Customer Churn Azure Project

Note: The dataset provided for this assignment was already clean, with no missing values or Personally Identifiable Information (PII). However, for demonstration purposes, a filter condition was implemented to show how missing values can be handled using an Azure Data Factory (ADF) pipeline.

Project Objective

The goal of this project is to migrate data from an API source to an Azure SQL Database and establish an automated orchestration workflow using Azure Data Factory.

Technology Stack

- Resource Group
- Storage Accounts
- Azure Data Factory (ADF)
- Azure Databricks
- Azure SQL Database
- Azure Key Vault

ADF Activities Used

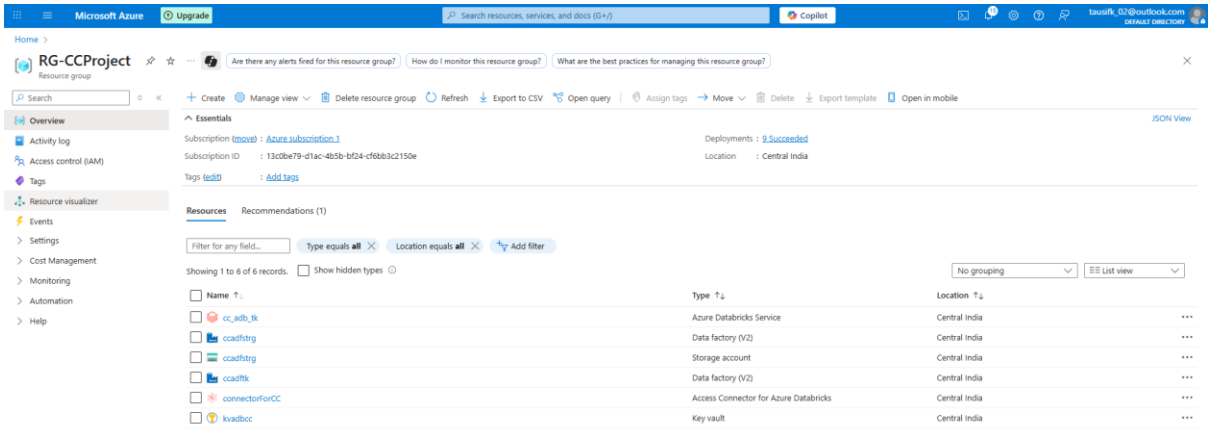
- Lookup Activity
- Copy Data Activity
- Azure Databricks Notebook
- Data Flow

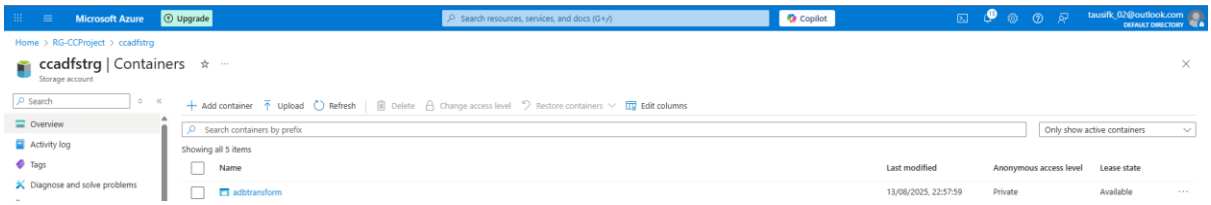
Project Workflow

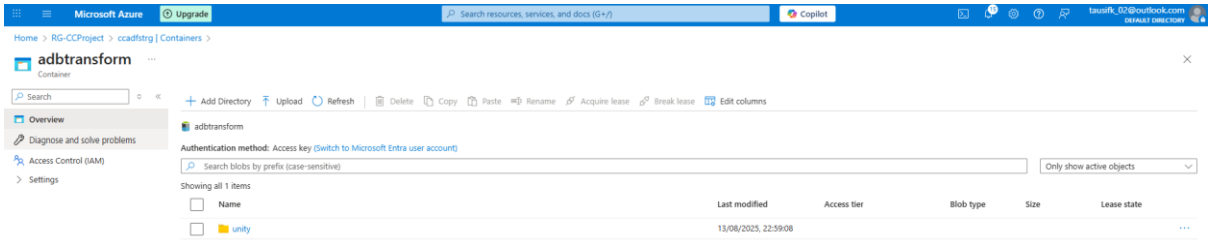
Azure Resources Setup

1. Created a Resource Group in Azure.
2. Added a Storage Account and Azure Data Factory instance under this resource group.
3. Within the Storage Account, created one container **'adbtransform'** and **'unity'** folder under the container.

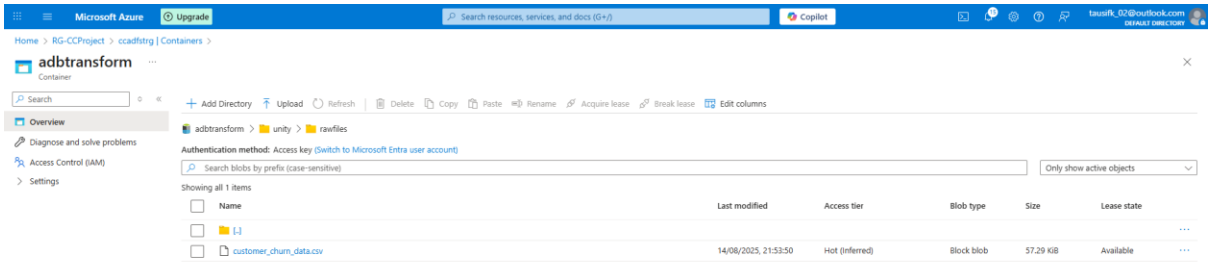
The following screenshot shows all the resources created under the Resource Group.



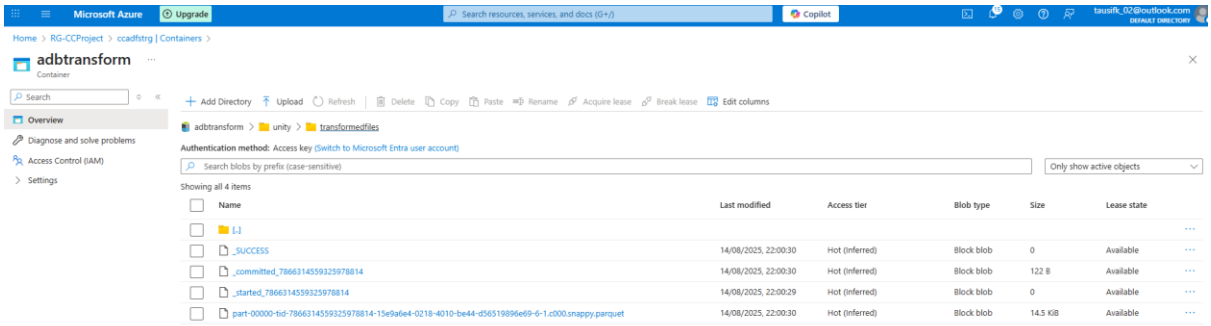




CSV file is loaded using lookup activity dynamically using ADF pipeline

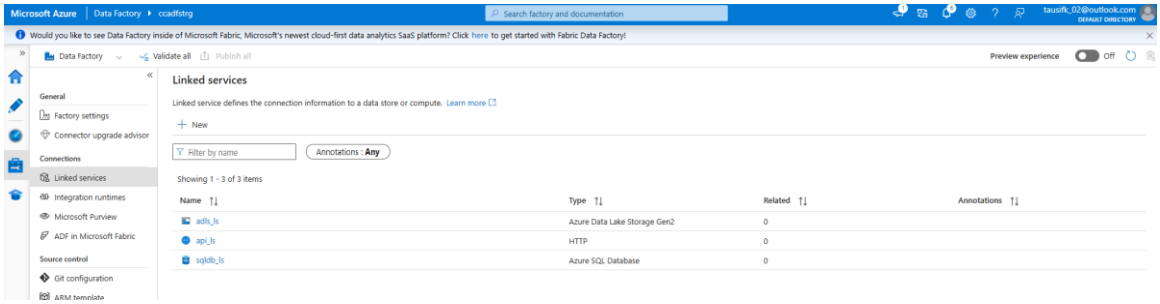


Parquet file is created using Azure Databricks notebook.



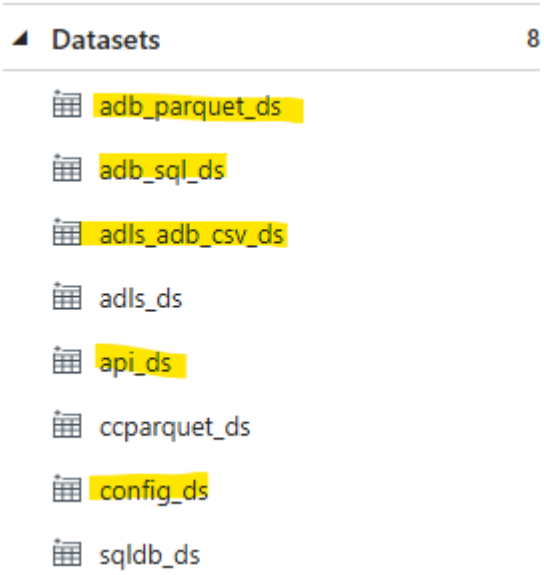
Linked Services Configuration

Configured Linked Services in ADF for connections to:
- Azure Storage Account
- Github Account
- Azure SQL Database



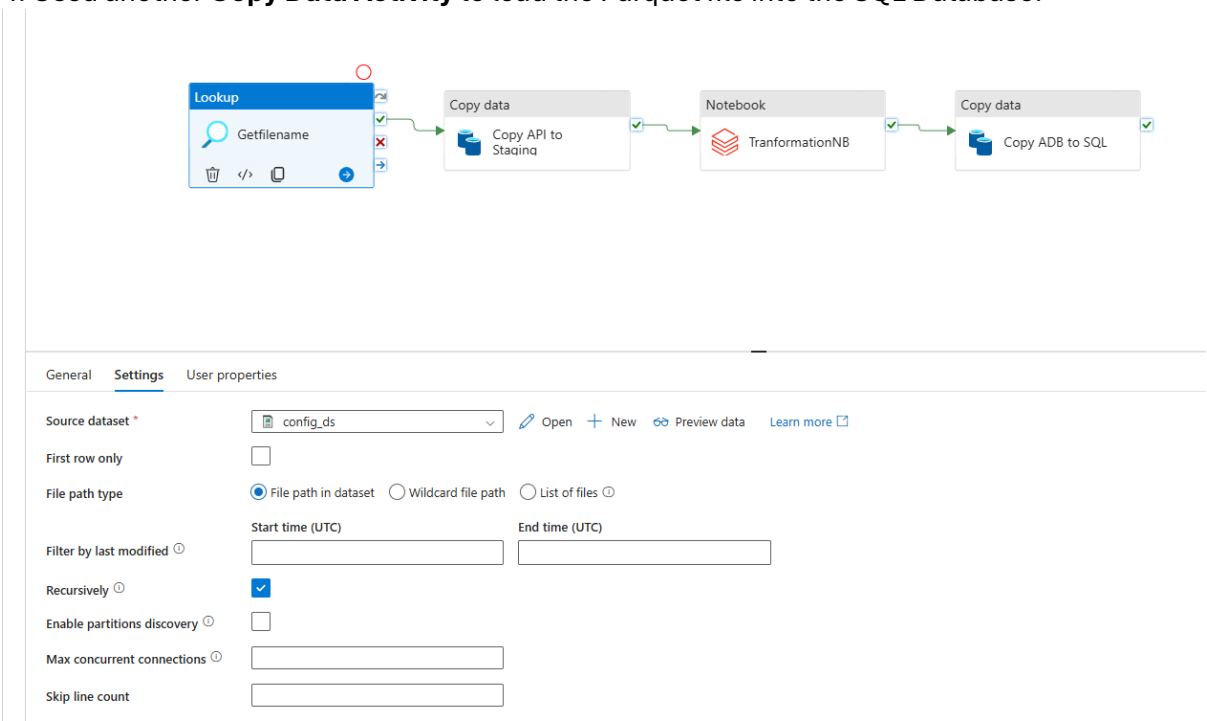
Dataset Creation

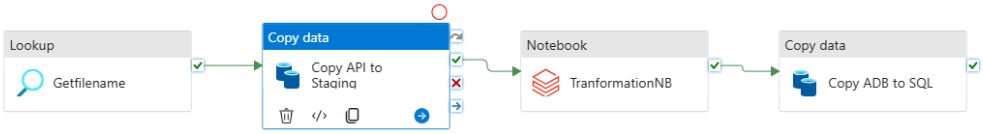
- The config file in the Storage Account – config_ds
- Loading file from GitHub – api_ds
- Create CSV file in ADLS Gen 2 – adls_adb_csv_ds
- Creating Parquet file using Notebook – adb_parquet_ds
- Create table in SQL Database – adb_Sql_ds



ETL Pipeline Development

Pipeline Workflow:
1. The config file containing the file name is passed through a Lookup Activity to the Copy Data Activity.
2. In **Azure Databricks notebook**, data is filtered/transformed by applying filter condition as “InternetService” is not equal to "None".
3. Saved the filtered data as a Parquet file in the transformedfiles folder.
4. Used another **Copy Data Activity** to load the Parquet file into the SQL Database.





General | **Source** | Sink | Mapping | Settings | User properties

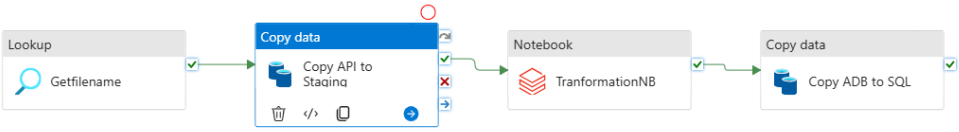
Source dataset * api_ds Open + New Preview data Learn more

Dataset properties

Name	Value
load	@activity('Getfilename').output.value...

Request method * GET

Additional headers



General | Source | **Sink** | Mapping | Settings | User properties

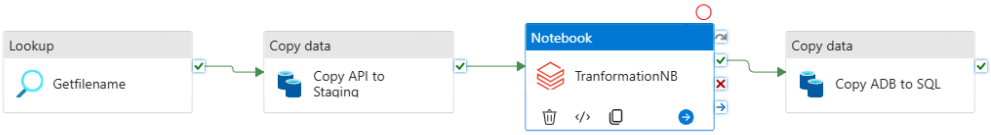
Sink dataset * adls_adb_csv_ds Open + New Learn more

Dataset properties

Name	Value
filename	@activity('Getfilename').output.value...

Copy behavior Select...

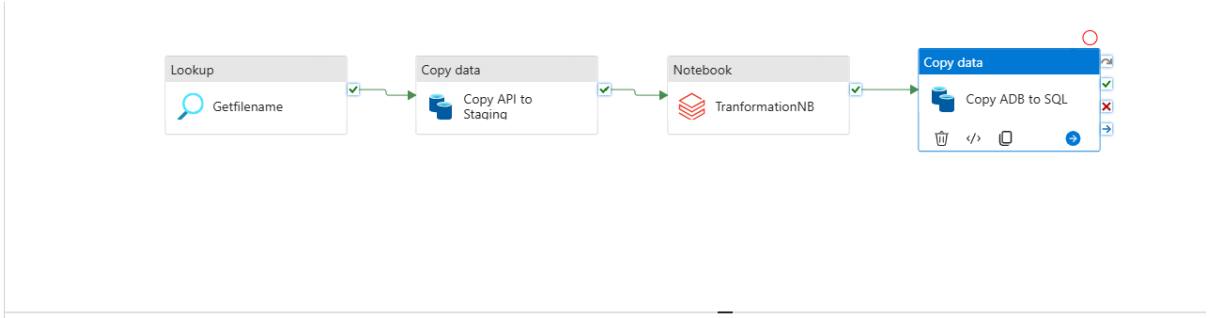
Max concurrent connections

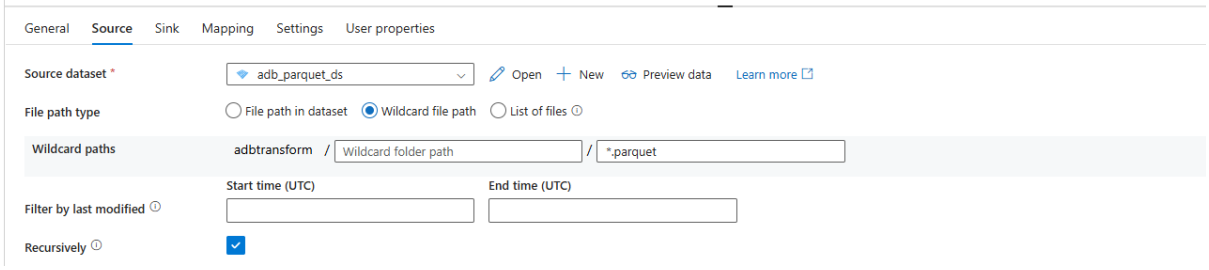


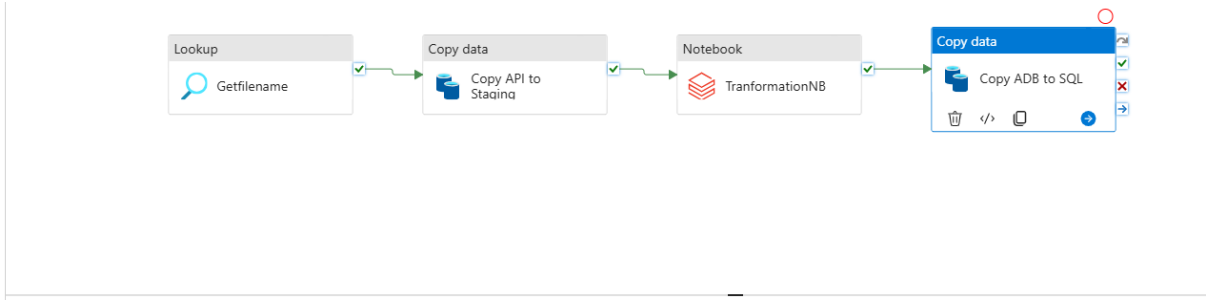
Expand toolbox pane

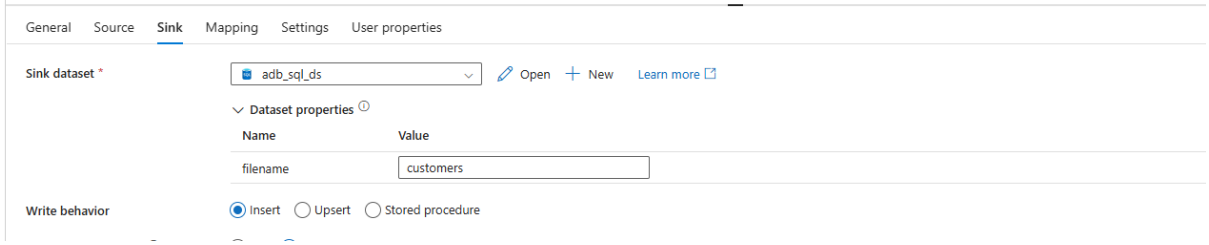
General | **Azure Databricks** | Settings | User properties

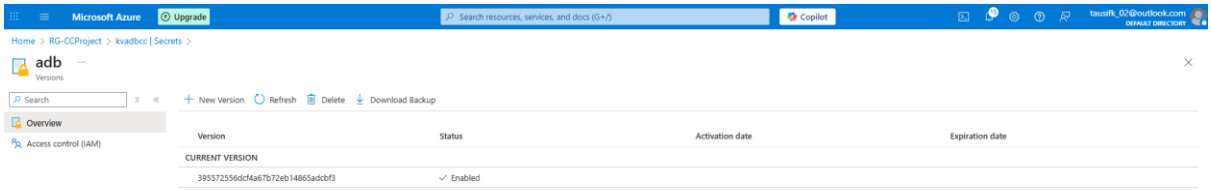
Databricks linked service * AzureDatabricks1 Test connection Edit + New

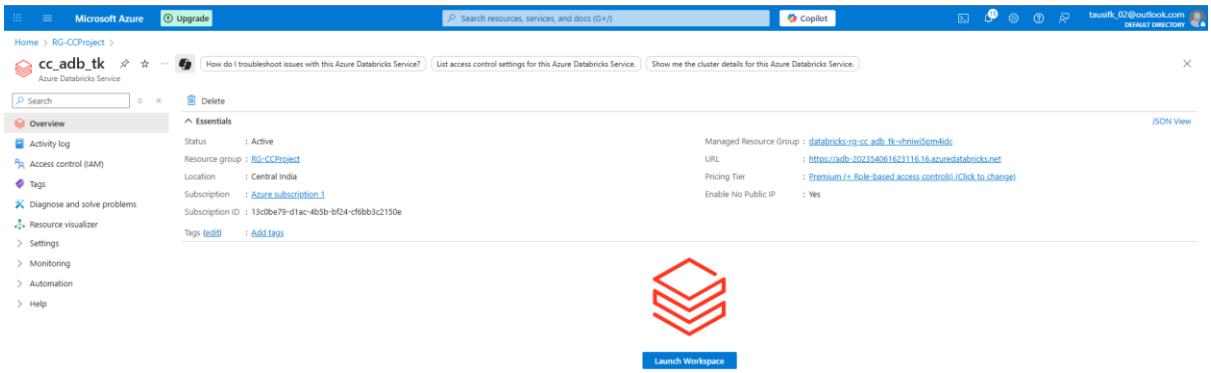

Lookup
Getfilename
Copy data
Copy API to Staging
Notebook
TranformationNB
Copy data
Copy ADB to SQL


General
Source
Sink
Mapping
Settings
User properties
Source dataset *
adb_parquet_ds
Open
New
Preview data
Learn more
File path type
File path in dataset
Wildcard file path
List of files
Wildcard paths
adbtransform
Wildcard folder path
*.parquet
Start time (UTC)
End time (UTC)
Filter by last modified
Recursively


Lookup
Getfilename
Copy data
Copy API to Staging
Notebook
TranformationNB
Copy data
Copy ADB to SQL


General
Source
Sink
Mapping
Settings
User properties
Sink dataset *
adb_sql_ds
Open
New
Learn more
Dataset properties
Name
Value
filename
customers
Write behavior
Insert
Upsert
Stored procedure


Microsoft Azure
Upgrade
Copilot
Home > RG-CCProject > kvadbcc | Secrets >
adb
Versions
Overview
Access control (IAM)
New Version
Refresh
Delete
Download Backup
Version
Status
Activation date
Expiration date
CURRENT VERSION
395572556dcf4a67b72eb14865adcbf3
Enabled


Microsoft Azure
Upgrade
Copilot
Home > RG-CCProject >
cc_adb_tk
Azure Databricks Service
Delete
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Monitoring
Automation
Help
Essentials
JSON View
Status : Active
Resource group : RG-CCProject
Location : Central India
Subscription : Azure subscription 1
Subscription ID : 13c0be79-d1ac-4b5b-bf24-cf6bb3c2150e
Tags (edit) : Add tags
Managed Resource Group : databricks-rg-cc_adb_tk-vhniwi5pm4idc
URL : https://adb-2023540616231 16.azuredatabricks.net
Pricing Tier : Premium (+ Role-based access controls) (Click to change)
Enable No Public IP : Yes
Launch Workspace

08:13 PM (1m) 1 Python

```python
df = spark.read.format("csv")\
    .option("header", "true")\
    .option("inferSchema", "true")\
    .load("abfss://adbtransform@ccadfstrg.dfs.core.windows.net/unity/rawfiles/customer_churn_data.csv")

display(df)
```

See performance (1) Optimize

df: pyspark.sql.connect.dataframe.DataFrame = [CustomerID: integer, Age: integer ... 8 more fields]

Table

	CustomerID	Age	Gender	Tenure	MonthlyCharges	ContractType	InternetService	TotalCharges	TechSupport
1	1	49	Male	4	88.35	Month-to-Month	Fiber Optic	353.4	Yes
2	2	43	Male	0	36.67	Month-to-Month	Fiber Optic	0	Yes
3	3	51	Female	2	63.79	Month-to-Month	Fiber Optic	127.58	No
4	4	60	Female	8	102.34	One-Year	DSL	818.72	Yes
5	5	42	Male	32	69.01	Month-to-Month	None	2208.32	No
6	6	42	Female	16	119.75	Two-Year	DSL	1916	Yes
7	7	60	Male	14	80.32	One-Year	None	1124.48	No
8	8	52	Female	6	58.9	One-Year	None	353.4	No
9	9	40	Female	53	49.81	Two-Year	Fiber Optic	2639.9300000000003	Yes
10	10	50	Female	10	61.55	Month-to-Month	Fiber Optic	615.5	Yes
11	11	40	Female	1	63.53	Month-to-Month	Fiber Optic	63.53	Yes
12	12	40	Female	19	36.18	One-Year	Fiber Optic	687.42	Yes
13	13	47	Male	2	63.26	Two-Year	None	126.52	No
14	14	25	Female	8	71.78	One-Year	None	574.24	No
15									

1,000 rows | 1m 11s runtime Refreshed 2 hours ago

08:16 PM (3s) 2: er Python

```python
from pyspark.sql.functions import *
from pyspark.sql.types import *

df_filter = df.filter(col("InternetService") != "None")

df_filter.display()
```

See performance (1) Optimize

df_filter: pyspark.sql.connect.dataframe.DataFrame = [CustomerID: integer, Age: integer ... 8 more fields]

Table

	CustomerID	Age	Gender	Tenure	MonthlyCharges	ContractType	InternetService	TotalCharges	TechSupport
1	1	49	Male	4	88.35	Month-to-Month	Fiber Optic	353.4	Yes
2	2	43	Male	0	36.67	Month-to-Month	Fiber Optic	0	Yes
3	3	51	Female	2	63.79	Month-to-Month	Fiber Optic	127.58	No
4	4	60	Female	8	102.34	One-Year	DSL	818.72	Yes
5	6	42	Female	16	119.75	Two-Year	DSL	1916	Yes
6	9	40	Female	53	49.81	Two-Year	Fiber Optic	2639.9300000000003	Yes
7	10	50	Female	10	61.55	Month-to-Month	Fiber Optic	615.5	Yes
8	11	40	Female	1	63.53	Month-to-Month	Fiber Optic	63.53	Yes
9	12	40	Female	19	36.18	One-Year	Fiber Optic	687.42	Yes
10	15	27	Male	14	95.05	Month-to-Month	Fiber Optic	1330.7	Yes
11	16	39	Female	41	89.11	Two-Year	Fiber Optic	3653.5099999999998	Yes
12	19	35	Male	98	49.59	Two-Year	Fiber Optic	4859.820000000001	Yes
13	20	30	Female	2	89.55	Month-to-Month	Fiber Optic	179.1	No
14	21	59	Female	6	73.56	One-Year	DSL	441.36	No
15									

703 rows | 2.73s runtime Refreshed 2 hours ago

08:16 PM (5s) 5 Python

```python
df_filter.write.mode("overwrite").format("parquet").save("abfss://adbtransform@ccadfstrg.dfs.core.windows.net/unity/transformedfiles")
```

See performance (1) Optimize

03:09 PM (1s) 6

```python
df_filter.createOrReplaceTempView("customers")
```

See performance (1) Optimize

03:10 PM (5s) 7

```sql
%sql

CREATE TABLE customerchurn.default.customers
AS
select * from customers
```

See performance (1) Optimize

Table

num_affected_rows	num_inserted_rows

03:11 PM (7s) 8 SQL

```sql
%sql
select * from customerchurn.default.customers
```

See performance (1) Optimize

_sqldf: pyspark.sql.connect.dataframe.DataFrame = [CustomerID: integer, Age: integer ... 8 more fields]

Table

	CustomerID	Age	Gender	Tenure	MonthlyCharges	ContractType	InternetService	TotalCharges	TechSupport
1	1	49	Male	4	88.35	Month-to-Month	Fiber Optic	353.4	Yes
2	2	43	Male	0	36.67	Month-to-Month	Fiber Optic	0	Yes
3	3	51	Female	2	63.79	Month-to-Month	Fiber Optic	127.58	No
4	4	60	Female	8	102.34	One-Year	DSL	818.72	Yes
5	6	42	Female	16	119.75	Two-Year	DSL	1916	Yes
6	9	40	Female	53	49.81	Two-Year	Fiber Optic	2639.9300000000003	Yes
7	10	50	Female	10	61.55	Month-to-Month	Fiber Optic	615.5	Yes
8	11	40	Female	1	63.53	Month-to-Month	Fiber Optic	63.53	Yes
9	12	40	Female	19	36.18	One-Year	Fiber Optic	687.42	Yes
10	15	27	Male	14	95.05	Month-to-Month	Fiber Optic	1330.7	Yes
11	16	39	Female	41	89.11	Two-Year	Fiber Optic	3653.5099999999998	Yes
12	19	35	Male	98	49.59	Two-Year	Fiber Optic	4859.820000000001	Yes
13	20	30	Female	2	89.55	Month-to-Month	Fiber Optic	179.1	No
14	21	59	Female	6	73.56	One-Year	DSL	441.36	No
15									

703 rows | 7.34s runtime Refreshed 7 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

Microsoft Azure Upgrade Search resources, services, and docs (G+/) Copilot tausifk_02@outlook.com DEFAULT DIRECTORY

Home > tk-azure-sql-server > CCpresentationlayer (tk-azure-sql-server/CCpresentationlayer)

CCpresentationlayer (tk-azure-sql-server/CCpresentationlayer) | Query editor (preview)
SQL database

Search
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Data management
Integrations
Power Platform
Security
Intelligent performance
Monitoring
Automation
Help

Login New Query Open query Feedback Getting started

CCpresentationlayer (tausifk_02@outlo...

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

Tables
dbo.Customer_churn
dbo.customers
Views
Stored Procedures

Query 1 Query 2

Run Cancel query Save query Export data as Show only Editor

```sql
SELECT TOP (1000) * FROM [dbo].[customers]
```

Results Messages

Search to filter items...

CustomerID	Age	Gender	Tenure	MonthlyCharges	ContractType	InternetServ
1	49	Male	4	88.35	Month-to-Month	Fiber Optic
2	43	Male	0	36.67	Month-to-Month	Fiber Optic
3	51	Female	2	63.79	Month-to-Month	Fiber Optic
4	60	Female	8	102.34	One-Year	DSL
6	42	Female	16	119.75	Two-Year	DSL
9	40	Female	53	49.81	Two-Year	Fiber Optic
10	50	Female	10	61.55	Month-to-Month	Fiber Optic

Orchestration and Automation

- Set a trigger so the pipeline runs automatically at scheduled times. *As per assignment it is to be scheduled for every hour.*
- Screenshots of the setup are included for reference.