Power BI

TOKYO 2020

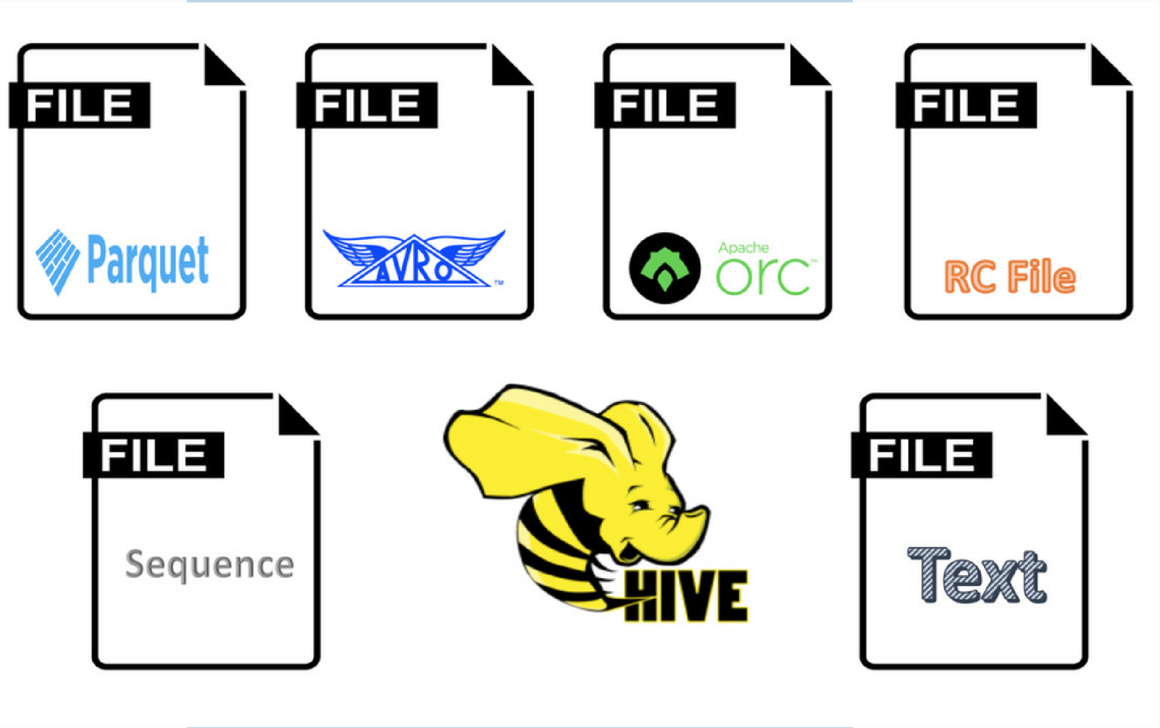
BUILD 2020 OLYMPIC GAMES DATA PROCESSING PIPELINE ON AZURE

Azure

Mentor/Intern: Nguyen Ngoc Thien / Tran Quoc Hai
Le Trung Viet / Tran Thi Anh Thu
Phan Thien Huu

TABLE OF CONTENT

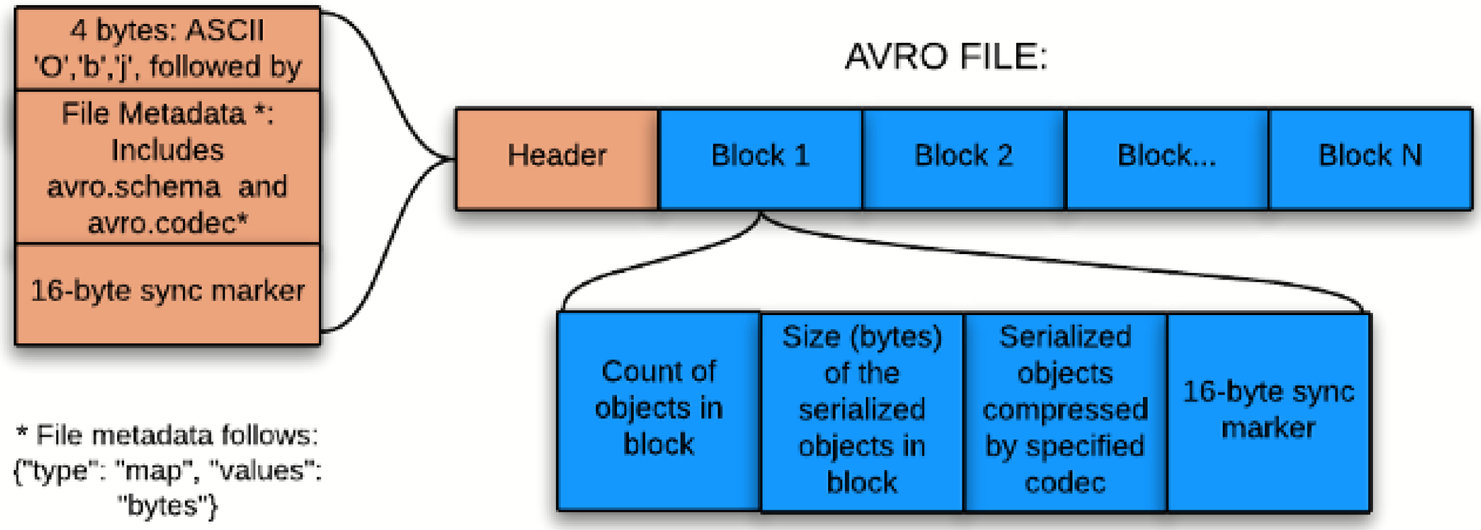
BIG DATA FORMAT



AVRO

Apache Avro is a data serialization format.

Apache Avro is a row-based storage.



AVRO

Flexibility

allows adding, editing, deleting

Compact size

stored as compressed binary

High speed access

provides efficient and fast write-to-read

Multi-language support

Java, python, c++, and many more languages

Easy integration

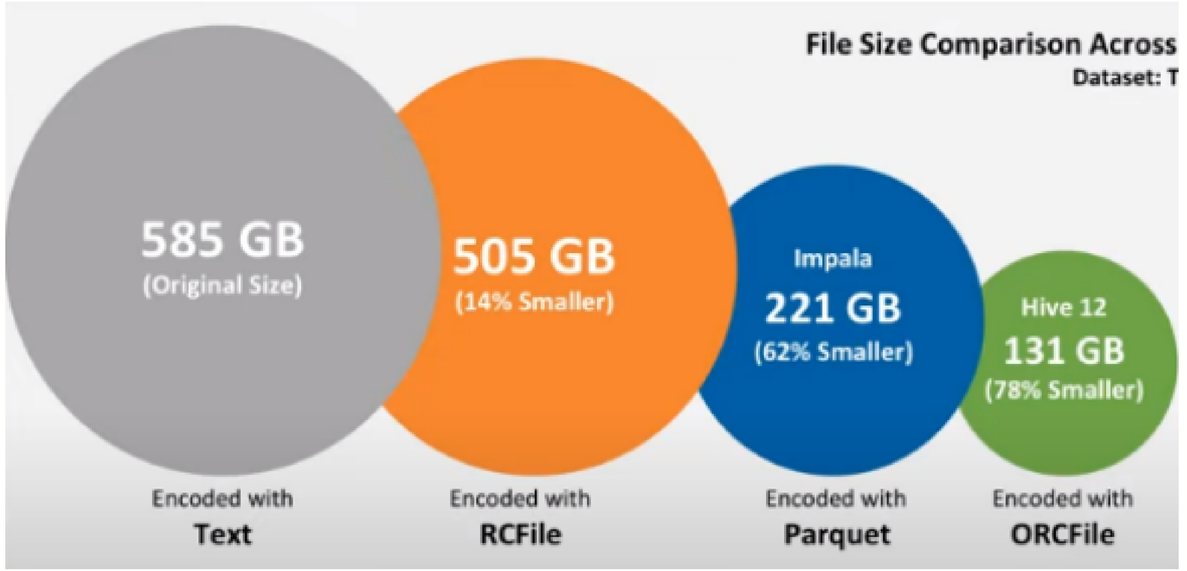
Hadoop, Spark, Kafka

ORC

- **Column-based**

Tên	Địa chỉ	SĐT	Email	Ngày sinh	Tổng tiền đã mua
John Smith	123 Main st	12345	john@...	1985-07-10	100000
Jane Doe	456 Oak St	6789	jane@...	1990-07-02	2000000

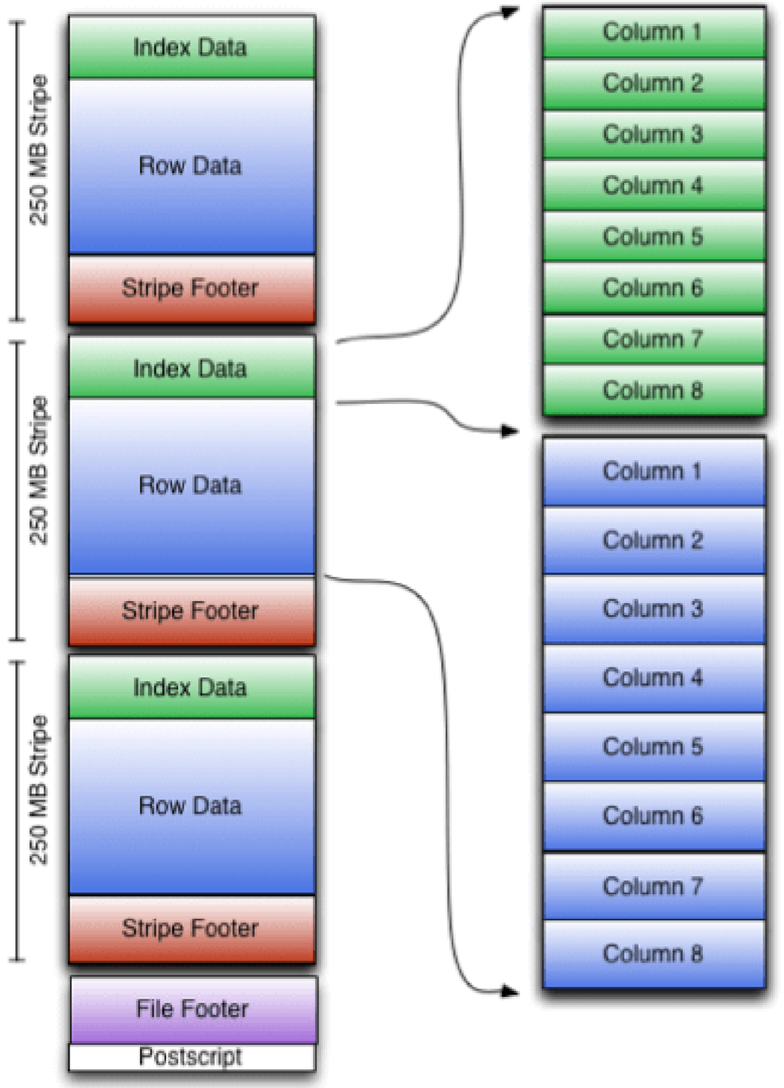
- **Compression**



ORC

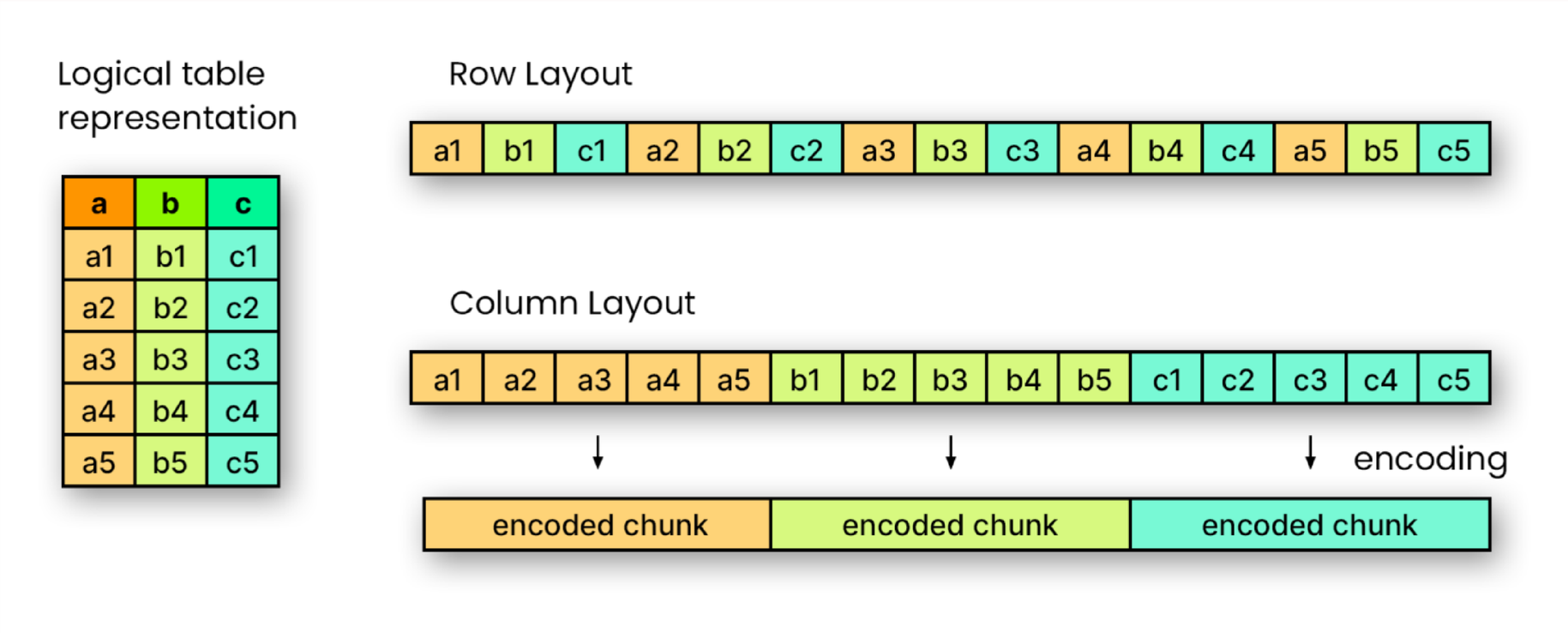
Purpose:

- Speed Up Query
- Save Storage Space
- Efficiency in Big Data Processing
- Support Read



PARQUET

Apache Parquet is a column-oriented data file format



PARQUET

Free & Open source file

Language Agnostic

Column-based format

Highly efficent

Supports complex data types

BENEFITS

Stores big data of any kind

Saves on cloud storage space

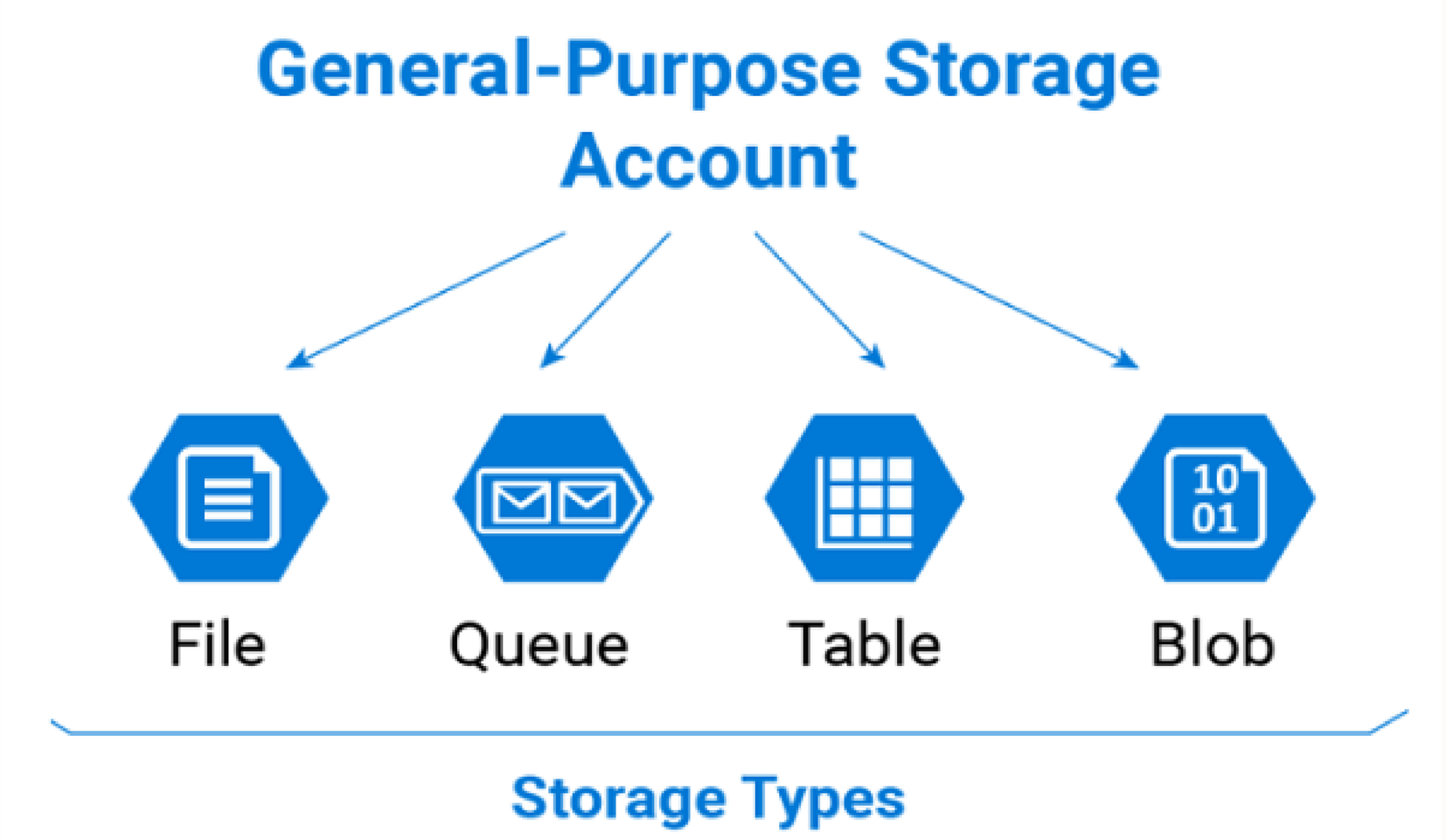
Increases data throughput & performance

COMPARISON

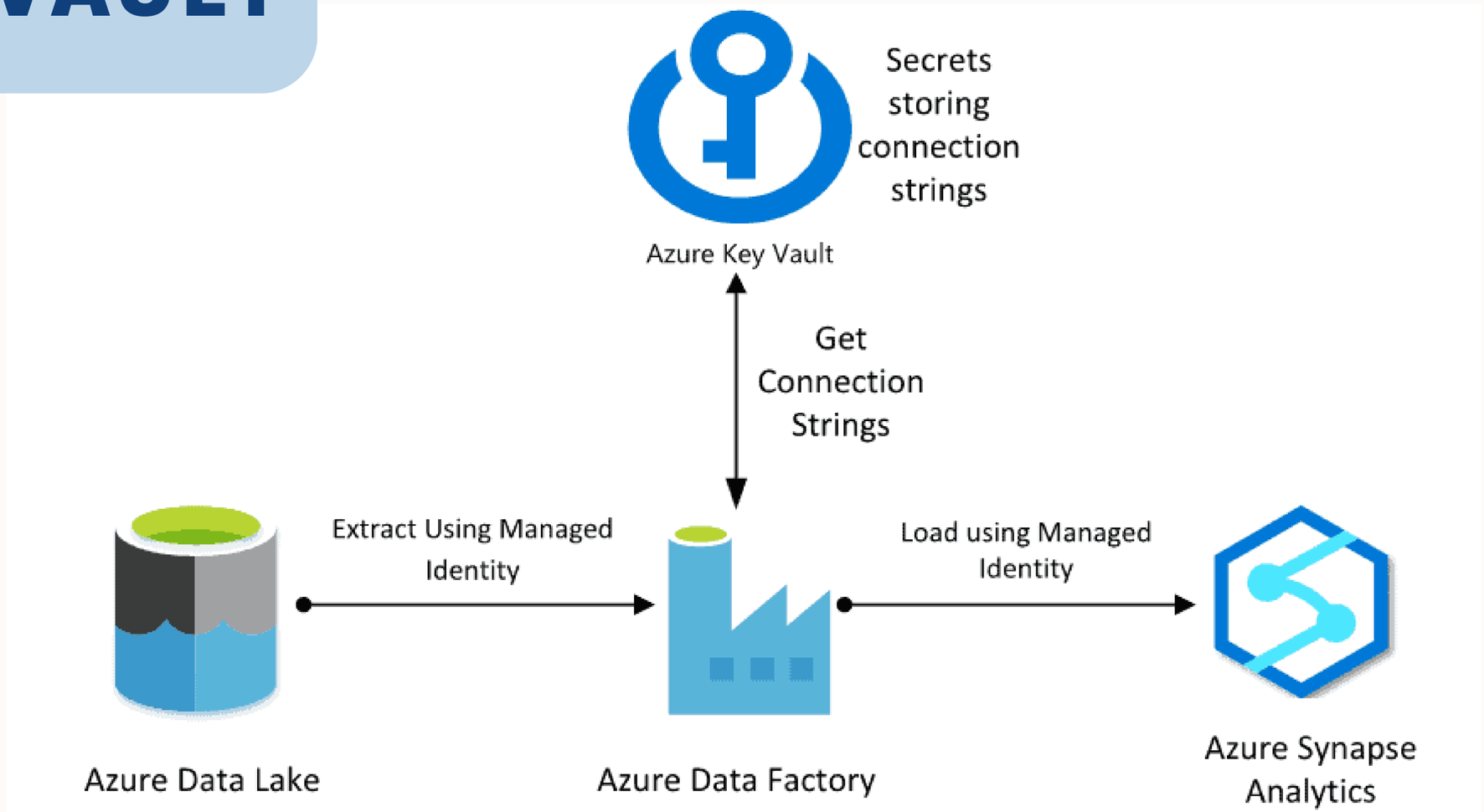
	AVRO	PARQUET	ORC
Schema Evolution	Best	Good	Better
Compression	Good	Better	Best
Row/Column Oriented	Row	Column	Column
Read/Write	Write	Read	Read
Splittability	Good	Good	Best

PIPELINE & DEMO

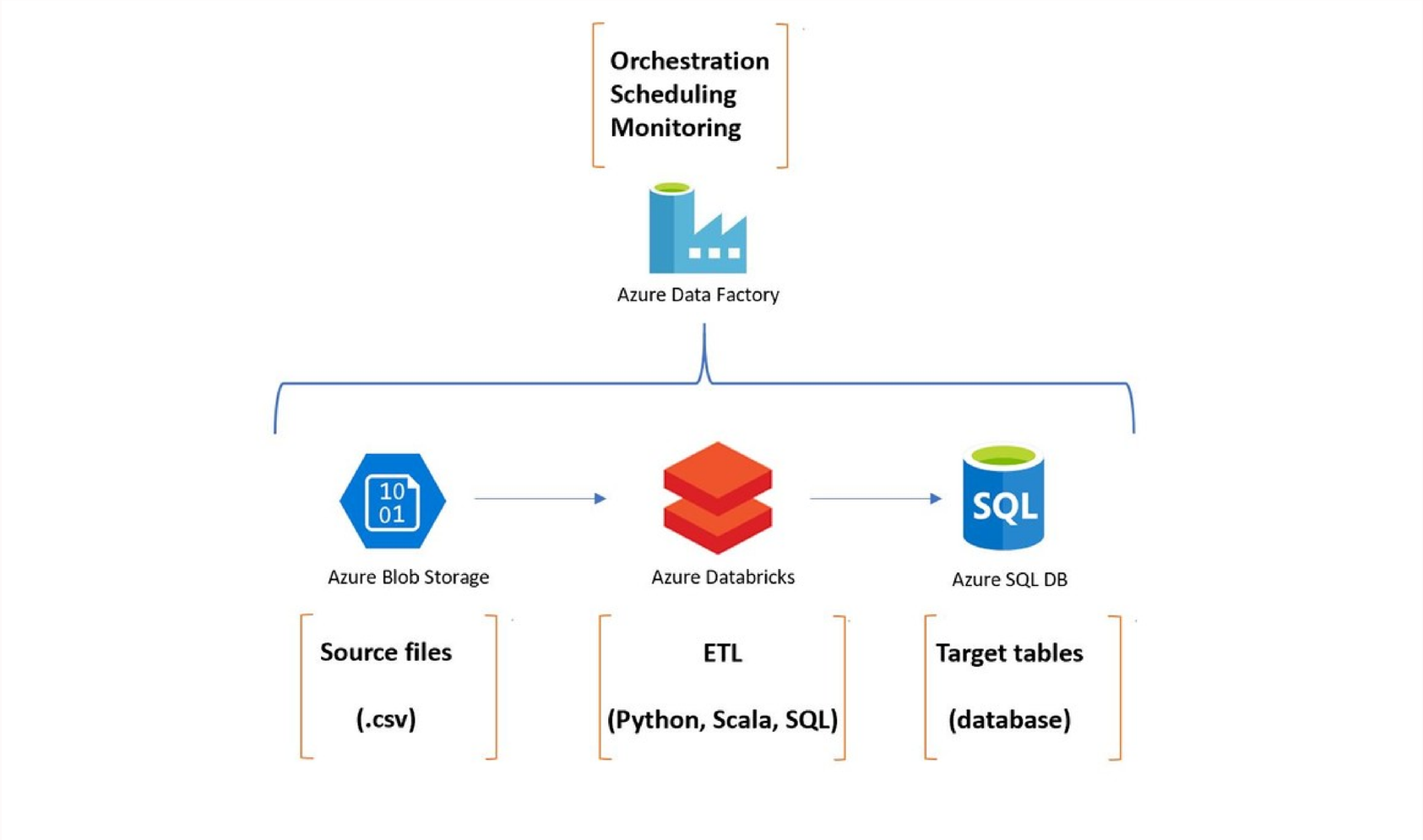
STORAGE ACCOUNT



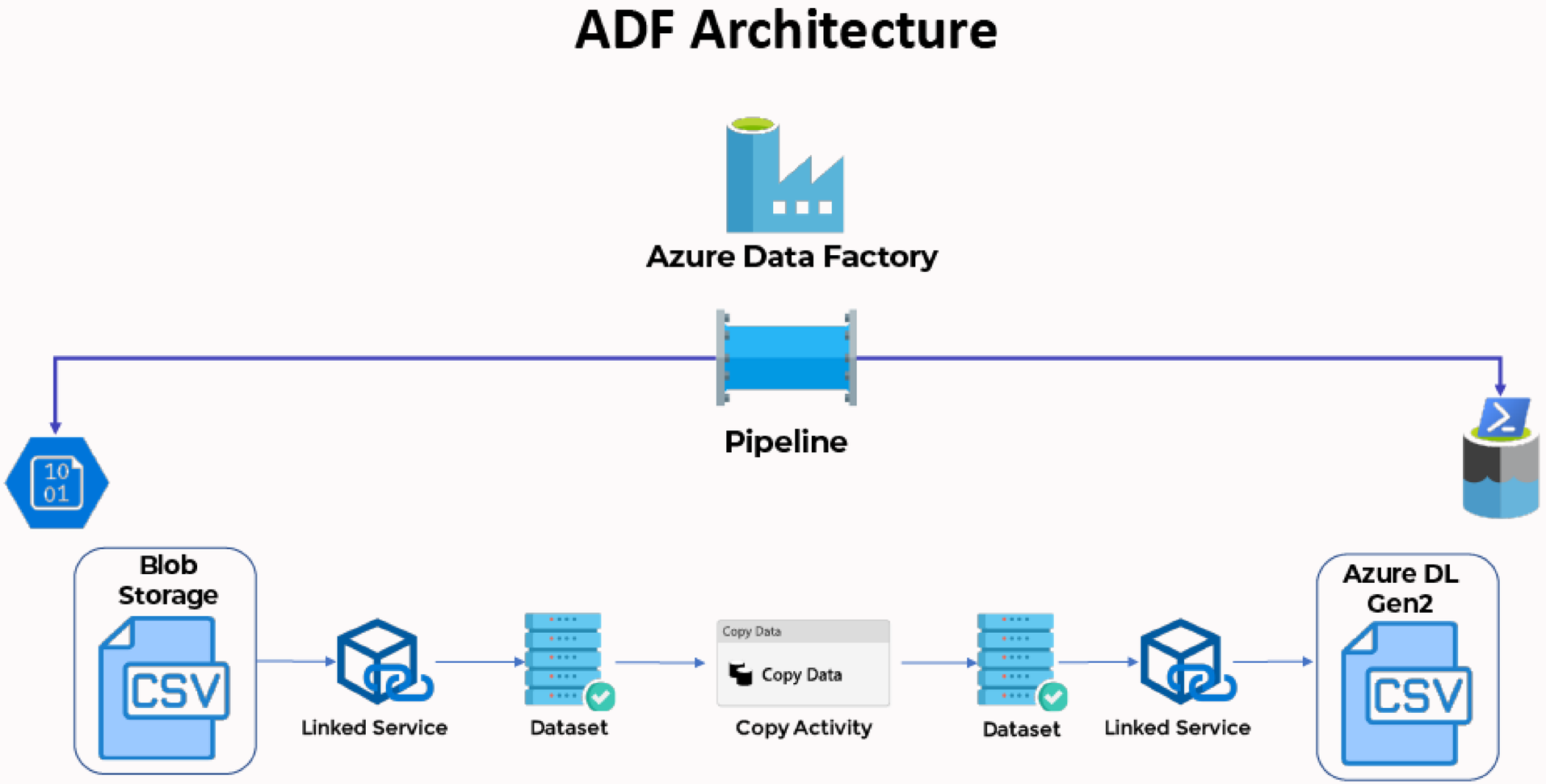
KEY VAULT


Secrets storing connection strings
Azure Key Vault
Get Connection Strings
Extract Using Managed Identity
Load using Managed Identity
Azure Data Lake
Azure Data Factory
Azure Synapse Analytics

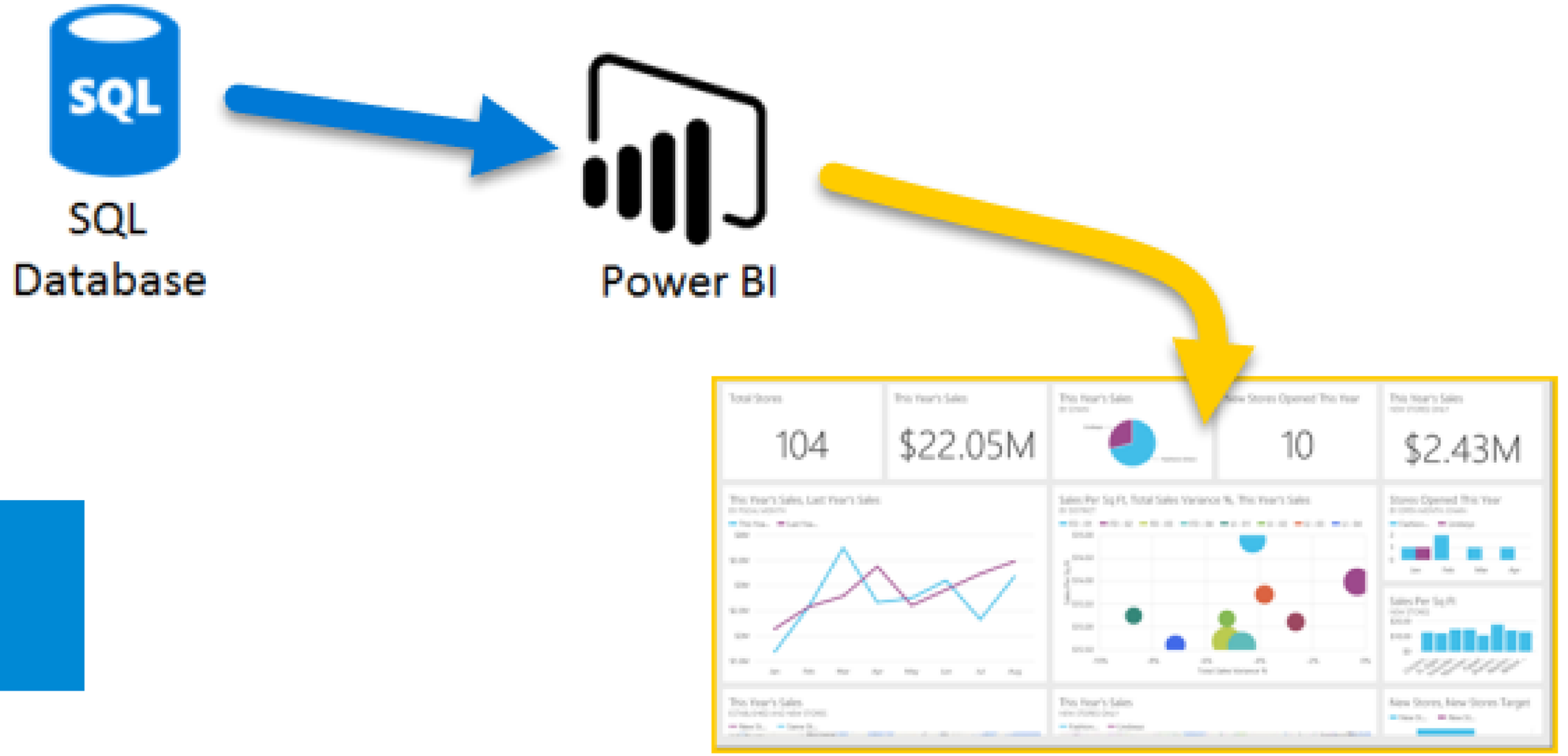
DATABRICKS - SQL DATABASE



DATA FACTORY



POWER BI



DATA

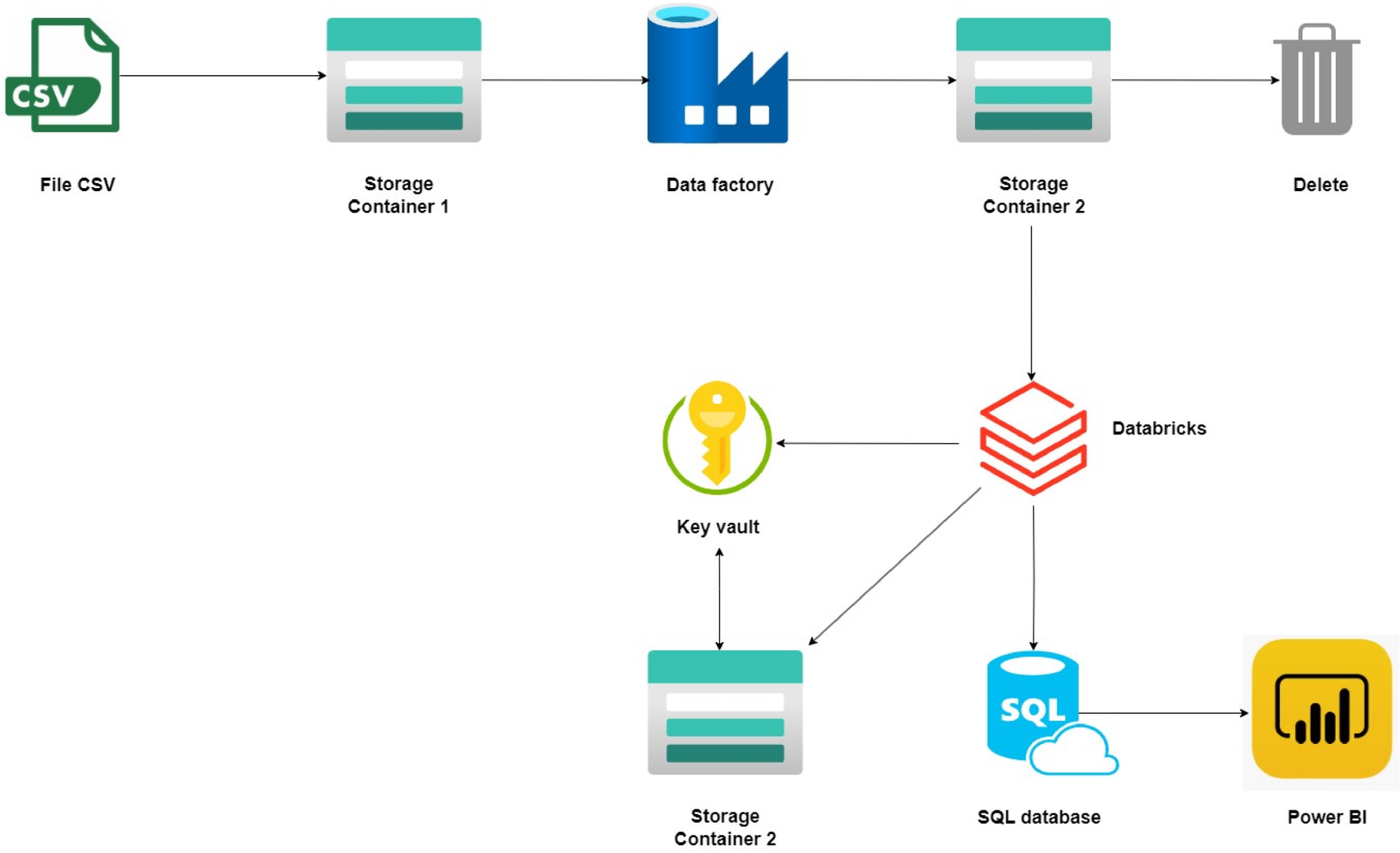
Tokyo 2020 Olympic Summer Games

Athletes

Coaches

Medals

Azure Pipelines



Read data by access key

Test try data Python

File Edit View Run Help Last edit was 23 minutes ago Provide feedback

Run all 22/8 Schedule Share

Cmd 3

Python

```
container ="container1"
storage_account_name = "huuphan"
spark.conf.set(
  "fs.azure.account.key."+storage_account_name+".blob.core.windows.net",
  dbutils.secrets.get("keyofme", "top"))
file_location = f"wasbs://{container}@{storage_account_name}.blob.core.windows.net/medals2.csv"
file_type = "csv"
df = spark.read.format(file_type).option("header", "true").load(file_location)
display(df)
```

(2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [_c0: string, medal_type: string ... 11 more fields]

Table +

	_c0	medal_type	medal_code	medal_date	athlete_short_name	athlete_name	athlete_sex
1	0	Gold Medal	1	2021-07-24 00:00:00.0	KIM JD	KIM Je Deok	null
2	1	Gold Medal	1	2021-07-24 00:00:00.0	AN S	AN San	X
3	2	Silver Medal	2	2021-07-24 00:00:00.0	SCHLOESSER G	SCHLOESSER Gabriela	X
4	3	Silver Medal	2	2021-07-24 00:00:00.0	WIJLER S	WIJLER Steve	null
5	4	Bronze Medal	3	2021-07-24 00:00:00.0	ALVAREZ L	ALVAREZ Luis	X
6	5	Bronze Medal	3	2021-07-24 00:00:00.0	VALENCIA A	VALENCIA Alejandra	null
7	6	Gold Medal	1	2021-07-24 00:00:00.0	CARAPAZ R	CARAPAZ Richard	M

Delete unnecessary columns

```
1  df1 = df.drop('_c0', 'medal_date')
```

▸ df1: pyspark.sql.dataframe.DataFrame = [medal_type: string, medal_code: string ... 9 more fields]

Command took 0.08 seconds -- by ph.thienhuu@gmail.com at 22:29:54 22/8/2023 on 22/8

handle null value and drop columns

```
from pyspark.sql.functions import *
from pyspark.sql.window import Window
# Xác định giá trị khác null trong cột theo từng nhóm
wd = Window.partitionBy('event')
df2 = df1.withColumn('NonNullValues', last('athlete_sex', ignorenulls=True).over(wd)) #lấy giá trị cuối cùng khi sắp xếp tăng dần bỏ qua các gtri null

# Thay các giá trị null bằng giá trị khác null đã tìm ra
df2 = df2.withColumn('athlete_sex', col('NonNullValues'))
df2 = df2.drop('NonNullValues')

# thay các giá trị null còn lại bằng ý nghĩa của dữ liệu
df3 = df2.withColumn('athlete_sex', when((col('athlete_sex').isNull()) & (col('event').like('%Women%')),'W').otherwise(col('athlete_sex')))
df4 = df3.withColumn('athlete_sex', when((col('athlete_sex').isNull()) & (col('event').like('%Men%')),'M').otherwise(col('athlete_sex')))

# từ ý dữ liệu của cột athlete_sex(thể hiện giới tính của nội dung thi đấu) chuyển qua event_sex
df4 = df4.withColumnRenamed('athlete_sex', 'event_sex')
medal_df = df4.withColumnRenamed('athlete_link', 'url')
medal_df = medal_df.drop('country_code','country','discipline_code','discipline')
display(medal_df)
```

read and handle null value in file Athletes

Read Athletes CSV and transform

```
#Đọc đỉa chỉ file athletes
file_location1 = f"wasbs://{container}@{storage_account_name}.blob.core.windows.net/athletes.csv"
file_type = "csv"
athlete_df = spark.read.format(file_type).option("header", "true").load(file_location1)
display(athlete_df)
```

```

from pyspark.sql.functions import *

# thay thế các giá trị null trong bảng
athlete_df_1 = athlete_df.withColumn('birth_place', when((col('birth_place').isNull()),(col('birth_country'))).otherwise(col('birth_place')))

athlete_df_1 = athlete_df_1.withColumn('birth_place', when((col('birth_place').isNull()),(col('country'))).otherwise(col('birth_place')))
athlete_df_1 = athlete_df_1.withColumn('birth_country', when((col('birth_country').isNull()),(col('country'))).otherwise(col('birth_country')))

athlete_df_1 = athlete_df_1.withColumn('residence_country', when((col('residence_country').isNull()),(col('residence_place'))).otherwise(col('residence_country')))
athlete_df_1 = athlete_df_1.withColumn('residence_country', when((col('residence_country').isNull()),(col('country'))).otherwise(col('residence_country')))
athlete_df_1 = athlete_df_1.withColumn('residence_place', when((col('residence_place').isNull()),(col('country'))).otherwise(col('residence_place')))

athlete_df_1 = athlete_df_1.withColumn('height_m/ft', when(col('height_m/ft').isNull(),'not collected yet').otherwise(col('height_m/ft')))
display(athlete_df_1)
```

create dim_country and Id for athlete table

```
# sau khi thực hiện làm sạch
# tạo country_df
country_df = athlete_df_1.select('country_code', 'country')
country_df = country_df.distinct()
display(country_df)
```

```
#kiểm tra trc khi join
if athlete_df_1.select('url').distinct().count() == athlete_df.select('url').count() :
    print('ko có dữ liệu trùng lặp')
else :
    print('Có dữ liệu trùng lặp')
```

▸ (5) Spark Jobs

```
ko có dữ liệu trùng lặp
```

```
# tạo id cbo bảng athlete
athlete_df_2 = athlete_df_1.withColumn('Athletes_Id', monotonically_increasing_id()+1)
athlete_df_2 = athlete_df_2.select('Athletes_Id', *athlete_df_2.columns[0:-1]).sort('Athletes_Id')
display(athlete_df_2)
```

Join Athlete and Medals2

```
#Nối bảng athlete và bảng medal để tạo nên bảng fact
fact_df = athlete_df_2.join(medal_df, on='url', how='left_outer')
display(fact_df)
```

```
#Thay đổi những giá trị null ở medal_type thành no medal
fact_df_1 = fact_df.withColumn('medal_type', when(col('medal_type').isNull(),'No medal').otherwise(col('medal_type')))
fact_df_2 = fact_df_1.select('Athletes_Id', 'medal_type','country_code', 'discipline_code', 'discipline', 'event',
'event_sex')
display(fact_df_2)

```

```
# tạo discipline_df
wdl = Window.partitionBy('discipline_code')
fact_df_2 = fact_df_1.withColumn('NonNullValues', last('discipline', ignorenulls=True).over(wdl))
#lấy giá trị cuối cùng khi sắp xếp tăng dần bỏ qua các gtri null
# Thay các giá trị null bằng giá trị khác null đã tìm ra
fact_df_2 = fact_df_2.withColumn('discipline', col('NonNullValues'))
fact_df_2 = fact_df_2.drop('NonNullValues')

discipline_df = fact_df_2.select('discipline_code', 'discipline')
discipline_df = discipline_df.distinct()
display(discipline_df)
```

```python
# Tạo bảng event
event_df = fact_df_2.select('discipline_code', 'event', 'event_sex')
event_df = event_df.distinct().dropna()
event_df = event_df.withColumn('Event_Id', monotonically_increasing_id()+1)
event_df = event_df.select('Event_Id', *event_df.columns[0:-1]).sort('Event_Id')

# vì fact không có id event nên thực hiện tạo 1 khóa phụ để join bảng fact lấy id event
event_df = event_df.withColumn('event_sub', concat(col('discipline_code'),col('event'),col('event_sex')))
event_df = event_df.withColumnRenamed('discipline_code', 'E_discipline_code')
display(event_df)
```

```python
# tạo sub_key
fact_df_3 = fact_df_2.withColumn('discipline', concat(col('country_code'),col('discipline')))
fact_df_3 = fact_df_3.withColumnRenamed('discipline', 'sub_key')
# tạo khóa phụ để join với event
fact_df_3 = fact_df_3.withColumn('event_sub', concat(col('discipline_code'),col('event'),col('event_sex')))
display(fact_df_3)
```

```python
# join vs bảng event
fact_table = fact_df_3.join(event_df, on='event_sub', how='left_outer')
fact_table = fact_table.select('Athletes_Id', 'medal_type', 'country_code', 'discipline_code', 'Event_Id')
display(fact_table)
```

```python
# xóa khóa phụ đã tạo đẻ đc lại bảng event ban đầu
event_df = event_df.drop('event_sub')
event_df = event_df.withColumnRenamed('E_discipline_code', 'discipline_code')
display(event_df)
```

```python
# import bảng coaches, làm sạch và tạo ID
file_location2 = f"wasbs://{container}@{storage_account_name}.blob.core.windows.net/coaches.csv"
file_type = "csv"
coaches_df = spark.read.format(file_type).option("header", "true").load(file_location2)
coaches_df_1 = coaches_df.withColumn('sub_key', concat (col('country_code'), col('discipline')))
display(coaches_df_1)
```

```python
#Tạo id
coaches_df_2 = coaches_df_1.withColumn('Coaches_Id', monotonically_increasing_id()+1)
coaches_df_2 = coaches_df_2.select('Coaches_Id', *coaches_df_2.columns[0:-1]).sort('Coaches_Id')

display(coaches_df_2)
```

```python
# tạo bảng total huy chương theo từng Quốc gia
total = fact_table.filter(col('medal_type') != 'No medal')
total.groupBy('country_code').count().display()
```

Save dataframes to Azure Storage

```python
# Đường dẫn đến Azure Blob Storage
blob_storage_path = f"wasbs://container2@huuphan.blob.core.windows.net/transformed/event"
blob_storage_path1 = f"wasbs://container2@huuphan.blob.core.windows.net/transformed/country"
blob_storage_path2 = f"wasbs://container2@huuphan.blob.core.windows.net/transformed/discipline"
blob_storage_path3 = f"wasbs://container2@huuphan.blob.core.windows.net/transformed/coaches"
blob_storage_path4 = f"wasbs://container2@huuphan.blob.core.windows.net/transformed/athlete"
blob_storage_path5 = f"wasbs://container2@huuphan.blob.core.windows.net/transformed/fact_table"

# Lưu DataFrame vào Azure Blob Storage dưới dạng file CSV
event_df.coalesce(1).write.parquet(blob_storage_path,  mode="overwrite")
country_df.coalesce(1).write.parquet(blob_storage_path1,  mode="overwrite")
discipline_df.coalesce(1).write.parquet(blob_storage_path2, mode="overwrite")
coaches_df_2.coalesce(1).write.parquet(blob_storage_path3,  mode="overwrite")
athlete_df_2.coalesce(1).write.parquet(blob_storage_path4,  mode="overwrite")
fact_table.coalesce(1).write.parquet(blob_storage_path5,  mode="overwrite")
```

▸ (25) Spark Jobs

```
Command took 8.62 seconds -- by ph.thienhuu@gmail.com at 22:34:52 22/8/2023 on 22/8
```

Recording data to SQL database

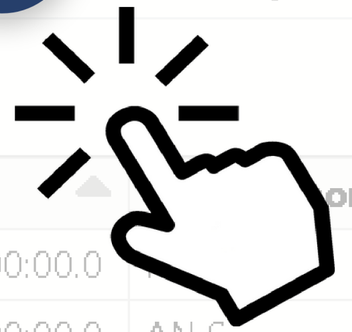
```python
server_name = 'huu-server'
database_name = 'olympic2020'
#Tạo secrets cho username và password sql
user_name = dbutils.secrets.get(scope ='keyofme', key ='usernamesql')
pass_word = dbutils.secrets.get(scope ='keyofme', key ='passwordsql')

list_table =  [
    (event_df, 'Dim_Events'),
    (country_df, 'Dim_Countries'),
    (discipline_df, 'Dim_Disciplines'),
    (athlete_df_2, 'Dim_Athletes'),
    (coaches_df_2, 'Dim_Coaches'),
    (fact_table, 'Fact_Table')
]
for i, j in list_table :
    i.write\
  .mode("overwrite")\
  .format("jdbc")\
  .option("url", f"jdbc:sqlserver://{server_name}.database.windows.net;databaseName={database_name};")\
  .option("dbtable", j)\
  .option("user", user_name)\
  .option("password", pass_word)\
  .option("driver", "com.microsoft.sqlserver.jdbc.SQLServerDriver")\
  .save()
```

```
container ="container1"
storage_account_name = "huuphan"
spark.conf.set(
  "fs.azure.account.key."+storage_account_name+".blob.core.windows.net",
  dbutils.secrets.get("keyofme", "top"))
file_location = "wasbs://{container}@{storage_account_name}.blob.core.windows.net/medals2.csv"
file_type = "csv"
df = spark.read.format(file_type).option("header", "true").load(file_location)
display(df)
```

Azure

DEMO

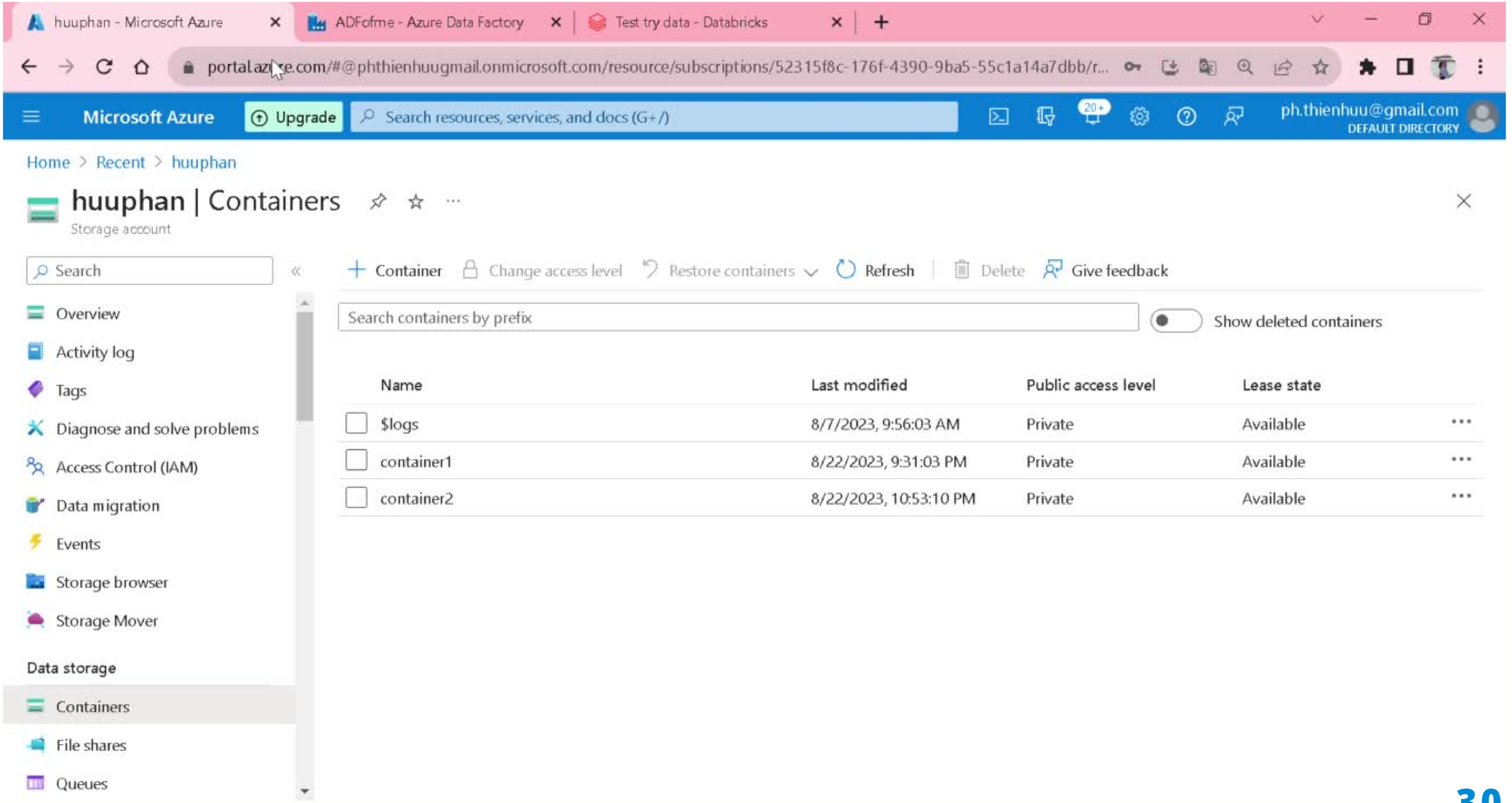


(2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [_c0: string, medal_type: string ... 4 more fields]

Table +

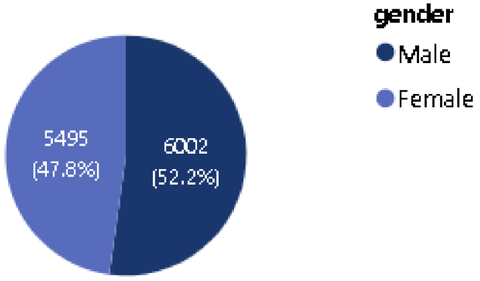
	_c0	medal_type	medal_code	medal_date	ort_name	athlete_name
1	0	Gold Medal	1	2021-07-24 00:00:00.0		KIM Je Deok
2	1	Gold Medal	1	2021-07-24 00:00:00.0	AN S	AN San
3	2	Silver Medal	2	2021-07-24 00:00:00.0	SCHLOESSER G	SCHLOESSER Gabriela
4	3	Silver Medal	2	2021-07-24 00:00:00.0	WIJLER S	WIJLER Steve
5	4	Bronze Medal	3	2021-07-24 00:00:00.0	ALVAREZ L	ALVAREZ Luis
6	5	Bronze Medal	3	2021-07-24 00:00:00.0	VALENCIA A	VALENCIA Alejandra
7	6	Gold Medal	1	2021-07-24 00:00:00.0	CARAPAZ R	CARAPAZ Richard


huuphan - Microsoft Azure
ADFofme - Azure Data Factory
Test try data - Databricks
portal.azure.com/#@phthienhuugmail.onmicrosoft.com/resource/subscriptions/52315f8c-176f-4390-9ba5-55c1a14a7dbb/r...
Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
ph.thienhuu@gmail.com
DEFAULT DIRECTORY
Home > Recent > huuphan
huuphan | Containers
Storage account
Search
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage browser
Storage Mover
Data storage
Containers
File shares
Queues
Container
Change access level
Restore containers
Refresh
Delete
Give feedback
Search containers by prefix
Show deleted containers
Name
Last modified
Public access level
Lease state
$logs
8/7/2023, 9:56:03 AM
Private
Available
container1
8/22/2023, 9:31:03 PM
Private
Available
container2
8/22/2023, 10:53:10 PM
Private
Available

VISUALIZATION

TOKYO 2020 OLYMPIC SUMMER GAMES

Distribution of medals by gender



Total Country

206

Count of country_code

Total Athletes

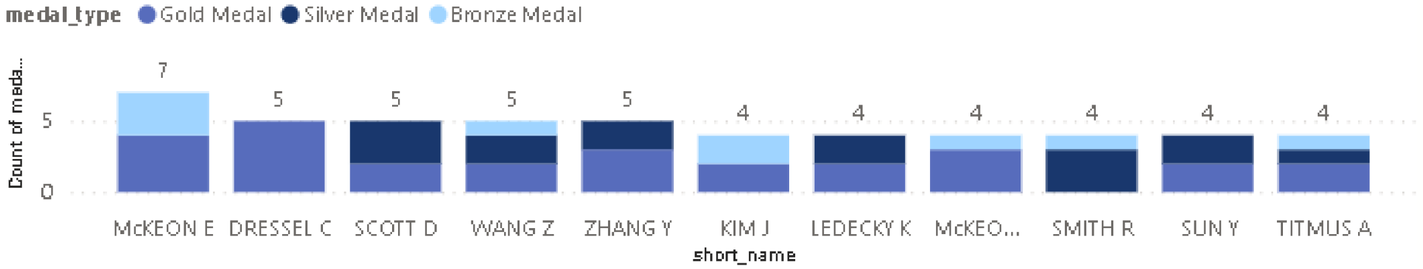
11.656K

Count of Athletes_Id

Medal Type: All

Country: All

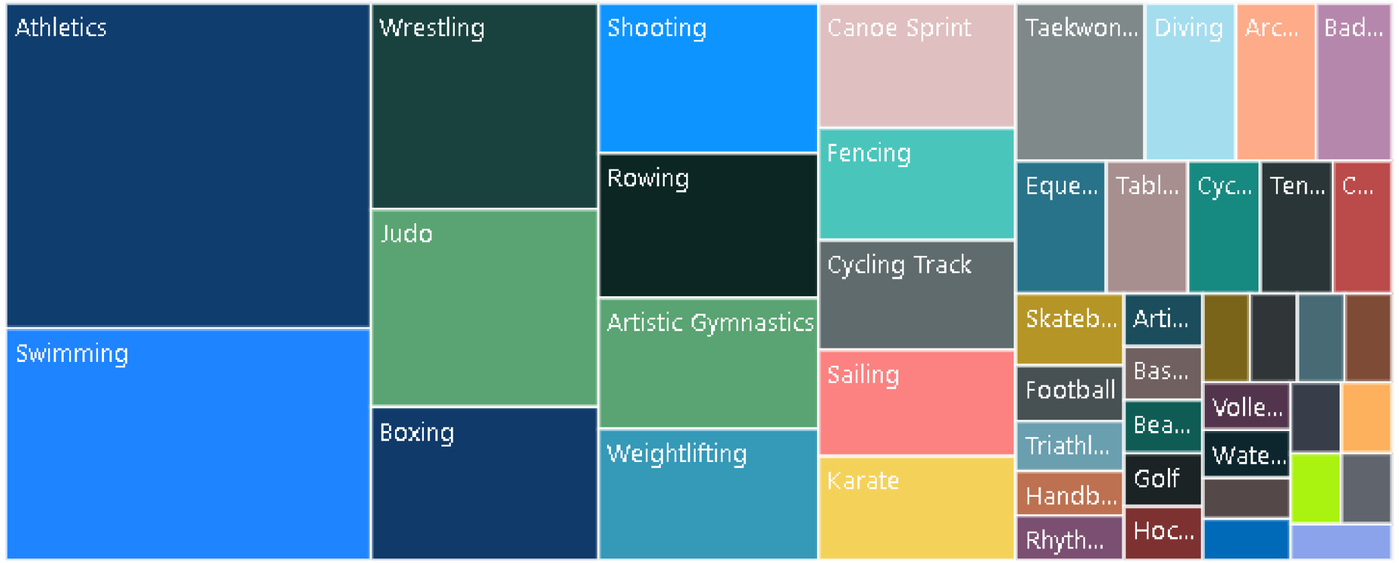
Distribution of medal by name



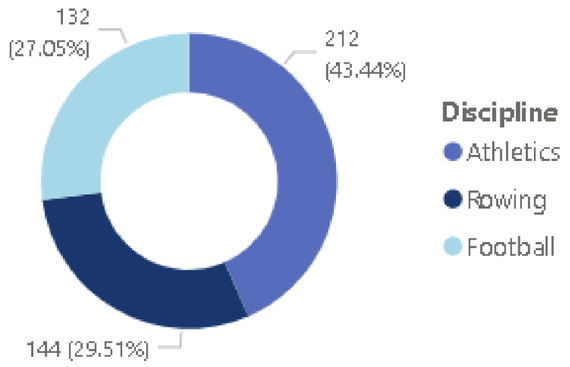
Top Country

Country	Gold Medal	Silver Medal	Bronze Medal	Total
United States of America	39	41	33	**113**
People's Republic of China	38	32	18	**88**
ROC	20	28	23	**71**
Great Britain	22	21	22	**65**
Japan	27	14	17	**58**
Australia	17	7	22	**46**
Italy	10	10	20	**40**
Germany	10	11	16	**37**
Netherlands	10	12	14	**36**
France	10	12	11	**33**
Canada	7	6	11	**24**
Brazil	7	6	8	**21**
Hungary	6	7	7	**20**
New Zealand	7	6	7	**20**
Republic of Korea	6	4	10	**20**
Ukraine	1	6	12	**19**
Total	**340**	**338**	**402**	**1080**

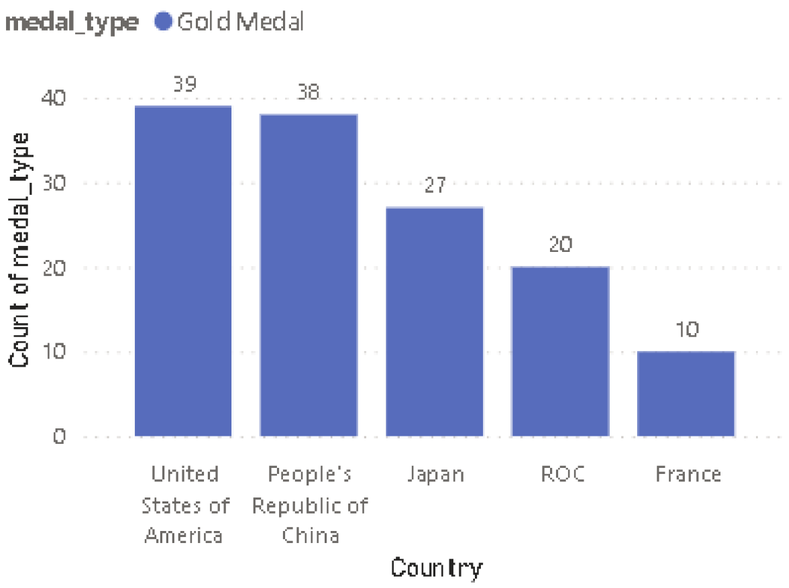
Distribution of medals by discipline



Top 3 Discipline with most Athletes



Top 5 Country with most Gold Medal

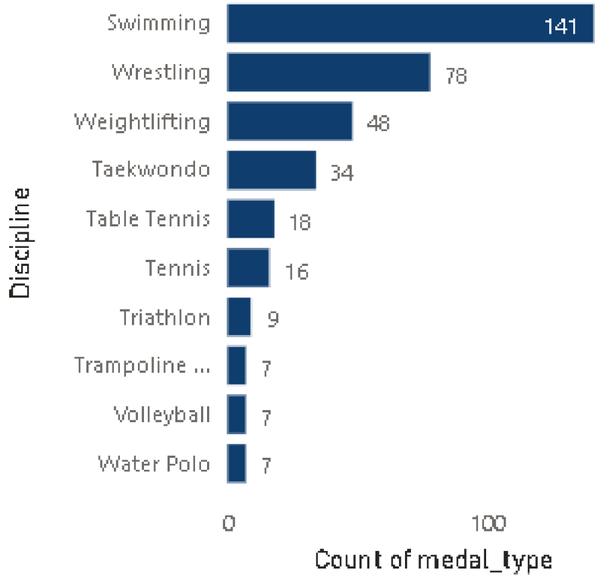


GOLD	SILVER	BRONZE
340	338	402

Medals: All

Country: All

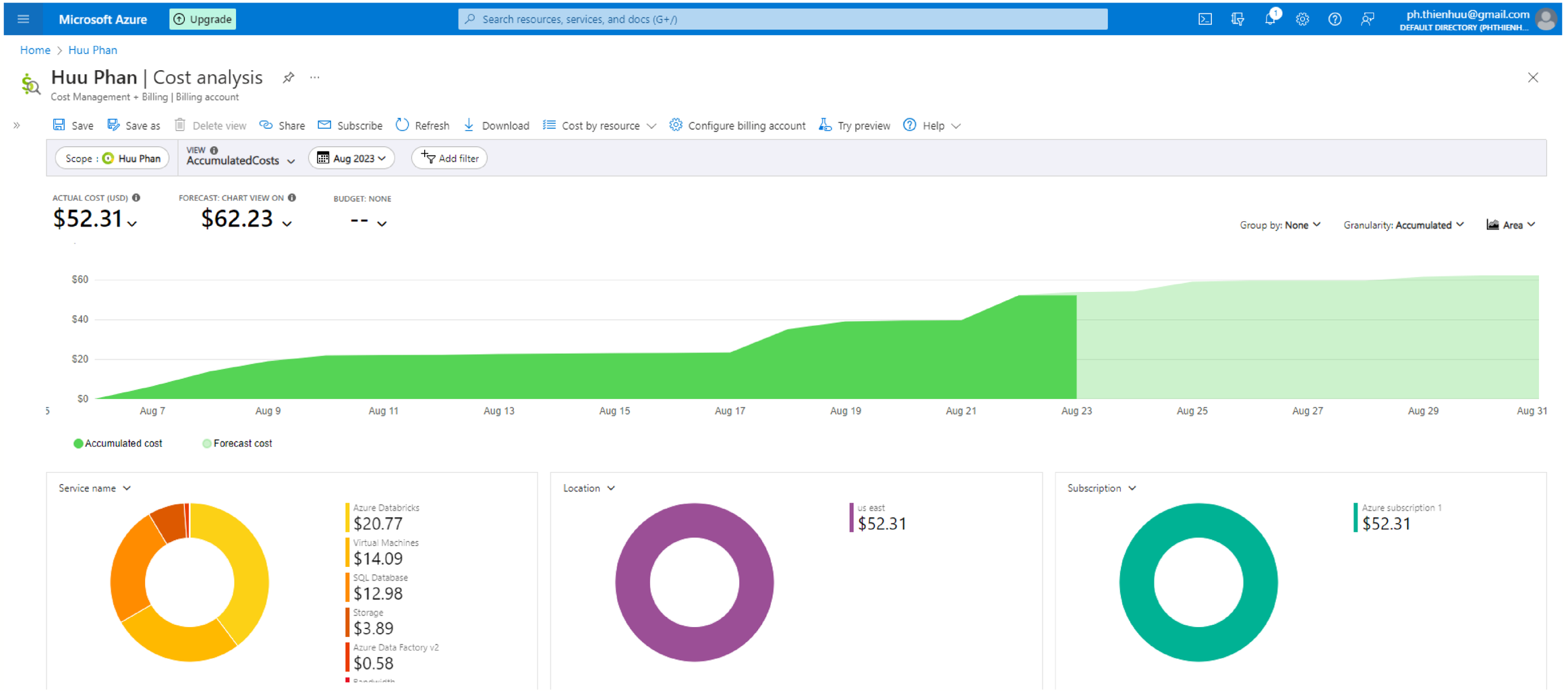
Top 10 Discipline with most Medals

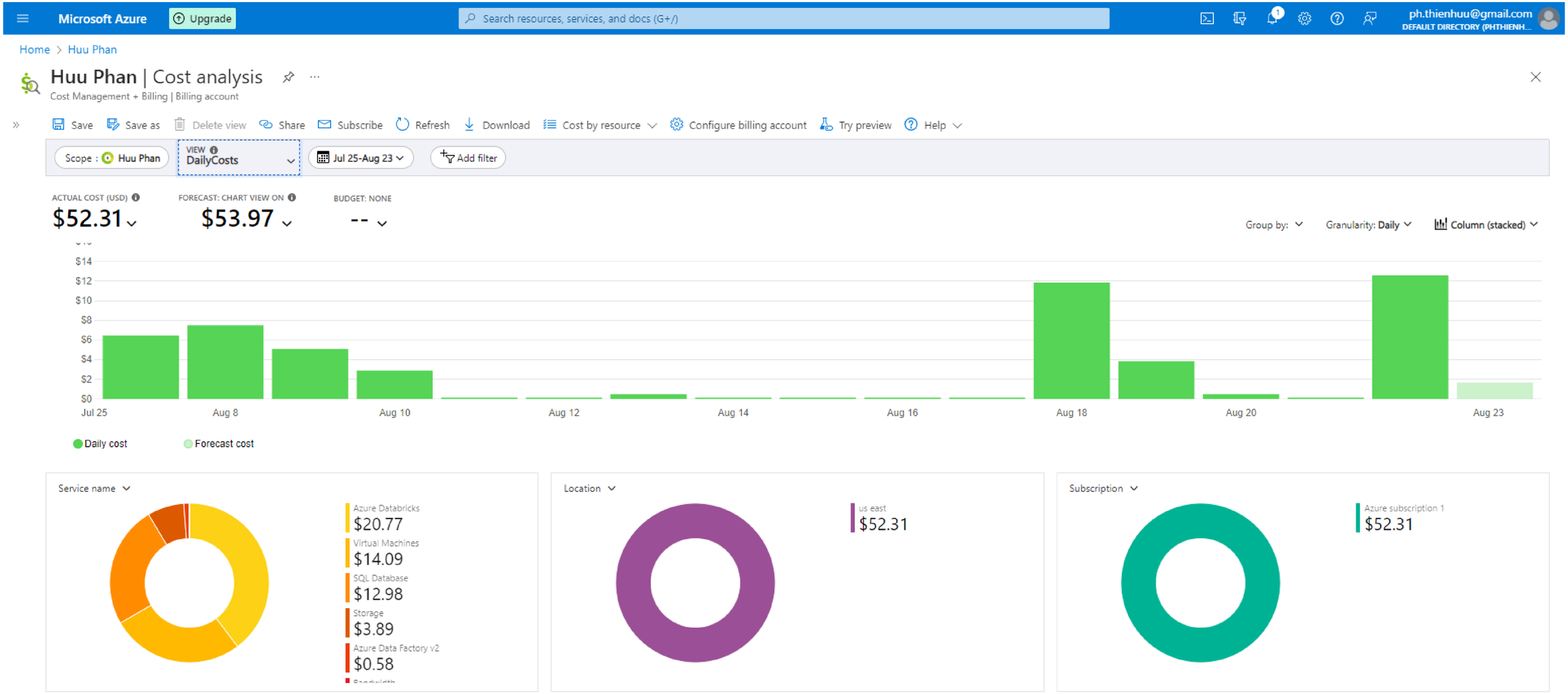


Top Athletes

Athletes Name	Bronze Medal	Gold Medal	Silver Medal	Total
McKEON Emma	3	4		7
DRESSEL Caeleb		5		5
LEDECKY Kathleen		2	2	4
McKEOWN Kaylee	1	3		4
SCOTT Duncan		1	3	4
TITMUS Ariarne	1	2	1	4
ZHANG Yufei		2	2	4
AN San		3		3
BATSARASHKINA Vitalina		2	1	3
CAMPBELL Cate	1	2		3
CARRINGTON Lisa		3		3
Total	**25**	**45**	**22**	**92**

COST MANAGEMENT


Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
ph.thienhuu@gmail.com
DEFAULT DIRECTORY (PHTHIENH...
Home > Huu Phan
Huu Phan | Cost analysis
Cost Management + Billing | Billing account
Save
Save as
Delete view
Share
Subscribe
Refresh
Download
Cost by resource
Configure billing account
Try preview
Help
Scope : Huu Phan
VIEW
AccumulatedCosts
Aug 2023
Add filter
ACTUAL COST (USD)
$52.31
FORECAST: CHART VIEW ON
$62.23
BUDGET: NONE
--
Group by: None
Granularity: Accumulated
Area
$60
$40
$20
$0
Aug 7
Aug 9
Aug 11
Aug 13
Aug 15
Aug 17
Aug 19
Aug 21
Aug 23
Aug 25
Aug 27
Aug 29
Aug 31
Accumulated cost
Forecast cost
Service name
Azure Databricks
$20.77
Virtual Machines
$14.09
SQL Database
$12.98
Storage
$3.89
Azure Data Factory v2
$0.58
Location
us east
$52.31
Subscription
Azure subscription 1
$52.31


Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
ph.thienhuu@gmail.com
DEFAULT DIRECTORY (PHTHIENH...
Home > Huu Phan
Huu Phan | Cost analysis
Cost Management + Billing | Billing account
Save
Save as
Delete view
Share
Subscribe
Refresh
Download
Cost by resource
Configure billing account
Try preview
Help
Scope : Huu Phan
VIEW DailyCosts
Jul 25-Aug 23
Add filter
ACTUAL COST (USD)
$52.31
FORECAST: CHART VIEW ON
$53.97
BUDGET: NONE
--
Group by:
Granularity: Daily
Column (stacked)
$14
$12
$10
$8
$6
$4
$2
$0
Jul 25
Aug 8
Aug 10
Aug 12
Aug 14
Aug 16
Aug 18
Aug 20
Aug 23
Daily cost
Forecast cost
Service name
Azure Databricks
$20.77
Virtual Machines
$14.09
SQL Database
$12.98
Storage
$3.89
Azure Data Factory v2
$0.58
Location
us east
$52.31
Subscription
Azure subscription 1
$52.31

THANK YOU