

# Project 2

## Domain-Specific Data Engineering

### Retail Inventory Management and Sales Forecasting

#### **Step 1: Define the Scope**

1. **Business Objective:** Optimize inventory management and sales forecasting.
2. **Data Sources:**
  - Sales transactions (customer purchases, product details, quantities, timestamps)
  - Inventory logs (stock levels, product movements, dates)
  - Customer data (demographics, purchase history, preferences)
3. **Success Metrics/Goal:** Optimize inventory levels to reduce stockouts by 20% within the next 12 months, while improving sales forecast accuracy by 15%.

#### **Step 2: Design the Data Pipeline:**

1. **Data Ingestion:**
  - **Sales transactions:** Through data transfer via linked service to onprem network for batch processing (on-premises to Azure SQL data base)
  - **Inventory logs:** Copy activity, source: Synapse Warehouse, sink: Azure SQL Database
  - **Customer data:** Through Database connectors, Notebook Activity, Source: Azure database for MySQL (Azure data studio), sink: Azure SQL database
2. **Data Storage:** Azure SQL Database
3. **Data Transformation:** Pyspark (notebook)
4. **Data Warehousing and Analytics:** Snowflake, Power BI

#### **Step 3. Implementation and Monitoring: Adf monitor**

#### **Step 4: Advanced Considerations:**

1. **Data Format Exploration:** (CSV vs. Parquet) comparisons in data bricks
2. **Data Pipeline Automation:** No real time data implemented in this project
3. **Medallion Architecture:** Implemented in data bricks
4. **Distribution Indexing:** Used indexing and table distribution on the source tables

#### **Implementation of project:**

1. **Data Ingestion:**
  - **Sales transactions:** Through copy activity transfer via linked service to on-premises (batch processing) to Azure SQL data base)

SQLQuery2.sql - RAKSHITHA\RAKSHITHAMAIN.salesdatabase (RAKSHITHA\raksh (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute

Quick Launch (Ctrl+Q)

Object Explorer

RAKSHITHA\RAKSHITHAMAIN (SQL Server 15.0.1000.34 - Rakshitha\raksh)

salesdatabase

SQLQuery2.sql - RA\\_SITHA\ra(sh (59))

```
SELECT COUNT(*) FROM [SourceSalesTransactions];
SELECT * FROM [SourceSalesTransactions];
```

Results Messages

TransactionID	CustomerID	ProductID	Quantity	Timestamp
1	139	P001	9	2024-01-06 07:57:00.000
2	2	P004	5	2024-01-06 10:45:00.000
3	3	P006	2	2024-01-06 10:45:00.000
4	4	P009	2	2024-01-06 10:03:00.000
5	5	P001	8	2024-01-06 07:13:00.000
6	6	P021	8	2024-01-06 13:32:00.000
7	7	P003	3	2024-01-06 14:59:00.000
8	8	P004	1	2024-01-06 02:39:00.000
9	9	P010	1	2024-01-06 00:34:00.000
10	10	P002	2	2024-01-06 01:19:00.000
11	11	P003	6	2024-01-06 01:19:00.000
12	12	P005	7	2024-01-06 02:55:00.000
13	13	P003	3	2024-01-06 02:58:00.000
14	14	P010	9	2024-01-06 13:01:00.000
15	15	P003	4	2024-01-06 13:51:00.000
16	16	P004	1	2024-01-06 10:59:00.000
17	17	P006	2	2024-01-06 08:12:00.000
18	18	P008	8	2024-01-06 08:12:00.000
19	19	P010	2	2024-01-06 08:45:00.000
20	20	P007	1	2024-01-06 02:53:00.000
21	21	P006	2	2024-01-06 09:41:00.000
22	22	P001	9	2024-01-06 14:45:00.000
23	23	P007	8	2024-01-06 09:34:00.000
24	24	P006	10	2024-01-06 20:59:00.000
25	25	P007	9	2024-01-06 20:34:00.000
26	26	P008	9	2024-01-06 20:27:00.000
27	27	P006	10	2024-01-06 21:41:00.000
28	28	P001	10	2024-01-06 02:46:00.000

Query executed successfully.

RAKSHITHA\RAKSHITHAMAIN (15... RAKSHITHA\raksh (59) salesdatabase 00:00:00 700 rows

Ready

InPrivate

Customer x mywkrak x mydataf x SqlData x Tutorial x All resou x mydataf x Copilot x ChatGPT x +

https://adf.azure.com/en/authoring/pipeline/pipeline?factory=%2Fsubscriptions%2Fbf9ac98f-4827-452b-bd2c-930ad6cd590b%2FresourceGroups%2FETL...

keerhanaakkula@gmail.com

Microsoft Azure | Data Factory | mydatafactory | Search factory and documentation

Data Factory | Validate all | Publish all | Auto Save | Preview experience On

pipeline1 | AzureSqlTable

Activities | copy | Validate | Validate copy runtime | Debug | Add trigger

Copy data

onpremToSQLDB

General Source Sink Mapping Settings User properties

Source dataset \* SqlServerTable1

Open New Preview data Learn more

Use query Table Query Stored procedure

Query timeout (minutes) 120

Isolation level Read uncommitted

Partition option None Physical partitions of table Dynamic range

Please preview data to validate the partition settings.

Additional columns New

Microsoft Azure | Data Factory > mydatafactory Search factory and documentation

Validate all Publish all 4 Auto Save Preview experience On

Activities <> Validate Validate copy runtime Debug Add trigger

copy

< Move and transform

Copy data

Copy data

onpremToAzSQLdb

General Source Sink Mapping Settings User properties

Sink dataset \* AzureSqlTable Open New Learn more

Write behavior Insert Upsert Stored procedure

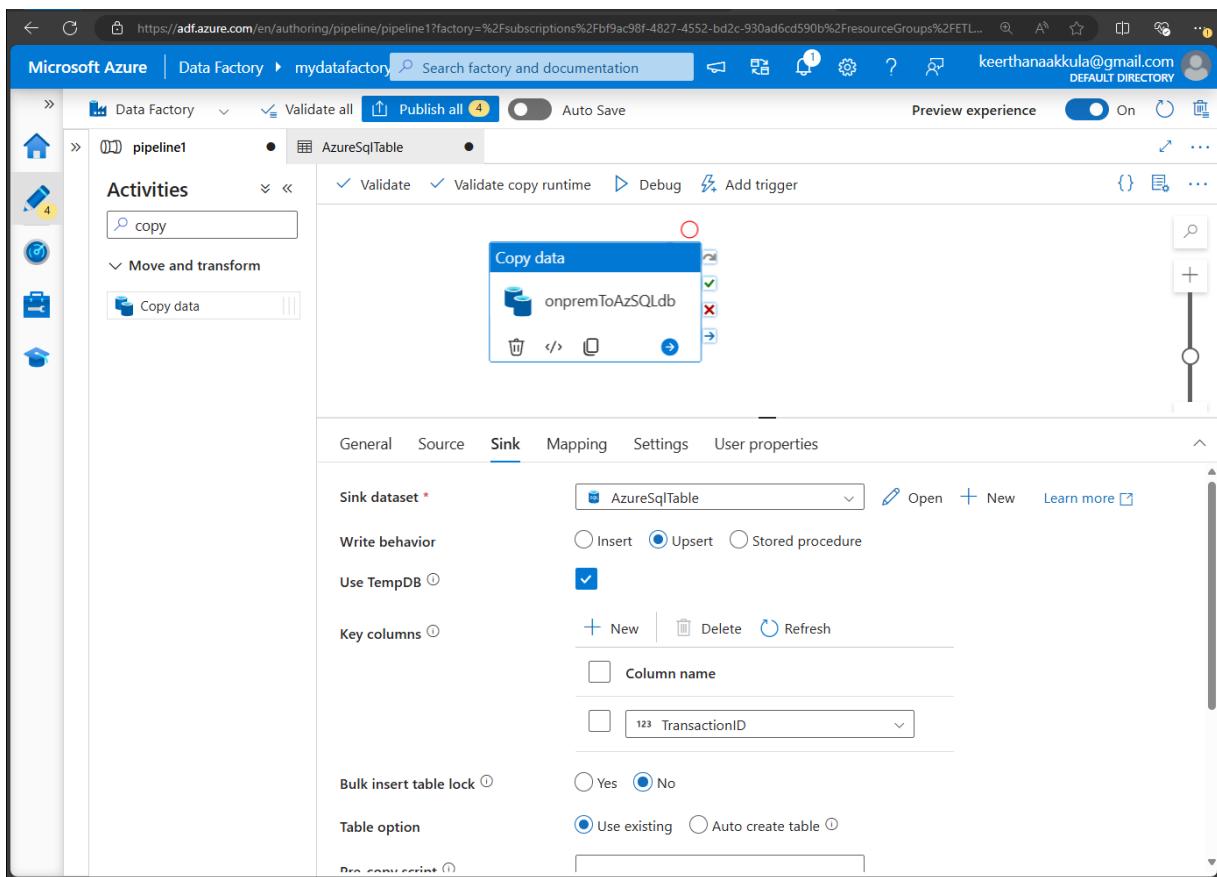
Use TempDB

Key columns  Column name  TransactionID

Bulk insert table lock  Yes  No

Table option Use existing Auto create table

Pre copy script



Microsoft Azure | Data Factory > mydatafactory Search factory and documentation

Validate all Publish all 4 Auto Save Preview experience On

<> Validate Debug Add trigger

Copy data

onpremToAzSQLdb

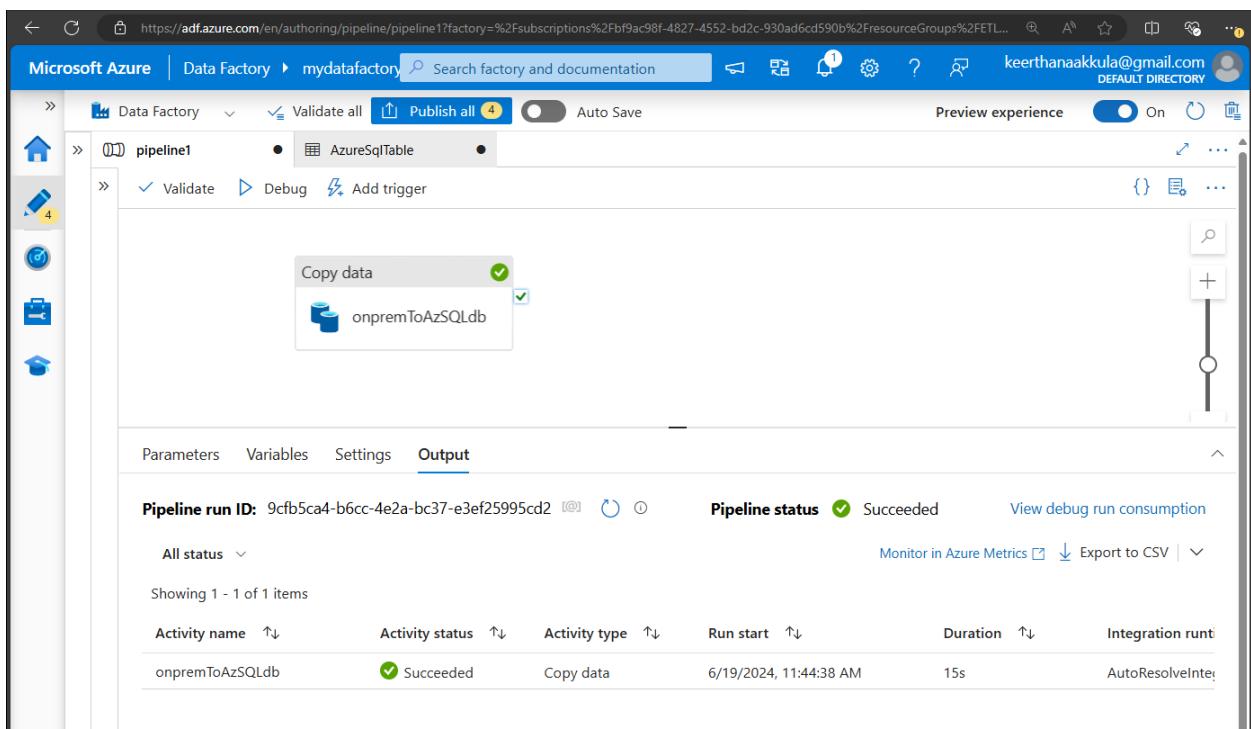
Parameters Variables Settings Output

Pipeline run ID: 9cfb5ca4-b6cc-4e2a-bc37-e3ef25995cd2 Pipeline status Succeeded View debug run consumption

All status ▾ Monitor in Azure Metrics Export to CSV

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runt
onpremToAzSQLdb	Succeeded	Copy data	6/19/2024, 11:44:38 AM	15s	AutoResolveInteg



The screenshot shows the Microsoft Azure SQL Database Query Editor interface. The left sidebar displays the database structure for 'SqlDatabase (rakshitha)'. The 'Tables' section lists 'dbo.SourceCustomerData', 'dbo.SourceInventoryLogs', and 'dbo.SourceSalesTransactions'. The 'dbo.SourceSalesTransactions' table is currently selected. The main area contains a query editor window titled 'Query 4' with the following SQL code:

```
1  SELECT TOP (1000) * FROM [dbo].[SourceSalesTransactions]
```

Below the query editor is a results grid showing the following data:

TransactionID	CustomerID	ProductID	Quantity	Timestamp
1	186	P010	9	2024-06-01T23:23:00.0000
2	117	P004	6	2024-06-01T16:16:00.0000
3	137	P001	5	2024-06-01T00:16:00.0000
4	198	P007	6	2024-06-01T13:09:00.0000
5	199	P009	10	2024-06-01T07:04:00.0000
6	144	P008	6	2024-06-01T22:21:00.0000
7	146	P004	2	2024-06-01T20:58:00.0000

A green status bar at the bottom indicates 'Query succeeded | 0s'.

The screenshot shows the Microsoft Azure SQL Database Query Editor interface, identical to the one above but with a different query. The left sidebar displays the database structure for 'SqlDatabase (rakshitha)'. The 'Tables' section lists 'dbo.SourceCustomerData', 'dbo.SourceInventoryLogs', and 'dbo.SourceSalesTransactions'. The 'dbo.SourceSalesTransactions' table is currently selected. The main area contains a query editor window titled 'Query 4' with the following SQL code:

```
1  SELECT count(*) FROM [dbo].[SourceSalesTransactions]
```

The results grid shows a single row with the value '700'.

```

SELECT COUNT(*) FROM [SourceSalesTransactions];
SELECT * FROM [SourceSalesTransactions];

```

Results (No column name)

1	700
---	-----

Query executed successfully.

- **Inventory logs: Copy activity, source: Synapse Warehouse, sink: Azure SQL Database**

Name	Type	Related	Annotations
AzureDatabricks1	Azure Databricks	1	
AzureDatabricksDeltaLakeLin...	Azure Databricks Delta Lake	1	
AzureSqlDatabase	Azure SQL Database	1	
AzureSynapseAnalytics	Azure Synapse Analytics	1	
linkedService1	Azure Blob Storage	1	
SnowflakeLinkedservice	Snowflake	1	
SqlServer	SQL server	0	

InPrivate Catalog Exp. LakeHouse\_ mydatafactory mydatafacto SqlDatabase Tutorial - Cr Secret scope

Data Factory > mydatafactoryspace Search factory and documentation

Validate all Publish all Auto Save Preview experience On

MainPipeline AzureSynapseAnalyti... pipeline2\_datasour...

Validate Validate copy runtime Debug Add trigger

Copy data SynapseToSQL

General Source Sink Mapping Settings User properties

Source dataset \* AzureSynapseAnalyticsDatasetTable Open New Preview data Learn more

Use query Table Query Stored procedure

Query timeout (minutes) 120

Isolation level Select...

Partition option None Physical partitions of table Dynamic range

Please preview data to validate the partition settings.

Additional columns New

InPrivate Catalog Exp. LakeHouse\_ mydatafactory mydatafacto SqlDatabase Tutorial - Cr Secret scope

Data Factory > mydatafactoryspace Search factory and documentation

Validate all Publish all Auto Save Preview experience On

MainPipeline AzureSynapseAnalyti... pipeline2\_datasour...

Azure Synapse Analytics AzureSynapseAnalyticsDatasetTable

Connection Schema Parameters

Linked service \* AzureSynapseAnalytics Test connection Edit New Learn more Connection successful

Table dbo . InventoryLogs Preview data Enter manually

The screenshot shows the Azure Synapse Studio interface with the following details:

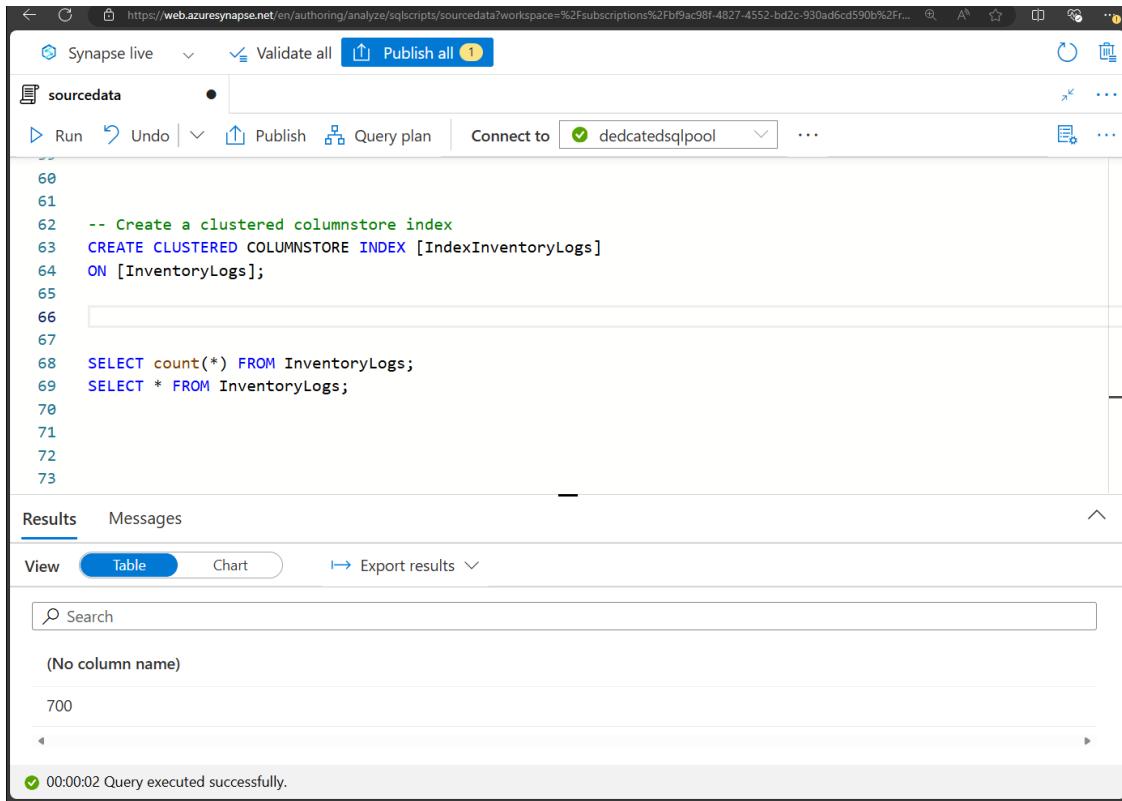
- Header:** The URL is <https://web.azure-synapse.net/en/authoring/analyze/sqlscripts/sourcedata?workspace=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2Fr...>. The top bar includes icons for back, forward, search, and other navigation.
- Toolbar:** Includes "Synapse live", "Validate all", "Publish all" (with a count of 1), and a refresh/circular arrow icon.
- Section Header:** "sourcedata" with a "..." dropdown.
- Tool Buttons:** Run, Undo, Publish, Query plan, Connect to (set to "dedicatedsqlpool"), and a "..." button.
- Code Area:** A large text area containing a SQL script to create the "InventoryLogs" table. The code is numbered from 1 to 24. It includes comments for dropping the table if it exists and creating it with specific columns, distribution, and indexing.

```
1 -- Check if the table exists and drop it if it does
2 IF EXISTS (SELECT * FROM sys.tables WHERE name = 'InventoryLogs' AND schema_id = SCHEMA_ID('dbo'))
3 BEGIN
4     DROP TABLE [InventoryLogs];
5 END
6
7
8 -- Create the InventoryLogs table with distribution and indexing
9 CREATE TABLE InventoryLogs (
10     LogID INT,
11     ProductID VARCHAR(10),
12     Action VARCHAR(20),
13     Quantity INT,
14     [Date] DATE
15 )
16 WITH
17 (
18     DISTRIBUTION = ROUND_ROBIN
19 );
20
21
22
23
24
--
```

The screenshot shows the Azure Synapse Studio interface with the following details:

- Header:** The URL is <https://web.azure-synapse.net/en/authoring/analyze/sqlscripts/sourcedata?workspace=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930adfc590b%2Fr...>. The top navigation bar includes tabs for "Synapse live", "Validate all", and "Publish all" (with 1 pending). There are also icons for refresh, search, and other common operations.
- Toolbar:** Below the header is a toolbar with buttons for "Run", "Undo", "Publish", "Query plan", "Connect to" (set to "dedicatedsqlpool"), and a "..." menu. To the right of the toolbar are icons for copy/paste and a refresh.
- Code Editor:** The main area contains a T-SQL script for generating random inventory logs. The code uses variables for product IDs, actions, start and end dates, and a loop to insert data into the "InventoryLogs" table. The script is numbered from 33 to 56.
- Status Bar:** At the bottom left, it says "Results" and "Messages". The "Messages" tab is active, showing a single message: "00:00:02 Query executed successfully."

Used clustered column store index for speed up data retrieval and analysis tasks, especially for large datasets.



The screenshot shows the Azure Synapse Studio interface. At the top, there's a navigation bar with 'Synapse live' and 'Validate all' buttons, followed by a 'Publish all' button with a count of 1. Below the navigation is a toolbar with 'Run', 'Undo', 'Publish', 'Query plan', 'Connect to' (set to 'dedicatedsqlpool'), and other options. The main area contains a code editor with the following SQL script:

```
60
61
62 -- Create a clustered columnstore index
63 CREATE CLUSTERED COLUMNSTORE INDEX [IndexInventoryLogs]
64 ON [InventoryLogs];
65
66
67
68 SELECT count(*) FROM InventoryLogs;
69 SELECT * FROM InventoryLogs;
70
71
72
73
```

Below the code editor is a results pane. It has tabs for 'Results' (selected) and 'Messages'. Under 'Results', there are 'View' and 'Table' buttons (the latter is selected), and an 'Export results' dropdown. A search bar is also present. The results table shows one row with the value '700'. At the bottom of the results pane, a message states '00:00:02 Query executed successfully.'

https://adf.azure.com/en/authoring/dataset/AzureSynapseAnalyticsDatasetTable?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd59... keerthanaakkula@gmail.com DEFAULT DIRECTORY

Data Factory > mydatafactoryspace Search factory and documentation

Edit linked service

Azure Synapse Analytics Learn more

Connect via integration runtime \* ⓘ

AutoResolveIntegrationRuntime

Version

Recommended (radio button) Legacy (radio button)

Account selection method ⓘ

From Azure subscription (radio button) Enter manually (radio button)

Fully qualified domain name \*

mywkrakshi.sql.azuresynapse.net

Database name \*

dedicatedsqlpool

Authentication type \*

SQL authentication

User name \*

sqladminuser

Password (radio button) Azure Key Vault (radio button)

Password \*

.....

Save Cancel Test connection

Azure Synapse Analytics Dataset Table

pipeline2\_datasource MainPipeline

Connection Schema Parameters

Linked service \* AzureSynapseAnalytics

Table dbo Enter ma

https://adf.azure.com/en/authoring/pipeline/pipeline2\_datasource?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd59... keerthanaakkula@gmail.com DEFAULT DIRECTORY

Data Factory > mydatafactoryspace Search factory and documentation

Preview experience On

Preview data

Linked service: AzureSynapseAnalytics

Object: dbo.InventoryLogs

#	LogID	ProductID	Action	Quantity	Date
1	16	P002	Received	9	05/25/2024
2	55	P001	Received	9	03/29/2024
3	24	P002	Sold	5	01/12/2024
4	88	P001	Received	5	04/19/2024
5	4	P002	Sold	7	05/15/2024
6	10	P002	Sold	5	03/11/2024
7	66	P001	Sold	8	03/16/2024
8	59	P001	Sold	2	05/22/2024
9	18	P003	Received	9	03/10/2024
10	90	P002	Sold	3	03/22/2024

Microsoft Azure | Data Factory > mydatafactory | Search factory and documentation

InPrivate CustomerData\_ mywkrakshi - A SqlDatabase All resources mydatafactory Copilot Fixing Sales Tra + keerthanaakkula@gmail.com DEFAULT DIRECTORY

Validate all Publish all Auto Save Preview experience On

pipeline AzureSqlTable MainPipeline

Validate Validate copy runtime Cancel options Add trigger

Copy data

SynapseToSQL

General Source Sink Mapping Settings User properties

Sink dataset \* AzureSqlTable Open New Learn more

Write behavior Insert Upsert Stored procedure

Use TempDB

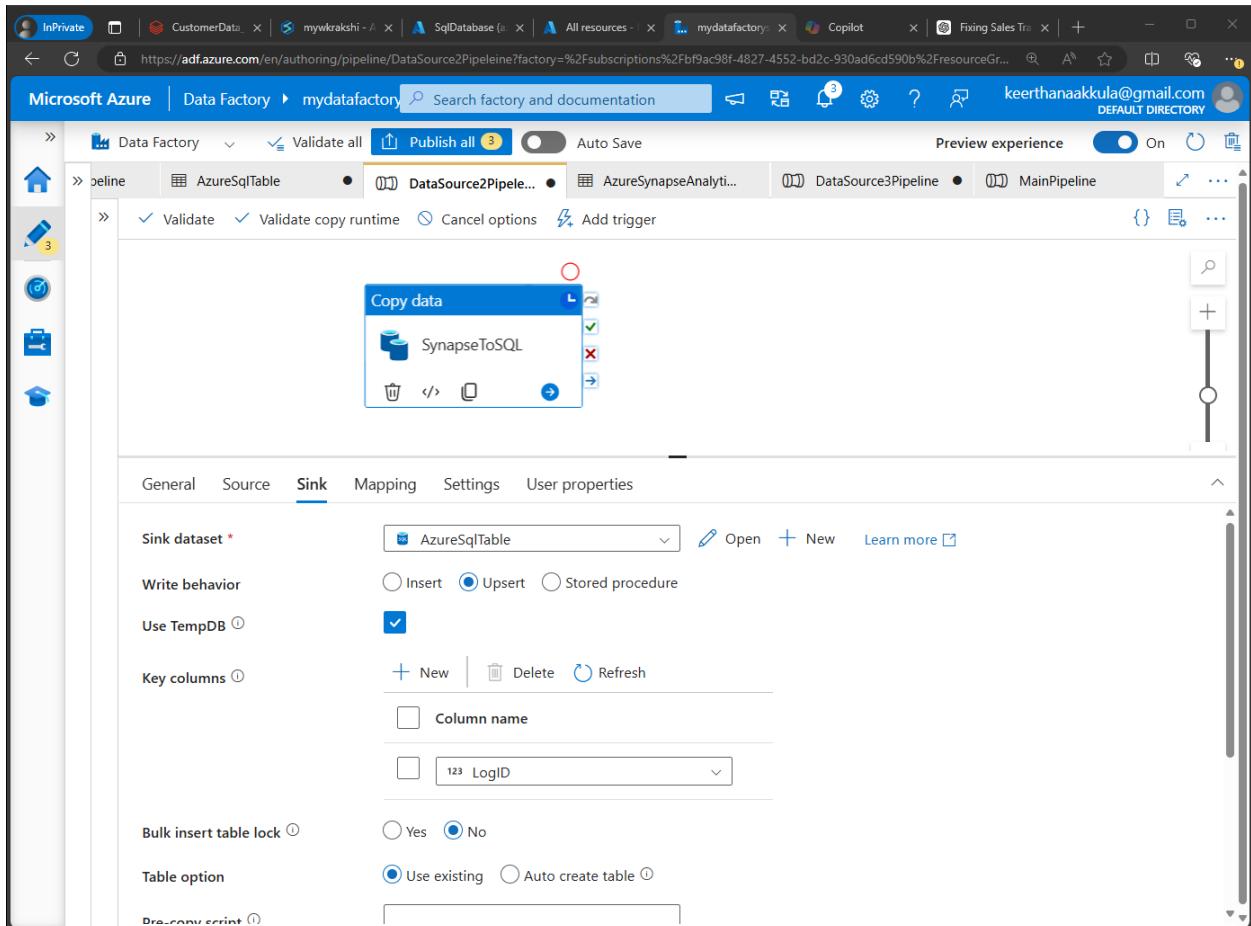
Key columns New Delete Refresh

Column name LogID

Bulk insert table lock Yes No

Table option Use existing Auto create table

Drop-copy script



InPrivate Catalog LakeH mywkr mydatafactory SqlDat dedcat others Tutorial Secret + keerthanaakkula@gmail.com DEFAULT DIRECTORY

Data Factory > mydatafactoriespace | Search factory and documentation

Validate all Publish all Auto Save Preview experience On

pipeline2\_datasource MainPipeline AzureSqlTable

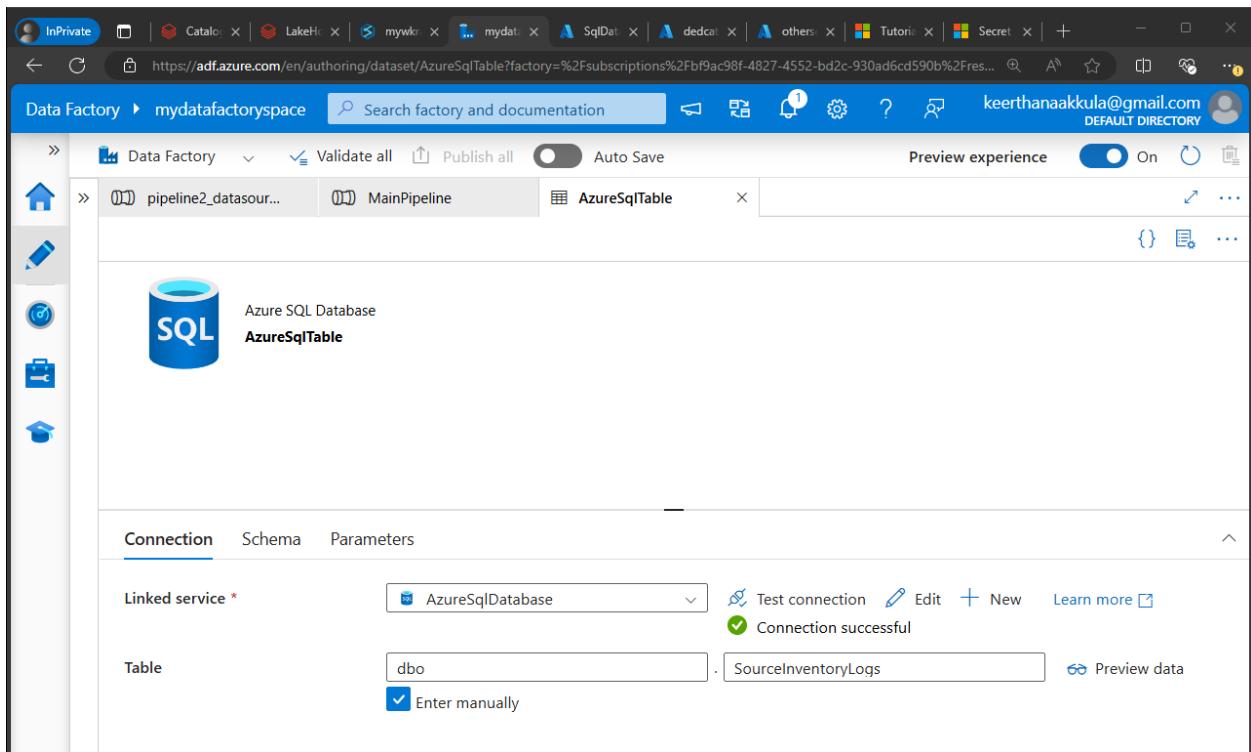
Azure SQL Database

AzureSqlTable

Connection Schema Parameters

Linked service \* AzureSqlDatabase Test connection Edit New Learn more Connection successful

Table dbo SourceInventoryLogs Enter manually Preview data



https://adf.azure.com/en/authoring/dataset/AzureSqlTable?factory=%2Fsubscriptions%2Fb9ac98f-4827-4552-bd2c-930ad6cd590b%2Fres...

Data Factory > mydatafactoryspace

Edit linked service

Azure SQL Database [Learn more](#)

Connect via integration runtime \* [AutoResolveIntegrationRuntime](#)

Version

Recommended  Legacy

Account selection method [From Azure subscription](#)  Enter manually

Fully qualified domain name \*

azureserverrakshitha.database.windows.net

Database name \*

SqlDatabase

Authentication type \*

SQL authentication

User name \*

rakshitha

Password [Azure Key Vault](#)

Password \*

.....

Save Cancel Test connection

https://adf.azure.com/en/authoring/pipeline/DataSource2Pipeline?factory=%2Fsubscriptions%2Fb9ac98f-4827-4552-bd2c-930ad6cd590b%2FresourceGr...

Microsoft Azure | Data Factory > mydatafactory

Validate all Publish all Auto Save Preview experience On

DataSource3Pipeline MainPipeline DataSource2Pipeline

Validate Debug Add trigger

Copy data SynapseToSQL

Parameters Variables Settings Output

Pipeline run ID: 8818ec5c-62e6-4b12-8e9a-1effaefa9c2b

Pipeline status Succeeded

View debug run consumption

All status

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	Use
SynapseToSQL	Succeeded	Copy data	6/19/2024, 12:15:19 PM	13s	AutoResolveIntegratio	

Monitor in Azure Metrics Export to CSV

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The URL is https://web.azure-synapse.net/en/authoring/analyze/sqlscripts/sourcedata?workspace=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2Fr... . The workspace name is mywkrakshi. The notebook titled 'sourcedata' is open, showing the following SQL code:

```
48  
49  
50  SELECT count(*) FROM InventoryLogs;  
51  SELECT * FROM InventoryLogs;  
52
```

The results pane shows the output of the query:

(No column name)  
700

A message at the bottom indicates: 00:00:02 Query executed successfully.

The screenshot shows the Microsoft Azure SQL Database Query editor (preview) interface. The URL is https://portal.azure.com/#@keerthanaakkulagmail.onmicrosoft.com/resource/subscriptions/bf9ac98f-4827-4552-bd2c-930ad6cd590b/resourceGroups/ETL... . The database name is SqlDatabase (azureserverrakshitha/SqlDatabase). The query editor contains the following SQL code:

```
1  SELECT count(*) FROM [dbo].[SourceInventoryLogs]
```

The results pane shows the output of the query:

700

A message at the bottom indicates: Query succeeded | 0s

- **Customer data: Through Database connectors, Notebook Activity, Source: Azure database for MySQL (Azure data studio), sink: Azure SQL database**



We need to import the mysql connector jar file into databricks and connect to Azure database for MySQL via JDBC connections

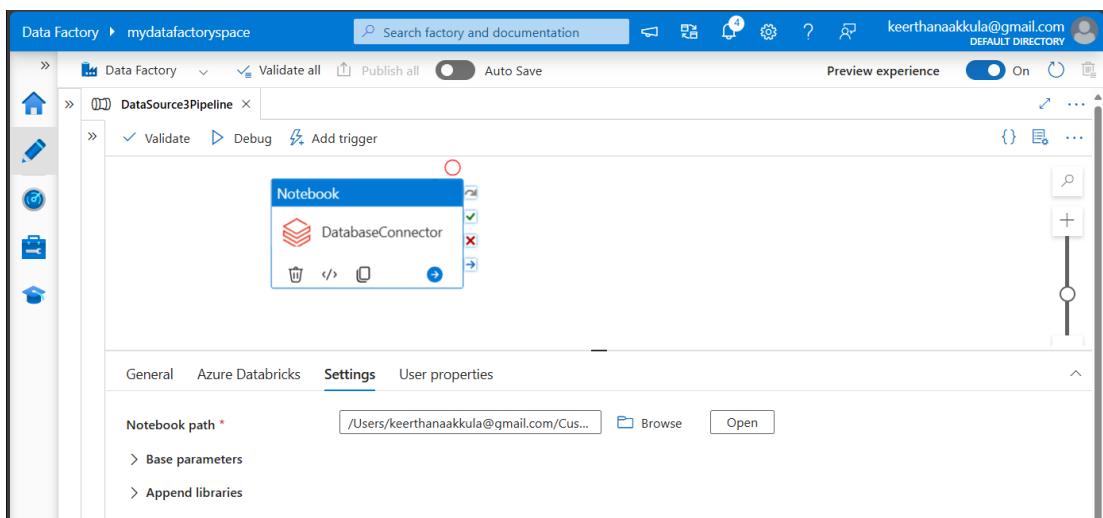
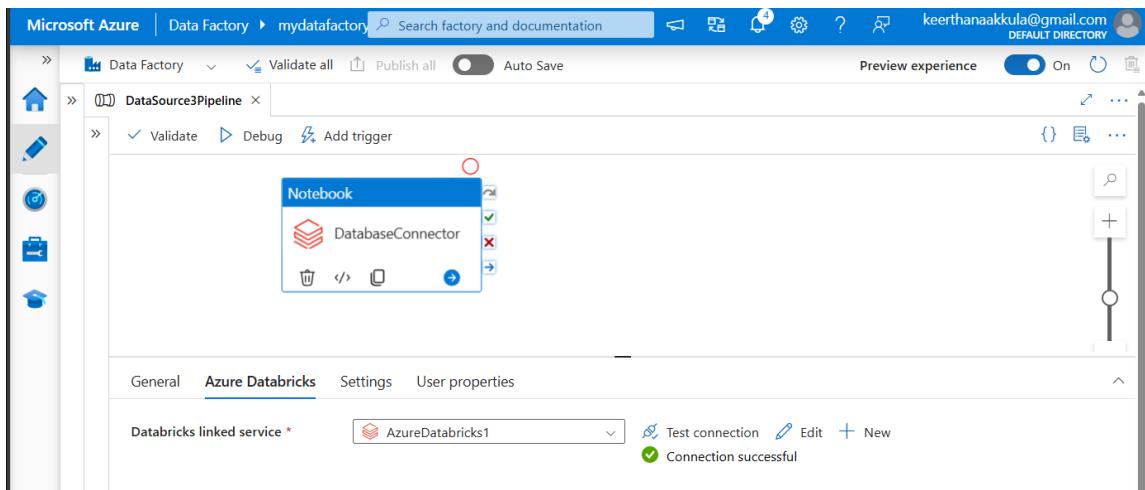
To connect to an Azure Database for MySQL from Databricks using JDBC, you need to import the MySQL Connector/J jar file into your Databricks environment. This connector allows Databricks to interact with the MySQL database over JDBC (Java Database Connectivity)

The screenshot shows the 'Compute > HEM MARYADA's Cluster' page in Databricks. The 'Libraries' tab is selected. A table lists the imported JAR file:

Status	Name	Type	Source
<input type="checkbox"/>	mysql-connector-j-8.4.0.jar	JAR	/Workspace/Users/keerthanaakkula@gmail.c...

The screenshot shows the 'mydatafactoryspace' Data Factory pipeline named 'DataSource3Pipeline'. The 'Notebook' activity is selected, showing a 'DatabaseConnector' component. The pipeline configuration details are as follows:

- General** tab:
  - Name \***: DatabaseConnector
  - Description**: This pipeline will get the source data from Azure database for MySQL (via Azure Data Studio) to Azure SQL
  - Activity state**: Activated (radio button selected)
  - Timeout**: 0.12:00:00
  - Retry**: 0



**Edit linked service**

Azure Databricks [Learn more](#)

Autoscale in Integration Runtime

Account selection method \*

From Azure subscription  Enter manually

Databrick Workspace URL \* ⓘ

Authentication type \*

[Access token](#) [Azure Key Vault](#)

Access token \* ⓘ

Select cluster

New job cluster  Existing interactive cluster  Existing instance pool

Existing cluster ID \* ⓘ

 Connection successful

 Test connection

[Save](#) [Cancel](#)

InPrivate CustomerD... mywkrash... SqlDatabase... otherserver... mydatafact... CustomerD... Copilot Fixing Sales... https://portal.azure.com/#@keerthanaakkula@gmail.onmicrosoft.com/resource/subscriptions/bf9ac98f-4827-4552-bd2c-930ad6cd590b/resourceGroups/ETL...

Microsoft Azure Search resources, services, and docs (G+/-) keerthanaakkula@gmail... DEFAULT DIRECTORY

Home > otherservername Azure Database for MySQL flexible server

Search Connect View process list Delete Reset password Restore Restart Stop Refresh ...

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Learning center

Settings Compute + storage Networking Databases Connect Server parameters Replication Maintenance High availability Backup and restore Advisor recommendations Locks

**Essentials**

Subscription ([move](#)) [Azure subscription 1](#) Server name otherservername.mysql.database.azure.com

Subscription ID bf9ac98f-4827-4552-bd2c-930ad6cd590b Server admin login name rakshitha

Resource group ([move](#)) [ETLRG](#) Configuration [General Purpose, D2ads v5, 2 vCores, 8 GiB RAM, 64 GiB st...](#)

Status Available MySQL version 8.0

Location East US Availability zone 3

Created On 2024-06-17 16:56:05.8728877 UTC

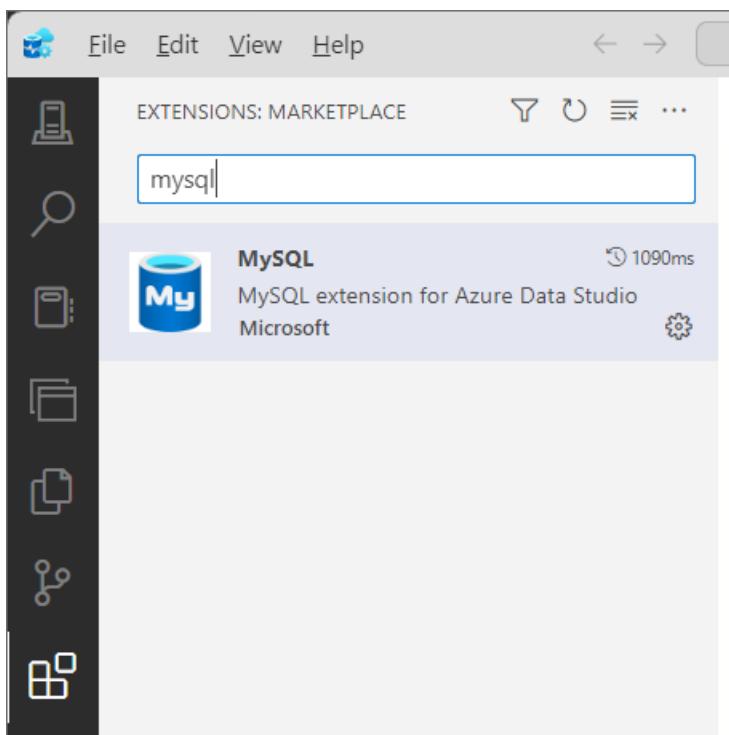
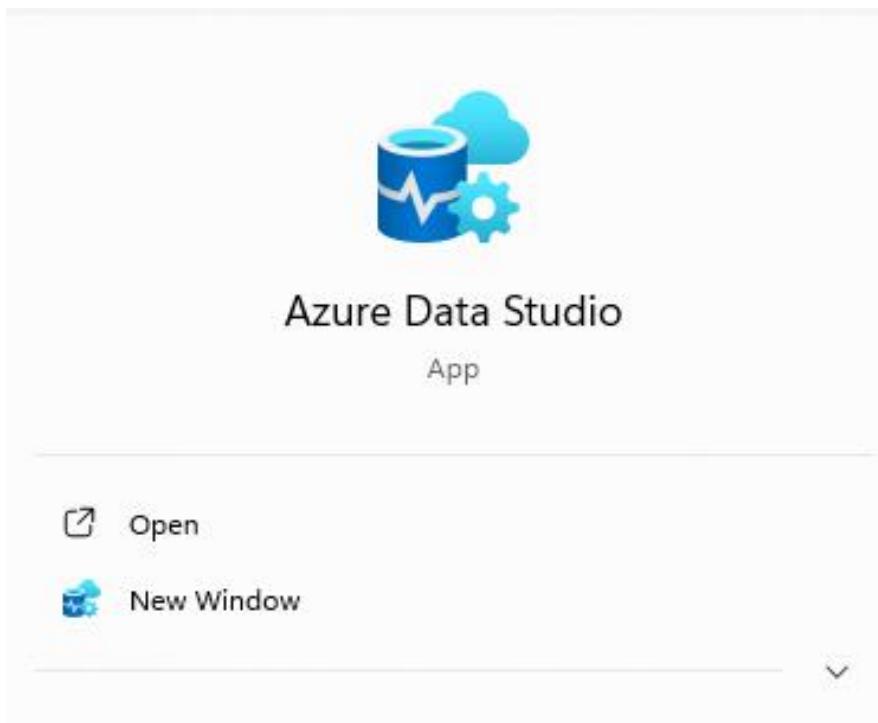
Tags ([edit](#)) [Add tags](#)

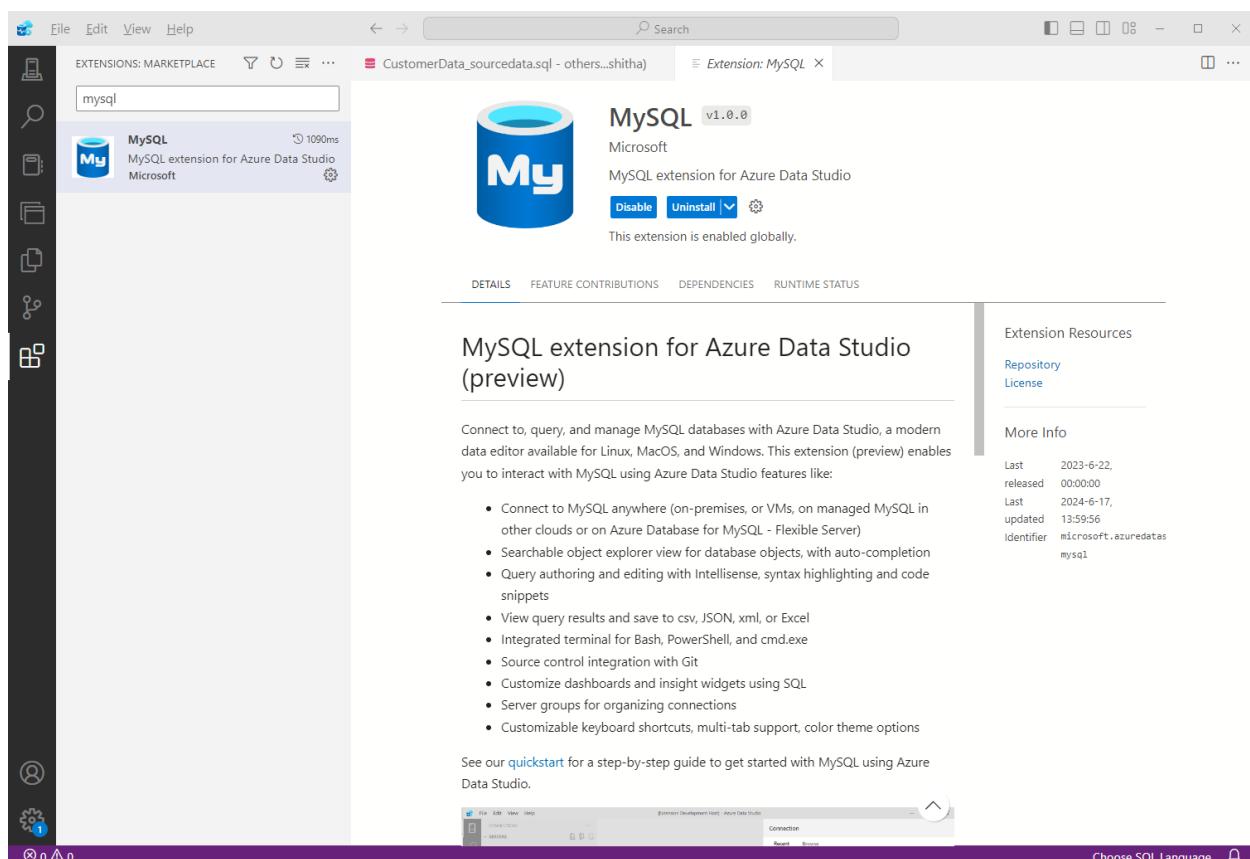
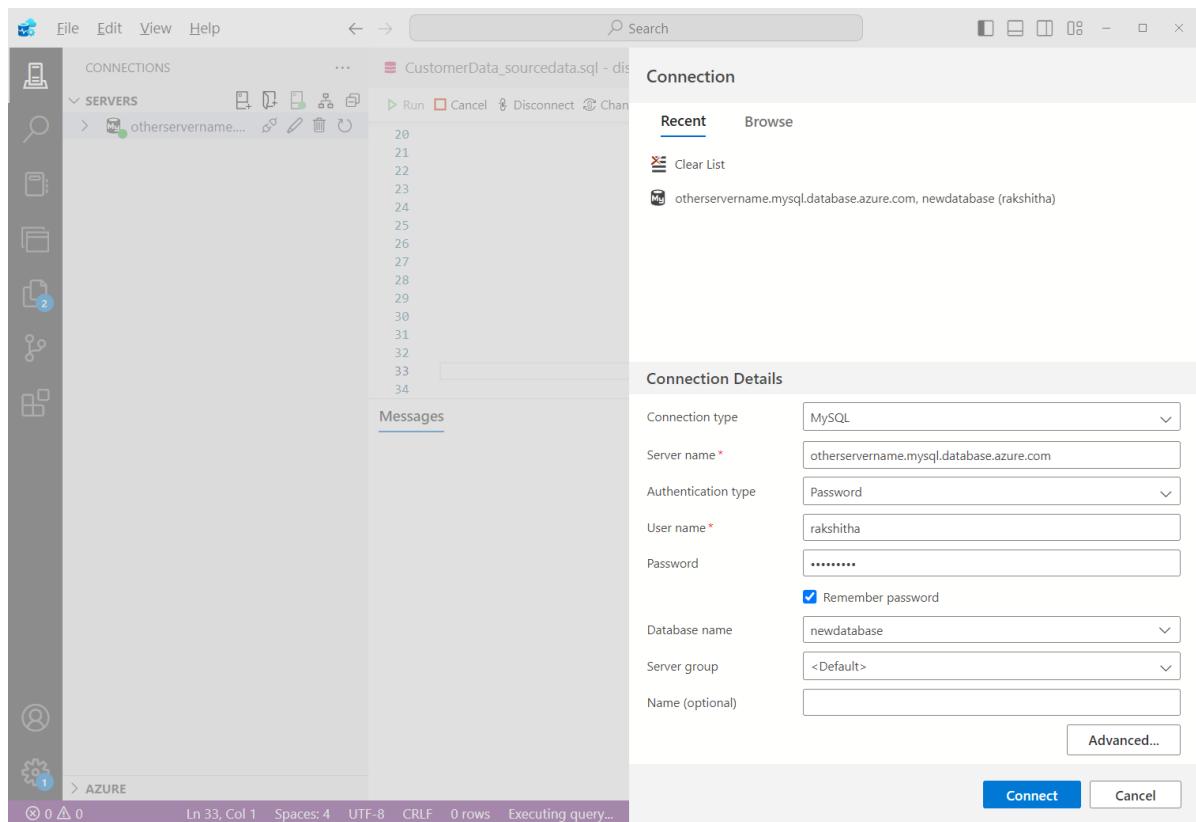
[Getting started](#) Properties Recommendations Monitoring Tutorials

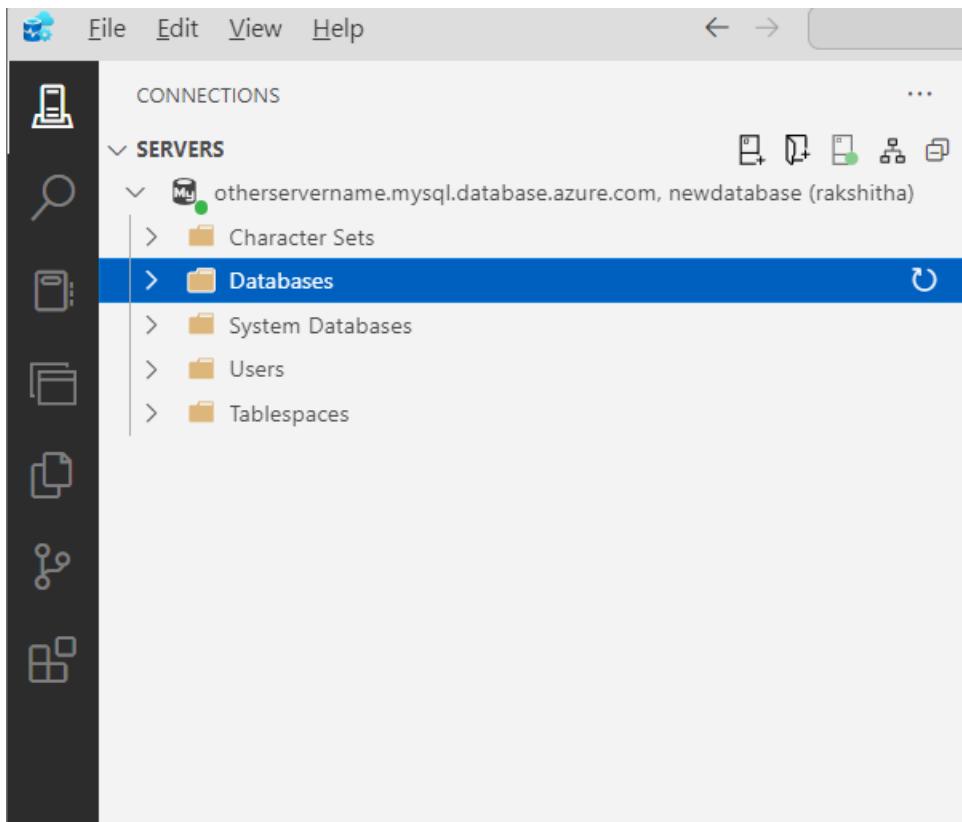
**Start your project**

Connect to your database for the first time with a few simple steps.

101 Learn Allow access







The screenshot shows the MySQL Workbench interface with the SQL editor open. The connection 'otherservername.mysql.database.azure.com, newdatabase (rakshitha)' is selected. In the SQL pane, the following query is run:

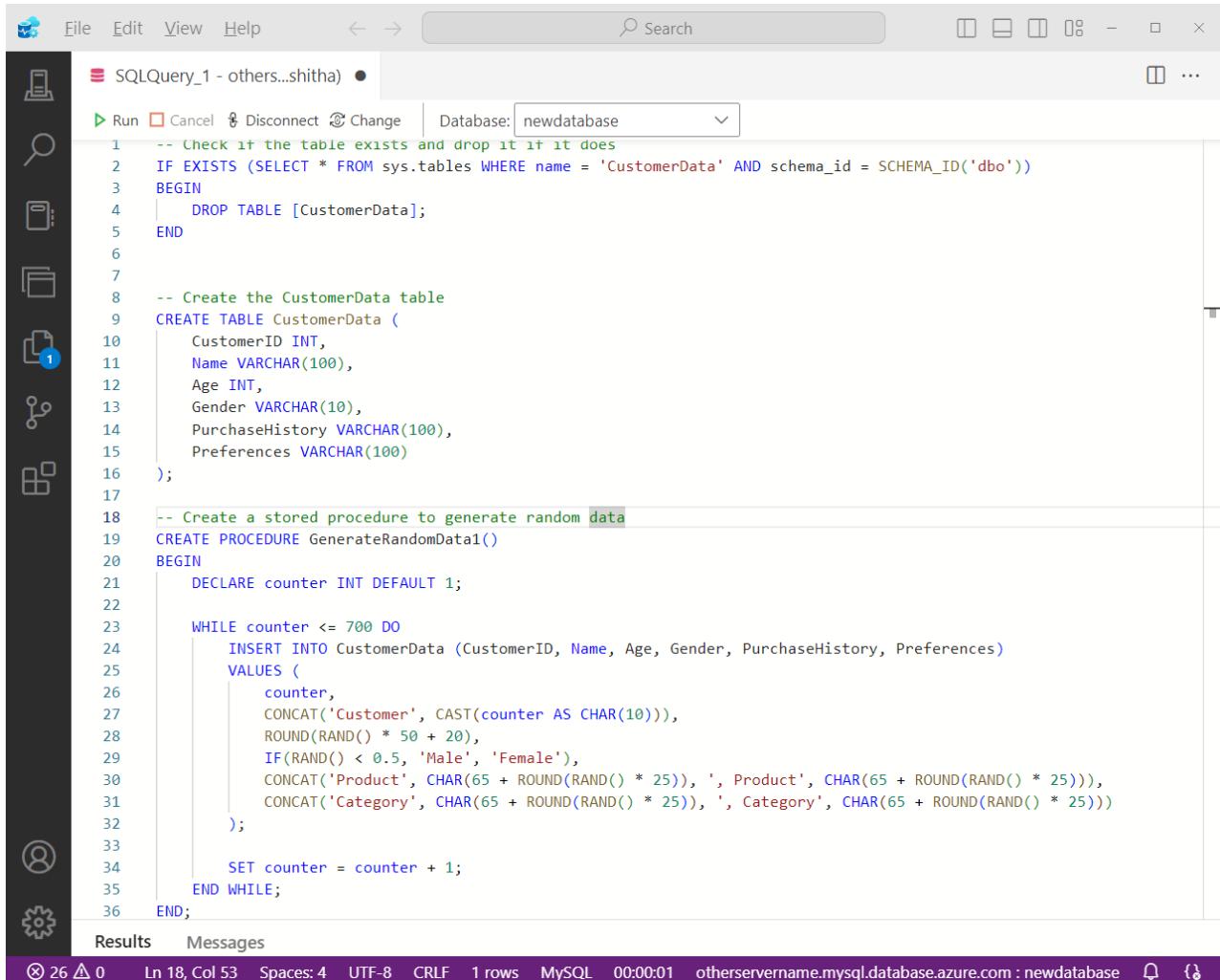
```
41 -- Output the generated customerdata
42
43
44 SELECT count(*) FROM CustomerData;
45
46
```

The results pane displays the output of the query:

count(*)
1 700

The bottom status bar shows the following information: Line 26, Column 0; Line 42, Column 1; Spaces: 4; UTF-8; CRLF; 1 rows; MySQL; 00:00:01; otherservername.mysql.database.azure.com : newdatabase.

Created a random customer data in Azure Database for MySQL flexible server via Azure data studio



The screenshot shows the Microsoft Azure Data Studio interface. The left sidebar has icons for file operations, database management, and connectivity. The main window is titled "SQLQuery\_1 - others...shitha". It contains the following SQL script:

```
1 -- Check if the table exists and drop it if it does
2 IF EXISTS (SELECT * FROM sys.tables WHERE name = 'CustomerData' AND schema_id = SCHEMA_ID('dbo'))
3 BEGIN
4     DROP TABLE [CustomerData];
5 END
6
7
8 -- Create the CustomerData table
9 CREATE TABLE CustomerData (
10     CustomerID INT,
11     Name VARCHAR(100),
12     Age INT,
13     Gender VARCHAR(10),
14     PurchaseHistory VARCHAR(100),
15     Preferences VARCHAR(100)
16 );
17
18 -- Create a stored procedure to generate random data
19 CREATE PROCEDURE GenerateRandomData1()
20 BEGIN
21     DECLARE counter INT DEFAULT 1;
22
23     WHILE counter <= 700 DO
24         INSERT INTO CustomerData (CustomerID, Name, Age, Gender, PurchaseHistory, Preferences)
25             VALUES (
26                 counter,
27                 CONCAT('Customer', CAST(counter AS CHAR(10))),
28                 ROUND(RAND() * 50 + 20),
29                 IF(RAND() < 0.5, 'Male', 'Female'),
30                 CONCAT('Product', CHAR(65 + ROUND(RAND() * 25)), ', Product', CHAR(65 + ROUND(RAND() * 25))),
31                 CONCAT('Category', CHAR(65 + ROUND(RAND() * 25)), ', Category', CHAR(65 + ROUND(RAND() * 25)))
32             );
33
34             SET counter = counter + 1;
35     END WHILE;
36 END;
```

The status bar at the bottom shows: ⑧ 26 △ 0 Ln 18, Col 53 Spaces: 4 UTF-8 CRLF 1 rows MySQL 00:00:01 otherservername.mysql.database.azure.com : newdatabase



CustomerData\_DatabaseConnector.py.html

Microsoft Azure | databricks

CustomerData\_DatabaseConnector.py Python

```
from pyspark.sql import SparkSession
# Initialize the SparkSession
spark = SparkSession.builder \
    .appName("MySQL to SQL Database Transfer") \
    .getOrCreate()

# JDBC URL for MySQL
mysql_url = "jdbc:mysql://otherverservername.mysql.database.azure.com:3306/newdatabase"
mysql_properties = {
    "user": "rakshitha",
    "password": "Vasavi@06",
    "driver": "com.mysql.cj.jdbc.Driver"
}

# JDBC URL for SQL Server
sqlserver_url = "jdbc:sqlserver://azureserverrakshitha.database.windows.net:1433;database=SqlDatabase"
sqlserver_properties = {
    "user": "rakshitha",
    "password": "Vasavi@06",
    "driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
}

# Read data from MySQL
df = spark.read.jdbc(url=mysql_url, table="CustomerData", properties=mysql_properties)

display(df)
df.write.jdbc(url=sqlserver_url, table="TargetCustomerData", mode="append", properties=sqlserver_properties)
```

Microsoft Azure | databricks

CustomerData\_DatabaseConnector.py Python

Table

#	CustomerID	Name	Age	Gender	PurchaseHistory	Preferences
1	1	Customer1	51	Female	ProductI, ProductT	CategoryU, CategoryR
2	2	Customer2	22	Male	ProductR, ProductW	CategoryJ, CategoryG
3	3	Customer3	23	Female	ProductX, ProductR	CategoryQ, CategoryE
4	4	Customer4	67	Male	ProductD, ProductB	CategoryW, Categor...
5	5	Customer5	47	Male	ProductL, ProductC	CategoryE, CategoryO
6	6	Customer6	36	Female	ProductL, ProductP	CategoryQ, CategoryJ
7	7	Customer7	68	Female	ProductS, ProductL	CategoryB, CategoryX
8	8	Customer8	42	Male	ProductY, ProductK	CategoryD, CategoryM
9	9	Customer9	69	Female	ProductP, ProductK	CategoryF, CategoryV
10	10	Customer10	46	Male	ProductB, ProductT	CategoryT, CategoryM
11	11	Customer11	24	Male	ProductW, ProductI	CategoryW, Categor...
12	12	Customer12	69	Male	ProductT, ProductU	CategoryR, CategoryY
13	13	Customer13	59	Female	ProductQ, ProductH	CategoryN, CategoryQ
14	14	Customer14	53	Male	ProductX, ProductL	CategoryK, CategoryR
15	15	Customer15	36	Male	ProductJ, ProductK	CategoryZ, CategoryR
16	16	Customer16	40	Male	ProductY, ProductS	CategoryS, CategoryJ
17	17	Customer17	57	Female	ProductS, ProductU	CategoryU, CategoryR
18	18	Customer18	21	Male	ProductZ, ProductW	CategoryN, Category...
19	19	Customer19	61	Female	ProductE, ProductG	CategoryS, CategoryU
20	20	Customer20	66	Male	ProductD, ProductB	CategoryV, CategoryD

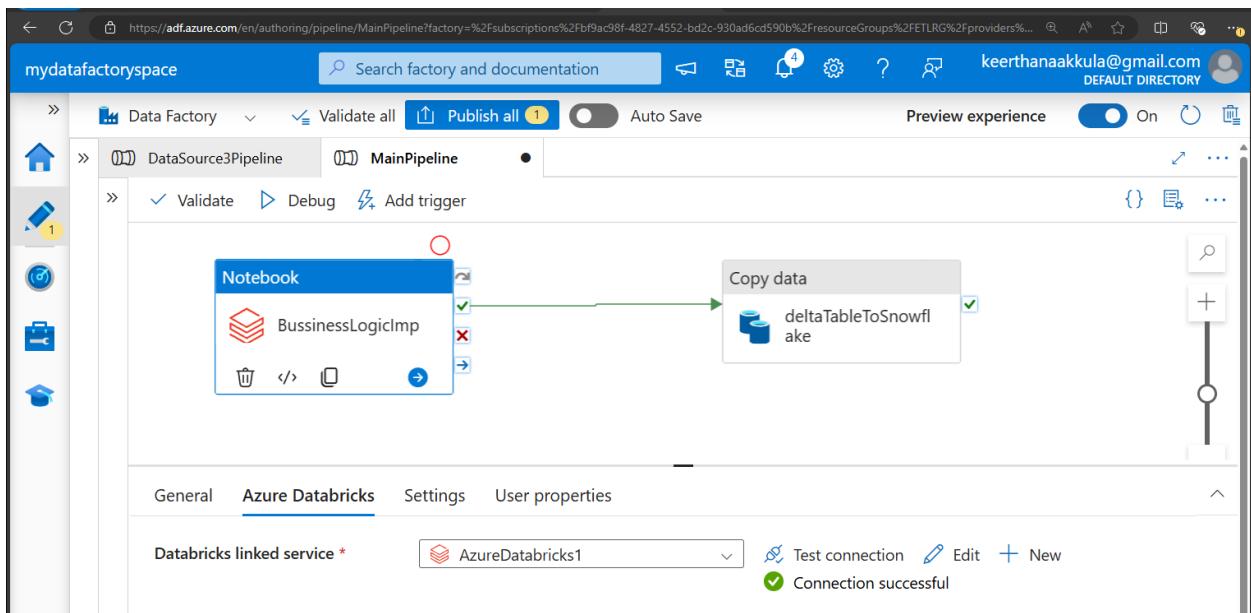
700 rows | 17.18 seconds runtime

Refreshed 2 minutes ago

The screenshot shows the Azure Data Factory pipeline run status page for a pipeline named "DataSource3Pipeline". The pipeline run ID is 578d4796-b676-4fb0-a9e9-9361f0018d32. The status is Succeeded. The pipeline consists of one activity, "DatabaseConnector", which is also marked as Succeeded. The activity type is Notebook. The run started at 6/19/2024, 1:10:23 PM and completed in 35s. The integration runtime used was AutoResolve.

This screenshot is identical to the one above, but it includes a modal window titled "Details" for the first activity, "DatabaseConnector". The modal shows the duration as 00:00:31 and the run page URL as <https://adb-953747807129243.3.azuredatabricks.net/?o=953747807129243#job/313940340361468/run/62426287>. The rest of the interface is the same, showing the pipeline run status and activity list.

- 2. Data Storage: Now all the data sources are stored in Azure SQL Database:** Now all the 3 data sources are in Azure SQL Database, I have created a pipeline to copy the data from Azure SQL Database to Snowflake
- 3. Data Transformation: Pyspark (notebook):** Created a notebook that implements the **Lakehouse Architecture** (medallion)
- 4. Data Warehousing and Analytics: Snowflake, Power BI,** I have created a pipeline to copy the data from Azure SQL Database to Snowflake



The Medallion Architecture, also known as the Lakehouse Architecture, is a data management pattern that organizes data in layers to improve data quality, processing efficiency, and maintainability. This architecture typically involves three layers:

- 1. Bronze (Raw Data) Layer:** Stores raw data in its original format.
- 2. Silver (Refined Data) Layer:** Contains cleaned and processed data, ready for analytics.
- 3. Gold (Serving) Layer:** Consists of highly refined data used for reporting and dashboards.

LakeHouse\_ArchMainTransformation.py Python

```
1 ##### This code connects to an Azure SQL Database, reads specified tables, and ingests the data into a designated directory in Delta format, overwriting any existing data. The ingested data is part of the "Bronze layer," which typically represents raw or minimally processed data in a data lake or data warehouse architecture.
2
3 from pyspark.sql import SparkSession
4
5 # Initialize Spark Session
6 spark = SparkSession.builder.appName("BronzeLayer").getOrCreate()
7
8 # Define connection properties for Azure SQL Database
9 jdbc_url = "jdbc:sqlserver://azureserverrakshitha.database.windows.net:1433;database=SqlDatabase"
10 sqlserver_properties = {
11     "user": "rakshitha",
12     "password": "Vasavi@06",
13     "driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
14 }
15
16 # List of tables to be ingested from Azure SQL Database
17 tables = ["SourceSalesTransactions", "SourceInventoryLogs", "SourceCustomerData"]
18
19 # Base path for storing Bronze layer data
20 bronze_base_path = "/bronze"
21
22 # Function to read and write each table
23 def ingest_table(table_name):
24     query = f"(SELECT * FROM {table_name}) AS {table_name}"
25     df = spark.read.jdbc(url=jdbc_url, table=query, properties=sqlserver_properties)
26     df.write.format("delta").mode("overwrite").save(bronze_base_path + table_name)
27
28 # Iterate over the list of tables and ingest each one
29 for table in tables:
30     ingest_table(table)
31
32 print("Data ingestion to Bronze layer completed.")
33
```

LakeHouse\_ArchMainTransf... Python

```
1 # Dictionary to store DataFrames
2 dataframes = {}
3
4 # Function to read each table into a DataFrame
5 def read_table_to_df(table_name):
6     df = spark.read.format("delta").load(bronze_base_path + table_name)
7     return df
8
9 # Iterate over the list of tables and read each one into a DataFrame
10 for table in tables:
11     dataframes[table] = read_table_to_df(table)
12
13 # Now you can access the DataFrames from the dictionary
14 df_transaction = dataframes["SourceSalesTransactions"]
15 df_inventory = dataframes["SourceInventoryLogs"]
16 df_customer = dataframes["SourceCustomerData"]
17
18 # Display the DataFrames to verify the data
19 df_customer.show()
20 df_transaction.show()
21 df_inventory.show()
22
```

LakeHouse\_ArchMainTransformation.py Python

```
1 from pyspark.sql.functions import col, split, explode, trim, sum, avg
2
3 # Explode PurchaseHistory and Preferences
4 df_customer = df_customer.withColumn("PurchaseHistory", explode(split(col("PurchaseHistory"), ", ")))
5 df_customer = df_customer.withColumn("Preferences", explode(split(col("Preferences"), ", ")))
6
7 # Example data cleansing: Trimming spaces from strings
8 df_customer = df_customer.withColumn("PurchaseHistory", trim(col("PurchaseHistory")))
9 df_customer = df_customer.withColumn("Preferences", trim(col("Preferences")))
10
11 # Join transaction data with customer data on CustomerID and drop duplicate CustomerID column
12 df_combined = df_transaction.join(df_customer, df_transaction.CustomerID == df_customer.CustomerID, "inner") \
13 .drop(df_customer.CustomerID)
14
15 # Selecting relevant columns for analysis
16 df_combined = df_combined.select(
17     df_combined.TransactionID,
18     df_combined.CustomerID,
19     df_combined.Name,
20     df_combined.Age,
21     df_combined.Gender,
22     df_combined.ProductID,
23     df_combined.Quantity,
24     df_combined.Timestamp,
25     df_combined.PurchaseHistory,
26     df_combined.Preferences
27 )
28
29
```

LakeHouse\_ArchMainTransformation.py Python

```
1 # Calculate total purchases and average purchase quantity per customer
2 df_features = df_combined.groupBy("CustomerID").agg(
3     sum("Quantity").alias("TotalPurchases"),
4     avg("Quantity").alias("AveragePurchaseQuantity")
5 )
6
7 # Join the features back with the combined DataFrame
8 df_combined = df_combined.join(df_features, "CustomerID", "inner")
9
10 # No missing values detected, Removing duplicates and Standardizing timestamp format
11
12 # Define schema for the data
13 schema = "CustomerID INT, TransactionID INT, Name STRING, Age INT, Gender STRING, ProductID STRING, Quantity INT, Timestamp TIMESTAMP, PurchaseHistory STRING, Preferences STRING, TotalPurchases INT, AveragePurchaseQuantity DOUBLE"
14
15 # Cleanse data
16 df_cleaned = df_combined.dropDuplicates() # Remove duplicates
17 df_cleaned = df_cleaned.withColumn("Timestamp", col("Timestamp").cast("timestamp"))
18
19 # Show cleansed data
20 print("Cleansed Data:")
21 df_cleaned.show(truncate=False)
22
23 # Show the final DataFrame with enriched data and new features
24 df_combined.show(truncate=False)
25 print("Row count:", df_combined.count())
26
27 # Define the Silver layer base path
28 silver_base_path = "/silver/"
29
30 # Write the transformed DataFrame to the Silver layer in Delta format
31 df_combined.write.format("delta").mode("overwrite").save(silver_base_path + "enriched_data")
```

LakeHouse\_ArchMainTransformation.py Python

```
from pyspark.sql.functions import sum, avg, month, year
# Path to the Silver layer
silver_base_path = "/silver/"

# Read the enriched data from the Silver layer
df_silver = spark.read.format("delta").load(silver_base_path + "enriched_data")

# Perform additional aggregations and transformations
df_gold = df_silver.groupBy(
    "CustomerID",
    year("Timestamp").alias("Year"),
    month("Timestamp").alias("Month")
).agg(
    sum("Quantity").alias("TotalPurchases"),
    avg("Quantity").alias("AveragePurchaseQuantity"
)

# Show the final DataFrame with aggregated data
display(df_gold)
print("Row count:", df_gold.count())

# Define the Gold layer base path
gold_base_path = "/gold/"

# Write the transformed DataFrame to the Gold layer in Delta format
df_gold.write.format("delta").mode("overwrite").save(gold_base_path + "monthly_customer_purchases")
```

LakeHouse\_ArchMainTransf... Python

```
# Read the Delta table from the Gold layer
df_gold = spark.read.format("delta").load("/gold/monthly_customer_purchases")

# Show the contents of the DataFrame
df_gold.show()

# Print the schema of the DataFrame
df_gold.printSchema()
```

The screenshot shows the Microsoft Azure portal interface. The user is navigating through the Azure Storage service, specifically the Container section. A new Shared Access Token (SAS) is being created for a container named 'container'. The token is configured with the following parameters:

- Permissions:** Read
- Start:** 2024-06-19 at 14:48:40 (UTC-05:00) Eastern Time (US & Canada)
- Expiry:** 2024-06-19 at 22:48:40 (UTC-05:00) Eastern Time (US & Canada)
- Allowed IP addresses:** None (example: 168.1.5.65 or 168.1.5.65-168.1...)
- Allowed protocols:** HTTPS only

Below these settings, two URLs are generated:

- Blob SAS token:** `sp=r&st=2024-06-19T18:48:40Z&se=2024-06-20T02:48:40Z&spr=https&sv=2022-11-02&sr=c&sig=X1g%2BjwDpWOWkfITz%2BTzXb6OIG4mYOxE7mv3a...`
- Blob SAS URL:** `https://adlsrakshitha.blob.core.windows.net/container?sp=r&st=2024-06-19T18:48:40Z&se=2024-06-20T02:48:40Z&spr=https&sv=2022-11-02&sr=c&sig=X1g%2BjwDpWOWkfITz%2BTzXb6OIG4mYOxE7mv3a...`

Keep the Blob SAS URL in the SAS URL and leave the SAS token blank in data factory as shown below

The screenshot shows the Microsoft Data Factory interface. A linked service named 'MainPipeline' is being edited. In the 'Activities' pane, an 'Azure Blob Storage' activity is selected. The 'Authentication type' is set to 'SAS URI', and the 'SAS URI' field contains the previously generated Blob SAS URL: `https://adlsrakshitha.blob.core.windows.net/container?sp=r&st=2024-06-19T18:48:40Z&se=2024-06-20T02:48:40Z&spr=https&sv=2022-11-02&sr=c&sig=X1g%2BjwDpWOWkfITz%2BTzXb6OIG4mYOxE7mv3a...`. The 'Test connection' button is highlighted, showing a successful connection.

[https://portal.azure.com/#view/Microsoft\\_Azure\\_Storage/ContainerMenuBlade/~/sas/storageAccountid/%2Fsubscriptions%2Fb9ac38f-4827-4552-bd2c-930ad6cd590b%2FresourceGroups%2FETLRG%2Fproviders%2F...](https://portal.azure.com/#view/Microsoft_Azure_Storage/ContainerMenuBlade/~/sas/storageAccountid/%2Fsubscriptions%2Fb9ac38f-4827-4552-bd2c-930ad6cd590b%2FresourceGroups%2FETLRG%2Fproviders%2F...)

Microsoft Azure  Azure Container Shared access tokens Container keerthanaakkula@gmail.com DEFAULT DIRECTORY (KEERTHAN...)

Home > adlsrakshitha | Containers > container

## container | Shared access tokens

Container

Start and expiry date/time Edit

Start  14:53:12

(UTC-05:00) Eastern Time (US & Canada) ▼

Expiry  22:53:12

(UTC-05:00) Eastern Time (US & Canada) ▼

Allowed IP addresses Edit  
for example, 168.1.5.65 or 168.1.5.65-168.1...

Allowed protocols Edit  
 HTTPS only  HTTPS and HTTP

Generate SAS token and URL

Blob SAS token Edit

Blob SAS URL Edit

<https://adlsrakshitha.blob.core.windows.net/container?sp=racwdlmeop&st=2024-06-19T18:53:12Z&se=2024-07-05T02:53:12Z&spr=https&sv=2022-11-02&sr=c&sig=nx9TJh%2FxXVXMxGuYXKf6UzKQFG4zjuD8kkwjcaWS0ws%3D>

mydatafactoryspace - Azure Data Factory

MainPipeline Validate all Publish all Auto Save

Activities Search activities

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

AzureDatabricksDelt... SnowflakeTable1

General Source Sink Mapping Settings User properties

You will be charged # of used DIUs \* copy duration \* \$0.25/DIU-hour. Local current

Maximum data integration unit   Use custom value

Degree of copy parallelism

Fault tolerance

Enable logging

Enable staging

Staging settings

Staging account linked service linkedService1

Storage path

Enable compression

Edit linked service Azure Blob Storage Learn more

Connect via integration runtime AutoResolveIntegrationRuntime

Authentication type SAS URI

SAS URL SAS token Azure Key Vault

SAS token Azure Key Vault

sample: ?sv=<storage services version>&st=<start time>&se=<expire time>&sr=<resc>

Test connection To linked service To file path

Annotations

Connection successful ✓ Apply Cancel Test connection

https://adf.azure.com/en/authoring/pipeline/MainPipeline\_copy1?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b...

Data Factory Validate all Publish all Auto Save Preview experience On

AzureDatabricksDeltaTable1 SnowflakeTable1 MainPipeline MainPipeline\_copy1

Activities Search activities

Notebook BussinessLogicImp

Copy data deltaTableToSnowflake

Name \* deltaTableToSnowflake Learn more

Description

Activity state Activated Deactivated

Timeout 0:12:00:00

Retry 0

Retry interval (sec) 30

Secure output Secure input

```
graph LR; Notebook[BussinessLogicImp] --> Copy[deltaTableToSnowflake]
```

InPrivate [2] mydatafactoryspace - Azure Data container - Microsoft Azure https://adf.azure.com/en/authoring/pipeline/MainPipeline\_copy1?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b...

Data Factory Validate all Publish all Auto Save Preview experience On

AzureDatabricksDeltaTable1 SnowflakeTable1 MainPipeline MainPipeline\_copy1

Activities Search activities

Notebook BussinessLogicImp

Copy data deltaTableToSnowflake

Source dataset \* AzureDatabricksDeltaLakeDataset1 Open New Preview data Learn more

Use query Table Query

Date format

Timestamp format

Additional columns New

```
graph LR; Notebook[BussinessLogicImp] --> Copy[deltaTableToSnowflake]
```

InPrivate [2] mydatafactoriespace - Azure Data... container - Microsoft Azure

https://adf.azure.com/en/authoring/dataset/AzureDatabricksDeltaLakeDataset1?factory=%2Fsubscriptions%2Fbfb9ac98f-4827-4552-bd2c-...

Data Factory Validate all Publish all Auto Save

AzureDatabricksDelt... SnowflakeTable1 MainPipeline

**Edit linked service**

Azure Databricks Delta Lake Learn more

Name \* AzureDatabricksDeltaLakeLinkedservice

Description

Connect via integration runtime \* AutoResolveIntegrationRuntime

Authentication method \* Access token

Account selection method From Azure subscription Enter manually

Domain \* https://adb-953747807129243.azuredatabricks.net

Existing cluster ID \* 0617-170452-lsnfmp78

Access token Azure Key Vault

Access token \* .....

Annotations

New Parameters Advanced

Save Cancel Test connection

Azure Databricks Delta Lake AzureDatabricksDeltaLakeDataset1

Connection Schema Parameters

Linked service \* AzureDatabricksDeltaLakeLinkedserv...

Database my\_database Edit

Table monthly\_customer\_purchases Edit

InPrivate [2] mydatafactoriespace - Azure Data... container - Microsoft Azure

https://adf.azure.com/en/authoring/dataset/AzureDatabricksDeltaLakeDataset1?factory=%2Fsubscriptions%2Fbfb9ac98f-4827-4552-bd2c-...

Data Factory Validate all Publish all Auto Save Preview experience On

AzureDatabricksDelt... SnowflakeTable1 MainPipeline MainPipeline\_copy1

**Azure Databricks Delta Lake** AzureDatabricksDeltaLakeDataset1

Connection Schema Parameters

Linked service \* AzureDatabricksDeltaLakeLinkedserv...

Test connection Edit New Learn more

Connection successful

Database my\_database Refresh

Table monthly\_customer\_purchases Refresh Preview data

InPrivate (2) mydatafactoryspace - Azure Data Factory container - Microsoft Azure

https://adf.azure.com/en/authoring/dataset/AzureDatabricksDeltaLakeDataset1?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2Fres...

Data Factory Validate all Publish all Auto Save Preview experience On

AzureDatabricksDelt... SnowflakeTable1 MainPipeline MainPipeline\_copy1

Azure Databricks Delta Lake  
AzureDatabricksDeltaLakeDataset1

Connection Schema Parameters

Import schema Clear

Column name	Type
CustomerID	integer
Year	integer
Month	integer
TotalPurchases	long
AveragePurchaseQuantity	double

InPrivate (2) mydatafactoryspace - Azure Data Factory container - Microsoft Azure

https://adf.azure.com/en/authoring/pipeline/MainPipeline?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2Fres...

Data Factory Validate all Publish all Auto Save Preview experience On

AzureDatabricksDelt... SnowflakeTable1 MainPipeline MainPipeline\_copy1

Activities < >

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

Validate Validate copy runtime Debug Add trigger

Notebook BussinessLogicImp

General Source Sink Mapping Settings User properties

Sink dataset \* SnowflakeTable1 Open New Learn more

Pre-copy script

Additional Snowflake copy options  
+ New

Additional Snowflake format options  
+ New

InPrivate (2) mydatafactoriespace - Azure Data container - Microsoft Azure

https://adf.azure.com/en/authoring/dataset/SnowflakeTable1?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2F...

Data Factory Validate all Publish all Auto Save Preview experience On

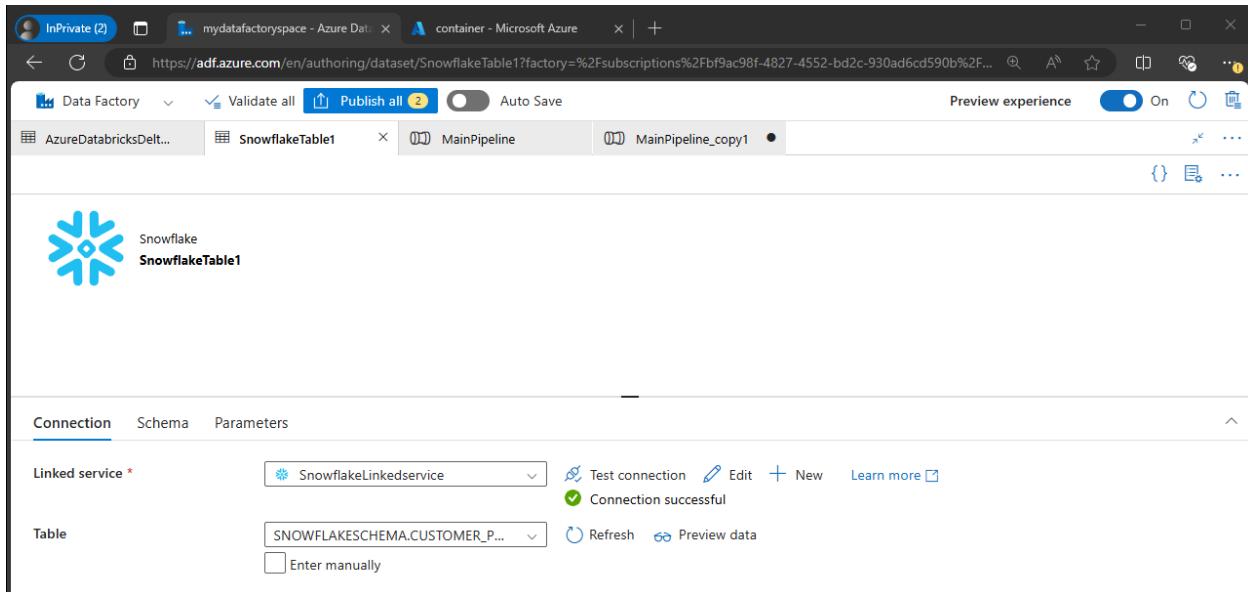
AzureDatabricksDelt... SnowflakeTable1 MainPipeline MainPipeline\_copy1

Snowflake SnowflakeTable1

Connection Schema Parameters

Linked service \* SnowflakeLinkedservice Test connection Edit + New Learn more Connection successful

Table SNOWFLAKESCHEMA.CUSTOMER\_P... Refresh Preview data Enter manually



InPrivate (2) mydatafactoriespace - Azure Data container - Microsoft Azure

https://adf.azure.com/en/authoring/dataset/SnowflakeTable1?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2F...

Data Factory Validate all Publish all Auto Save

AzureDatabricksDelt... SnowflakeTable1 MainPipeline

Edit linked service Snowflake Learn more

Name \* SnowflakeLinkedservice

Description

Connect via integration runtime \* AutoResolveIntegrationRuntime

Account name \* cuoipnx-qg65241

Database \* SNOWFLAKEDATABASE

Warehouse \* COMPUTE\_WH

Authentication type \* Basic

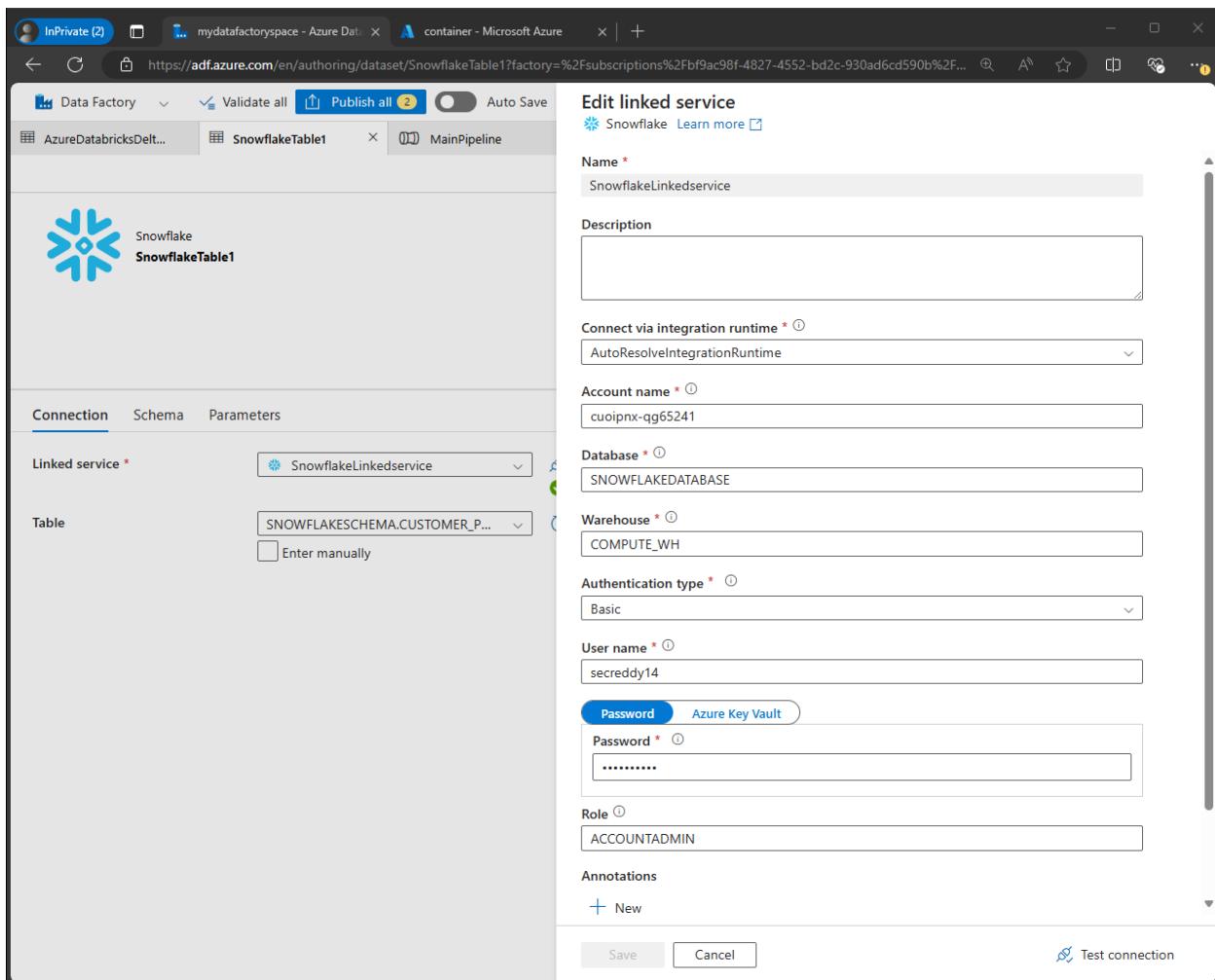
User name \* secreddy14

Password Azure Key Vault

Role ACCOUNTADMIN

Annotations + New

Save Cancel Test connection



InPrivate [2] mydatafactoryspace - Azure Data ... container - Microsoft Azure x | +

https://adf.azure.com/en/authoring/dataset/SnowflakeTable1?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2F...

Data Factory Validate all Publish all Auto Save Preview experience On

AzureDatabricksDelt... SnowflakeTable1 MainPipeline MainPipeline\_copy1

Snowflake SnowflakeTable1

Connection Schema Parameters

Import schema Clear

Column name	Type
CUSTOMERID	NUMBER
YEAR	NUMBER
MONTH	NUMBER
TOTALPURCHASES	NUMBER
AVERAGEPURCHASEQUANTITY	FLOAT

https://adf.azure.com/en/authoring/pipeline/MainPipeline\_copy1?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b...

Data Factory Validate all Publish all Auto Save Preview experience On

AzureDatabricksDelt... SnowflakeTable1 MainPipeline MainPipeline\_copy1

Activities

Search activities

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

Notebook BussinessLogicImp

Copy data deltaTableToSnowflake

Validate Validate copy runtime Debug Add trigger

General Source Sink Mapping Settings User properties

Import schemas Preview source New mapping Clear Reset Delete

Source	Type	Destination	Type
CustomerID	integer	CUSTOMERID	NUMBER
Year	integer	YEAR	NUMBER
Month	integer	MONTH	NUMBER
TotalPurchases	12 long	TOTALPURCHASES	12 NUMBER
AveragePurchaseQua...	1.2 double	AVERAGEPURCHASEQUA...	1.2 FLOAT

Add dynamic content [Alt+Shift+D]

The above tried approach is loading the data directly from delta lake table (DATABRICKS) to snowflake in data factory using linked services (SAS URL)

Alternative approach to load the data from blob storage(csv) to snowflake

---

This screenshot shows the Azure Storage Container settings for 'samplecontainername'. The container contains one blob named 'FinalData.csv' which was modified on 19/06/2024, 18:32:10 and is in the 'Hot (Inferred)' access tier. The blob type is a 'Block blob'.

This screenshot shows the Azure Data Factory pipeline editor. A 'Copy data' activity is connected from a 'BusinessLogicImp' notebook step to a 'blobTableToSnowflake' sink. The pipeline is named 'MainPipeline\_copy1'.

https://adf.azure.com/en/authoring/pipeline/MainPipeline\_copy1?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b...

Data Factory > mydatafactoriespace

main branch / saved / pipeline1 / MainPipeline\_copy1 / MainPipeline

Preview experience: On

Notebook: BusinessLogicImp

Copy data: blobTableToSnowflake

Source dataset: sourcedata

File path type: File path in dataset

Start time (UTC):

End time (UTC):

Filter by last modified:

Recursively: checked

Enable partitions discovery:

Max concurrent connections:

Skip line count:

Additional columns: + New

https://adf.azure.com/en/authoring/dataset/sourcedata?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2Fresou...

Data Factory > mydatafactoriespace

main branch / saved / pipeline1 / MainPipeline\_copy1 / MainPipeline

Preview experience: On

Dataset: DelimitedText sourcedata

Connection: snowflakedemo

Test connection: Connection successful

File path: samplecontainername / Directory / FinalData.csv

Compression type: Select...

Column delimiter: Comma (,)

Row delimiter: Line feed (\n)

Encoding: Default(UTF-8)

Quote character: Double quote (")

Escape character: Backslash (\)

First row as header:

Null value:

https://adf.azure.com/en/authoring/dataset/sourcedata?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2Fresource... keerthanaakkula@gmail.com DEFAULT DIRECTORY

Data Factory > mydatafactoryspace

main branch | Validate all | Save all | Publish | Auto Save | Preview experience | On

sourcedata | sinkdata | pipeline1 | MainPipeline\_copy1 | MainPipeline

Preview data

Linked service: snowflakedemo

Object: FinalData.csv

	Prop_0	Prop_1	Prop_2	Prop_3	Prop_4
1	102	2024	6	496	5.166666666666667
2	146	2024	6	248	5.166666666666667
3	180	2024	6	280	4.375
4	111	2024	6	488	6.777777777777778
5	154	2024	6	112	4.666666666666667
6	195	2024	6	424	5.888888888888889
7	122	2024	6	184	5.75
8	131	2024	6	496	5.166666666666667
9	124	2024	6	120	7.5
10	158	2024	6	384	6.857142857142857

Connection  
Linked service  
File path \*  
Compression t  
Column delimi  
Row delimiter  
Encoding  
Quote charact  
Escape character  
First row as header  
Null value

w data Detect format

https://adf.azure.com/en/authoring/dataset/sinkdata?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2Fresource... keerthanaakkula@gmail.com DEFAULT DIRECTORY

Data Factory > mydatafactoryspace

main branch | Validate all | Save all | Publish | Auto Save | Preview experience | On

sourcedata | sinkdata | pipeline1 | MainPipeline\_copy1 | MainPipeline

Snowflake sinkdata

Connection Schema Parameters

Linked service \* SnowflakeLinkedservice Test connection Edit + New Learn more Connection successful

Table SNOWFLAKESCHEMA.CUSTOMER\_P... Refresh Preview data Enter manually

https://adf.azure.com/en/authoring/dataset/sinkdata?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2Fresource... keerthanaakkula@gmail.com DEFAULT DIRECTORY

Data Factory > mydatafactoriespace

Search factory and documentation

Validate all Save all Publish Auto Save Preview experience On

main branch sourcadata sinkdata pipeline1 MainPipeline\_copy1 MainPipeline

Saved

Preview data

Linked service: SnowflakeLinkedservice

Object: SNOWFLAKESCHEMA.CUSTOMER\_PURCHASES

	CUSTOMERID	YEAR	MONTH	TOTALPURCHASES	AVERAGEPURCHASEQUANTITY
1	102	2024	6	496	5.166666666666667
2	146	2024	6	248	5.166666666666667
3	180	2024	6	280	4.375
4	111	2024	6	488	6.777777777777778
5	154	2024	6	112	4.666666666666667
6	195	2024	6	424	5.888888888888889
7	122	2024	6	184	5.75
8	131	2024	6	496	5.166666666666667
9	124	2024	6	120	7.5
10	158	2024	6	384	6.857142857142857

https://adf.azure.com/en/monitoring/pipelineruns/65e8ecf5-8bf6-4333-a2a2-e94f5a8d8728?factory=%2Fsubscriptions%2Fbf9ac98f-4827-4552-bd2c-930ad6cd590b%2FresourceGroups... keerthanaakkula@gmail.com DEFAULT DIRECTORY

Microsoft Azure | Data Factory > mydatafactoriespace

All pipeline runs > MainPipeline\_copy1 - Activity runs

Run by Manual trigger

Start time 6/19/2024, 6:47:13 PM

End time 6/19/2024, 6:55:07 PM

Status Succeeded

Parameters 0

Pipeline run ID 65e8ecf5-8bf6-4333-a2a2-e94f5a8d8728

Notebook → Copy data

Activity runs

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User prop
blobTableToSnowflake	Succeeded	Copy data	6/19/2024, 6:54:37 PM	29s	AutoResolveIntegratio	
BusinessLogicImp	Succeeded	Notebook	6/19/2024, 6:47:15 PM	7m 21s	AutoResolveIntegratio	

Microsoft Azure | Data Factory > mydatafactoryspace | Search factory and documentation

All pipeline runs > MainPipeline\_copy1 - Activity runs

Rerun Cancel Refresh Update pipeline List Gantt

Notebook BussinessLogicImp → Copy data blobTableToSnowflake

**Activity runs**

All status Monitor in Azure Metrics View run detail Export to CSV

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity run ID
blobTableToSnowflake	Succeeded	Copy data	6/19/2024, 6:54:37 PM	29s	AutoResolveIntegrator		faa723d3-
BussinessLogicImp	Succeeded	Notebook	6/19/2024, 6:47:15 PM	7m 21s	AutoResolveIntegrator		0868e8f6-

InPrivate [2] https://adb-953747807129243.azuredatabricks.net/jobs/704254141502331/runs/531198042607886?o=953747807129243

Microsoft Azure | databricks | Search data, notebooks, recents, and more... CTRL + P databricksworkspace H

Workflows > Runs > ADF\_mydatafactoryspace\_MainPipeline\_copy1\_BussinessLogicImp\_0868e8f6-e61c-4571-a9bb-deb90447addb run

Delete job run Repair run

**Output**

```
✓ 37.69 seconds Reading Data from Azure SQL Database
#####
# The code connects to an Azure SQL Database, reads specified tables, and ingests the data into a designated directory in Delta format, overwriting any existing data. The ingested data is part of the "Bronze layer," which typically represents raw or minimally processed data in a data lake or data warehouse architecture.

from pyspark.sql import SparkSession

# Initialize Spark Session
spark = SparkSession.builder.appName("BronzeLayer").getOrCreate()

# Define connection properties for Azure SQL Database
jdbc_url = "jdbc:sqlserver://azureserverrakshitha.database.windows.net:1433;database=SqlDatabase"
sqlserver_properties = {
    "user": "rakshitha",
    "password": "Vasavi@06",
    "driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
}

# List of tables to be ingested from Azure SQL Database
tables = ["SourceSalesTransactions", "SourceInventoryLogs", "SourceCustomerData"]

# Base path for storing Bronze layer data
bronze_base_path = "/bronze"

# Function to read and write each table
def ingest_table(table_name):
    query = f"SELECT * FROM {table_name} AS {table_name}"
    df = spark.read.jdbc(url=jdbc_url, table=query, properties=sqlserver_properties)
    df.write.format("delta").mode("overwrite").save(bronze_base_path + table_name)
```

**Task run details**

- Job ID: 704254141502331
- Task run ID: 531198042607886
- Run as: HEM MARYADA
- Started: 06/19/2024, 06:52:34 PM
- Ended: 06/19/2024, 06:54:24 PM
- Duration: 1m 50s
- Queue duration: -
- Status: Succeeded

**Notebook**

/Users/keerthanaakkula@gmail.com/LakeHouse\_ArchMainTransformation.py

**Compute**

HEM MARYADA's Cluster

Single node: Standard\_DS3\_v2 - DBR: 14.3.x-photon-scala2.12

**Permissions**

No permissions

Count: 100

Microsoft Azure | databricks

LakeHouse\_ArchMainTransformatio... Python Search data, notebooks, recents, and more... CTRL + P databricksworkspace ▾ ? H

File Edit View Run Help Last edit wa... Provide feedback

```
# Load the table from the database
df_gold = spark.sql("SELECT * FROM my_database.monthly_customer_purchases")

# Show the contents of the DataFrame
df_gold.count()

▶ (2) Spark Jobs
▶ df_gold: pyspark.sql.dataframe.DataFrame = [CustomerID: integer, Year: integer ... 3 more fields]
```

[Shift+Enter] to run and move to next cell  
[Esc H] to see all keyboard shortcuts

Microsoft Azure | Microsoft Azure Storage Blob Properties Blade V2

https://portal.azure.com/#view/Microsoft\_Azure\_Storage/BlobPropertiesBladeV2/storageAccountName/%2Fsubscriptions%2Fb9ac98f-4827-4552-bd2c-930ad6cd590b%2FresourceGroups%2Fmydata... | keerthanaakkula@gmail... DEFAULT DIRECTORY (KEERTHAN...)

Home > namestoragerakshitha | Containers > samplecontainername >

**samplecontainername** Container

Search Overview Diagnose and solve problems Access Control (IAM) Shared access tokens Access policy Properties Metadata

Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: samplecontainername

Search blobs by prefix (case-insensitive): Show deleted blobs Add filter

**FinalData.csv**

Save Discard Download Refresh Delete

Overview Versions Snapshots Edit Generate SAS

	Name
75	139,2024,6,528,8.25
76	156,2024,6,232,5.8
77	134,2024,6,448,5.890909090909091
78	197,2024,6,480,6
79	175,2024,6,232,4.833333333333333
80	164,2024,6,200,5
81	183,2024,6,136,3.4
82	194,2024,6,456,8.142857142857142
83	138,2024,6,288,8
84	136,2024,6,336,7
85	148,2024,6,200,5
86	169,2024,6,456,6.333333333333333
87	178,2024,6,616,7.7
88	125,2024,6,248,5.1666666666666667
89	163,2024,6,144,6
90	118,2024,6,192,4
91	123,2024,6,256,4
92	141,2024,6,264,4.125
93	192,2024,6,184,7.6666666666666667
94	153,2024,6,272,5.6666666666666667
95	177,2024,6,240,7.5
96	186,2024,6,240,3.75
97	179,2024,6,384,6
98	129,2024,6,432,5.4
99	166,2024,6,336,5.25
100	150,2024,6,216,5.4
101	

Csv Preview

The screenshot shows the Snowflake web interface with the CUSTOMER\_PURCHASES table selected. The left sidebar shows the navigation menu with 'Data' and 'Databases' selected. The main panel displays the table definition:

```
1 create or replace TABLE
SNOWFLAKEDATABASE.SNOWFLAKESCHEMA.CUSTOMER_PURCHASES (
2   CUSTOMERID NUMBER(38,0),
3   YEAR NUMBER(38,0),
4   MONTH NUMBER(38,0),
5   TOTALPURCHASES NUMBER(38,0),
6   AVERAGEPURCHASEQUANTITY FLOAT
7 );
```

Below the table definition, there is a 'Privileges' section showing 'ACCOUNTADMIN' with 'Current Role' and 'Ownership'.

The screenshot shows the Snowflake web interface with the CUSTOMER\_PURCHASES table selected. The left sidebar shows the navigation menu with 'Data' and 'Databases' selected. The main panel displays the table columns:

NAME	TYPE	NULLABLE	DEFAULT
AVERAGEPURCHASEQUANTITY	# Float	Yes	NULL
CUSTOMERID	# Number	Yes	NULL
MONTH	# Number	Yes	NULL
TOTALPURCHASES	# Number	Yes	NULL
YEAR	# Number	Yes	NULL

Snowflake Data Preview for CUSTOMER\_PURCHASES Table

Table Details: ACCOUNTADMIN, 3 hours ago, 200 rows, 3.0KB

Data Preview: COMPUTE\_WH (100 of 200 Rows • Updated just now)

	CUSTOMERID	YEAR	MONTH	TOTALPURCHASES	AVERAGEPURCHASEQUANTITY
85	148	2024	6	200	5
86	169	2024	6	456	6.333333333
87	178	2024	6	616	7.7
88	125	2024	6	248	5.166666667
89	163	2024	6	144	6
90	110	2024	6	192	4
91	123	2024	6	256	4
92	141	2024	6	264	4.125
93	192	2024	6	184	7.666666667
94	153	2024	6	272	5.666666667
95	177	2024	6	240	7.5
96	186	2024	6	240	3.75
97	179	2024	6	384	6
98	129	2024	6	432	5.4
99	166	2024	6	336	5.25
100	150	2024	6	216	5.4

Snowflake Copy History for CUSTOMER\_PURCHASES Table

Table Details: ACCOUNTADMIN, 3 hours ago, 200 rows, 3.0KB

Copy History: Copies over time (Last 7 days)

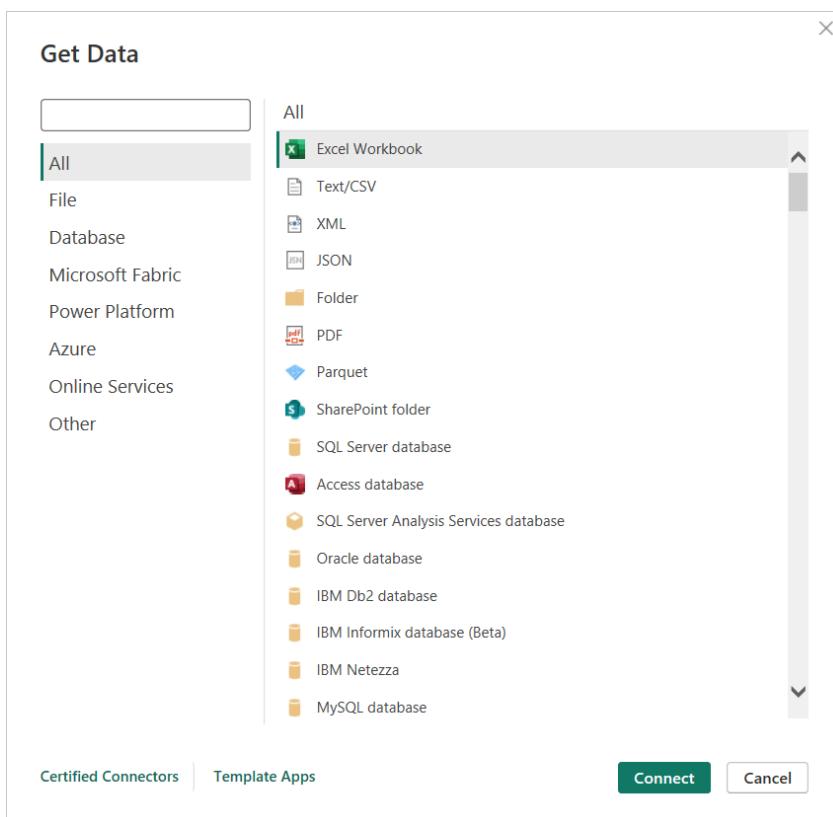
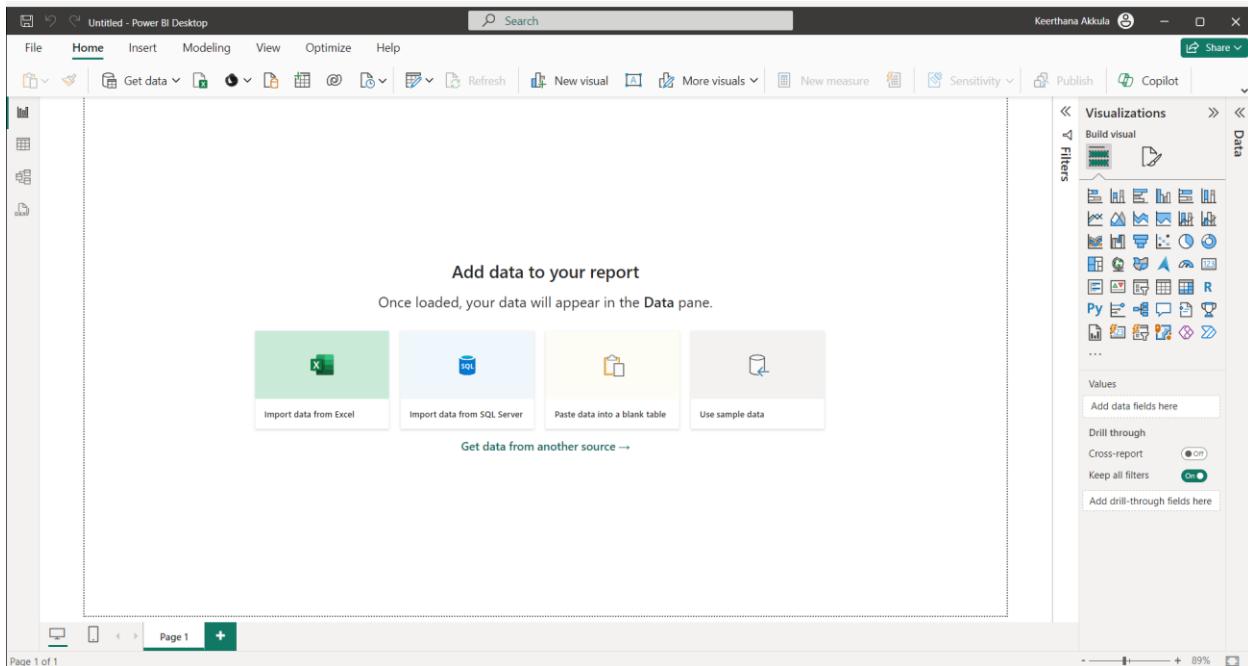
2 Copies (Jun 13, 2024, 12 AM - Jun 20, 2024, 12 AM)

FILE NAME	LOADED	SIZE	ROWS	STATUS	SOURCE	PIPE
	6m ago	2.3KB	100	Loaded	—	—
	21m ago	2.3KB	100	Loaded	—	—

## Snowflake to Power BI Desktop

Server name of snowflake Accountname.region.snowflakecomputing.com

ai16587.canada-central.azure.snowflakecomputing.com



Microsoft Wallet

10.100 US\$0.00

Passwords / snowflakecomputing.com

secreddy14

secreddy14

Password:

Site: <https://ai16587.canada-central.azure.snowflakecomputing.com/>

Notes: No note added

Edit Delete

secreddy14

Password:

Site: <https://cuoipnx-qg65241.snowflakecomputing.com/c...>

Notes: No note added

Edit Delete

+10 10

Level 1

Satisfied with Wallet?

Sign in to Snowflake

Username: secreddy14

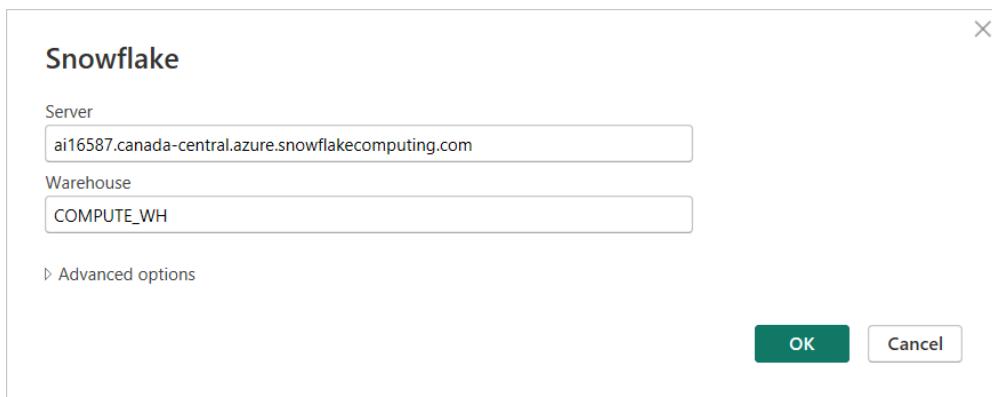
Password:

[Forgot password?](#)

We process your personal information according to our  
[Privacy Notice](#)

The screenshot shows the Snowflake web interface with the URL <https://app.snowflake.com/cuciprx/qg65241/#/compute/warehouses?whType=all&status=all&columns=name%2Cstatus%2Csize%2Ctype%2Cclusters%2Crunning%2Cqueued%2Cowner%2Cresumed>. The left sidebar is open, showing the 'Warehouses' section under 'Admin'. The main content area displays a table titled 'Warehouses' with one row:

NAME	SIZE	STATUS	RUNNING	QUEUED	OWNER	RESUMED
COMPUTE_WH	M	Started	2	0	ACCOUNTADMIN	19 minutes ago



The screenshot shows the Power BI Desktop interface with a 'Untitled - Power BI Desktop' tab. A 'Get data' dialog box is open, specifically for 'Snowflake'. It shows the server 'ai16587.canada-central.azure.snowflakecomputing.com' and a 'Microsoft Account' selected. The dialog has 'User name' set to 'secreddy14' and 'Password' masked. At the bottom are 'Back', 'Connect', and 'Cancel' buttons. To the right of the dialog, the Power BI canvas shows a visualization with a timer showing '00:21:38' and various data exploration tools like 'Sensitivity' and 'Drill through'.

## Navigator

Display Options ▾

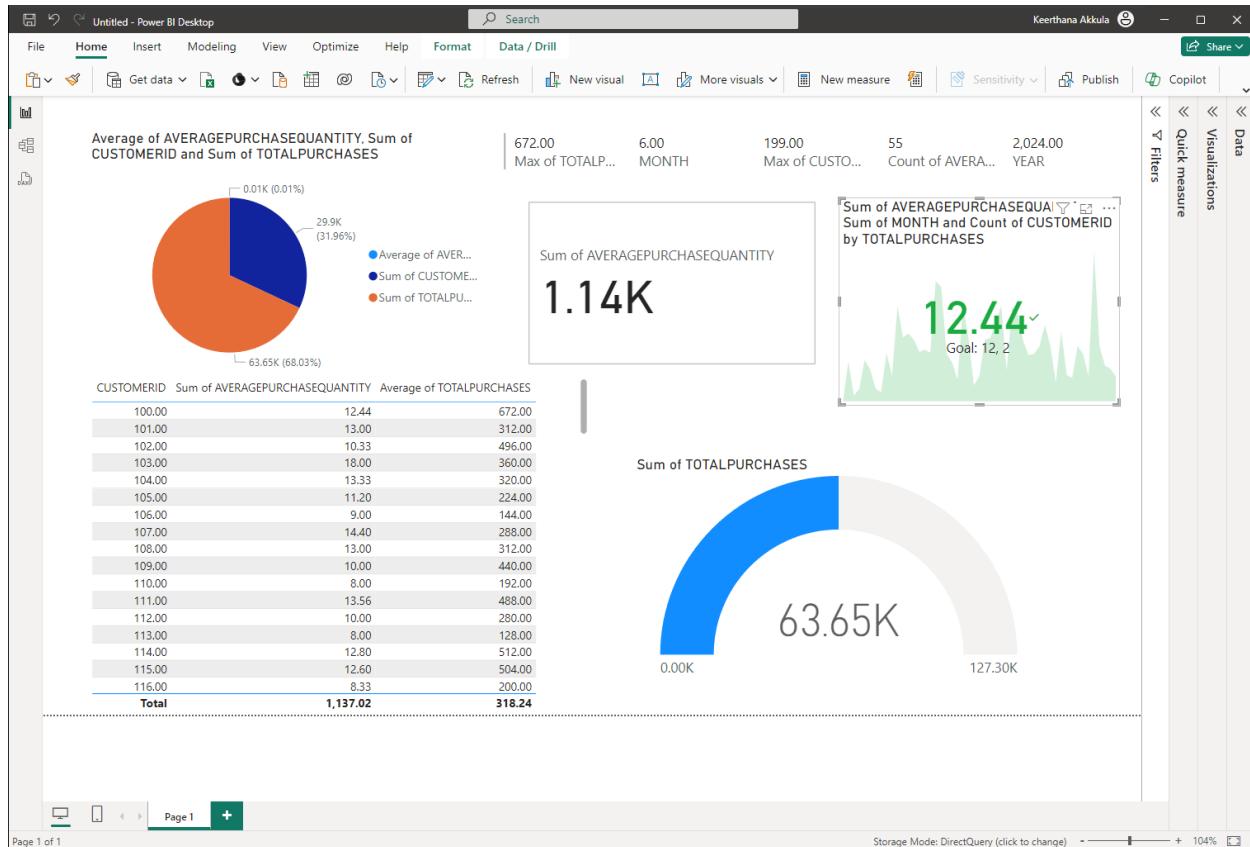
- ai16587.canada-central.azure.snowflakecompute...
- ▶ SNOWFLAKE
- ▶ SNOWFLAKE\_SAMPLE\_DATA
- ▶ SNOWFLAKEDATABASE [2]
- ▶ PUBLIC
- ▶ SNOWFLAKESCHEMA [1]
- CUSTOMER\_PURCHASES

### CUSTOMER\_PURCHASES

CUSTOMERID	YEAR	MONTH	TOTALPURCHASES	AVERAGEPURCHASEQUANT...
102	2024	6	496	5.1666666
146	2024	6	248	5.1666666
180	2024	6	280	4.3
111	2024	6	488	6.7777777
154	2024	6	112	4.6666666
195	2024	6	424	5.8888888
122	2024	6	184	5.
131	2024	6	496	5.1666666
124	2024	6	120	
158	2024	6	384	6.8571428
190	2024	6	296	4.6
112	2024	6	280	
107	2024	6	288	
133	2024	6	72	
105	2024	6	224	
121	2024	6	184	5.
173	2024	6	248	
167	2024	6	320	6.6666666
119	2024	6	256	
101	2024	6	312	
155	2024	6	416	
162	2024	6	192	3.4285714
170	2024	6	208	

Select Related Tables Transform Data

## Used Pie chart, Tabular, Donut chart, Line chart



## Data Format Exploration:

- Experiment with different data formats (CSV vs. Parquet) and compare processing efficiency.**
- Parquet offers better compression and faster query performance for analytical workloads.**

The screenshot shows a Databricks notebook interface with the following details:

- Code Cell:** Contains Python code for mounting ADLS in Databricks.

```

1 # Mount ADLS in Databricks
2 configs = {
3     "fs.azure.account.auth.type": "OAuth",
4     "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
5     "fs.azure.account.oauth2.client.id": "22b45241-70c5-4ea2-a23a-fb7efc40d683",
6     "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="rakshithaSecretScope", key="secretforServicePrincipal"),
7     "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/cb876379-1061-49ab-964b-7e5d840d62a1/oauth2/token"
}

```

- Output Tab:** Shows an error message: "ExecutionError: An error occurred while calling s396.mount. : java.rmi.RemoteException: java.lang.IllegalArgumentException: requirement failed: Directory already mounted: /mnt/adls; nested exception is: ...".

Microsoft Azure | databricks | Search data, notebooks, recents, a... CTRL + P | databricksworkspace | H

### CSV vs. Parq...

Python | Run all | Schedule | Share | HEM MARYADA's Cluster

```
mount_point = "/mnt/adls"
# Check if the mount point already exists
mounts = dbutils.fs.mounts()
mount_points = [mount.mountPoint for mount in mounts]
if mount_point not in mount_points:
    dbutils.fs.mount(
        source = "abfss://dataformatcontainer@adlsrakshitha.dfs.core.windows.net/",
        mount_point = mount_point,
        extra_configs = configs
)
else:
    print(f"Mount point {mount_point} already exists.")
```

Output Terminal

Mount point /mnt/adls already exists.

Microsoft Azure | databricks | Search data, notebooks, recents, a... CTRL + P | databricksworkspace | H

### CSV vs. Parq...

Python | Run all | Schedule | Share | HEM MARYADA's Cluster

```
# Define the path for Parquet file
parquet_file_path = "/mnt/adls/sample_data.parquet"
# Write the DataFrame as Parquet with overwrite mode
df_csv.write.mode("overwrite").parquet(parquet_file_path)
```

Microsoft Azure | databricks

CSV vs. Parq... Python Search data, notebooks, recents, a... CTRL + P databricksworkspace Share

File Edit View Run Help L Pr Run all HEM MARYADA's Cluster Schedule Share

08:05 PM (12s) Load Data in Databricks Python

```
1 # Load the CSV file into a DataFrame
2 csv_file_path = "/mnt/adls/salesdata.csv"
3 df_csv = spark.read.csv(csv_file_path, header=True, inferSchema=True)
4
5 # Display the DataFrame
6 df_csv.show()
```

Output Terminal

```
df_csv: pyspark.sql.dataframe.DataFrame = [Country: string, Company: string ... 4 more fields]
+-----+-----+-----+-----+-----+
|Country|Company|Month|Sale2018|Sale2019|Sale2020|
+-----+-----+-----+-----+-----+
| India|SequelGate| January| 3,234.00| NULL| 1,935.00|
| India|SequelGate| February| 6,270.00| 7,059.00| 12,418.26|
| India|SequelGate| March| 4,352.00| 16,638.85| 31,770.26|
| India|SequelGate| April| 3,814.00| 11,864.76| 8,785.89|
| India|SequelGate| May| 6,234.00| 2,819.82| 119.99|
| India|SequelGate| June| 5,571.00| 3,216.00| 2,819.82|
```

InPrivate [2] https://adb-953747807129243.azuredatabricks.net/?o=953747807129243#notebook/2129975719267786/command/1117397613705365

Microsoft Azure | databricks

CSV vs. Parq... Python Search data, notebooks, recents, a... CTRL + P databricksworkspace Share

File Edit View Run Help L Pr Run all HEM MARYADA's Cluster Schedule Share

08:06 PM (2s) Convert CSV to Parquet Python

```
1 # Define the path for Parquet file
2 parquet_file_path = "/mnt/adls/sample_data.parquet"
3
4 # Write the DataFrame as Parquet with overwrite mode
5 df_csv.write.mode("overwrite").parquet(parquet_file_path)
```

Output Terminal

Microsoft Azure |  databricks | Search data, notebooks, recents, a... CTRL + P | databricksworkspace | ? H

### CSV vs. Parq...

Python | Run all | HEM MARYADA's Cluster | Schedule | Share

```
1 #CSV Processing
2
3 import time
4
5 # Measure the time taken to read the CSV file
6 start_time = time.time()
```

**Output** Terminal

```
df_csv: pyspark.sql.dataframe.DataFrame = [Country: string, Company: string ... 4 more fields]
CSV Read Time: 3.044142484664917 seconds
```

InPrivate (2) | CSV vs. | CSV vs. | mywkr | SqlDat | LakeH | Catalog | Intro | Datab | mydat | ADF\_n | All res | + | Microsoft Edge

Microsoft Azure |  databricks | Search data, notebooks, recents, a... CTRL + P | databricksworkspace | ? H

### CSV vs. Parq...

Python | Run all | Cell 6 | Python | Schedule | Share

```
1 #Parquet Processing
2
3 # Measure the time taken to read the Parquet file
4 start_time = time.time()
5 df_parquet = spark.read.parquet(parquet_file_path)
6 df_parquet.count() # Trigger an action to ensure the DataFrame is fully loaded
7 end_time = time.time()
8
9 print("Parquet Read Time: {} seconds".format(end_time - start_time))
10
```

**Output** Terminal

```
df_parquet: pyspark.sql.dataframe.DataFrame = [Country: string, Company: string ... 4 more fields]
Parquet Read Time: 3.071917772293091 seconds
```

**The reason why the reading a CSV file takes less time than parquet file is because of the size of the file we are reading. Make sure we have the same file size to see the time difference between both of comparisons.**

**Overview of project** – Optimized the inventory management and improved the sales forecasting accuracy. I have achieved this by analyzing data from sales transactions, inventory logs, and customer information. There are 4 phases in my project

**Question 1: Why synapse to relational data base (Azure SQL DB)?** A retail company wants to analyze customer behavior to improve its marketing strategies. They have a data warehouse that stores massive amounts of historical sales, customer interactions, and product data. However, for more focused and faster analysis, they need to transfer specific subsets of this data to a relational database and then perform ETL operations on this data.

### **Simplified form**

They have lots of historical data stored in a data warehouse. To analyze this data more quickly and easily we will transfer specific parts of it to a relational database and then perform ETL.

Aggregated data in warehouse (history data and aggregated data)

### **Question 2: Delta to snowflake?**

1. Set up a Delta Lake to Snowflake integration using manifest files and query Delta tables.
2. Implement delta sharing in Databricks to Snowflake direct connect
3. Databricks provides a Snowflake connector in the Databricks Runtime to support writing data from Snowflake.
4. Pipeline should also work (delta table to snowflake)

<https://community.databricks.com/t5/data-engineering/how-do-we-use-delta-sharing-between-databricks-to-snowflake/td-p/55415#:%text=To%20implement%20delta%20sharing%20in%20Databricks%20to%20Snowflake,users%20who%20have%20access%20to%20delta%20sharing.%20>

<https://docs.databricks.com/en/data-sharing/index.html>

<https://docs.databricks.com/en/connect/external-systems/snowflake.html>

**Direct delta to snowflake is not recommended because it is not a real time scenario – so whatever I used blob to snowflake is a real time case.**

### **Question 3: How can you get the data stored in silver or bronze layer ?**

Table name and the connection where you want to write the data

Mounting in ADLS (Azure Data Lake Storage) and Databricks means connecting a storage account to Databricks so you can access the data stored in ADLS directly from Databricks as if it were part of your local file system. This simplifies data access and management within your Databricks environment.

#### **In simpler form**

Mounting in ADLS and Databricks means linking your data storage to Databricks so you can easily access and use your data stored in Azure within Databricks, just like you would with local files on your computer.

#### **And why do you use access key, sas key, iam ?**

Access keys (or SAS keys) are used in Azure services like Azure Data Lake Storage (ADLS) and Databricks for authentication and authorization purposes. Here's a simpler explanation:

Access keys or Shared Access Signature (SAS) keys are used to securely connect services like Azure Data Lake Storage (ADLS) to Databricks. They act like passwords, allowing Databricks to access and manage data stored in ADLS, ensuring secure and controlled access to your cloud storage resources.