

Cloud Olympics: Transforming Insights in Olympic Data with Azure Cloud.

TEAM MEMBERS PICTURES:



Deepika Bezwada
16347826



Naga Sahithi Vunnam
16355633



Greeshmitha Reddy Veeradimmu
16355034



Sai Harshith Nimmalagottu
16352577

Names:

- 1) Deepika, Bezwada (Student ID: 16347826, Section-1, 11:30am Class)
- 2) Greeshmitha Reddy, Veeradimmu (Student ID: 16355034, Section-2, 1:00 PM Class)
- 3) Naga Sahithi, Vunnam (Student ID: 16355633, Section-2, 1:00 PM Class)
- 4) Sai Harshith, Nimmalagottu (Student ID: 16352577, Section-2, 1:00 PM Class)

Roles:

- Greeshmitha is responsible for data ingestion task. She will set up data pipelines to extract and load data from various sources into Azure using Azure Data Factory. She is involved in collecting the data from various data sources and then loads that raw data into the data lake.
- Deepika is responsible for Transformation tasks. She is involved in preparing and cleaning the raw data. Uses data bricks to clean the data. This involves handling missing values, removing duplicates, and addressing data quality issues. And then perform wide range of data transformation operation that include filtering. Finally make the raw data suitable for analysis.
- Sahithi is responsible for data analysis using Azure Synapse Analytics. Using Azure Synapse Analytics, she writes SQL Queries to extract insights from the cleaned and transformed data. She is responsible for performing analysis of the data. It includes querying for medal counts and more.
- Harshith is responsible for creating dashboards using power BI or any other tool. He will use this tool to create graphs, charts, and tables to provide some insights from Olympic data.

Motivation and Purpose:

- The Olympic Data Analysis project on Azure is motivated by the pursuit of informed decision making in sports. By harnessing Azure's advanced analytics and cloud capabilities, the project seeks to uncover nuanced patterns and insights within the vast sea of Olympic data.
- The aim is to empower athletes, coaches, and sports enthusiasts with data-driven strategies, enhance training methodologies, and enable precise performance predictions.
- This endeavor not only advances the field of sports analytics but also enriches the overall Olympic experience, fostering a new era of intelligent sports management and athlete development.
- With the help of Azure's data services and cloud computing, this study aims to show how to effectively extract insightful information from past Olympic data. By combining Azure Databricks, Azure Data Factory, and other Azure services, the project provides a scalable and efficient method for handling, converting, and evaluating large-scale Olympic datasets.

Cloud Technologies/Services Used:

- Azure Databricks
- Azure Data Factory
- Azure Data Lake Gen2
- Azure Synapse Analytics

System Architecture:

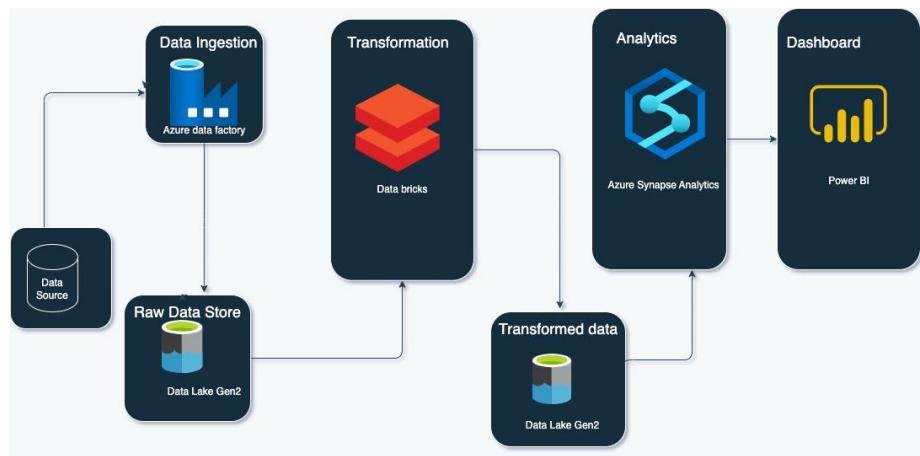


Fig: Olympic data analysis architecture

The project's architecture is made up of the following parts:

- **Azure Databricks:** For data analysis, transformation, and processing. It offers an interactive and cooperative workspace for executing Spark-based tasks.
- **Azure Data Factory:** Coordinates and oversees the data processing process. It is in charge of job scheduling, data transformation, and data ingestion from several sources.
- **Azure Storage:** Acts as the raw and processed data lake. Additionally, it is capable of housing interim findings produced throughout the examination.
- **Azure SQL database:** The data that has been cleaned and transformed is stored in an Azure SQL database, which enables access for reporting and visualization.
- Using an Azure SQL database connection, Power BI allows users to construct dynamic, eye-catching dashboards for data analysis.

Features (with screenshots):

• Data Collection:

We have collected the 2021 Olympic dataset from [Kaggle](#).

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, Teams participating and the Entries by gender. It contains their names, countries represented, discipline, gender of competitors, name of the coaches.

This dataset has the following csv files. It contains details about participating Athletes, their country, medals, sport, events, gender, and rankings.

	A	B	C	D	E	F	G
1	Name	NOC	Discipline				
2	AALERUD Norway		Cycling Road				
3	ABAD Nest Spain		Artistic Gymnastics				
4	ABAGNALI Italy		Rowing				
5	ABALDE Al Spain		Basketball				
6	ABALDE T Spain		Basketball				
7	ABALO Luc France		Handball				
8	ABAROA C Chile		Rowing				
9	ABASS Abd Sudan		Swimming				
10	ABBASALI Islamic Re	Karate					
11	ABBASOV Azerbaijan		Wrestling				
12	ABBINGH Netherlan	I	Handball				
13	ABBOT Em Australia		Rhythmic Gymnastics				
14	ABBOTT M United Sta		Baseball/Softball				
15	ABDALLA I Qatar		Athletics				
16	ABDALLA I Egypt		Artistic Swimming				

Fig 1: Athletes.csv

	A	B	C	D	E	F
1	Discipline	Female	Male	Total		
2	3x3 Basket	32	32	64		
3	Archery	64	64	128		
4	Artistic Gy	98	98	196		
5	Artistic Sw	105	0	105		
6	Athletics	969	1072	2041		
7	Badminton	86	87	173		
8	Baseball/S	90	144	234		
9	Basketball	144	144	288		
10	Beach Voll	48	48	96		
11	Boxing	102	187	289		
12	Canoe Slal	41	41	82		
13	Canoe Spr	123	126	249		
14	Cycling BM	10	9	19		
15	Cycling BM	24	24	48		
16	DSHEHRI Saudi Arab	Football	Men			

Fig 2: Coaches.csv

	A	B	C	D	E	F
1	Discipline	Female	Male	Total		
2	3x3 Basket	32	32	64		
3	Archery	64	64	128		
4	Artistic Gy	98	98	196		
5	Artistic Sw	105	0	105		
6	Athletics	969	1072	2041		
7	Badminton	86	87	173		
8	Baseball/S	90	144	234		
9	Basketball	144	144	288		
10	Beach Voll	48	48	96		
11	Boxing	102	187	289		
12	Canoe Slal	41	41	82		
13	Canoe Spr	123	126	249		
14	Cycling BM	10	9	19		
15	Cycling BM	24	24	48		
16	DSHEHRI Saudi Arab	Football	Men			

Fig 3: EntriesGender.csv

	A	B	C	D	E	F	G
1	Name	Discipline	NOC	Event			
2	Belgium	3x3 Basket	Belgium	Men			
3	China	3x3 Basket	People's R	Men			
4	China	3x3 Basket	People's R	Women			
5	France	3x3 Basket	France	Women			
6	Italy	3x3 Basket	Italy	Women			
7	Japan	3x3 Basket	Japan	Men			
8	Japan	3x3 Basket	Japan	Women			
9	Latvia	3x3 Basket	Latvia	Men			
10	Mongolia	3x3 Basket	Mongolia	Women			
11	Netherlands	3x3 Basket	Netherlands	Men			
12	Poland	3x3 Basket	Poland	Men			
13	ROC	3x3 Basket	ROC	Men			
14	ROC	3x3 Basket	ROC	Women			
15	Romania	3x3 Basket	Romania	Women			

Fig 4: Teams.csv

	A	B	C	D	E	F	G	H
1	Rank	Team/NOC	Gold	Silver	Bronze	Total	Rank by Total	
2	1	United Sta	39	41	33	113	1	
3	2	People's R	38	32	18	88	2	
4	3	Japan	27	14	17	58	5	
5	4	Great Brita	22	21	22	65	4	
6	5	ROC	20	28	23	71	3	
7	6	Australia	17	7	22	46	6	
8	7	Netherland	10	12	14	36	9	
9	8	France	10	12	11	33	10	
10	9	Germany	10	11	16	37	8	
11	10	Italy	10	10	20	40	7	
12	11	Canada	7	6	11	24	11	
13	12	Brazil	7	6	8	21	12	
14	13	New Zeala	7	6	7	20	13	
15	14	Cuba	7	3	5	15	18	
16	15	Hungary	6	7	7	20	13	
17	16	Republic o	6	4	10	20	13	

Fig 5: Medals.csv

• Data Ingestion:

We need to ingest data from external data source onto azure. To do so we used azure data factory to set up a pipeline and extract data from an external data source and load that data onto azure data lake storage.

Here's a detailed breakdown of the steps involved in creating a data factory, pipeline, and extracting data from an external data source:

1. Created a storage account.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes tabs for 'New Tab', 'New Tab', and 'olympicdatacloudproject3_1702001944806'. Below the navigation bar, the main content area displays the 'Deployment' blade for the project. The title is 'olympicdatacloudproject3_1702001944806 | Overview'. The status message 'Your deployment is complete' is prominently displayed. Deployment details show the name 'olympicdatacloudproject3_1702001944806', subscription 'Azure subscription 1', and resource group 'CloudProject'. The start time is listed as 07/12/2023, 20:19:06. A correlation ID is also provided. On the right side of the blade, there are several promotional cards: 'Cost Management' (Get notified to stay within your budget and prevent unexpected charges on your bill. Set up cost alerts >), 'Microsoft Defender for Cloud' (Secure your apps and infrastructure. Go to Microsoft Defender for Cloud >), 'Free Microsoft tutorials' (Start learning today >), and 'Work with an expert' (Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support. Find an Azure expert >).

2. Created a container.

The screenshot shows the Microsoft Azure portal interface for a storage account named 'olympicdatacloudproject3'. The left sidebar includes links for Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, and Storage browser. Under 'Data storage', 'Containers' is selected, showing a list of existing containers. The 'Containers' table has columns for Name, Last modified, Anonymous access level, and Lease state. Two containers are listed:

Name	Last modified	Anonymous access level	Lease state
Logs	07/12/2023, 20:19:30	Private	Available
olympicdata-cloudproject3	07/12/2023, 20:26:25	Private	Available

Inside the container we created two folders one is for raw data this where we store all our raw data as it is from the data source that we extract. Next folder is for transformed data.

The screenshot shows the Microsoft Azure portal interface for a specific container named 'olympicdata-cloudproject3'. The left sidebar includes links for Overview, Diagnose and solve problems, Access Control (IAM), and Settings (Shared access tokens, Manage ACL, Access policy, Properties, Metadata). The main area shows authentication method as 'Access key' and location as 'olympicdata-cloudproject3'. The 'Blobs' section displays a list of blobs with columns for Name, Modified, Access tier, Archive status, Blob type, Size, and Lease state. Two blobs are listed:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
rawdata	-	-	-	-	-	---
transformeddata	-	-	-	-	-	---

A success message box is visible in the top right corner: 'Successfully added directory' and 'Successfully added directory 'transformeddata''. The URL in the address bar is: https://portal.azure.com/#view/Microsoft_Azure_Storage/ContainerMenuBlade/~/overview/storageAccountId/%2Fsubscriptions%2F674104f0-dfe7-4f1d-9d8f-511e0cd4cbe5/resourceGroups/CloudProject/providers/Microsoft.Storage/storageAccounts/olympicdatacloudproject3.

Created an azure data factory...

The screenshot shows the Microsoft Azure Data Factory - Overview page. The deployment status is marked as "complete" with a green checkmark. Deployment details include:

- Deployment name: Microsoft.DataFactory-20231207204204
- Subscription: Azure subscription 1
- Resource group: CloudProject
- Start time: 12/7/2023, 8:46:25 PM
- Correlation ID: a47cfafc-ad75-47cb-b949-663a0c030473

Next steps include "Go to resource" and "Give feedback". A sidebar on the right provides links to Cost management, Microsoft Defender for Cloud, Free Microsoft tutorials, and Work with an expert.

The screenshot shows the Azure Data Factory Studio interface. The main page displays the Data Factory overview, including its name (olympicdatacloudprojectdatafactory3), type (Data factory (V2)), and essential details like Resource group, Status, Location, and Subscription information. A large blue icon of a factory building is prominently displayed. Below the main area, there are sections for Monitoring (Alerts, Metrics, Diagnostic settings, Logs) and Automation (PipelineRuns, ActivityRuns). A "Launch studio" button is located at the top right of the monitoring section.

Now the data factory is ready. This is where we create the pipeline to extract the data from data source and upload that data onto their target location.

Next, Ingested the downloaded dataset into the GitHub repository. Used this GitHub repository as our data source and extracted the raw formats (using raw data URL) using the azure data factory tool.

Click on Athletes.csv file and click on raw button.

The screenshot shows a GitHub repository named "CloudProject". In the "Files" section, the "Athletes.csv" file is selected. The file content is displayed in a table:

	PersonName	Country	Discipline
1	AALERUD Katrine	Norway	Cycling Road
2	ABAD Nestor	Spain	Artistic Gymnastics
3	ABAGNALE Giovanni	Italy	Rowing
4	ABALDE Alberto	Spain	Basketball
5	ABALDE Tamara	Spain	Basketball
6	ABALO Luc	France	Handball
7	ABAROA Cesar	Chile	Rowing
8	ABASS Abobakr	Sudan	Swimming
10	ABBASALI Hamideh	Islamic Republic of Iran	Karate
11	ABBASOV Islam	Azerbaijan	Wrestling
12	ABBINGH Lois	Netherlands	Handball
13	ABBOTT Emily	Australia	Rhythmic Gymnastics

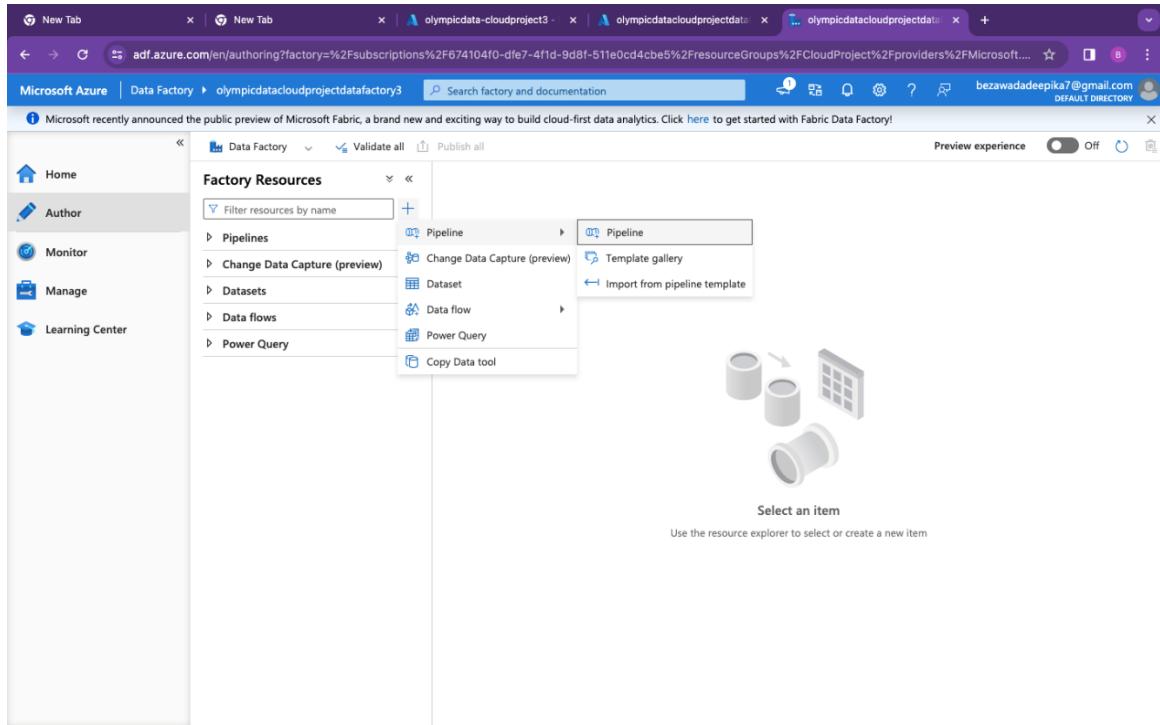
At the bottom of the page, the URL <https://github.com/BezawadaDeepika/CloudProject/raw/main/Athletes.csv> is shown, preceded by "DTT Monica".

It redirects us to a page. That page contains URL which is the raw URL of this CSV file. And connected it to a source.

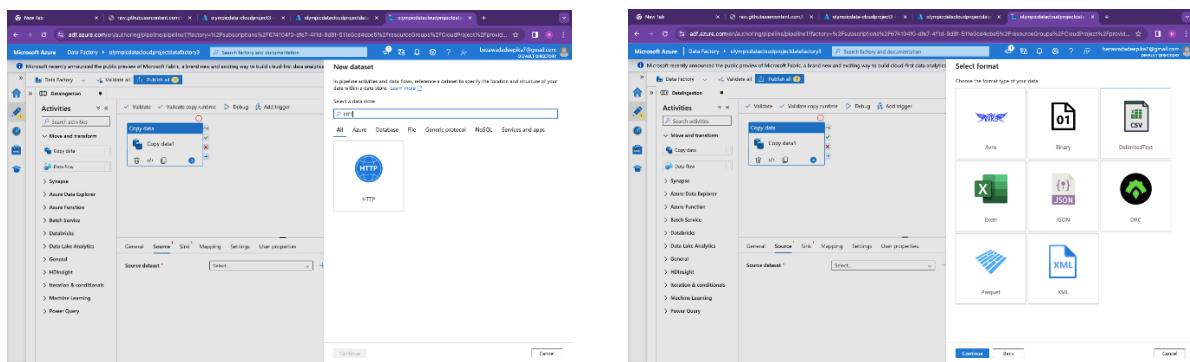
The screenshot shows a browser window displaying the raw content of the Athletes.csv file. The content is a single, long line of text separated by commas, representing the CSV data from the previous screenshot.

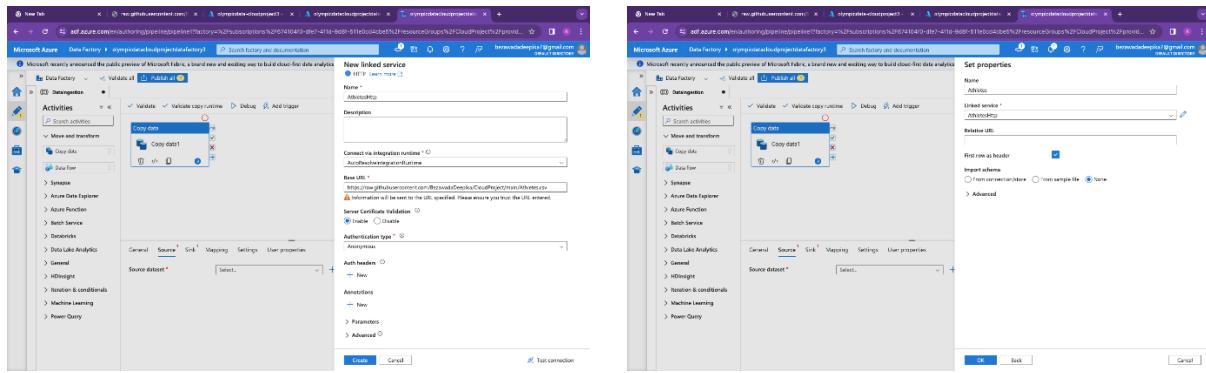
```
PersonName,Country,Discipline
AALERUD Katrine,Norway,Cycling Road
ABAD Nestor,Spain,Artistic Gymnastics
ABAGNALE Giovanni,Italy,Rowing
ABALDE Alberto,Spain,Basketball
ABALDE Tamara,Spain,Basketball
ABALO Luc,France,Handball
ABAROA Cesar,Chile,Rowing
ABASS Abobakr,Sudan,Swimming
ABBASALI Hamideh,Islamic Republic of Iran,Karate
ABBASOV Islam,Azerbaijan,Wrestling
ABBOTT Emily,Australia,Rhythmic Gymnastics
ABBOTT Monica,United States of America,Baseball/Softball
ABDALLA Abubaker Haydar,Qatar,Athletics
ABDALLA Maryam,Egypt,Artistic Swimming
ABDALLA Shahid,Egypt,Artistic Swimming
ABDELMASSOUD Ahmed,Egypt,Handball
ABDEL RAZEK Samy,Egypt,Shooting
ABDELAZIZ Abdalla,Egypt,Karate
ABDELAZIZ Farah,Egypt,Table Tennis
ABDELMANSOUR Feryal,Egypt,Karate
ABDELMOTTALEB Dlalaedin Kamal Gouda,Egypt,Wrestling
ABDELRAMAN Ihab,Egypt,Athletics
ABDELSALAM Mohamed,Egypt,Football
ABDELSALAM Nour,Egypt,Taekwondo
ABDELWAHAB Ahmed,Italy,Athletics
ABDUL Bashir,Maldives,Boxing
ABDURAHMAN Abdi,United States of America,Athletics
ABDUL HADI Farah Ann,Malaysia,Artistic Gymnastics
ABDUL RAHMAN Kiria Tikanah,Singapore,Fencing
ABDUL RAZZAQ Fathimath Nabaha,Maldives,Badminton
ABDUL RAHIM,Iraq,Saudi Arabia,Football
ABDUL JABBAR Amin,Riad,Judo
ABDULLAEV Gulomjon,Uzbekistan,Wrestling
ABDULLAEV Mumunjon,Uzbekistan,Wrestling
ABDULLAH Rahmat Erwin,Indonesia,Weightlifting
ABDULLIN Ilfat,Kazakhstan,Archery
ABDULKARIM Mohamed,Uzbekistan,Handball
ABDURAINOV Elmur,Uzbekistan,Boxing
ABDURAKHMONOV Ravuljon,Uzbekistan,Artistic Gymnastics
ABDURAKHMONOV Bekzod,Uzbekistan,Wrestling
ABE Hifumi,Japan,Judo
ABE Takatoshi,Japan,Athletics
ABE Utsuro,Japan,Athletics
ABEBE Melides,Ethiopia,Athletics
ABEDINI Mojtaba,Islamic Republic of Iran,Fencing
ABEL Jennifer,Canada,Diving
ABELA Matthew,Malta,Badminton
ABELVIK ROED Magnus,Norway,Handball
ABEYSINGHE Matthew,Sri Lanka,Swimming
```

Pipeline creation: Next to create a pipeline, click on pipeline tab and give your pipeline a name.



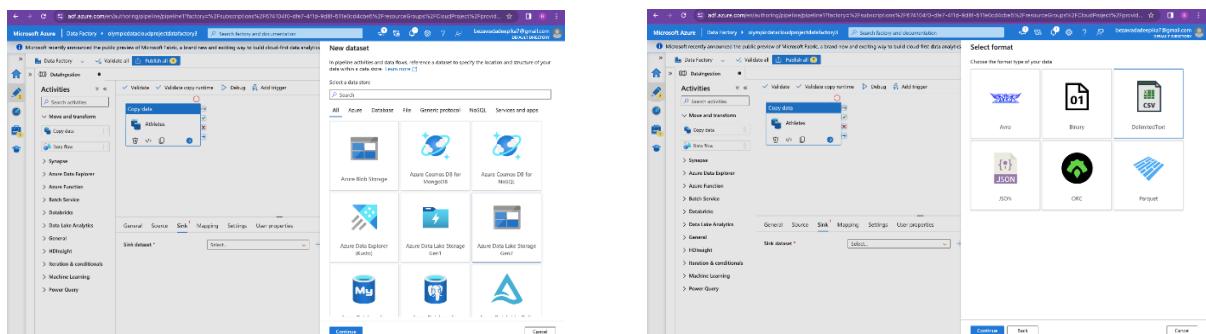
Drag a copy activity onto the canvas tool. Next click on source tab, we need to create the link from our data factory to GitHub repository for this file. click on new icon we can see we can extract data from multiple places we have the Amazon RDS, Azure Blob Storage, Azure data lake Gen2 then again, we have multiple options available and from that data source you can easily access and integrate the data factories from these options. We are interested in HTTP. Because this raw data is accessed through http server.

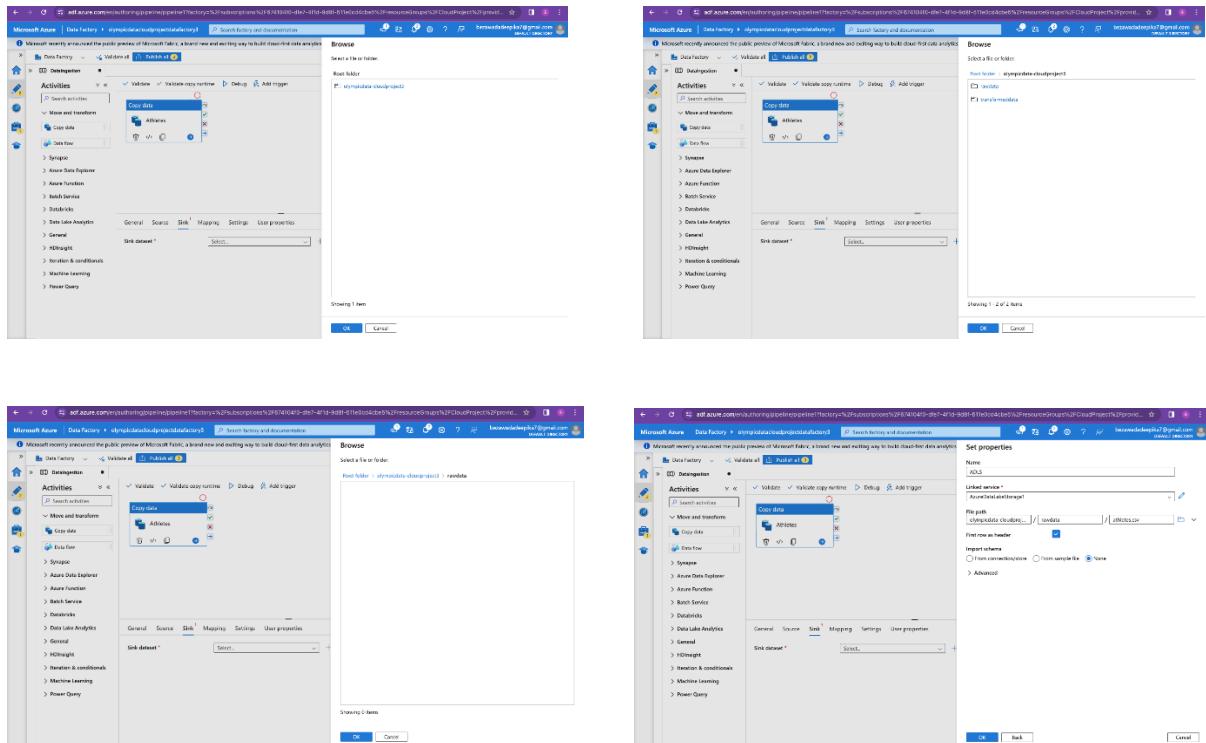




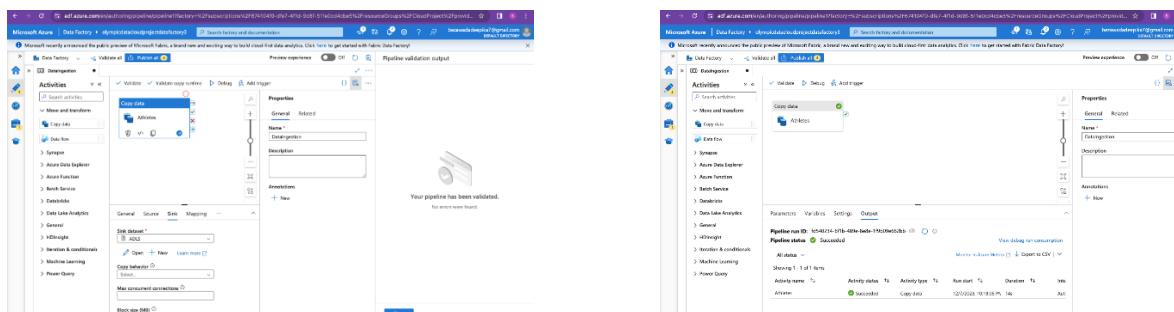
	PersonName	Country	Discipline
1	AALERUD Katrine	Norway	Cycling Road
2	ABAD Nestor	Spain	Artistic Gymnastics
3	ABAGNALE Giovanni	Italy	Rowing
4	ABALDE Alberto	Spain	Basketball
5	ABALDE Tamara	Spain	Basketball
6	ABALO Luc	France	Handball
7	ABAROA Cesar	Chile	Rowing
8	ABASS Abobakr	Sudan	Swimming
9	ABBASALI Hamideh	Islamic Republic of Iran	Karate
10	ABBAZOV Islam	Azerbaijan	Wrestling

In the "Sink" tab of the "Copy" activity, configure the destination where you want to store the extracted data. Choose Azure Data Lake Store2 and Configure the connection and path to the destination storage.

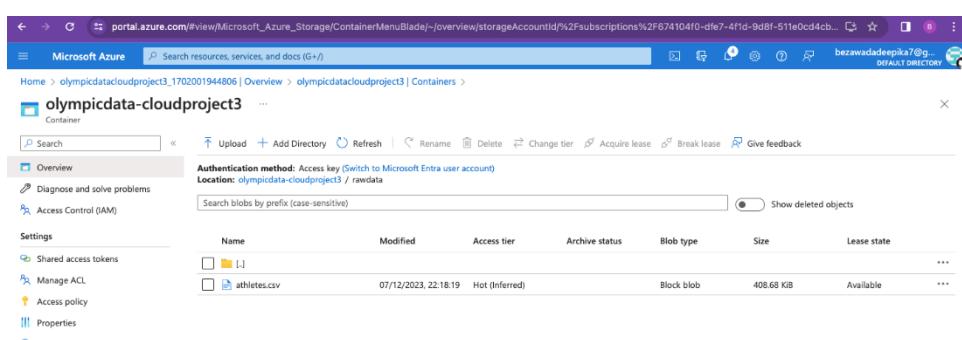




Test and run the pipeline: Click the "Debug" button to test the pipeline with a small dataset. Once satisfied, click the "Publish All" button to make the pipeline available for execution.



Now we can see athletes.csv is available in our storage account. We were able to extract data from GitHub repository using azure data factory.



By following these steps for all the five .csv files, we successfully created a data factory pipeline and ingested all the data onto azure.

Loaded datasets into azure data lake storage by creating a simple pipeline using azure data factory.

Microsoft Azure | portal.azure.com#view/Microsoft_Azure_Storage/ContainerMenuBlade/~/overview/storageAccountid%2Fsubscriptions%2F674104f0-dfe7-4f1d-9d8f-511e0cd4cbe%2FresourceGroups%2FCloudProject%2Fprovider%2F... DEFAULT DIRECTORY (BEZAWADADEEPIKA7@GMAIL.COM)

Home > **olympicdata-cloudproject3** Container

Search resources, services, and docs (G+)

Search blobs by prefix (case-sensitive)

Show deleted objects

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[...]						...
athletes.csv	07/12/2023, 22:44:35	Hot (Inferred)		Block blob	408.68 KIB	Available
coaches.csv	07/12/2023, 22:44:51	Hot (Inferred)		Block blob	16.49 KIB	Available
entriesgender.csv	07/12/2023, 22:45:03	Hot (Inferred)		Block blob	1.1 KIB	Available
medals.csv	07/12/2023, 22:45:16	Hot (Inferred)		Block blob	2.36 KIB	Available
teams.csv	07/12/2023, 22:45:29	Hot (Inferred)		Block blob	34.44 KIB	Available

Created a pipeline.

Microsoft Azure | Data Factory > olympicdatacloudprojectdatafactory3 | Search factory and documentation

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Data Factory Validate all Publish all

Preview experience Off

DataIngestion

Validate Debug Add trigger

Properties

General Related

Name * DataIngestion

Description

Annotations

+ New

Copy data Athletes

Copy data Coaches

Copy data EntriesGender

Copy data Medals

Copy data Teams

Parameters Variables Settings Output

Pipeline run ID: 9571a010-da88-4dc7-9eef-8e67da764790 Pipeline status: Succeeded

All status ▾

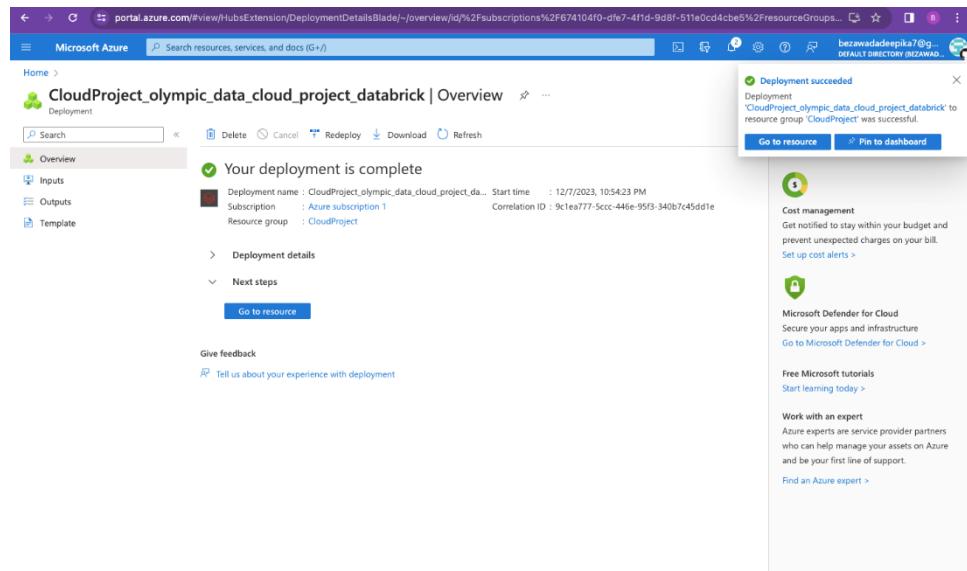
Showing 1 - 5 of 5 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User prop
Teams	Succeeded	Copy data	12/7/2023, 10:45:19 PM	12s	AutoResolveIntegration	
Medals	Succeeded	Copy data	12/7/2023, 10:45:06 PM	12s	AutoResolveIntegration	
EntriesGender	Succeeded	Copy data	12/7/2023, 10:44:53 PM	12s	AutoResolveIntegration	
Coaches	Succeeded	Copy data	12/7/2023, 10:44:37 PM	15s	AutoResolveIntegration	
Athletes	Succeeded	Copy data	12/7/2023, 10:44:22 PM	15s	AutoResolveIntegration	

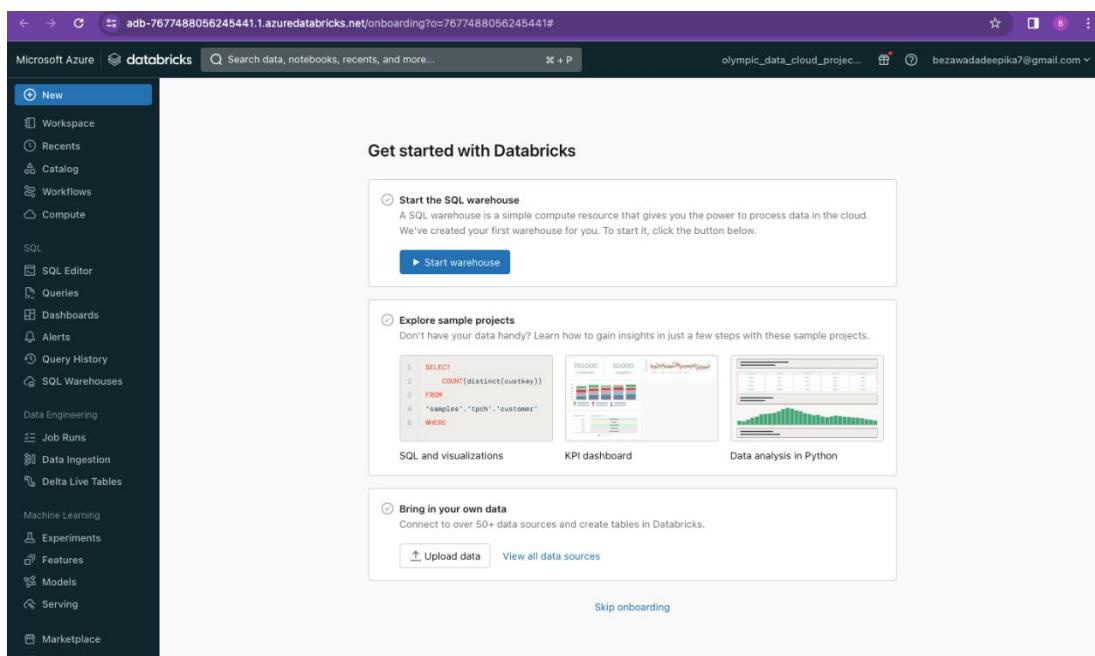
- **Data Transformation:**

This stage involves the successful creation of Data bricks. This step takes advantage of Azure Databrick's distributed computing capabilities for efficient processing.

Created azure data bricks service.



Go to resource and click on launch workspace and we will be redirected to the Azure database workspace where we can create the notebook and instance and so on.



Created the compute because we want to write our spark code. For that click on create compute.

The screenshot shows the Azure Databricks Compute page. On the left, there is a sidebar with a 'Compute' section selected. The main area displays a table of clusters. One cluster is listed: 'Deepika Bezawada's Cluster'. The table includes columns for State, Name, Policy, Runtime, Active memory, Active cores, Active DBs, Source, Creator, Notebooks, and a more button. At the top right, there are buttons for 'Create with Personal Compute' and 'Create compute'.

Next, we need to create a connection from azure data bricks to our azure data storage so that we can easily access the data. To create a connection, we need client Id and tenant id. So for that we created app registrations and got those credentials.

The screenshot shows the Azure portal's App registrations page for an application named 'app03'. The left sidebar lists various management options like Overview, Quickstart, Integration assistant, Manage, Support + Troubleshooting, and more. The main content area shows the 'Overview' tab for the app. It displays the display name 'app03', application (client) ID 'a0130042-1e05-4a31-94be-1b5d23548d92', object ID '22b443f4-541a-4ccf-aeb8-0cd82df8f9fc', directory (tenant) ID 'c0efb7c6-6844-48b3-b9aa-9a4fb7b7312c', and supported account types 'My organization only'. There are also sections for Client credentials, Redirect URIs, Application ID URI, and Managed application in L. At the bottom, there are links for 'Get Started' and 'Documentation', and a section titled 'Build your application with the Microsoft identity platform'.

Go to certificates & secrets and get the secret id.

Azure data bricks workspace:

1. Create connection from this azure data bricks to our azure data storage so that we can easily access the data.

The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** OlympicDataTransformation
- Languages:** Python
- Last edited:** 23 hours ago
- Feedback:** Provide feedback
- Run all:** Run all cells
- Share:** Share the notebook

The code cell contains the following Python code:

```
1 configs = {"fs.azure.account.auth.type": "OAuth",
2 "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
3 "fs.azure.account.oauth.provider.extra.configs": "{'client_id': '00130042-1e05-4311-940e-1b5d23548d92',
4 'client_secret': 'MTbQo-JNaUxVr-fPhUsRx_GENQKZfsgDZfcIz',
5 'fs.azure.account.oauth2.client.endpoint': 'https://login.microsoftonline.com/c0efb7c6-6844-48b3-b9aa-9af4fb7b7312/oauth2/token'}",
6
7
8 dbutils.fs.mount(
9 source = "abfss://olympicdata-cloudproject3@olympicdatacloudproject3.dfs.core.windows.net", # contrainer@storageacc
10 mount_point = "/mnt/olympicdatacloudproject",
11 extra_configs = configs)
```

Below the code cell, the status bar indicates:

Command took 11.16 seconds -- by bezawadadeepika7@gmail.com at 08/12/2023, 00:10:37 on Deepika Bezawada's Cluster

2. We successfully created the connection to the azure data factory. We can check the same by running the following command. If connection is successful, then we would be able to see all our files from this mount location.

```
Cmd 2

1 %fs
2 ls "/mnt/olympicdatacloudproject"
```

Result: Able to see all files from mount location

Table + New result table: OFF ▾

path	name	size	modificationTime
1 dbfs:/mnt/olympicdatacloudproject/rawdata/	rawdata/	0	1702002911000
2 dbfs:/mnt/olympicdatacloudproject/transformeddata/	transformeddata/	0	1702002929000

↓ 2 rows | 1.00 second runtime Refreshed 2 days ago

Command took 1.00 second -- by bezawadadeepika7@gmail.com at 08/12/2023, 0:01:22 on Deepika Bezawada's Cluster

3. Importing specific functions and types from PySpark to work with Spark Data Frames.

4. Read the files

```

Cd 10

1 teams = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/olympicdatacloud/project/rpdata/teams.csv")

# [2] Spark Jobs
2 teams = pyspark.sql.DataFrame[TeamName: string, Discipline: string, ... 2 more fields]
Command took 1.15 seconds -- by bezzadepareshuk@gmail.com at 08/12/2021, 01:19:55 on Deepika Bezzad's Cluster
Cd 20

1 teams.show(1)

# [1] Spark Jobs
+-----+-----+-----+-----+
|TeamName|Discipline|Country|Event|
+-----+-----+-----+-----+
|Belgium|Basketball|Belgium|Men|
|China|Basketball|People's Republic of China|Men|
|China|Basketball|People's Republic of China|Women|
|France|Basketball|France|Men|
|Italy|Basketball|Italy|Men|
|Japan|Basketball|Japan|Men|
|Japan|Basketball|Japan|Women|
|Latvia|Basketball|Latvia|Men|
|Mongolia|Basketball|Mongolia|Men|
|Mongolia|Basketball|Mongolia|Women|
|Netherlands|Basketball|Netherlands|Men|
|Pakistan|Basketball|Pakistan|Men|
|ROC|Basketball|ROC|Men|
|ROC|Basketball|ROC|Women|
|Romania|Basketball|Romania|Men|
|Serbia|Basketball|Serbia|Men|
|United States|Basketball|United States of America|Men|
|United States|Basketball|United States of America|Women|
|Australia|Archery|Australia|Mixed Team|
|Australia|Archery|Australia|Women|
+-----+-----+-----+-----+
Command took 0.37 seconds -- by bezzadepareshuk@gmail.com at 08/12/2021, 01:20:51 on Deepika Bezzad's Cluster

```

5. Transformed the datatypes

For Entriesgender.csv and Medals.csv

```
Cmd 11

1 entriesgender.printSchema()

root
 |-- Discipline: string (nullable = true)
 |-- Female: string (nullable = true)
 |-- Male: string (nullable = true)
 |-- Total: string (nullable = true)

Command took 0.11 seconds -- by bezawadadeepika@gmail.com at 08/12/2023, 01:01:44 on Deepika Bezawada's Cluster

Cmd 12

1 entriesgender = entriesgender.withColumn("Female",col("Female").cast(IntegerType()))
2 .withColumn("Male",col("Male").cast(IntegerType()))
3 .withColumn("Total",col("Total").cast(IntegerType()))

+---+ entriesgender = entriesgender.withColumn("Female",col("Female").cast(IntegerType()))
|Discipline|Female|Integer
|Male|Integer
|Total|Integer

Command took 0.20 seconds -- by bezawadadeepika@gmail.com at 08/12/2023, 01:11:39 on Deepika Bezawada's Cluster

Cmd 13

1 entriesgender.printSchema()

root
 |-- Discipline: string (nullable = true)
 |-- Female: integer (nullable = true)
 |-- Male: integer (nullable = true)
 |-- Total: integer (nullable = true)

Command took 0.05 seconds -- by bezawadadeepika@gmail.com at 08/12/2023, 01:11:55 on Deepika Bezawada's Cluster
```

```
1 medals.printSchema()

root
 |-- Rank: string (nullable = true)
 |-- Team_Country: string (nullable = true)
 |-- Gold: string (nullable = true)
 |-- Silver: string (nullable = true)
 |-- Bronze: string (nullable = true)
 |-- Total: string (nullable = true)
 |-- Rank by Total: string (nullable = true)

Command took 0.84 seconds -- by bezawadadeepeka@gmail.com at 08/12/2023, 01:15:56 on Deepika Bezawada's Cluster
Cmd 17

1 #transform data types
2 medals = medals.withColumn("Gold",col("Gold").cast(IntegerType()))
3 .withColumn("Silver",col("Silver").cast(IntegerType()))
4 .withColumn("Bronze",col("Bronze").cast(IntegerType()))
5 .withColumn("Total",col("Total").cast(IntegerType()))

# medals: pyspark.sql.dataframe.DataFrame = [Rank: string, Team_Country: string ... 6 more fields]

Command took 0.27 seconds -- by bezawadadeepeka@gmail.com at 08/12/2023, 01:16:39 on Deepika Bezawada's Cluster
Cmd 18

1 medals.printSchema()

root
 |-- Rank: string (nullable = true)
 |-- Team_Country: string (nullable = true)
 |-- Gold: integer (nullable = true)
 |-- Silver: integer (nullable = true)
 |-- Bronze: integer (nullable = true)
 |-- Total: integer (nullable = true)
 |-- Rank by Total: string (nullable = true)
```

6. After performing transformation the data is loaded into data lake (i.e the data will be loaded into transformed folder).

```
Cmd 25

1 #load the tables after transformation into datalake
2 athletes.write.option("header","true").csv("/mnt/olympicdatacloudproject/transformeddata/athletes")

▶ (1) Spark Jobs
Command took 0.78 seconds -- by bezawadadeepika7@gmail.com at 08/12/2023, 02:15:51 on Deepika Bezawada's Cluster
Cmd 26

1 medals.write.option("header","true").csv("/mnt/olympicdatacloudproject/transformeddata/medals")

▶ (1) Spark Jobs
Command took 0.97 seconds -- by bezawadadeepika7@gmail.com at 08/12/2023, 02:17:01 on Deepika Bezawada's Cluster
Cmd 27

1 entriesgender.write.option("header","true").csv("/mnt/olympicdatacloudproject/transformeddata/entriesgender")

▶ (1) Spark Jobs
Command took 0.94 seconds -- by bezawadadeepika7@gmail.com at 08/12/2023, 02:18:25 on Deepika Bezawada's Cluster
Cmd 28

1 coaches.write.option("header","true").csv("/mnt/olympicdatacloudproject/transformeddata/coaches")

▶ (1) Spark Jobs
Command took 0.89 seconds -- by bezawadadeepika7@gmail.com at 08/12/2023, 02:20:24 on Deepika Bezawada's Cluster
Cmd 29

1 teams.write.option("header","true").csv("/mnt/olympicdatacloudproject/transformeddata/teams")

▶ (1) Spark Jobs
Command took 0.77 seconds -- by bezawadadeepika7@gmail.com at 08/12/2023, 02:20:53 on Deepika Bezawada's Cluster
```

Data Analysis:

Used azure synapse analytics to perform data analysis. We need to load that transformed data into azure synapse analytics.

Created azure synapse analytics.

Name	Type	Size
SQL pools		

Load the data. For that you can just click onto the data part and click on plus icon and click on lake database.

Next repeat the above steps for all the five files to load the data into database. All the tables were loaded into the database. And then validate and publish all.

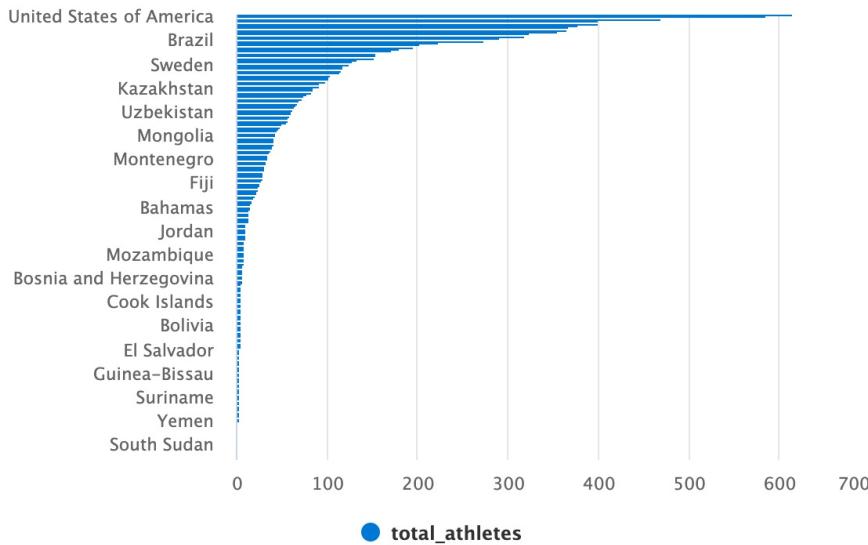
Analysis part:

1. This query provides the count of athletes per country.

```

1 --Analysis1:Athletes Count from each country
2 --Count the no of athletes from each country;
3 SELECT Country, count(*) AS total_athletes
4   from table_athletes
5 GROUP BY Country
6   order by total_athletes Desc;
7

```



The United States has the most athletes competing in the Olympics by a significant margin, with over 600 athletes participating. Brazil is the second-highest country, with over 400 athletes competing. The remaining countries in the top 10 all have between 100 and 300 athletes competing.

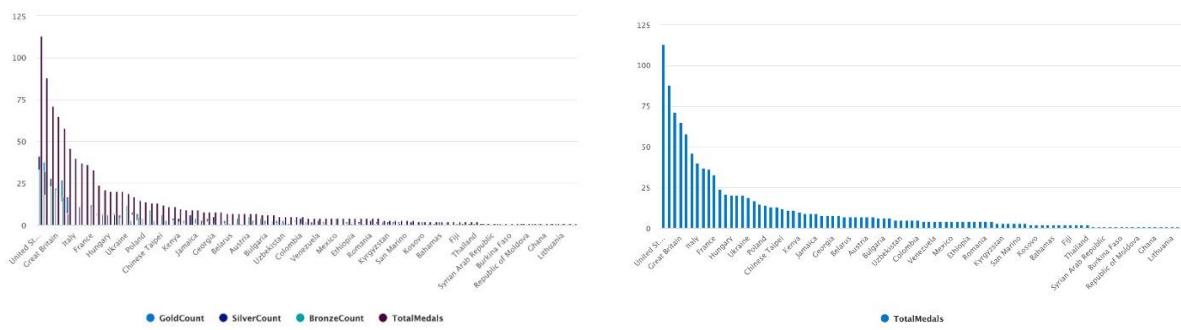
We also observed that there is a significant disparity in the number of athletes competing from different countries.

2. Calculated the total number of medals won by each country in the Olympics.

```

7
8    -- Analysis 2:Medals conunt by each country
9    --calculate the total medals won by each country;
10   SELECT Team_country AS Country,
11       SUM(Gold) AS GoldCount,
12       SUM(Silver) AS SilverCount,
13       SUM(Bronze) AS BronzeCount,
14       SUM(Total) AS TotalMedals
15   FROM medals
16   GROUP BY Team_country
17   ORDER BY TotalMedals DESC;
18

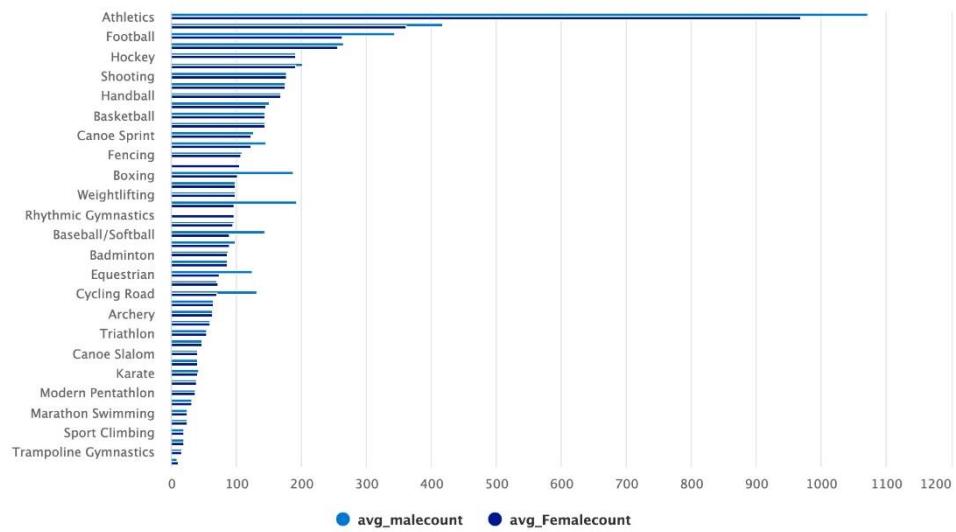
```



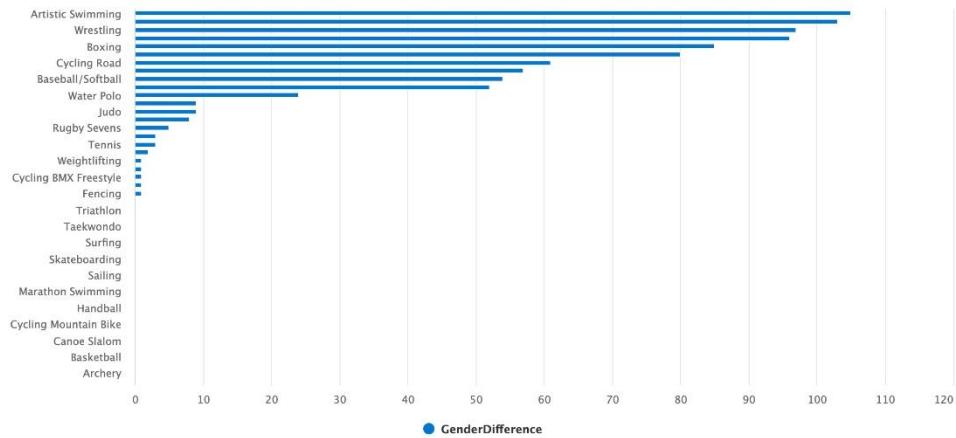
This visualization showed that the United States has been the most successful country in the Olympics (2021) in terms of medals won.

3. Calculated the average number of male and female entries for each discipline in the Olympics.

```
-- Analysis 3: Gender distribution by discipline
--calculate the average number of entries by gender for each discipline;
select Discipline, AVG(Male) AS avg_malecount, AVG(Female) as avg_Femalecount
FROM entriesgender
GROUP BY Discipline
ORDER by avg_Femalecount DESC;
```



```
25
26 -- Which discipline had the most significant difference in the number of entries by gender?
27 WITH GenderEntryDiff AS (
28     SELECT Discipline,
29             ABS(SUM(Female) - SUM(Male)) AS GenderDifference
30     FROM entriesgender
31     GROUP BY Discipline
32 )
33 SELECT Discipline, GenderDifference
34 FROM GenderEntryDiff
35 ORDER BY GenderDifference DESC;
36
```

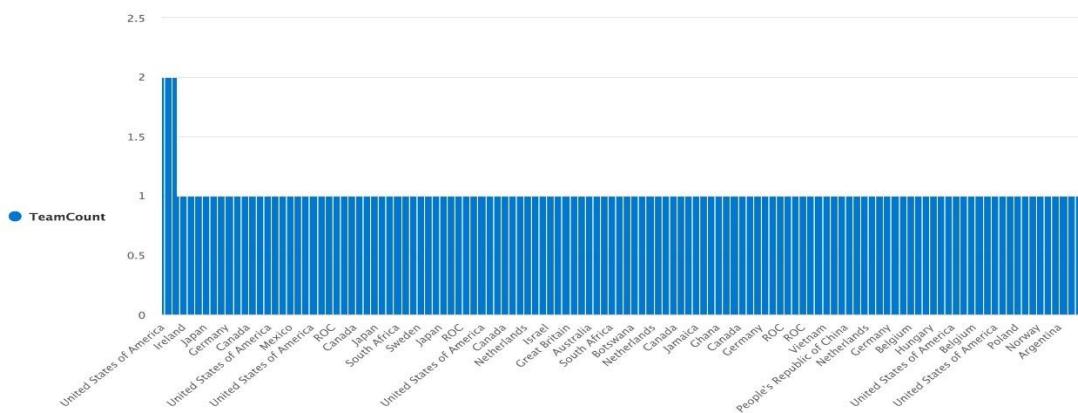


There is a significant disparity in the average number of male and female entries for many disciplines, but there is a growing trend of female participation in many traditionally male-dominated sports.

4. This query shows the number of teams participating in each event for different countries and disciplines.

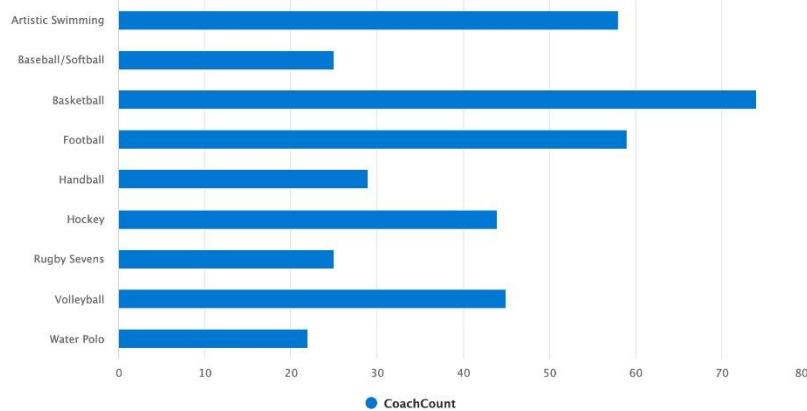
```
36
37  --Analysis 4: Event participation by teams
38  --Count of teams participating in each event for different countries and disciplines;
39  SELECT t.Discipline,
40    |   t.Country,
41    |   t.Event,
42    |   COUNT(*) AS TeamCount
43  FROM teams t
44  GROUP BY t.Discipline, t.Country, t.Event
45  ORDER BY TeamCount DESC;
```

Discipline	Country	Event	TeamCount
Beach Volleyball	United States of America	Women	2
Beach Volleyball	United States of America	Men	2
Beach Volleyball	Switzerland	Women	2
Beach Volleyball	ROC	Men	2
Beach Volleyball	Poland	Men	2
Beach Volleyball	People's Republic of China	Women	2
Beach Volleyball	Netherlands	Women	2
Beach Volleyball	Brazil	Men	2
Beach Volleyball	Brazil	Women	2
Beach Volleyball	Canada	Women	2
Beach Volleyball	Italy	Men	2
Beach Volleyball	Germany	Women	2
Dunkin' Donuts	New Zealand	Men	1



5. This query counts the number of unique coaches per discipline.

```
46  --Analysis 5: coaches per discipline
47  SELECT c.Discipline,
48  |   COUNT(DISTINCT c.Name) AS CoachCount
49  FROM coaches c
50  GROUP BY c.Discipline;
```



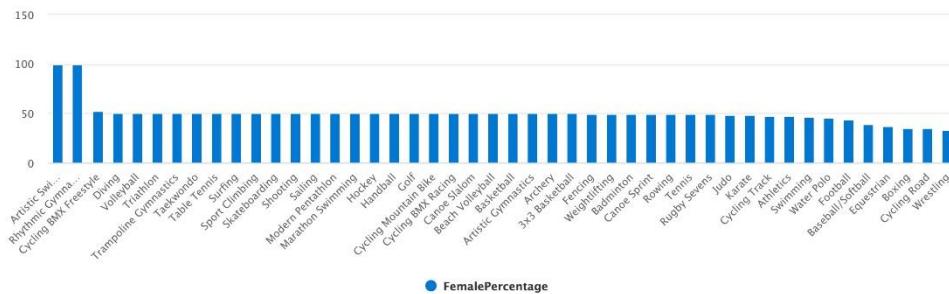
We can see that Basketball discipline has the highest number of coaches.

6. Calculates the percentage of female participation in each discipline, helping identify disciplines with the highest female involvement.

```

51 -- Analysis 6: Discipline with Highest Female Participation percentage
52 WITH TotalGenderCounts AS (
53     SELECT Discipline,
54         SUM(Female) AS TotalFemale,
55         SUM(Male) AS TotalMale
56     FROM entriesgender
57     GROUP BY Discipline
58 )
59     SELECT Discipline,
60         (TotalFemale * 100.0) / (TotalFemale + TotalMale) AS FemalePercentage
61     FROM TotalGenderCounts
62     ORDER BY FemalePercentage DESC;

```



The visualization shows that Artistic Swimming and Rhythmic Gymnastics has the highest female participation percentage.

7. Coaches handling multiple disciplines:

```

--Analysis 7: Coaches handling multiple disciplines
SELECT Name AS CoachName,
       COUNT(DISTINCT Discipline) AS DisciplineCount
FROM coaches
GROUP BY Name
HAVING COUNT(DISTINCT Discipline) >1
ORDER BY DisciplineCount DESC;

```

DATA VISUALIZATION:



Conclusion:

In conclusion, leveraging Azure for Olympic data analysis provided a robust platform for scalable data processing and insightful analytics. The integration of Azure services facilitated efficient data storage, seamless data querying, and advanced analytics. Through Azure Synapse Analytics, the project achieved high-performance data processing, enabling real-time insights and predictive modelling. Azure's diverse toolset, including Synapse Analytics, enhanced data-driven decision-making, offering a comprehensive solution for Olympic data analysis, fostering agility, scalability, and in-depth understanding of athletic performance, trends, and future projections.

Challenges:

- We've made good progress with our project and managed most aspects smoothly. However, accessing raw data from Kaggle became a hurdle due to issues with API access and authentication keys. Our workaround involved downloading the dataset and uploading it to a GitHub repository. From there, we extracted the data and loaded it onto Azure storage using the repository's raw data URL.
- Yet, we hit another roadblock when trying to create an app repository necessary for linking Azure Data Lake Storage to the data factory. We encountered an access issue.
- We didn't get the access for the resources of Azure Databricks, so we opted pay-as-you-go option and continued with the next steps.

Video recording Link:

https://drive.google.com/file/d/1CFqifxQlQvRXIDm6_FV9DSIxYQqncJeh/view?usp=sharing

https://drive.google.com/drive/folders/1YFQzTsRXc7VxD6RWQ8edRaCzzjq7hXJe?usp=drive_link