**Project: Hybrid Cloud Data Movement**

**Team Members-**

**Subrat Shukla, DE-1**

**Aniroop Gupta, DE-1**

**Harish ER, DE-1**

* **Project Overview:**

Implement a solution that involves moving data between on-premises data sources and Azure cloud using Azure Data Factory, and perform data processing tasks in Azure Databricks.

* **Project Description:**

This project focuses on implementing a scalable and efficient solution to move data between on-premises data sources and Azure cloud while performing advanced data processing using Azure Data Factory (ADF) and Azure Databricks. The aim is to establish a robust data pipeline for seamless integration, transformation, and analytics in the cloud environment.

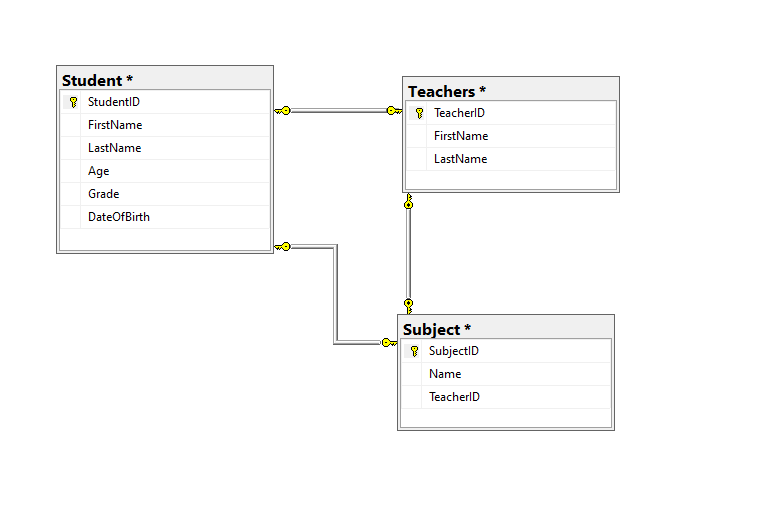
* **Data Overview:**

The **Student Database** includes a primary table named **Student**, designed to manage essential information about students. This table contains the following columns:

* **StudentID** (unique identifier for each student),
* **FirstName** and **LastName** (to capture the student’s name),
* **Age** (representing the student's current age),
* **Gender** (to identify the gender of the student), and
* **DOB** (Date of Birth, to record the student's birth date).

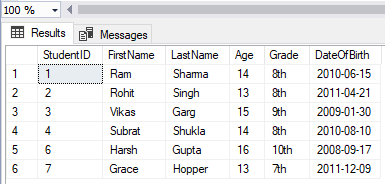
The structure ensures the efficient storage and retrieval of student records for administrative or analytical purposes. This database is crucial for maintaining up-to-date student information and can support operations such as reporting, data analysis, and integration with other systems.

* **ER Diagram:**



* **How it works:**

We created a sql database named StudentDB, in that we made tables names as Student, Teachers, Subject with required details. Here is the overview of data file.



* **Execution Overview:**

The Hybrid Cloud Data Movement project involves creating an end-to-end pipeline to move and process data using Azure Data Factory and Azure Databricks.

**Steps to Implement the Solution:**

1. **Understanding Requirements**:
   * Identify the **on-premises data source(s)** (e.g., SQL Server, Oracle, flat files).
   * Define the **destination** in Azure (e.g., Azure Blob Storage, Azure SQL Database, Data Lake).
   * Specify data processing requirements (e.g., data transformations, cleaning, aggregations).
2. **Setting Up Azure Environment**:
   * Creating a new Azure Data Factory (ADF) instance.
   * Creating a new storage account container in Azure.
   * Set up Azure Databricks workspace in Azure.
   * Create linked services in ADF to connect with on-premises sources using the **Self-hosted Integration Runtime** for secure data movement.
3. **Data Movement (ADF)**:
   * **Pipelines**:
     + Build an ADF pipeline to:
       1. Extract data from on-premises sources.
       2. Load it into Azure storage (Blob Storage or Data Lake).
   * **Triggers**:
     + Schedule the pipelines using triggers (time-based or event-driven).
4. **Data Processing (Azure Databricks)**:
   * **Databricks Notebooks**:
     + Develop notebooks to process the ingested data.
     + Perform tasks like filtering, aggregations, joins, and formatting data for downstream use.
   * **Integrate with ADF**:
     + Use ADF to trigger Databricks jobs by calling its API or Notebook activity.
5. **End-to-End Workflow**:
   * Combine data movement and data processing workflows into a cohesive ADF pipeline.
   * Ensure proper error handling, logging, and retries for resilience.

* **Azure resources used for this project:**

1. Azure Data Factory (ADF)
2. Azure Storage Account
3. Azure Data factory
4. Azure Databricks

* **Project Requirements:**

1. Create new Azure Data Factory
2. Create a new container from Storage accounts, to store data.
3. Download on premise data store, we used SQL Server.
4. Download Microsoft Integration Runtime.

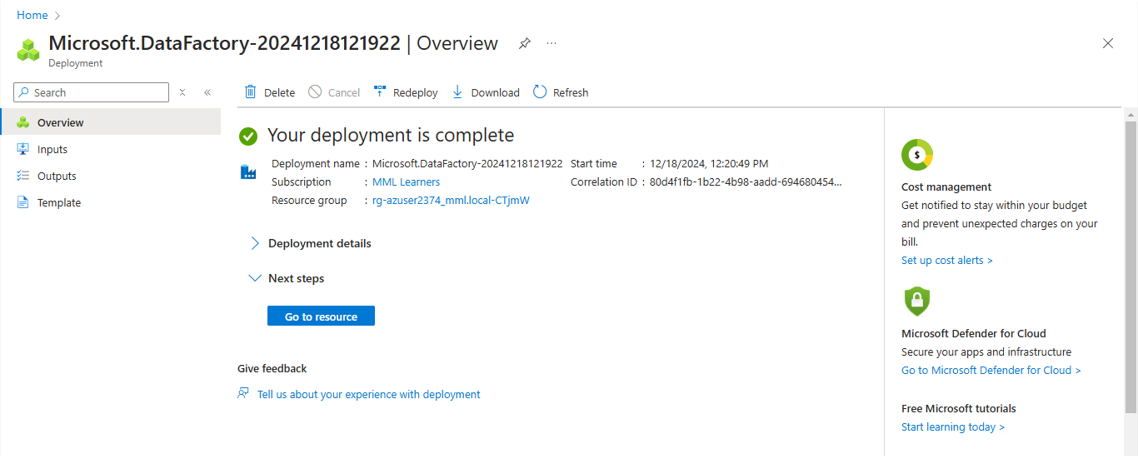
* **Tasks Performed:**

1. Configured **Azure Data Factory (ADF)** and set up linked services to connect with on-premises data sources and Azure storage.
2. Created integration runtime to connect self-host node to cloud service
3. Built pipelines in ADF to **extract, transform, and load (ETL)** data from on-premises systems to Azure Blob Storage.
4. Scheduled and tested pipelines using **triggers** (time-based or manual runs).
5. Connected SQL Server Database with the Azure Blob Storage.
6. Integrated Databricks with ADF for automated workflow execution.
7. Developed **Databricks notebooks** to perform data transformations.
8. Conducted thorough testing to ensure data integrity during migration.

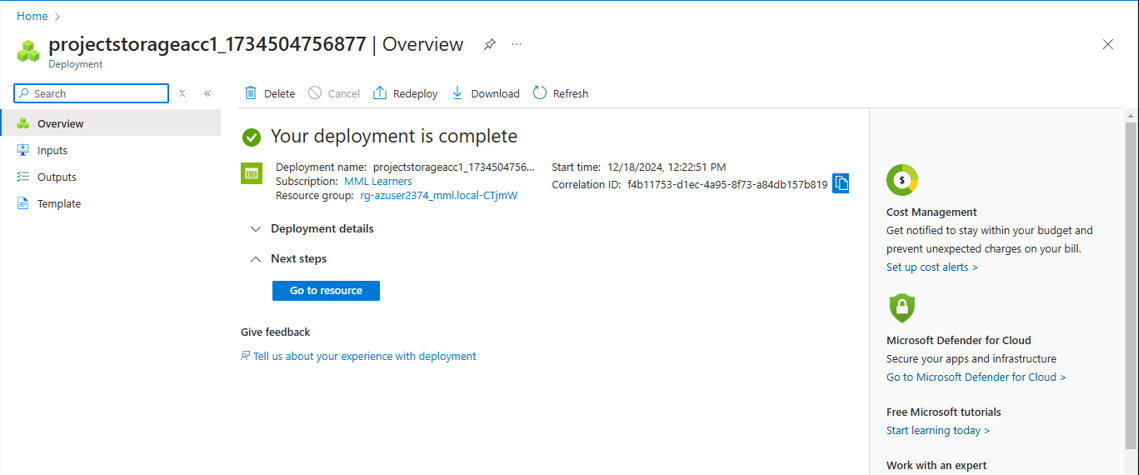
This implementation enables a seamless, scalable pipeline for hybrid cloud data movement and processing, leveraging Azure’s capabilities.

* **Analysis Results:**

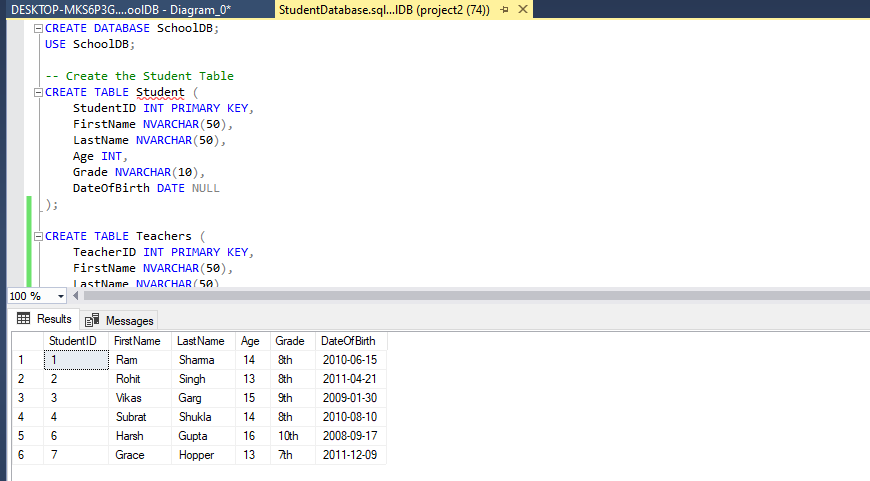
1. **Creating new Data Factory:**



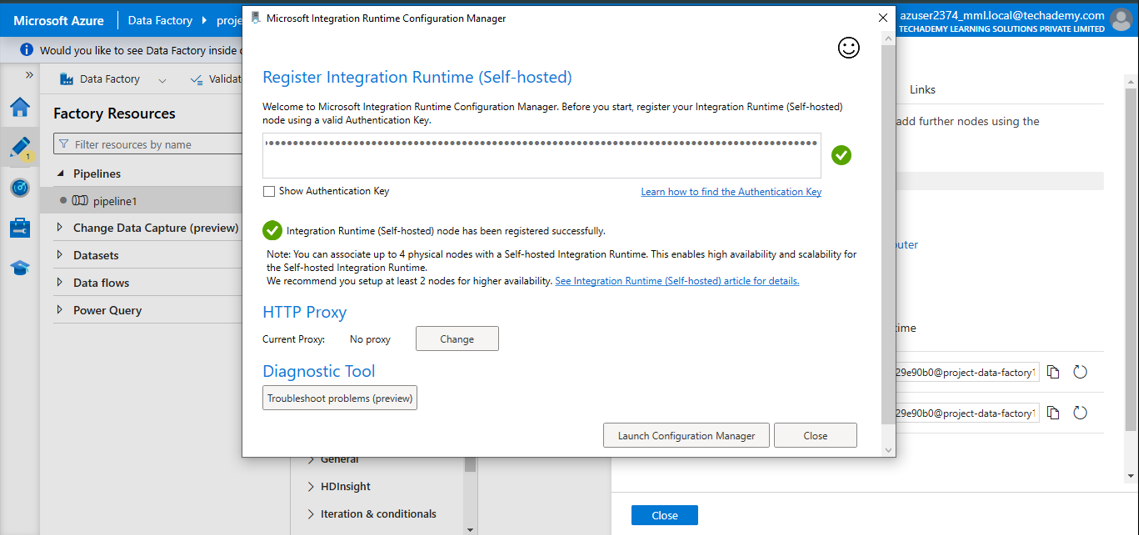
1. **Creating storage account:**

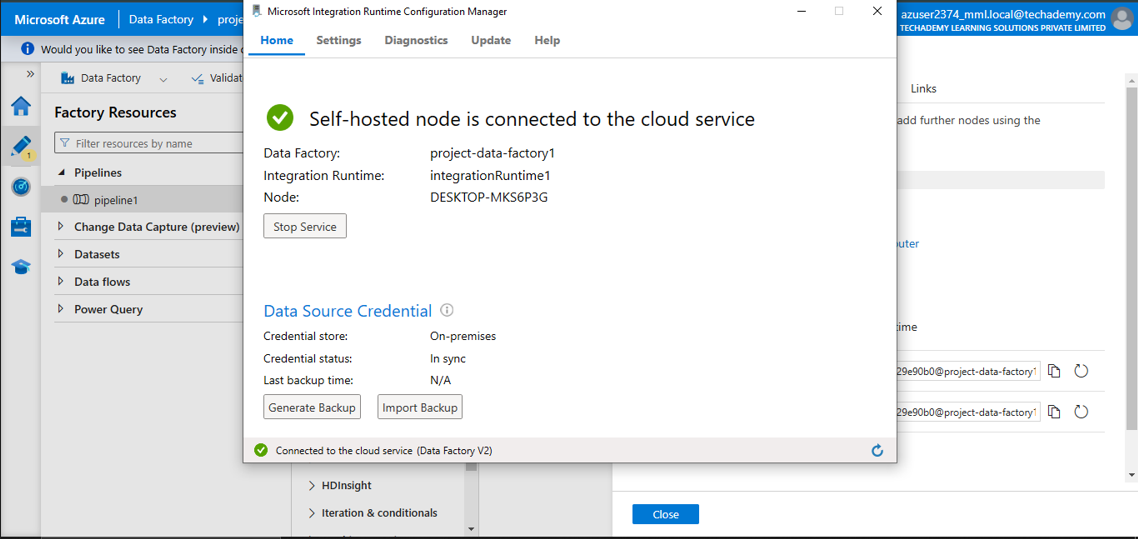


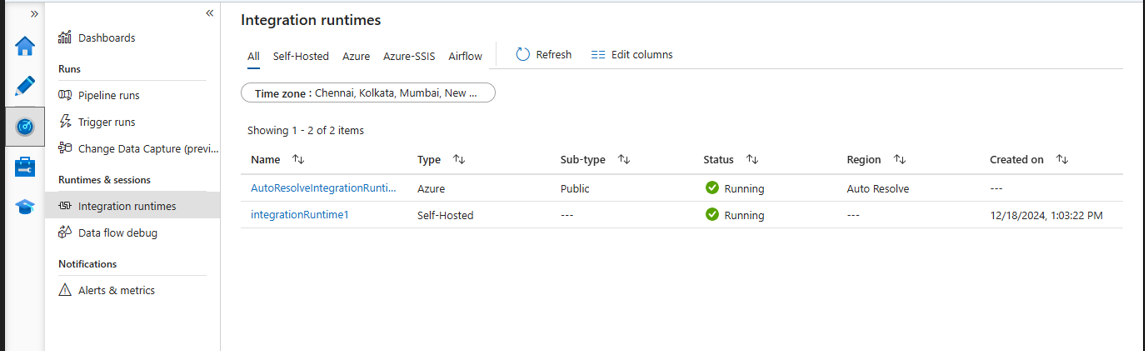
1. **Creating database on premise SQL Server:**



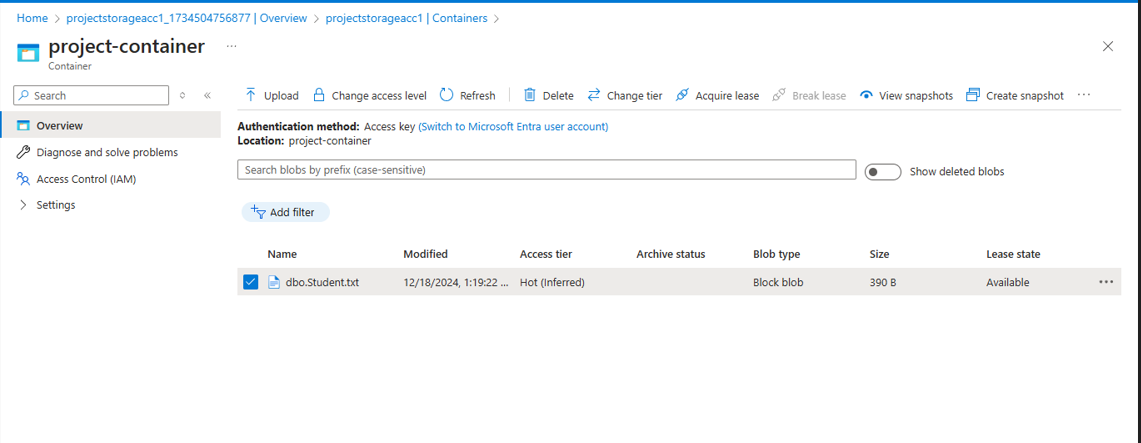
1. **Integrating Integration Runtime to connect self-host node with cloud:**

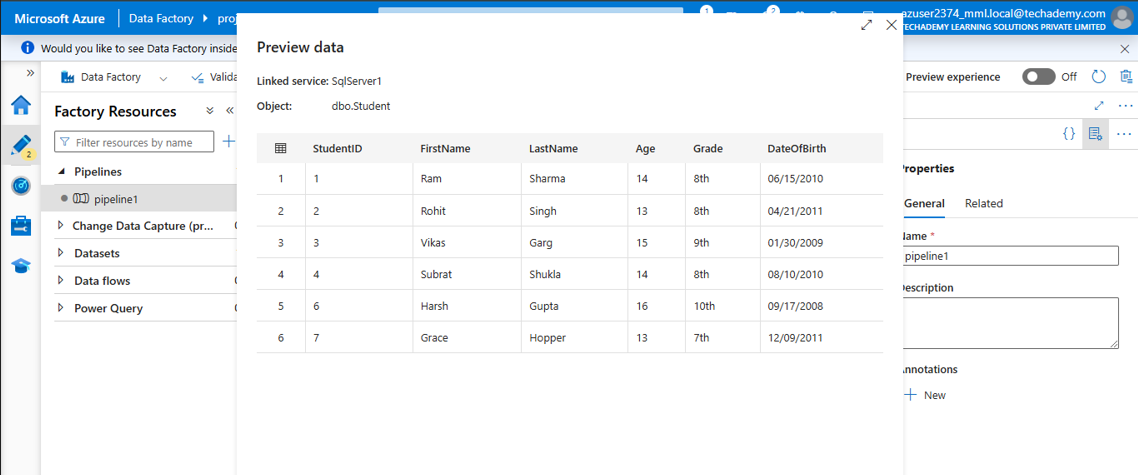




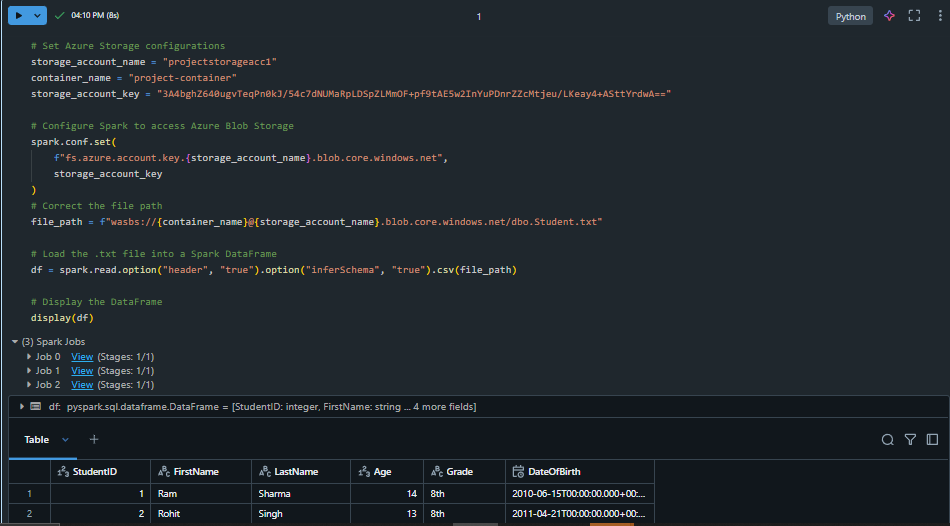


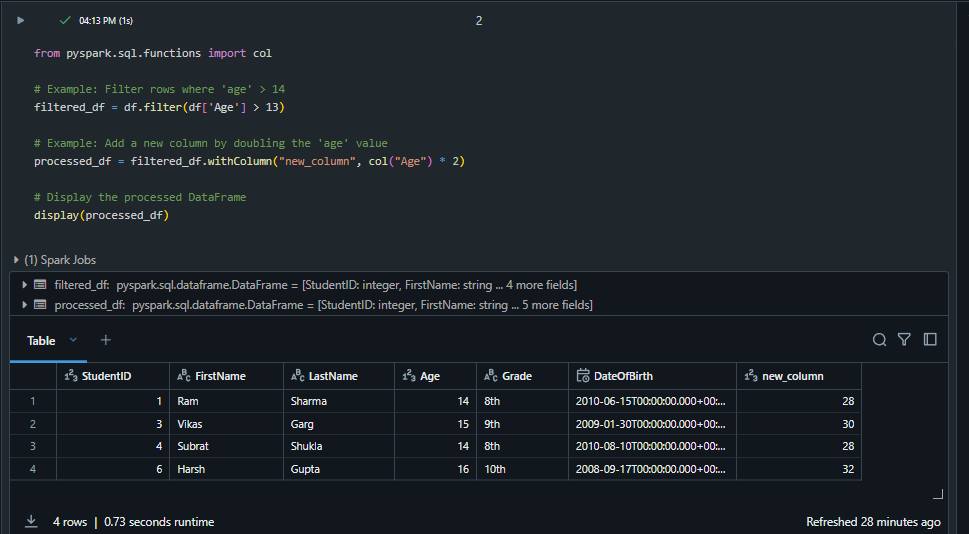
1. **Migrated data to the Storage container:**

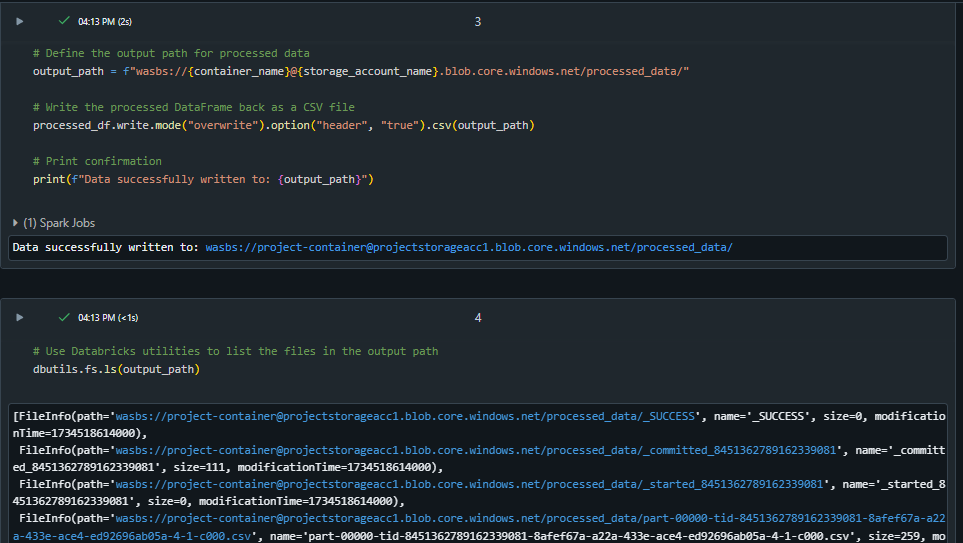




1. **Integrate Databricks with ADF for automated workflow:**



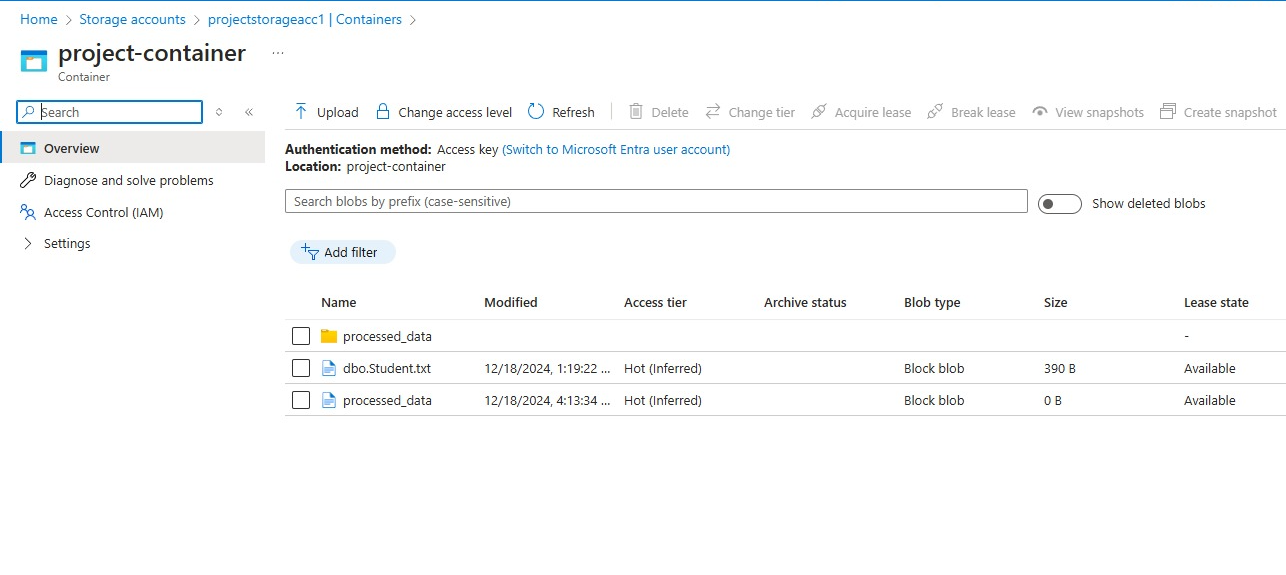




This ETL process ensures the data first gets converted into processed form and then migrates the processed data to the Blob Storage Container.

Let us see the final processed data folder in Blob Storage.

1. **Processed Data migrated to Blob Storage:**



* **Conclusion:**

The **Hybrid Cloud Data Movement Project** successfully demonstrates a robust and efficient solution for integrating on-premises data with the Azure cloud. By leveraging **Azure Data Factory** for seamless data movement and **Azure Databricks** for advanced processing, the project enables scalable, high-performance data pipelines suitable for real-world scenarios. The solution ensures secure, reliable, and automated workflows, optimized to handle complex transformations and large data volumes. With comprehensive testing, monitoring, and documentation, this project lays a strong foundation for hybrid cloud implementations, empowering businesses to unlock insights and drive data-driven decisions with ease.

**Submitted By-**

**Subrat Shukla, DE-1**

**Harish ER, DE-1**

**Aniroop Gupta, DE-1**

**--Thank You!**