

Read Different File Formats from ADLS Gen2 Using SQL Syntax

1. Created Sample Data

- Prepared a CSV file customers.csv with 500 rows of sample customer records.
- Included fields: CustomerID, FirstName, LastName, Email, Phone, City, Country.

	A	B	C	D	E	F	G
1	id	first_name	last_name	email	gender	ip_address	
2	1	Deidre	McLeod	dmcleoid0@github.io	Female	241.79.115.71	
3	2	Jacqui	Gaines	jgaines1@free.fr	Female	195.87.58.156	
4	3	Cassandra	Biggam	cbiggam2@thetimes.co.uk	Female	252.241.154.87	
5	4	Morie	Sartain	msartain3@aboutads.info	Male	170.82.69.153	
6	5	Tommie	Longworthy	tlongworthy4@ehow.com	Male	16.12.144.36	
7	6	Oliviero	Parkhouse	oparkhouse5@yolasite.com	Bigender	65.3.164.12	
8	7	Cornie	Sellar	csellar6@sun.com	Male	30.104.150.51	
9	8	Demetris	Prudence	dprudence7@taobao.com	Female	20.53.25.133	
10	9	Francesca	Hamerton	fhamerton8@epa.gov	Female	2.200.9.122	
11	10	Dave	Richardt	drichardt9@biblegateway.com	Male	108.91.69.218	
12	11	Gayleen	Testin	gtestina@newyorker.com	Female	79.77.236.112	
13	12	Tabbi	Alberti	talbertib@google.com.hk	Female	255.161.132.116	
14	13	Becka	Bambrick	bbambrickc@mysql.com	Female	72.242.69.255	
15	14	Ebeneser	Purcell	epurcellid@adobe.com	Male	43.91.219.143	
16	15	Nollie	Tandy	ntandye@clickbank.net	Female	138.160.54.194	
17	16	Clemente	Overall	coverallf@vkontakte.ru	Male	29.9.161.129	
18	17	Vlad	Holby	vholbyg@uol.com.br	Male	109.116.9.221	
19	18	Audrey	Pales	apalesh@shutterfly.com	Female	162.18.171.235	
20	19	Konstance	Longhi	klonghihi@friendfeed.com	Female	251.78.234.118	
21	20	Leyla	Curry	lcurryj@dropbox.com	Female	207.77.180.107	
22	21	Vincenty	Coppeard	vcoppeardk@prnewswire.com	Male	127.144.182.48	
23	22	Graehme	Twynning	gtwynningl@adobe.com	Male	250.123.15.166	
24	23	Mathias	Tondeur	mtondeurm@cnbc.com	Male	30.194.99.49	
25	24	Amelie	Antoons	aantoonsn@elpais.com	Female	57.188.242.197	
26	25	Hirsch	Tyers	htyerso@bloomberg.com	Male	202.135.214.221	

2. Created Azure Resources

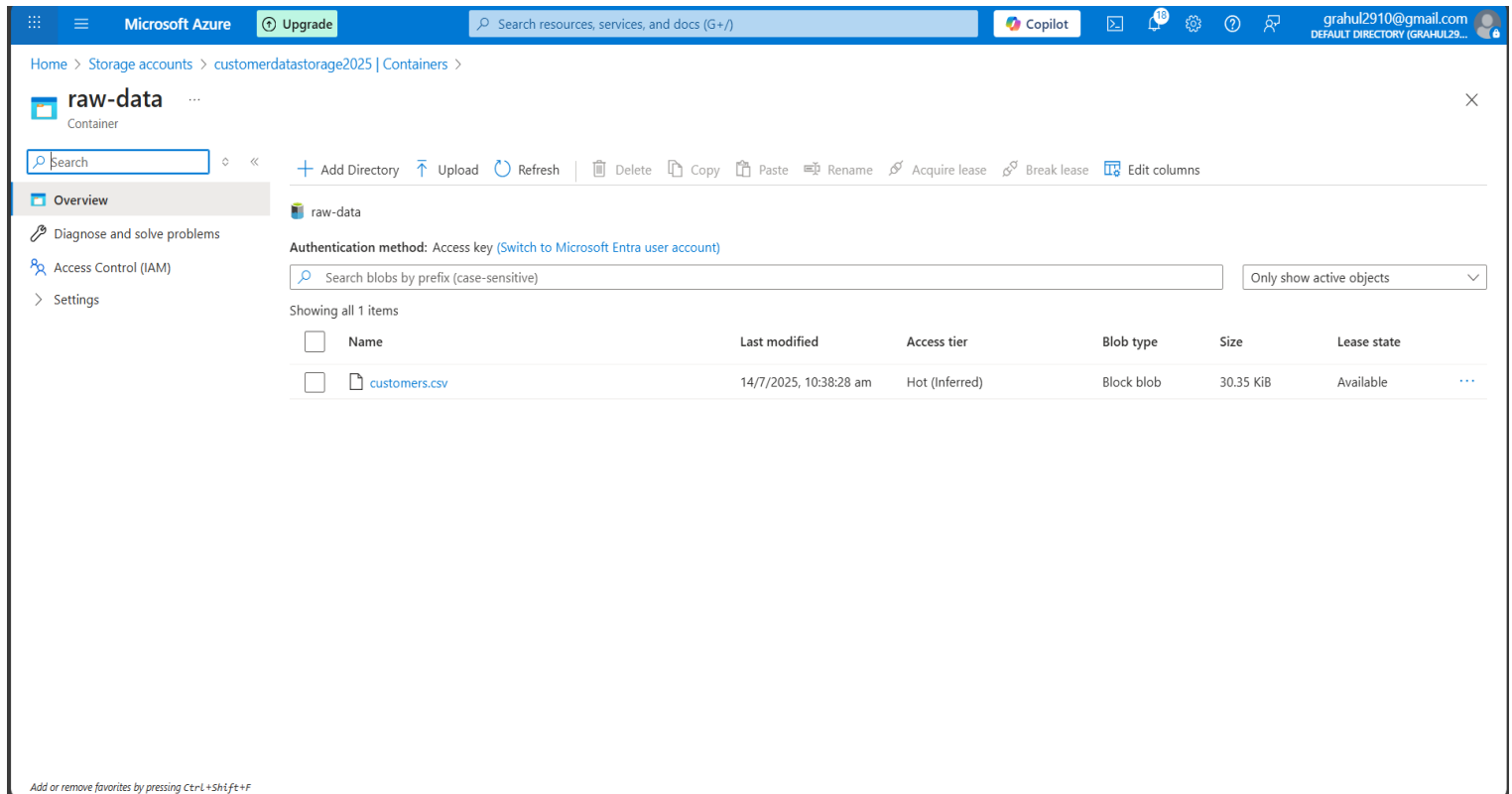
- Resource Group: customerdata-rg-final

The screenshot displays the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, an 'Upgrade' button, a search bar, and a 'Copilot' button. The user's profile 'grahul2910@gmail.com' is visible in the top right corner.

The main content area is titled 'Resource groups' and shows a list of resource groups on the left sidebar. The selected resource group is 'customerdata-rg-final'. The main panel displays the 'Resources' tab for this group, showing a table of resources:

Name	Type	Location
customer-databricks-final	Azure Databricks Service	Central India
customerdata-adf-final	Data factory (V2)	Central India
customerdatastorage2025	Storage account	Central India

- Storage Account: Enabled ADLS Gen2 by checking Hierarchical Namespace
- Containers Created:
 - raw-data: for original customers.csv
 - converted: for output in multiple formats

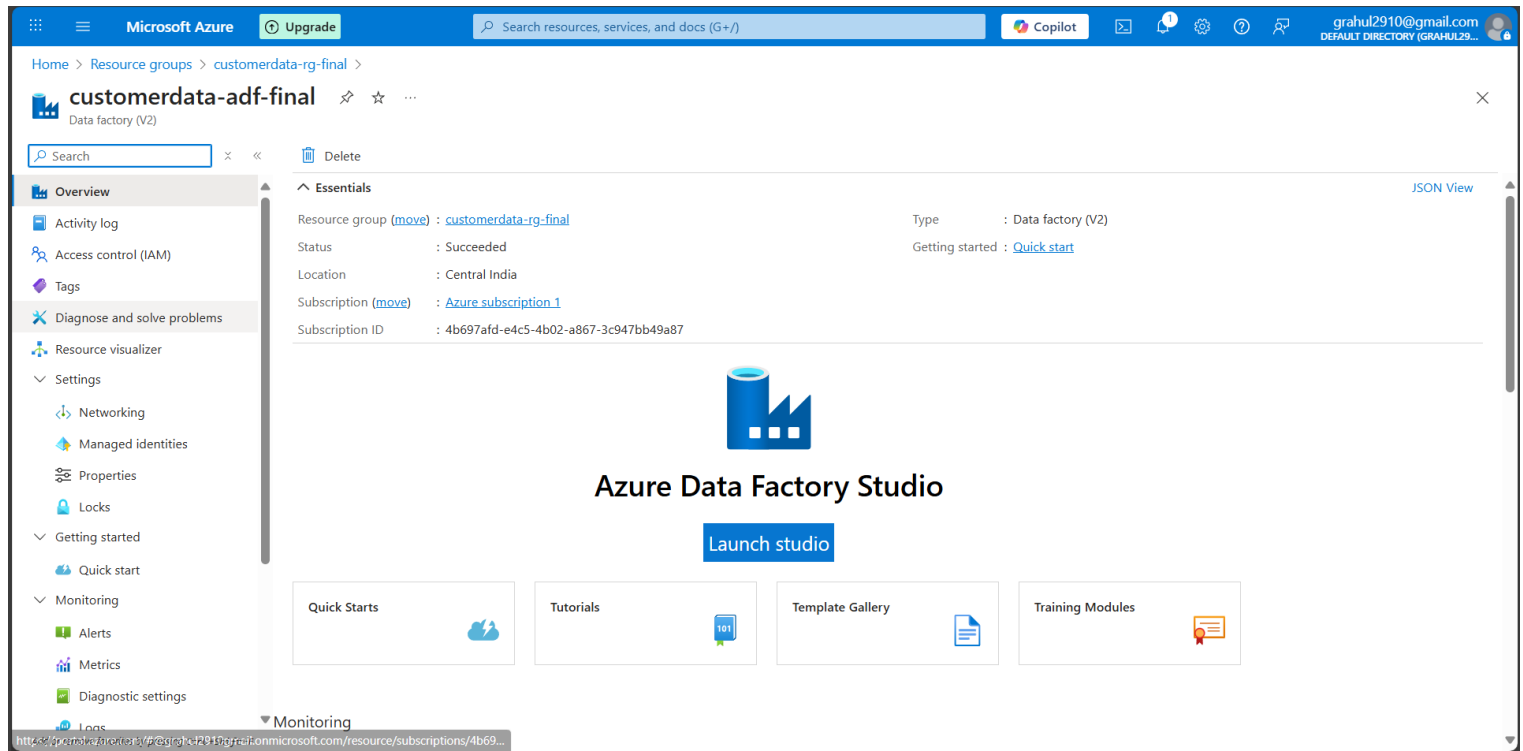


3. Uploaded CSV File

- Uploaded customers.csv to the raw-data container using the Azure portal.

4. Created Azure Data Factory

- Created a data factory named customerdata-adf-final.
- Launched ADF Studio for pipeline development.



5. Created Linked Services

- Created a linked service (LS_ADLSGen2) to connect ADF to the ADLS Gen2 storage.

Edit linked service

Azure Data Lake Storage Gen2 [Learn more](#)

Name *

LS_ADLSGen2

Description

Connect via integration runtime *

✓ AutoResolveIntegrationRuntime

Authentication type

Account key

Account selection method

☐ From Azure subscription ☒ Enter manually

URL *

https://customerdatastorage2025.dfs.core.windows.net/

Storage account key **Azure Key Vault**

Storage account key *

.....

Test connection

☒ To linked service ☐ To file path

Annotations

Save Cancel Test connection

6. Built ADF Pipeline for File Format Conversion

- Pipeline Name: CSV_Conversion_Pipeline
- Created one Copy Data Activity each to convert customers.csv into:
 - Parquet → stored in converted/parquet/
 - Avro → stored in converted/avro/
 - ORC → stored in converted/orc/

Delta was not supported directly in ADF, so it was generated later via Databricks.

The screenshot shows the Microsoft Azure Data Factory Author interface for a pipeline named 'CSV_Conversion_Pipeline'. The left sidebar contains navigation options: Home, Author, Monitor, Manage, and Learning Center. The main area displays the pipeline's configuration, including a 'Factory Resources' pane on the left and an 'Activities' pane in the center. The 'Activities' pane lists various activities like Move and transform, Synapse, Azure Data Explorer, Azure Function, Batch Service, Databricks, Data Lake Analytics, General, HDInsight, Iteration & conditionals, Machine Learning, and Power Query. The right pane shows the pipeline's visual representation with four 'Copy data' activities: 'Copy_data_avro', 'Copy_data_ORC', 'Copy_data_parquet', and 'Copy_data_JSON'. The 'Source' tab is selected, showing the 'Source dataset' as 'Source_DataSet' and the 'File path type' as 'File path in dataset'. The 'Filter by last modified' section is also visible, with 'Start time (UTC)' and 'End time (UTC)' input fields.

Run the pipeline to convert CSV to various file formats.

The screenshot shows the Microsoft Azure Data Factory Monitor interface for the 'CSV_Conversion_Pipeline'. The left sidebar contains navigation options: Home, Author, Monitor, Manage, and Learning Center. The main area displays the 'All pipeline runs' section, showing a list of pipeline runs with a status of 'Succeeded'. Below this, the 'Activity runs' section shows a table of activity runs for the pipeline run ID 'a16ec890-c221-40df-be67-5b1fa8a59942'. The table lists four activities: 'Copy_data_avro', 'Copy_data_parquet', 'Copy_data_ORC', and 'Copy_data_JSON', all of which have a status of 'Succeeded'. The table also includes columns for 'Activity name', 'Activity status', 'Activity name', 'Run start', 'Duration', 'Integration runtime', and 'User'.

Activity name	Activity status	Activity name	Run start	Duration	Integration runtime	User
Copy_data_avro	Succeeded	Copy data	7/14/2025, 11:42:07 AM	14s	AutoResolveIntegrationRuntime (Central India)	
Copy_data_parquet	Succeeded	Copy data	7/14/2025, 11:42:07 AM	14s	AutoResolveIntegrationRuntime (Central India)	
Copy_data_ORC	Succeeded	Copy data	7/14/2025, 11:42:07 AM	14s	AutoResolveIntegrationRuntime (Central India)	
Copy_data_JSON	Succeeded	Copy data	7/14/2025, 11:42:07 AM	13s	AutoResolveIntegrationRuntime (Central India)	

7. Created Databricks Workspace & Cluster

- Workspace: customer-databricks

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, an 'Upgrade' button, a search bar, and a user profile for 'grahul2910@gmail.com'. The main content area is titled 'customerdata-rg-final_customer-databricks-final | Overview'. On the left, there is a sidebar with 'Overview', 'Inputs', 'Outputs', and 'Template'. The main area displays 'Deployment is in progress' with details: Deployment name: customerdata-rg-final_customer-databricks-final, Subscription: Azure subscription 1, Resource group: customerdata-rg-final, Start time: 7/14/2025, 11:46:27 AM, and Correlation ID: 2908bc97-5ec2-4f20-a338-50852cef1705. Below this, a table shows the deployment details for the 'customer-databricks-final' resource, which is an 'Azure Databricks Service' with a status of 'Created'. On the right, there are links for 'Microsoft Defender for Cloud', 'Free Microsoft tutorials', and 'Work with an expert'.

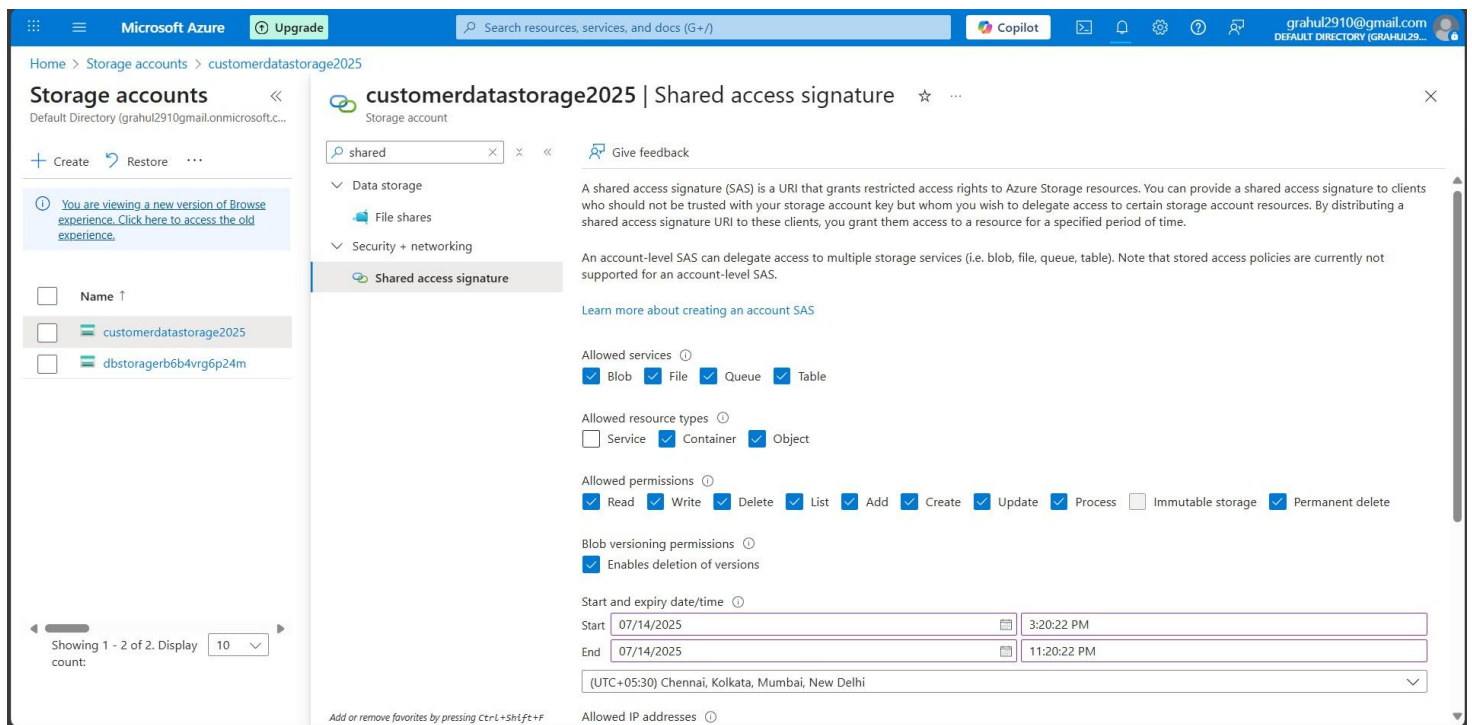
Resource	Type	Status	Operation details
customer-databricks-final	Azure Databricks Service	Created	Operation details

The screenshot shows the Databricks workspace configuration page for a new cluster named 'customer-cluster'. The left sidebar contains navigation options like 'New', 'Workspace', 'Recents', 'Catalog', 'Jobs & Pipelines', 'Compute', 'Data Engineering', 'Job Runs', 'AI/ML', 'Playground', 'Experiments', 'Features', 'Models', and 'Serving'. The main area is titled 'Compute > New compute > Simple form: OFF'. It shows the cluster configuration for 'customer-cluster'. The 'Access mode' is set to 'Single user' and 'Single user or group access' is set to 'Rahul Gupta'. The 'Performance' section shows 'Databricks runtime version' as 'Runtime: 16.4 LTS (Scala 2.12) (Scala 2.12, Spark 3.5.2)' and 'Use Photon Acceleration' is checked. The 'Worker type' is 'Standard_D4ds_v5' with '16 GB Memory, 4 Cores' and '1' worker. 'Spot instances' are checked. The 'Driver type' is 'Same as worker' with '16 GB Memory, 4 Cores'. 'Enable autoscaling' is unchecked, and 'Terminate after' is set to '120 minutes of inactivity'. The 'Tags' section is empty. A 'Summary' panel on the right shows the configuration: 1 Worker (16 GB Memory, 4 Cores), 1 Driver (16 GB Memory, 4 Cores), Runtime (16.4.x-scala2.12), and Photon Standard_D4ds_v5 4 DBU/h. At the bottom, there are 'Create compute' and 'Cancel' buttons.

Summary

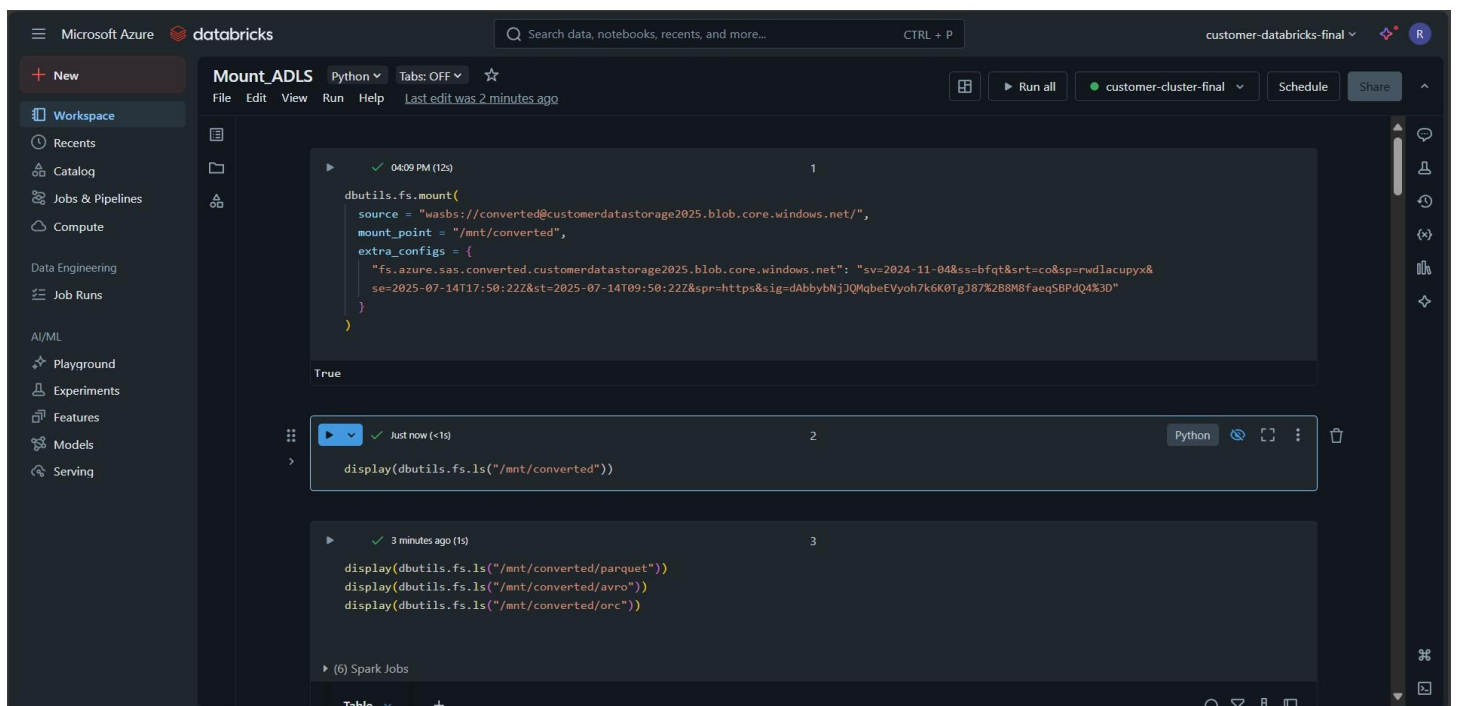
Component	Configuration
1 Worker	16 GB Memory, 4 Cores
1 Driver	16 GB Memory, 4 Cores
Runtime	16.4.x-scala2.12
Photon	Standard_D4ds_v5 4 DBU/h

Configure Shared Access Signature, put the generated key in databricks.



8. Mounted ADLS Containers in Databricks

- Mounted both raw-data and converted containers using `dbutils.fs.mount()`



- Verified mount with `dbutils.fs.ls("/mnt/raw-data")` and `dbutils.fs.ls("/mnt/converted")`

The screenshot shows a Databricks workspace with a notebook titled "Mount_ADLS". The notebook contains two SQL queries. The first query, executed 3 minutes ago, is:

```
%sql
CREATE OR REPLACE TABLE delta_parquet
USING DELTA
AS SELECT * FROM parquet.`/mnt/converted/parquet/`;
```

The second query, executed 2 minutes ago, is:

```
%sql
CREATE OR REPLACE TABLE delta_avro
USING DELTA
AS SELECT * FROM avro.`/mnt/converted/avro/`;
```

Both queries show "0 rows" returned. The interface indicates "No rows returned" and "0 rows | 13.17s runtime". A message states: "This result is stored as `_sqlidf` and can be used in other Python and SQL cells."

9. Created Delta File Using Databricks (for Delta format)

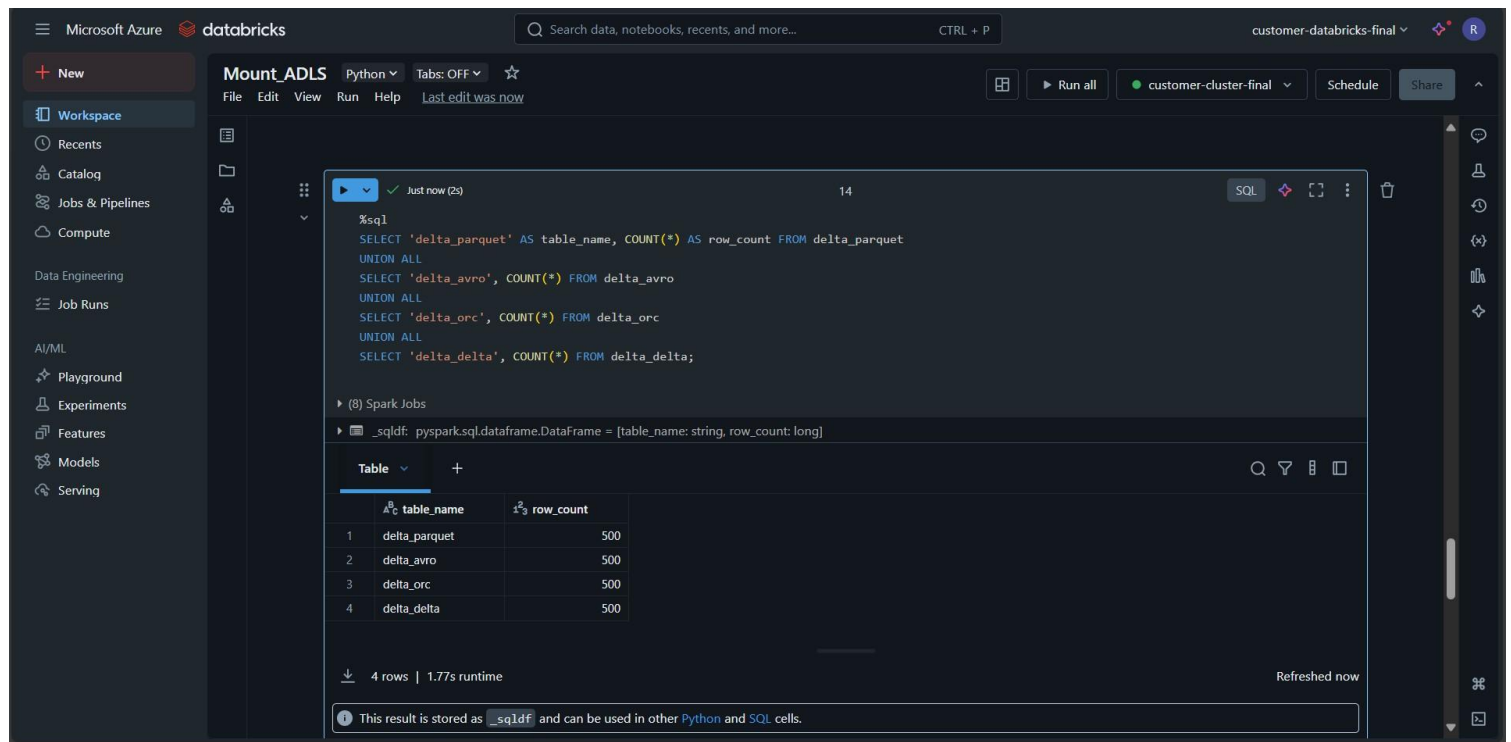
The screenshot shows a Databricks workspace with a notebook titled "Mount_ADLS". The notebook contains a SQL query executed at 04:42 PM (2s):

```
%sql
SELECT * FROM delta_parquet
```

The query result is displayed as a table with 15 rows and 7 columns. The columns are: `id`, `first_name`, `last_name`, `email`, `gender`, and `ip_address`.

	id	first_name	last_name	email	gender	ip_address
1	1	Deidre	McLeoid	dmcleoid0@github.io	Female	241.79.115.71
2	2	Jacqui	Gaines	jjgaines1@free.fr	Female	195.87.58.156
3	3	Cassandra	Biggam	cbiggam2@thetimes.co.uk	Female	252.241.154.87
4	4	Morie	Sartain	msartain3@aboutads.info	Male	170.82.69.153
5	5	Tommie	Longworthy	tlongworthy4@ehow.com	Male	16.12.144.36
6	6	Oliviero	Parkhouse	oparkhouse5@yolasite.com	Bigender	65.3.164.12
7	7	Cornie	Sellar	csellar6@sun.com	Male	30.104.150.51
8	8	Demetris	Prudence	dprudence7@taobao.com	Female	20.53.25.133
9	9	Francesca	Hamerton	thamerton8@epa.gov	Female	2.200.9.122
10	10	Dave	Richardt	drichardt9@biblegateway.com	Male	108.91.69.218
11	11	Gayleen	Testin	gtestina@newyorker.com	Female	79.77.236.112
12	12	Tabbi	Alberti	talbertib@google.com.hk	Female	255.161.132.116
13	13	Becka	Bambrick	bbambrick@mysql.com	Female	72.242.69.255
14	14	Ebeneser	Purcell	epurcell@adobe.com	Male	43.91.219.143
15	15	Nollie	Tandy	ntandy@clickbank.net	Female	138.160.54.194

10. Verify and Validate that 500 rows are created



The screenshot shows the Databricks interface with a workspace named 'Mount_ADLS'. A SQL query is executed, and the results are displayed in a table. The query counts the number of rows in four Delta tables: delta_parquet, delta_avro, delta_orc, and delta_delta. The results show that each table contains 500 rows.

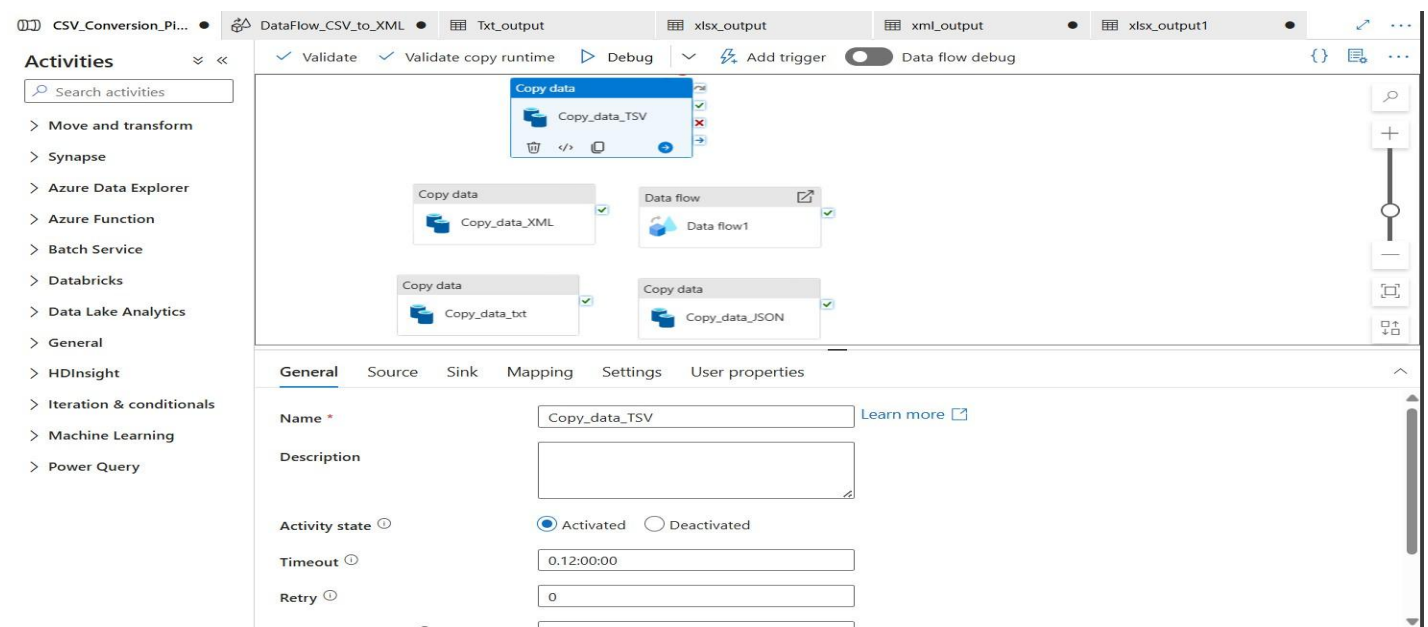
```
%sql
SELECT 'delta_parquet' AS table_name, COUNT(*) AS row_count FROM delta_parquet
UNION ALL
SELECT 'delta_avro', COUNT(*) FROM delta_avro
UNION ALL
SELECT 'delta_orc', COUNT(*) FROM delta_orc
UNION ALL
SELECT 'delta_delta', COUNT(*) FROM delta_delta;
```

table_name	row_count
delta_parquet	500
delta_avro	500
delta_orc	500
delta_delta	500

For CSV, TSV, JSON, XML,XLSX, TXT(using temporary view then converting it into Delta table):

1.) Changes to pipeline

- Make changes to the pipeline so can it can convert sample data to various file formats



The screenshot shows the Azure Data Factory pipeline editor. The pipeline is named 'CSV_Conversion_Pipeline'. It contains several activities: 'Copy data' (Copy_data_TSV), 'Copy data' (Copy_data_XML), 'Copy data' (Copy_data_txt), 'Data flow' (Data flow1), and 'Copy data' (Copy_data_JSON). The 'Copy_data_TSV' activity is selected, and its properties are shown in the 'General' tab. The activity is activated and has a timeout of 0.12:00:00.

Activities

- Move and transform
- Synapse
- Azure Data Explorer
- Azure Function
- Batch Service
- Databricks
- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Power Query

Copy_data_TSV

General

Name: Copy_data_TSV

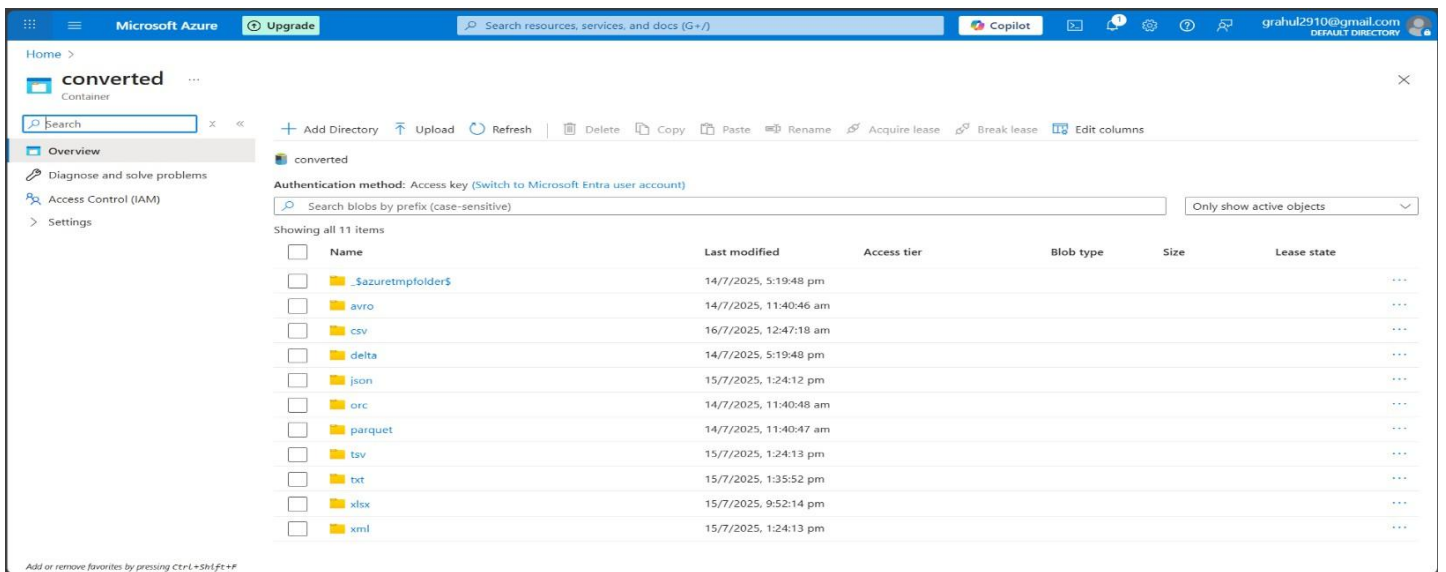
Description:

Activity state: ☒ Activated ☐ Deactivated

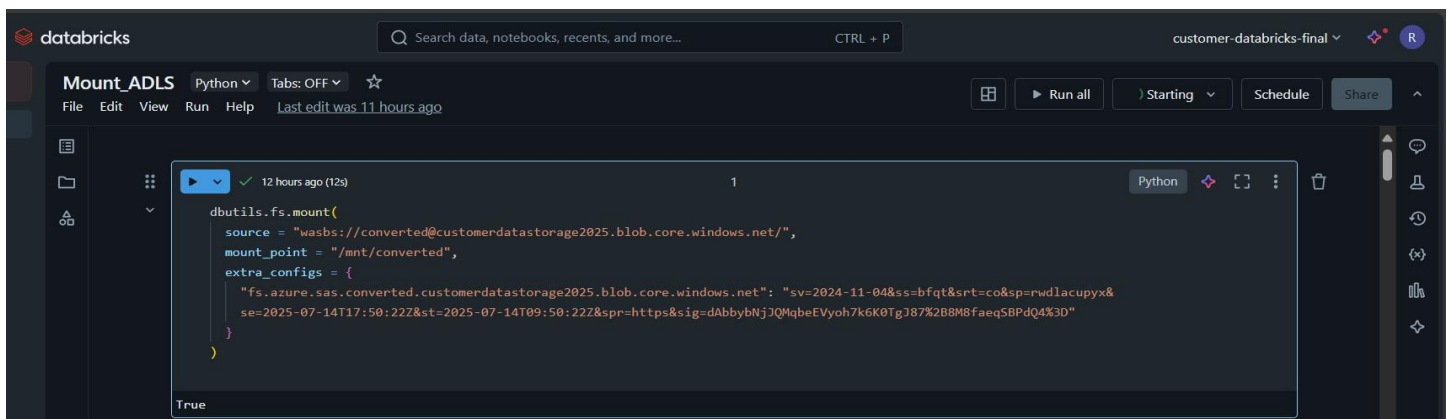
Timeout: 0.12:00:00

Retry: 0

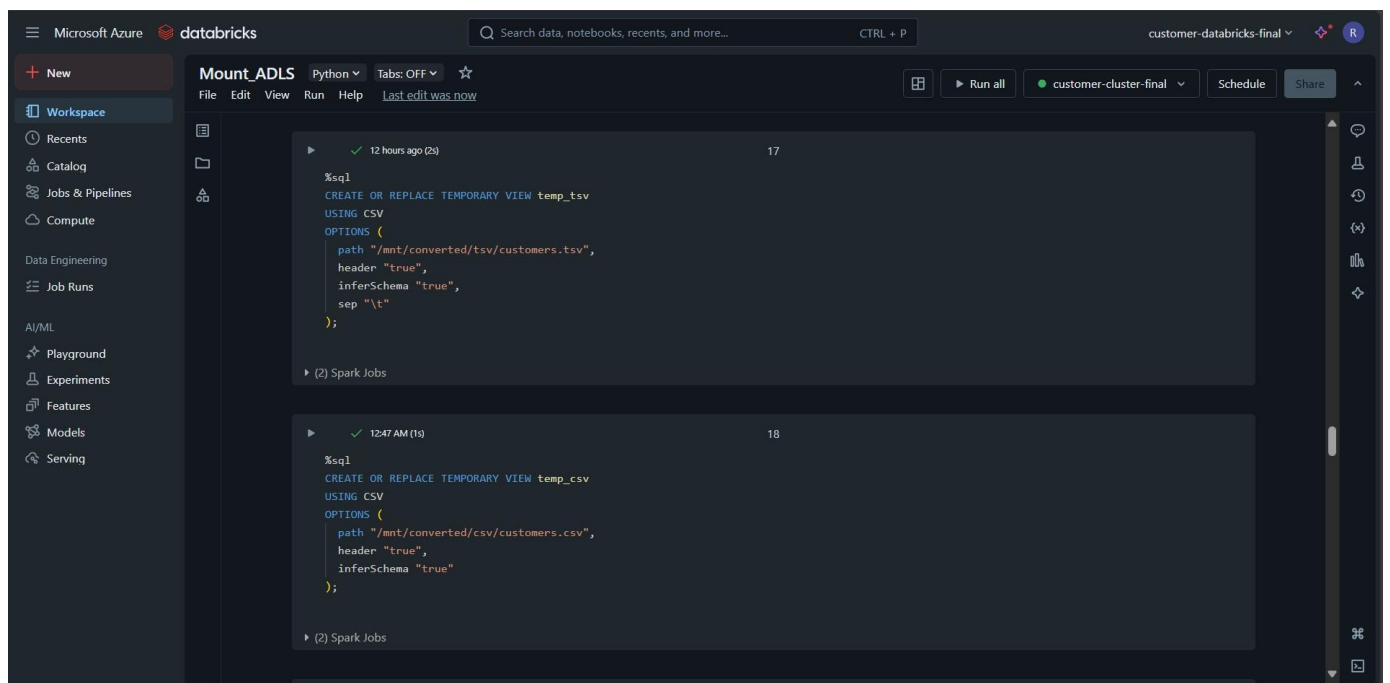
2.) Check the updated output



3.) Mount Data on Databricks



4.) Create Temporary View for all the file formats



5.) Create Table As Select using the temporary View

The screenshot shows the Databricks interface with a notebook titled "Mount_ADLS". The notebook is in Python mode. The first cell (12:21 AM) contains the following SQL code:

```
%sql
CREATE OR REPLACE TABLE delta_xlsx AS SELECT * FROM temp_xlsx;
```

The second cell (12:22 AM) contains the following SQL code:

```
%sql
CREATE OR REPLACE TABLE delta_txt AS SELECT * FROM temp_txt;
```

The notebook shows the execution results for the first cell, indicating that 0 rows were returned. The second cell is also shown, but its results are not visible in the screenshot.

6.) Validate 500 rows

The screenshot shows the Databricks interface with a notebook titled "Mount_ADLS". The notebook is in Python mode. The first cell (01:01 AM) contains the following SQL code:

```
%sql
SELECT 'delta_csv' AS table_name, COUNT(*) AS row_count FROM delta_csv
UNION ALL
SELECT 'delta_tsv', COUNT(*) FROM delta_tsv
UNION ALL
SELECT 'delta_json', COUNT(*) FROM delta_json
UNION ALL
SELECT 'delta_xlsx', COUNT(*) FROM delta_xlsx
UNION ALL
SELECT 'delta_txt', COUNT(*) FROM delta_txt;
```

The notebook shows the execution results for the first cell, indicating that 5 rows were returned. The results are displayed in a table with 2 columns: table_name and row_count.

table_name	row_count
delta_csv	500
delta_tsv	500
delta_json	500
delta_xlsx	500
delta_txt	500