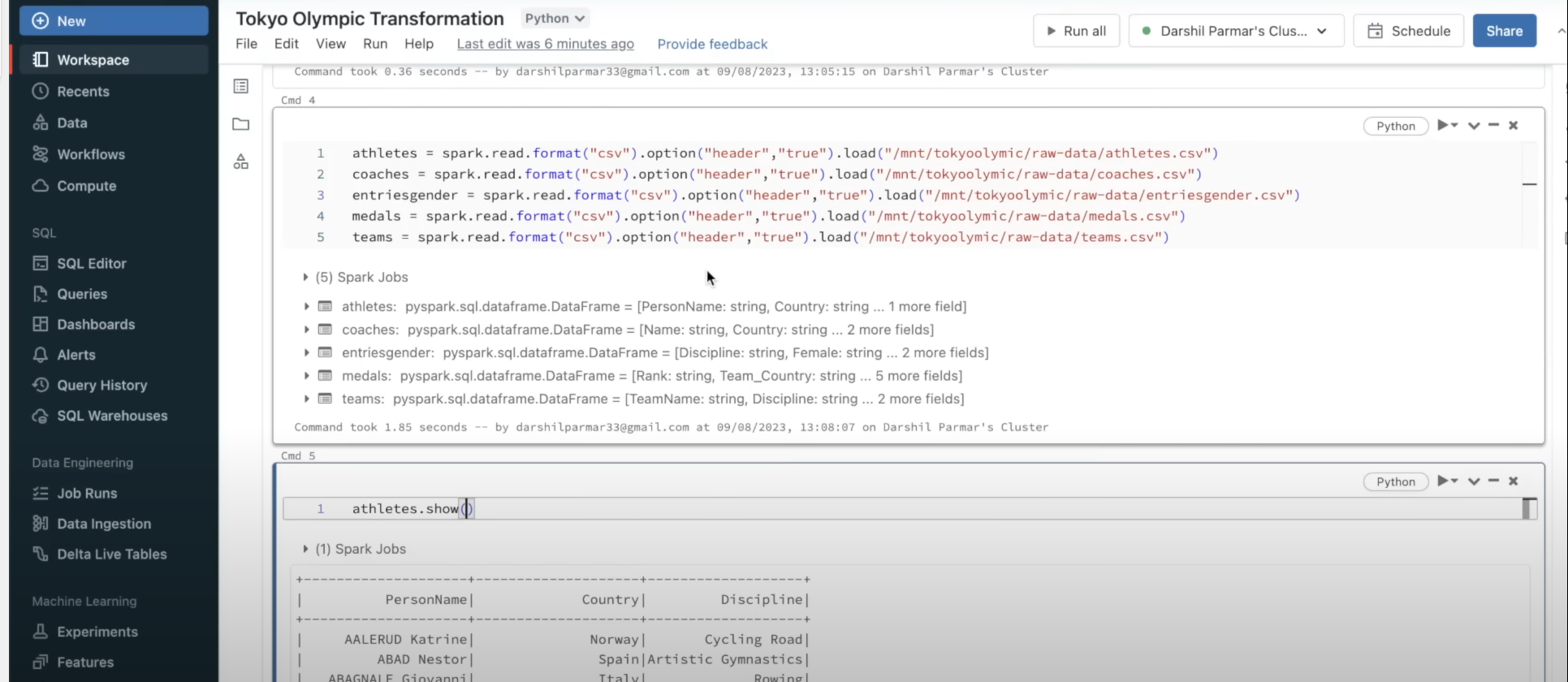
# **Tokyo Olympic Data Analytics | End-To-End Azure Data Engineering Project**

* This project will extract the data from the API using Azure data factory which is kind of like the data pipeline tool available on Azure. It will build a flow like this and load our data onto the Azure data lake storage.

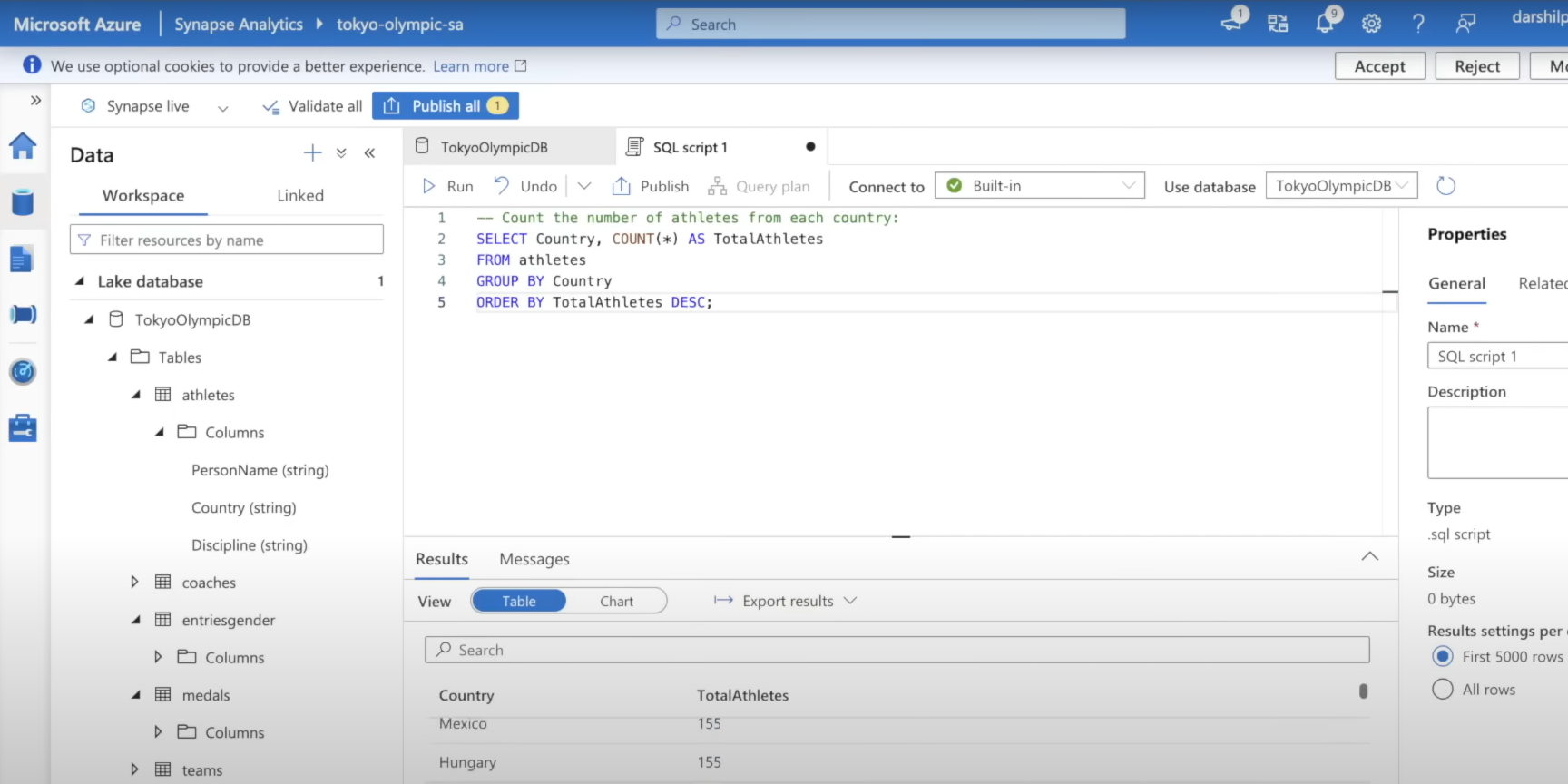
A screenshot of a computer

Description automatically generated

* So first will load the raw data and then use Azure Databricks will write spark code and transform data and load data back to transform data lake storage.



* Once that is done, will use Synapse Analytics to run the SQL queries on top of the transform data so that we can find the insights and get the visualization on top of it.



* So, will execute the entire data engineering project from start to end. From extracting data, doing the transformation, and then loading the data and doing analysis on top of it.



# **Project Details:**

* This project will take the Olympic data that is available on Kaggle, and I will build an end-to-end Azure data engineering project.
* So, will use the major services available on Azure Cloud and try to build a simple data pipeline, will copy the data, do some basic transformation, and then load the data onto some target location. So, this is the overall project.

# **The Architecture Diagram:**

End-To-End Data Engineer Project

* We have a data source then we have some ingestion mechanism, we are storing that data in some location then we are using some transformation code using Apache Spark then again, we are storing this transformation block using the analytics function and then we have the dashboard at the end.

# **Understanding the Architecture Diagram**

* **ETL(extract, transform, load)**: So, in ETL we basically extract data from multiple sources and then apply some transformation logic the transformation logic can be anything such as cleaning the data, removing duplicate values, or null values, or applying any business logic and it then we load that data onto some target location so, this particular architecture diagram is built on top of that using Azure Cloud services.

1. **Data source**: We have a data source that the data that we will use from Kaggle, and we will understand the data and what types of files we have, the columns we have.
2. **Data factory service**: Data factory is an integration service it can build a data pipeline, connect to multiple sources, and do some basic transformations there are a lot of different functionalities available and this is the service available on Azure that can use to build a simple data pipeline, or it can also do some complex work.

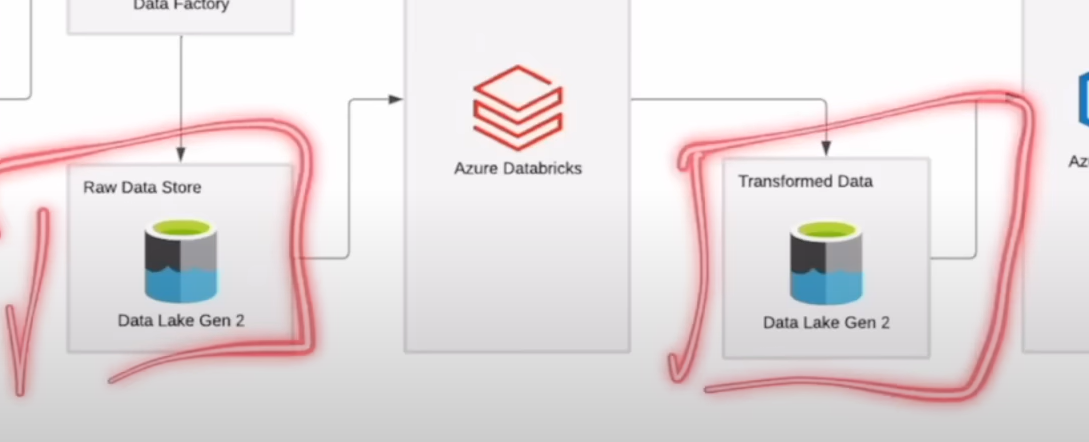
So, we will use Azure data factory to extract data from our data source and then we will load that data onto our data lake storage.

1. **Data lake storage service**: It is basically an object storage where you can store your file. So, on Azure Cloud we have something called a storage account. Inside the storage account you can create the containers where we can store the blog files, or we can also create the tables and there are multiple things available. The overall goal of the data lake is to store the structured and unstructured data onto the storage.

So, what we will do, we will extract the raw data and we will store the raw data as it is on our storage location after that we will want to do some transformations code on top of this.

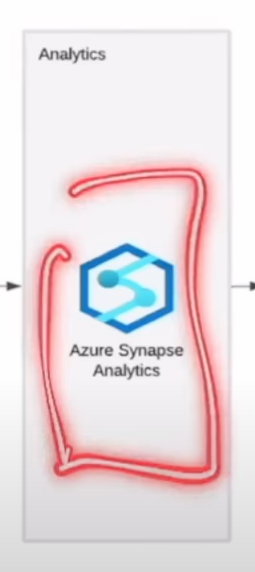
1. **Databricks service**: What we will do here, we will get this raw data, we will write our code in Apache Spark on this Databricks environment and do some basic transformations and again load our data onto the transformed data.

So, after cleaning the data and doing some basic transformations we will again load the data onto the data lake storage. So, we will have the raw data stored and we will have the transformed data stored, both databases get stored in our data lake storage.



1. **Synapse Analytics service**: Once we have all the things we discuss before done then we can use this analytics service called Azure Synapse Analytics.

In Azure Synapse Analytics we can use multiple things we can use Notebook we can use SQL functions (run the analytics query on top of it so if we want to understand who won the highest gold medal by country or which player won the highest gold medal we can do that using the SQL queries also).

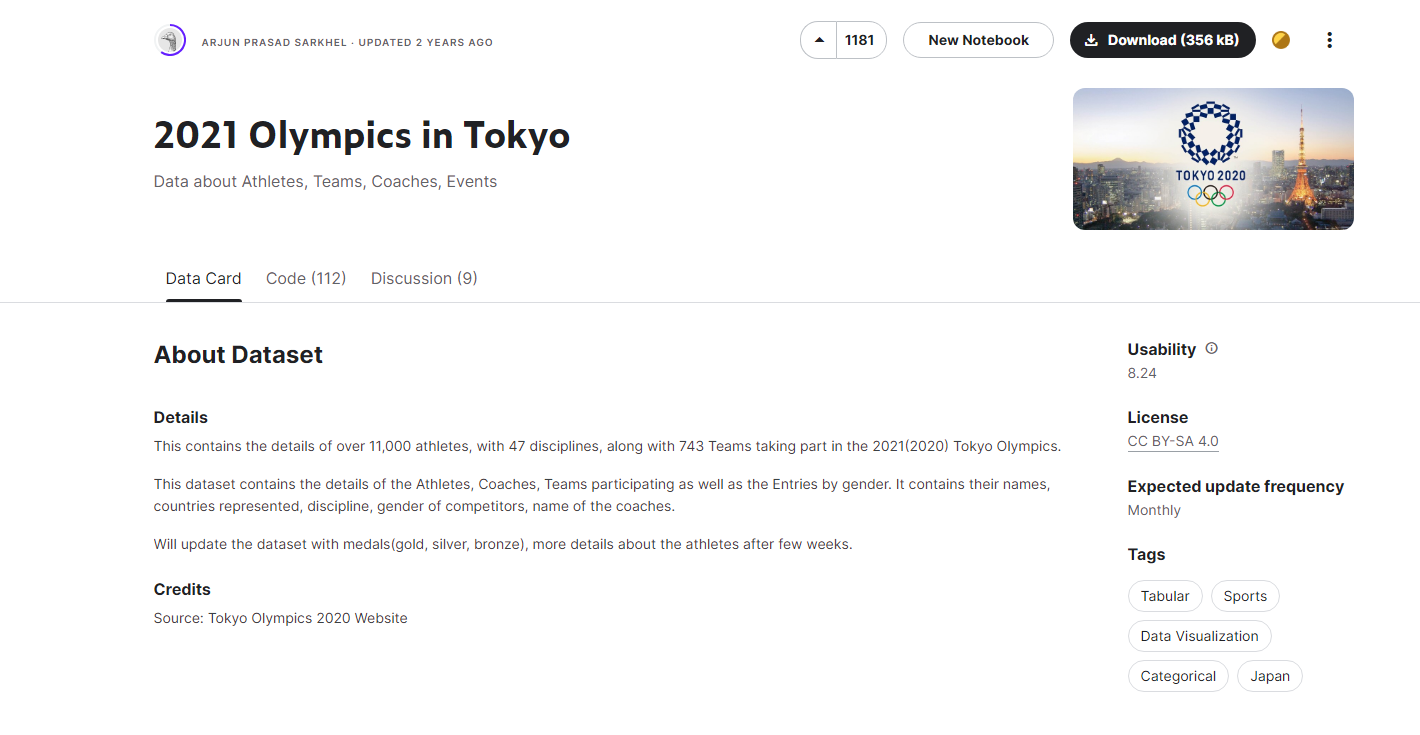


1. **Dashboard**: At the end we can build a dashboard using multiple tools such as Power BI, Looker Studio, Tableau.

A screenshot of a computer

Description automatically generated

# **Understanding each of the services in a little bit of detail and what they do**

**Data Source:** [2021 Olympics in Tokyo](https://www.kaggle.com/datasets/arjunprasadsarkhel/2021-olympics-in-tokyo) (Kaggle)

1. **Understand our data**: This data set contains the information about 11,000 athletes, with 47 disciplines, along with 743 Teams taking part in the 2021(2020) Tokyo Olympics and we have multiple different files available. As you can see below, we have the Excel format files available, but I have converted this into the CSV for the better data processing, so we have the Athletes, Coaches, EntriesGender, Medals, Teams.

**Datasets:**

**Athletes.xlsx (316.28 KB)**

About this file: Contains details about the participating Athletes.

Columns:

1. PersonName: The athlete’s name.
2. Country: The country they are from.

3- Discipline: Such as Basketball, Rowing, Wrestling, and karate all other discipline that are performed in the Olympics.

**Coaches.xlsx (15.01 kB)**

About this file: Contains details about the Coach(Country, Discipline, Event).

Information about the coaches of that athlete.

Columns: Name, Country, Discipline and Event.

**EntriesGender.xlsx (9.7 kB)**

About this file:Entries by Discipline and number of females and males taking part in it.

In this data in each discipline how many were male and how many were female and the total count.

Columns: Discipline, Female, Male and Total.

**Medals.xlsx (6.87 kB)**

About this file:Medals as on 29th July 2021.

Information about Medals so for which country won the number of medals such as Gold, Silver, Bronze and total and rank of it.

The medals contain the information on the country level about the gold, Silver, Bronze and the total base and the rank.

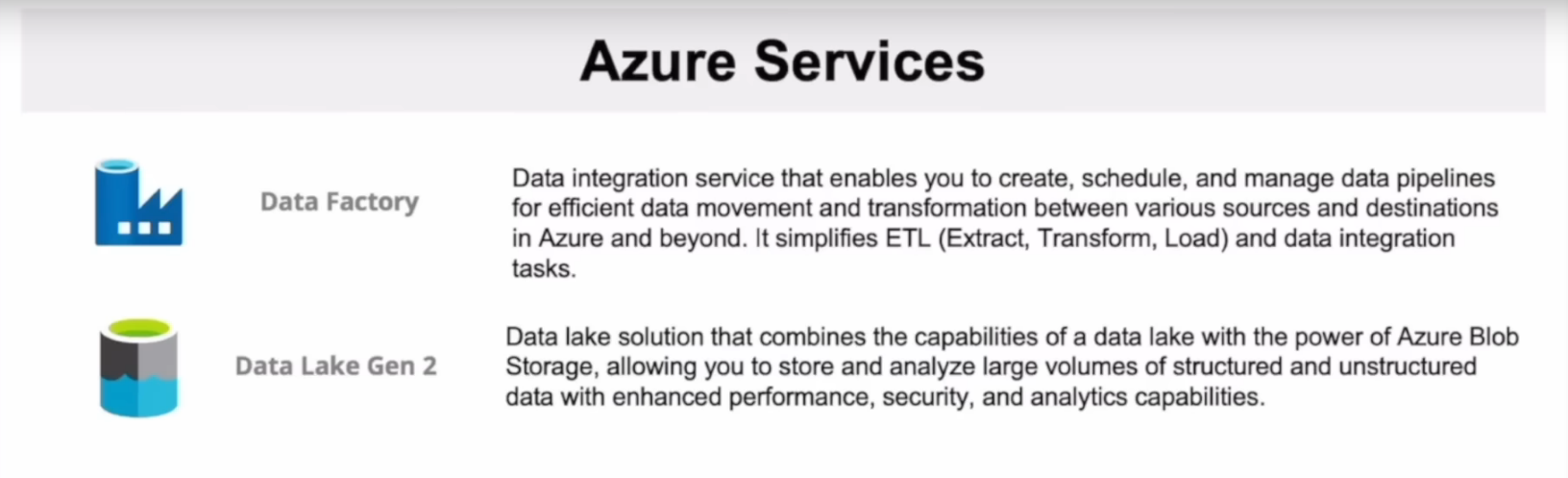
Columns: Rank, Team\_Country, Gold, Silver, Bronze, Total and Rank by Total.

**Teams.xlsx (24.98 kB)**

About this file:Contains the details of all the Teams(Country, event, Discipline, Event).

The Teams contain the team name, in that discipline and the country name, with the event.

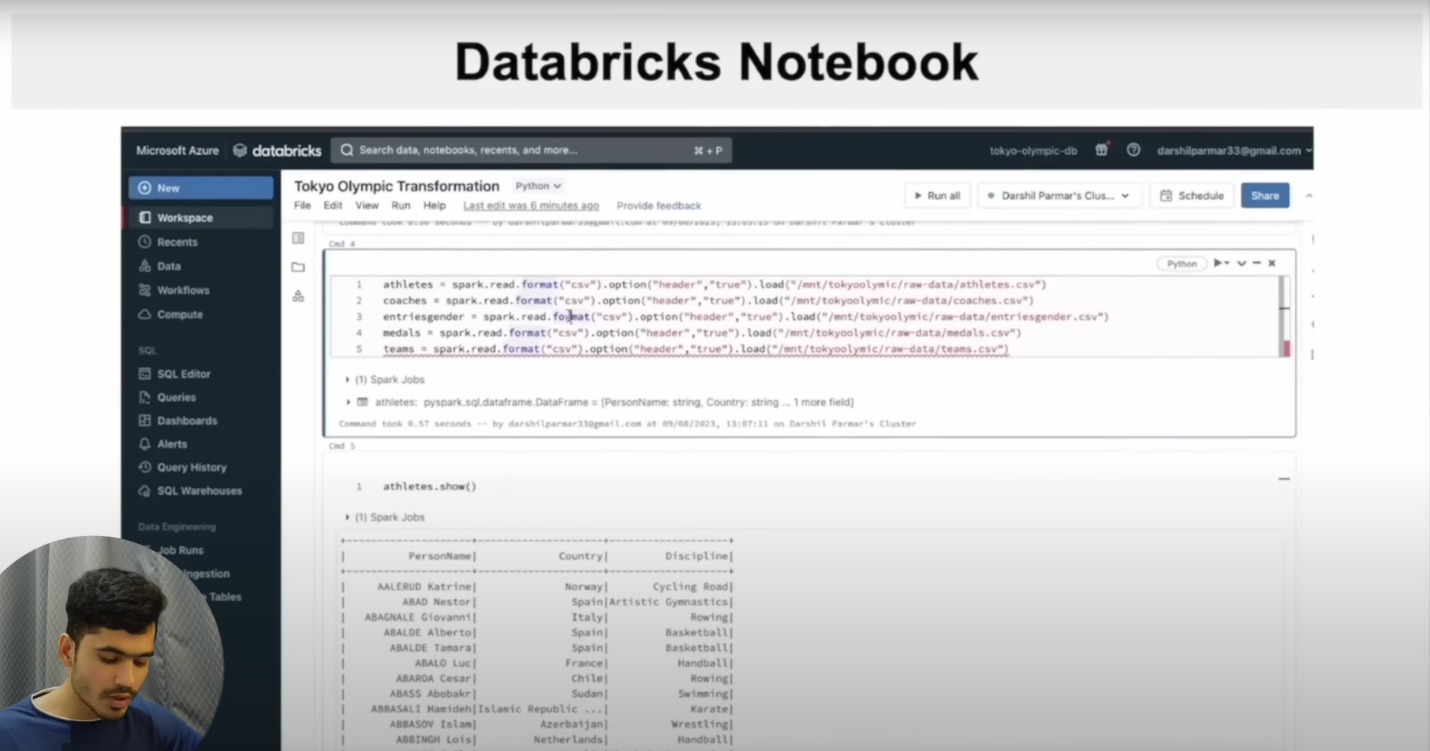
1. A close-up of a text

   Description automatically generated**Azure Services****:**
2. **Data Factory UI**

**A blue factory logo

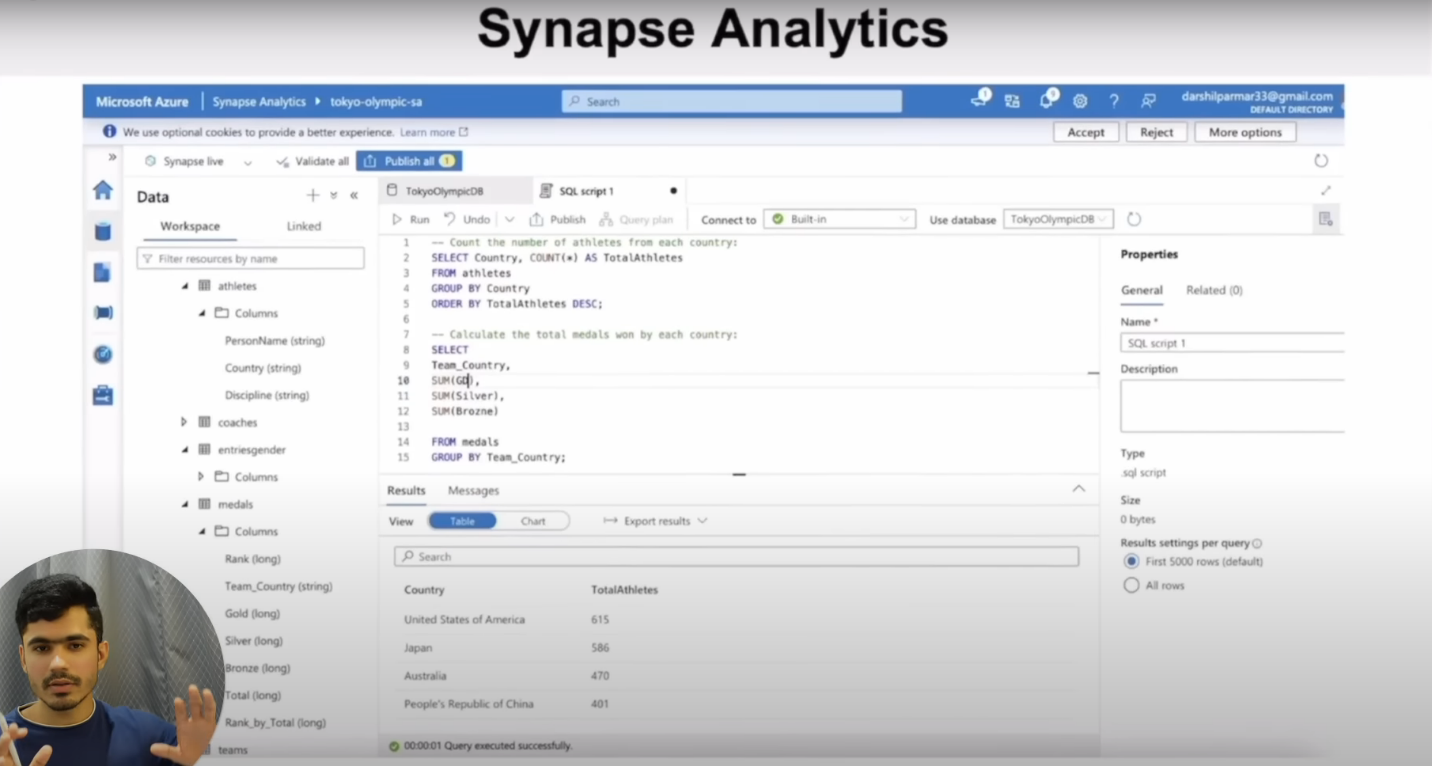
Description automatically generated**

1. Databricks UI





1. Synapse Analytics UI



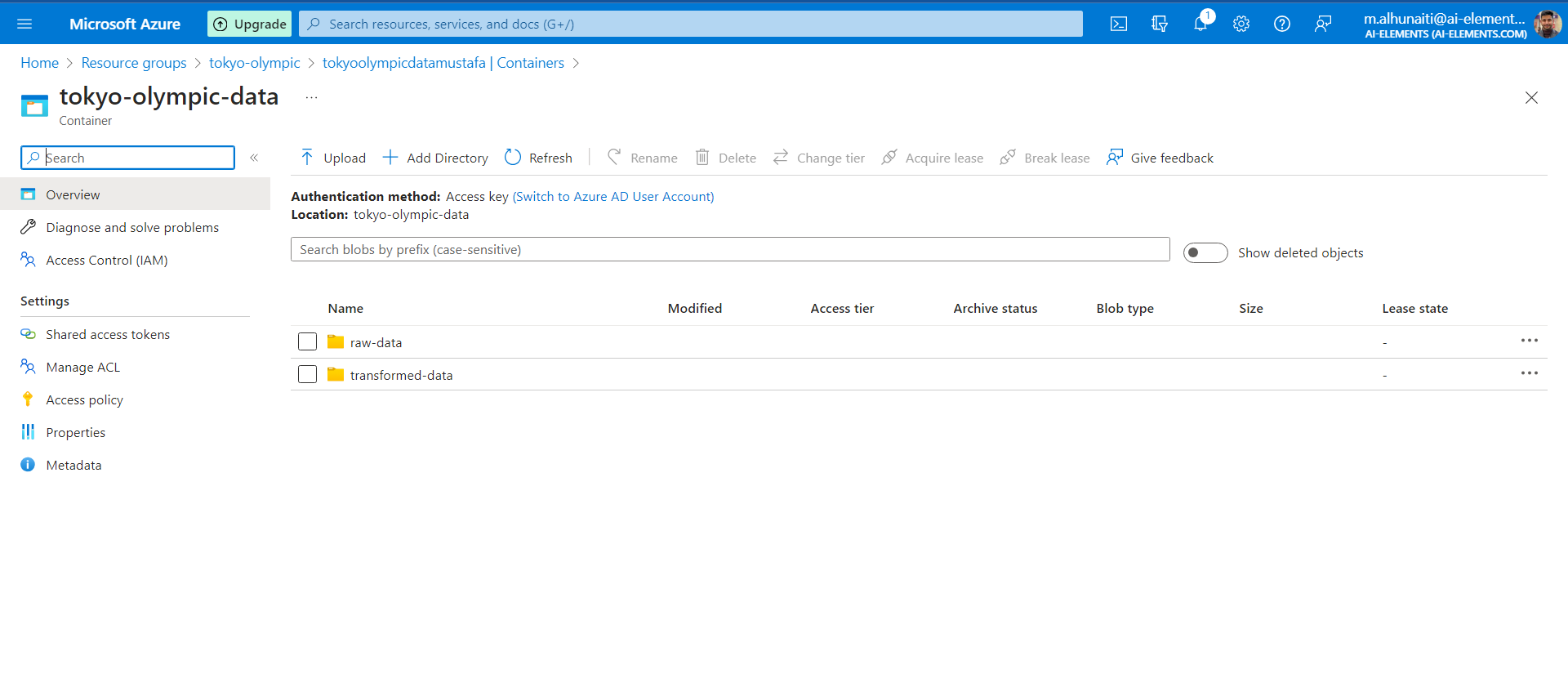
A blue hexagon with blue dots and a blue circle

Description automatically generated

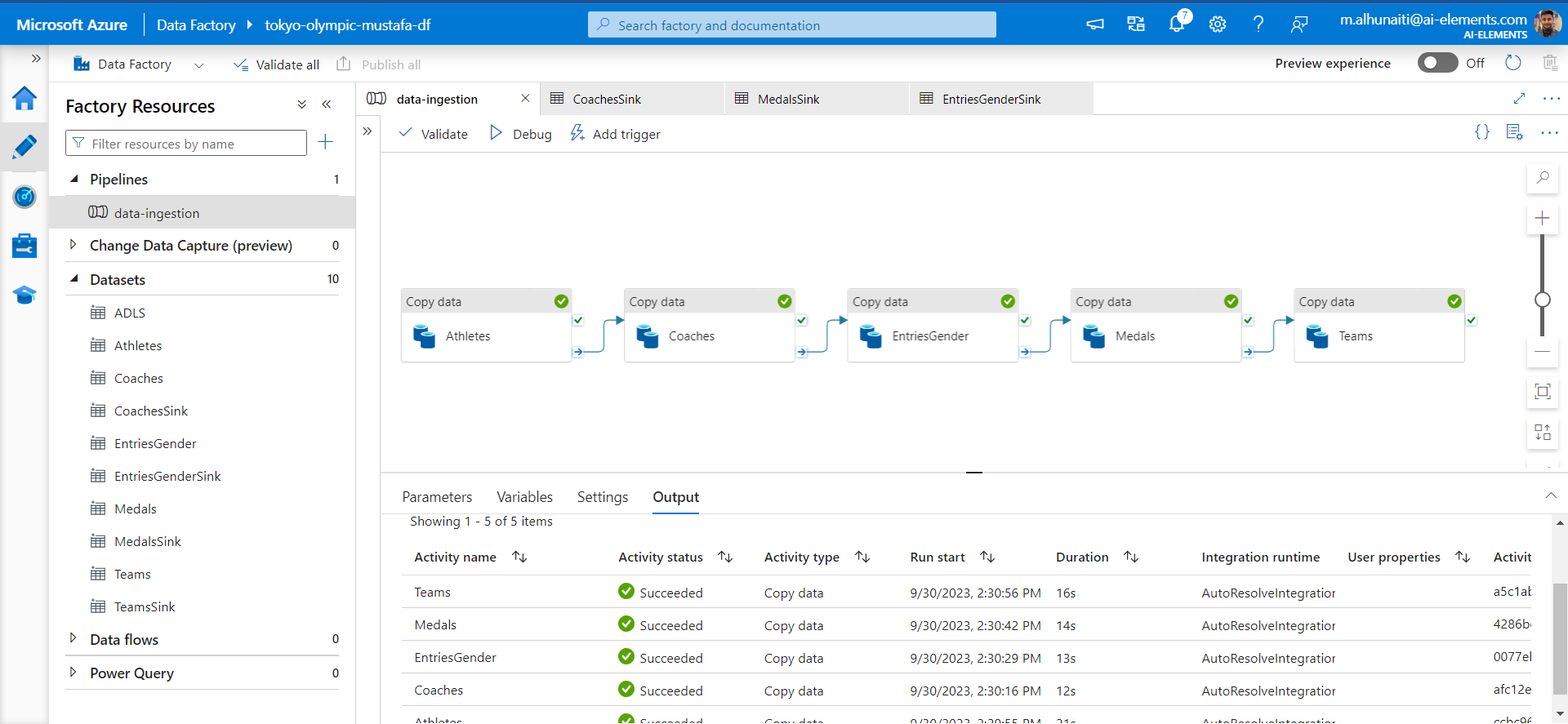
# **Snapshots Project**

**A screenshot of a computer

Description automatically generatedResource Group**

**Azure Data Lake Storage Gen2**

**Azure Data Factory**



**A number and numbers on a white background

Description automatically generated**A screenshot of a computer

Description automatically generated**Azure Databricks**

Secretkey: dtyxxxxxxxxxxxxxxxxxxxxxx

eaxxxxxxxxxxxxxxxxxxx

ecxxxxxxxxxxxxxxxxxxx



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

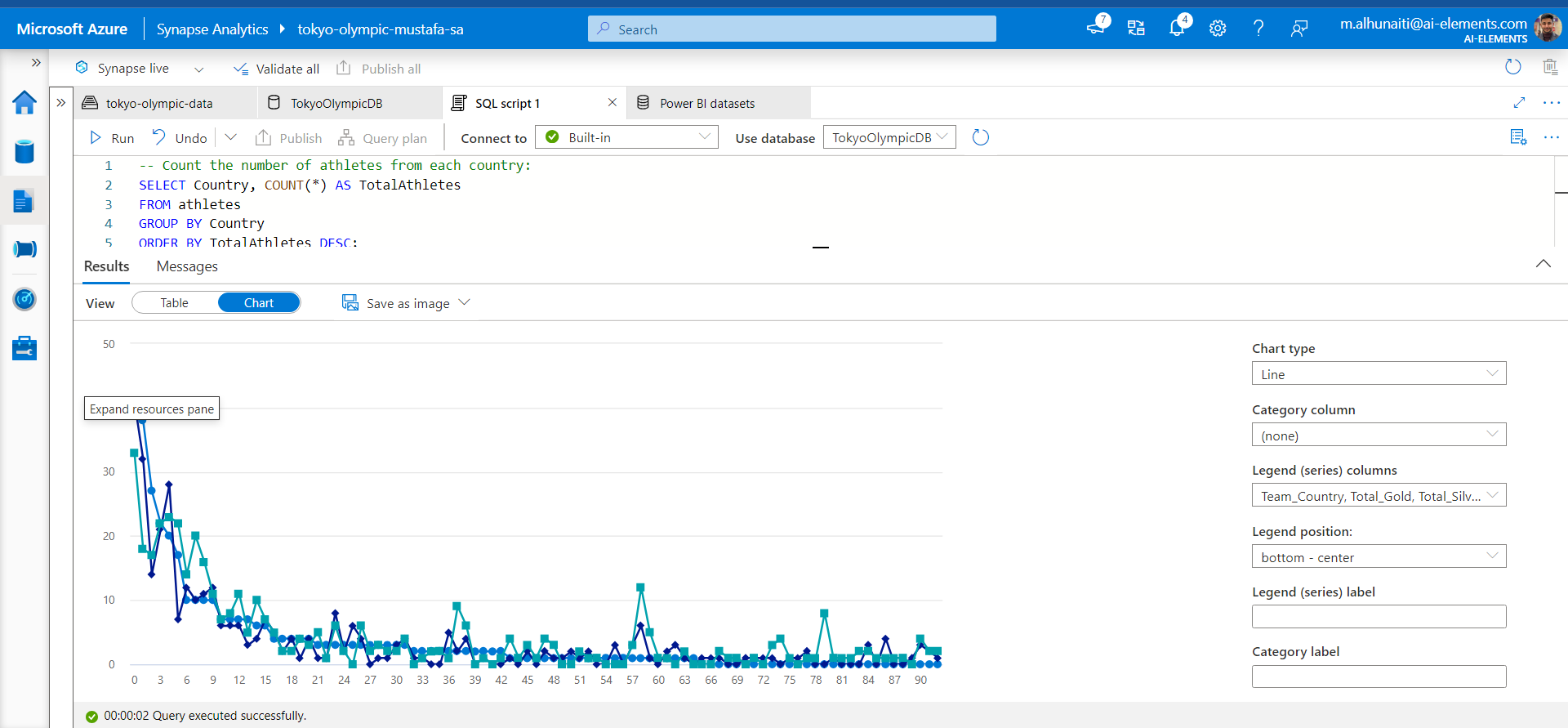
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Azure Synapse Analytics**



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated