Enterprise Data Warehouse for Olympic Games

# Description: -

*•* Enterprise Data Warehouse for Olympic game management is an independent software system developed to store the history of all activities happening in the Olympic game like Player, country, game, schedule and Medal .

Ranking. The data stored could be subsequently used for reporting.

*•* In this project I have used AWS S3 service, AWS Glue, AWS Redshift. AWS Glue is fully managed ETL service .

that categorize Olympic data, clean it, enrich it, and move to Redshift data warehouse by using JDBC.

*•* AWS Redshift data warehousing system is being developed to run complex SOL queries and enhance the decision support.

# Step-by-Step Process:-

**1.** **Set Up AWS S3 for Data Storage**

- Service: Amazon S3

- Steps:

1. Create an S3 bucket to store raw Olympic data.

2. Upload the raw data files (e.g., CSV, JSON) to the S3 bucket.

**2. Use AWS Glue for ETL Operations**

- Service: AWS Glue

- Steps:

1. Create a Glue Crawler:

- Configure the crawler to scan the S3 bucket and identify the data schema.

- Run the crawler to create metadata in the AWS Glue Data Catalog.

2. Set Up Glue ETL Jobs:

- Create a Glue ETL job to process the raw data.

- Use Glue scripts to clean, transform, and enrich the data.

- Configure the job to load the processed data into Amazon Redshift using JDBC.

**3. Set Up AWS Redshift for Data Warehousing**

- Service: Amazon Redshift

- Steps:

1. Create a Redshift Cluster:

- Launch an Amazon Redshift cluster.

- Configure the cluster (node type, number of nodes, etc.).

2. Create Redshift Tables:

- Define the schema for the tables in Redshift based on the processed data.

- Use SQL commands to create tables in Redshift.

3. Load Data into Redshift:

- Use the COPY command in Redshift to load data from S3.

- Alternatively, configure AWS Glue ETL jobs to load data directly into Redshift via JDBC.

**4. Run Queries and Generate Reports**

- Service: Amazon Redshift

- Steps:

1. Use SQL to run complex queries on the Redshift data warehouse.

2. Generate reports and perform data analysis to support decision-making.

----------------------------------------------------------------------------------------------------------------

# Summary of AWS Services Used :

1. Amazon S3: For storing raw Olympic data files.

2. AWS Glue: For crawling, cataloging, transforming, and loading data.

- Glue Crawler: To discover and catalog metadata.

- Glue ETL Jobs: To clean, transform, and move data to Redshift.

3. Amazon Redshift: For storing and querying the processed data.

- Redshift Cluster: The data warehousing environment.

- Redshift COPY Command: For loading data from S3 to Redshift tables.

----------------------------------------------------------------------------------------------------------------

# Example Workflow :-

1. Upload Data to S3:

- Raw data about players, countries, games, schedules, and medal rankings are uploaded to an S3 bucket.

2. AWS Glue Crawler:

- A Glue Crawler scans the S3 bucket, identifies the schema, and creates metadata in the Glue Data Catalog.

3. AWS Glue ETL Job:

- A Glue ETL job reads the raw data from S3, cleans and transforms it, and writes the processed data back to S3 or directly to Redshift using JDBC.

4. Amazon Redshift:

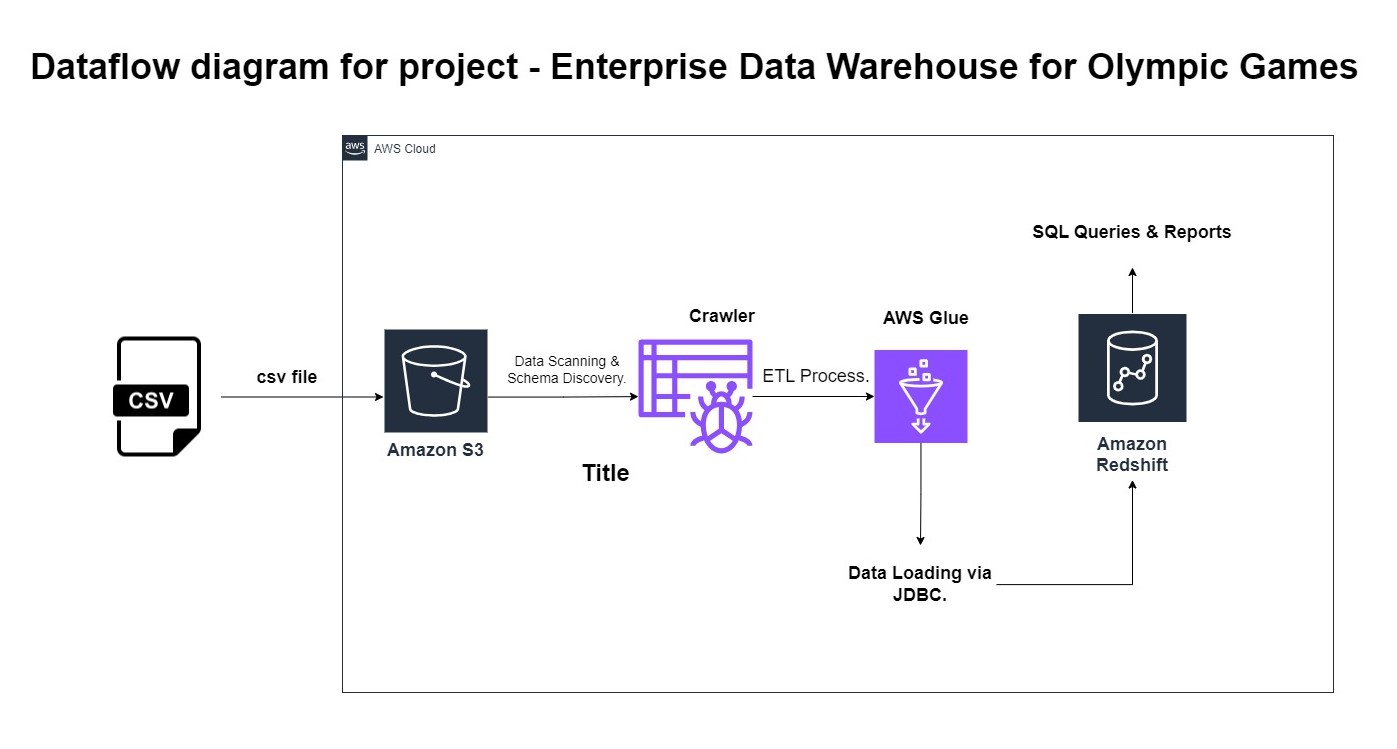
- Create tables in Redshift to match the schema of the processed data.

- Use the Redshift COPY command to load data from S3 into Redshift, or have the Glue job load data directly into Redshift.

5. Query and Analyze:

- Run complex SQL queries on the data stored in Redshift to generate insights and reports.

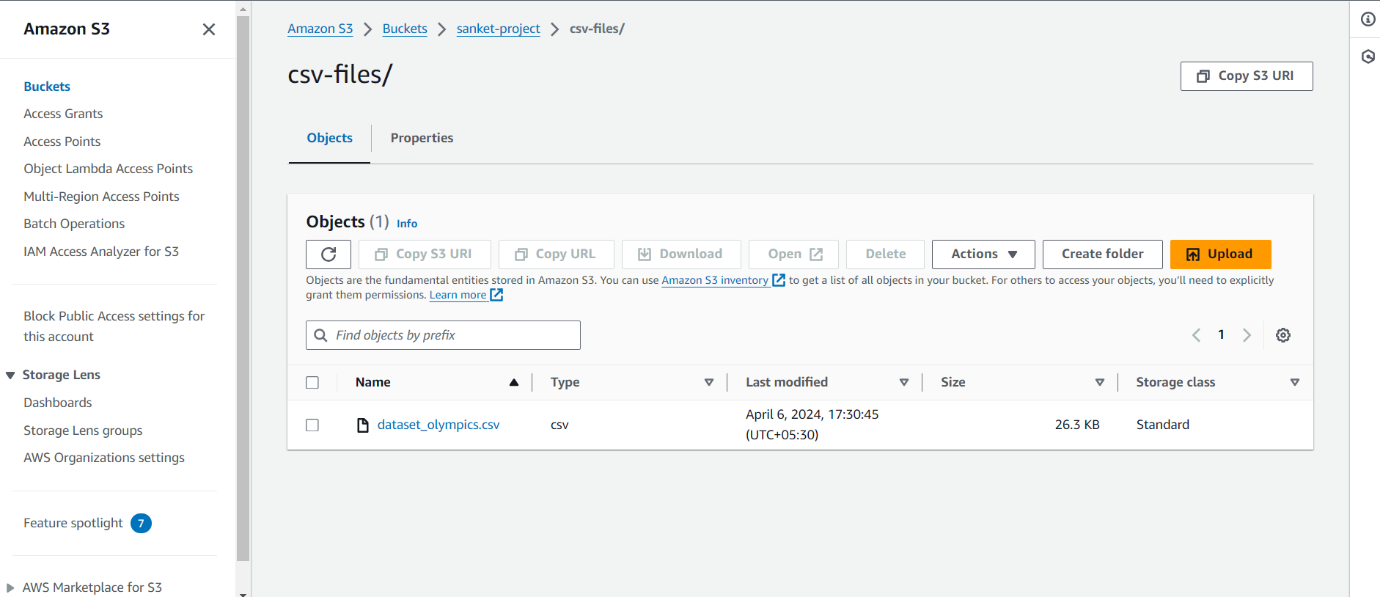
By following these steps and using these AWS services, you can build a robust Enterprise Data Warehouse for managing and analyzing Olympic Games data.



# Implementation Screenshots of project -

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Now further, will use AWS Redshift to store and run complex queries.

But AWS Redshift is not under AWS Free tier , Hence I skipped that part .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*