## Integrating Olympic Data with Microsoft SQL Server and SSIS

## Introduction:

In this project, you will explore and integrate historical data from the Olympic Games from 1896 to 2022.

Your task is to create a comprehensive database that captures detailed information about athletes, events, and their performances.   
You will perform data ingestion, transformation, and loading (ETL) processes to clean and consolidate the data from multiple sources, making it suitable for analysis.

## About the Dataset:

It contains detailed information about 154,902 unique athletes, including:

**1. Olympic\_Athlete\_Bio**

* **Table Description:** Contains biographical information about Olympic athletes.
* **Columns:**
  + athlete\_id: Unique identifier assigned to each athlete.
  + name: Full name of the athlete.
  + sex: Gender of the athlete (Male or Female).
  + born: Date of birth of the athlete.
  + height: Height of the athlete in centimeters.
  + weight: Weight of the athlete in kilograms.
  + country: Name of the country the athlete represents.
  + country\_noc: National Olympic Committee code representing the country.
  + description: A paragraph describing the athlete.
  + special\_notes: Special notes about the athlete in bullet points.

**2. Olympic\_Athlete\_Event\_Results**

* **Table Description:** Records the results of athletes in various Olympic events.
* **Columns:**
  + edition: The edition of the Olympic Games.
  + edition\_id: Unique identifier for the edition.
  + country\_noc: National Olympic Committee code for the country represented.
  + sport: The sport in which the athlete competed.
  + event: Specific event in the sport.
  + result\_id: Unique identifier for the event result.
  + athlete: Name of the athlete.
  + athlete\_id: Unique identifier for the athlete.
  + pos: Position or rank achieved by the athlete in the event.
  + medal: Type of medal won (Gold, Silver, Bronze, or None).

**3. Olympic\_Games\_Medal\_Tally**

* **Table Description:** Provides a summary of medal counts by country for each edition of the Olympic Games.
* **Columns:**
  + edition: The edition of the Olympic Games.
  + edition\_id: Unique identifier for the edition.
  + year: The year the Olympic Games took place.
  + country: Name of the country.
  + country\_noc: National Olympic Committee code for the country.
  + gold: Number of gold medals won by the country.
  + silver: Number of silver medals won by the country.
  + bronze: Number of bronze medals won by the country.
  + total: Total number of medals won by the country.

**4. Olympic\_Results**

* **Table Description:** Contains detailed information about event results in the Olympic Games.
* **Columns:**
  + result\_id: Unique identifier for each result.
  + event\_title: Title of the event.
  + edition: The edition of the Olympic Games.
  + edition\_id: Unique identifier for the edition.
  + sport: The sport category.
  + sport\_url: URL link to additional sport information.
  + result\_date: Date when the result was recorded.
  + result\_location: Location where the event took place.
  + result\_participants: Number of participants in the event.
  + result\_format: Format of the event result (e.g., time, score).

**5. Olympics\_Country**

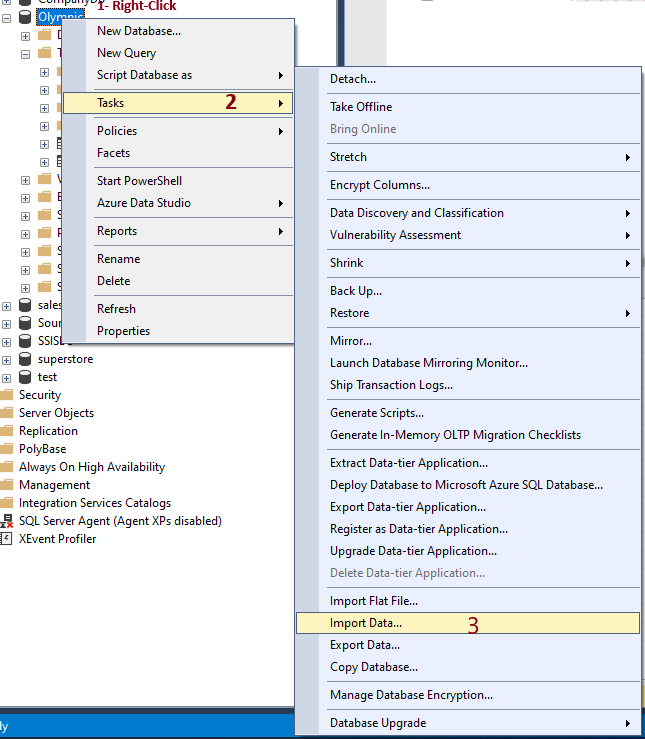
* **Table Description:** Lists countries and their corresponding National Olympic Committee codes.
* **Columns:**
  + noc: National Olympic Committee code.
  + country: Name of the country.

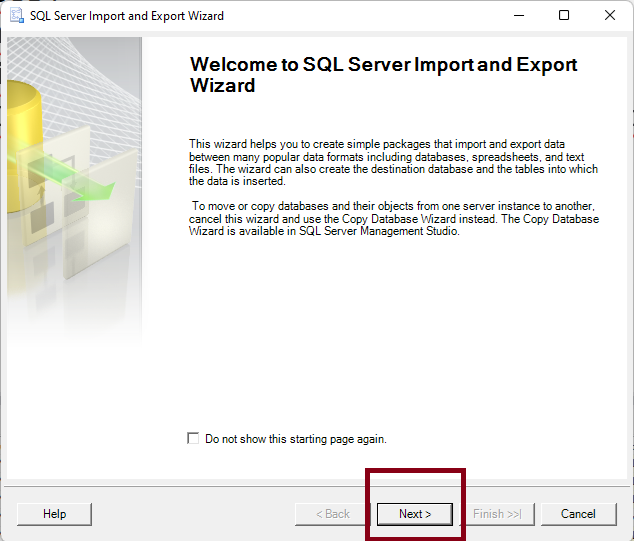
**6. Olympics\_Games**

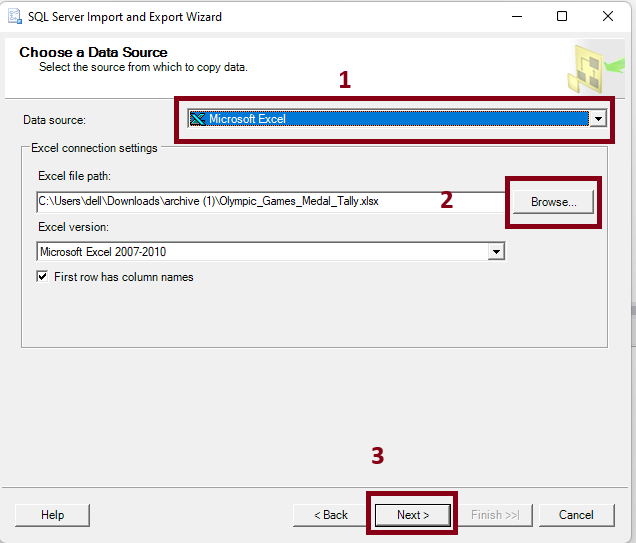
* **Table Description:** Provides information about each edition of the Olympic Games.
* **Columns:**
  + edition: The edition of the Olympic Games.
  + edition\_id: Unique identifier for the edition.
  + edition\_url: URL link to more details about the edition.
  + year: The year the Olympic Games took place.
  + city: The host city of the Olympic Games.
  + country\_flag\_url: URL link to the image of the host country's flag.
  + country\_noc: National Olympic Committee code for the host country.
  + start\_date: Start date of the Olympic Games.
  + end\_date: End date of the Olympic Games.
  + competition\_date: Date(s) of the competitions.

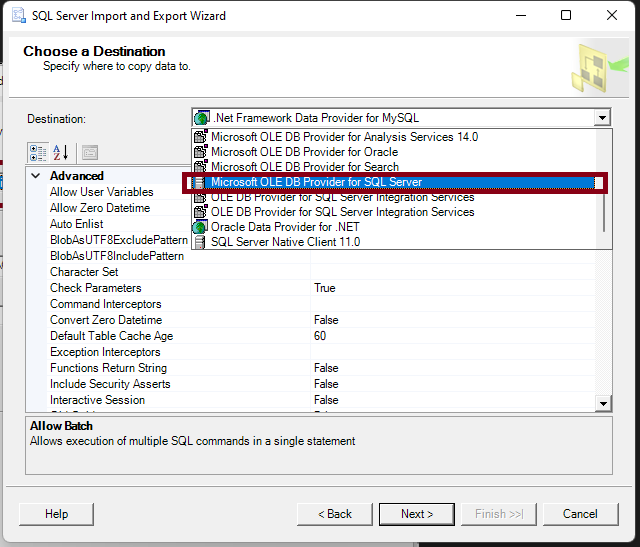
## Project Preparation:

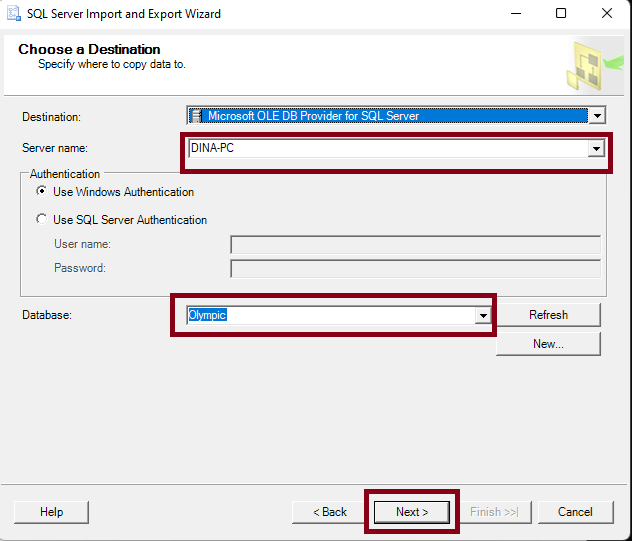
### Import the dataset to the Microsoft SQL Server:

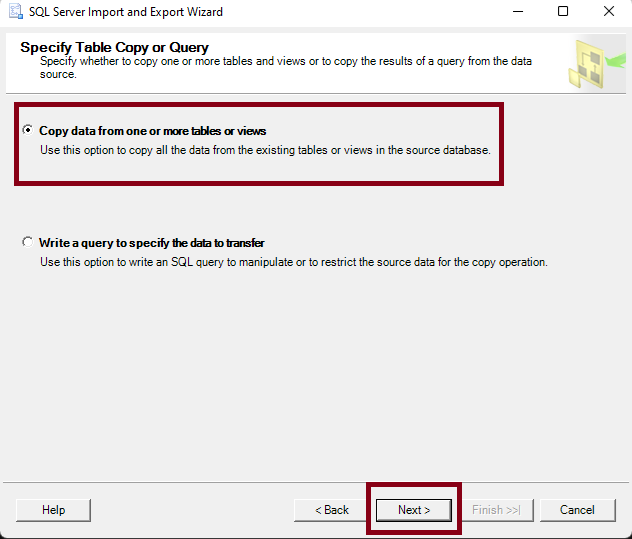
1. **Create a database called:** *Olympic.*
2. **Import all the Excel files into this database, follow these steps to import the files one by one:**

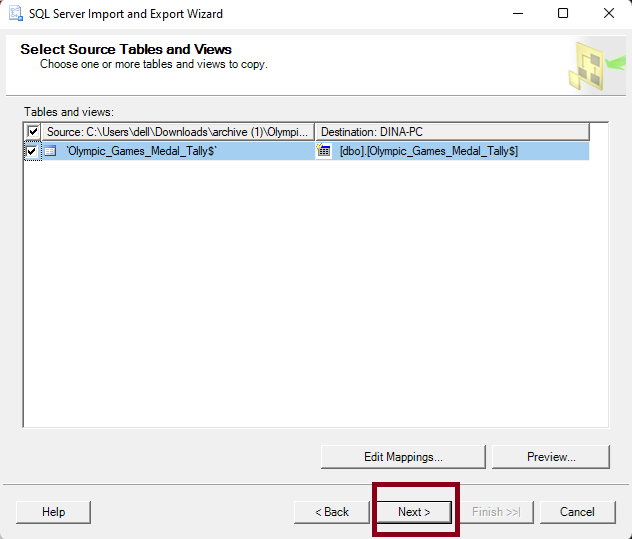


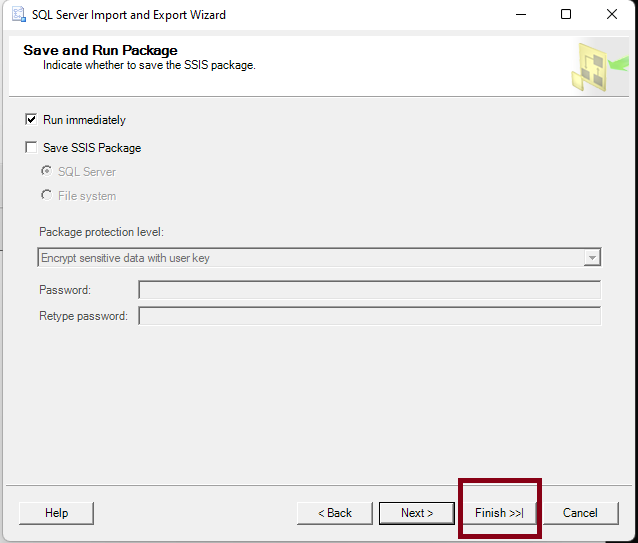


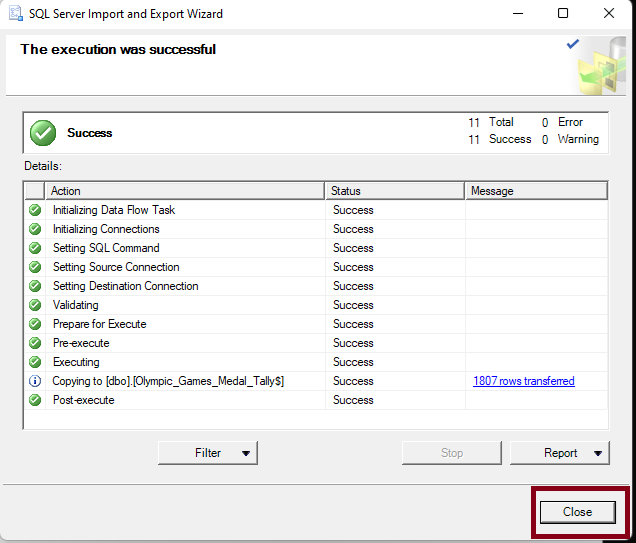












## Project Requirements:

### Basic Requirements:

1. **Create a new table to store the average height and weight of athletes by country.  
   The table name is** *Athlete\_Avg\_Stats***, and it contains** *country\_noc***,** *avg\_height***, and** *avg\_weight* **columns.**
2. **Insert a new record into the** *Olympics\_Country* **table for a newly recognized country.  
   The noc is ‘NEW’ and the country is ‘Newland’/**
3. **Find all distinct sports from the** *Olympic\_Results* **table where the number of participants is greater than 20.**
4. **Classify athletes' performance in** *Olympic\_Athlete\_Event\_Results* **as 'Winner', 'Runner-up', 'Finalist', or 'Participant' based on their position.  
   (1 = Winner, 2 = Runner-Up, less than or equal 8 = Finalist, other is Participant).**
5. **Retrieve the top 10 events with the most participants, ordered by the number of participants in descending order.**
6. **Get the top 5 athletes who have won the most medals.**
7. **Replace null values in the description column of Olympic\_Athlete\_Bio with 'No description available'. (Use SELECT)**
8. **Convert the born date in Olympic\_Athlete\_Bio to the year only.   
   Give an alias ‘birth\_year’ to the new column. (Use SELECT)**
9. **Retrieve the athlete name and their country in one column.**
10. **Retrieve the first three letters of the athletes' names in uppercase.**
11. **Find the current date and time of the server as current\_datetime.**
12. **Create a new table called ‘Country\_Medal\_Count’ with the total number of medals won by each country and insert the data.**
13. **Find the total number of medals won by each country that has won more than 10 medals.**
14. **Rank athletes within each sport by the number of medals won.**
15. **Classify countries based on total medals won as 'High', 'Medium', or 'Low'.  
    (more than 50 = ‘High’, between 20 and 50 = ‘Medium, other = ‘Low’).**
16. **Create a stored procedure to get the medal tally for a specific country and year.**
17. **Store the total number of medals won by a specific country -ex: ‘USA’- in a variable.**
18. **Create a dynamic SQL statement to retrieve medal data for a specific sport.**
19. **Check if a country has won more than 50 medals then print the country name and ‘High Medal Count’, if not, print ‘Low Medal Count’ behind the country name.**
20. **Loop through each athlete in a list and print their name along with their country.**
21. **Find the athletes who have participated in more than one edition of the Olympics.**
22. **List the athletes who have won medals in both Summer and Winter Olympics.**
23. **Create a stored procedure** *UpdateAthleteInfo* **that takes an athlete's ID, a column name, and a new value as input parameters. It updates the specified column for the given athlete with the new value.**
24. **Create a stored procedure** *GetAthletesByMedalType* **that takes a medal type as an input parameter and dynamically generates a report of athletes who have won that type of medal.**

### Business Reports:

1. **Analyze the Performance of Athletes Across Multiple Editions and Determine Medal Trends**

**Description**: Create a report that shows the total number of medals won by athletes who have participated in at least two editions of the Olympics. For each athlete, display their name, country, the total medals won, the average position, and a classification based on the total number of medals as 'Exceptional', 'Outstanding', or 'Remarkable'. Include athletes' participation and performance details. The total medals should order the results won, and handle cases where some athletes might miss data.

1. **Identify Countries with the Most Consistent Medal Performance**

**Description**: Create a report with the countries that have won medals in every edition they participated in. For each country, list the number of editions they participated in, the total number of medals won, and their average medal count per edition. Classify the consistency of performance as 'Highly Consistent', 'Moderately Consistent', or 'Inconsistent' based on the ratio of total medals to editions. Use a dynamic SQL query to display results for either Summer or Winter Olympics based on the input.

**Evaluate Athlete's Peak Performance Ages and Their Influence on Medal Wins**

**Description**: Create a report that finds each athlete's age at the time of their medal wins and determines the age range during which athletes are most likely to win medals. Categorize the athletes into different age brackets (under 20, 20-24, 25-29, 30-34, 35+) and count the total medals won in each bracket. Also, identify the top sport for each age bracket where athletes have won the most medals.

### Build SSIS pipelines:

1. **Create a data flow pipeline that generates a medal summary report for each country by edition and outputs the report in a flat-file format.**

Notes:

1. Use *Olympic\_Games\_Medal\_Tally* table.
2. Calculate the total number of gold, silver, and bronze medals won by each country for each edition.
3. Add a new column to classify countries as "High Performance" or "Low Performance" (more than 50 medals = High Performance, otherwise Low).
4. Output should be a flat file (CSV report with medal summary by country and edition).
5. **Create a data flow pipeline to migrate the** *Olympic\_Athlete\_Event\_Results* **data from the current schema to a new SQL Server schema, performing a few transformations along the way.**

Notes:

1. Use *Olympic\_Athlete\_Event\_Results* table.
2. Ensure that all records have valid medal information (i.e., no null values for the medal column).
3. Change medal type values from "Gold", "Silver", and "Bronze" to integer representations (1 for Gold, 2 for Silver, 3 for Bronze).
4. Output should be a new SQL Server schema containing the migrated data.
5. **Create a data flow pipeline that exports the top 10 athletes by the number of medals won from each sport into a CSV file. It should filter the data, rank the athletes, and output the results.**

Notes:

1. Use *Olympic\_Athlete\_Event\_Results* table.
2. Ensure that you include only athletes who have won medals.
3. Rank athletes by the number of medals won.
4. The output should be a CSV report contains the top 10 athletes for each sport.
5. **Create a data flow pipeline to generate a report that summarizes the total number of medals won by each country in a given year and save it as a flat file. It filters the data for the specified year and aggregates the results.**

Notes:

1. Use *Olympic\_Games\_Medal\_Tally* table.
2. Ensure that you filter by the year given by the user who will run the pipeline.
3. Calculate the total number of medals won by each country for that year.
4. The output should be a CSV file.