**Olympic Data Analytics**

**GROUP 10**

**Prerak Panwar (Roll number 30), Yi Hsiang Hung (Roll number 17)**

**ABOUT THE PROJECT:**

In this project, our objective is to conduct a comprehensive analysis of the Olympic dataset available on Kaggle. To achieve this, we will implement an end-to-end data engineering project, incorporating various Azure services.

**1. Data Extraction using Azure Data Factory:**

We will initiate the project by extracting data from the Olympic API using Azure Data Factory.

Azure Data Factory, a powerful data pipeline tool in the Azure ecosystem, will be employed to create a seamless flow for retrieving and loading our data into Azure Data Lake storage.

**2. Data Transformation with Azure Databricks:**

Following the data extraction phase, we will utilize Azure Databricks to employ Spark code for data transformation.

This step involves cleaning, structuring, and enhancing the raw data. The transformed data will then be stored back into the Azure Data Lake storage for further processing.

**3. Data Analysis using Azure Synapse Analytics:**

Azure Synapse Analytics will be employed to run SQL queries on the transformed data.

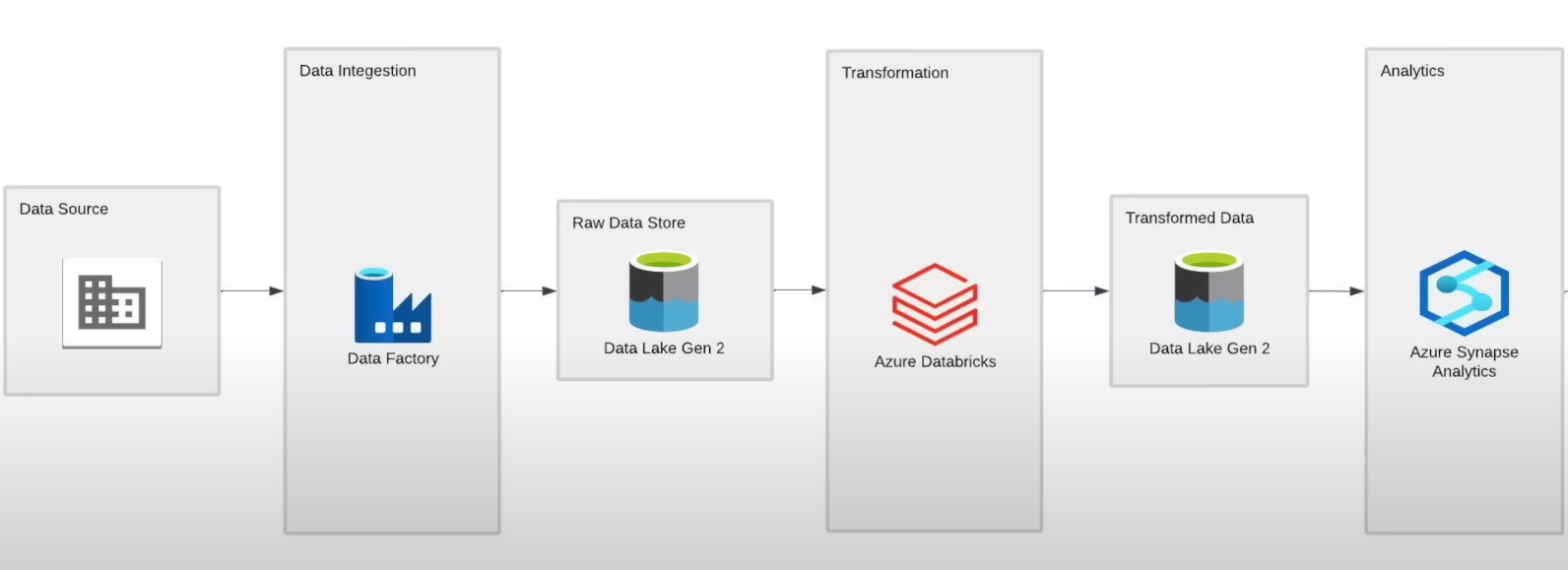
This step will provide valuable insights into the dataset, enabling us to derive meaningful information.

**4. Data Visualization in PowerBI (Optional):**

For enhanced visualization and presentation of our findings, we have the option to integrate with business intelligence tools like PowerBI. This step will allow us to create visually appealing and informative dashboards based on the analyzed Olympic dataset.

This structured approach ensures a systematic and efficient execution of the Olympic dataset analysis, leveraging the capabilities of Azure services at each stage of the data engineering process.

**This picture represents the complete Architecture of the Project work:**



**More information about the Azure services:**

**1-Azure Data Factory** (for Data Ingestion):

Data integration service that enables you to create, schedule, and manage data pipelines for efficient data movement and transformation between various sources and destinations in Azure and beyond. It simplifies ETL (Extract, Transform, Load) and data integration tasks.

**2-Data Lake Gen 2** (for Data Storage):

Data lake solution that combines the capabilities of a data lake with the power of Azure Blob Storage, allowing you to store and analyze large volumes of structured and unstructured data with enhanced performance, security, and analytics capabilities.

**3-Azure Databricks** (for Data Transformation)

Databricks is a unified analytics platform built on top of Apache Spark, designed to help data engineers and data scientists collaborate on big data processing and machine learning tasks. It provides tools for data exploration, data processing, and building machine learning models in a collaborative and scalable environment.

**4-Azure Synapse Analytics** (for Data Analysis)

SQL Data Warehouse is a cloud-based analytics service provided by Microsoft Azure. It combines big data and data warehousing into a single integrated platform, allowing organizations to analyze and process large volumes of data for business intelligence and data analytics purposes.

**DETAILS ON THE DATASET USED:**

The source of data is Kaggle and the data can be accessed from the following link:

<https://www.kaggle.com/datasets/arjunprasadsarkhel/2021-olympics-in-tokyo>

This dataset has 5 tables (Athletes, coaches, EntriesGender, Medals, and Teams).

This contains the details of over 11,000 athletes, with 47 disciplines, along with 743 Teams taking part in the 2021(2020) Tokyo Olympics.

This dataset contains the details of the Athletes, Coaches, and Teams participating as well as the Entries by gender. It contains their names, countries represented discipline, gender of competitors, and names of the coaches.

We first converted the .XLSX format to .CSV format for ease of use. After converting it we made a GitHub Repository and loaded the data there.

The link of the GitHub repository is: <https://github.com/prerakpanwar/Data_Engineering_Azure>

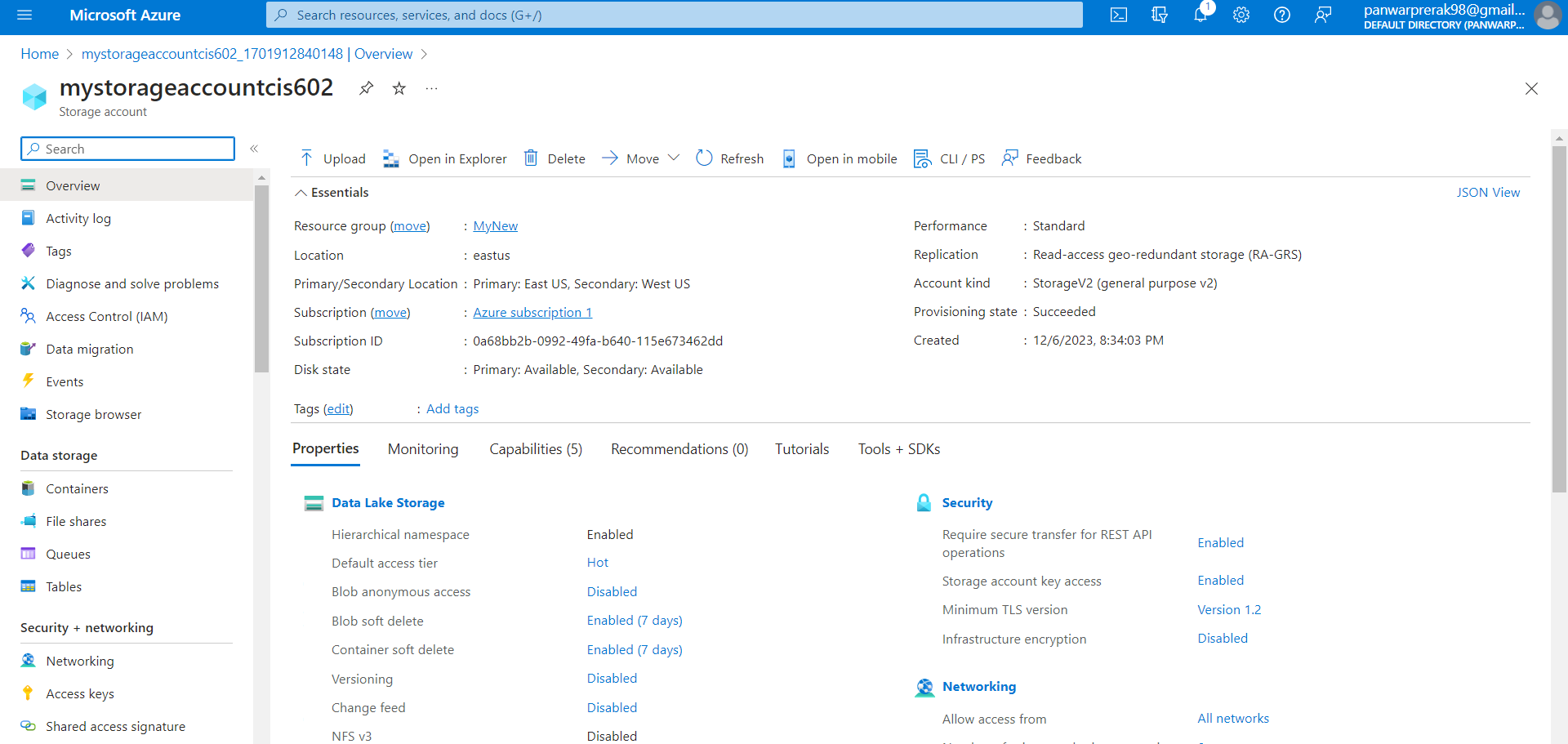
We did this step as it was necessary to save our work without losing it on Azure.

After loading the data we extracted the data from the GitHub repository using the Azure Data Factory Source option which we will explain in the next steps.

**AZURE STORAGE ACCOUNTS STEPS:**

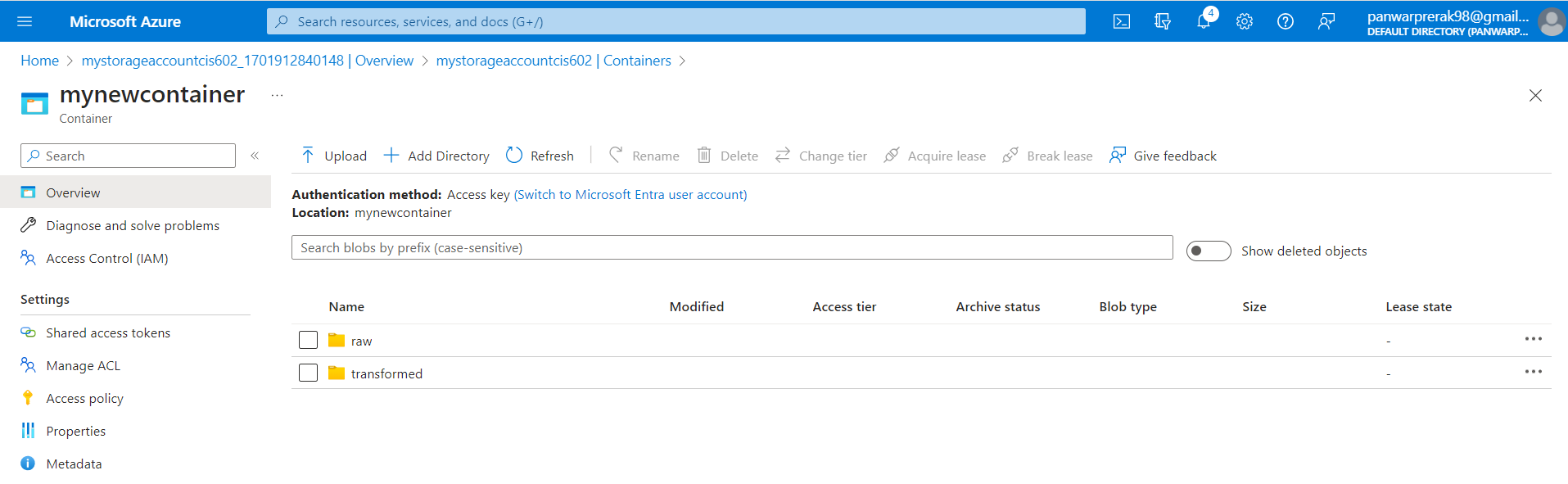
1. Go to the Azure Storage accounts.
2. Click ‘+Create’/ ‘Create storage account’ option.
3. For the Basics tab choose the following: (Subscription, Resource group, Storage account name, Region, Performance, and Redundancy)
4. For the Advanced tab enable the Hierarchical Namespace option (This allows the collection of objects/files within an account to be organized into a hierarchy of directories and nested subdirectories in the same way that the file system on your computer is organized.)
5. Leave the rest settings as it is OR change as per your requirements.
6. Review and Create the storage account.

**The below screenshot tells us about the details of the storage account created.**



**After this we will Create Data storage:**

1. Go to Containers in the Data storage types.
2. Click on +Container 🡪give the name of the container 🡪 click Create.
3. Now open the container you just created and click on +Add Directory to create the required folders.
4. Make 2 Directory names (raw and transformed). You can name it as per your choice, here raw folder will store raw data and similarly transformed data will be stored in the transformed folder.



**After doing these steps we will go to Azure Data Factory.**

**AZURE DATA FACTORY STEPS:**

1. Go to the Azure Data factories.
2. Click ‘+Create’/ ‘Create data factory’ option.
3. For the Basics tab choose the following: (Subscription, Resource group, and Instance details) NOTE: choose the same Resource Group.
4. Leave the rest settings as it is OR change as per your requirements.
5. Click Review + Create 🡪 and then click Create.
6. Now go to Resource and Launch Studio.
7. Go to the Manage tab 🡪 Git configuration 🡪 Configure

On repository type choose GitHub and give the name of the GitHub repository owner (GitHub username).

Configure a repository by giving the repository name and details and hit apply.

NOTE: This will help us in saving our work.

1. Now Go to the Author tab (This is where we will create our pipeline).
2. Click on + to add a new resource and select pipeline.
3. On the activities tab select the Move and Transform option, then drag and drop the Copy data option to the blank space on the right of it.
4. Now provide the Name, Source, and Sink for the pipeline.

SOURCE: click +New and search for ‘HTTP’, select it and after that select the format type of your data. Here we have .CSV so we will choose DelimitedText as the format.

Now for set properties: give Name and linked service.

For Linked service: create +new 🡪 give Name, Base URL, and Authentication type as Anonymous.

For the Base URL: Open your CSV file on GitHub and click on Raw, after clicking on Raw Copy the HTTP link and paste it on the Base URL option.

After doing all these configurations a Set Properties tab will come up where you have to choose the First row as the header option and click OK.

1. Similarly, for Sink choose Azure Data Lake Storage Gen2 and CSV format.

In the Set Properties tab give name and Linked service.

For linked service provide the subscription, and storage account name which we have created before.

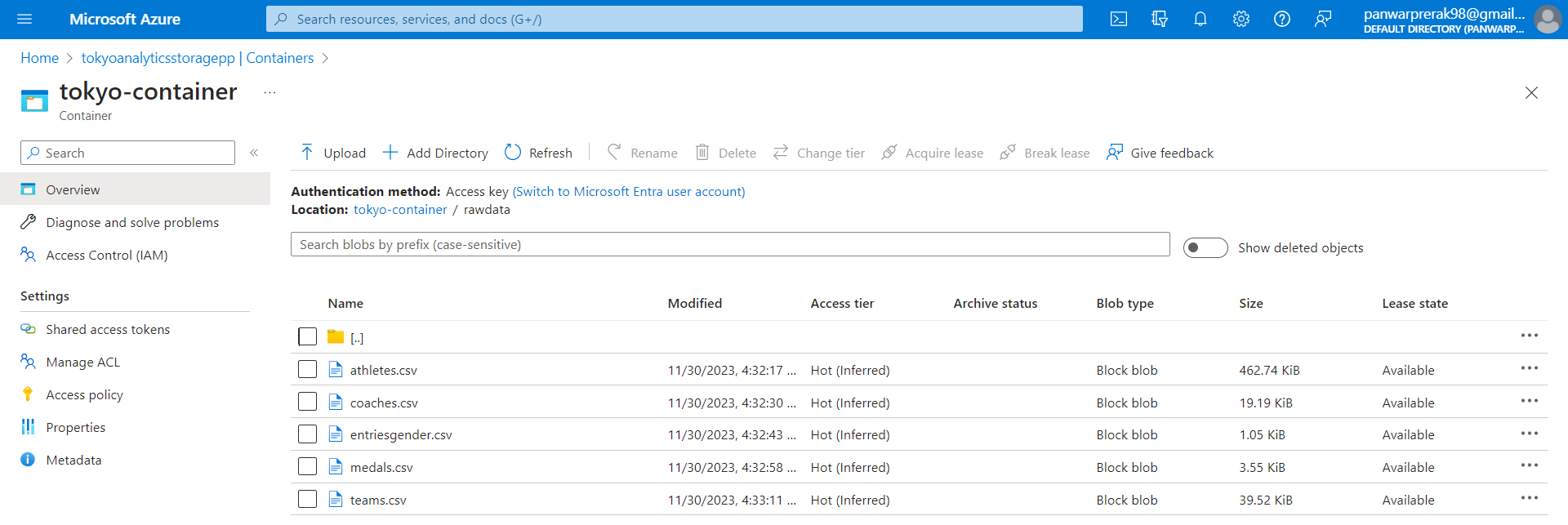
Click Create.

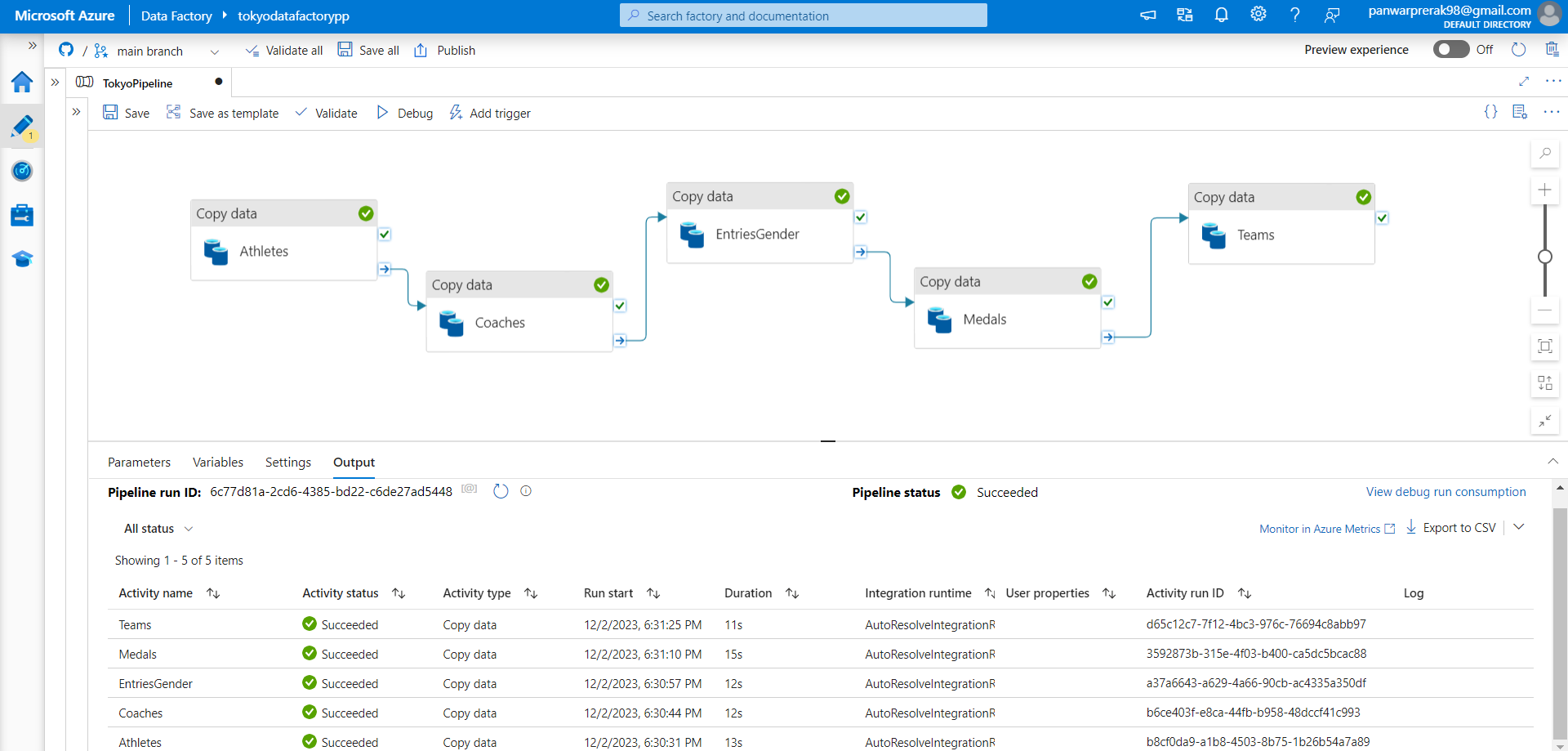
Now provide the file path where you want your Raw data to be stored. In our case, we will provide a file path to our raw folder.

NOTE: This is why we create a storage account first.

Now repeat steps 10 – 12 to ingest the next 4 tables.

**After doing these steps, validation, and debug test the pipeline should look like this.**

****

**Once the pipeline is set successfully with no errors, then you will be able to see all the raw data of 5 tables inside the raw folder as shown above.**

**After doing these steps we will go to Azure Databricks service.**

**AZURE DATABRICKS STEPS:**

1-Go to the Azure Databricks service.

2-Click ‘+Create’/ ‘Create Azure Databricks service’ option.

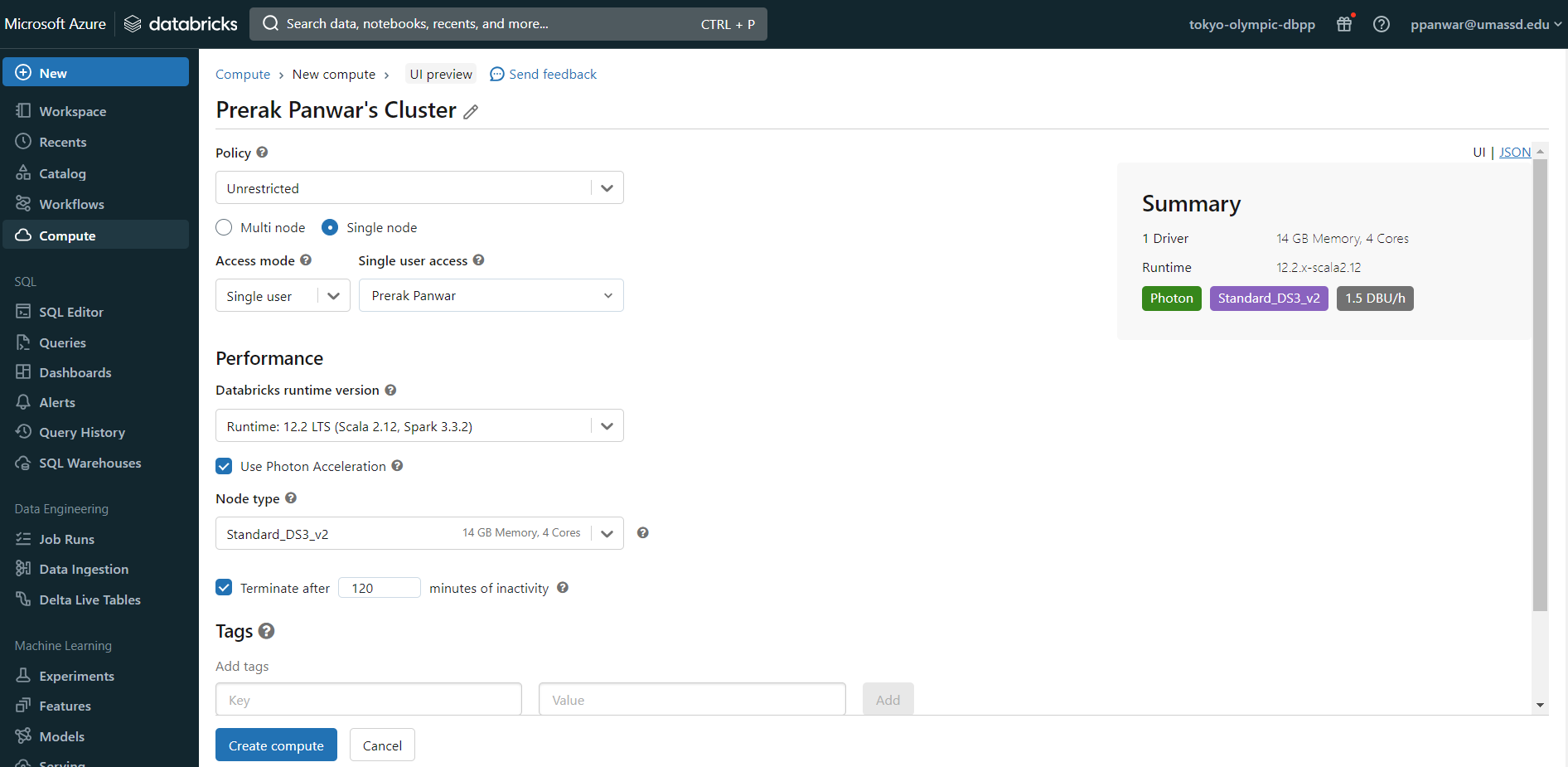
3-For the Basics tab choose the following: (Subscription, Resource group, and Instance details) NOTE: choose the same Resource Group.

4-Leave the rest settings as it is OR change as per your requirements.

5-Click Review + Create 🡪 and then click Create.

6-Now go to Resource and LaunchWorkspace.

7-Once you are logged in to Databricks, go to the Compute tab and configure your Cluster with details as shown in the screenshot/ or as per your workload requirements.

****

8-After configuring all the desired requirements of the cluster click on Create Compute.

Wait for some time for the cluster to get created.

**After the cluster is created successfully we will move to the data transformation steps.**

1-Go to the workspace tab where you will be writing the spark code for data transformation.

Note: make sure you have selected you spark cluster which you created in the previous step.

2-Now search for the App registrations service of Azure.

Click +New registration 🡪 give your app a name 🡪 Click on register.

Now copy the Application (client) ID and Directory (tenant) ID.

Now on the Manage tab select Certificates and secrets where we will create our secret ID. Click on +New client secret 🡪 give some description to it 🡪 Click on add.

Now Copy the Value of the secret key.

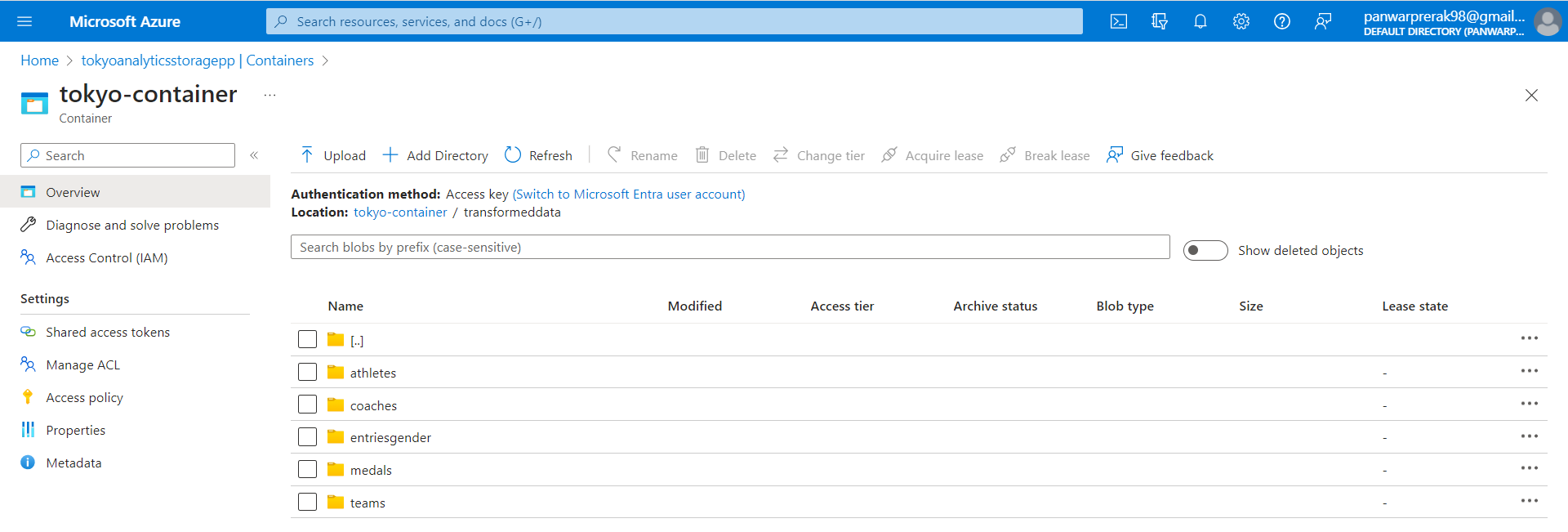
**We have done all the above 2nd step to access the Azure Data Lake Storage from Databricks.**

In our code, we will first make the configuration format using the keys we have copied above.

**The code can be accessed from the link:**

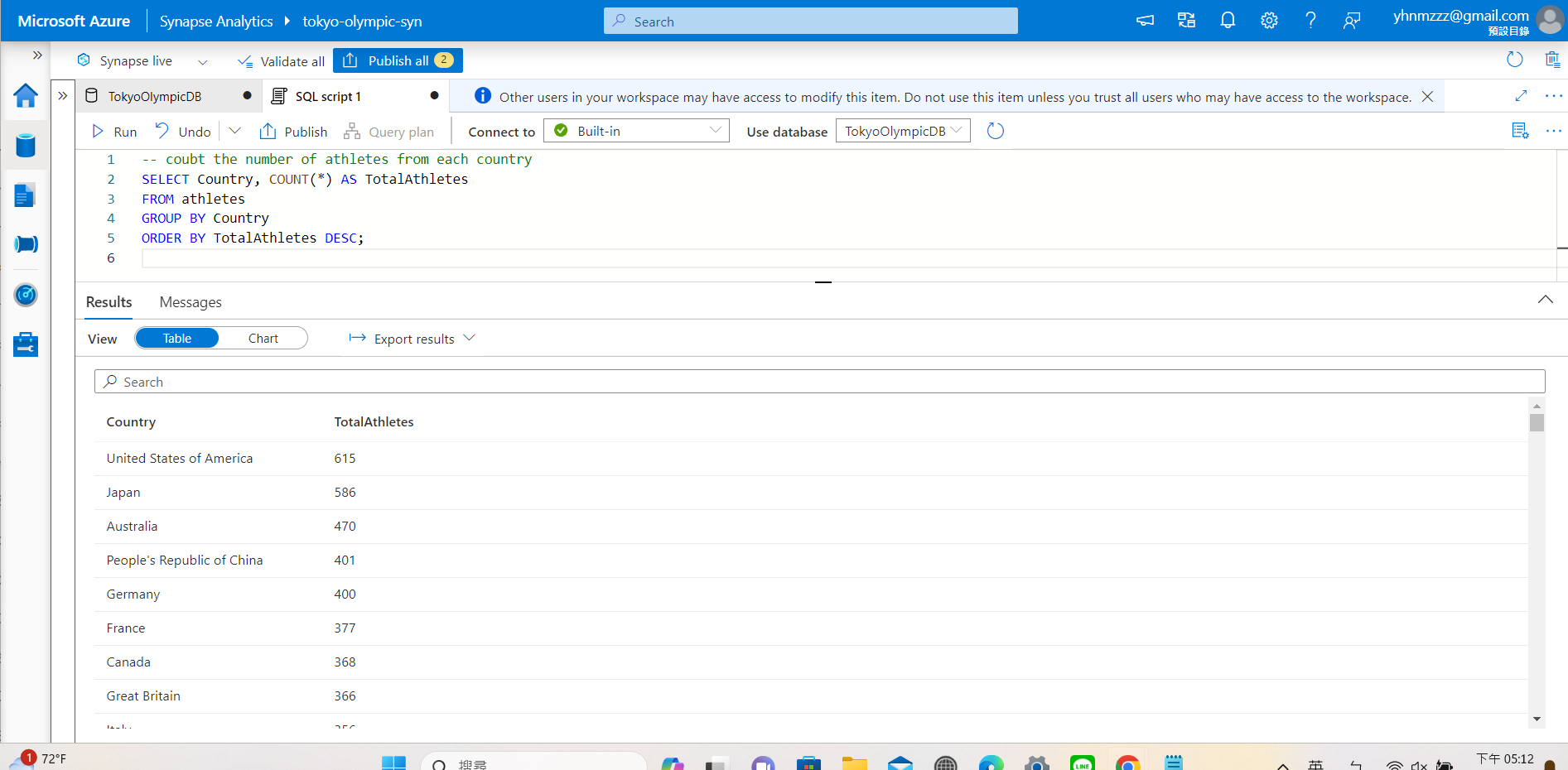
<https://github.com/prerakpanwar/Data_Engineering_Azure/blob/main/Code_for_simple_transformation_on_databricks.ipynb>

**After the transformation is done successfully the transformed data will be available in the transformed folder in our container. The following screenshot represents the same.**

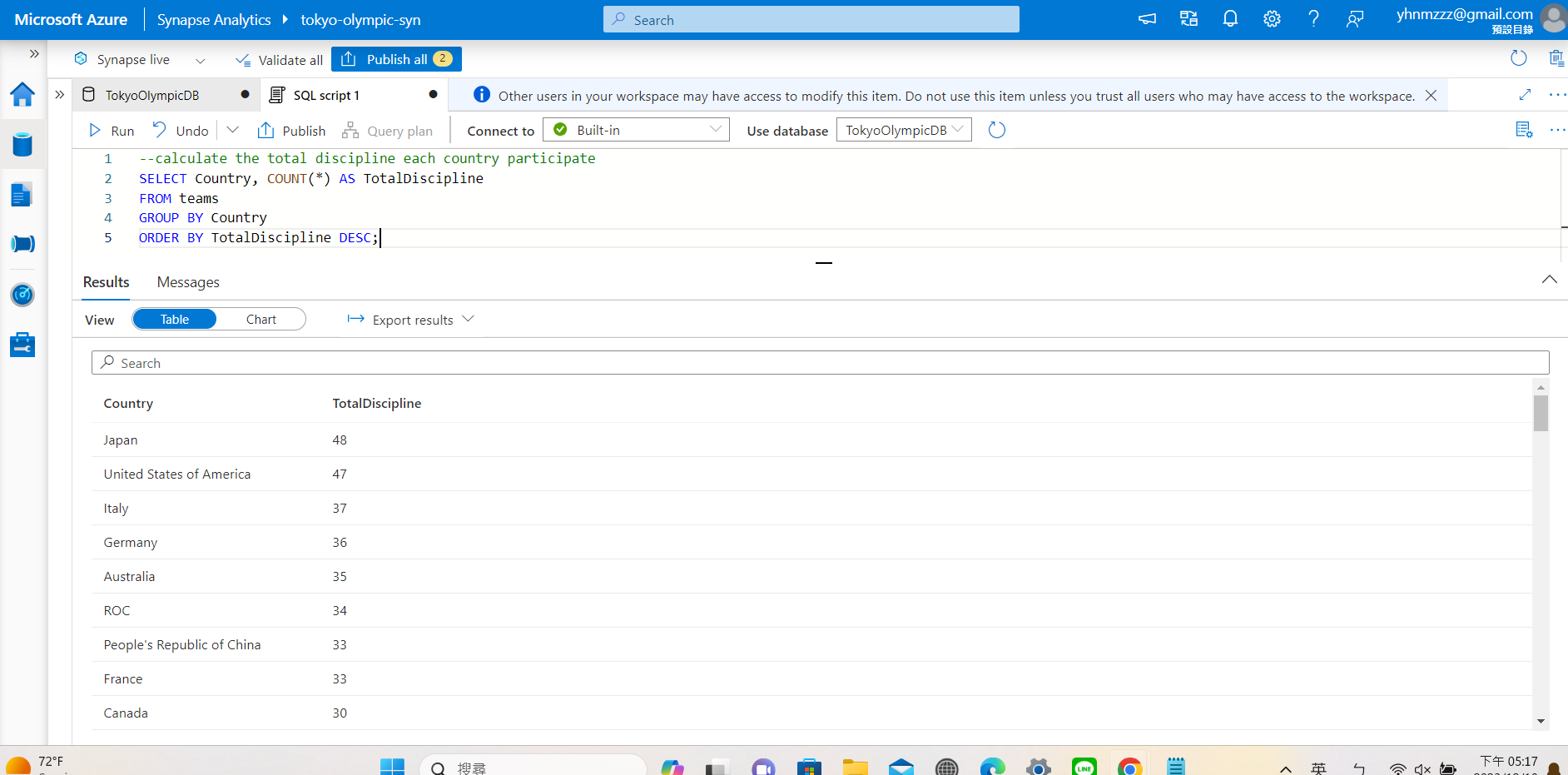


**AZURE SYNAPSE ANALYTICS STEPS:**

1. Create a database in synapse for data analytics and connect the database to the data lake where we stored the dataset.
2. Create each table by loading the dataset from Azure Data Lake Gen 2.
3. Do the analysis using SQL to run different queries to get the data we require.



The query above shows us the number of athletes from each country. From this analysis, we can understand the participation status of each country. Moreover, based on the results we get, the organization may be able to prepare more for the next Olympics.



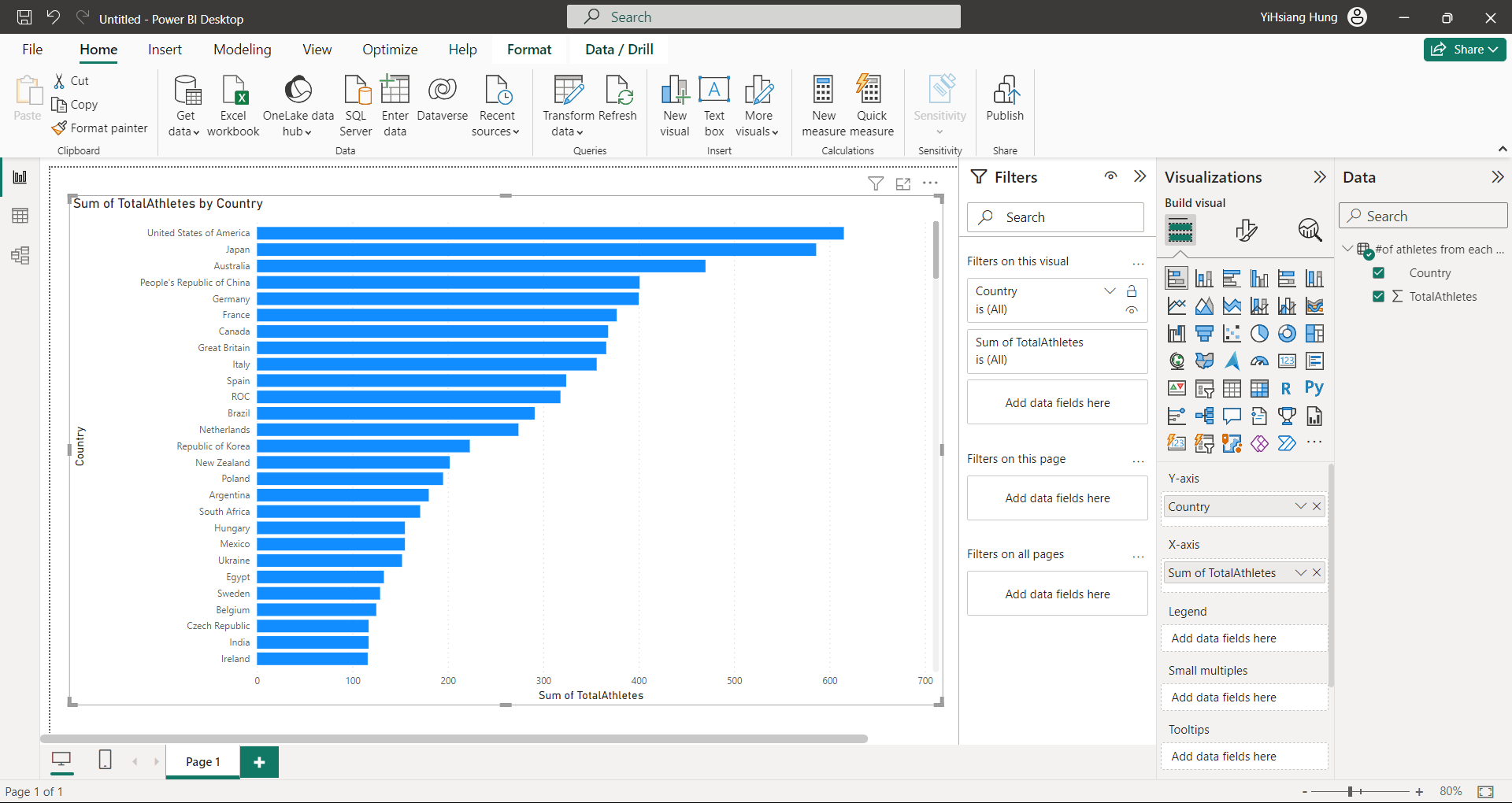
The above query shows the number of disciplines each country participates in. Knowing the result helps us to analyze the reason why some countries are only participating in a few disciplines. For example, if a country only participated in single-digit disciplines, we can investigate deeply to see what disciplines the country participates in and why the country only participates in those disciplines. This helps us to figure out the reason and the organization may make some changes so that the country may participate in more disciplines in the next Olympics.



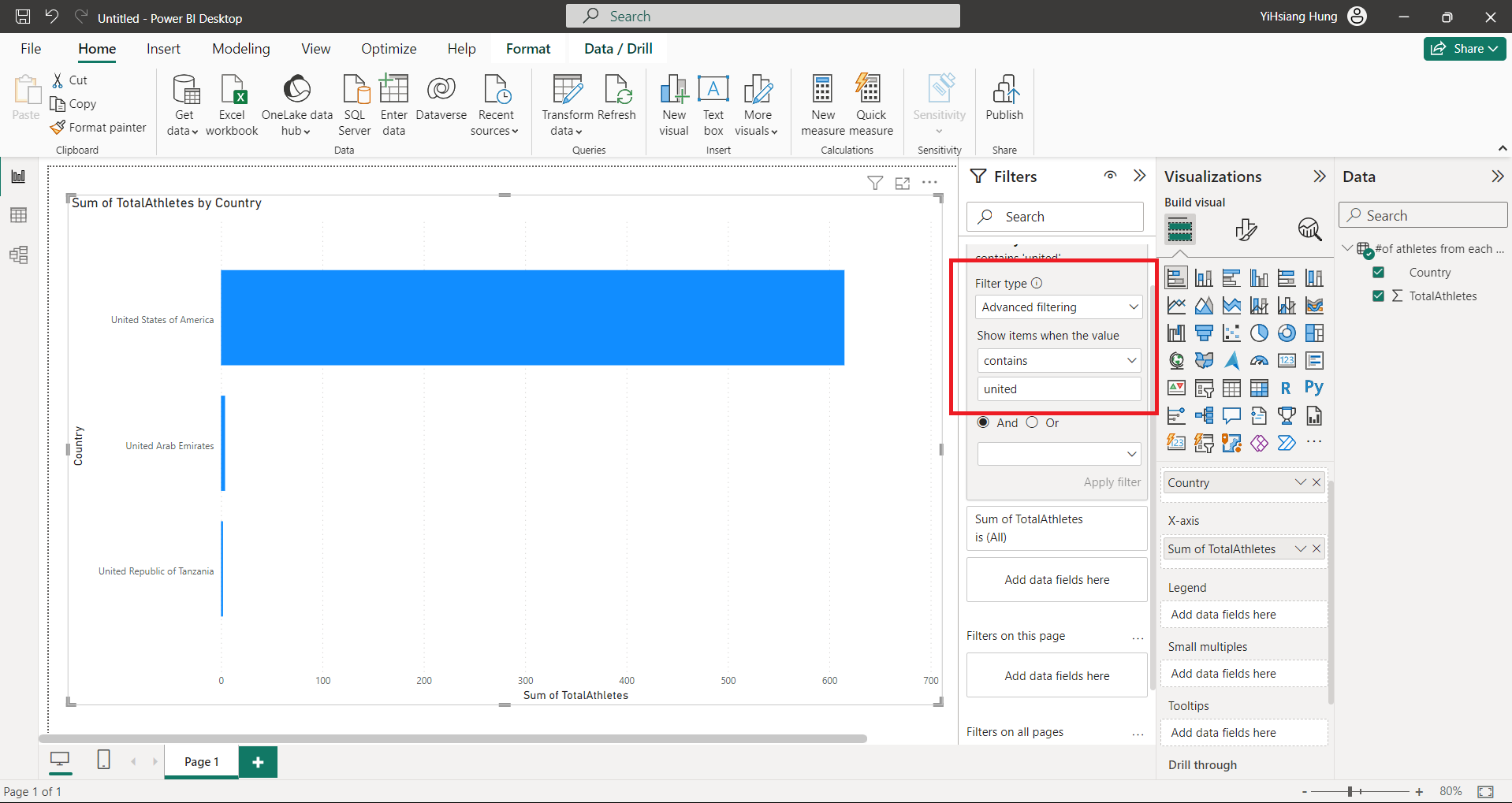
Azure Synapse analytics also provides the data visualization. However, the features are imperfect and there are only a few choices of the graph.

**MICROSOFT POWER BI STEPS:**

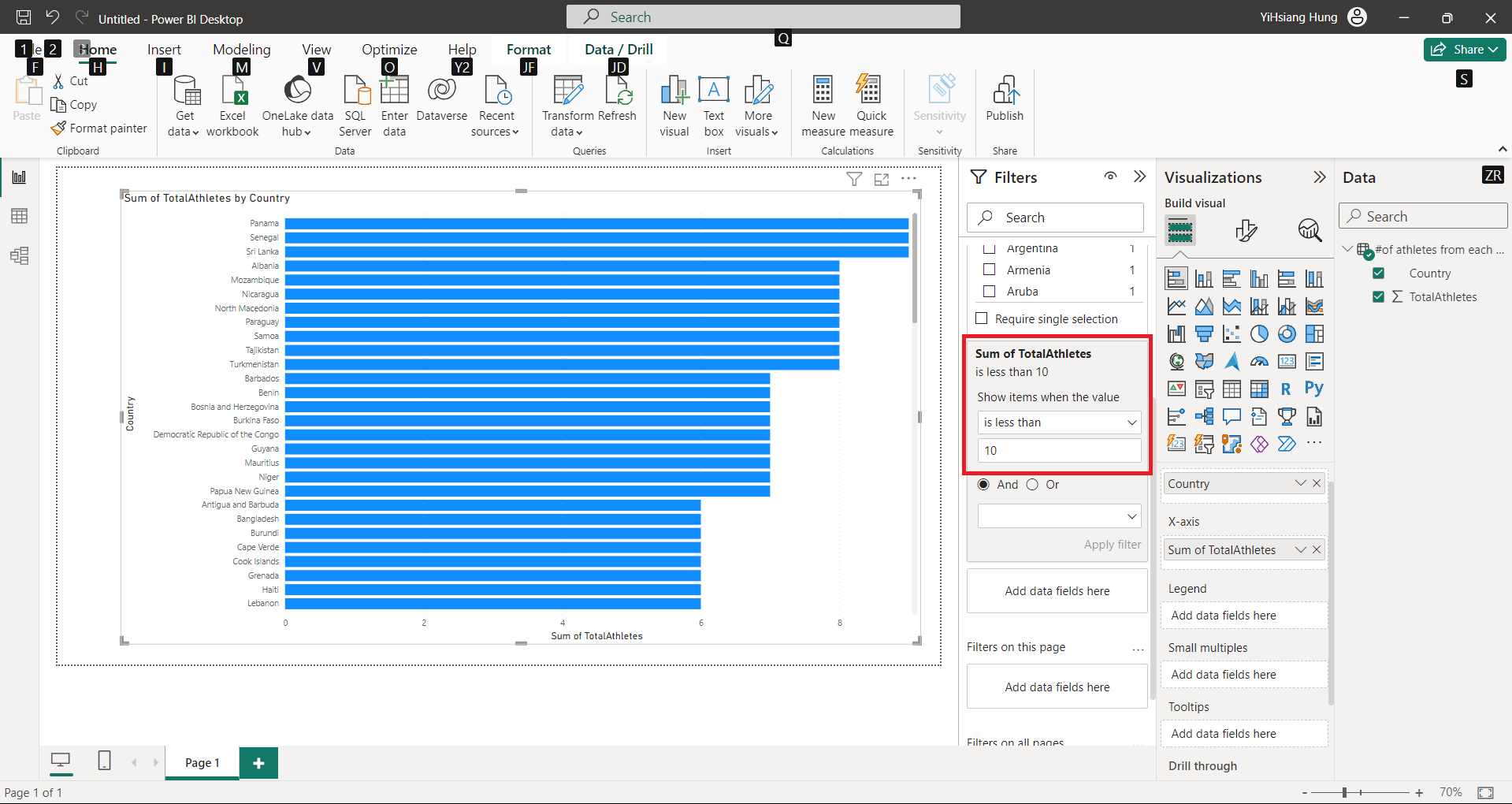
1. In Azure Data Synapse we export the analytics we did into CSV files.
2. Import the CSV files into Microsoft Power BI.
3. Try different kinds of graphs to find the best visualization we want.



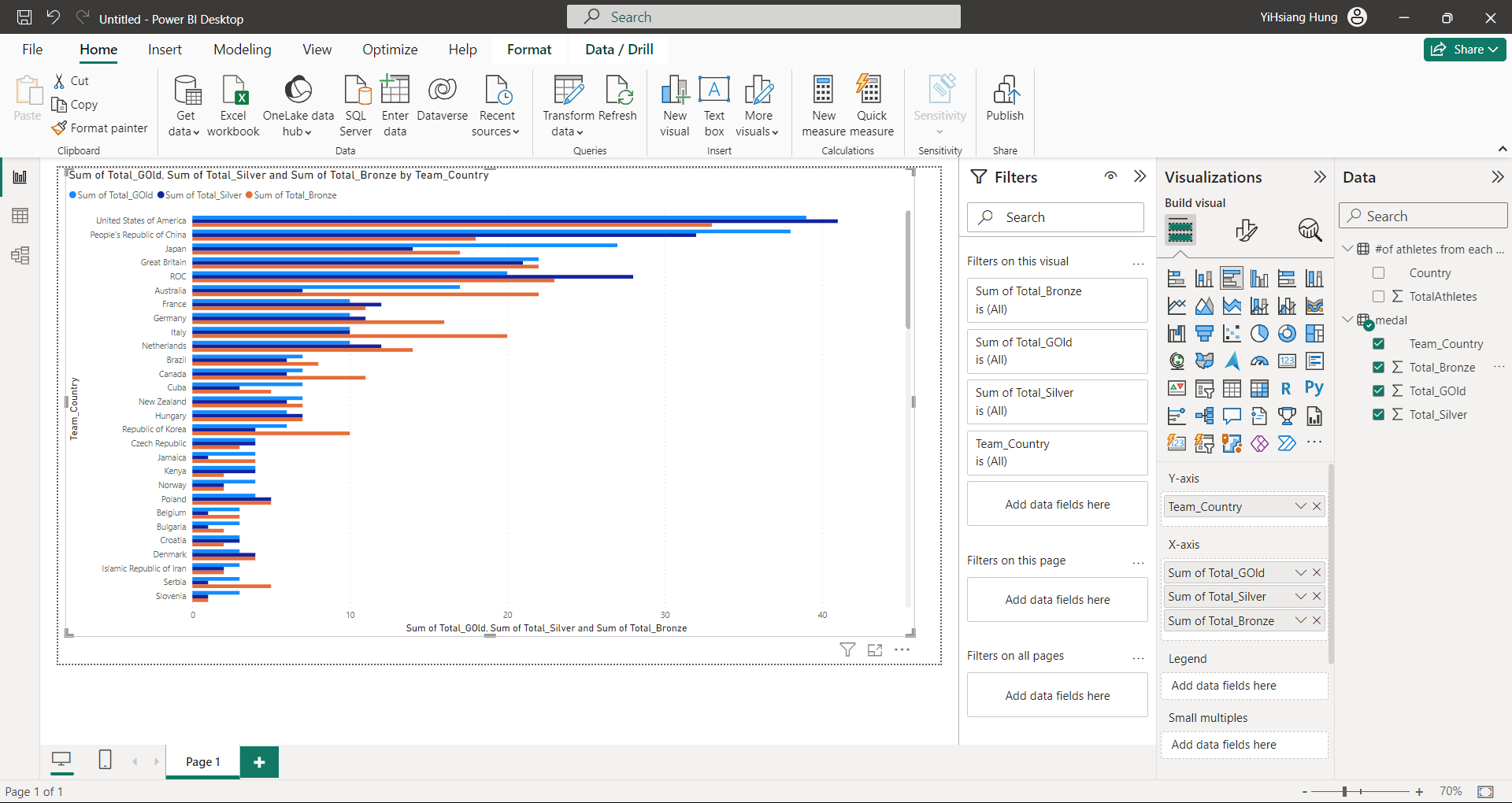
The above graph shows the number of athletes from each country.



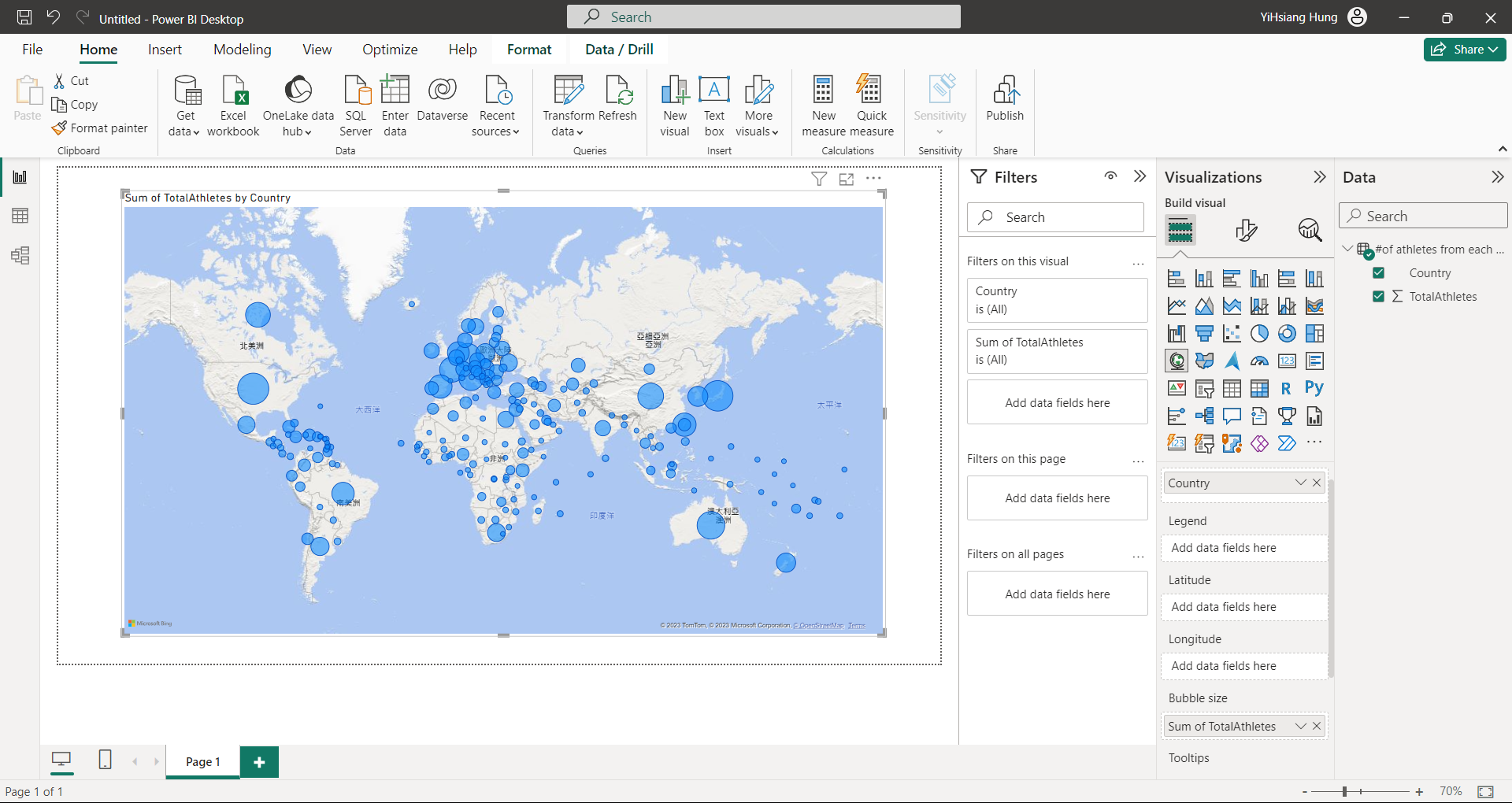
We can also do some advanced filters for the visualization. For example, the above graph shows the visualization only if the country name contains “united”.



The above graph shows us a country where the number of athletes is less than 10.



The above shows us the numbers of each medal which were won by each country. By using this graph, we can also look deeply at a specific country and compare the number of gold, silver, and bronze medals of the country.



This is the graph of the number of athletes from each country. Power BI provides some special graphs that can achieve more visualization effects. In this graph, we can compare the number of athletes from each country by looking at the size of the circle of each country. If we need to investigate deeply, we can switch to another kind of graph.

**REFERENCES:**

**Microsoft tools used:**

<https://learn.microsoft.com/en-us/azure/data-factory/>

<https://learn.microsoft.com/en-us/azure/data-factory/load-azure-data-lake-storage-gen2>

<https://learn.microsoft.com/en-us/azure/databricks/getting-started/>

<https://learn.microsoft.com/en-us/azure/synapse-analytics/overview-what-is>

<https://learn.microsoft.com/en-us/power-bi/>

**Websites used:**

<https://www.kaggle.com/>

<https://github.com/>

**Full Code + Data Repository:**

<https://github.com/prerakpanwar/Data_Engineering_Azure>

**Conclusion:**

In this project, we got the dataset from Kaggle and uploaded it to GitHub. The first step was to create a Resource group where all the services used will be under the same resource group. After this we learned how to create a Storage account where we made 2 folders to store Raw data and Transformed data. Then we load the dataset into Azure Data Factory using data pipeline. Then we stored the raw data in Azure Data Lake Gen 2. Once we are done with storing our raw data we now moved to the data transformation step. For this first we need to connect Azure Data lake to Azure Databricks via Azure app registrations and also provide access to the data to the created application so that I can give that data to Azure Databricks. After Databricks had access to the data using a client ID, tenant ID, and secret key we started cleaning the raw data, removing some incorrect data, modifying some of the data storing types, and changing the names of some columns. After data transformation, we stored the transformed data in Azure Data Lake Gen 2 in a separate folder. Hence, we created a database in Azure Synapse Analytics and loaded the dataset from Azure Data Lake Gen 2 to create tables. We do the SQL queries in Azure Synapse Analytics to get the data we want and do the analysis. Additionally, we had the option to enhance our project by integrating with PowerBI for data visualization and creating visually appealing and informative dashboards based on the analyzed Olympic dataset. So to get a better visualization, we use Microsoft Power BI to show different kinds of graphs of our dataset. Our systematic approach and the integration of Azure services provided a robust framework for conducting in-depth analysis, ensuring the project's success in deriving meaningful insights from the Olympic dataset. The detailed process and findings are also documented in our GitHub repository which is provided above in the references section, providing transparency and traceability to our work.

Through this project, we learned to seamlessly integrate Azure services for end-to-end data engineering, encompassing data extraction, transformation, storage, and analysis. Leveraging tools like Azure Data Factory, Data Lake Gen 2, Databricks, and Synapse Analytics, we successfully processed the Kaggle Olympic dataset, deriving valuable insights. The integration of PowerBI for visualization enhanced our ability to communicate findings effectively.