The Godfather of Talent | Futurense

# OLYMPIC HISTORY DE SOLUTION

**TEAM DATA LAKE**

1. ANIRUDH KURUVA (FT738)
2. KOUSTAV SARKAR (FT746)
3. MOKSHA H S (FT750)
4. SAI VENKAT SEELAM (FT756)
5. SAKSHI SHINDE (FT757)

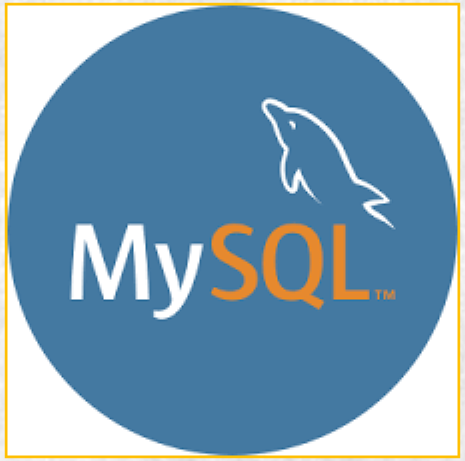
# Overview:

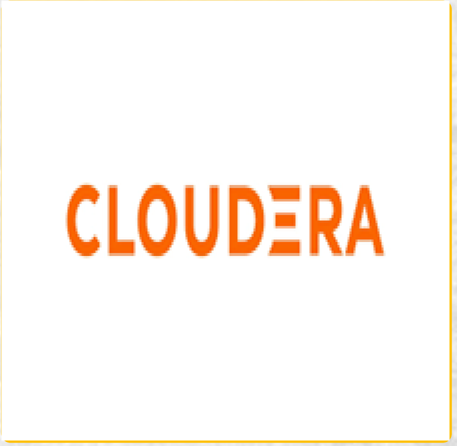
**Brief Description:** This project utilizes Olympic data to analyze historical performance, participation, and medal distribution across various countries and sports. It employs HBase with a Sqoop pipeline for data import, AWS S3 Snowpipe for external staging, Azure Copy for additional external staging, and Power BI for data visualization.

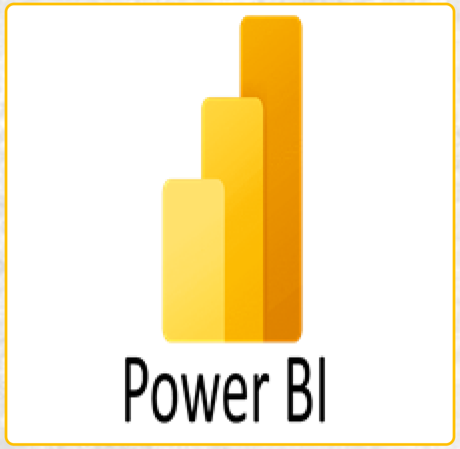
**Business Problem:**The business seeks insights into Olympic performance trends, country-wise medal distribution, and sport-specific achievements. The goal is to provide actionable insights to stakeholders, sponsors, and athletes to inform strategic decisions and investments.

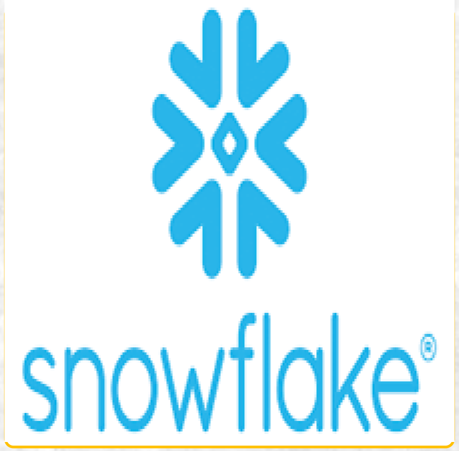
**Solution Approach:** 1. Data Collection 2. Data Storage 3.Data Staging 4. Data Integration 5. Data Analysis 6. Insights Delivery

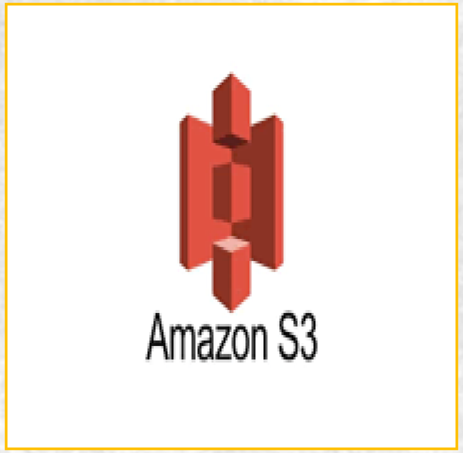
# Tools and Technologies Used:

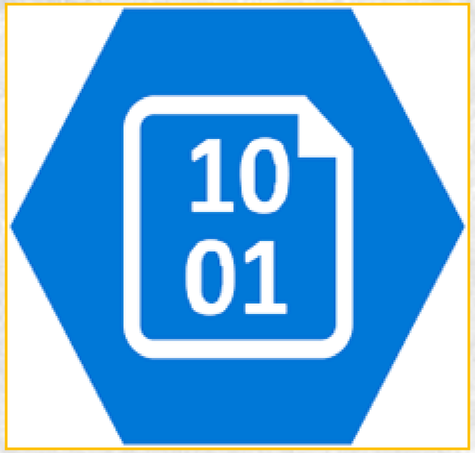




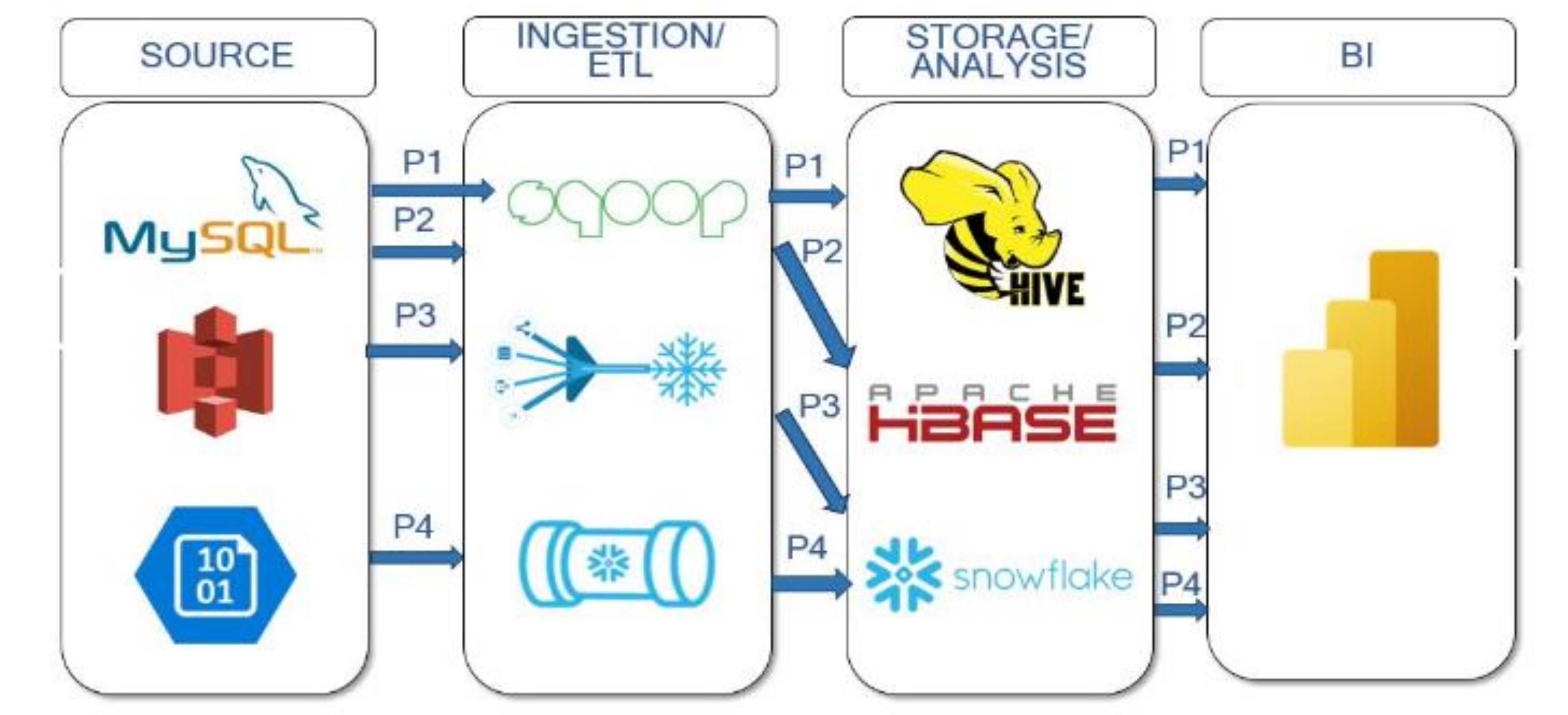




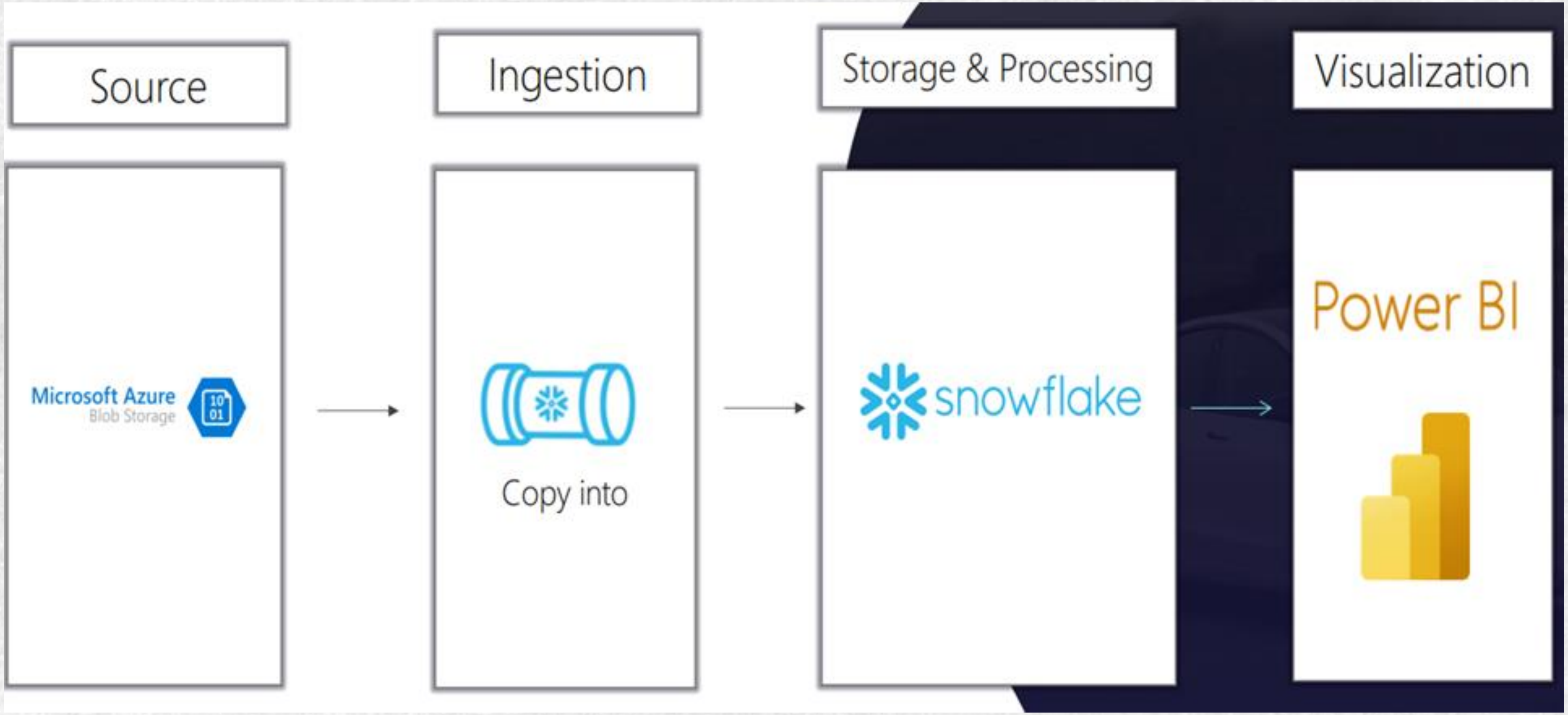




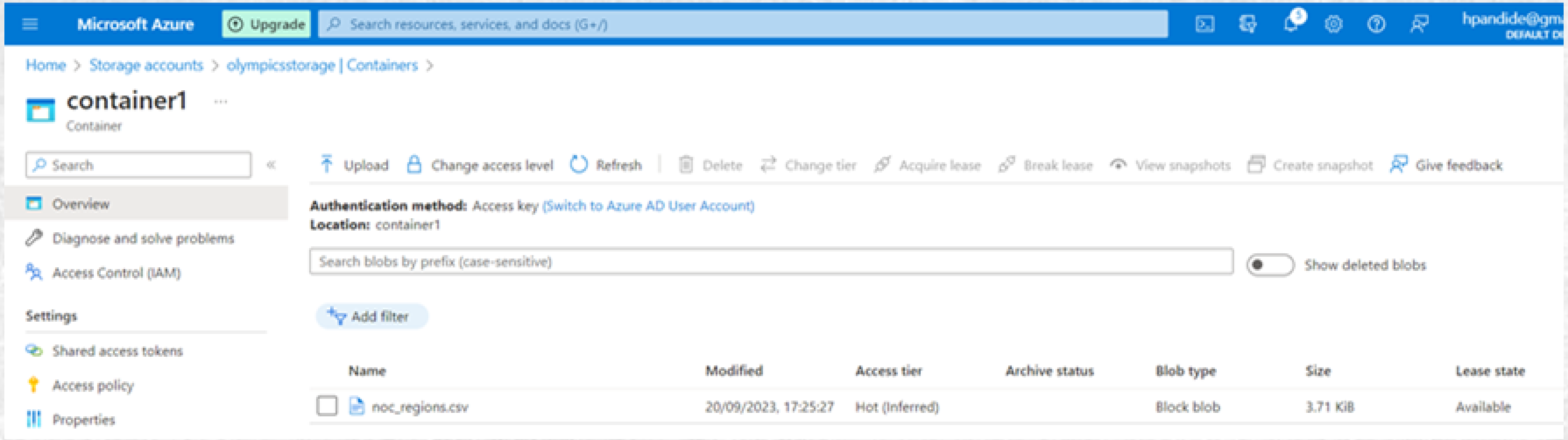
# DE SOLUTION ARCHITECTURE



# Azure BLOB to Snowflake Pipeline



## Azure Data input–



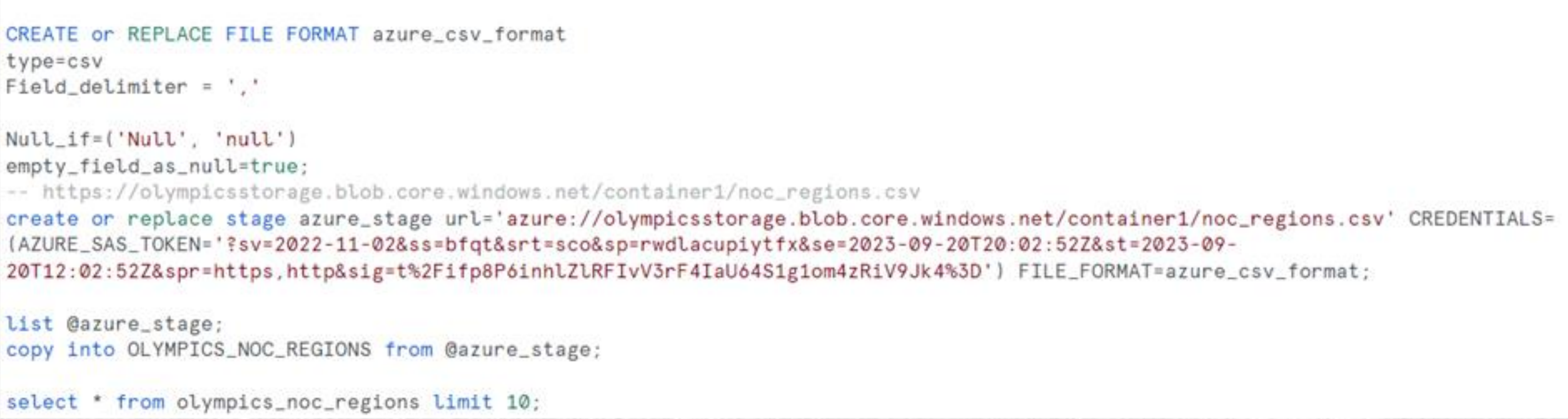
Home > Storage accounts > olympicsstorage | Containers >

container1 ...
Container

**Authentication method:** Access key (Switch to Azure AD User Account)
**Location:** container1

| | Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|---|---|---|---|---|---|---|---|
| □ | noc_regions.csv | 20/09/2023, 17:25:27 | Hot (Inferred) | | Block blob | 3.71 KiB | Available |

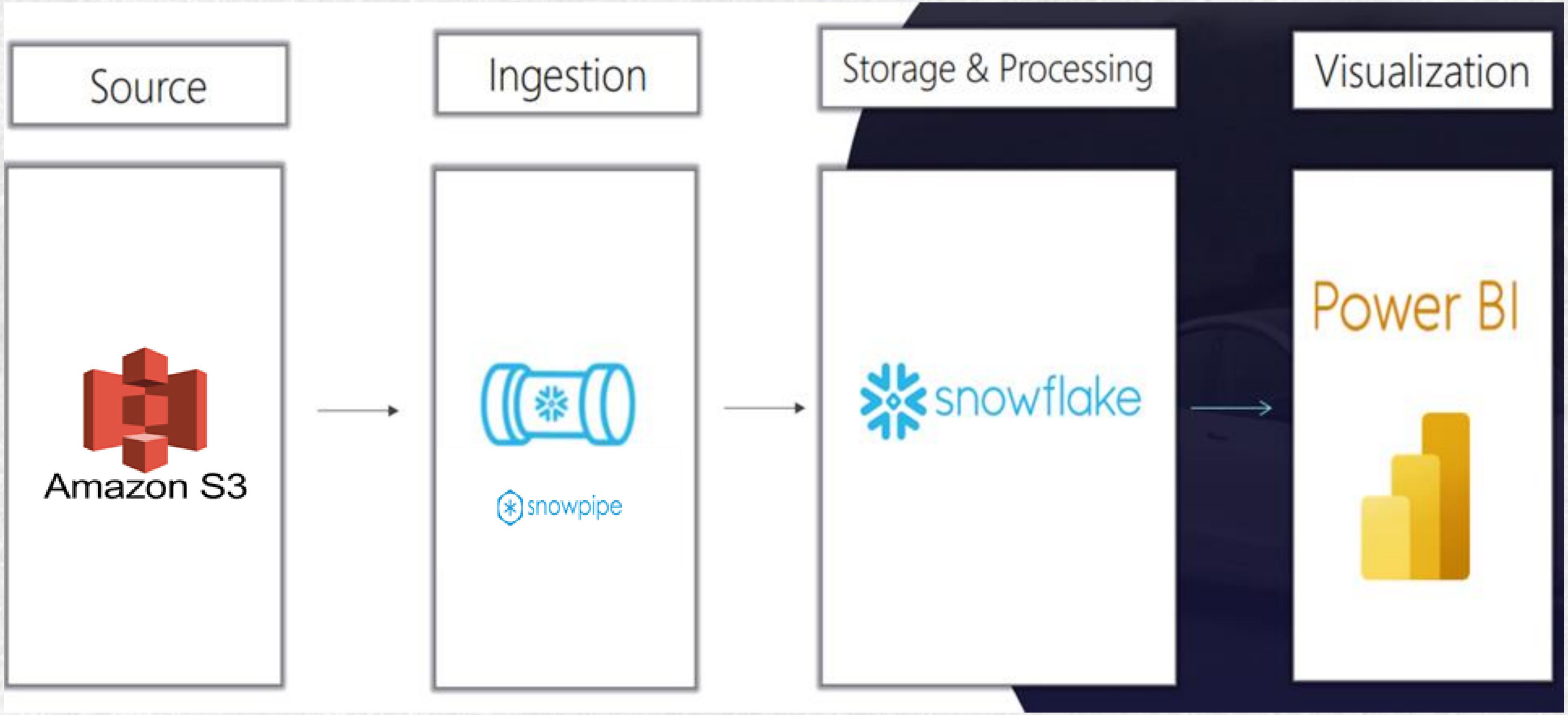
## External Stage & Copy Into Pipeline –



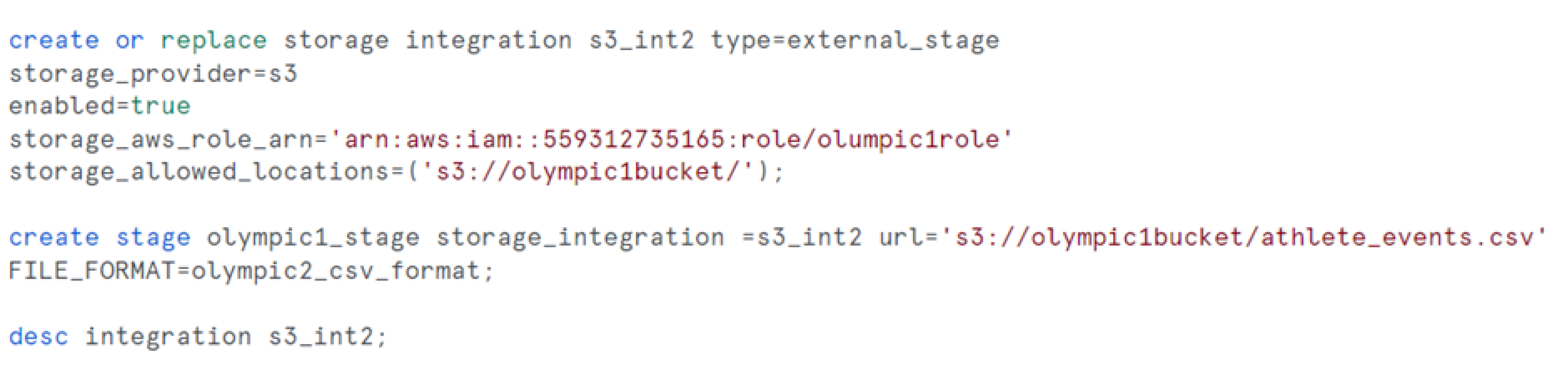
```
CREATE or REPLACE FILE FORMAT azure_csv_format
type=csv
Field_delimiter = ','

Null_if=('Null', 'null')
empty_field_as_null=true;
-- https://olympicsstorage.blob.core.windows.net/container1/noc_regions.csv
create or replace stage azure_stage url='azure://olympicsstorage.blob.core.windows.net/container1/noc_regions.csv' CREDENTIALS=
(AZURE_SAS_TOKEN='?sv=2022-11-02&ss=bfqt&srt=sco&sp=rwdlacupiytfx&se=2023-09-20T20:02:52Z&st=2023-09-
20T12:02:52Z&spr=https,http&sig=t%2Fifp8P6inhlZLRFIvV3rF4IaU64S1g1om4zRiV9Jk4%3D') FILE_FORMAT=azure_csv_format;

list @azure_stage;
copy into OLYMPICS_NOC_REGIONS from @azure_stage;

select * from olympics_noc_regions limit 10;
```

The Godfather of Talent | Futurense

# AWS S3 to Snowflake Pipeline

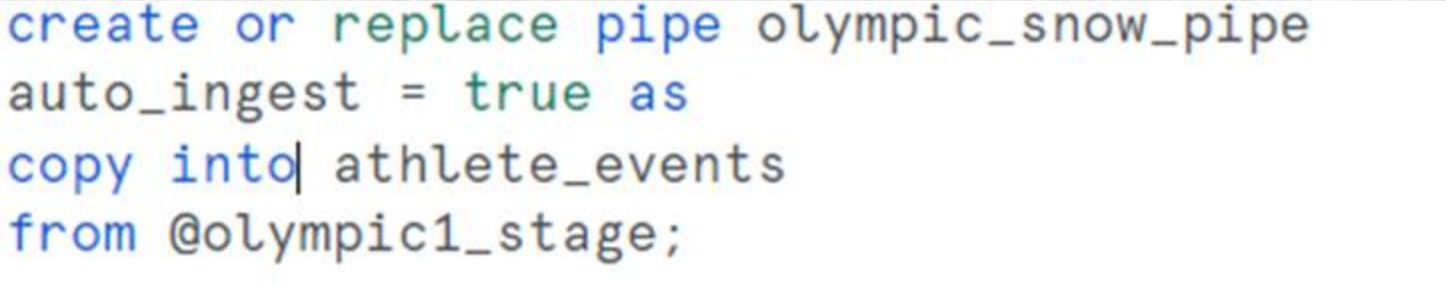


## Creating Storage Integration & External Stage for Aws Bucket



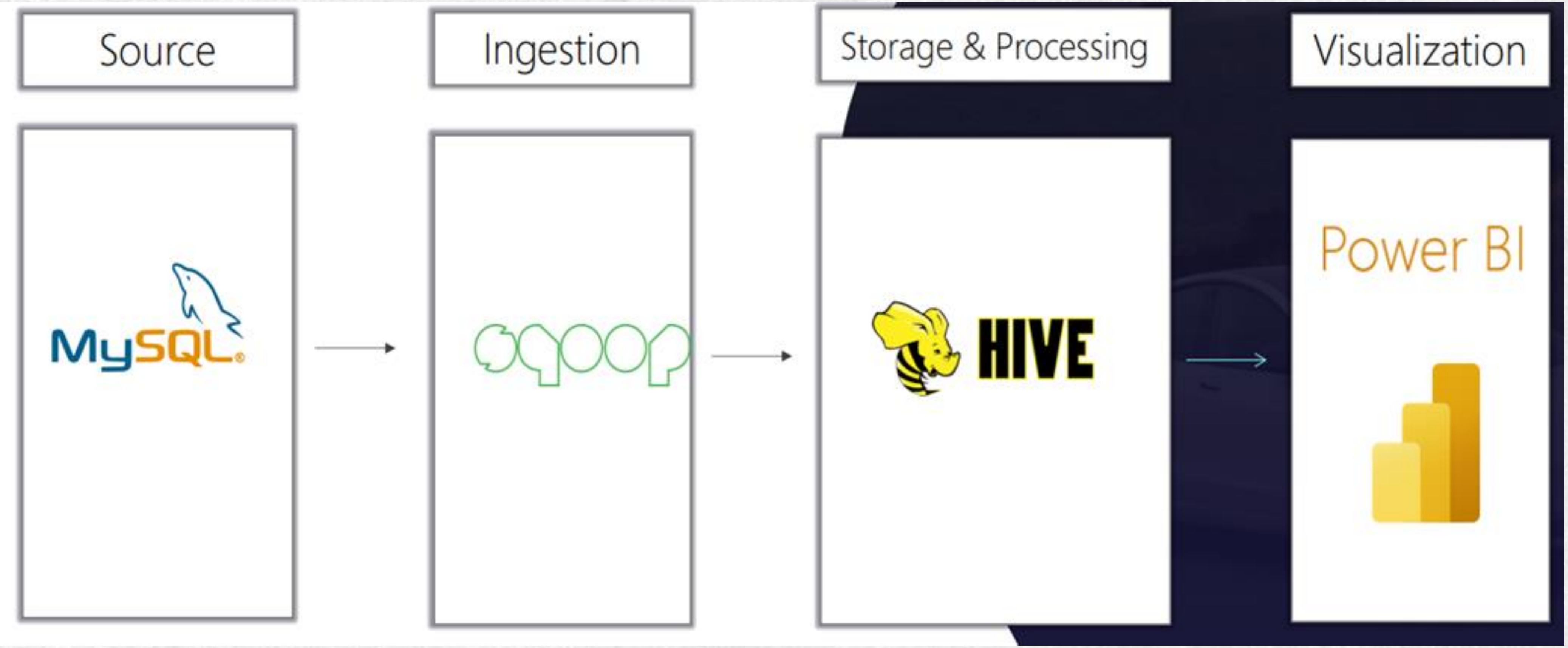
```
create or replace storage integration s3_int2 type=external_stage
storage_provider=s3
enabled=true
storage_aws_role_arn='arn:aws:iam::559312735165:role/olumpic1role'
storage_allowed_locations=('s3://olympic1bucket/');

create stage olympic1_stage storage_integration =s3_int2 url='s3://olympic1bucket/athlete_events.csv'
FILE_FORMAT=olympic2_csv_format;

desc integration s3_int2;
```

## Creating Snowpipe for importing from stage

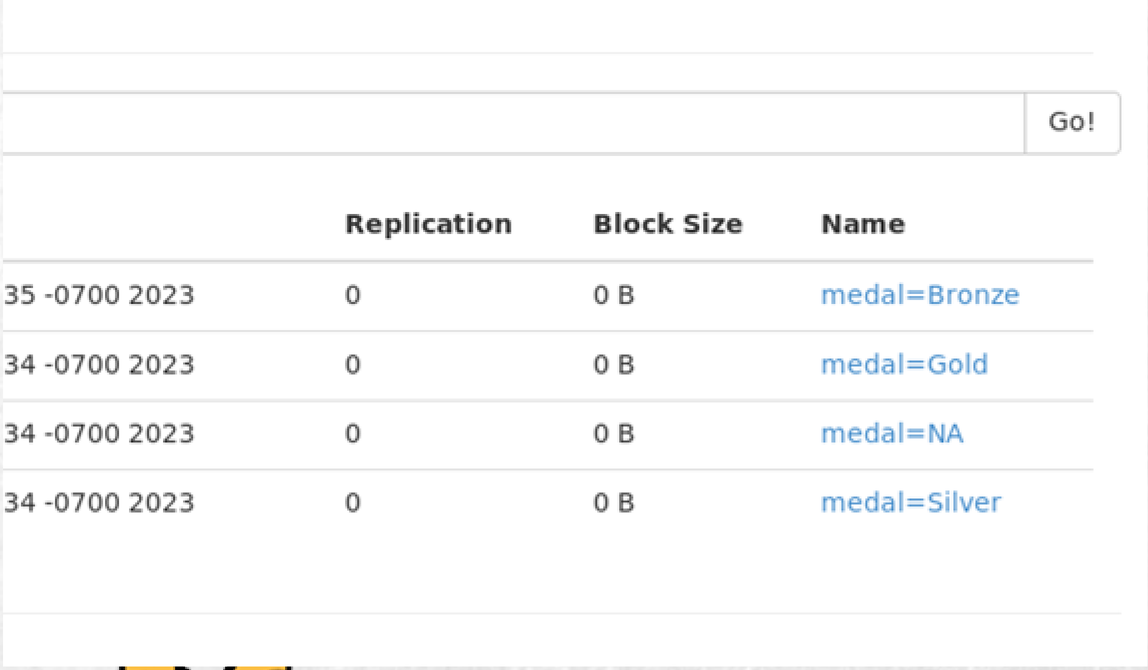


```
create or replace pipe olympic_snow_pipe
auto_ingest = true as
copy into athlete_events
from @olympic1_stage;
```

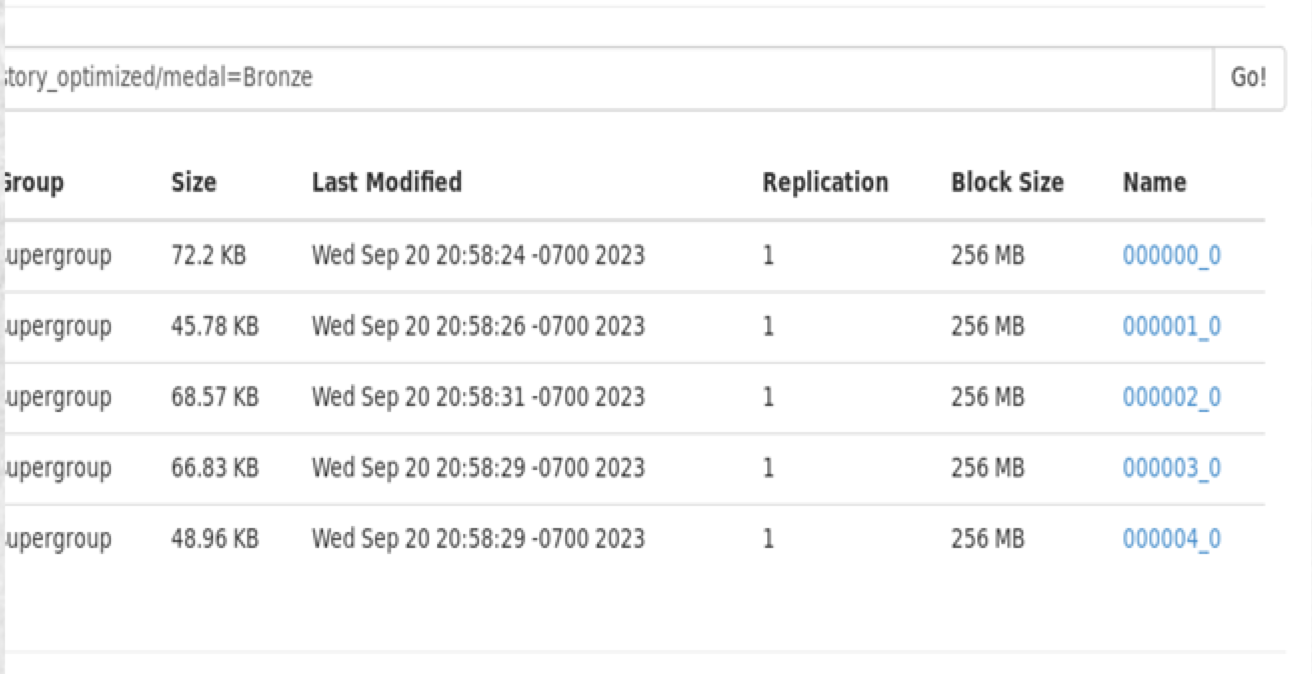
# MySQL To HIVE Pipeline



# Partitioned Data



| | Replication | Block Size | Name |
|---|---|---|---|
| 35 -0700 2023 | 0 | 0 B | medal=Bronze |
| 34 -0700 2023 | 0 | 0 B | medal=Gold |
| 34 -0700 2023 | 0 | 0 B | medal=NA |
| 34 -0700 2023 | 0 | 0 B | medal=Silver |

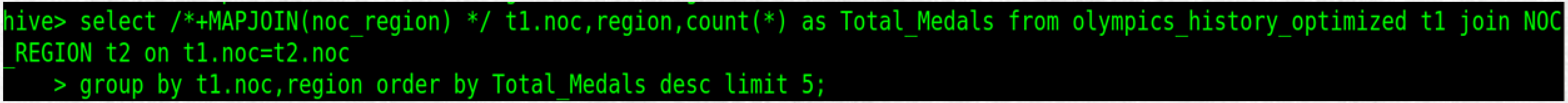


tory_optimized/medal=Bronze

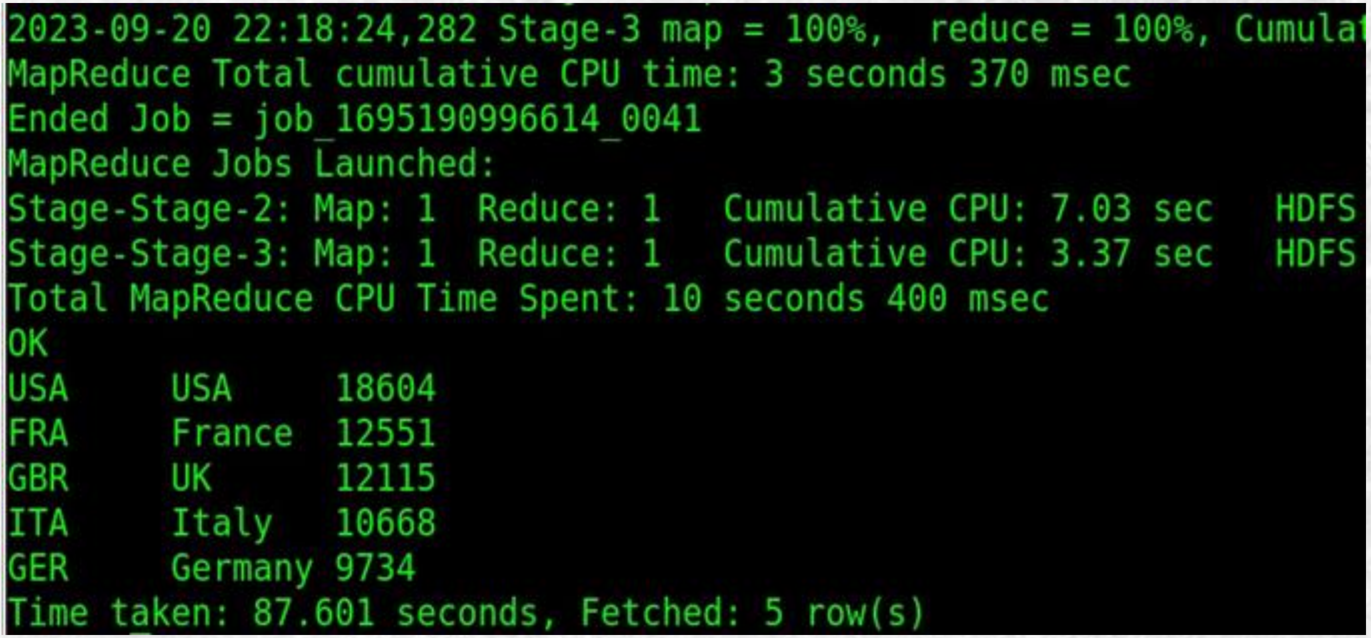
| iroup | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|
| upergroup | 72.2 KB | Wed Sep 20 20:58:24 -0700 2023 | 1 | 256 MB | 000000_0 |
| upergroup | 45.78 KB | Wed Sep 20 20:58:26 -0700 2023 | 1 | 256 MB | 000001_0 |
| upergroup | 68.57 KB | Wed Sep 20 20:58:31 -0700 2023 | 1 | 256 MB | 000002_0 |
| upergroup | 66.83 KB | Wed Sep 20 20:58:29 -0700 2023 | 1 | 256 MB | 000003_0 |
| upergroup | 48.96 KB | Wed Sep 20 20:58:29 -0700 2023 | 1 | 256 MB | 000004_0 |

# Hive User Story

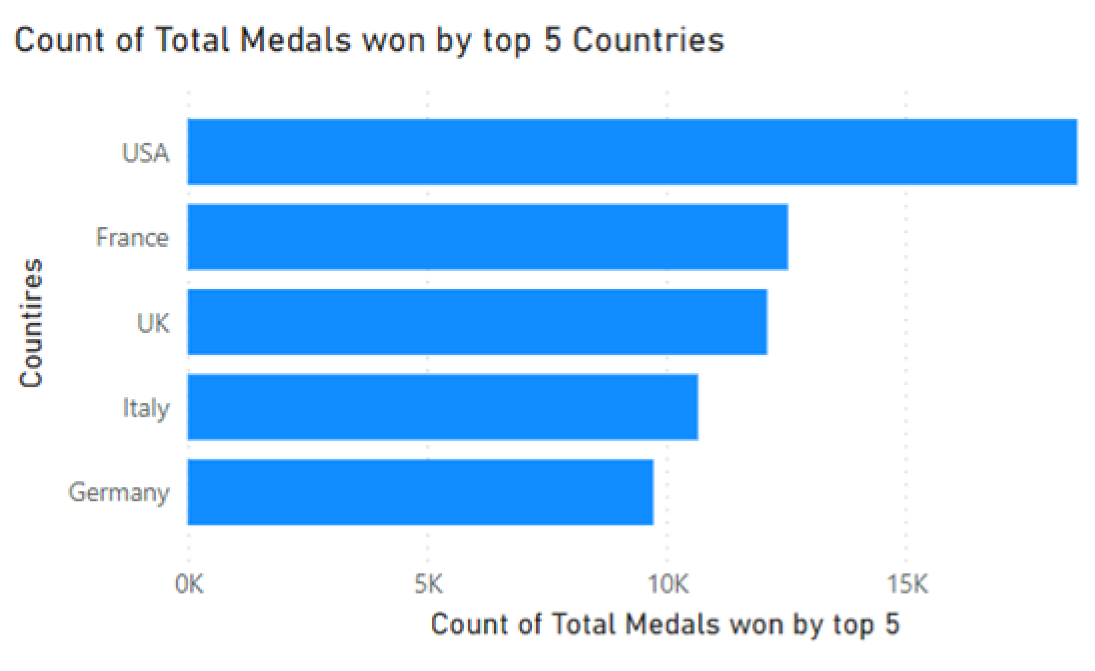
**Fetch the top 5 most successful countries in olympics. Success is defined by no of medals won.**



```
hive> select /*+MAPJOIN(noc_region) */ t1.noc,region,count(*) as Total_Medals from olympics_history_optimized t1 join NOC
_REGION t2 on t1.noc=t2.noc
    > group by t1.noc,region order by Total_Medals desc limit 5;
```

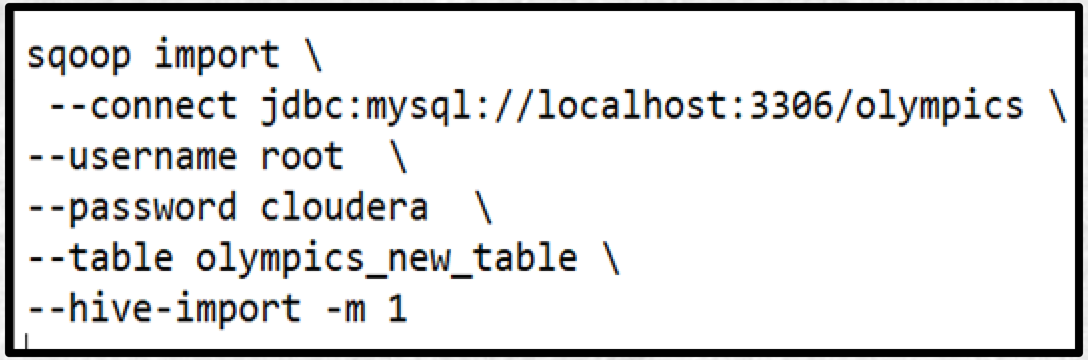


```
2023-09-20 22:18:24,282 Stage-3 map = 100%,  reduce = 100%, Cumulat
MapReduce Total cumulative CPU time: 3 seconds 370 msec
Ended Job = job_1695190996614_0041
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 7.03 sec   HDFS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 3.37 sec   HDFS
Total MapReduce CPU Time Spent: 10 seconds 400 msec
OK
USA     USA     18604
FRA     France  12551
GBR     UK      12115
ITA     Italy   10668
GER     Germany 9734
Time taken: 87.601 seconds, Fetched: 5 row(s)
```

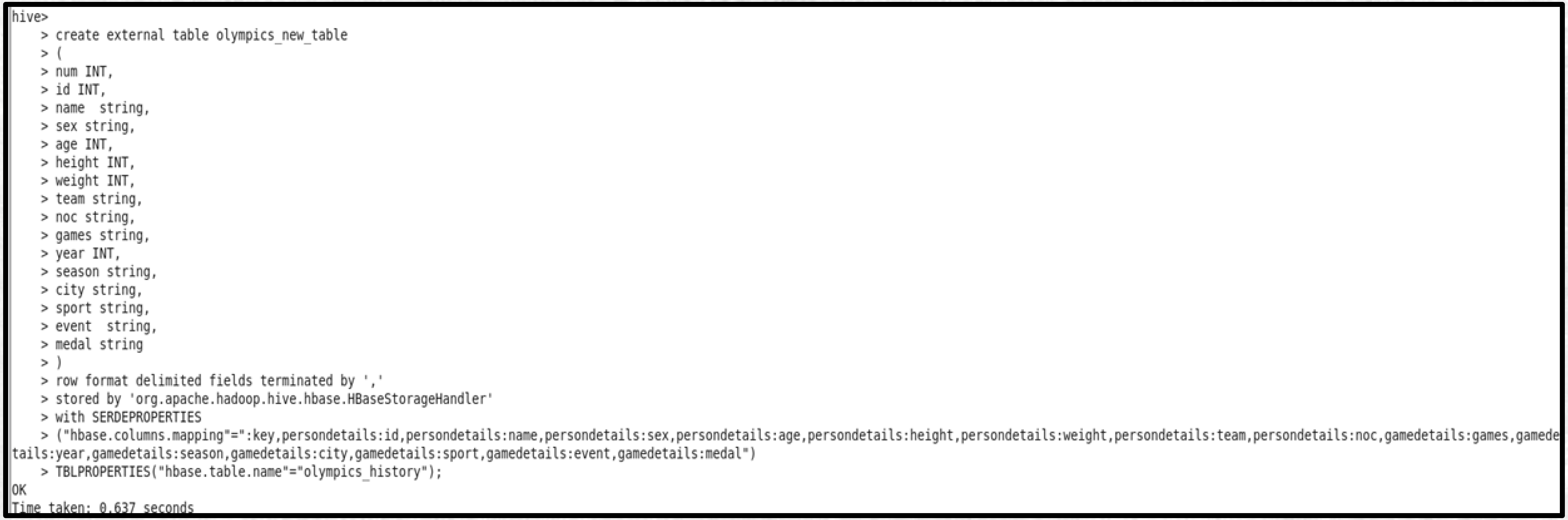


# HBase User Story

## Importing table from MySQL to Hive & Integrating Hive with HBase

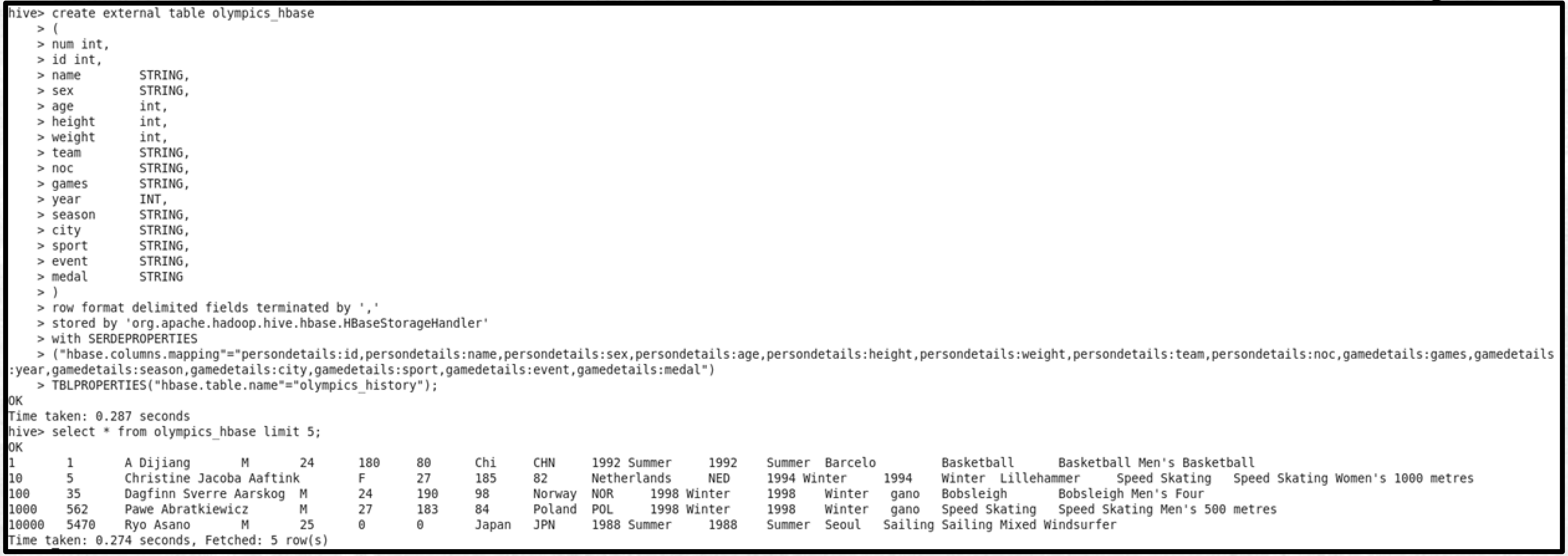


```
sqoop import \
 --connect jdbc:mysql://localhost:3306/olympics \
--username root  \
--password cloudera  \
--table olympics_new_table \
--hive-import -m 1
```



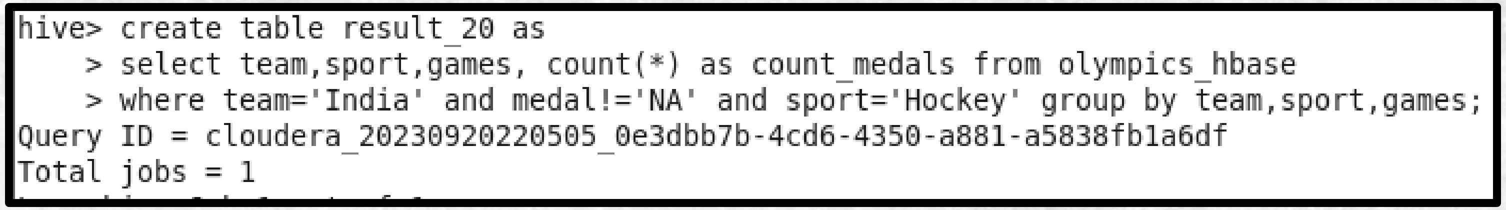
```
hive>
    > create external table olympics_new_table
    > (
    > num INT,
    > id INT,
    > name  string,
    > sex string,
    > age INT,
    > height INT,
    > weight INT,
    > team string,
    > noc string,
    > games string,
    > year INT,
    > season string,
    > city string,
    > sport string,
    > event  string,
    > medal string
    > )
    > row format delimited fields terminated by ','
    > stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > with SERDEPROPERTIES
    > ("hbase.columns.mapping"=":key,persondetails:id,persondetails:name,persondetails:sex,persondetails:age,persondetails:height,persondetails:weight,persondetails:team,persondetails:noc,gamedetails:games,gamede
tails:year,gamedetails:season,gamedetails:city,gamedetails:sport,gamedetails:event,gamedetails:medal")
    > TBLPROPERTIES("hbase.table.name"="olympics_history");
OK
Time taken: 0.637 seconds
```

## Integrating  HBase to Hive



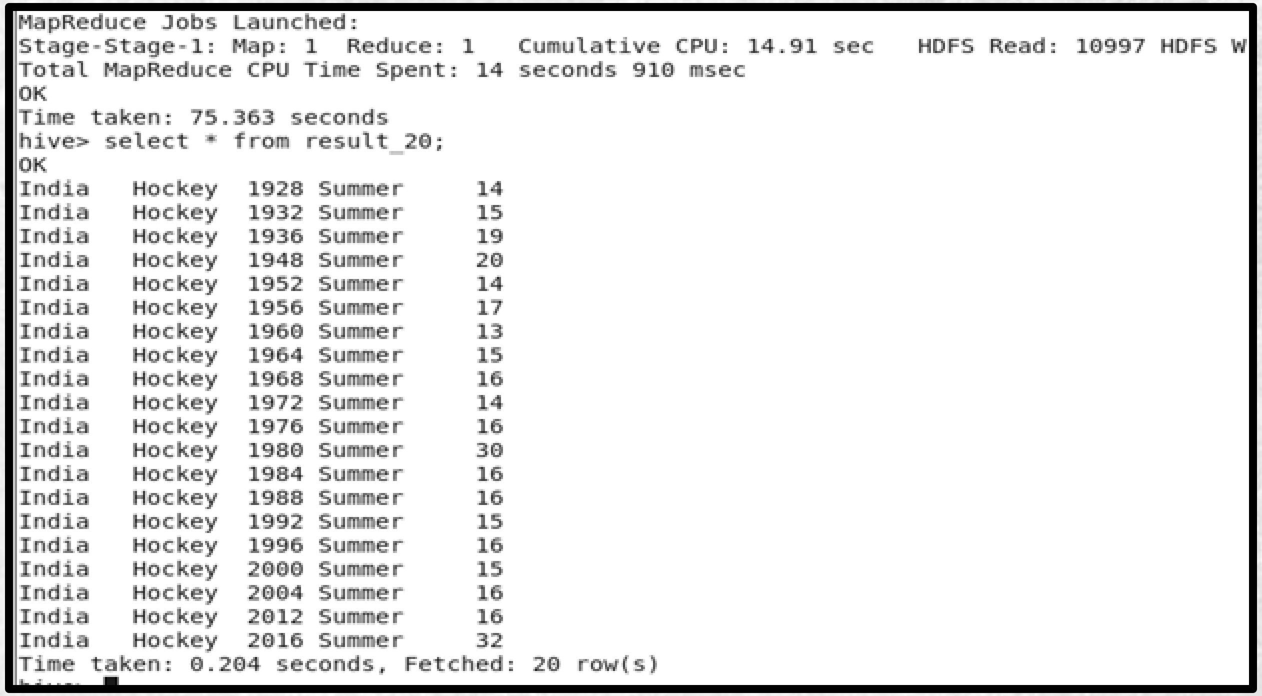
```
hive> create external table olympics_hbase
    > (
    > num int,
    > id int,
    > name        STRING,
    > sex         STRING,
    > age         int,
    > height      int,
    > weight      int,
    > team        STRING,
    > noc         STRING,
    > games       STRING,
    > year        INT,
    > season      STRING,
    > city        STRING,
    > sport       STRING,
    > event       STRING,
    > medal       STRING
    > )
    > row format delimited fields terminated by ','
    > stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > with SERDEPROPERTIES
    > ("hbase.columns.mapping"="persondetails:id,persondetails:name,persondetails:sex,persondetails:age,persondetails:height,persondetails:weight,persondetails:team,persondetails:noc,gamedetails:games,gamedetails
:year,gamedetails:season,gamedetails:city,gamedetails:sport,gamedetails:event,gamedetails:medal")
    > TBLPROPERTIES("hbase.table.name"="olympics_history");
OK
Time taken: 0.287 seconds
hive> select * from olympics_hbase limit 5;
OK
1       1       A Dijiang       M       24      180     80      Chi     CHN     1992 Summer     1992    Summer  Barcelo         Basketball      Basketball Men's Basketball
10      5       Christine Jacoba Aaftink        F       27      185     82      Netherlands     NED     1994 Winter     1994    Winter  Lillehammer     Speed Skating   Speed Skating Women's 1000 metres
100     35      Dagfinn Sverre Aarskog  M       24      190     98      Norway  NOR     1998 Winter     1998    Winter   gano   Bobsleigh       Bobsleigh Men's Four
1000    562     Pawe Abratkiewicz       M       27      183     84      Poland  POL     1998 Winter     1998    Winter   gano   Speed Skating   Speed Skating Men's 500 metres
10000   5470    Ryo Asano       M       25      0       0       Japan   JPN     1988 Summer     1988    Summer  Seoul   Sailing Sailing Mixed Windsurfer
Time taken: 0.274 seconds, Fetched: 5 row(s)
```

## Break down all olympic games where india won medal for Hockey and how many medals in each olympic games.

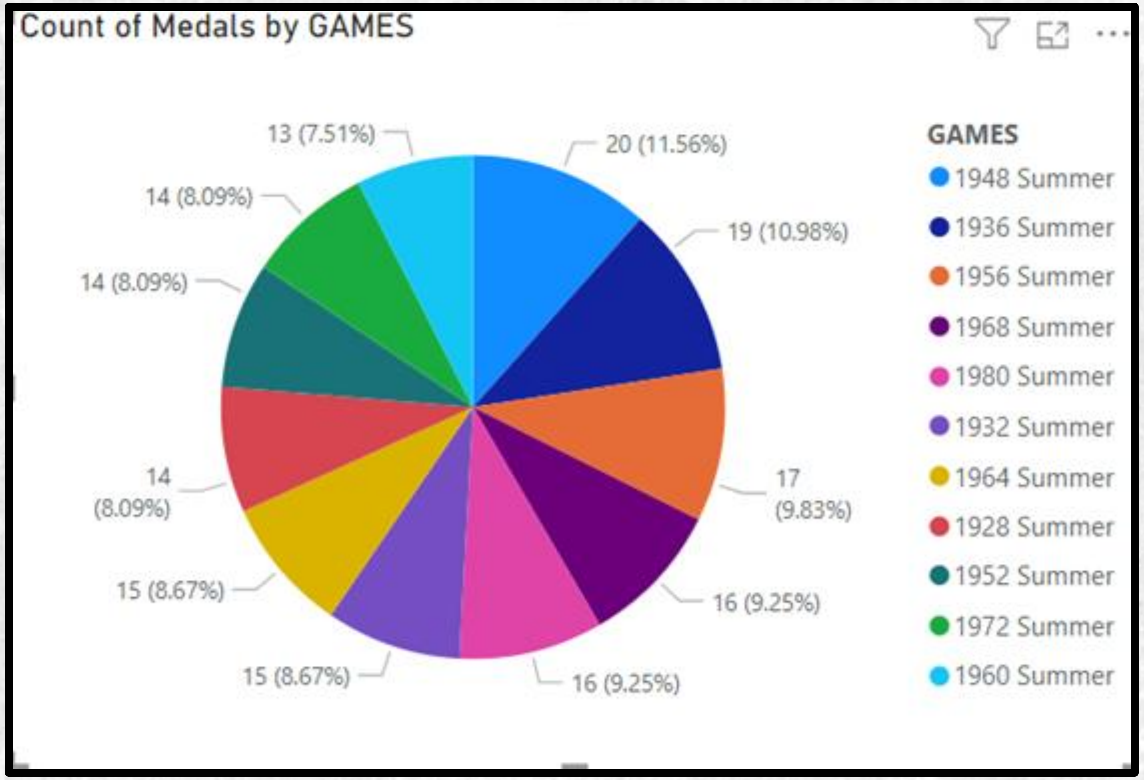


```
hive> create table result_20 as
    > select team,sport,games, count(*) as count_medals from olympics_hbase
    > where team='India' and medal!='NA' and sport='Hockey' group by team,sport,games;
Query ID = cloudera_20230920220505_0e3dbb7b-4cd6-4350-a881-a5838fb1a6df
Total jobs = 1
```

**Output:**



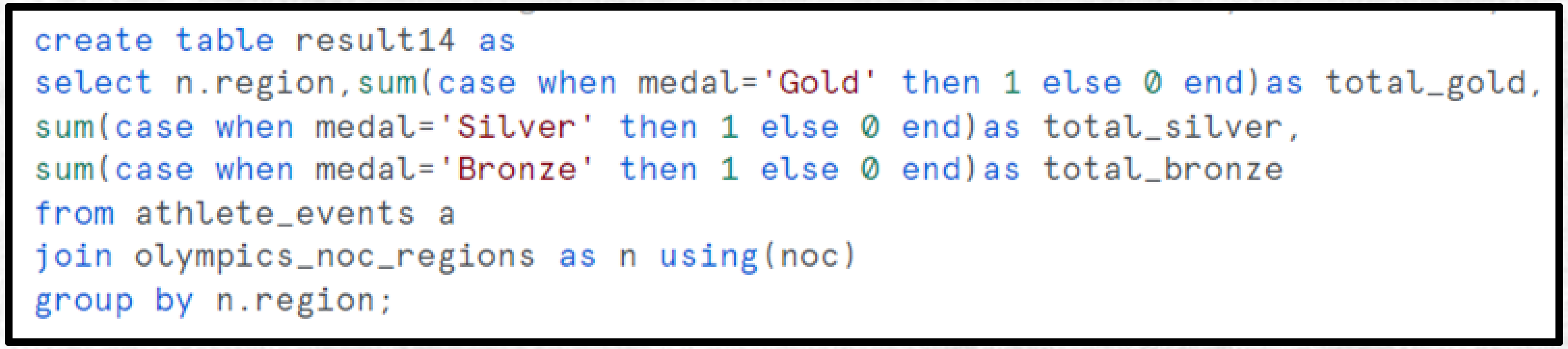
```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 14.91 sec   HDFS Read: 10997 HDFS W
Total MapReduce CPU Time Spent: 14 seconds 910 msec
OK
Time taken: 75.363 seconds
hive> select * from result_20;
OK
India   Hockey  1928 Summer     14
India   Hockey  1932 Summer     15
India   Hockey  1936 Summer     19
India   Hockey  1948 Summer     20
India   Hockey  1952 Summer     14
India   Hockey  1956 Summer     17
India   Hockey  1960 Summer     13
India   Hockey  1964 Summer     15
India   Hockey  1968 Summer     16
India   Hockey  1972 Summer     14
India   Hockey  1976 Summer     16
India   Hockey  1980 Summer     30
India   Hockey  1984 Summer     16
India   Hockey  1988 Summer     16
India   Hockey  1992 Summer     15
India   Hockey  1996 Summer     16
India   Hockey  2000 Summer     15
India   Hockey  2004 Summer     16
India   Hockey  2012 Summer     16
India   Hockey  2016 Summer     32
Time taken: 0.204 seconds, Fetched: 20 row(s)
```



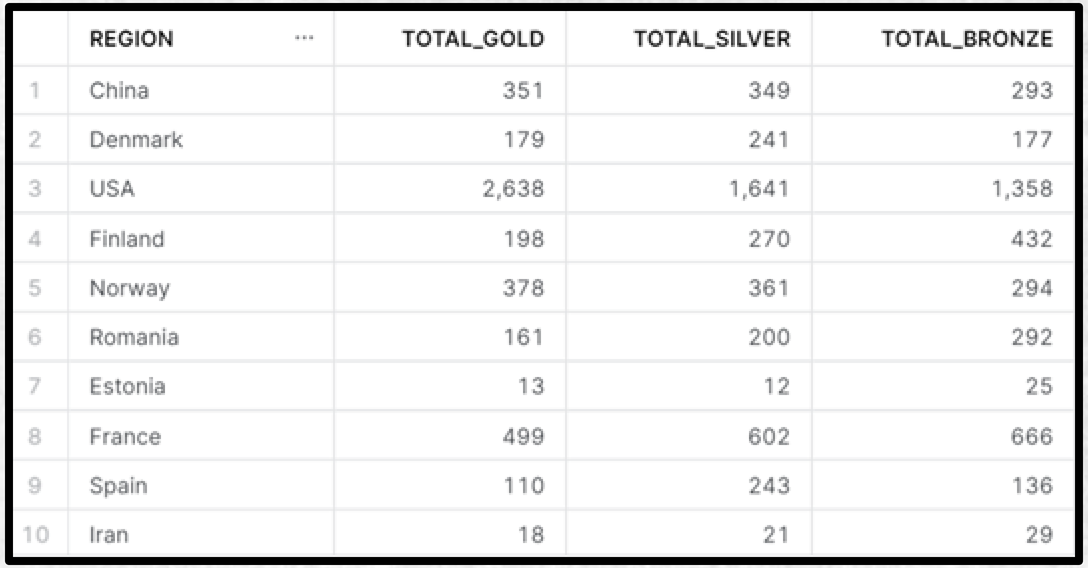
The Godfather of Talent | Futurense

# Snowflake User Story

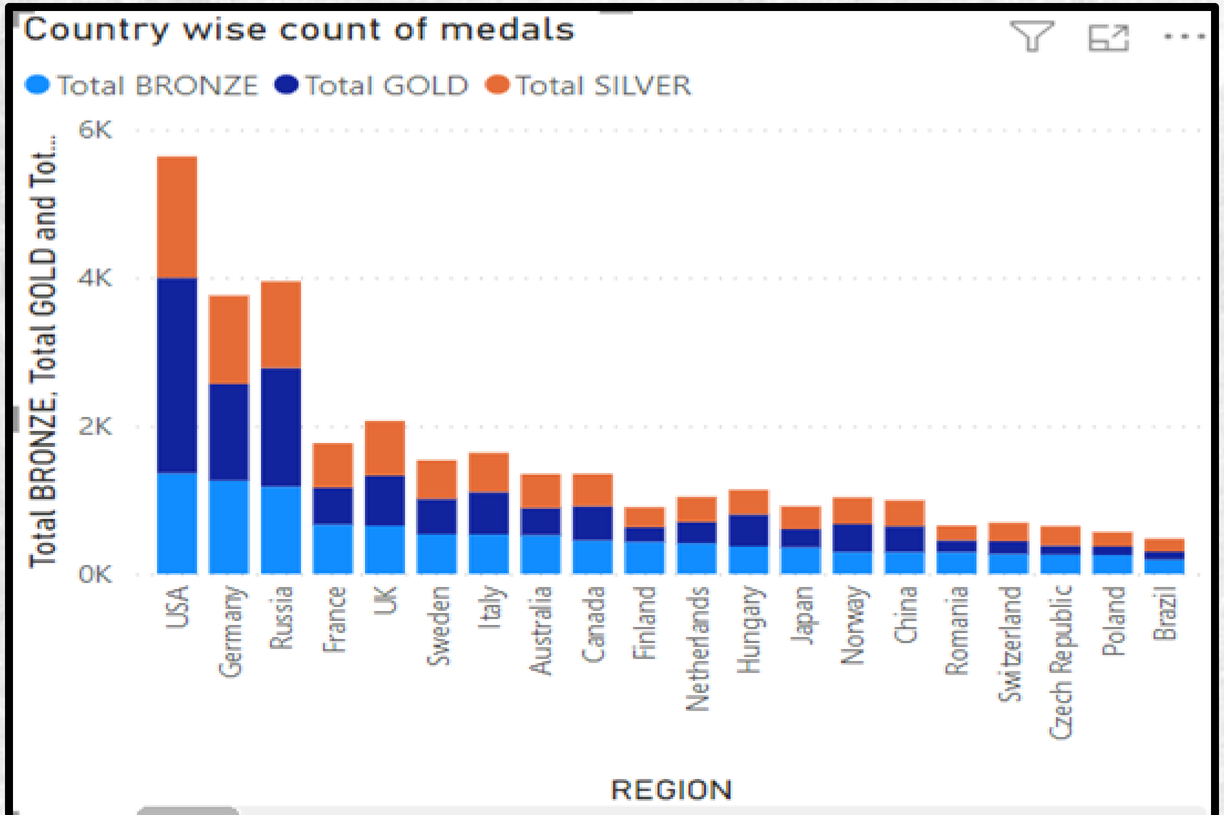
## List down total gold, silver and bronze medals won by each country.



```sql
create table result14 as
select n.region,sum(case when medal='Gold' then 1 else 0 end)as total_gold,
sum(case when medal='Silver' then 1 else 0 end)as total_silver,
sum(case when medal='Bronze' then 1 else 0 end)as total_bronze
from athlete_events a
join olympics_noc_regions as n using(noc)
group by n.region;
```



| | REGION | TOTAL_GOLD | TOTAL_SILVER | TOTAL_BRONZE |
|---|---|---|---|---|
| 1 | China | 351 | 349 | 293 |
| 2 | Denmark | 179 | 241 | 177 |
| 3 | USA | 2,638 | 1,641 | 1,358 |
| 4 | Finland | 198 | 270 | 432 |
| 5 | Norway | 378 | 361 | 294 |
| 6 | Romania | 161 | 200 | 292 |
| 7 | Estonia | 13 | 12 | 25 |
| 8 | France | 499 | 602 | 666 |
| 9 | Spain | 110 | 243 | 136 |
| 10 | Iran | 18 | 21 | 29 |



# Optimizations and Best Practices Used:

> As our solution is dealing with Big Data we have implemented optimizations like Partitioning and Bucketing for hive table for effective data access while analysis.

> We have used ORC file format which will give best compression and performance while analysis.

> Used Map side join while joining NOC_Region table .

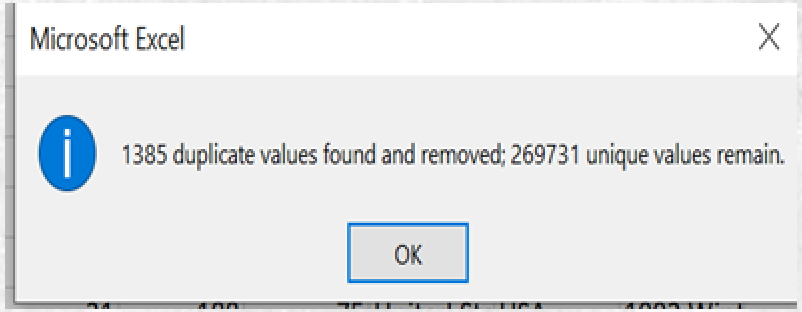
> While writing queries we have used Common Table Expression to optimize the query.

> Using external table to store the results as a best practice.

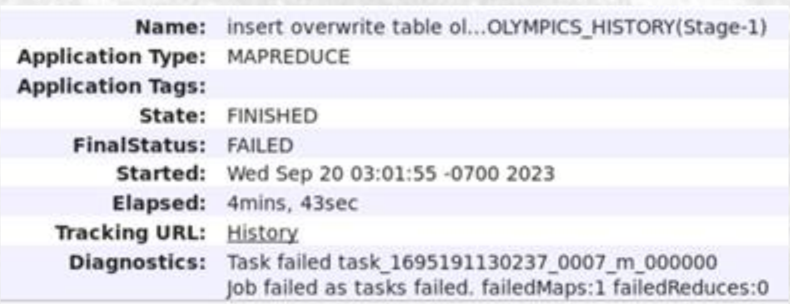
> We have used standard naming conviction as a best practice for readability and understanding.

# Roadblocks & Resolutions

Snapshots:



1385 duplicate values found and removed; 269731 unique values remain.



```
          Name:  insert overwrite table ol...OLYMPICS_HISTORY(Stage-1)
Application Type:  MAPREDUCE
Application Tags:
          State:  FINISHED
     FinalStatus:  FAILED
        Started:  Wed Sep 20 03:01:55 -0700 2023
        Elapsed:  4mins, 43sec
    Tracking URL:  History
     Diagnostics:  Task failed task_1695191130237_0007_m_000000
                   Job failed as tasks failed. failedMaps:1 failedReduces:0
```

**Roadblock 1:**
Data set given has duplicate records and commas in values of particular column.

**Resolution 1:**
We have removed duplicates and replaced comma as part of cleaning.

**Roadblock 2:**
As per user stories we have decided to use multiple columns for partitioning and use ORC file format but the we have encountered insufficient heap size error.

**Resolution 2:**
We can not compresize on fileformat so we have partitioned on best column to be partition.

The Godfather of Talent | Futurense

Venkata S Billa, DE Expert

Team Data Lake

Moksha HS
Trainee Data Specialist

Sakshi Shinde
Trainee Data Specialist

Sai Venkat Seelam
Trainee Data Specialist

Anirudh Kuruva
Trainee Data Specialist

Koustav Sarkar
Trainee Data Specialist

# Thank you!