#1. Import Libraries

```
from pyspark.sql.functions import col
from pyspark.sql.types import IntegerType, DoubleType, BooleanType,
DateType
```

#2. Need to create connections [Databricks <--> Azure Data Lake]

1. Create configuration format
2. Create mount

```
#1. create configuration format

configs = {"fs.azure.account.auth.type": "OAuth",
"fs.azure.account.oauth.provider.type":
"org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
"fs.azure.account.oauth2.client.id": "4e5fde0e-f726-4e0d-9e79-
4685413e2078", #clientid
"fs.azure.account.oauth2.client.secret":
'hWm8Q~Kbovgxczxg2yx.wZl6n_dv4Up.-w8YubU.', #secretid
"fs.azure.account.oauth2.client.endpoint":
"https://login.microsoftonline.com/9075e7fb-176e-4971-ae19-
2443cd22c3f2/oauth2/token"} #tenantid

#2. Mount

dbutils.fs.mount(
source = "abfss://ade-data@adeprojectdata.dfs.core.windows.net/", #
contrainer@storageacc
mount_point = "/mnt/tokyoolympic",
extra_configs = configs)
```

#3. To Check if the mount is successful or not

```
%fs
ls "/mnt/tokyoolympic"
```

#4. Read Each File's Data & Schema

1. Athletes
2. Coaches
3. Entries Gender
4. Medals
5. Teams

```
#filename =
spark.read.format("csv:fileformat").options("header","true").option("i
nferSchema","true").load("@(mountpoint:/mnt/tokyoolympic)/
@datalakename/athletes.csv")
```

```
athletes =
spark.read.format("csv").option("header","true").option("inferSchema",
"true").load("/mnt/tokyoolympic/Raw Data Lake/Athletes.csv")
coaches =
spark.read.format("csv").option("header","true").option("inferSchema",
"true").load("/mnt/tokyoolympic/Raw Data Lake/Coaches.csv")
entriesgender =
spark.read.format("csv").option("header","true").option("inferSchema",
"true").load("/mnt/tokyoolympic/Raw Data Lake/EntriesGender.csv")
medals =
spark.read.format("csv").option("header","true").option("inferSchema",
"true").load("/mnt/tokyoolympic/Raw Data Lake/Medals.csv")
teams =
spark.read.format("csv").option("header","true").option("inferSchema",
"true").load("/mnt/tokyoolympic/Raw Data Lake/Teams.csv")


#1. Athletes Data

athletes.show()

+--------------------+--------------------+-------------------+
|          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 ...|             Karate|
|       ABBASOV Islam|          Azerbaijan|          Wrestling|
|        ABBINGH Lois|         Netherlands|           Handball|
|         ABBOT Emily|           Australia|Rhythmic Gymnastics|
|       ABBOTT Monica|United States of ...|  Baseball/Softball|
|ABDALLA Abubaker ...|               Qatar|          Athletics|
|      ABDALLA Maryam|               Egypt|  Artistic Swimming|
|      ABDALLAH Shahd|               Egypt|  Artistic Swimming|
| ABDALRASOOL Mohamed|               Sudan|               Judo|
|   ABDEL LATIF Radwa|               Egypt|           Shooting|
|    ABDEL RAZEK Samy|               Egypt|           Shooting|
|   ABDELAZIZ Abdalla|               Egypt|             Karate|
+--------------------+--------------------+-------------------+
only showing top 20 rows
```

```
#Schema

athletes.printSchema()

root
 |-- PersonName: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- Discipline: string (nullable = true)


#2. Coaches Data

coaches.show()

+--------------------+--------------------+-----------------+--------+
|                Name|             Country|       Discipline|   Event|
+--------------------+--------------------+-----------------+--------+
|     ABDELMAGID Wael|               Egypt|         Football|    null|
|           ABE Junya|               Japan|       Volleyball|    null|
|       ABE Katsuhiko|               Japan|       Basketball|    null|
|        ADAMA Cherif|       C�te d'Ivoire|         Football|    null|
|          AGEBA Yuya|               Japan|       Volleyball|    null|
|AIKMAN Siegfried ...|               Japan|           Hockey|     Men|
|       AL SAADI Kais|             Germany|           Hockey|     Men|
|       ALAMEDA Lonni|              Canada|Baseball/Softball|Softball|
|     ALEKNO Vladimir|Islamic Republic ...|       Volleyball|     Men|
|     ALEKSEEV Alexey|                 ROC|         Handball|   Women|
|ALLER CARBALLO Ma...|               Spain|       Basketball|    null|
|       ALSHEHRI Saad|        Saudi Arabia|         Football|     Men|
|           ALY Kamal|               Egypt|         Football|    null|
| AMAYA GAITAN Fabian|         Puerto Rico|       Basketball|    null|
|    AMO AGUADO Pablo|               Spain|         Football|    null|
|   ANDONOVSKI Vlatko|United States of ...|         Football|   Women|
|        ANNAN Alyson|         Netherlands|           Hockey|   Women|
|  ARNAU CREUS Xavier|               Japan|           Hockey|   Women|
|       ARNOLD Graham|           Australia|         Football|     Men|
|         AXNER Tomas|              Sweden|         Handball|   Women|
+--------------------+--------------------+-----------------+--------+
only showing top 20 rows


#Schema

coaches.printSchema()

root
 |-- Name: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- Discipline: string (nullable = true)
 |-- Event: string (nullable = true)
```

```
#3. EntriesGender Data

entriesgender

+--------------------+------+----+-----+
|          Discipline|Female|Male|Total|
+--------------------+------+----+-----+
|      3x3 Basketball|    32|  32|   64|
|             Archery|    64|  64|  128|
| Artistic Gymnastics|    98|  98|  196|
|   Artistic Swimming|   105|   0|  105|
|           Athletics|   969|1072| 2041|
|           Badminton|    86|  87|  173|
|   Baseball/Softball|    90| 144|  234|
|          Basketball|   144| 144|  288|
|    Beach Volleyball|    48|  48|   96|
|              Boxing|   102| 187|  289|
|        Canoe Slalom|    41|  41|   82|
|        Canoe Sprint|   123| 126|  249|
|Cycling BMX Frees...|    10|   9|   19|
|  Cycling BMX Racing|    24|  24|   48|
|Cycling Mountain ...|    38|  38|   76|
|        Cycling Road|    70| 131|  201|
|       Cycling Track|    90|  99|  189|
|              Diving|    72|  71|  143|
|          Equestrian|    73| 125|  198|
|             Fencing|   107| 108|  215|
+--------------------+------+----+-----+
only showing top 20 rows


#Schema

entriesgender.printSchema()

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

Obervation: As we can see in the output that Female & Male should have Integer datatype, but it is string. Therefore we need to change the datatype from string to integer.

```
#To fix the datatype from string to integer. [run 1 for col]

entriesgender =
entriesgender.withColumn("Female",col("Female").cast(IntegerType()))\
    .withColumn("Male",col("Male").cast(IntegerType()))\
    .withColumn("Total",col("Total").cast(IntegerType()))


#After fixing the data type

entriesgender.printSchema()

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


#4. Medals Data

medals.show()

+----+--------------------+----+------+------+-----+-------------+
|Rank|         TeamCountry|Gold|Silver|Bronze|Total|Rank by Total|
+----+--------------------+----+------+------+-----+-------------+
|   1|United States of ...|  39|    41|    33|  113|            1|
|   2|People's Republic...|  38|    32|    18|   88|            2|
|   3|               Japan|  27|    14|    17|   58|            5|
|   4|       Great Britain|  22|    21|    22|   65|            4|
|   5|                 ROC|  20|    28|    23|   71|            3|
|   6|           Australia|  17|     7|    22|   46|            6|
|   7|         Netherlands|  10|    12|    14|   36|            9|
|   8|              France|  10|    12|    11|   33|           10|
|   9|             Germany|  10|    11|    16|   37|            8|
|  10|               Italy|  10|    10|    20|   40|            7|
|  11|              Canada|   7|     6|    11|   24|           11|
|  12|              Brazil|   7|     6|     8|   21|           12|
|  13|         New Zealand|   7|     6|     7|   20|           13|
|  14|                Cuba|   7|     3|     5|   15|           18|
|  15|             Hungary|   6|     7|     7|   20|           13|
|  16|   Republic of Korea|   6|     4|    10|   20|           13|
|  17|              Poland|   4|     5|     5|   14|           19|
|  18|      Czech Republic|   4|     4|     3|   11|           23|
|  19|               Kenya|   4|     4|     2|   10|           25|
|  20|              Norway|   4|     2|     2|    8|           29|
+----+--------------------+----+------+------+-----+-------------+
only showing top 20 rows


#Schema

medals.printSchema()

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


#5. Teams Data

teams.show()

+-------------+-------------+--------------------+------------+
|     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|
|   Bangladesh|      Archery|          Bangladesh|  Mixed Team|
|      Belarus|      Archery|             Belarus|Women's Team|
+-------------+-------------+--------------------+------------+
only showing top 20 rows


#Schema

teams.printSchema()

root
 |-- TeamName: string (nullable = true)
 |-- Discipline: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- Event: string (nullable = true)
```

All done for the files and schema

#5. Few Insights Extraction

```
#Find countries with highest number of gold medals.

top_gold_medal_by_country =
medals.select("TeamCountry","Gold").orderBy("Gold",
ascending=False).show()

+--------------------+----+
|         TeamCountry|Gold|
+--------------------+----+
|United States of ...|  39|
|People's Republic...|  38|
|               Japan|  27|
|       Great Britain|  22|
|                 ROC|  20|
|           Australia|  17|
|         Netherlands|  10|
|              France|  10|
|             Germany|  10|
|               Italy|  10|
|              Canada|   7|
|              Brazil|   7|
|         New Zealand|   7|
|                Cuba|   7|
|             Hungary|   6|
|   Republic of Korea|   6|
|              Poland|   4|
|      Czech Republic|   4|
|               Kenya|   4|
|              Norway|   4|
+--------------------+----+
only showing top 20 rows


# Calculate the average number of entries by gender for each
discipline

average_entries_by_gender = entriesgender.withColumn(
    'Avg_Female', entriesgender['Female'] / entriesgender['Total']
).withColumn(
    'Avg_Male', entriesgender['Male'] / entriesgender['Total']
)
average_entries_by_gender.show()

+--------------------+------+----+-----+------------------
+-------------------+
|          Discipline|Female|Male|Total|         Avg_Female|
Avg_Male|
+--------------------+------+----+-----+------------------
```

```
+-------------------+
|     3x3 Basketball|    32|  32|   64|                0.5|
0.5|
|            Archery|    64|  64|  128|                0.5|
0.5|
| Artistic Gymnastics|    98|  98|  196|                0.5|
0.5|
|   Artistic Swimming|   105|   0|  105|                1.0|
0.0|
|          Athletics|   969|1072| 2041| 0.4747672709456149|
0.5252327290543851|
|          Badminton|    86|  87|  173|0.49710982658959535|
0.5028901734104047|
|   Baseball/Softball|    90| 144|  234|0.38461538461538464|
0.6153846153846154|
|         Basketball|   144| 144|  288|                0.5|
0.5|
|    Beach Volleyball|    48|  48|   96|                0.5|
0.5|
|             Boxing|   102| 187|  289|0.35294117647058826|
0.6470588235294118|
|       Canoe Slalom|    41|  41|   82|                0.5|
0.5|
|       Canoe Sprint|   123| 126|  249| 0.4939759036144578|
0.5060240963855421|
|Cycling BMX Frees...|    10|   9|   19| 0.5263157894736842|
0.47368421052631576|
|  Cycling BMX Racing|    24|  24|   48|                0.5|
0.5|
|Cycling Mountain ...|    38|  38|   76|                0.5|
0.5|
|       Cycling Road|    70| 131|  201| 0.3482587064676617|
0.6517412935323383|
|      Cycling Track|    90|  99|  189|0.47619047619047616|
0.5238095238095238|
|             Diving|    72|  71|  143| 0.5034965034965035|
0.4965034965034965|
|         Equestrian|    73| 125|  198| 0.3686868686868687|
0.6313131313131313|
|            Fencing|   107| 108|  215|0.49767441860465117|
0.5023255813953489|
+-------------------+------+----+-----+-------------------
+-------------------+
only showing top 20 rows
```

All the connections, reading, transformation and analysis done. NEXT --> dump the transformed data into Transformed Data Lake

#Move the Transformed Cleaned Data to Transformed Data Lake.

```
#format
#filename.repartition(1).write.mode("overwrite").option("header",'true
').csv("/mnt/tokyoolympic/@transformed Data Lake/filename")

athletes.repartition(1).write.mode("overwrite").option("header",'true'
).csv("/mnt/tokyoolympic/Transformed Data Lake/Athletes")
coaches.repartition(1).write.mode("overwrite").option("header","true")
.csv("/mnt/tokyoolympic/Transformed Data Lake/Coaches")
entriesgender.repartition(1).write.mode("overwrite").option("header","
true").csv("/mnt/tokyoolympic/Transformed Data Lake/EntriesGender")
medals.repartition(1).write.mode("overwrite").option("header","true").
csv("/mnt/tokyoolympic/Transformed Data Lake/Medals")
teams.repartition(1).write.mode("overwrite").option("header","true").c
sv("/mnt/tokyoolympic/Transformed Data Lake/Teams")
```

Pulled the Data from 'Raw Data Lake" -> Transformed in Databricks using Apache Spark -> Dumped the transformed data in "Transformed Data Lake"

Next: Transformed ADLS --> Azure Synapse