**Use Databricks to Perform Extract, Transform, Load (ETL) Processes on Large Datasets.**

**Introduction:**

In this documentation we will explore the scenario of performing Extract, Transform, Load (ETL) using Databricks.

ETL is a process used to extract data from various sources, transform it into a desired format, and load it into a target data store.

Our workflow involves moving data from Azure SQL DB to Azure Databricks and ultimately to ADLS Gen2**.**

**What is Databricks?**

Databricks is an advanced analytics platform built on Apache Spark, designed for big data processing and analytics.

It offers a collaborative environment that enables data scientists, analysts, and engineers to work together on large-scale data projects.

Databricks supports multiple programming languages, including Python, Scala, and SQL, providing flexibility for different use cases.

**ETL Overview**

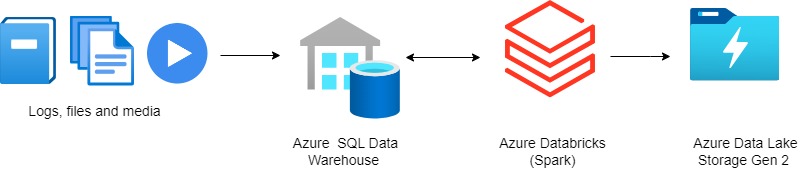
ETL stands for Extract, Transform, Load.

Extraction involves retrieving data from various sources, such as databases or APIs.

Transformation includes cleaning, aggregating, and restructuring the data to meet specific requirements.

Loading is the process of storing the transformed data into a target data store, ensuring it is ready for analysis or reporting.

**Overview Diagram:**

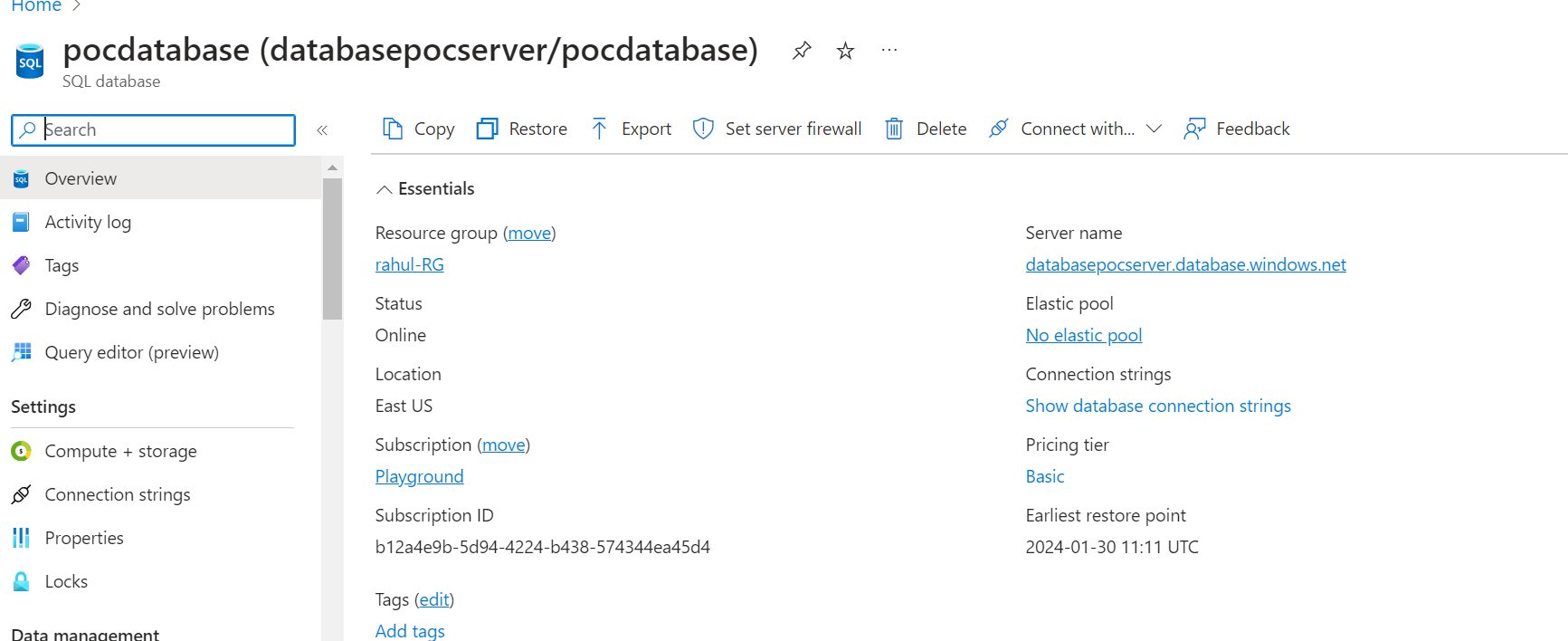


Our workflow begins with data in Azure SQL DB, then moves to Azure Databricks for processing, and finally lands in ADLS Gen2 for storage and accessibility.

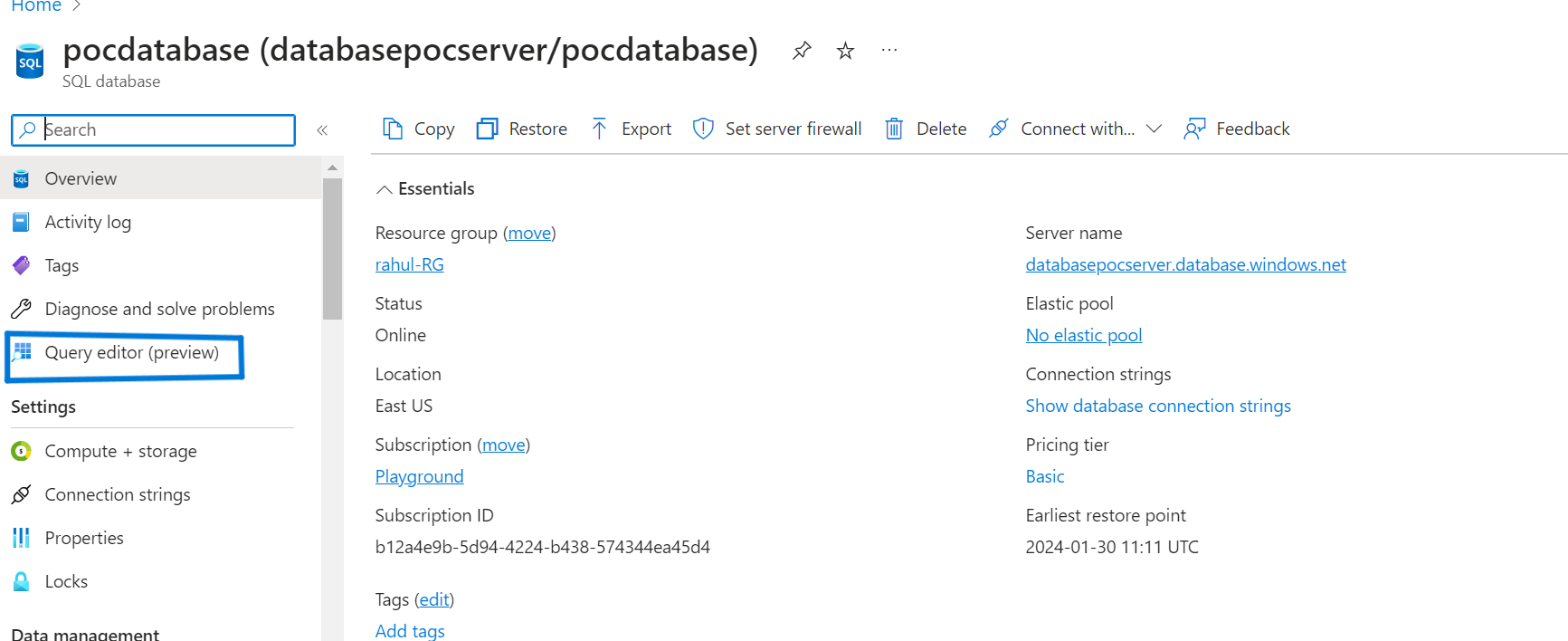
**Pre-Requisites**

For this Demo, ensure the successful creation of the following pre-requisites**.**

**i)** Create Azure SQL Database:

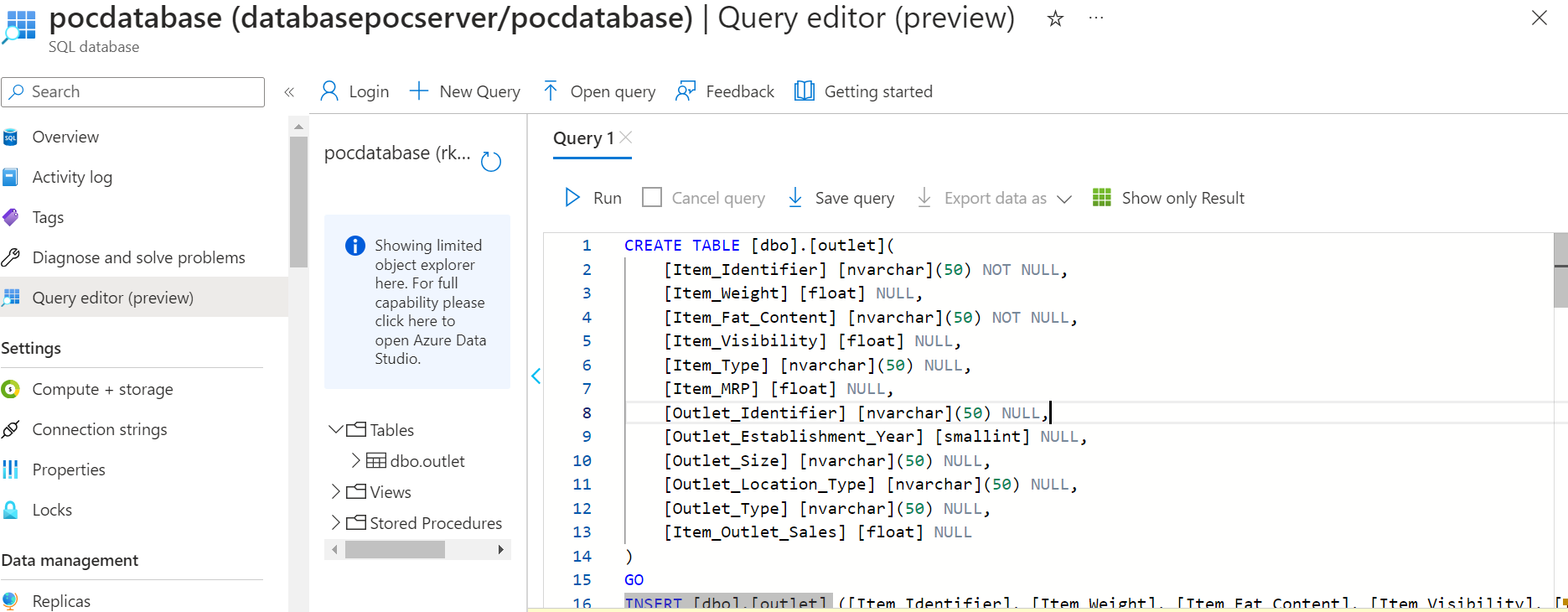


ii) Once we have created a database, we need to create a table inside it.

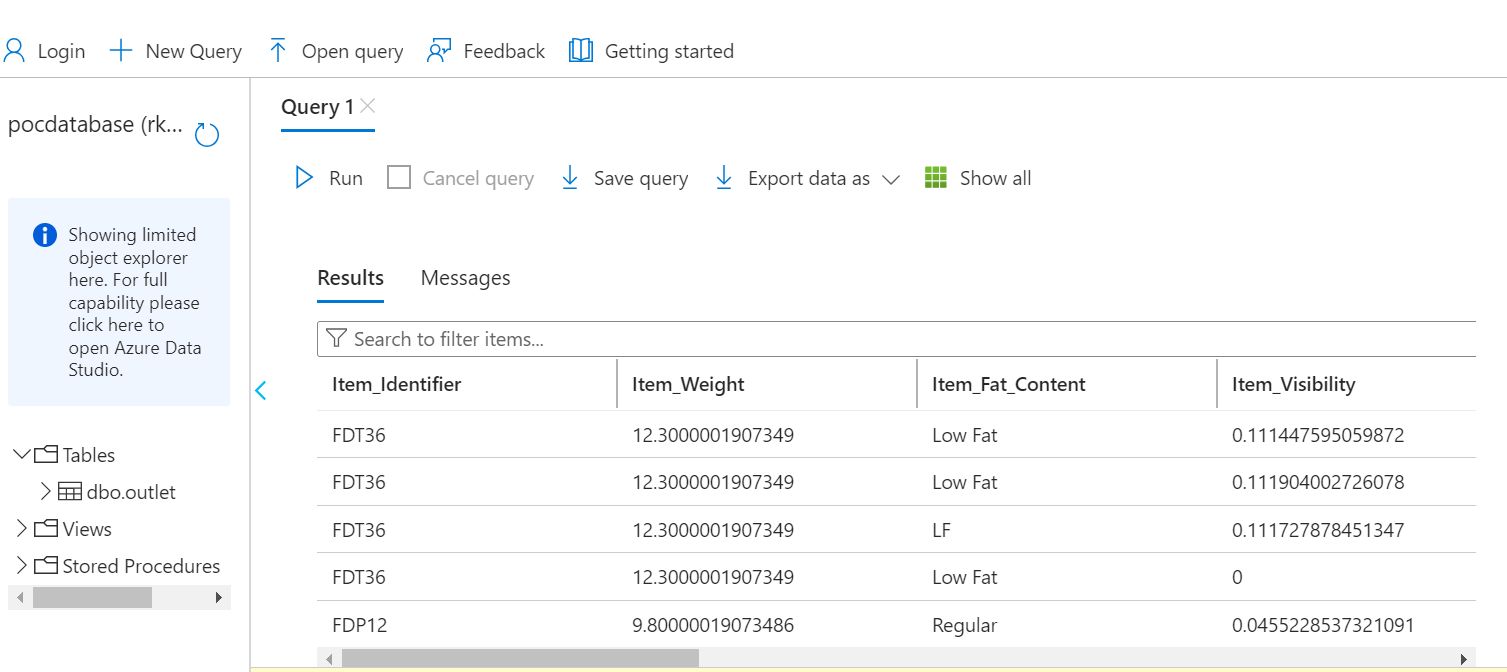


For this demo I’m using this data : <https://drive.google.com/file/d/1kbd1Ew8W8m_Dfeq5zKq3nP6kdeKzK61l/view>

Using the provided data, create a table and insert the data into it.

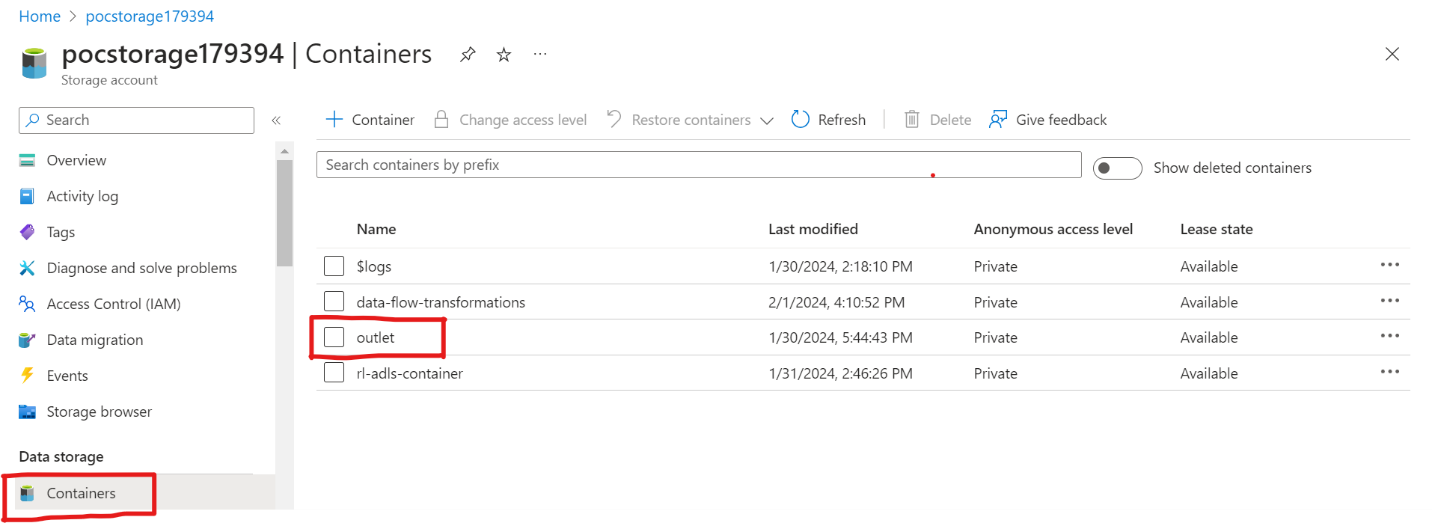


Here are the inserted records.



2) **Create a Data Lake Storage Gen2**: ADLSgen2 will be the Data Lake storage on top of which the Delta Lake will be created.

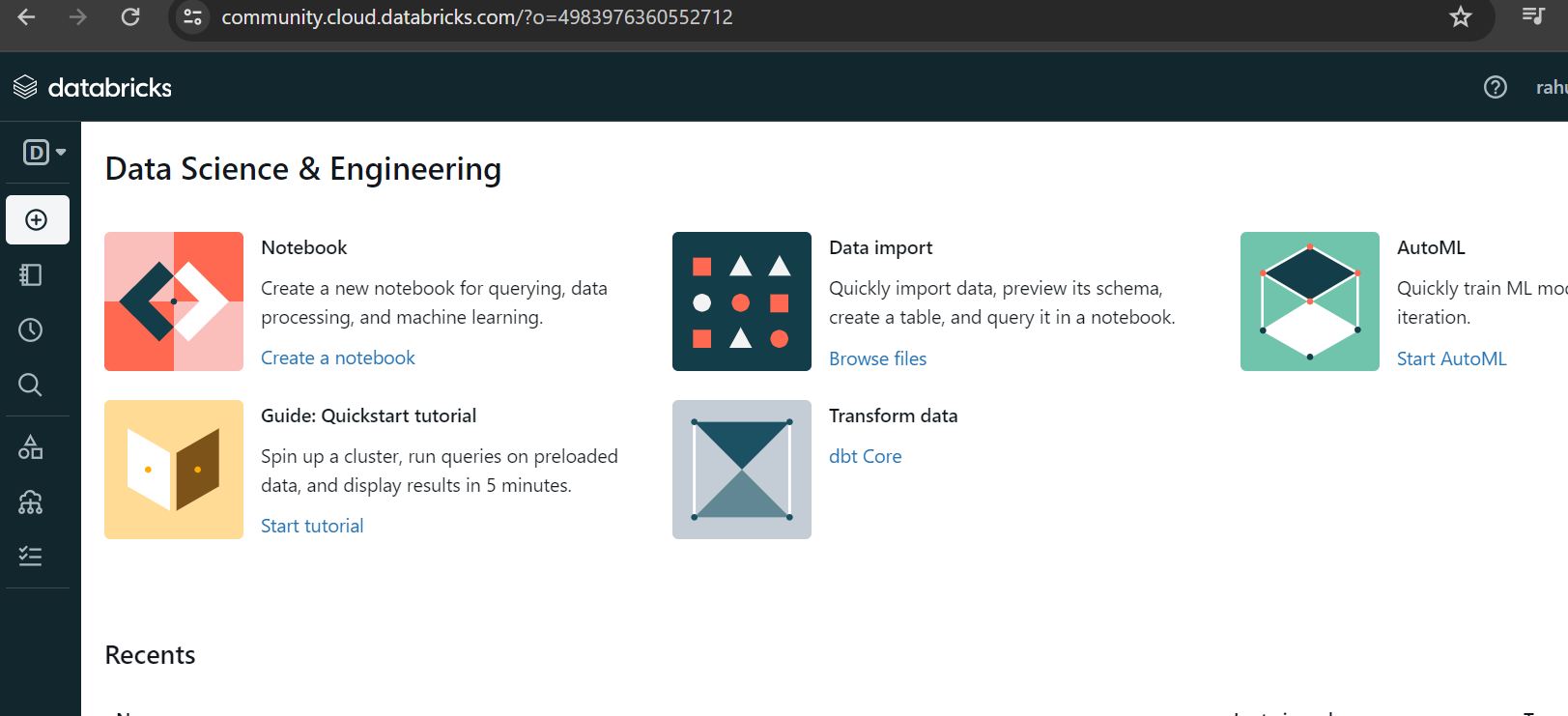
3) **Create Data Lake Storage Gen2 Container and Zones**: Once your Data Lake Gen2 is created, also create the appropriate containers .This demo will use the outlet container to store a final data(Load data).



Now that all pre-requisites are in place, we are ready to create **data bricks** to perform ETL transformation.

**Create a Databricks service**

For this demo, we'll use Databricks Community Edition. Follow the steps to create a Spark cluster in Databricks:

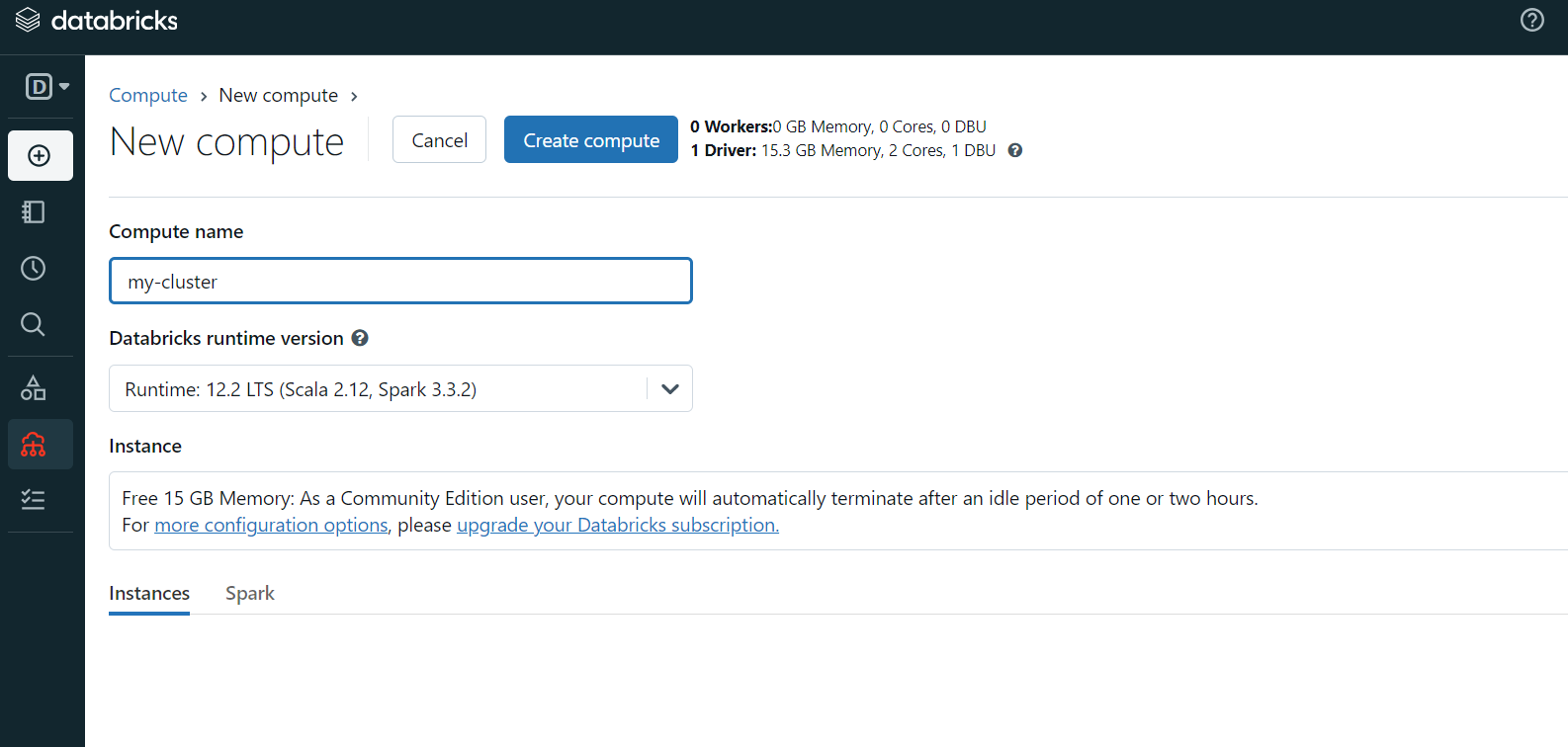


## Create a Spark cluster in Databricks

1)From the portal, select **compute then select create compute**.

## 

2)In the **New cluster** page, provide the values to create a cluster.



Fill in values for the following fields, and accept the default values for the other fields:

* + Enter a name for the **cluster**.
  + Select **Create cluster**. After the cluster is running, you can attach notebooks to the cluster and run Spark jobs.

## Create a Databricks Notebook

## 1) On the left, select Workspace. From the Workspace drop-down, select Create > Notebook.

## 

## 2) Enter a name for the notebook, select Python as the language, and choose the Spark cluster.

## 

## Extract data from the Azure SQL Database

Use the following sample code to extract data from Azure SQL database:

## %python

## df = spark.read.format("jdbc")\

## .option("url", "jdbc:sqlserver://{servername:portname};databaseName={databasename};")\

## .option("dbtable", "{table\_name}")\

## .option("user", "username")\

## .option("password", "\*\*\*\*\*\*")\

## .load()

## Note: Replace placeholders with actual values

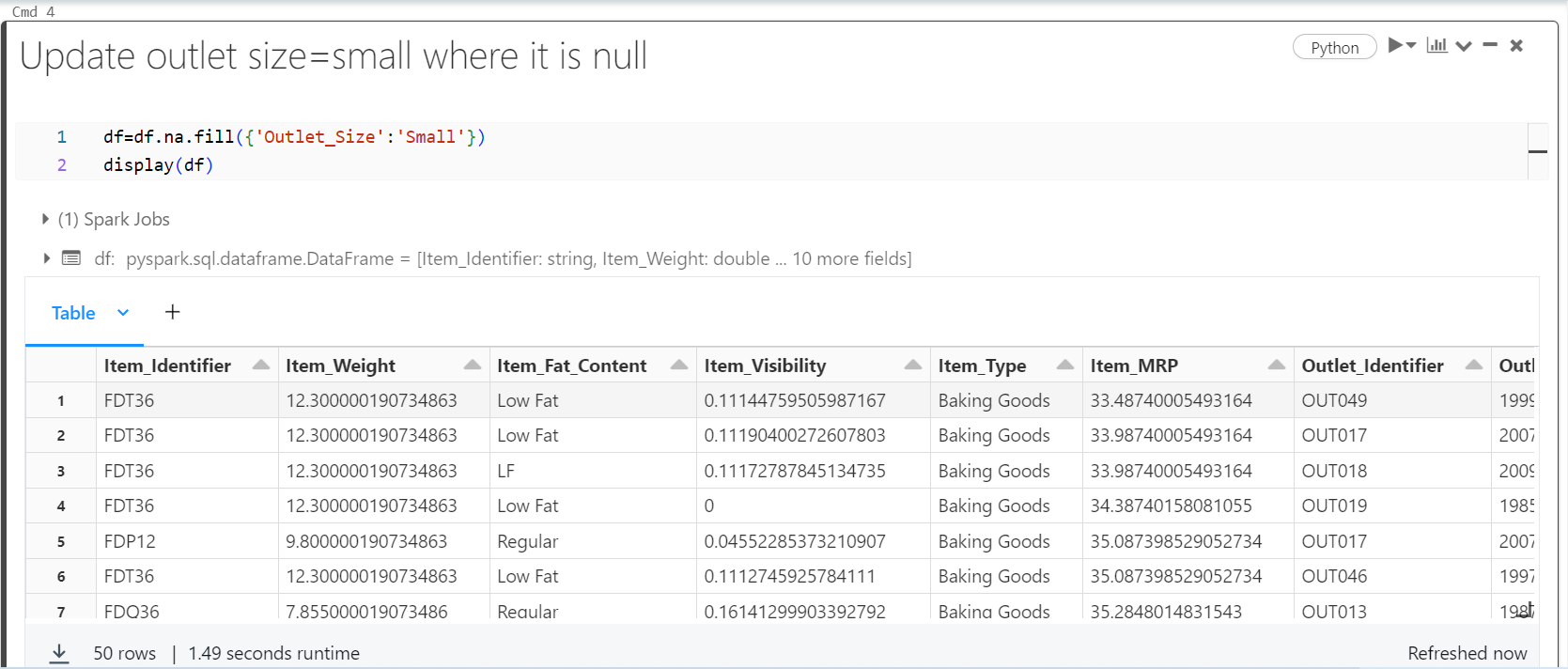
## 

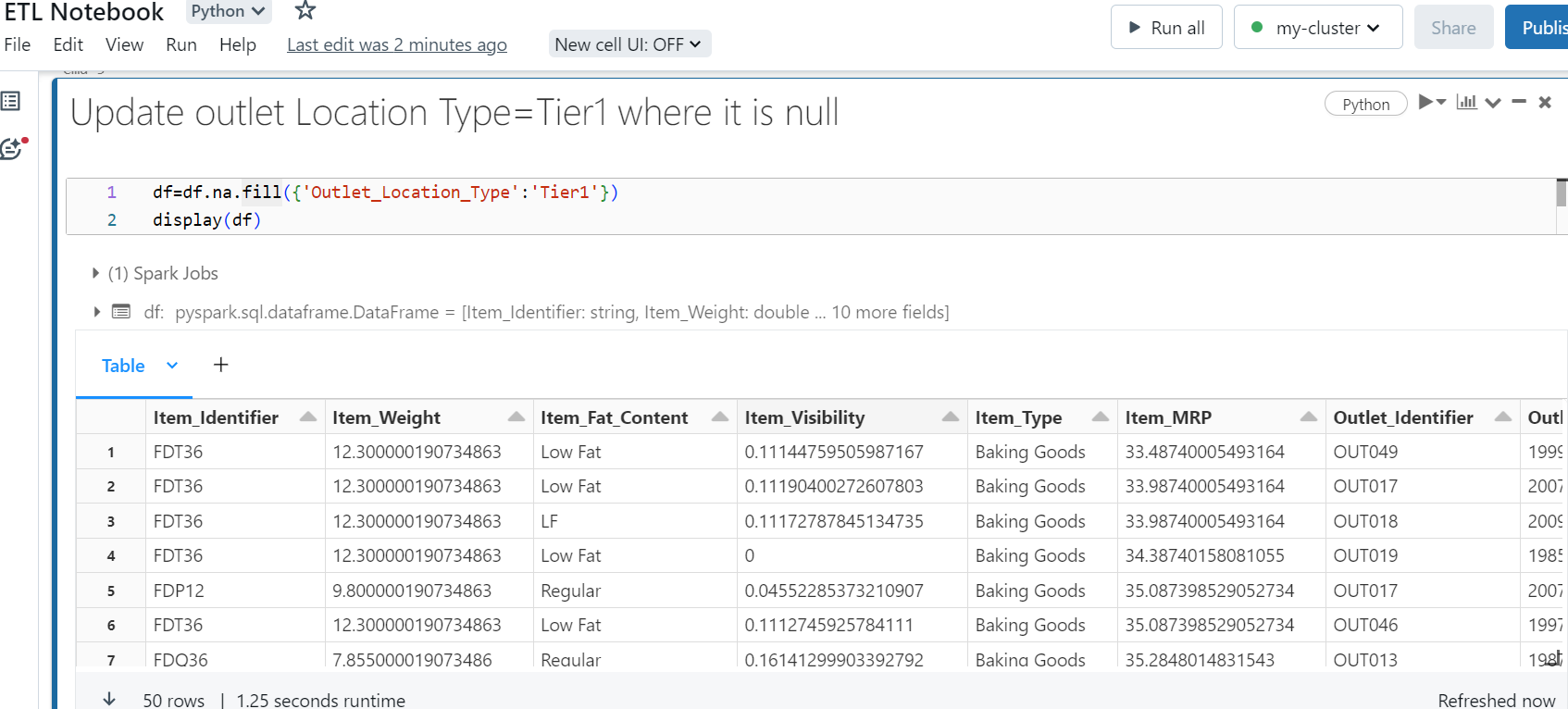
## Once we extract the data from Azure SQL run below command to see your SQL data into the data frame.

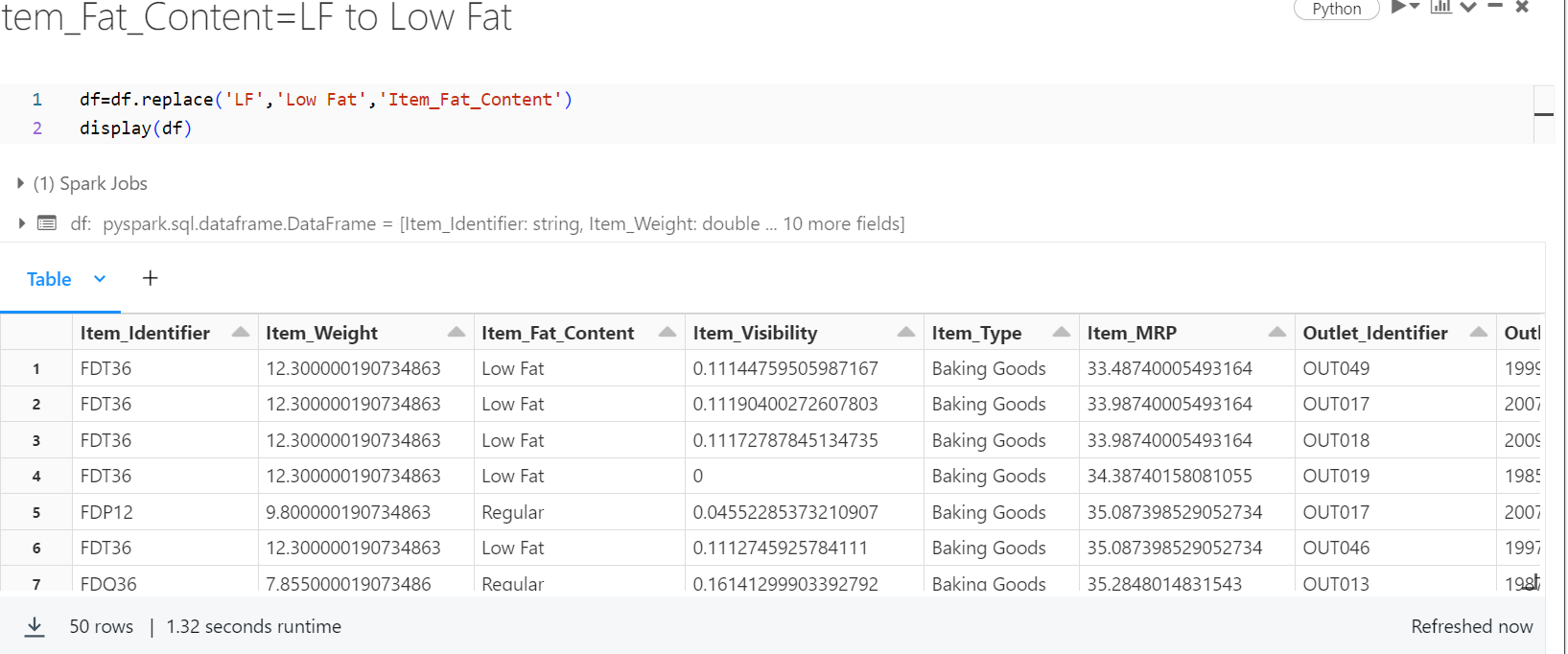
## 

## Transform data in Databricks

Perform transformation operations on the extracted data. [Include code snippets and examples]







In the above we perform some transformation operations. Now Its time to load this result set data into ADLS Gen2.

## Load data into Azure Data Lake Storage Gen2

## In this section, we upload the transformed data into Azure Data Lake Storage Gen2.

## Here is the sample code to load the data from Azure Data Lake Storage Gen2

%python

dbutils.fs.mount(

source = "wasbs://{container}@{storage}.blob.core.windows.net",

mount\_point="/mnt/outlet1",

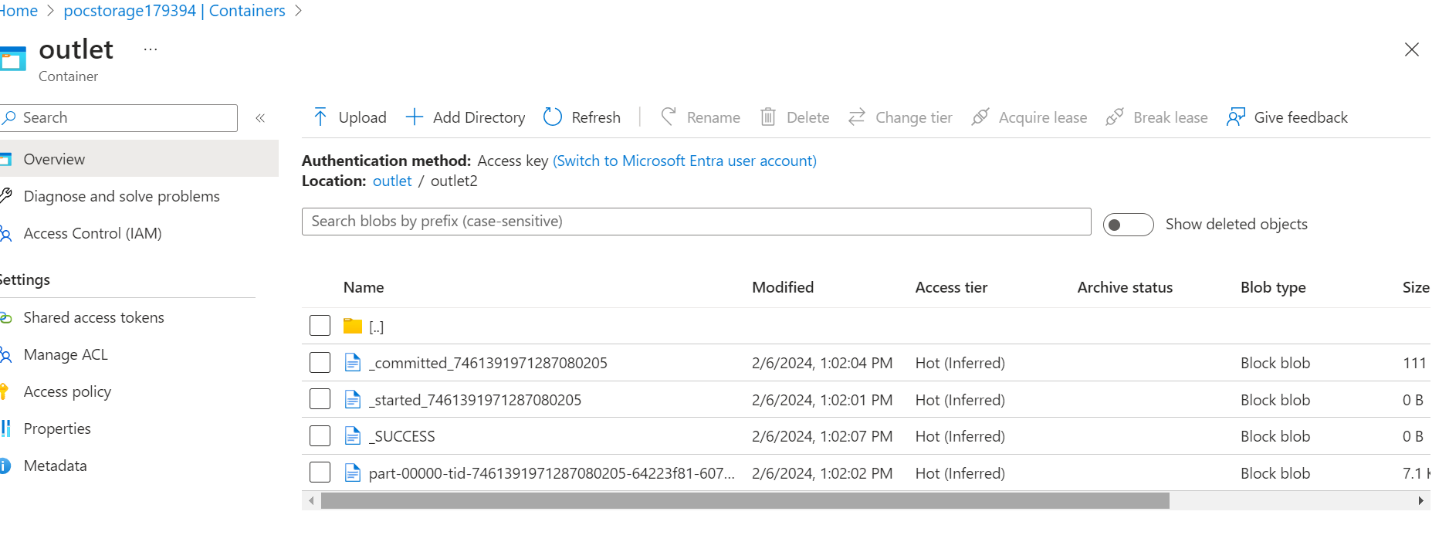
extra\_configs={"fs.azure.account.key.pocstorgate52662.blob.core.window.net":"{access\_key}"})

**Note**: Replace placeholders with actual values.

## 

## 

As expected, once the data is loaded into Azure Data Lake Storage Gen2, check the ADLS2 outlet folder to ensure records have been created.



This documentation provides a comprehensive guide for performing ETL processes using Databricks on large datasets. Adjust the code snippets based on your specific configurations and requirements