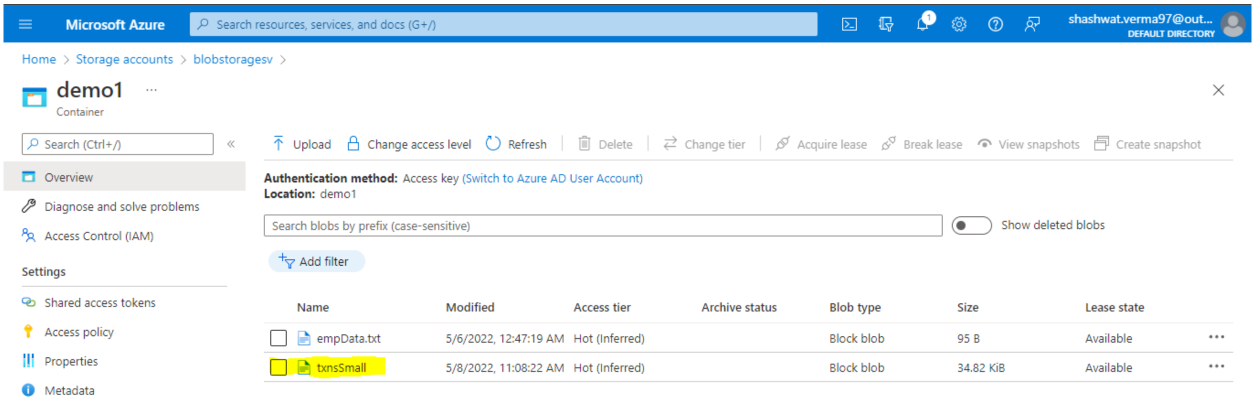
# Lab 3: Collect all data whose category is Gymnastics (Databricks)

**Dataset: txnsSmall**

**BlobStorage → ADLS → Databricks → ADLS → Synapse**

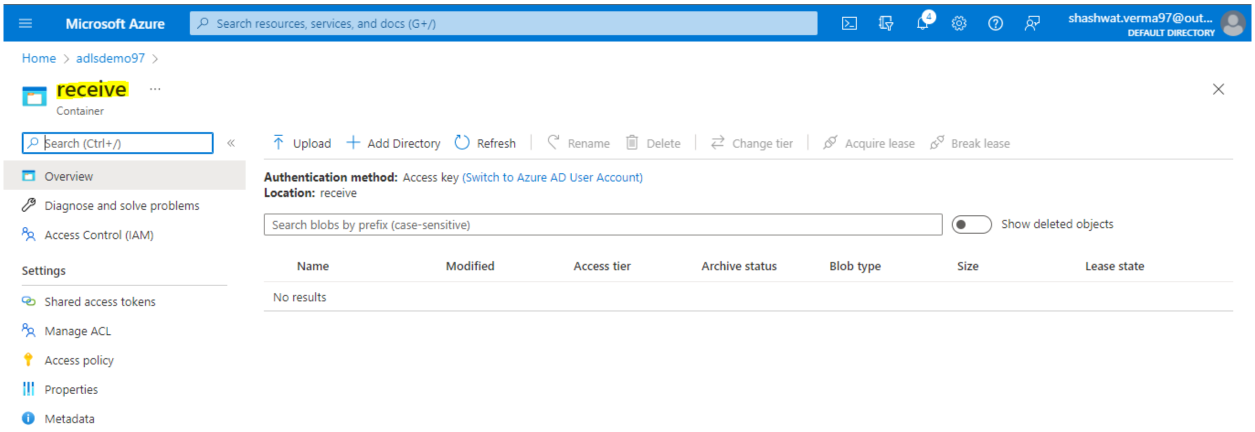
**1.** Upload ‘txnsSmall’ dataset in blob storage container ‘demo1’.



**Authentication method:** Access key (Switch to Azure AD User Account)
**Location:** demo1

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|---|---|---|---|---|---|---|
| empData.txt | 5/6/2022, 12:47:19 AM | Hot (Inferred) | | Block blob | 95 B | Available |
| txnsSmall | 5/8/2022, 11:08:22 AM | Hot (Inferred) | | Block blob | 34.82 KiB | Available |

**2.** We will use the ‘receive’ container in the adls storage ‘adlsdemo97’ to receive the copy data from blob storage.



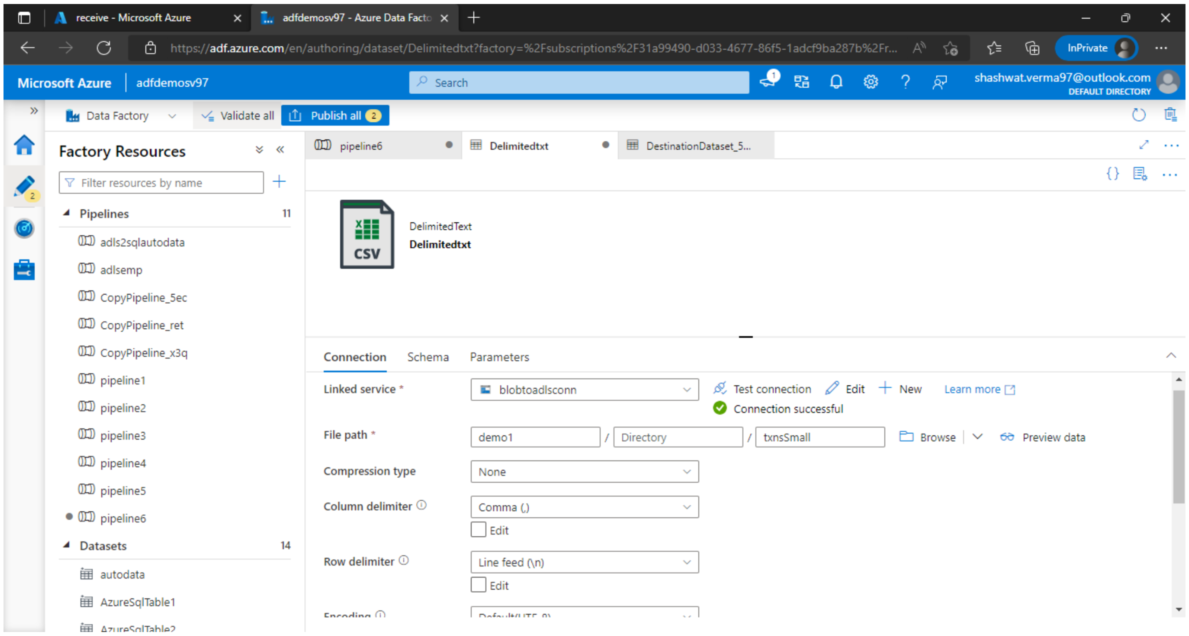
**Authentication method:** Access key (Switch to Azure AD User Account)
**Location:** receive

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|---|---|---|---|---|---|---|
| No results | | | | | | |

**3.** Created pipeline ‘blob2adls’ to copy the data from blob to adls

<u>Source configuration:</u>

We will use previously created source configuration ‘Delimitedtxt’ as source for the pipeline



DelimitedText
Delimitedtxt

Linked service * blobtoadlsconn

Connection successful

File path * demo1 / Directory / txnsSmall

Compression type None

Column delimiter Comma (,)

□ Edit

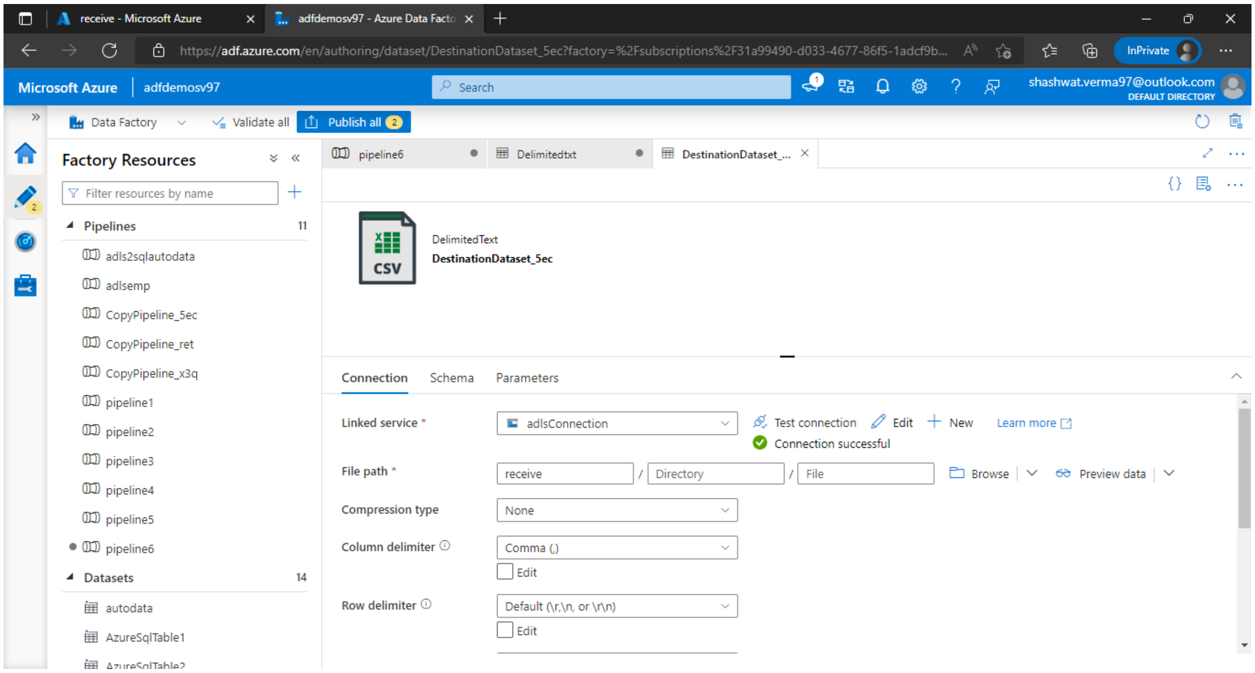
Row delimiter Line feed (\n)

□ Edit

Encoding Default(UTF-8)

## Sink configuration:

We will use previously created ‘DestinationDataset_5ec’ as sink configuration which uses ‘receive’ container as destination location to store copy data



DelimitedText
DestinationDataset_5ec

Linked service * adlsConnection

Connection successful

File path * receive / Directory / File

Compression type None

Column delimiter Comma (,)

□ Edit

Row delimiter Default (\r,\n, or \r\n)

□ Edit

**4.** Created a python notebook in Azure Databricks - ‘filterGymnastics’ in ‘mytestcluster9706’. To store filtered output obtained through this notebook, we created a container ‘databricksoutput’ in ADLS storage ‘adlsdemo97’

<u>Source Code:</u>

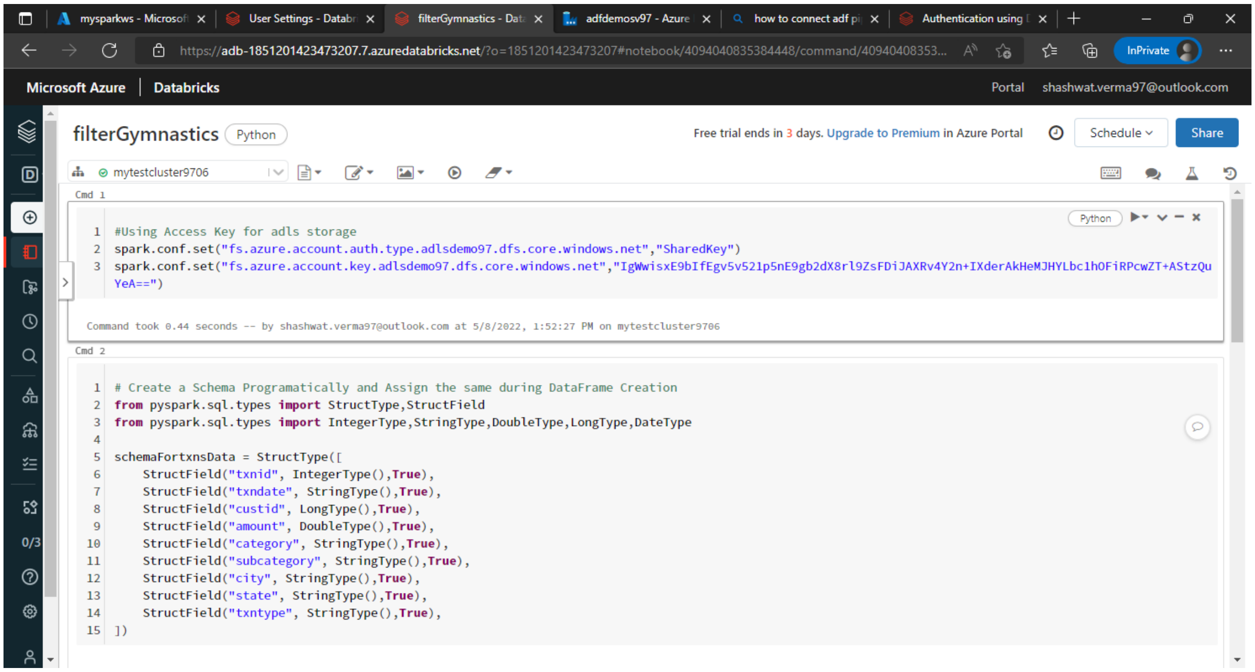
```
#Using Access Key for adls storage
spark.conf.set("fs.azure.account.auth.type.adlsdemo97.dfs.core.windows.net","SharedKey")
spark.conf.set("fs.azure.account.key.adlsdemo97.dfs.core.windows.net","IgWwisxE9bIfEgv5v521p5nE9gb2dX8rl9ZsFDiJAXRv4Y2n+IXderAkHeMJHYLbc1hOFiRPcwZT+AStzQuYeA==")

# Create a Schema Programatically and Assign the same during DataFrame Creation
from pyspark.sql.types import StructType,StructField
from pyspark.sql.types import IntegerType,StringType,DoubleType,LongType,DateType

schemaFortxnsData = StructType([
   StructField("txnid", IntegerType(),True),
   StructField("txndate", StringType(),True),
   StructField("custid", LongType(),True),
   StructField("amount", DoubleType(),True),
   StructField("category", StringType(),True),
   StructField("subcategory", StringType(),True),
   StructField("city", StringType(),True),
   StructField("state", StringType(),True),
   StructField("txntype", StringType(),True),
])

# Assigning the Schema
txnsDataDF =
spark.read.schema(schemaFortxnsData).option("delimiter",',').csv('abfss://receive@adlsdemo97.dfs.core.windows.net/')

txnsDataDF.filter("category =
'Gymnastics'").repartition(1).write.option("header",True).csv("abfss://databricksoutput@adlsdemo97.dfs.core.windows.net/gymnasticsoutput")
```



Cmd 1

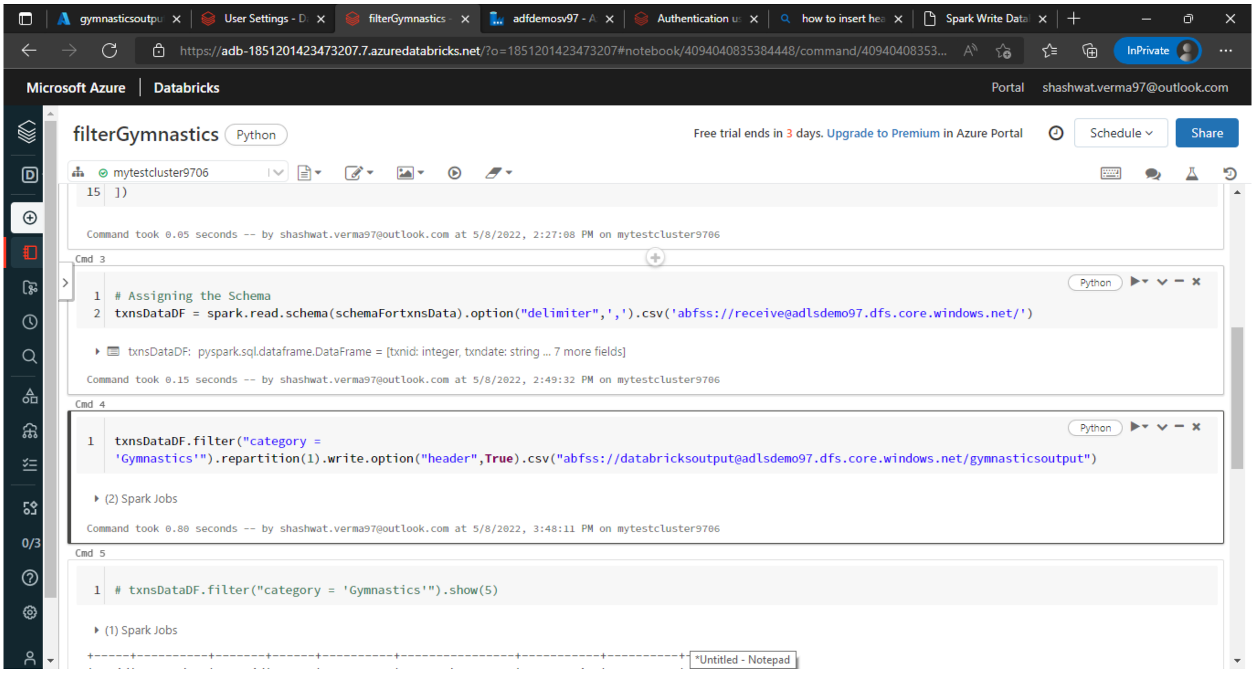
```python
#Using Access Key for adls storage
spark.conf.set("fs.azure.account.auth.type.adlsdemo97.dfs.core.windows.net","SharedKey")
spark.conf.set("fs.azure.account.key.adlsdemo97.dfs.core.windows.net","IgWwisxE9bIfEgv5v521p5nE9gb2dX8rl9ZsFDiJAXRv4Y2n+IXderAkHeMJHYLbc1hOFiRPcwZT+AStzQuYeA==")
```

```
Command took 0.44 seconds -- by shashwat.verma97@outlook.com at 5/8/2022, 1:52:27 PM on mytestcluster9706
```

Cmd 2

```python
# Create a Schema Programatically and Assign the same during DataFrame Creation
from pyspark.sql.types import StructType,StructField
from pyspark.sql.types import IntegerType,StringType,DoubleType,LongType,DateType

schemaFortxnsData = StructType([
    StructField("txnid", IntegerType(),True),
    StructField("txndate", StringType(),True),
    StructField("custid", LongType(),True),
    StructField("amount", DoubleType(),True),
    StructField("category", StringType(),True),
    StructField("subcategory", StringType(),True),
    StructField("city", StringType(),True),
    StructField("state", StringType(),True),
    StructField("txntype", StringType(),True),
])
```



```python
])
```

```
Command took 0.05 seconds -- by shashwat.verma97@outlook.com at 5/8/2022, 2:27:08 PM on mytestcluster9706
```

Cmd 3

```python
# Assigning the Schema
txnsDataDF = spark.read.schema(schemaFortxnsData).option("delimiter",',').csv('abfss://receive@adlsdemo97.dfs.core.windows.net/')
```

```
txnsDataDF:  pyspark.sql.dataframe.DataFrame = [txnid: integer, txndate: string ... 7 more fields]
Command took 0.15 seconds -- by shashwat.verma97@outlook.com at 5/8/2022, 2:49:32 PM on mytestcluster9706
```

Cmd 4

```python
txnsDataDF.filter("category = 'Gymnastics'").repartition(1).write.option("header",True).csv("abfss://databricksoutput@adlsdemo97.dfs.core.windows.net/gymnasticsoutput")
```

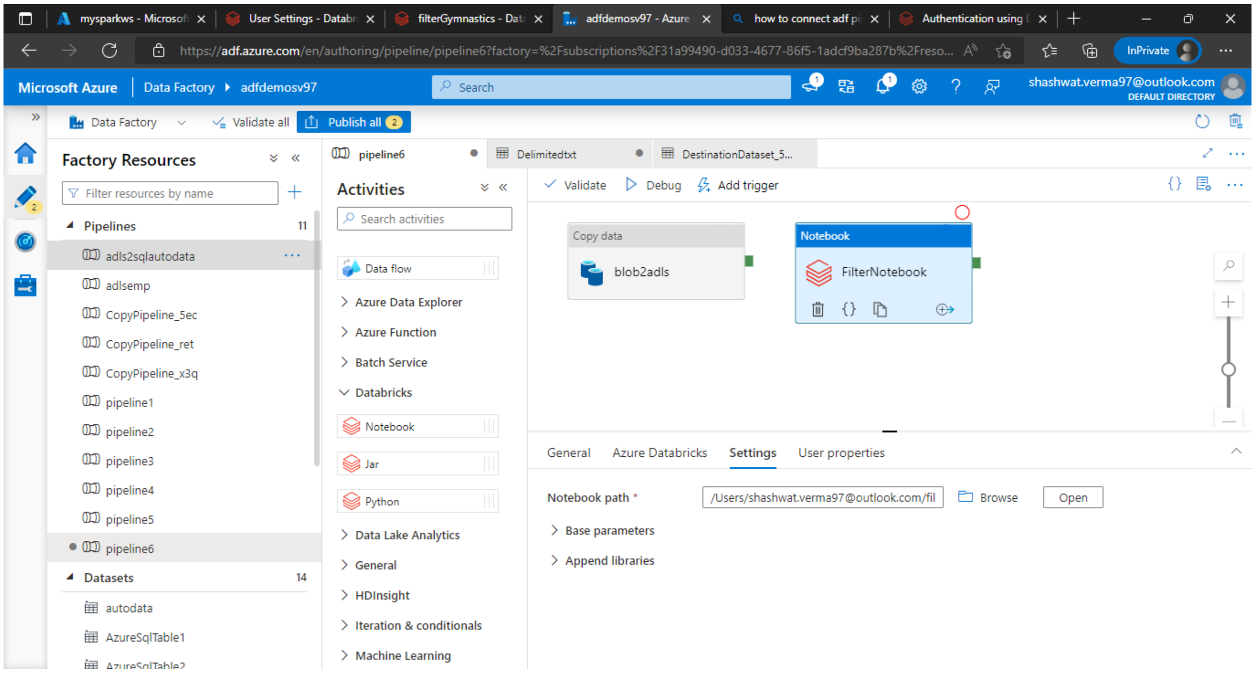
```
(2) Spark Jobs
Command took 0.80 seconds -- by shashwat.verma97@outlook.com at 5/8/2022, 3:48:11 PM on mytestcluster9706
```

Cmd 5

```python
# txnsDataDF.filter("category = 'Gymnastics'").show(5)
```

```
(1) Spark Jobs
+-----+----------+-------+------+----------+----------------+-----------+----------+
```

**5.** After creating the notebook we will add a Databricks notebook to the pipeline with following configurations:

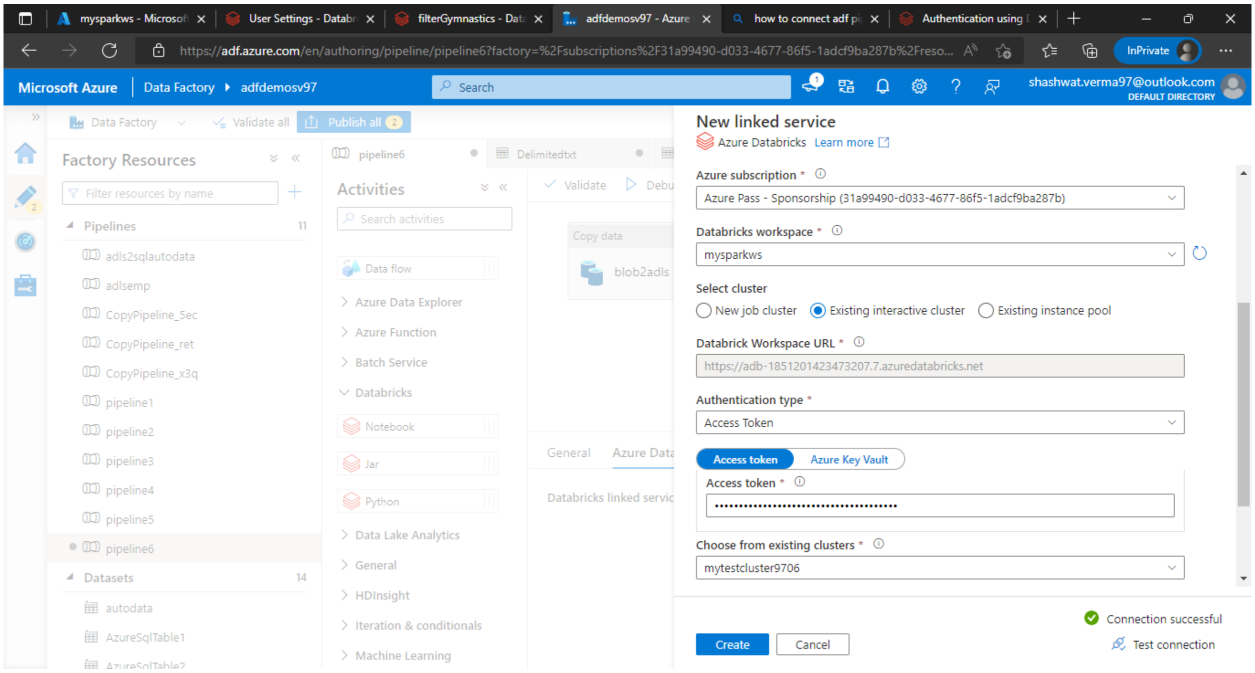


General | Azure Databricks | **Settings** | User properties

Notebook path * /Users/shashwat.verma97@outlook.com/fil Browse Open

Base parameters

Append libraries



New linked service

Azure Databricks Learn more

Azure subscription *
Azure Pass - Sponsorship (31a99490-d033-4677-86f5-1adcf9ba287b)

Databricks workspace *
mysparkws

Select cluster
○ New job cluster ● Existing interactive cluster ○ Existing instance pool

Databrick Workspace URL *
https://adb-1851201423473207.7.azuredatabricks.net

Authentication type *
Access Token

Access token | Azure Key Vault

Access token *
••••••••••••••••••••••••••••••••••••

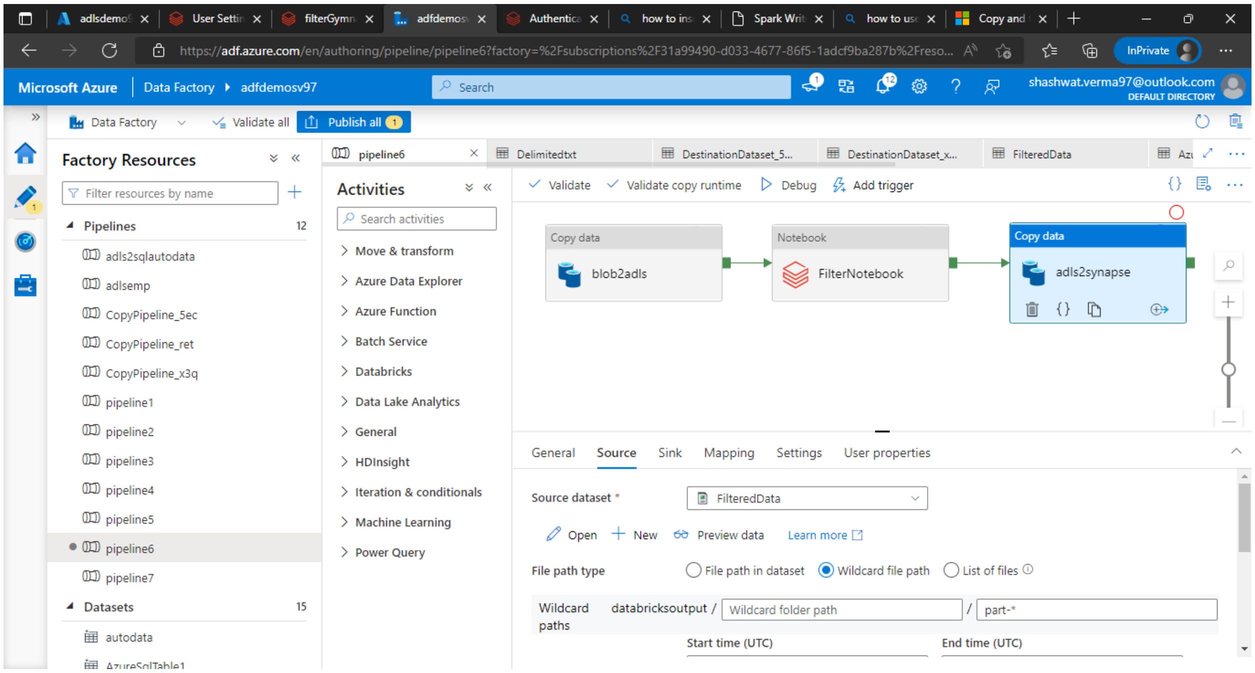
Choose from existing clusters *
mytestcluster9706

✓ Connection successful

Create Cancel Test connection

**6.** Now we have to fetch this filtered data from ADLS container ‘databricksoutput/gymnasticsoutput’ to Synapse. For that another copy data configuration is created.

# Source Configurations:



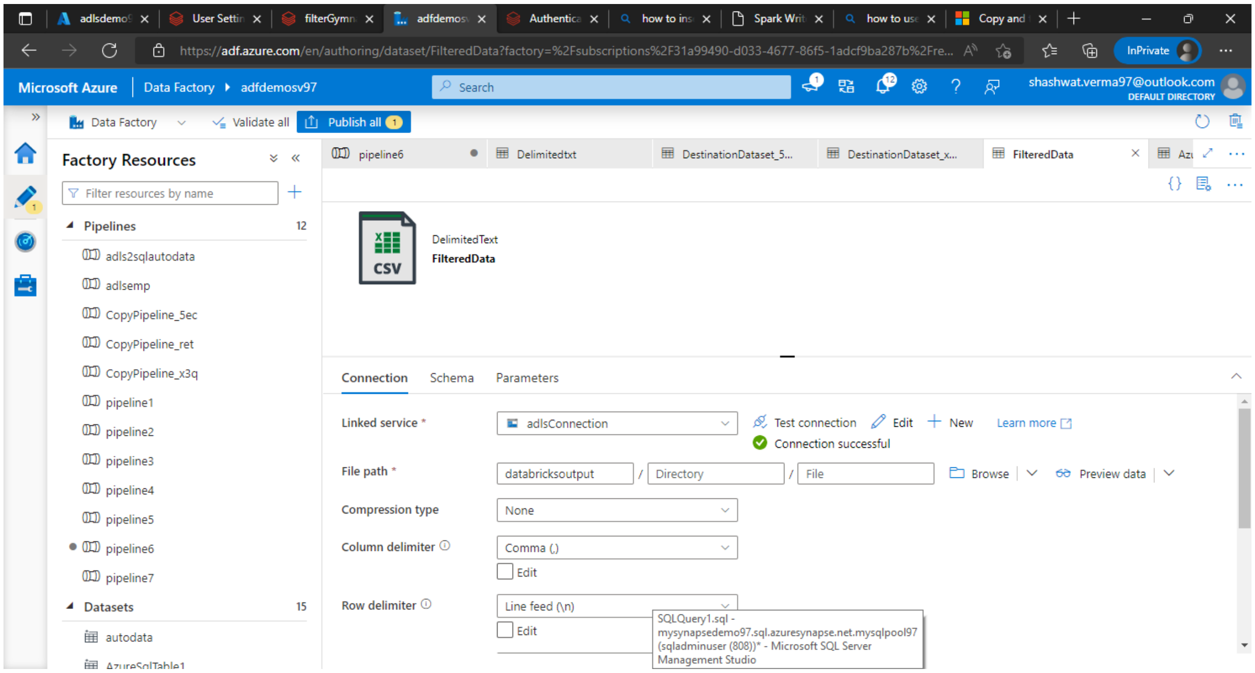
Source dataset * FilteredData

Open | New | Preview data | Learn more

File path type ○ File path in dataset ● Wildcard file path ○ List of files

Wildcard paths databricksoutput / Wildcard folder path / part-*

Start time (UTC) End time (UTC)



Linked service * adlsConnection

Test connection | Edit | New | Learn more

Connection successful

File path * databricksoutput / Directory / File

Browse | Preview data

Compression type None

Column delimiter Comma (,)

□ Edit

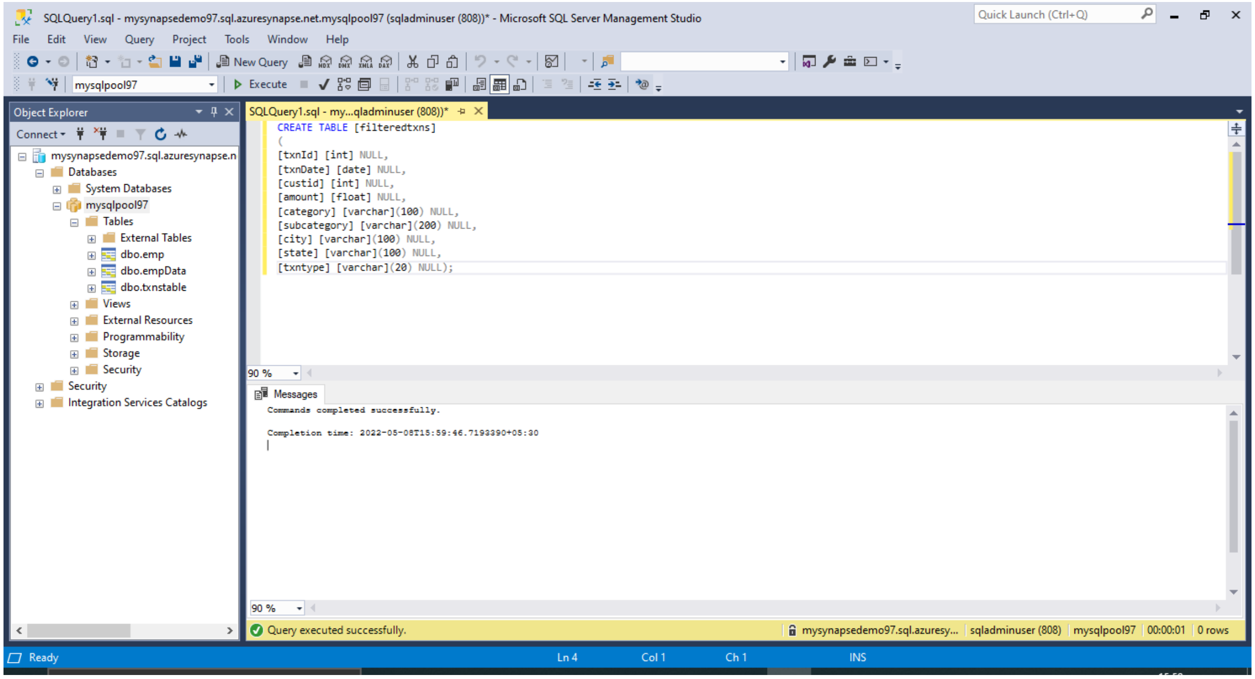
Row delimiter Line feed (\n)

□ Edit

SQLQuery1.sql - mysynapsedemo97.sql.azuresynapse.net.mysqlpool97 (sqladminuser (808))* - Microsoft SQL Server Management Studio

## Sink Configurations:

a. Created a table ‘filteredtxns’ in Synapse SQLPool


SQLQuery1.sql - mysynapsedemo97.sql.azuresynapse.net.mysqlpool97 (sqladminuser (808))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
mysqlpool97
Object Explorer
Connect
mysynapsedemo97.sql.azuresynapse.n
Databases
System Databases
mysqlpool97
Tables
External Tables
dbo.emp
dbo.empData
dbo.txnstable
Views
External Resources
Programmability
Storage
Security
Security
Integration Services Catalogs
SQLQuery1.sql - my...qladminuser (808))*
90 %
Messages
Ready
Ln 4
Col 1
Ch 1
INS

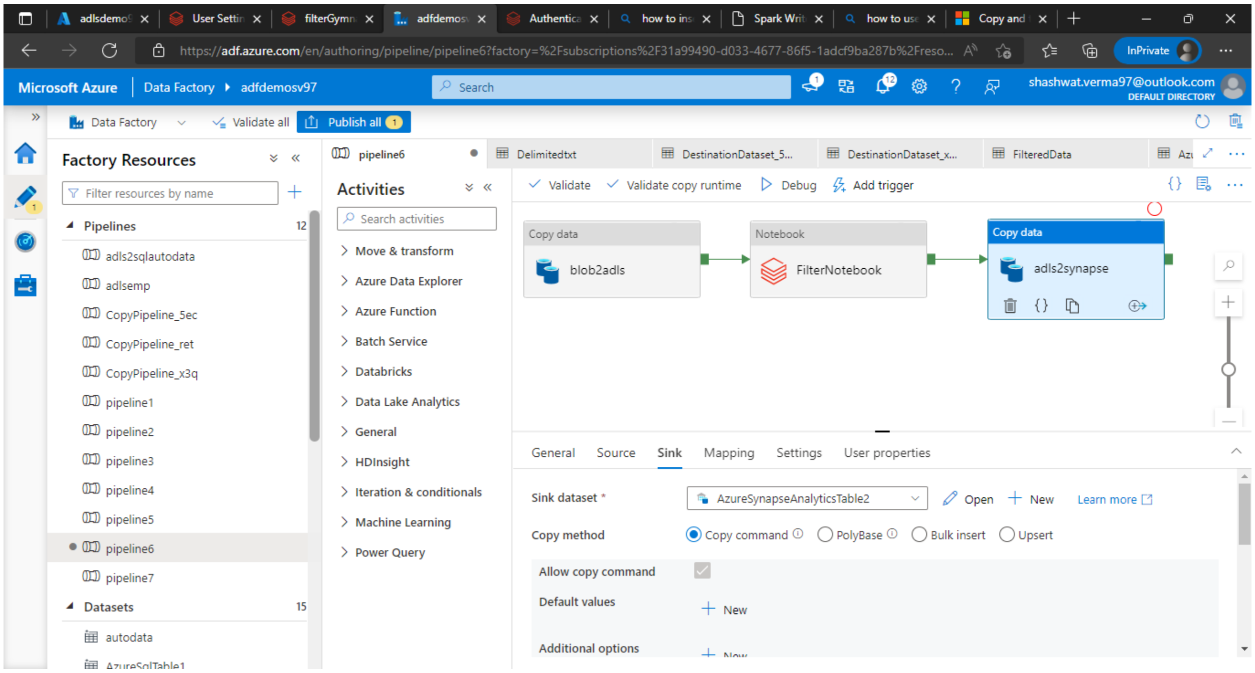
```
CREATE TABLE [filteredtxns]
(
[txnId] [int] NULL,
[txnDate] [date] NULL,
[custid] [int] NULL,
[amount] [float] NULL,
[category] [varchar](100) NULL,
[subcategory] [varchar](200) NULL,
[city] [varchar](100) NULL,
[state] [varchar](100) NULL,
[txntype] [varchar](20) NULL);
```

```
Commands completed successfully.

Completion time: 2022-05-08T15:59:46.7193390+05:30
```

Query executed successfully. | mysynapsedemo97.sql.azuresy... | sqladminuser (808) | mysqlpool97 | 00:00:01 | 0 rows

b. Selected Synapse SQLPool table dbo.filteredtxns



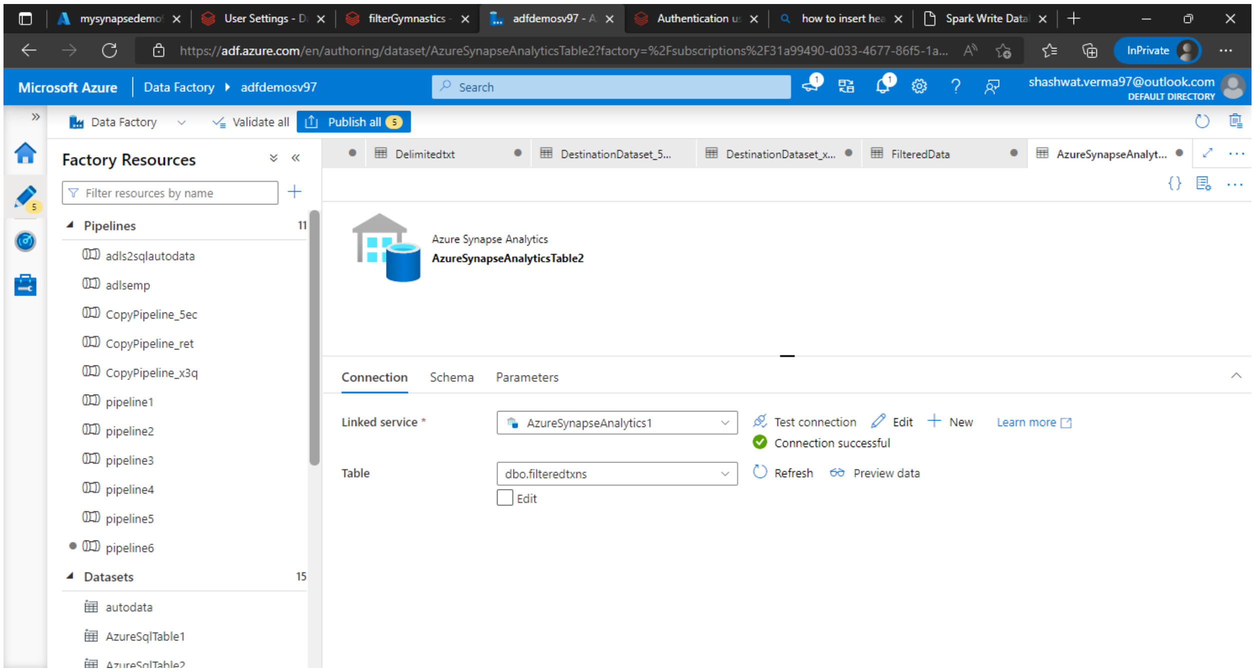
Sink dataset * AzureSynapseAnalyticsTable2 | Open | New | Learn more

Copy method ● Copy command ○ PolyBase ○ Bulk insert ○ Upsert

Allow copy command ☑

Default values + New

Additional options + New



https://adf.azure.com/en/authoring/dataset/AzureSynapseAnalyticsTable2?factory=%2Fsubscriptions%2F31a99490-d033-4677-86f5-1a...

| Pipelines | 11 |
|---|---|
| adls2sqlautodata | |
| adlsemp | |
| CopyPipeline_5ec | |
| CopyPipeline_ret | |
| CopyPipeline_x3q | |
| pipeline1 | |
| pipeline2 | |
| pipeline3 | |
| pipeline4 | |
| pipeline5 | |
| pipeline6 | |
| **Datasets** | **15** |
| autodata | |
| AzureSqlTable1 | |
| AzureSqlTable2 | |

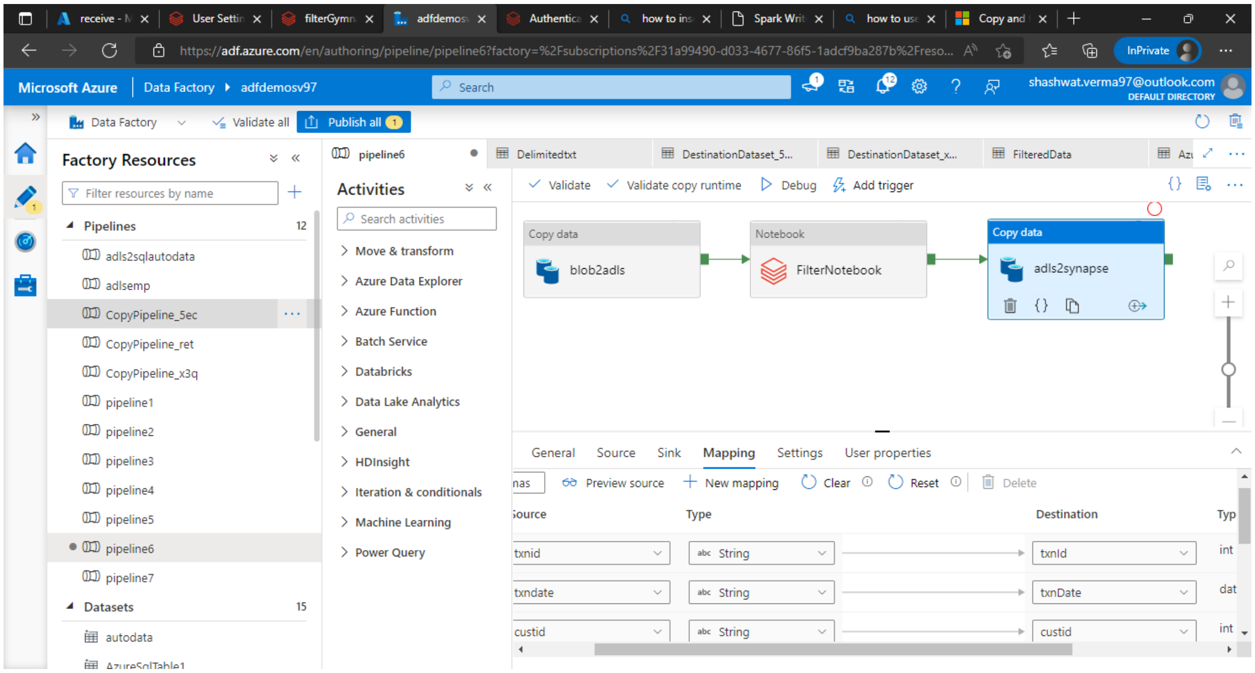
Azure Synapse Analytics
**AzureSynapseAnalyticsTable2**

Connection | Schema | Parameters

Linked service * : AzureSynapseAnalytics1 — Test connection | Edit | New | Learn more
✓ Connection successful

Table: dbo.filteredtxns — Refresh | Preview data
☐ Edit

# Mapping:

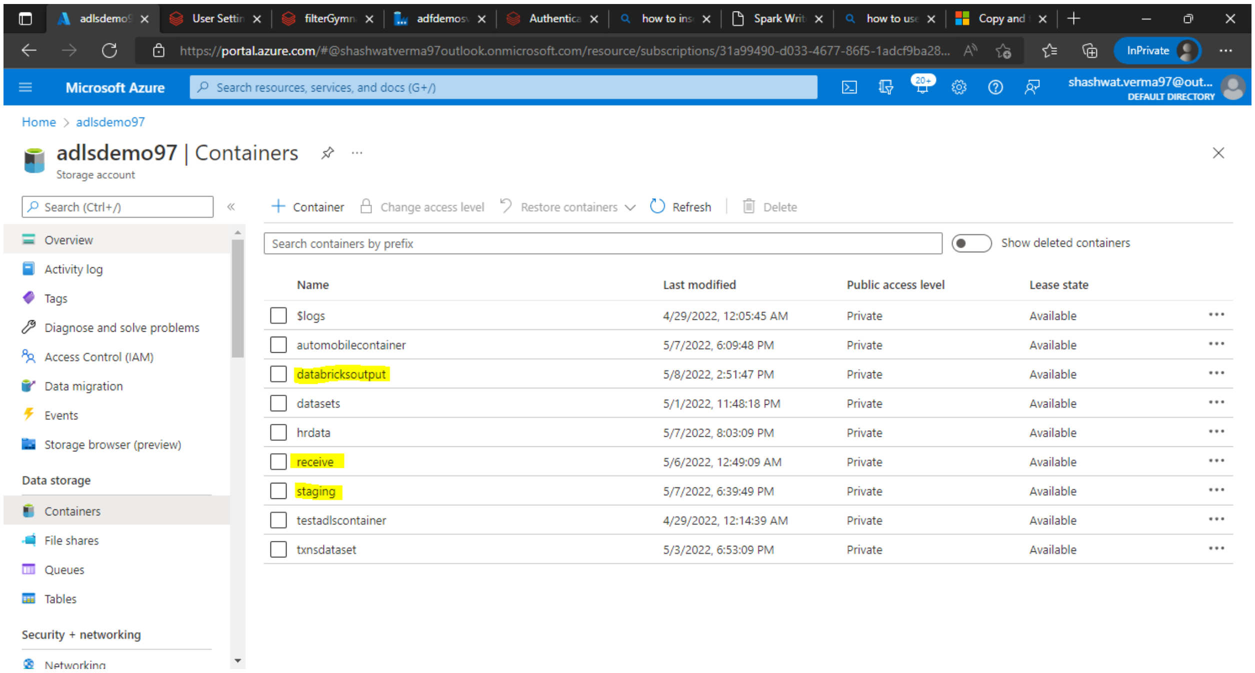


https://adf.azure.com/en/authoring/pipeline/pipeline6?factory=%2Fsubscriptions%2F31a99490-d033-4677-86f5-1adcf9ba287b%2Freso...

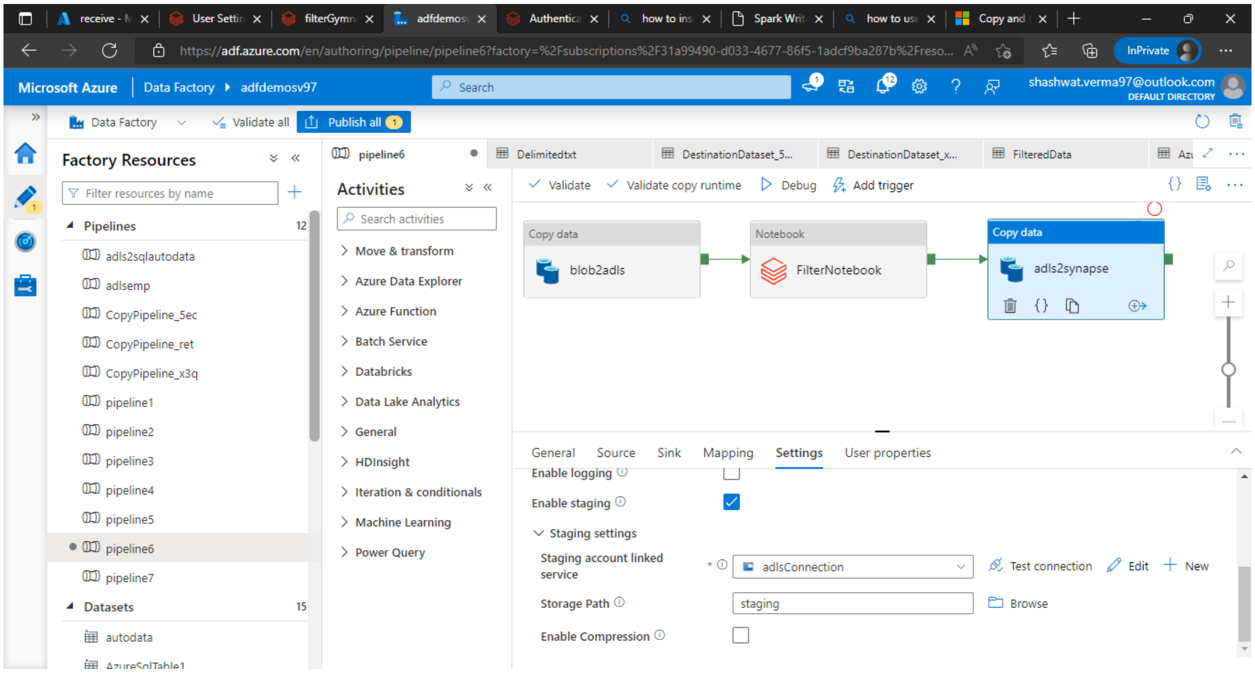
| Pipelines | 12 |
|---|---|
| adls2sqlautodata | |
| adlsemp | |
| CopyPipeline_5ec | |
| CopyPipeline_ret | |
| CopyPipeline_x3q | |
| pipeline1 | |
| pipeline2 | |
| pipeline3 | |
| pipeline4 | |
| pipeline5 | |
| pipeline6 | |
| pipeline7 | |
| **Datasets** | **15** |
| autodata | |
| AzureSqlTable1 | |

| ource | Type | Destination | Typ |
|---|---|---|---|
| txnid | abc String | txnId | int |
| txndate | abc String | txnDate | dat |
| custid | abc String | custid | int |

<u>Settings:</u> In settings we enabled staging at a container named ‘staging’ in ADLS storage account ‘adlsdemo97’.



| Name | Last modified | Public access level | Lease state |
|---|---|---|---|
| $logs | 4/29/2022, 12:05:45 AM | Private | Available |
| automobilecontainer | 5/7/2022, 6:09:48 PM | Private | Available |
| databricksoutput | 5/8/2022, 2:51:47 PM | Private | Available |
| datasets | 5/1/2022, 11:48:18 PM | Private | Available |
| hrdata | 5/7/2022, 8:03:09 PM | Private | Available |
| receive | 5/6/2022, 12:49:09 AM | Private | Available |
| staging | 5/7/2022, 6:39:49 PM | Private | Available |
| testadlscontainer | 4/29/2022, 12:14:39 AM | Private | Available |
| txnsdataset | 5/3/2022, 6:53:09 PM | Private | Available |



Enable logging ☐

Enable staging ☑

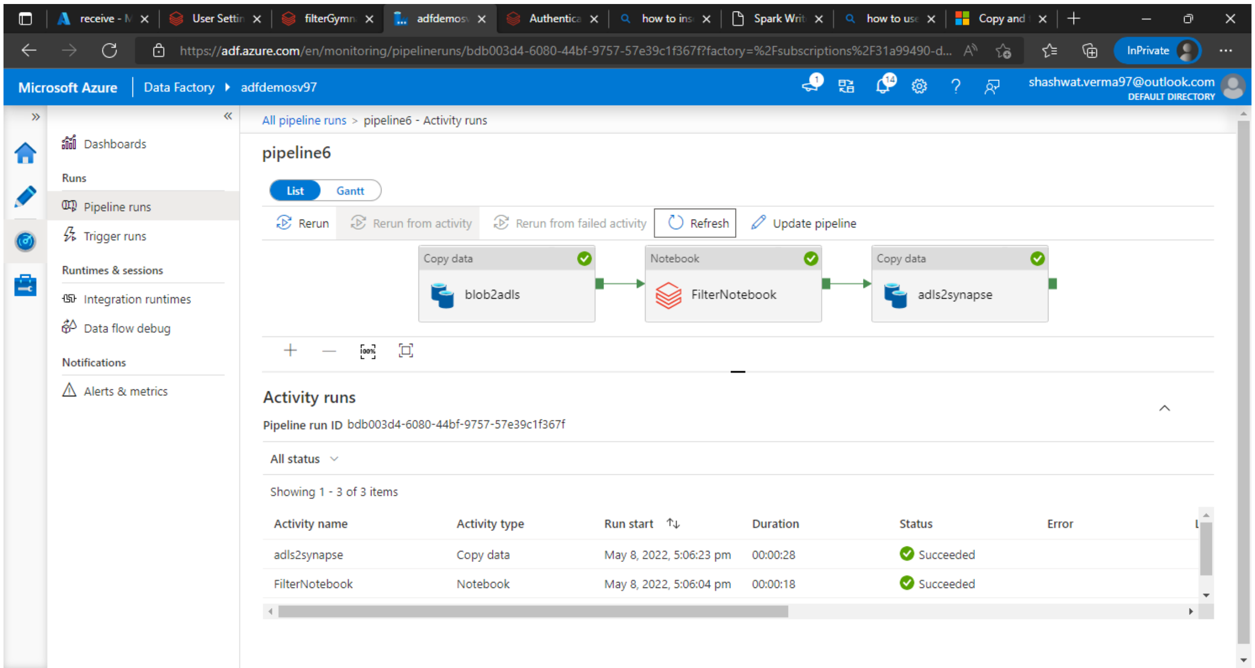
Staging settings

Staging account linked service * adlsConnection — Test connection — Edit — New

Storage Path: staging — Browse

Enable Compression ☐

**7.** Finally after validating the pipeline with no errors we published the pipeline and then triggered it.



pipeline6

Activity runs

Pipeline run ID bdb003d4-6080-44bf-9757-57e39c1f367f

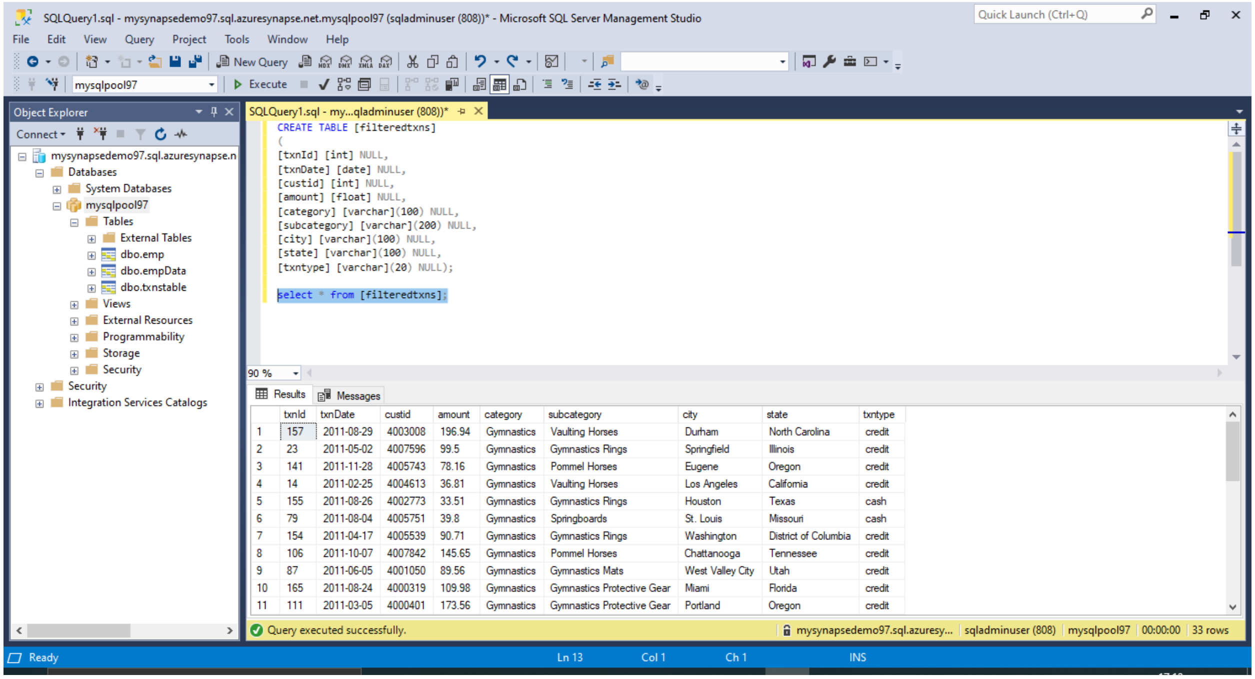
All status

Showing 1 - 3 of 3 items

| Activity name | Activity type | Run start | Duration | Status | Error |
|---|---|---|---|---|---|
| adls2synapse | Copy data | May 8, 2022, 5:06:23 pm | 00:00:28 | ✓ Succeeded | |
| FilterNotebook | Notebook | May 8, 2022, 5:06:04 pm | 00:00:18 | ✓ Succeeded | |

As shown above, it ran successfully.

To confirm we will run query in SSMS as “select * from [filteredtxns];” to fetch all records:



```
CREATE TABLE [filteredtxns]
(
[txnId] [int] NULL,
[txnDate] [date] NULL,
[custid] [int] NULL,
[amount] [float] NULL,
[category] [varchar](100) NULL,
[subcategory] [varchar](200) NULL,
[city] [varchar](100) NULL,
[state] [varchar](100) NULL,
[txntype] [varchar](20) NULL);

select * from [filteredtxns];
```

| | txnId | txnDate | custid | amount | category | subcategory | city | state | txntype |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 157 | 2011-08-29 | 4003008 | 196.94 | Gymnastics | Vaulting Horses | Durham | North Carolina | credit |
| 2 | 23 | 2011-05-02 | 4007596 | 99.5 | Gymnastics | Gymnastics Rings | Springfield | Illinois | credit |
| 3 | 141 | 2011-11-28 | 4005743 | 78.16 | Gymnastics | Pommel Horses | Eugene | Oregon | credit |
| 4 | 14 | 2011-02-25 | 4004613 | 36.81 | Gymnastics | Vaulting Horses | Los Angeles | California | credit |
| 5 | 155 | 2011-08-26 | 4002773 | 33.51 | Gymnastics | Gymnastics Rings | Houston | Texas | cash |
| 6 | 79 | 2011-08-04 | 4005751 | 39.8 | Gymnastics | Springboards | St. Louis | Missouri | cash |
| 7 | 154 | 2011-04-17 | 4005539 | 90.71 | Gymnastics | Gymnastics Rings | Washington | District of Columbia | credit |
| 8 | 106 | 2011-10-07 | 4007842 | 145.65 | Gymnastics | Pommel Horses | Chattanooga | Tennessee | credit |
| 9 | 87 | 2011-06-05 | 4001050 | 89.56 | Gymnastics | Gymnastics Mats | West Valley City | Utah | credit |
| 10 | 165 | 2011-08-24 | 4000319 | 109.98 | Gymnastics | Gymnastics Protective Gear | Miami | Florida | credit |
| 11 | 111 | 2011-03-05 | 4000401 | 173.56 | Gymnastics | Gymnastics Protective Gear | Portland | Oregon | credit |

✓ Query executed successfully. | mysynapsedemo97.sql.azuresy... | sqladminuser (808) | mysqlpool97 | 00:00:00 | 33 rows