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

TEAM MEMBERS PICTURES:

Deepika Bezawada
16347826

Naga Sahithi Vunnam
16355633

Greeshmitha Reddy Veeradimmu
16355034

Sai Harshith Nimmalagottu
16352577

Names:

1) Deepika, Bezawada (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:

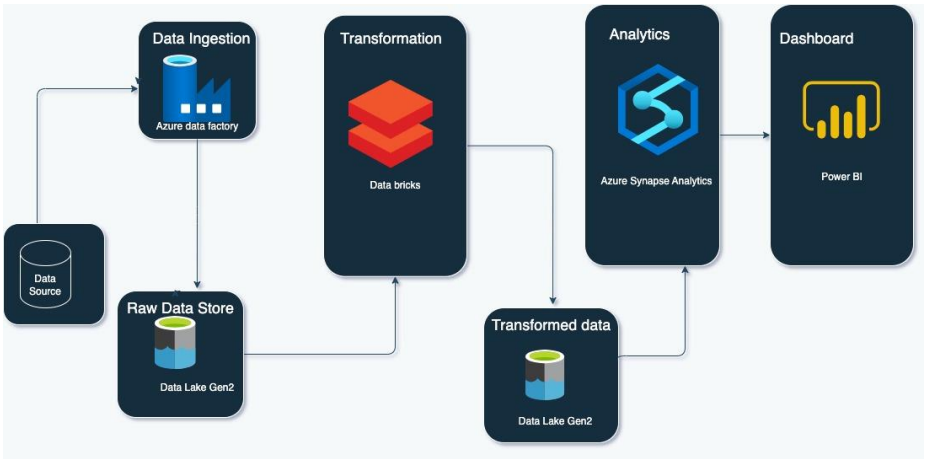


Fig: Olympic data analysis architecture

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.

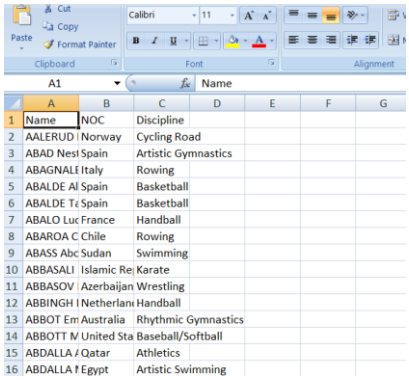
Features (with screenshots):

- ## Data Collection:

We have collected the 2021 Olympic dataset from **Kaggle.**

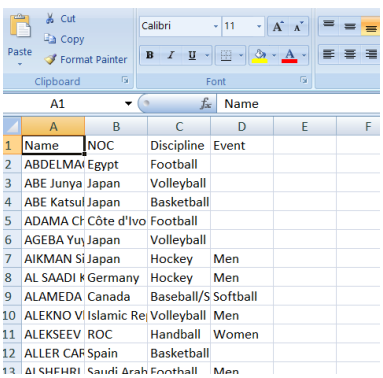
This contains the details of over 11,000 athletes, with 47 disciplines, along with 743 Teams taking part in the 2021(2020) Tokyo Olympics. This dataset contains the details of the Athletes, Coaches, Teams participating and the Entries by gender. It contains their names, countries represented, discipline, 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.



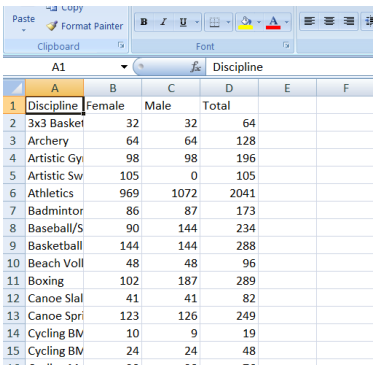
	A	B	C	D	E	F	G
1	Name	NOC	Discipline				
2	AALERUD I	Norway	Cycling Road				
3	ABAD Nest	Spain	Artistic Gymnastics				
4	ABAGNALE	Italy	Rowing				
5	ABALDE Al	Spain	Basketball				
6	ABALDE Te	Spain	Basketball				
7	ABALO Luc	France	Handball				
8	ABAROA C	Chile	Rowing				
9	ABASS Abc	Sudan	Swimming		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

Fig 1: Athletes.csv



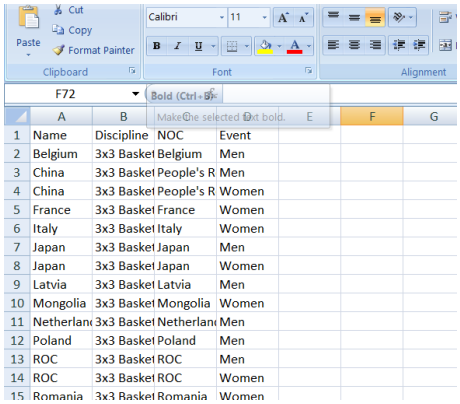
	A	B	C	D	E	F
1	Name	NOC	Discipline	Event		
2	ABDELMA	Egypt	Football			
3	ABE Junya	Japan	Volleyball			
4	ABE Katsul	Japan	Basketball			
5	ADAMA Ch	Côte d'Ivo	Football			
6	AGEBA Yu	Japan	Volleyball			
7	AIKMAN Si	Japan	Hockey	Men		
8	AL SAADI K	Germany	Hockey	Men		
9	ALAMEDA	Canada	Baseball/S	Softball		
10	ALEKNO Vl	Islamic Re	Volleyball	Men		
11	ALEKSEEV	ROC	Handball	Women		
12	ALLER CAF	Spain	Basketball			
13	AL SHEHRI	Saudi Arab	Football	Men		

Fig 2: Coaches.csv



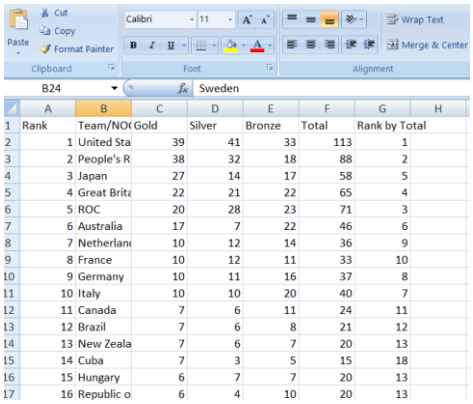
	A	B	C	D	E	F
1	Discipline	Female	Male	Total		
2	3x3 Basket	32	32	64		
3	Archery	64	64	128		
4	Artistic Gy	98	98	196		
5	Artistic Sw	105	0	105		
6	Athletics	969	1072	2041		
7	Badmintor	86	87	173		
8	Baseball/S	90	144	234		
9	Basketball	144	144	288		
10	Beach Voll	48	48	96		
11	Boxing	102	187	289		
12	Canoe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

Fig 3: EntriesGender.csv



	A	B	C	D	E
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	Netherlan	3x3 Basket	Netherlan	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/NO(	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	Netherlan	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	

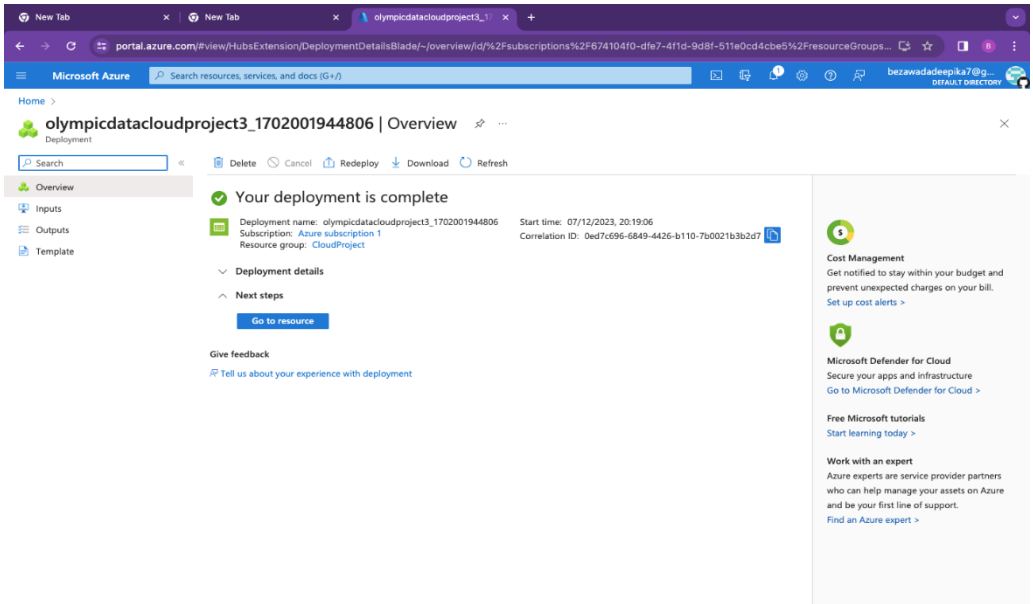
Fig 5: Medals.csv

- **Data Ingestion:**

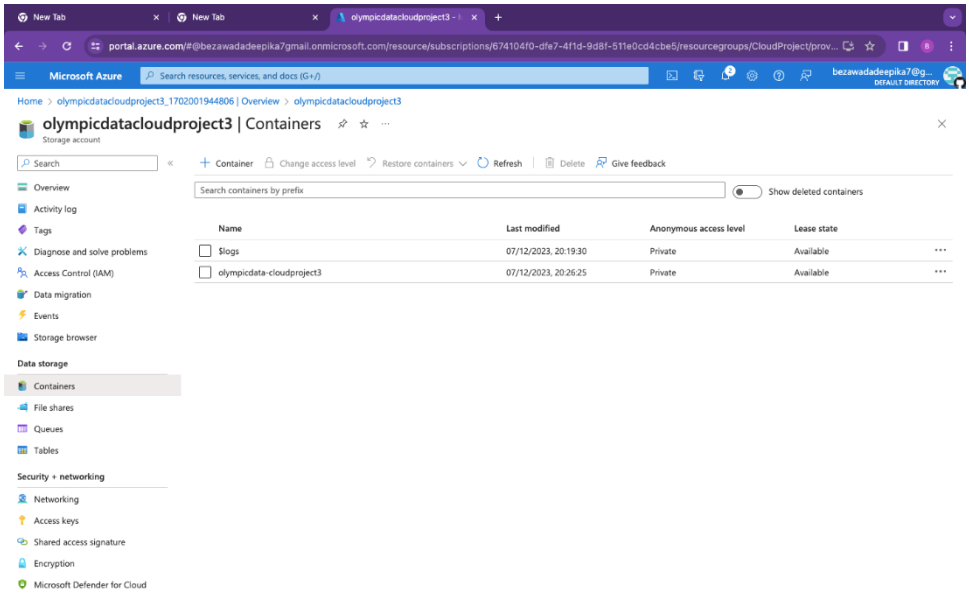
We need to ingest data from external data source onto azure. To do so we used azure data factory to set up a pipeline and extract data from an external data source and load that data onto azure data lake storage.

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

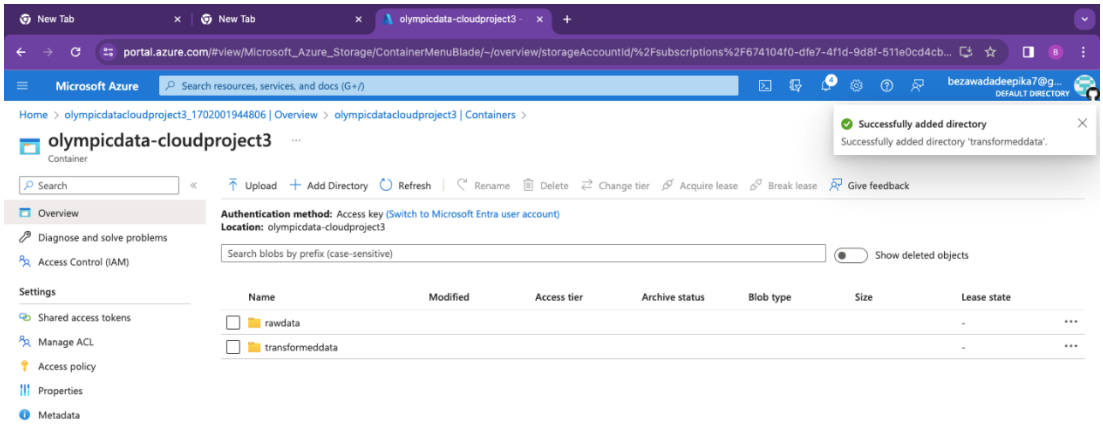
1. Created a storage account.



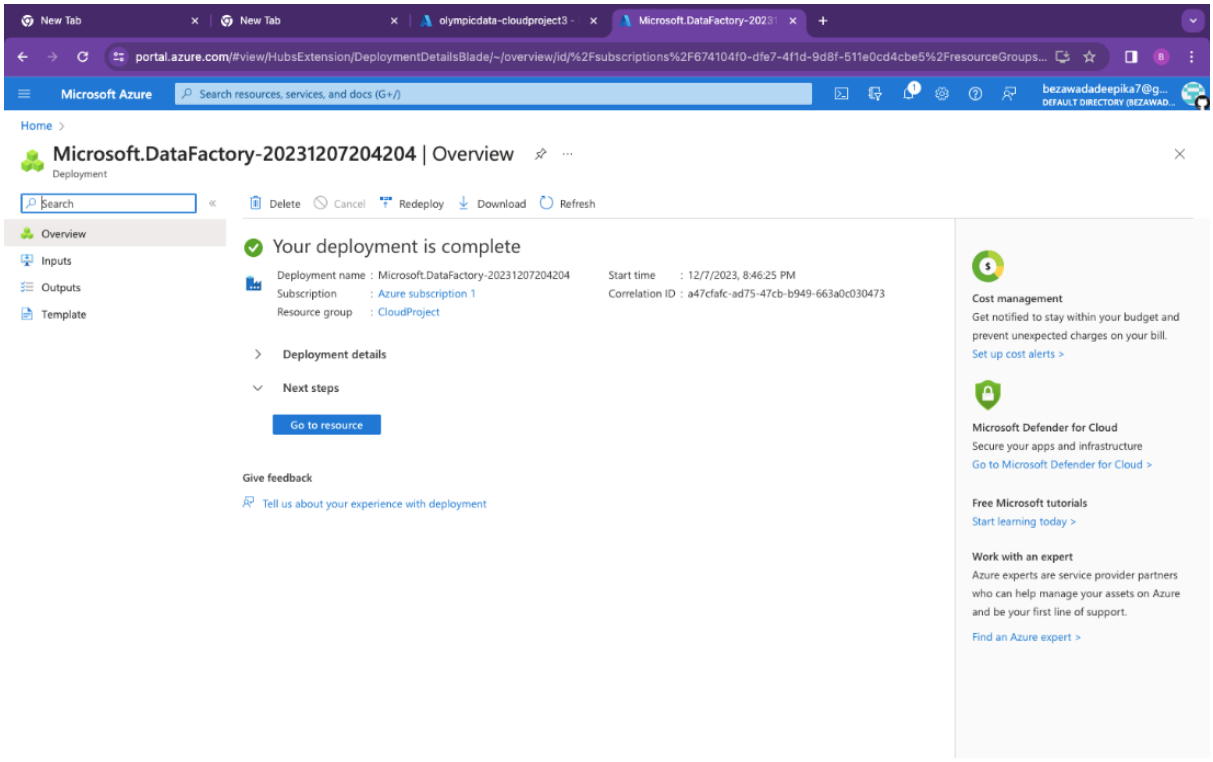
2. Created a container.

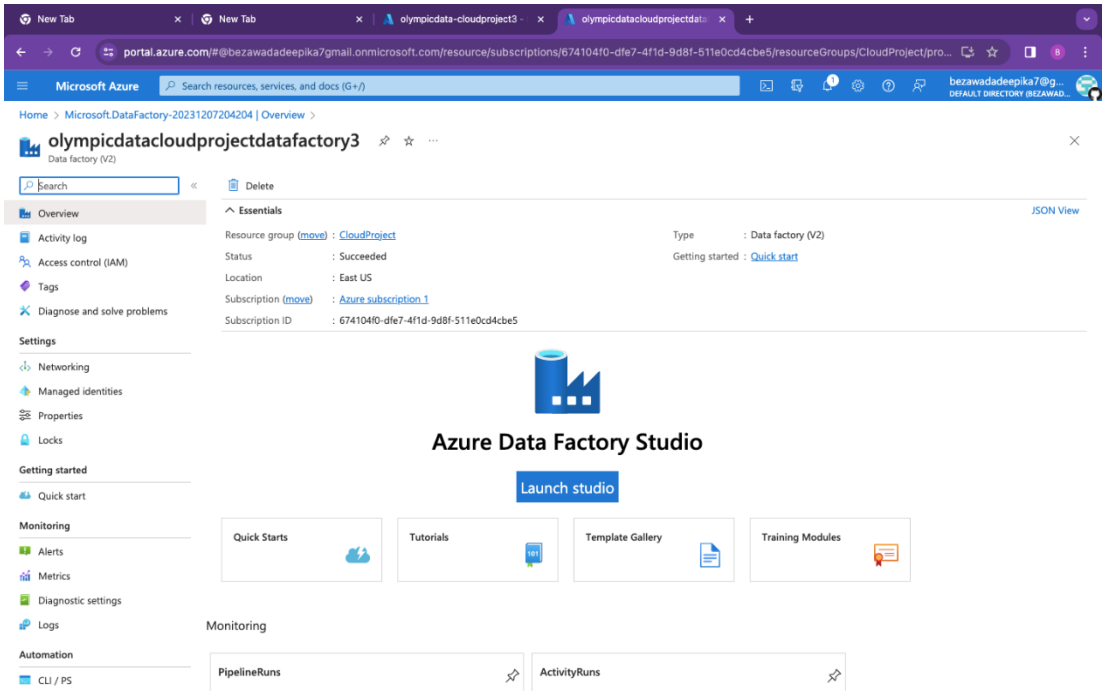


Inside the container we created two folders one is for raw data this where we store all our raw data as it is from the data source that we extract. Next folder is for transformed data.



Created an azure data factory…

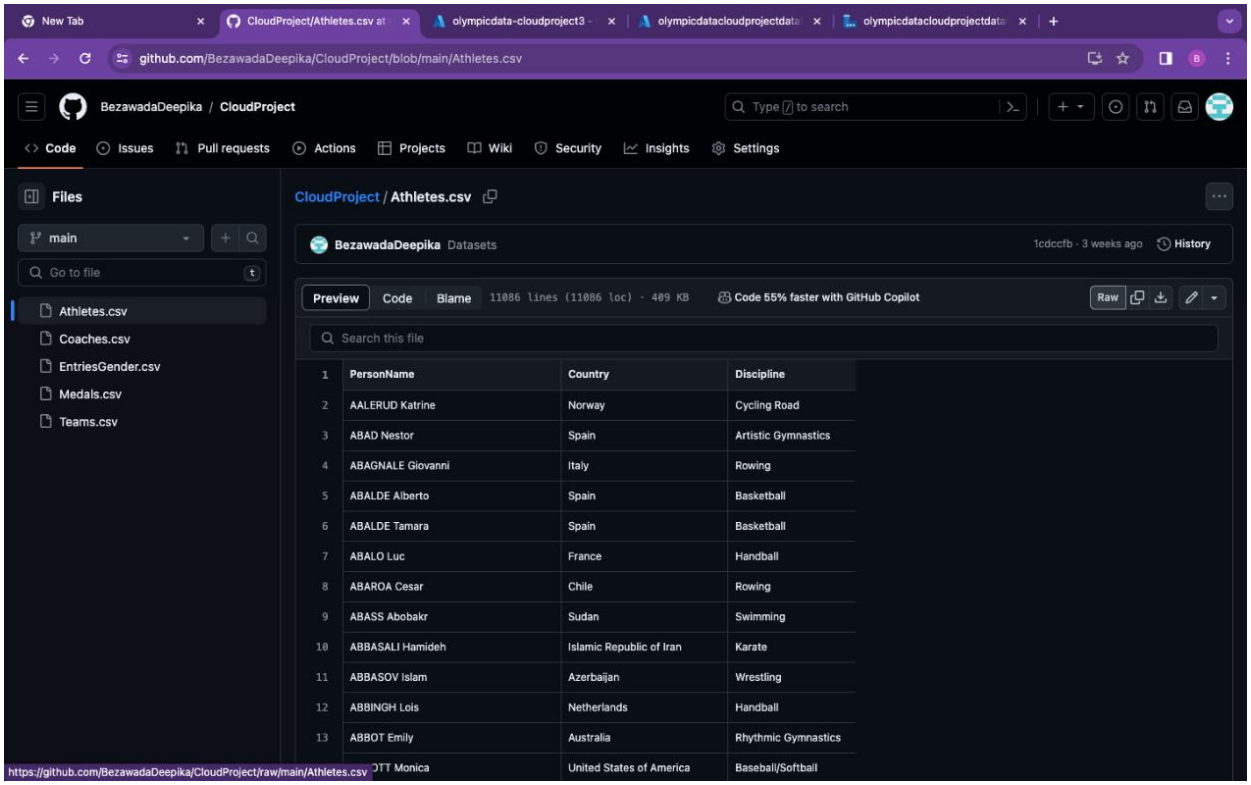




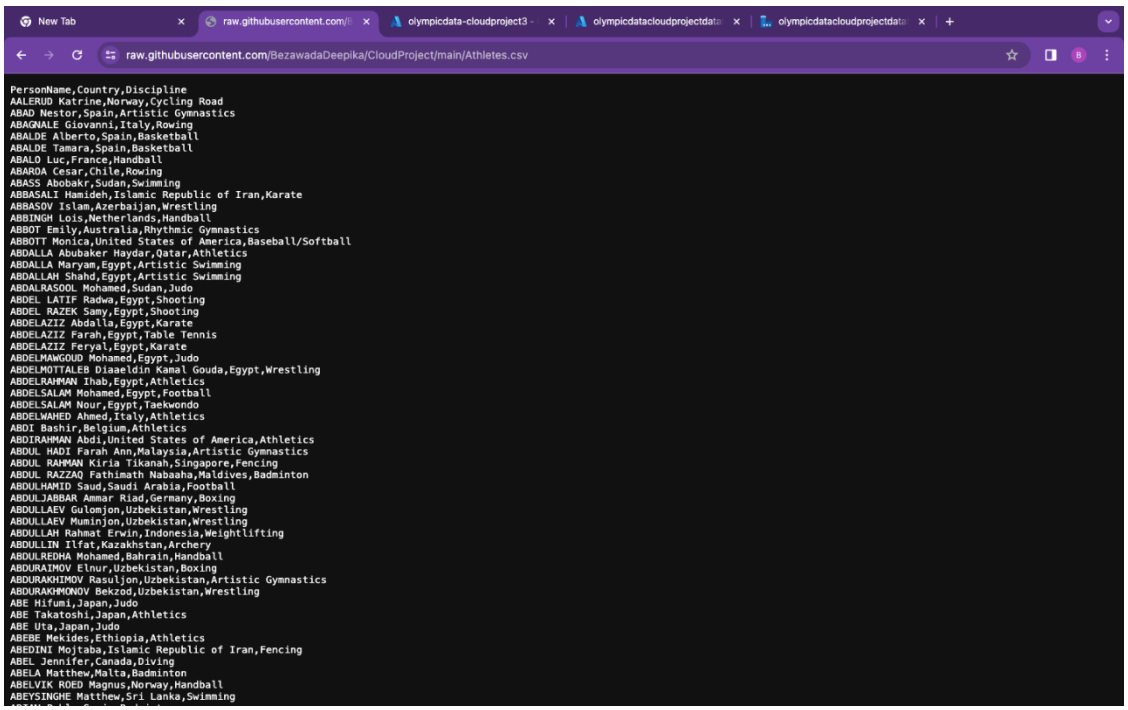
Now the data factory is ready. This is where we create the pipeline to extract the data from data source and upload that data onto their target location.

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.

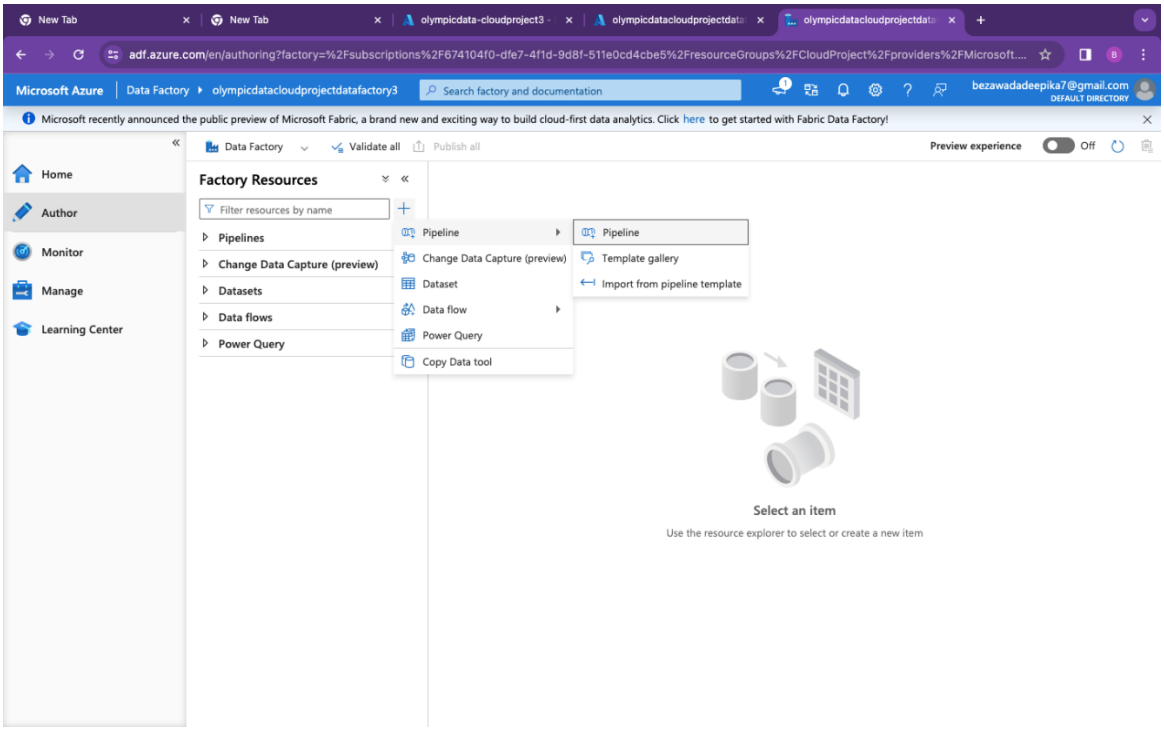
Click on Athletes.csv file and click on raw button.



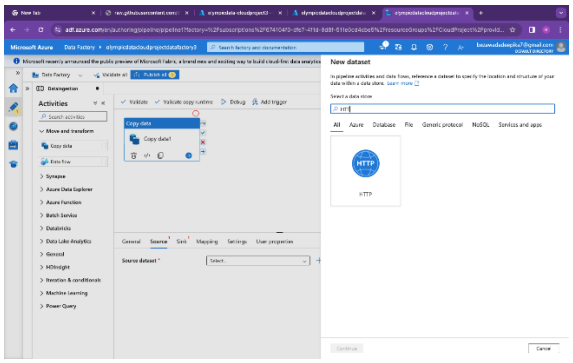
It redirects us to a page. That page contains URL which is the raw URL of this CSV file. And connected it to a source.

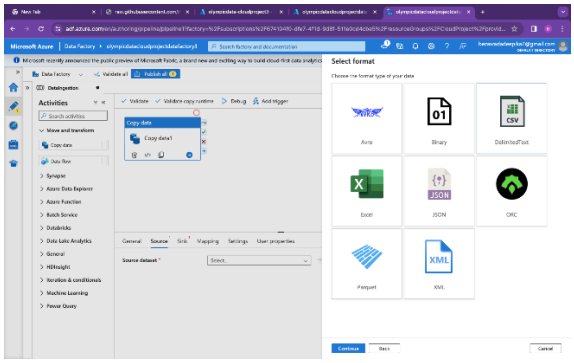


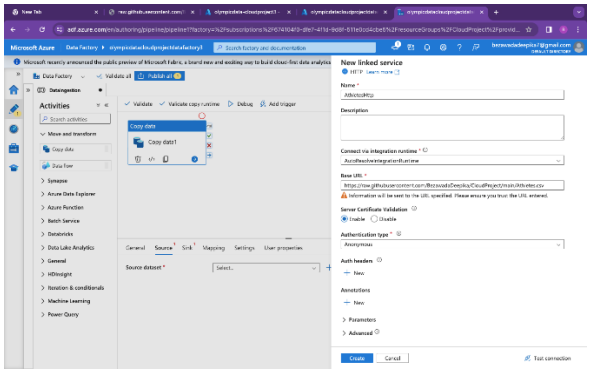
Pipeline creation: Next to create a pipeline, click on pipeline tab and give your pipeline a name.

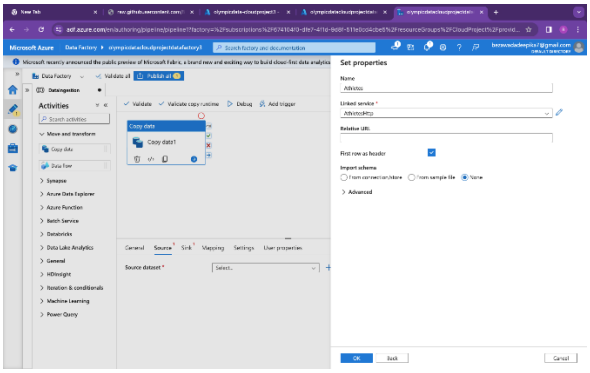


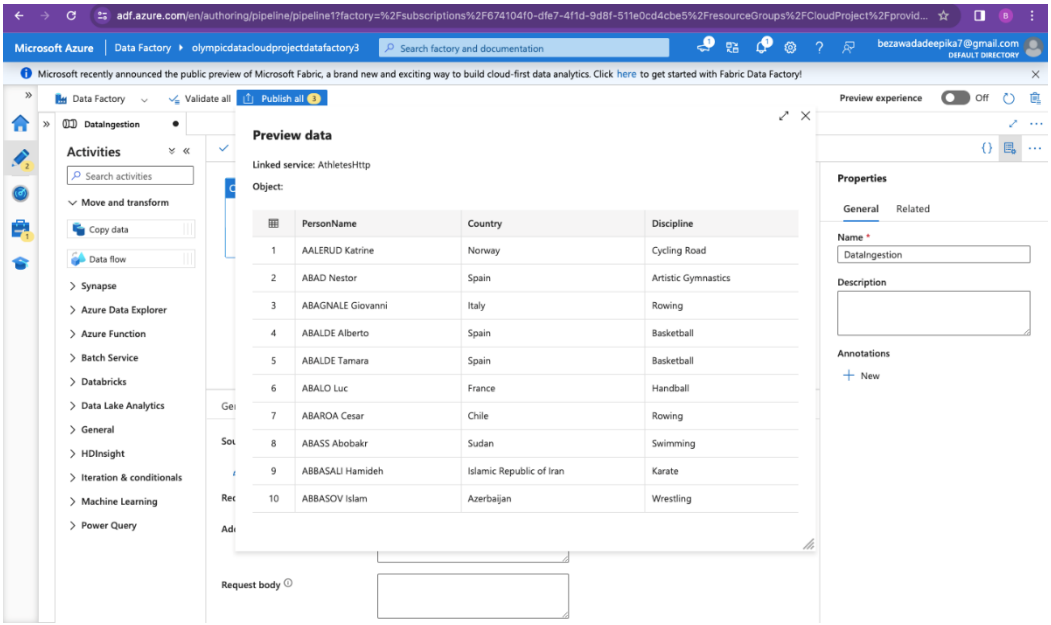
Drag a copy activity onto the canvas tool. Next click on source tab, we need to create the link from our data factory to GitHub repository for this file.
click on new icon we can see we can extract data from multiple places we have the Amazon RDS, Azure Blob Storage, Azure data lake Gen2 then again, 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.



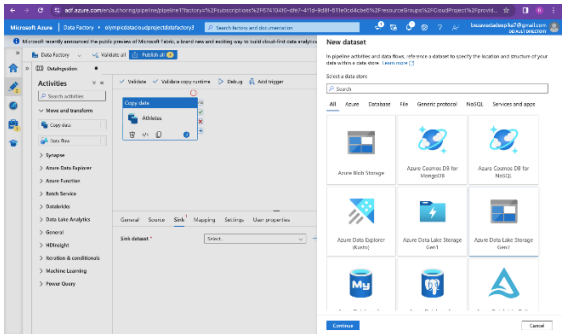


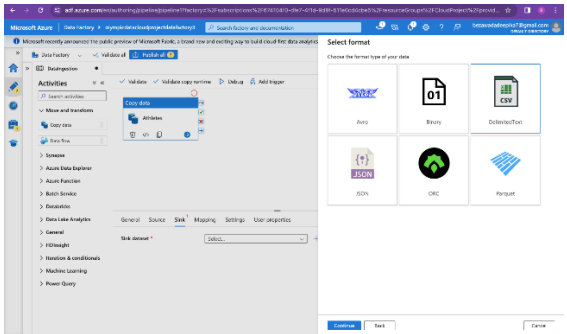


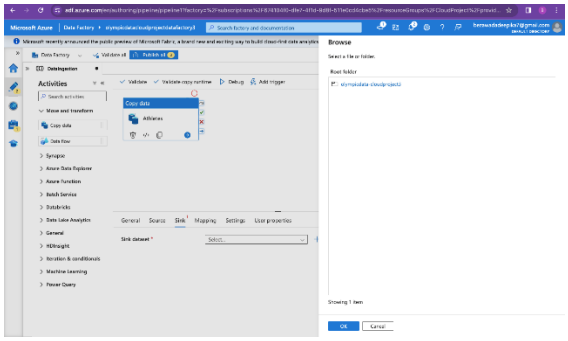


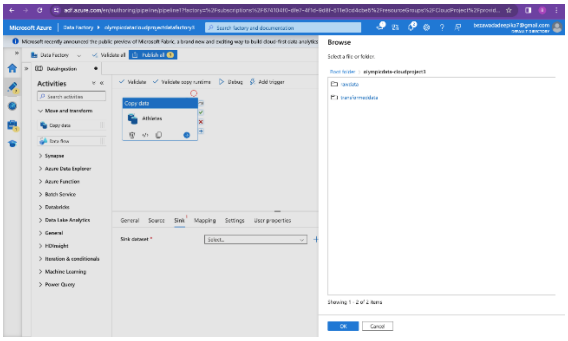


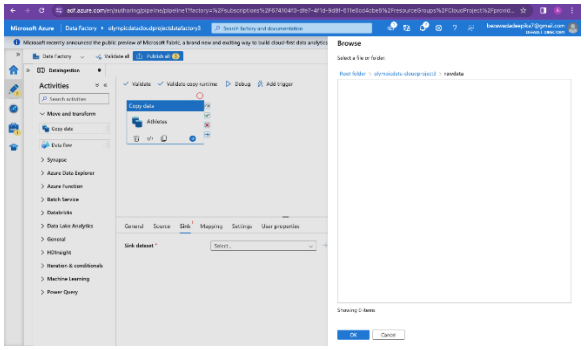
In the "Sink" tab of the "Copy" activity, configure the destination where you want to store the extracted data. Choose Azure Data Lake Store2 and Configure the connection and path to the destination storage.

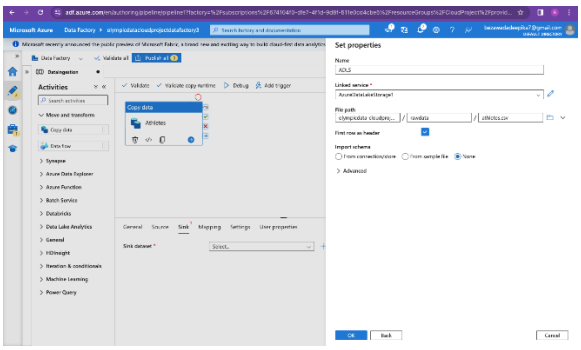




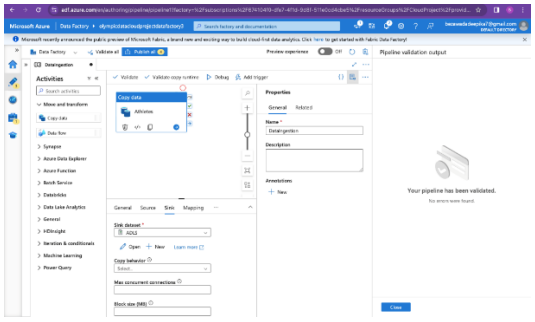


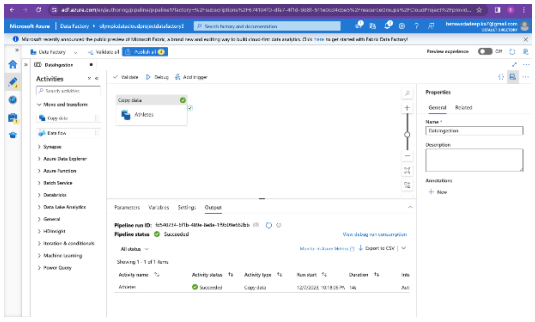




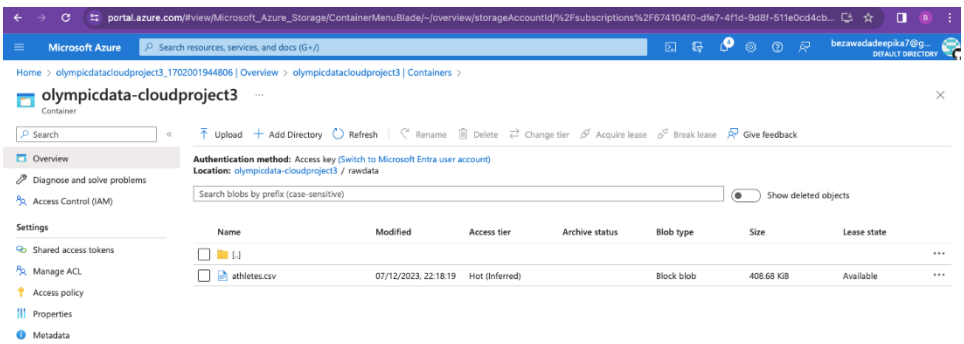


Test and run the pipeline: Click the "Debug" button to test the pipeline with a small dataset. Once satisfied, click the "Publish All" button to make the pipeline available for execution.



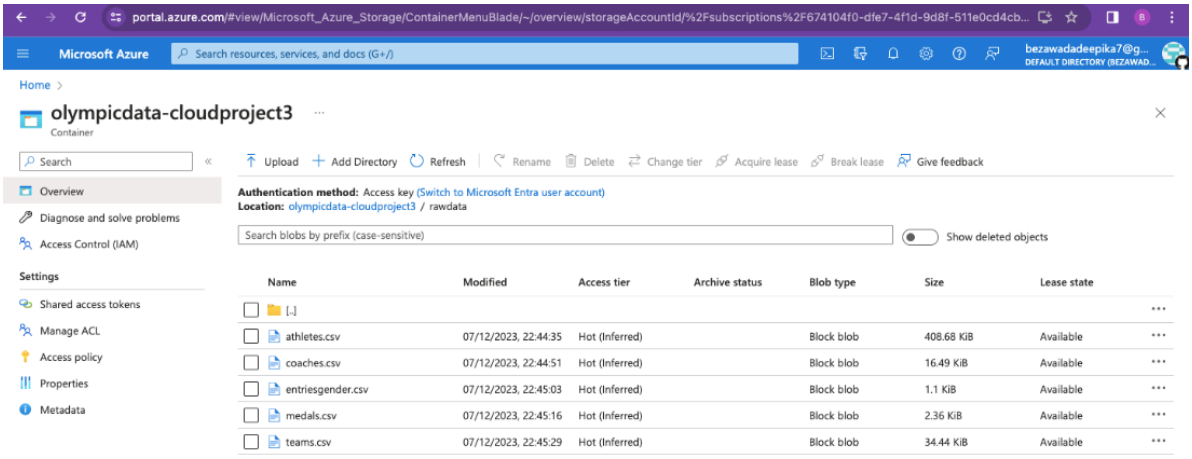


Now we can see athletes.csv is available in our storage account. We were able to extract data from GitHub repository using azure data factory.

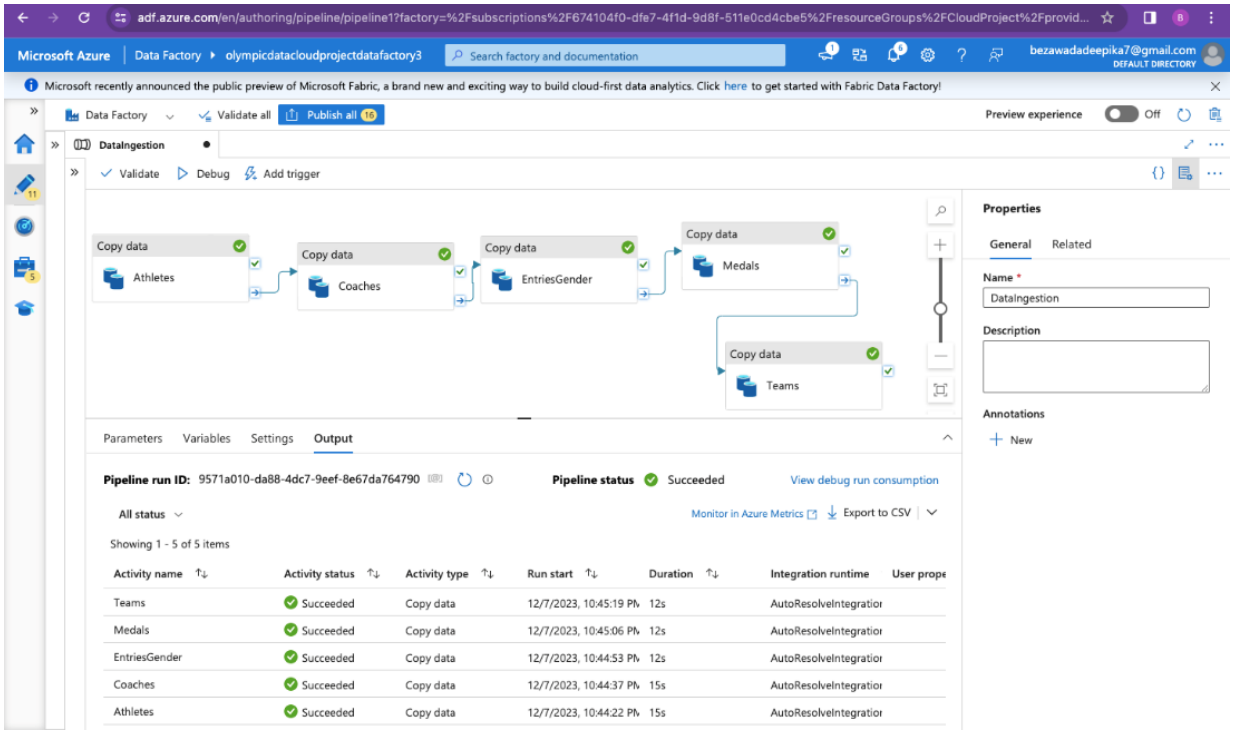


By following these steps for all the five .csv files, we successfully created a data factory pipeline and ingested all the data onto azure.

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



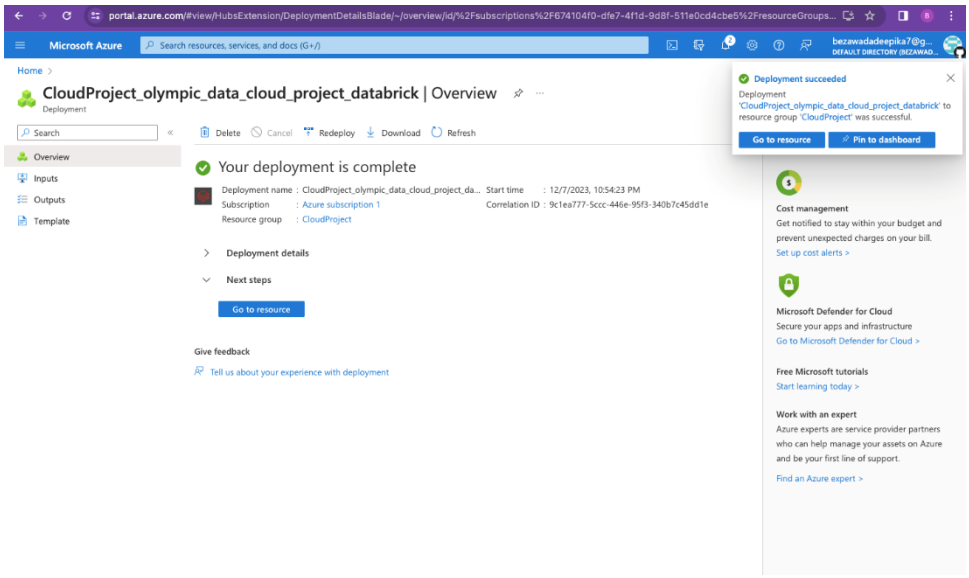
Created a pipeline.



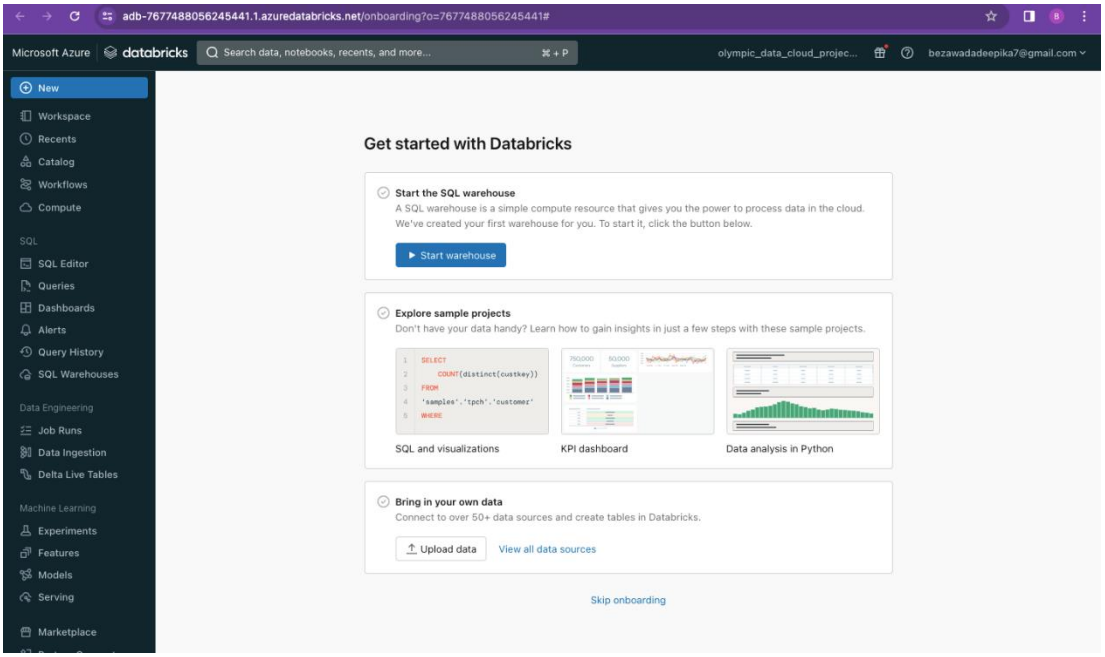
- **Data Transformation:**

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

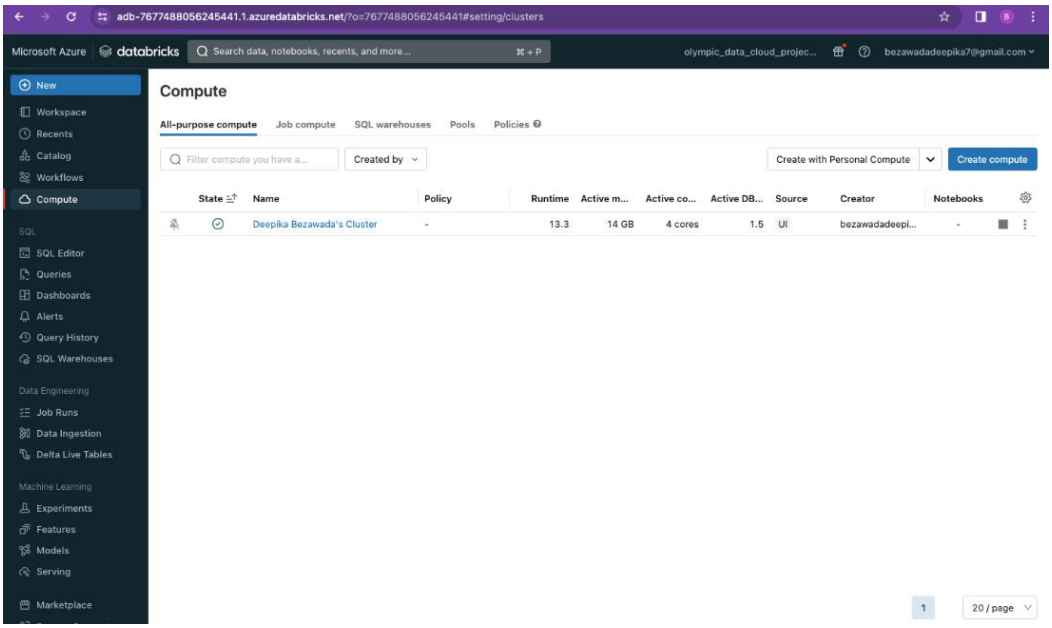
Created azure data bricks service.



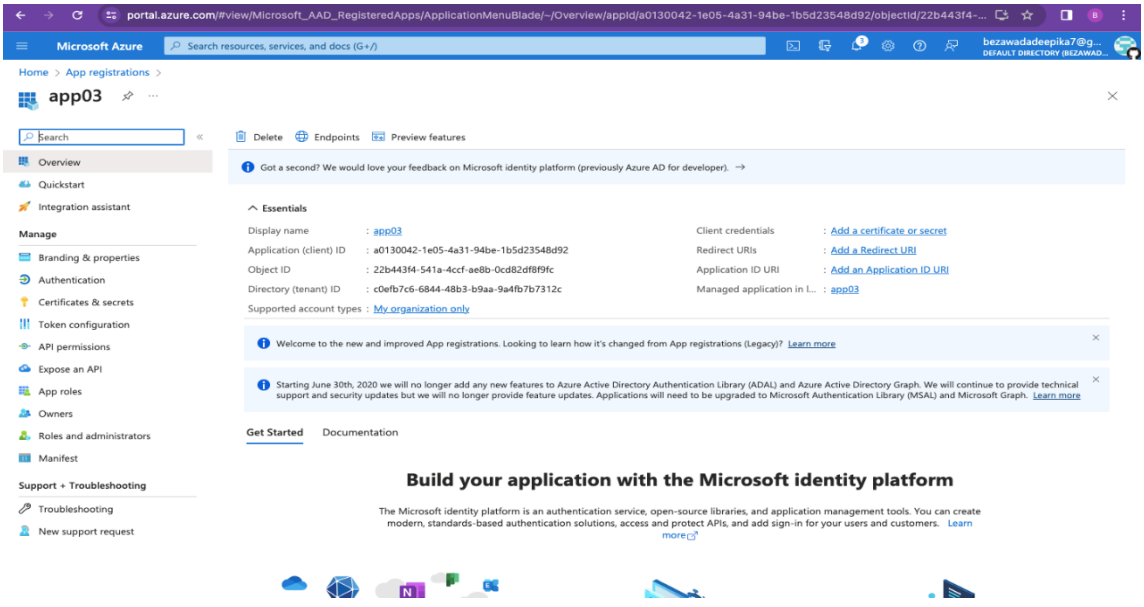
Go to resource and click on launch workspace and we will be redirected to the Azure database workspace where we can create the notebook and instance and so on.



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



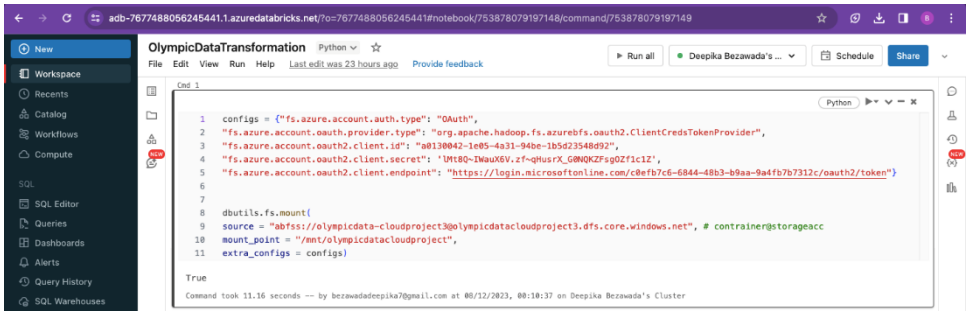
Next, we need to create a connection from azure data bricks to our azure data storage so that we can easily access the data. To create a connection, we need client Id and tenant id. So for that we created app registrations and got those credentials.



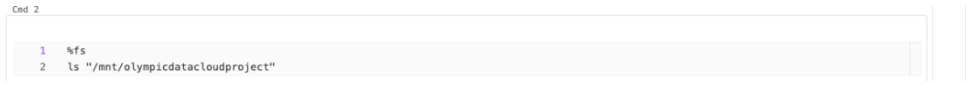
Go to certificates & secrets and get the secret id.

Azure data bricks workspace:

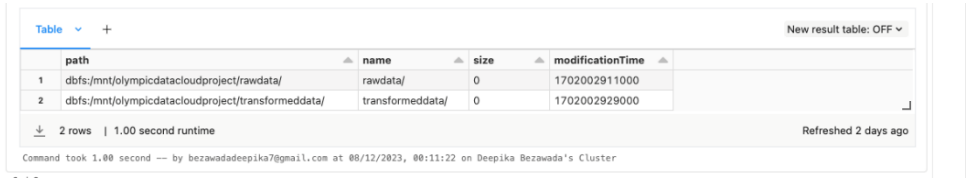
1. Create connection from this azure data bricks to our azure data storage so that we can easily access the data.



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.



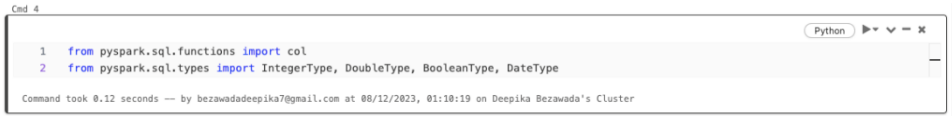
Result: Able to see all files from mount location



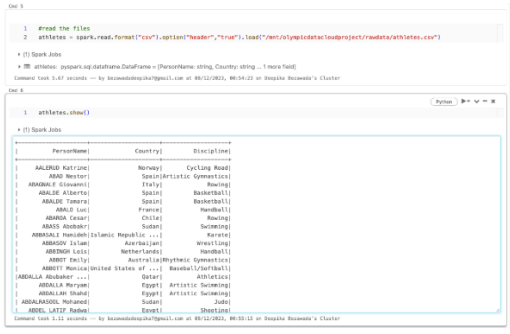
	path	name	size	modificationTime
1	dbfs:/mnt/olympicdatacloudproject/rawdata/	rawdata/	0	1702002911000
2	dbfs:/mnt/olympicdatacloudproject/transformeddata/	transformeddata/	0	1702002929000

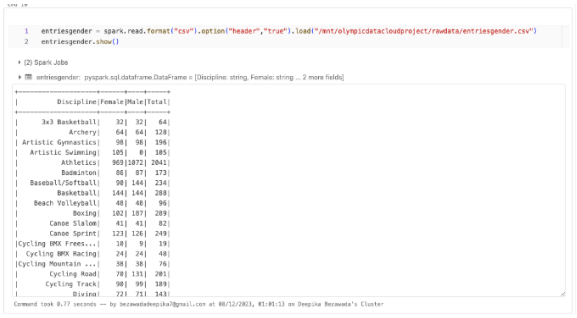
2 rows | 1.00 second runtime — Refreshed 2 days ago

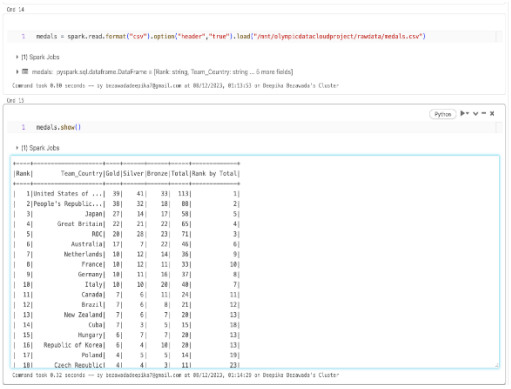
3. Importing specific functions and types from PySpark to work with Spark Data Frames.



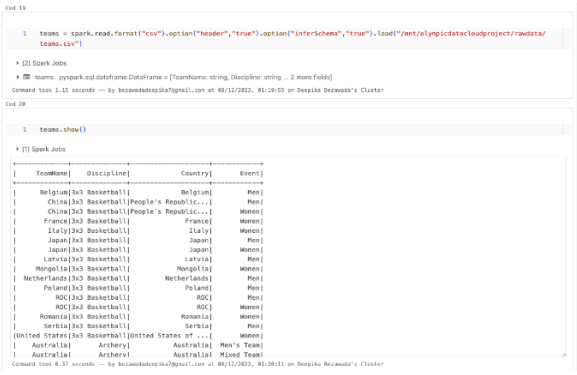
4. Read the files







```
Cmd 14
1  medals = spark.read.format("csv").option("header","true").load("/mnt/olympicdatacloudproject/rawdata/medals.csv")
▸ (1) Spark Jobs
▸ medals: pyspark.sql.dataframe.DataFrame = [Rank: string, Team_Country: string ... 5 more fields]
Command took 0.30 seconds — by bezawadadeepika7@gmail.com at 08/12/2023, 01:13:53 on Deepika Bezawada's Cluster

Cmd 15
1  medals.show()
▸ (1) Spark Jobs
+----+--------------+----+------+------+-----------+
|Rank|  Team_Country|Gold|Silver|Bronze|Rank by Total|
+----+--------------+----+------+------+-----------+
|   1|United States of ...|  39|  41|  33|        113|
|   2|People's Republic...|  38|  32|  18|         88|
|   3|         Japan|  27|  14|  17|         58|
|   4|Great Britain|  22|  21|  22|         65|
|   5|           ROC|  20|  28|  23|         71|
|   6|     Australia|  17|   7|  22|         46|
|   7|   Netherlands|  10|  12|  14|         36|
|   8|        France|  10|  12|  11|         33|
|   9|       Germany|  10|  11|  16|         37|
|  10|         Italy|  10|  10|  20|         40|
|  11|        Canada|   7|   6|  11|         24|
|  12|        Brazil|   7|   6|   8|         21|
|  13|   New Zealand|   7|   6|   7|         20|
|  14|          Cuba|   7|   3|   5|         15|
|  15|       Hungary|   6|   7|   7|         20|
|  16|Republic of Korea|   6|   4|  10|         20|
|  17|        Poland|   4|   5|   5|         14|
|  18|Czech Republic|   4|   4|   3|         11|
+----+--------------+----+------+------+-----------+
Command took 0.32 seconds — by bezawadadeepika7@gmail.com at 08/12/2023, 01:14:20 on Deepika Bezawada's Cluster
```

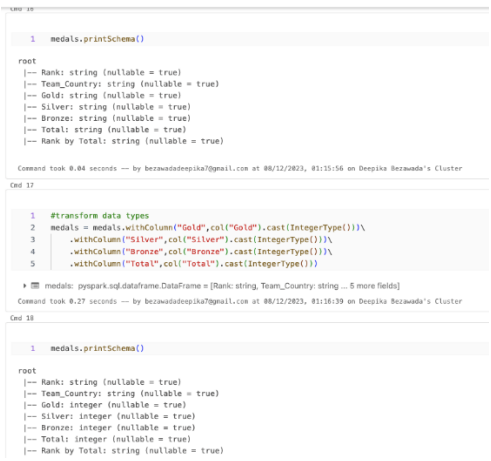


```
Cmd 19
1  teams = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/olympicdatacloudproject/rawdata/
   teams.csv")
▸ (2) Spark Jobs
▸ teams: pyspark.sql.dataframe.DataFrame = [TeamName: string, Discipline: string ... 2 more fields]
Command took 1.15 seconds — by bezawadadeepika7@gmail.com at 08/12/2023, 01:19:55 on Deepika Bezawada's Cluster

Cmd 20
1  teams.show()
▸ (1) Spark Jobs
+--------------+--------------+--------------+-----------+
|      TeamName|    Discipline|       Country|      Event|
+--------------+--------------+--------------+-----------+
|       Belgium|3x3 Basketball|       Belgium|        Men|
|       China|3x3 Basketball|People's Republic...|        Men|
|       China|3x3 Basketball|People's Republic...|      Women|
|        France|3x3 Basketball|        France|      Women|
|         Italy|3x3 Basketball|         Italy|      Women|
|         Japan|3x3 Basketball|         Japan|        Men|
|         Japan|3x3 Basketball|         Japan|      Women|
|        Latvia|3x3 Basketball|        Latvia|        Men|
|      Mongolia|3x3 Basketball|      Mongolia|      Women|
|   Netherlands|3x3 Basketball|   Netherlands|        Men|
|        Poland|3x3 Basketball|        Poland|        Men|
|         ROC|3x3 Basketball|           ROC|        Men|
|         ROC|3x3 Basketball|           ROC|      Women|
|      Romania|3x3 Basketball|       Romania|      Women|
|        Serbia|3x3 Basketball|        Serbia|        Men|
|United States|3x3 Basketball|United States of ...|      Women|
|     Australia|       Archery|     Australia|  Men's Team|
|     Australia|       Archery|     Australia|  Mixed Team|
+--------------+--------------+--------------+-----------+
Command took 0.37 seconds — by bezawadadeepika7@gmail.com at 08/12/2023, 01:20:11 on Deepika Bezawada's Cluster
```

5.Transformed the datatypes

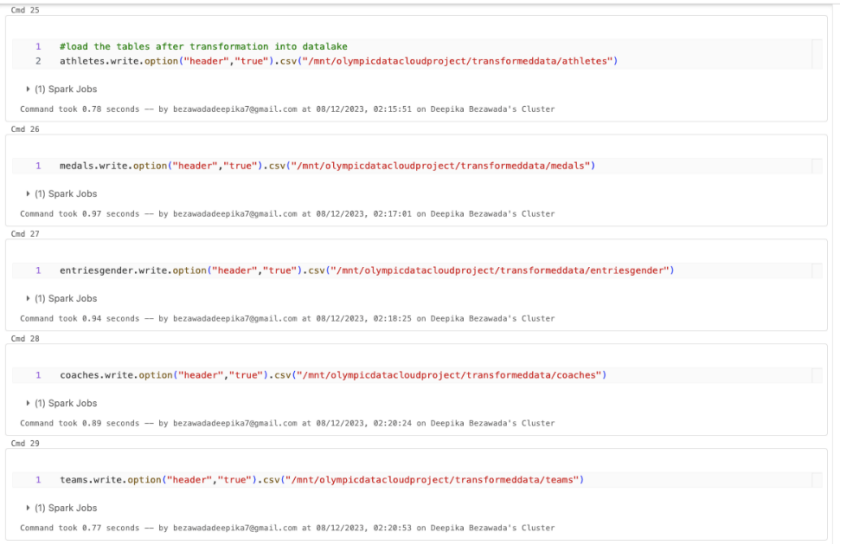
For **Entriesgender.csv and Medals.csv**

```
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 bezawadadeepika7@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: pyspark.sql.dataframe.DataFrame
    Discipline: string
    Female: integer
    Male: integer
    Total: integer
Command took 0.20 seconds — by bezawadadeepika7@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 bezawadadeepika7@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 0.04 seconds — by bezawadadeepika7@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 ... 5 more fields]
Command took 0.27 seconds — by bezawadadeepika7@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 (i5.e the data will be loaded into transformed folder.

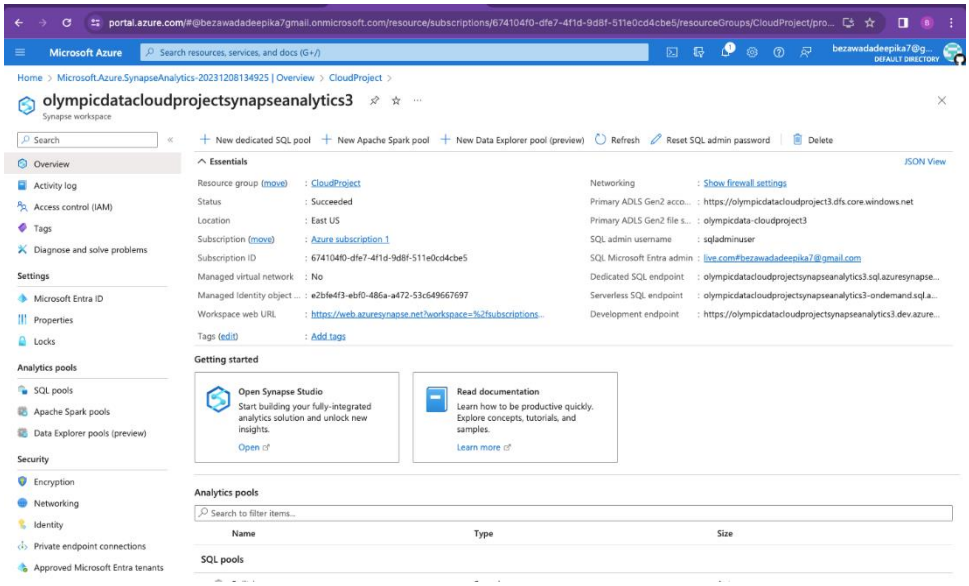


```
Cmd 25
1  #load the tables after transformation into datalake
2  athletes.write.option("header","true").csv("/mnt/olympicdatacloudproject/transformeddata/athletes")
▸ (1) Spark Jobs
Command took 0.78 seconds — by bezawadadeepika7@gmail.com at 08/12/2023, 02:15:51 on Deepika Bezawada's Cluster

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
```

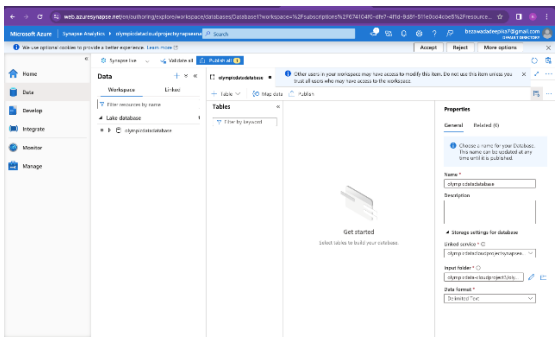
Data Analysis:

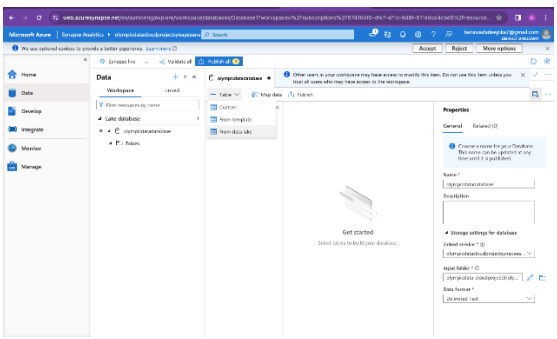
Used azure synapse analytics to perform data analysis. We need to load that transformed data into azure synapse analytics.

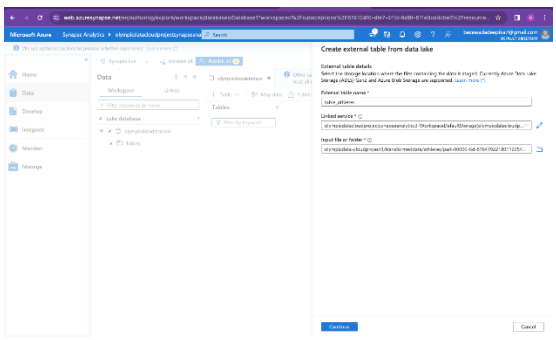
Created azure synapse analytics.

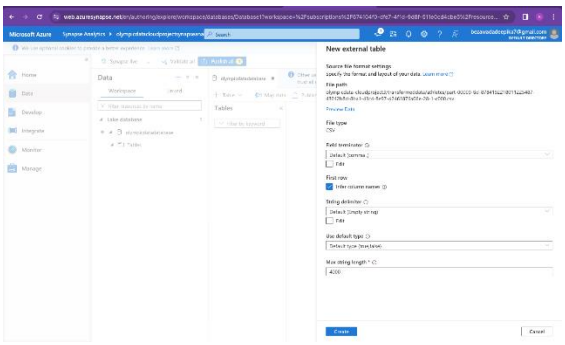


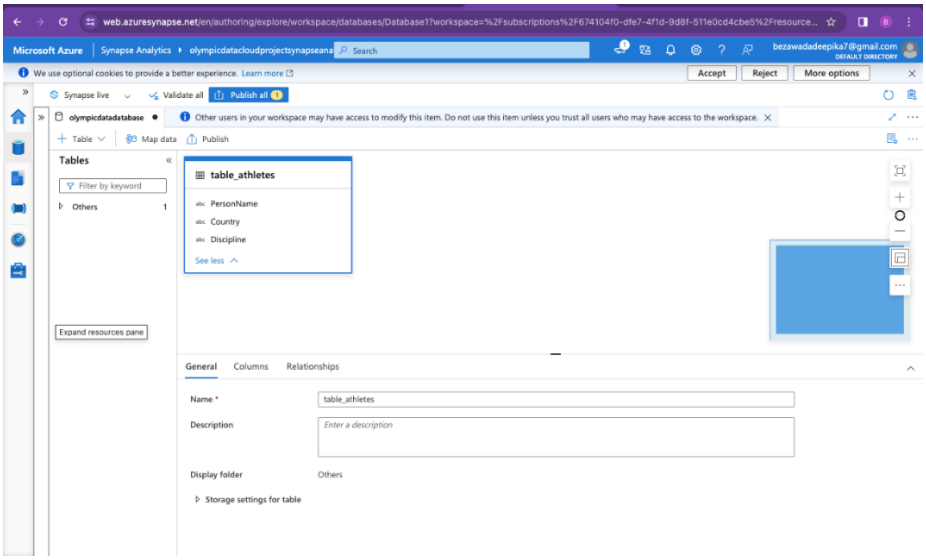
Load the data. For that you can just click onto the data part and click on plus icon and click on lake database.



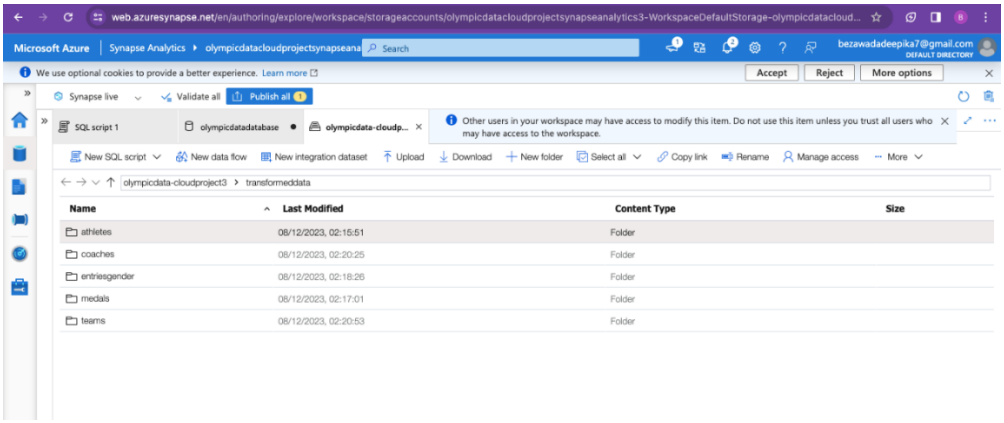






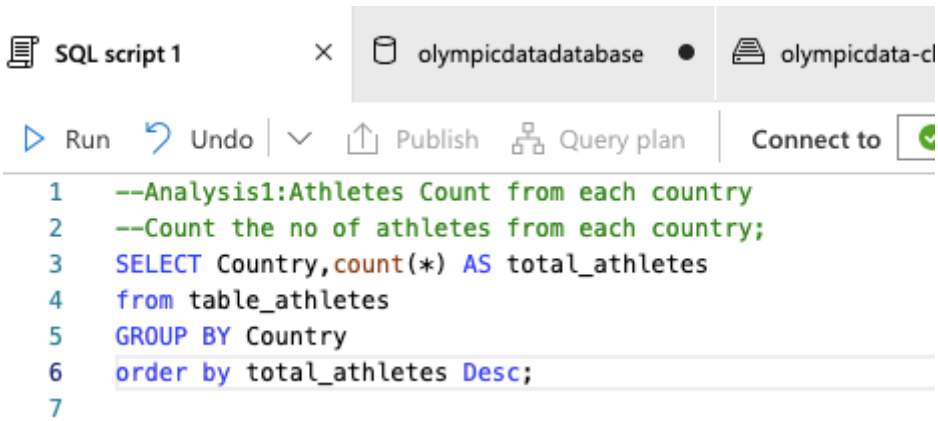


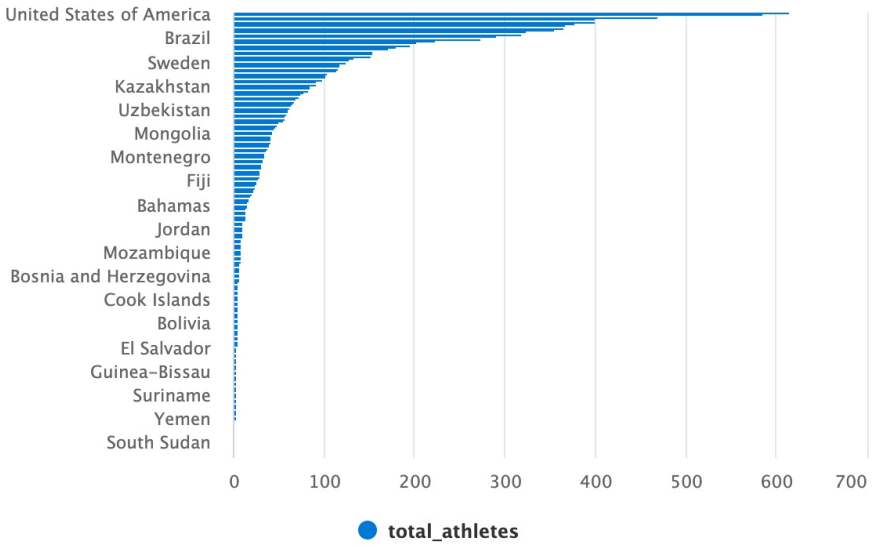
Next repeat the above steps for all the five files to load the data into database. All the tables were loaded into the database. And then validate and publish all.



Analysis part:

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

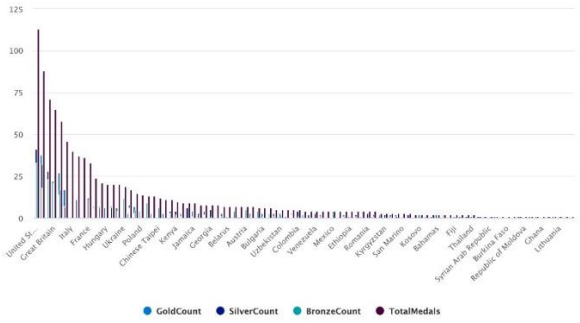


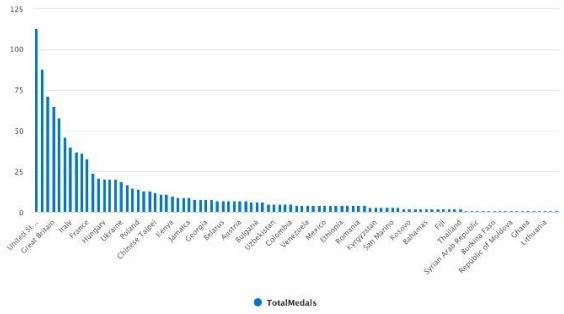


The United States has the most athletes competing in the Olympics by a significant margin, with over 600 athletes participating. Brazil is the second-highest country, with over 400 athletes competing. The remaining countries in the top 10 all have between 100 and 300 athletes competing.
We also observed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

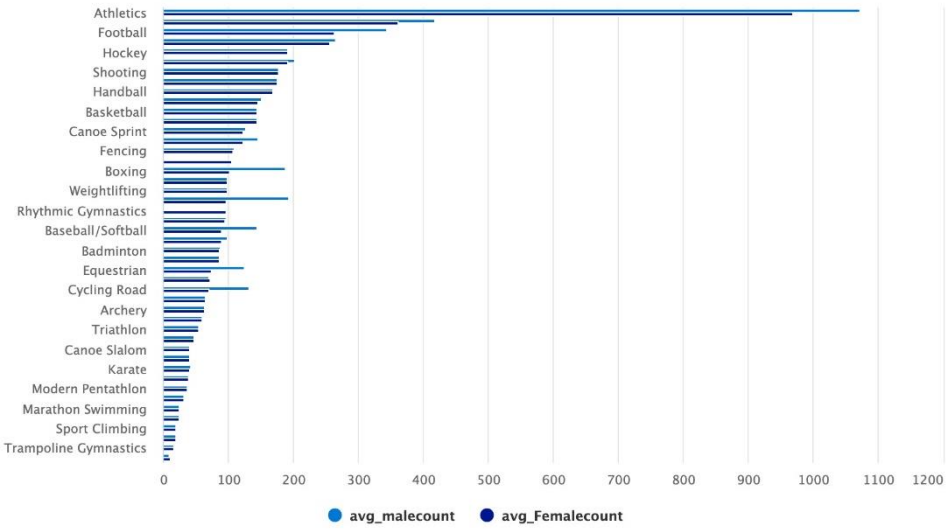




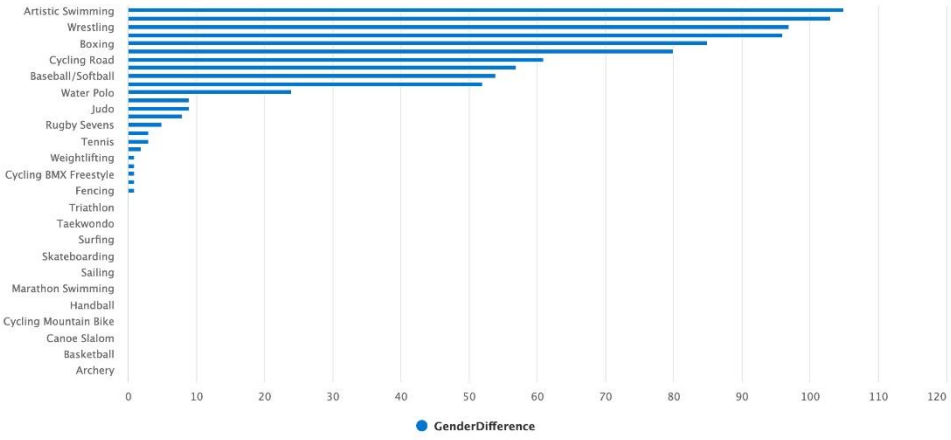
This visualization showed that the United States has been the most successful country in the Olympics (2021) in terms of medals won.

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

```sql
-- Analysis 3: Gender distribution by dsicipline
--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;
```




```sql
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
```

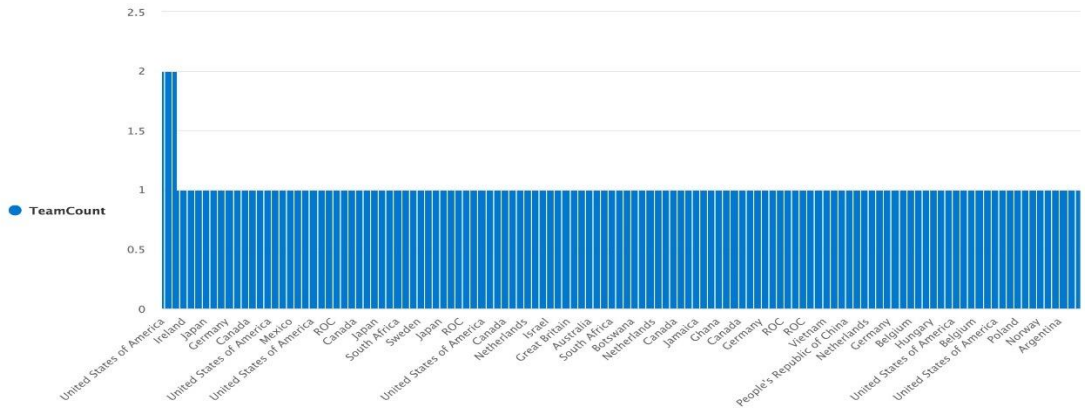


There is a significant disparity in the average number of male and female entries for many disciplines, but there is a growing trend of female participation in many traditionally male-dominated sports.

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

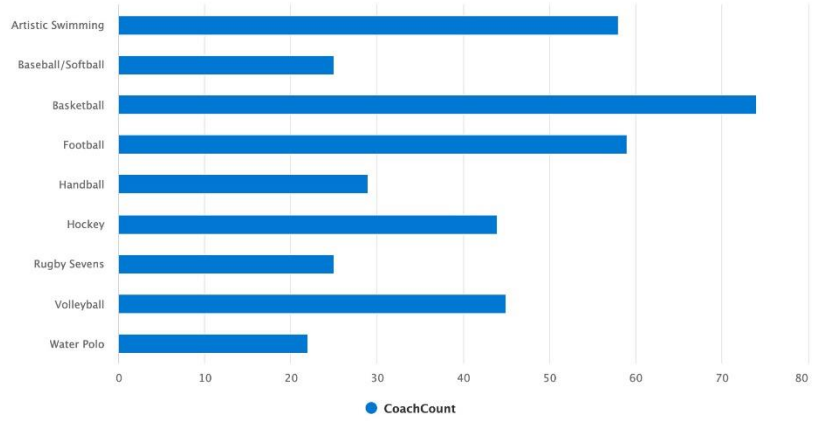
```sql
--Analysis 4: Event participation by teams
--Count of teams participating in each event for different countries and disciplines;
SELECT t.Discipline,
       t.Country,
       t.Event,
       COUNT(*) AS TeamCount
FROM teams t
GROUP BY t.Discipline, t.Country, t.Event
ORDER BY TeamCount DESC;
```

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



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

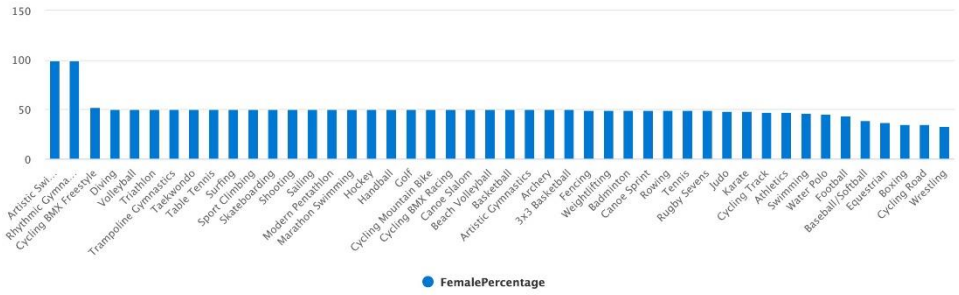
```sql
--Analysis 5: coaches per discipline
SELECT c.Discipline,
       COUNT(DISTINCT c.Name) AS CoachCount
FROM coaches c
GROUP BY c.Discipline;
```



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.

```sql
-- Analysis 6: Discipline with Highest Feamale Participation percentage
WITH TotalGenderCounts AS (
    SELECT Discipline,
           SUM(Female) AS TotalFemale,
           SUM(Male) AS TotalMale
    FROM entriesgender
    GROUP BY Discipline
)
SELECT Discipline,
       (TotalFemale * 100.0) / (TotalFemale + TotalMale) AS FemalePercentage
FROM TotalGenderCounts
ORDER BY FemalePercentage DESC;
```

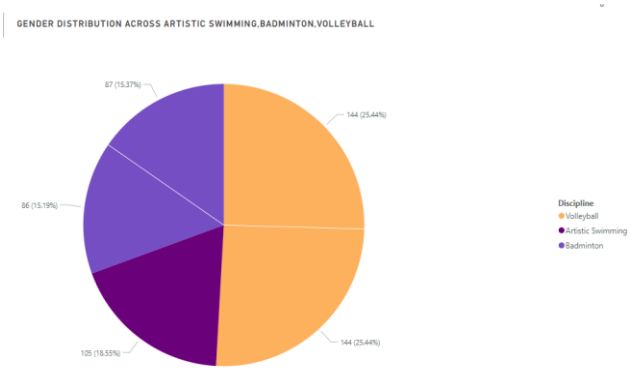


The visualization shows that Artistic Swimming and Rhythmic Gymnastics has the highest female participation percentage.

7. Coaches handling multiple disciplines:

```sql
--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;
```

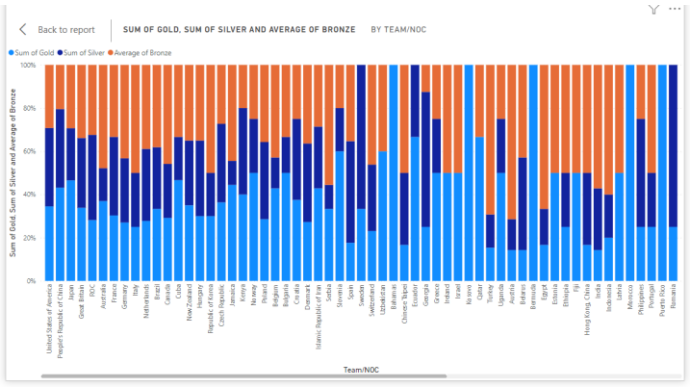
DATA VISUALIZATION:

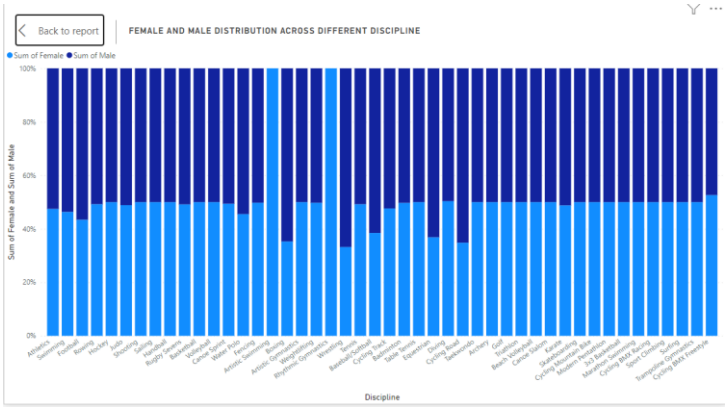


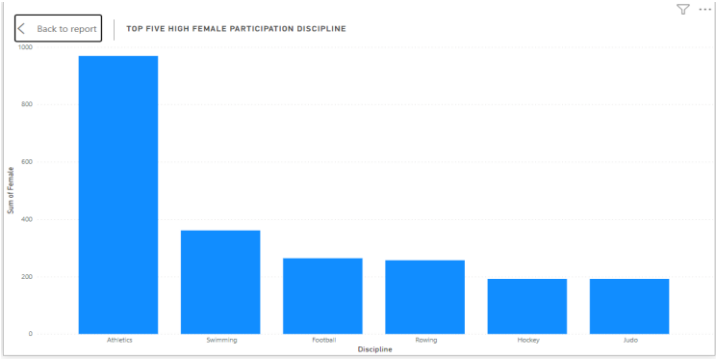
Back to report

TOP FIVE COUNTRY-WISE GOLD,SILVER,BRONZE MEDAL COUNTS

Team/NOC	Sum of Total	Sum of Bronze	Sum of Gold	Sum of Silver
United States of America	113	33	39	41
ROC	71	23	20	28
People's Republic of China	88	18	38	32
Japan	58	17	27	14
Great Britain	65	22	22	21
Total	**395**	**113**	**146**	**136**







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