Project: Olympic Games Analysis

ETL and Datawarehouse

Faiza Ayoun, Maria Holubowicz, and Deep Bambharoliya

Course: Database Systems for Analytics.
San Jose State University.

This document describes the steps to implement ETL using AWS Glue to load the athletes_events dataset into the Redshift data warehouse.

1. **Upload the athletes_events.csv file to a directory in an S3 bucket.**
   Created an S3 bucket and uploaded the atheltes_events.csv from Kaggle into a directory in the S3 bucket. Remove all duplicates with Python first.
2. **Create the IAM Role that allows crawler to access the data in S3**
   Following the tutorial in https://docs.aws.amazon.com/glue/latest/dg/create-service-policy.html we created the IAM role.

   The IAM role is called 'AWSGlueServiceRole-olympics' that has the AWSGlueServiceRole policy and AmazonS3FullAccess policy to access S3.

3. **Create a crawler called 'athletes-events'**

   This crawler reads the athletes_events.csv file stored in S3 and adds a table in the Data Catalog.

   - Specify the crawler source: Select the directory in the S3 bucket where the data is located.
   - Attach the IAM role created in the previous step.
   - Select the run on demand.
   - Configure the output for the crawler. In this case add the database 'olympics' which was created previously.
   - Run the crawler.

Name	athletes-events
Description	Reads the athletes_events.csv file.
Create a single schema for each S3 path	false
Table level	
Security configuration	
Tags	-
State	Ready
Schedule	
Last updated	Sat Nov 27 13:18:17 GMT-800 2021
Date created	Fri Nov 26 19:55:32 GMT-800 2021
Database	olympics
Service role	service-role/AWSGlueServiceRole-olympics
Selected classifiers	
Data store	S3
Include path	s3://olympics-bucket/athletes
Connection	
Exclude patterns	

Configuration options

Schema updates in the data store	Update the table definition in the data catalog for all data stores except S3. For tables that map to S3 data, add new columns only.
Object deletion in the data store	Mark the table as deprecated in the data catalog.

- The crawler added the table ‘athletes’ in the Glue Data Catalog. We named the columns because the crawler didn’t detect the names. The figure below shows the schema created by the crawler.

Schema

Showing: 1 - 16 of 16

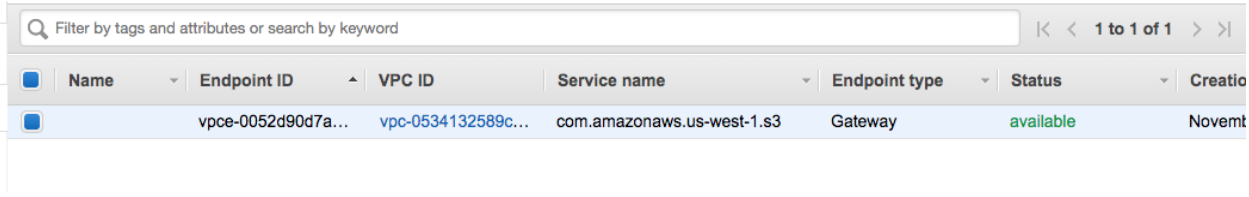
	Column name	Data type	Partition key	Comment
1	index	int		
2	id	int		
3	name	string		
4	sex	string		
5	age	float		
6	height	float		
7	weight	float		
8	team	string		
9	noc	string		
10	games	string		
11	year	int		
12	season	string		
13	city	string		
14	sport	string		
15	event	string		
16	medal	string		

4. **Create a Redshift cluster ‘redshift-cluster-olympics’**
5. **Create a database called ‘olympics’ with a table called ‘athletes_events’**

```
CREATE table athletes_events(
id int not null,
athlete_id int,
name varchar(150),
sex char(1),
age int,
height int,
weight int,
team varchar(100),
NOC varchar(3),
games varchar(100),
year int,
season varchar(100),
city varchar(100),
sport varchar(100),
event varchar(100),
medal varchar (100),
primary key(id)
);
```

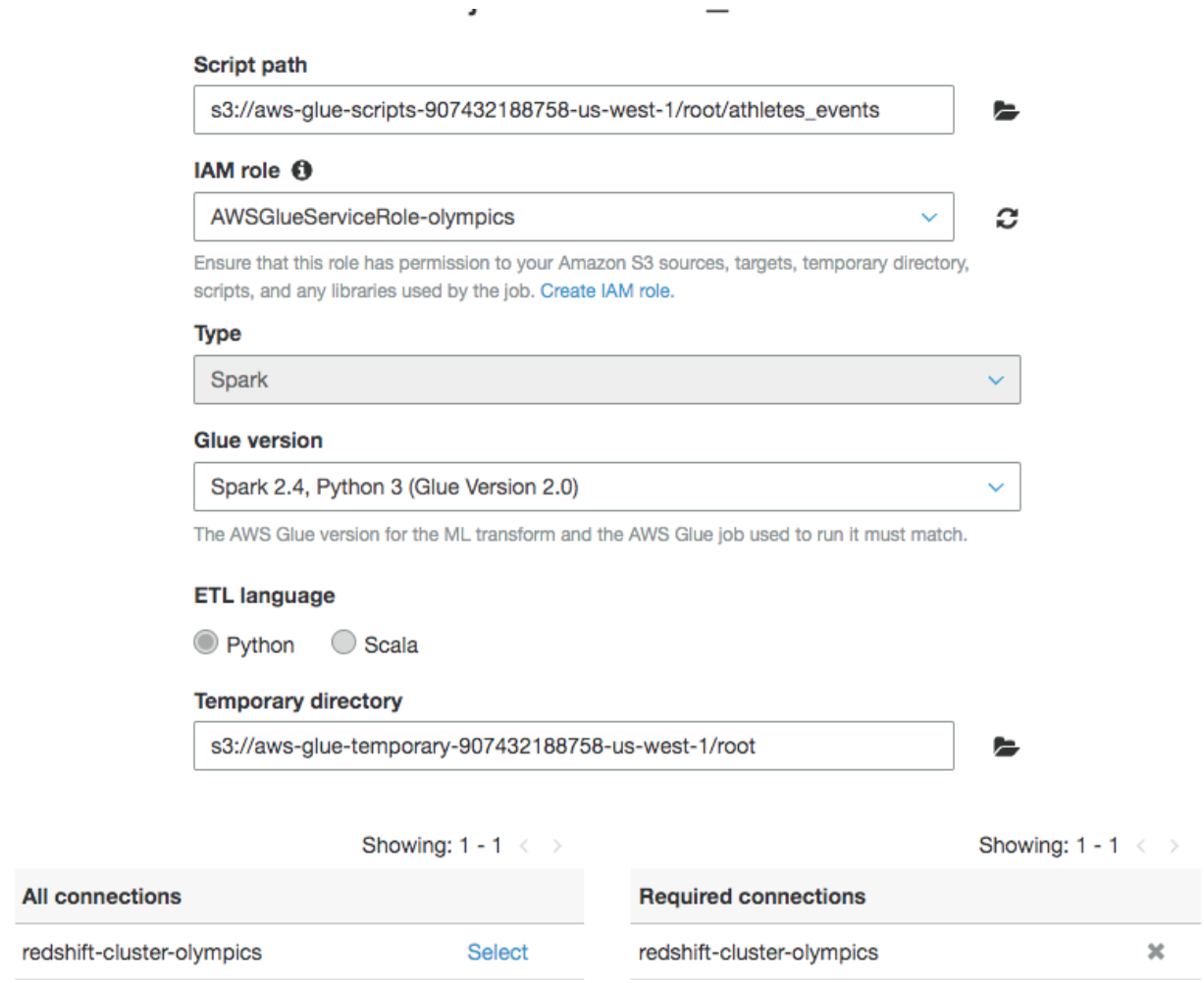
5. Create a VPC gateway to allow Glue to connect to S3

We followed the steps in (https://aws.amazon.com/premiumsupport/knowledge-center/glue-s3-endpoint-validation-failed/) in order to create the VPC gateway.

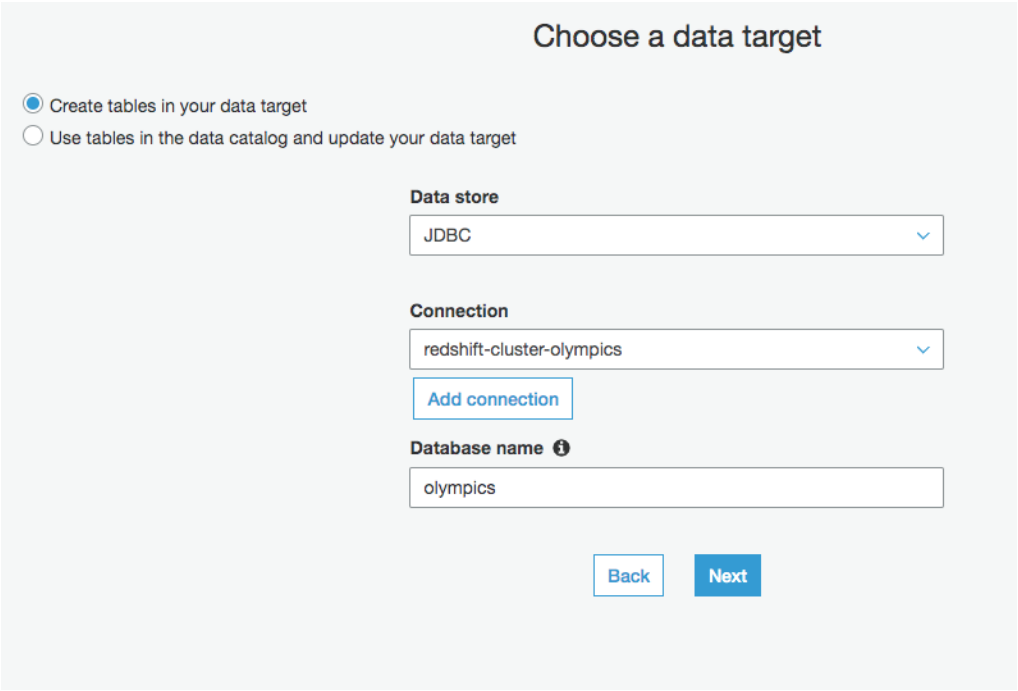


6. Create the Glue job to transform and load the data into Redshift

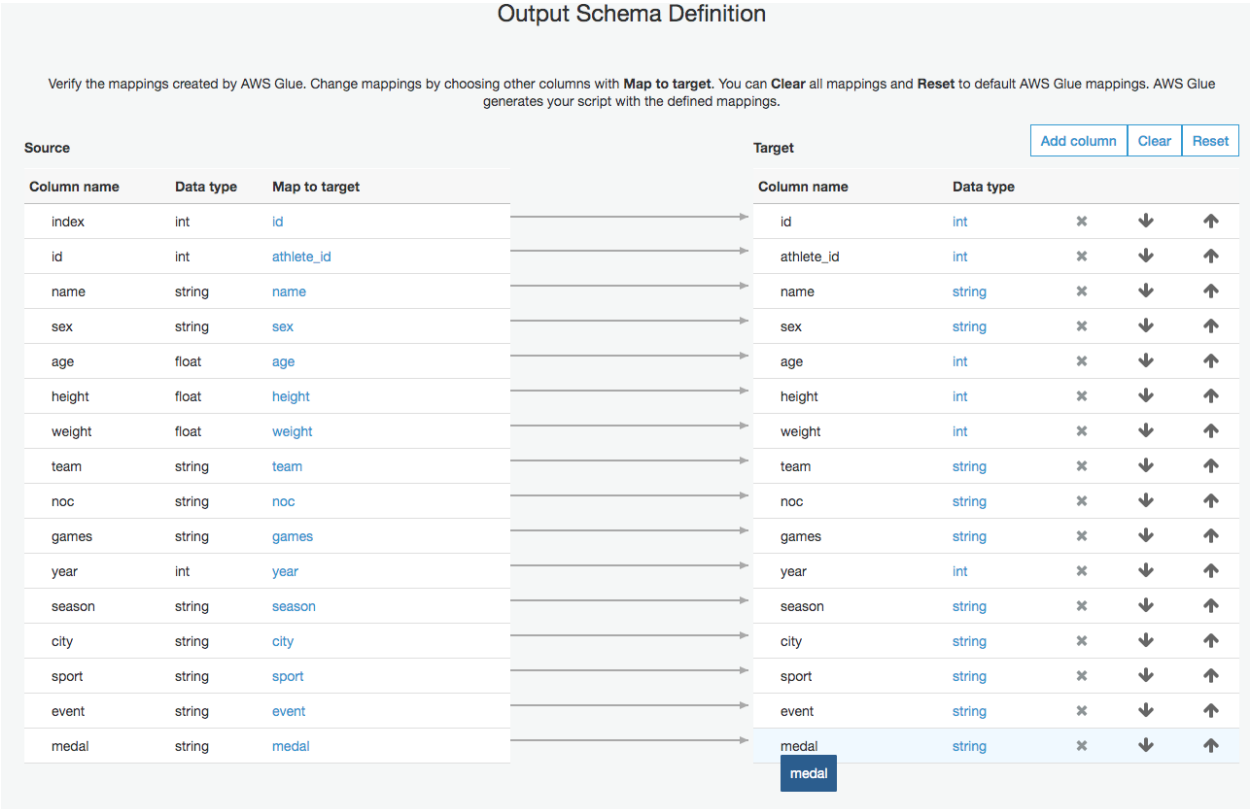
- Use the IAM-Role that has permissions to S3
- Choose Spark for GLue version, ETL language Python. Figure below illustrates this step.



- Choose the data source athletes_evetns.csv in S3, check the option ‘change schema’ click Next.
- For the data target choose create tables in your data target. For the Data store choose JDBC, connection: redshift-cluster-olympics, Database name: Olympics. Click Next. Below is the screenshot that illustrates this part.



- Set the changes to the schema: change data type of age, height, weight to int and change the column names so that it matches the column names in the table on Redshift. Changed index to id and id to athlete_id.



- Run job and the job will load the data into Redshift data warehouse.

Difficulties

- Omitted the file header as the job was failing. The glue job was reading the first row of the csv file in S3 and failed when trying to load it into Redshift. Deleted the header in the csv file.
- Changed the length of the name column in the table in Redshift. Changed from varchar(100) to varchar(150), as the job failed when trying to copy a name that was 109 characters long.

254368,127346,Max Emanuel Maria Alexander Vicot Bruno de la Santisima Trinidad y Todos los Santos von Hohenlohe Langenburg,M,24,@NULL@,@NULL@,Liechtenstein,LIE,1956 Winter,1956,Winter,Cortina d'Ampezzo,Alpine Skiing,Alpine Skiing Men's Downhill,"",@NULL@	Max Emanuel Maria Alexander Vicot Bruno de la Santisima Trinidad y Todos los Santos von Hohenlohe Langenburg	1204

Script link:
https://github.com/holubmaria/Olympic-Analysis-Project/commit/8446f991c6c45ef127f9fe197b4f038809be32a4

Query the data

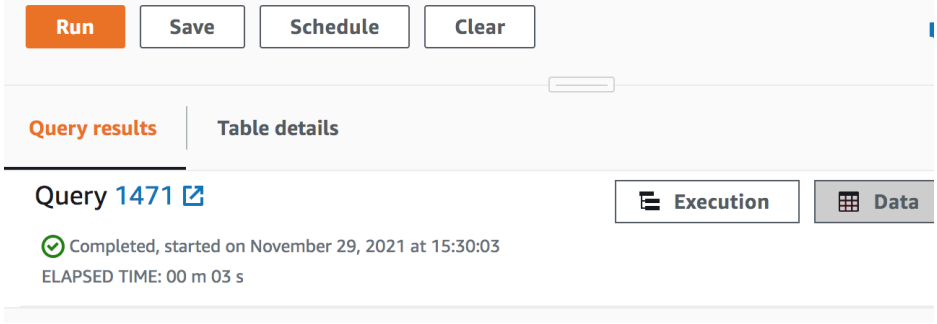
1. Participants by country

```
select count(*) as participants, team from athletes_events
group by team
order by participants desc;
```

participants	team
17598	United States
11817	France
11264	Great Britain
10213	Italy
9230	Germany
9226	Canada
8269	Japan
8004	Sweden
7512	Australia
6492	Hungary

2. **Top 10 country with the most gold medals**

```
1 select count(medal) as medals,  team  from athletes_events
2 group by team order by medals desc limit 10;
```



Query results | Table details

Query 1471

Execution | Data

Completed, started on November 29, 2021 at 15:30:03

ELAPSED TIME: 00 m 03 s

medals	team
17598	United States
11817	France
11264	Great Britain
10213	Italy
9230	Germany
9226	Canada
8269	Japan
8004	Sweden
7512	Australia
6492	Hungary

References:

1. AWS Glue tutorial https://docs.aws.amazon.com/glue/latest/dg/create-service-policy.html

2. Creating a Database in the data Catalog
   https://docs.aws.amazon.com/glue/latest/dg/define-database.html
3. Create Tables in the data catalog
   https://docs.aws.amazon.com/glue/latest/dg/tables-described.html
4. Created Glue jobs https://docs.aws.amazon.com/glue/latest/dg/add-job.html
5. Amazon Redshift tutorial
   https://docs.aws.amazon.com/redshift/latest/gsg/sample-data-load.html