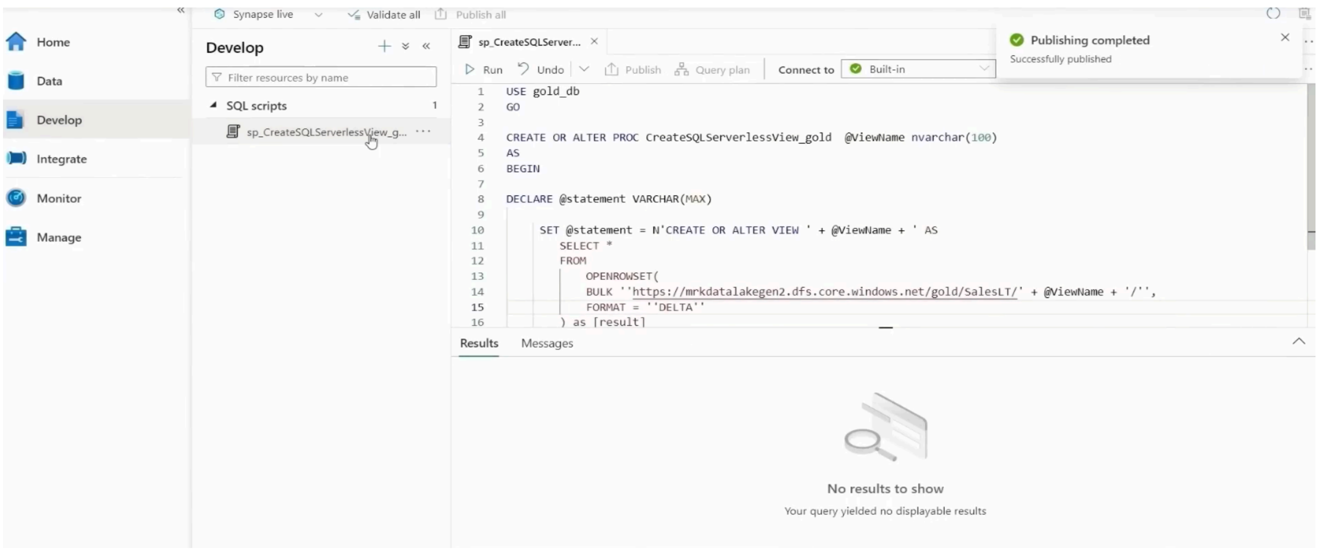
Showing 1 - 14 items

Activity name	Status	Activity type
Silver to Gold	✓ Succeeded	Notebook
Bronze to Silver	✓ Succeeded	Notebook
Copy Each Table	✓ Succeeded	Copy data
Copy Each Table	✓ Succeeded	Copy data
Copy Each Table	✓ Succeeded	Copy data
Copy Each Table	✓ Succeeded	Copy data

Part 4 - Data Loading - Gold container to Azure Synapse Serverless SQL Pool

```
USE gold_db
GO
CREATE OR ALTER PROC CreateSQLServerlessView_gold @ViewName nvarchar(100)
AS
BEGIN
DECLARE @statement VARCHAR (MAX)
SET @statement = N'CREATE OR ALTER VIEW ' + @ViewName + N' AS
                                SELECT *
                                FROM                          OPENROWSET (
                                                                       BULK
''https://mrkdatalakegen2.dfs.core.windows.net/gold/SalesLT/' + @ViewName + '/'',
                                                                       FORMAT =
''DELTA"
                                                                ) as [result]'

EXEC (@statement)
END
GO
```

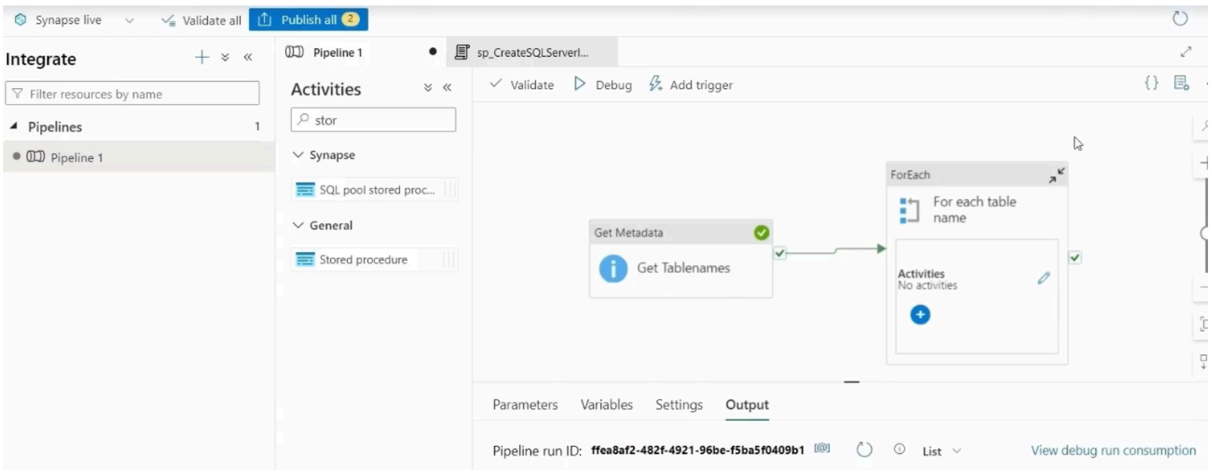
Mahesh Raut - Project - 31 May 2024



Publishing completed
Successfully published

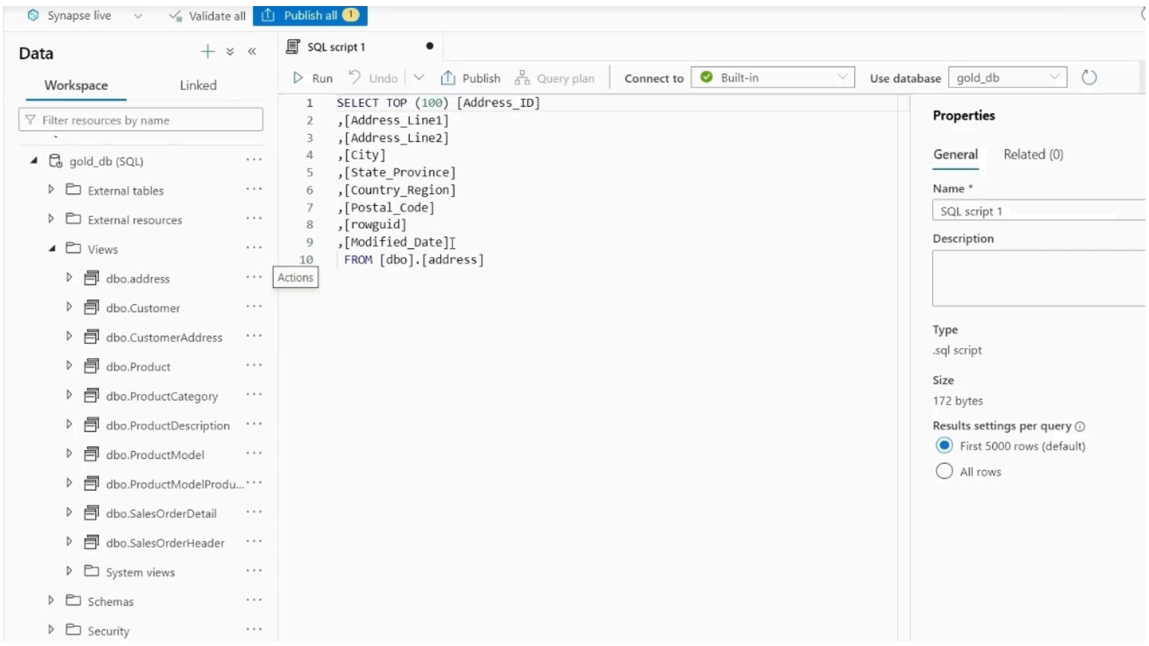
```
USE gold_db
GO

CREATE OR ALTER PROC CreateSQLServerlessView_gold  @ViewName nvarchar(100)
AS
BEGIN

DECLARE @statement VARCHAR(MAX)

    SET @statement = N'CREATE OR ALTER VIEW ' + @ViewName + ' AS
       SELECT *
       FROM
          OPENROWSET(
          BULK ''https://mrkdatalakegen2.dfs.core.windows.net/gold/SalesLT/' + @ViewName + '/'',
          FORMAT = ''DELTA''
       ) as [result]
```

No results to show
Your query yielded no displayable results



Pipeline run ID: ffea8af2-482f-4921-96be-f5ba5f0409b1

After trigger:



```
SELECT TOP (100) [Address_ID]
,[Address_Line1]
,[Address_Line2]
,[City]
,[State_Province]
,[Country_Region]
,[Postal_Code]
,[rowguid]
,[Modified_Date]
 FROM [dbo].[address]
```

Name *
SQL script 1
Description

Type
.sql script
Size
172 bytes
Results settings per query
[✓ First 5000 rows (default)]
All rows