"Analysis of Paris 2024 Olympic Games Performance and Strategic Insights for National Teams"

Olympic Data Analysis SQL Case Study

Project Overview

This case study uses data from the Kaggle dataset of the 2024 Olympic Games to conduct an insightful analysis of trends in medal distribution, athlete performance, and overall country performance. The datasets chosen, amongst all the ones available, cover athletes, events, medals, National Olympic Committees (NOCs), and venues, and are critical to understanding performance dynamics that can guide strategic decision-making. For a sponsor, this analysis can provide useful information in line with the company's commitment to sustainability, innovation, and sports excellence. By exploiting this information, the sponsor can optimize its sponsorship strategies, improve its sustainable mobility solutions for large-scale events, and reinforce its brand values globally.

The analysis uses SQL within the Snowflake data platform, chosen for its ability to efficiently handle large data sets and complex queries, ensuring robust and reliable results.

Graphs for each query are reported for a better understanding of the data.

Project Structure

1. Olympic Data Insights

The case study includes a series of SQL queries designed to extract relevant information from the Olympic data:

- **Medal Distribution Analysis**
  - *Medal Trends by Discipline*
  - *Medal Count by Athlete and Gender*
  - *Medal Distribution by Continent*
  - *Running Total of Medals for Each Country*
  - *Top Performing Countries by Sport*
- **Country Performance Analysis**
  - *Performance by Country and Sport*
  - *Medal Efficiency by Country (using CTE)*
- **Event Performance Analysis**
  - *Most Successful Events*
    - *Most successful Events & Sports (France)*
    - *Total Medals of a Country in an Event (France in Men's Events)*
- **Athlete Performance Analysis**
  - *Multi-Medal Athletes*
  - *Age Analysis of Medal Winners*
- **Business-Oriented Analysis**
  - *Medal Revenue Estimation for Sponsors*
  - *Medal Revenue Estimation by Sport and Athlete*
  - *Gender Disparity in Medal Distribution*

2. Database and Schema Setup

The first step in the case study was to create a dedicated database and schema for storing and organizing the data.

- *Database*: ‘olympic_analysis’: main container for all the data related to the 2024 Olympics.

- *Schema:* ‘olympic_data’: organizer for the tables used in the analysis.

- *Staging Area*: ‘olympic_stage’ : temporary storage location where the raw .csv files were uploaded before being loaded into the tables.

3. Data Loading and Staging

After setting up the database and the schema, the next step involved loading data into temporary tables. This approach allows for an initial review and cleaning of the data before it is finalized.

Temporary Tables:

- ‘*nocs_temp’*: holds information about the National Olympic Committees (NOCs), which represent the countries participating in the Olympics.

- ‘*events_temp’*: contains details about the Olympic events.

- ‘*athletes_temp’*: contains detailed information about athletes.

- ‘*medals_temp’*: stores data on medals awarded.

- ‘*venues_temp’*: stores information about the venues where the Olympic events are held.

Data Loading:

the data was loaded into the temporary tables, from the staging area.

4. Data Transformation and Final Table Creation

Once the data was successfully loaded into temporary tables, the next step was to transform and refine this data into final tables, which serve as the foundation for all subsequent analytical queries. These final tables are thoroughly cleaned by selecting distinct records to ensure accuracy and consistency, eliminating any duplicates that could skew analysis results.

Final Tables Created:

- ‘*nocs’:* a clean list of all National Olympic Committees (NOCs), ‘code’ as primary key to ensure each country is uniquely represented. This table has no duplicates, as all records are distinct.

- ‘*events’:* a distinct catalog of Olympic events, with ‘event’ set as the primary key, such that each event is uniquely defined to prevent any duplication (for better clarity in event analysis).

- ‘*athletes’*: a list of all athletes, with ‘code’ as primary key to uniquely identify each athlete. This table also includes a foreign key relationship linking ‘country_code’ to the nocs table, such that each athlete’s country is referenced.

- ‘*medals’:* a list of all medals awarded, with a composite primary key (‘code’ and ‘event’) to ensure each medal is uniquely identified by both the athlete and th event. This table also includes foreign keys relationships, linking ‘code’ to the athletes table, ‘country_code’ to the nocs table, and ‘event’ to the events table.

- ‘*venues’*: a list of venues, with a composite primary key (‘venue’ and ‘sports’), such that each venue and its associated sports are uniquely represented.

Duplicates, using SELECT DISTINCT, are not picked. These final tables maintain the integrity of the data, preventing issues such as inflated medal counts or incorrect participation statistics that could otherwise lead to misleading conclusions. The use of primary keys and foreign keys reinforces data integrity, ensuring that every record is unique and that relationships between tables are consistent and accurate.

5. Analytical Queries for Data Exploration

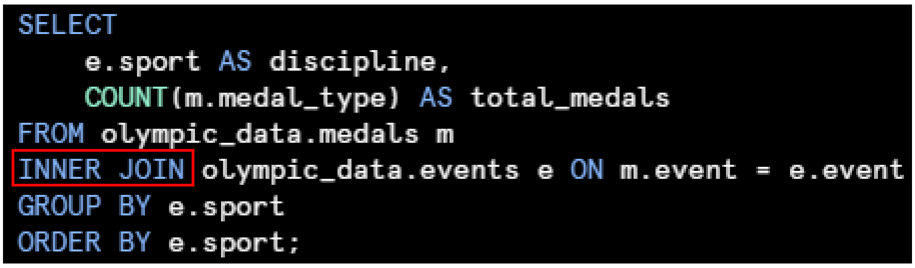
Next step of the pipeline was to develop analytical queries to explore and extract insights from the Olympic data. The queries cover several aspects:

5.1 Medal Distribution Analysis

I. Medal Trends by Discipline

- This query provides the total number of medals awarded in each sport.

  It joins the medals table with the events table to link medals to their respective sports. The query then groups the results by sport and counts the medals, ordering the results, alphabetically, by sport.



```
SELECT
    e.sport AS discipline,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.events e ON m.event = e.event
GROUP BY e.sport
ORDER BY e.sport;
```

Figure 1 – Query

- The INNER JOIN between medals and events links each medal to its specific event and sport. This join allows to grouping medals based on their associated sports (important for the aggregation of the count of medals by sport).

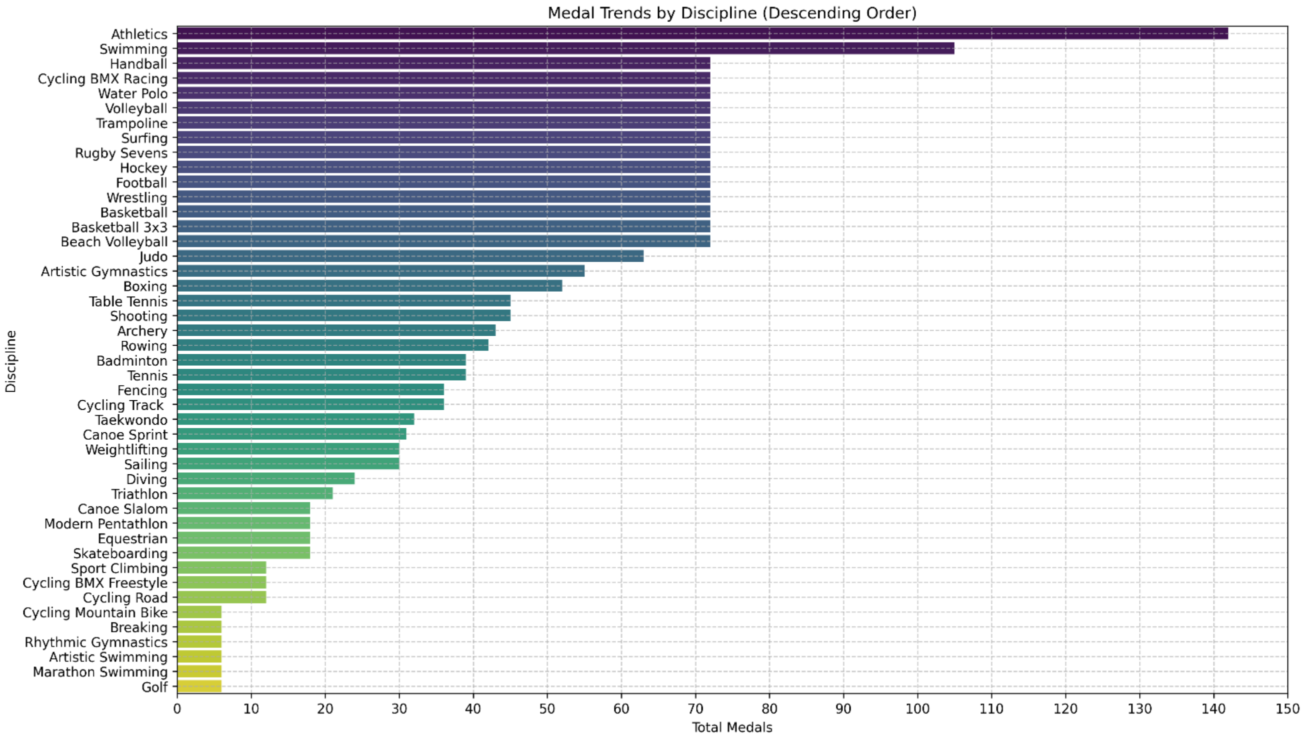
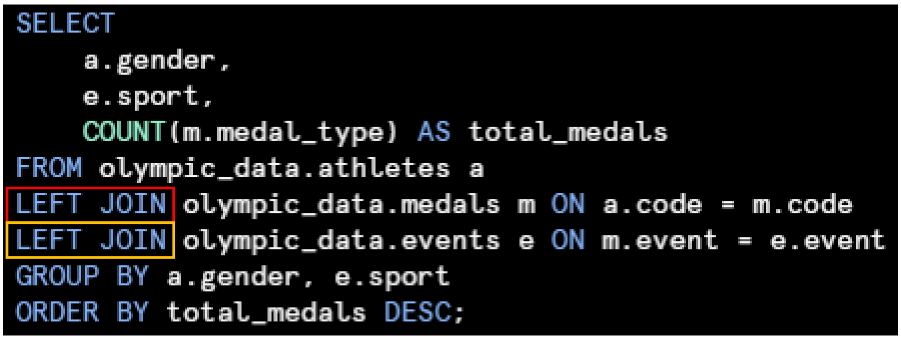


Figure 2 – Visual Representation of the Query

II. Medal Count by Athlete and Gender

- This query counts medals won by athletes, categorized by their gender and sport.

  It performs a LEFT JOIN between athletes and medals to include all athletes, even those without medals, and a LEFT JOIN between medals and events to link medals to their sports. The results are grouped by athlete gender and sport, then ordered by the number of medals in descending order.



```
SELECT
    a.gender,
    e.sport,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.athletes a
LEFT JOIN olympic_data.medals m ON a.code = m.code
LEFT JOIN olympic_data.events e ON m.event = e.event
GROUP BY a.gender, e.sport
ORDER BY total_medals DESC;
```

Figure 3 – Query

- The LEFT JOIN between athletes and medals ensures that all athletes are included in the results, even if they have not won any medals. This is important for counting medals by gender and sport, as it provides a complete picture of medal distribution.
- The second LEFT JOIN (the one with events) is used to link medals to their corresponding sports, allowing the query to aggregate medals by sport.

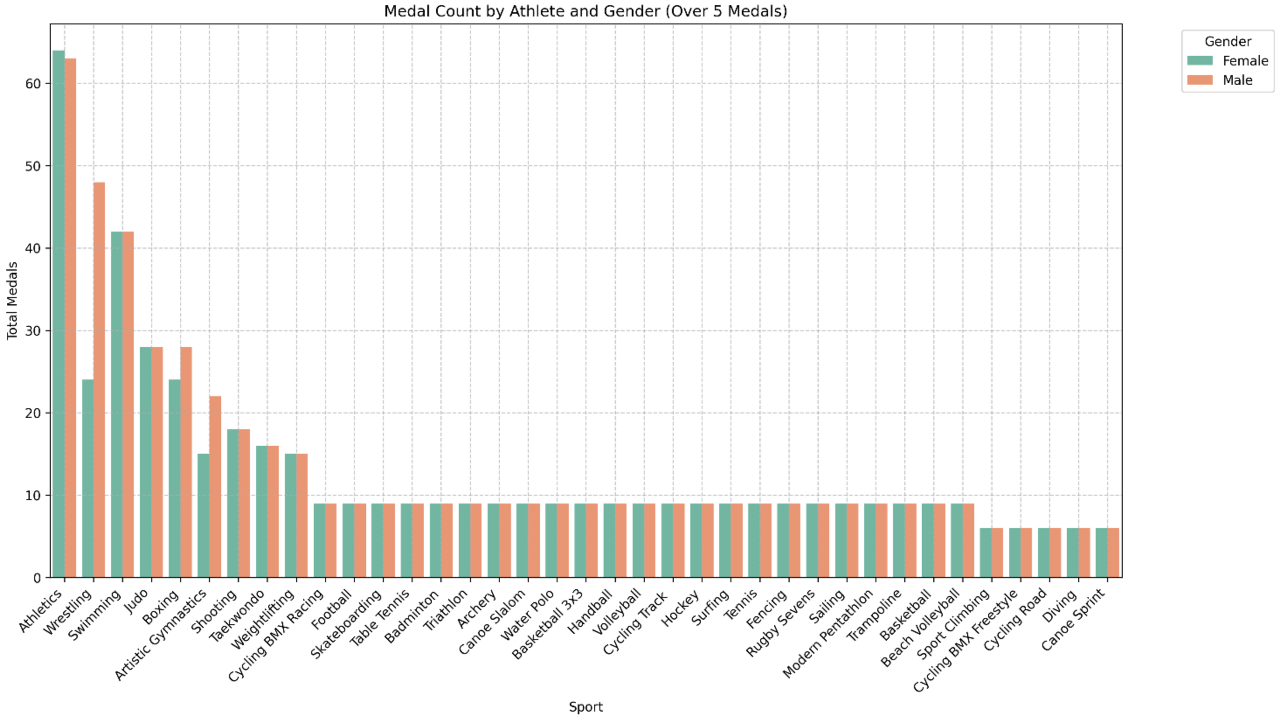
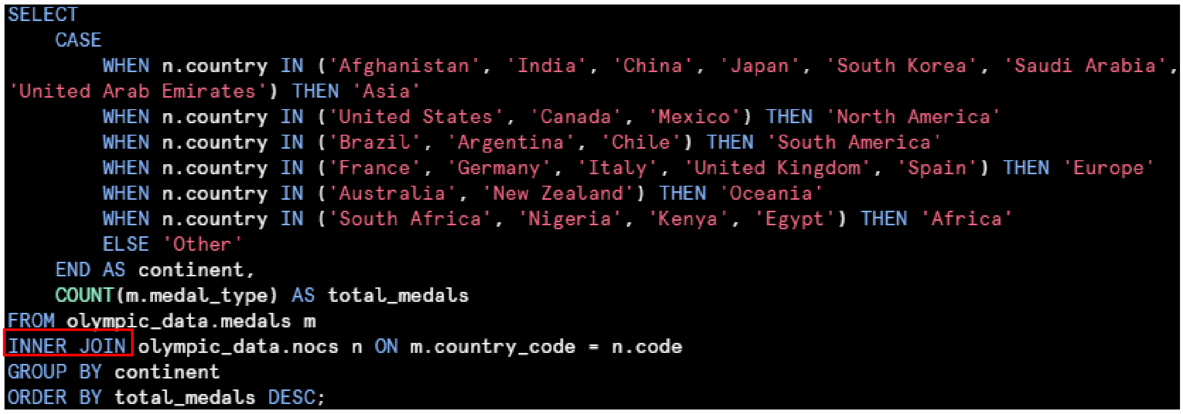


Figure 4 – Visual Representation of the Query

III. Medal Distribution by Content

- This query shows the distribution of medals by continent.

  It uses a CASE statement to categorize countries into continents based on their names. The query joins medals with nocs (National Olympic Committees) to link medals to countries. It then groups the results by continent and counts the medals, ordering the results by total medals in descending order.



```
SELECT
    CASE
        WHEN n.country IN ('Afghanistan', 'India', 'China', 'Japan', 'South Korea', 'Saudi Arabia',
'United Arab Emirates') THEN 'Asia'
        WHEN n.country IN ('United States', 'Canada', 'Mexico') THEN 'North America'
        WHEN n.country IN ('Brazil', 'Argentina', 'Chile') THEN 'South America'
        WHEN n.country IN ('France', 'Germany', 'Italy', 'United Kingdom', 'Spain') THEN 'Europe'
        WHEN n.country IN ('Australia', 'New Zealand') THEN 'Oceania'
        WHEN n.country IN ('South Africa', 'Nigeria', 'Kenya', 'Egypt') THEN 'Africa'
        ELSE 'Other'
    END AS continent,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
GROUP BY continent
ORDER BY total_medals DESC;
```

Figure 5 – Query

- The INNER JOIN between medals and nocs links medals to their respective countries. This join is necessary to group medals by continent based on the country of origin.

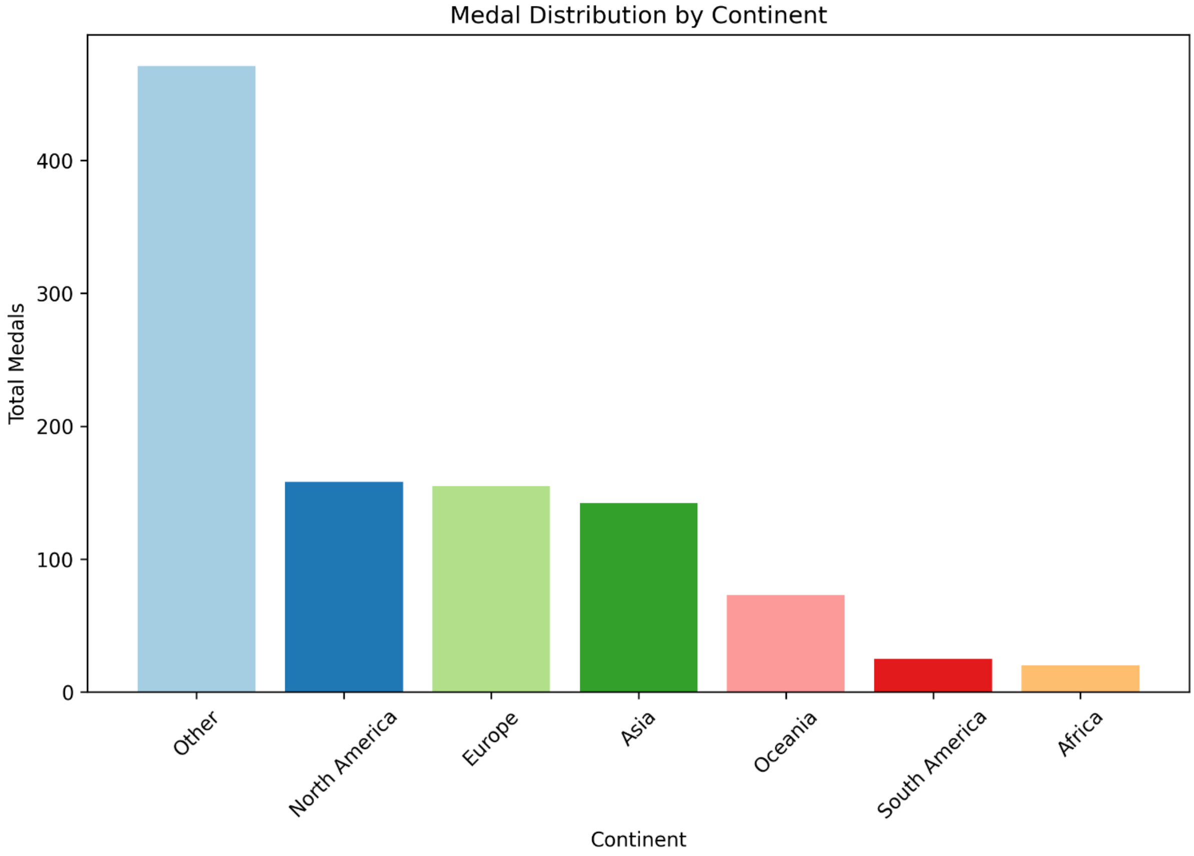
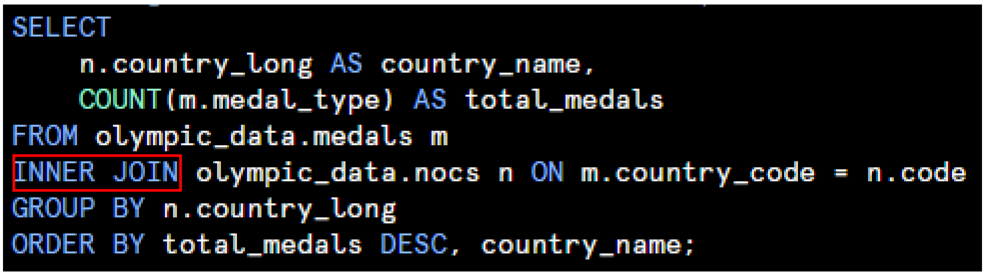


Figure 6 – Visual Representation of the Query

IV. Running Total of Medals for Each Country

- This query provides the total number of medals won by each country.

  It joins medals with nocs to associate medals with their respective countries. The results are grouped by country name and ordered first by the total number of medals and then by country name.



```sql
SELECT
    n.country_long AS country_name,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
GROUP BY n.country_long
ORDER BY total_medals DESC, country_name;
```

Figure 7 – Query

- The INNER JOIN between medals and nocs links the medals to the countries that won them. This join allows to group by country and calculate the total medal count for each (aggregation of the total number of medals per country).

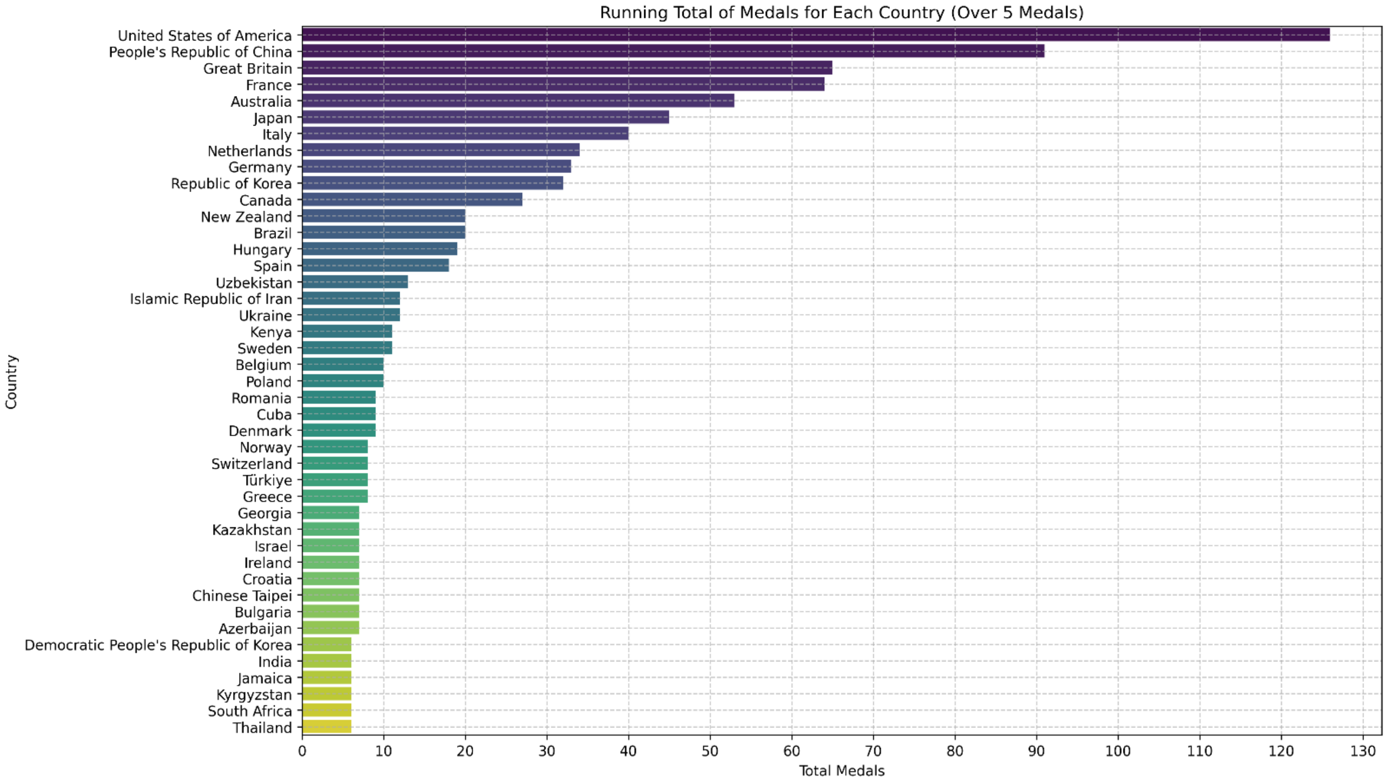
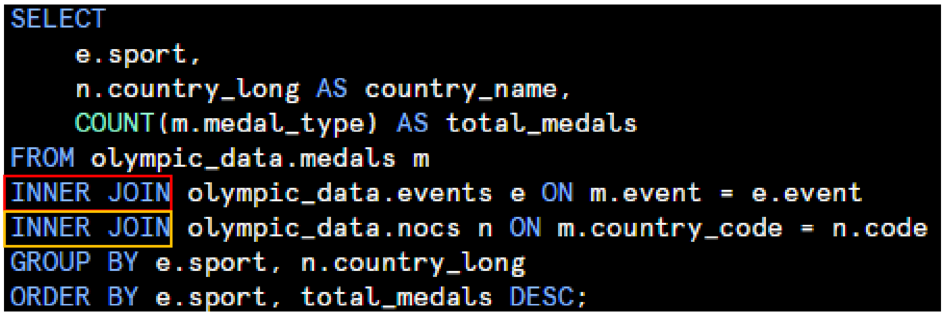


Figure 8 – Visual Representation of the Query

V. Top Performing Countries by Sport

- This query lists countries that perform best in each sport.

  It joins medals with events to get sport details, and with nocs to get country details. The results are grouped by sport and country, with medals counted and ordered by sport and the number of medals in descending order.



```
SELECT
    e.sport,
    n.country_long AS country_name,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.events e ON m.event = e.event
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
GROUP BY e.sport, n.country_long
ORDER BY e.sport, total_medals DESC;
```

Figure 9 – Query

- The first INNER JOIN between medals and events links medals to their sports, for grouping and counting medals by sport.
- The second INNER JOIN with nocs connects medals to the countries that won them. This join identifies which countries perform best in each sport.

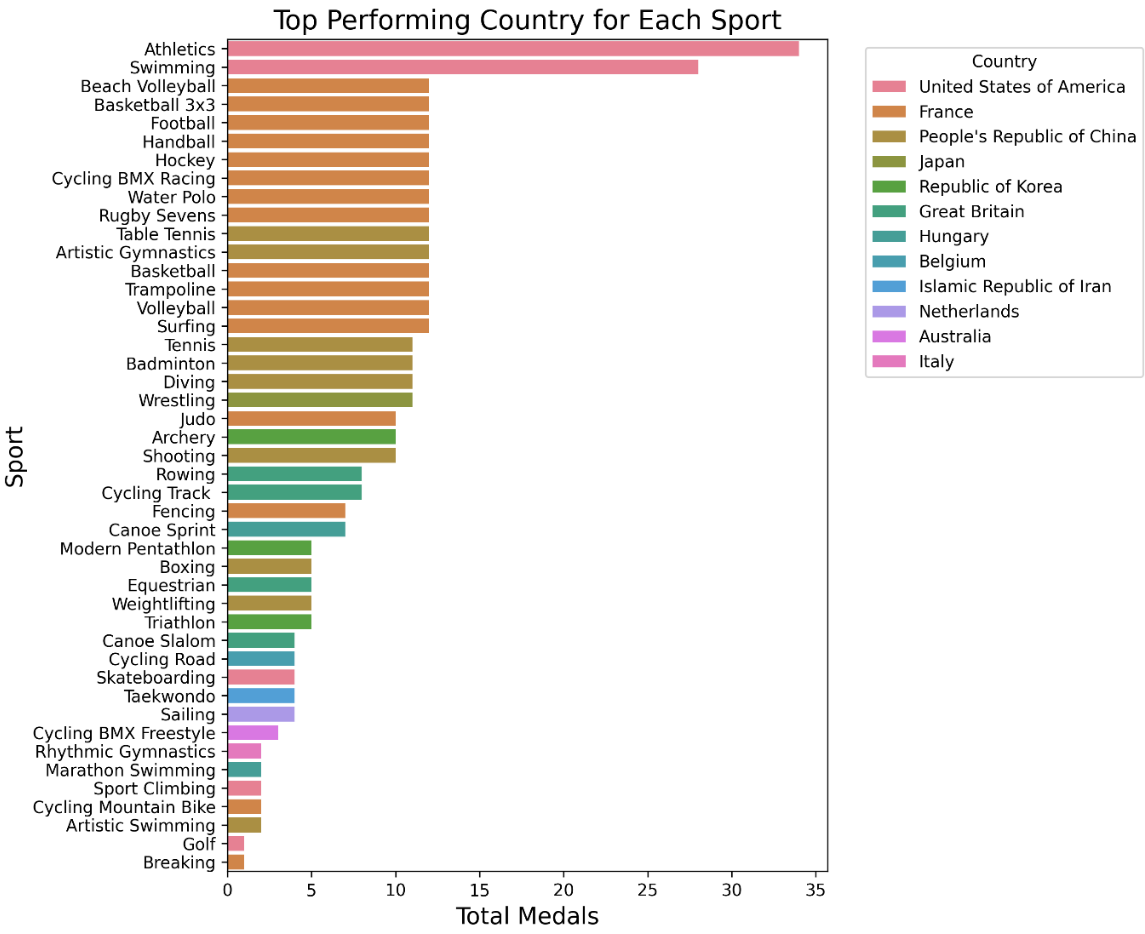


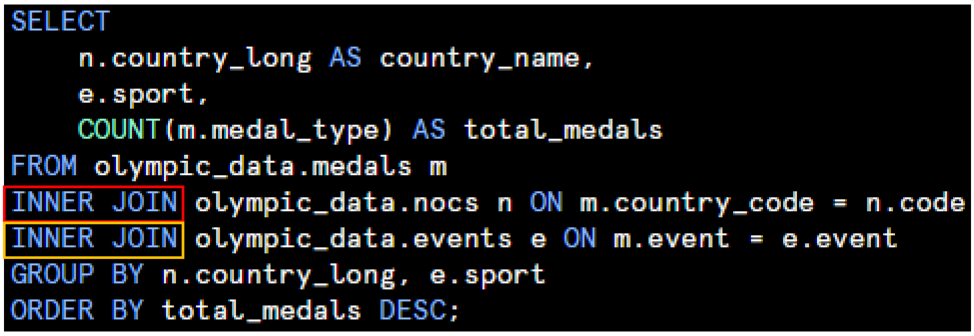
Figure 10 – Visual Representation of the Query

5.2 Country Performance Analysis

I. Performance by Country and Sport

- This query analyzes the total medals won by each country in different sports.

  It joins medals with nocs to link medals to countries and with events to get sport information. The results are grouped by country and sport, and the total number of medals is counted and ordered in descending order.



```
SELECT
    n.country_long AS country_name,
    e.sport,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
INNER JOIN olympic_data.events e ON m.event = e.event
GROUP BY n.country_long, e.sport
ORDER BY total_medals DESC;
```

Figure 11 – Query

- The INNER JOIN between medals and nocs links medals to the countries. This join is necessary to group medal counts by country.
- The second INNER JOIN, with events, associates medals with their sports, allowing the aggregation of medals by sport and country.

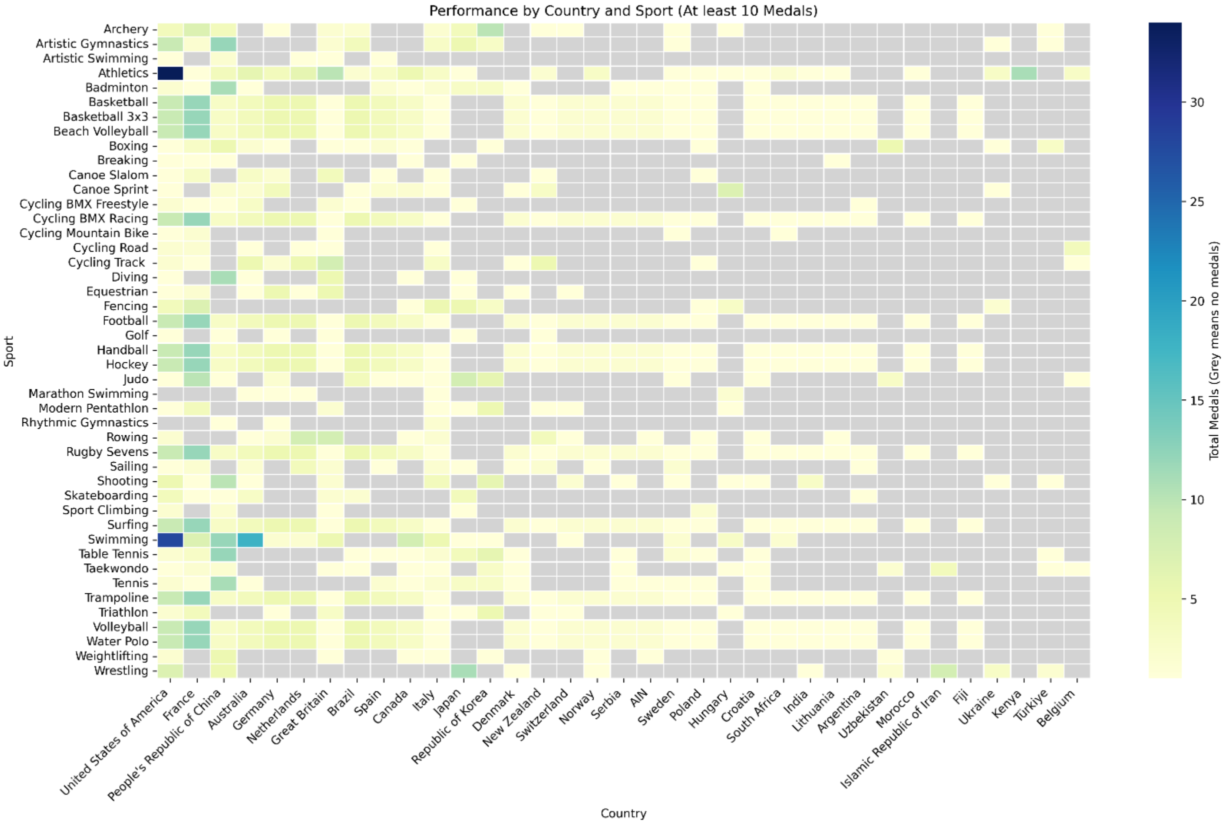
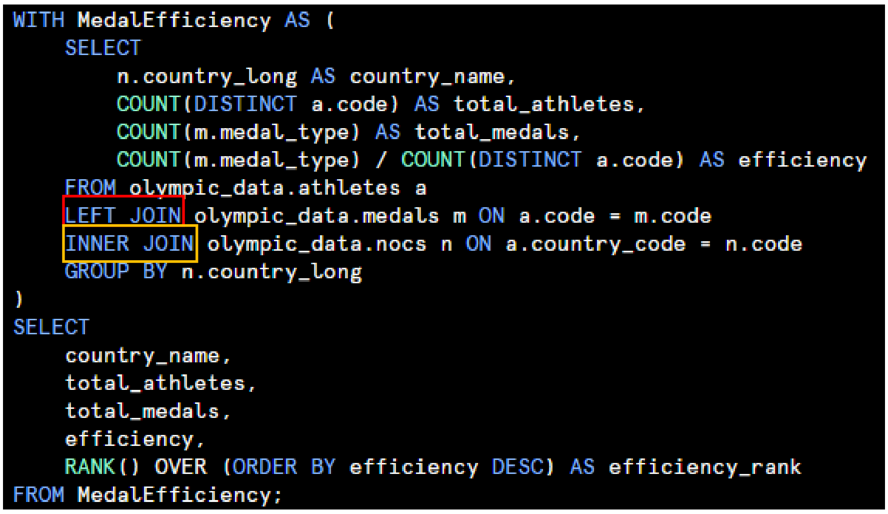


Figure 12 – Visual Representation of the Query

II. Medal Efficiency by Country Ranking Using CTE

- This query calculates and ranks countries by their medal efficiency.

  It first uses a Common Table Expression (CTE) to compute the efficiency of medal wins per athlete for each country. The main query then ranks countries based on this efficiency using the RANK() window function, ordering by efficiency in descending order.



```
WITH MedalEfficiency AS (
    SELECT
        n.country_long AS country_name,
        COUNT(DISTINCT a.code) AS total_athletes,
        COUNT(m.medal_type) AS total_medals,
        COUNT(m.medal_type) / COUNT(DISTINCT a.code) AS efficiency
    FROM olympic_data.athletes a
    LEFT JOIN olympic_data.medals m ON a.code = m.code
    INNER JOIN olympic_data.nocs n ON a.country_code = n.code
    GROUP BY n.country_long
)
SELECT
    country_name,
    total_athletes,
    total_medals,
    efficiency,
    RANK() OVER (ORDER BY efficiency DESC) AS efficiency_rank
FROM MedalEfficiency;
```

Figure 13 – Query

- The LEFT JOIN between athletes and medals ensures that the medal count includes all athletes, important for calculating accurate medal efficiency.
- The INNER JOIN with nocs links athletes to their countries, allowing grouping by country to compute the efficiency.

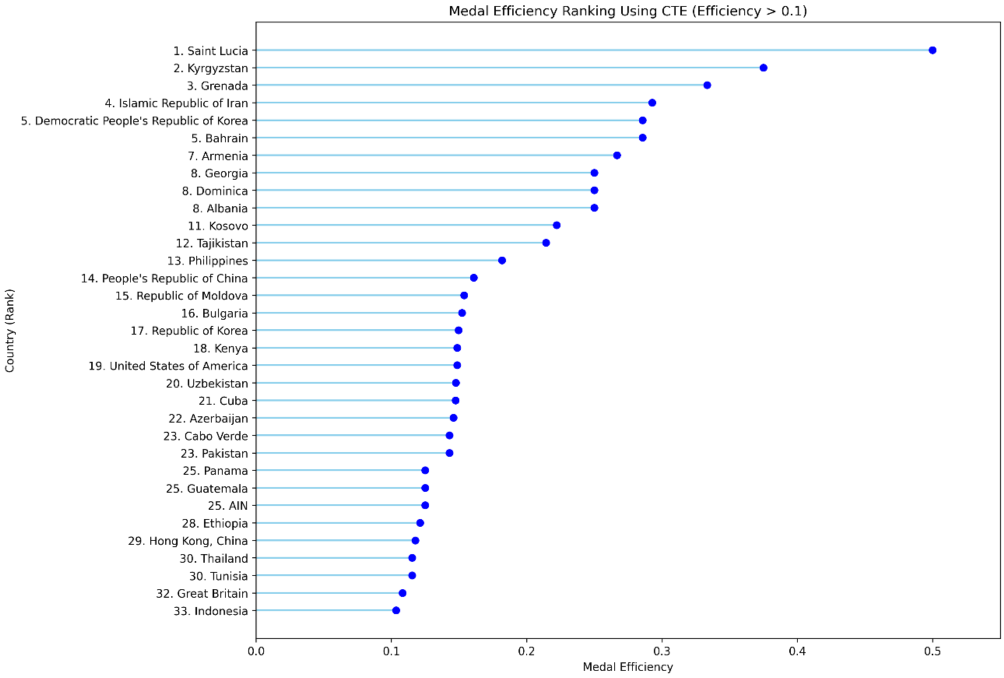


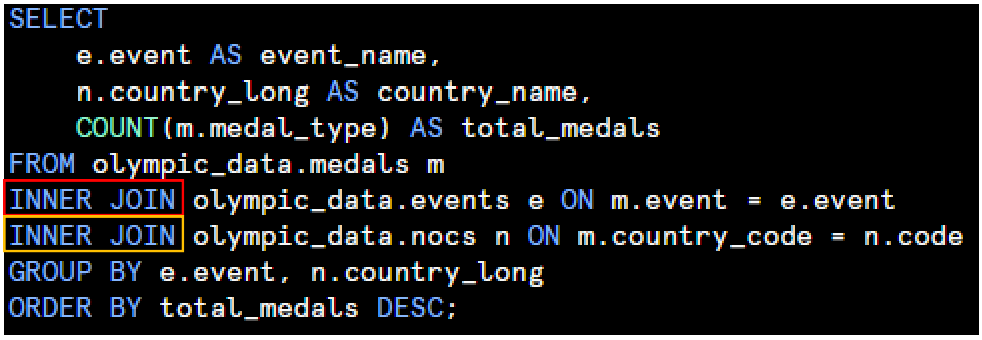
Figure 14 – Visual Representation of the Query

5.3 Event Performance Analysis

I. Most Successful Events

- This query identifies the most successful events based on total medals won by each country.

  It joins medals with events to get event names and with nocs to get country names. The results are grouped by event and country, with medals counted and ordered by total medals in descending order.



```
SELECT
    e.event AS event_name,
    n.country_long AS country_name,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.events e ON m.event = e.event
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
GROUP BY e.event, n.country_long
ORDER BY total_medals DESC;
```

Figure 15 – Query

- The INNER JOIN between medals and events links each medal to its event, allowing the query to aggregate the total number of medals by event.
- The INNER JOIN with nocs connects medals to their countries, allowing the grouping of medals by event and country.

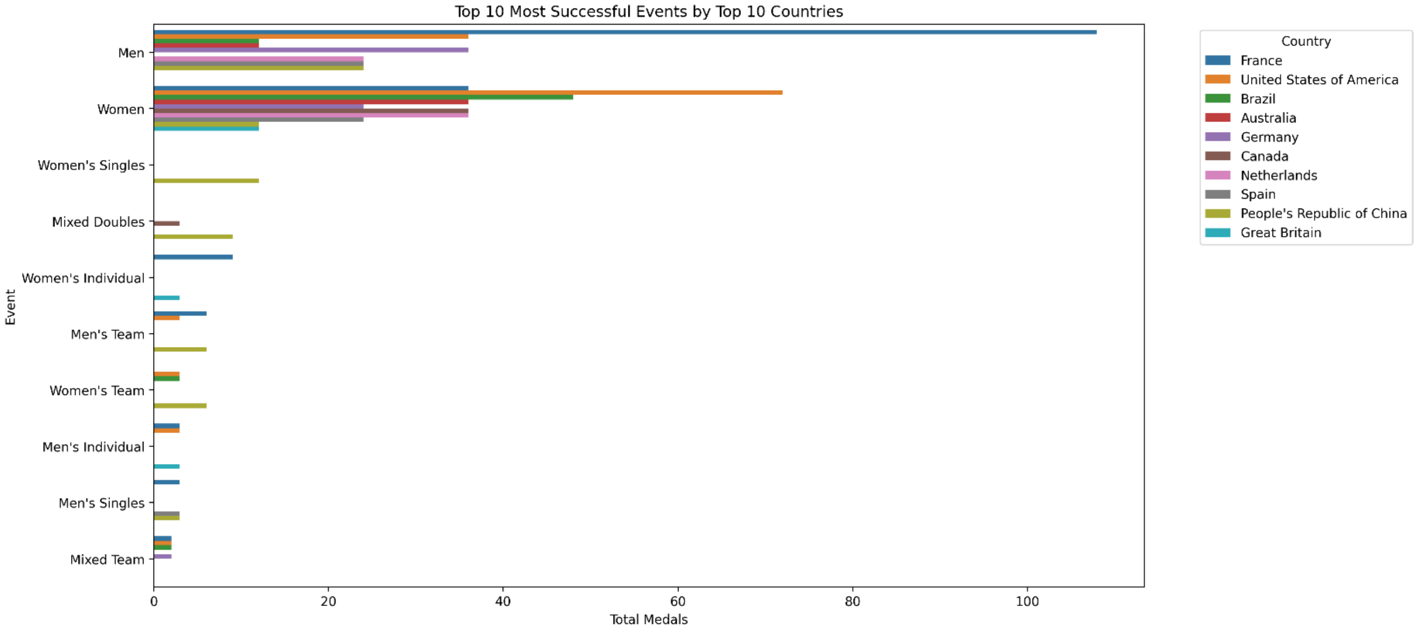
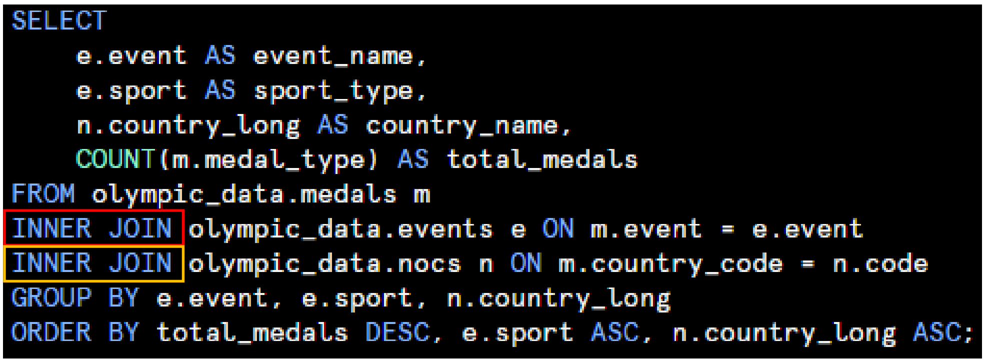


Figure 16 – Visual Representation of the Query

II. Most Successful Events & Sports

- This query provides the most successful events and sports, with medals won by each country.

  It joins medals with events to get both event and sport details, and with nocs for country information. The results are grouped by event, sport, and country, with medals counted and ordered by total medals in descending order. Finally there is a sorting by sport and country which ensures further breakdown.



```
SELECT
    e.event AS event_name,
    e.sport AS sport_type,
    n.country_long AS country_name,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.events e ON m.event = e.event
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
GROUP BY e.event, e.sport, n.country_long
ORDER BY total_medals DESC, e.sport ASC, n.country_long ASC;
```

Figure 17 – Query

- The INNER JOIN between medals and events links medals to events and sports, allowing the aggregation of medals by event and sport.
- The INNER JOIN with nocs connects medals to their countries, for a better grouping by event, sport, and country.

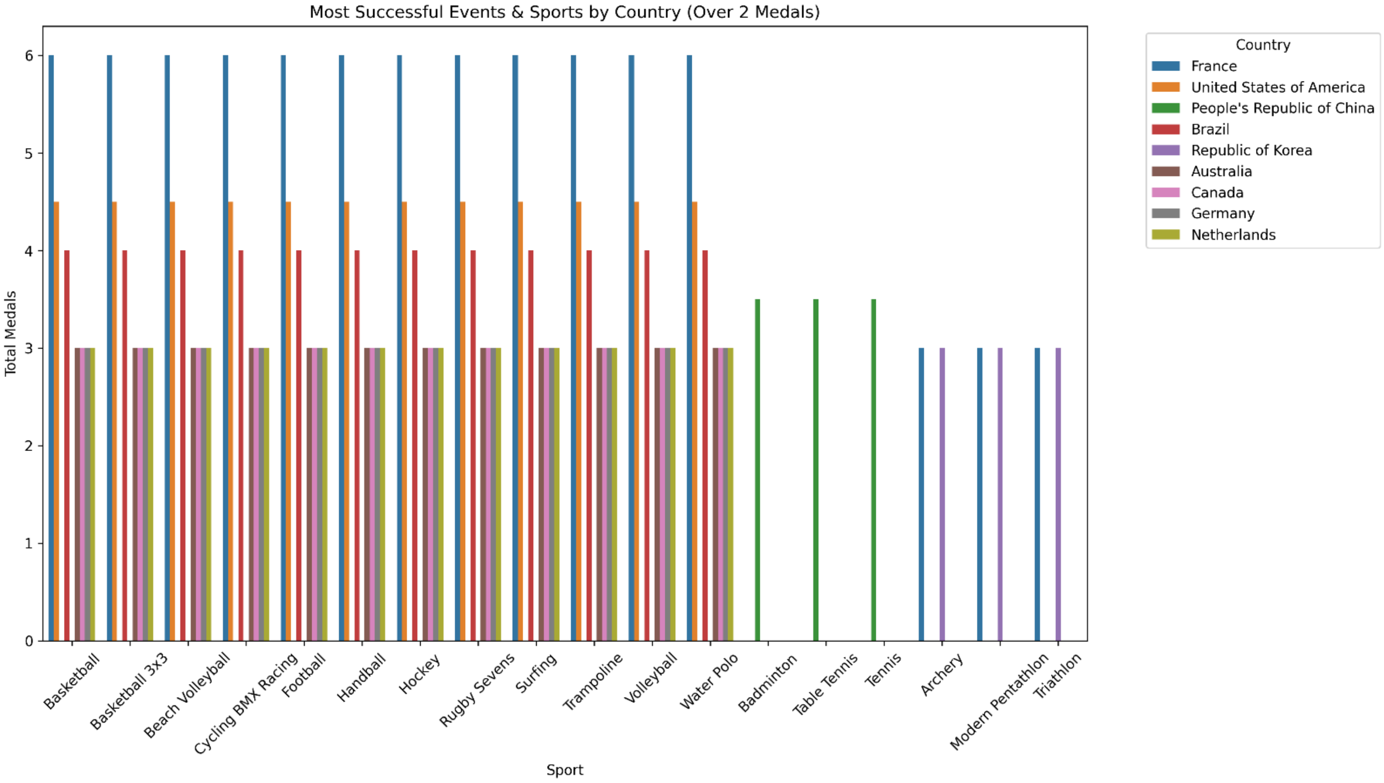
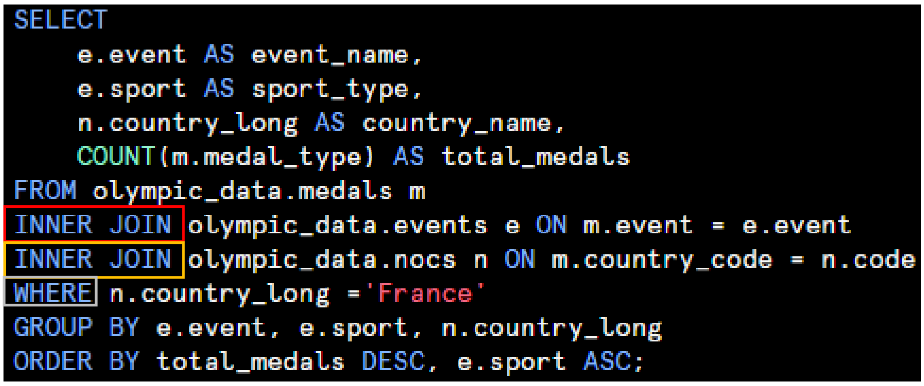


Figure 18 – Visual Representation of the Query

III. Most Successful Events & Sports (France)

- This query focuses on the performance of France, the host country, in different events and sports.

  It is similar to the previous query, but with a WHERE clause to filter results only for France. It groups by event and sport, counts medals, and orders the results by total medals in descending order, with an additional sorting by sport.



```
SELECT
    e.event AS event_name,
    e.sport AS sport_type,
    n.country_long AS country_name,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.events e ON m.event = e.event
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
WHERE n.country_long ='France'
GROUP BY e.event, e.sport, n.country_long
ORDER BY total_medals DESC, e.sport ASC;
```

Figure 19 – Query

- The WHERE clause filters results for France. The joins,like in the previous query, are important for associating medals with events, sports, and countries.

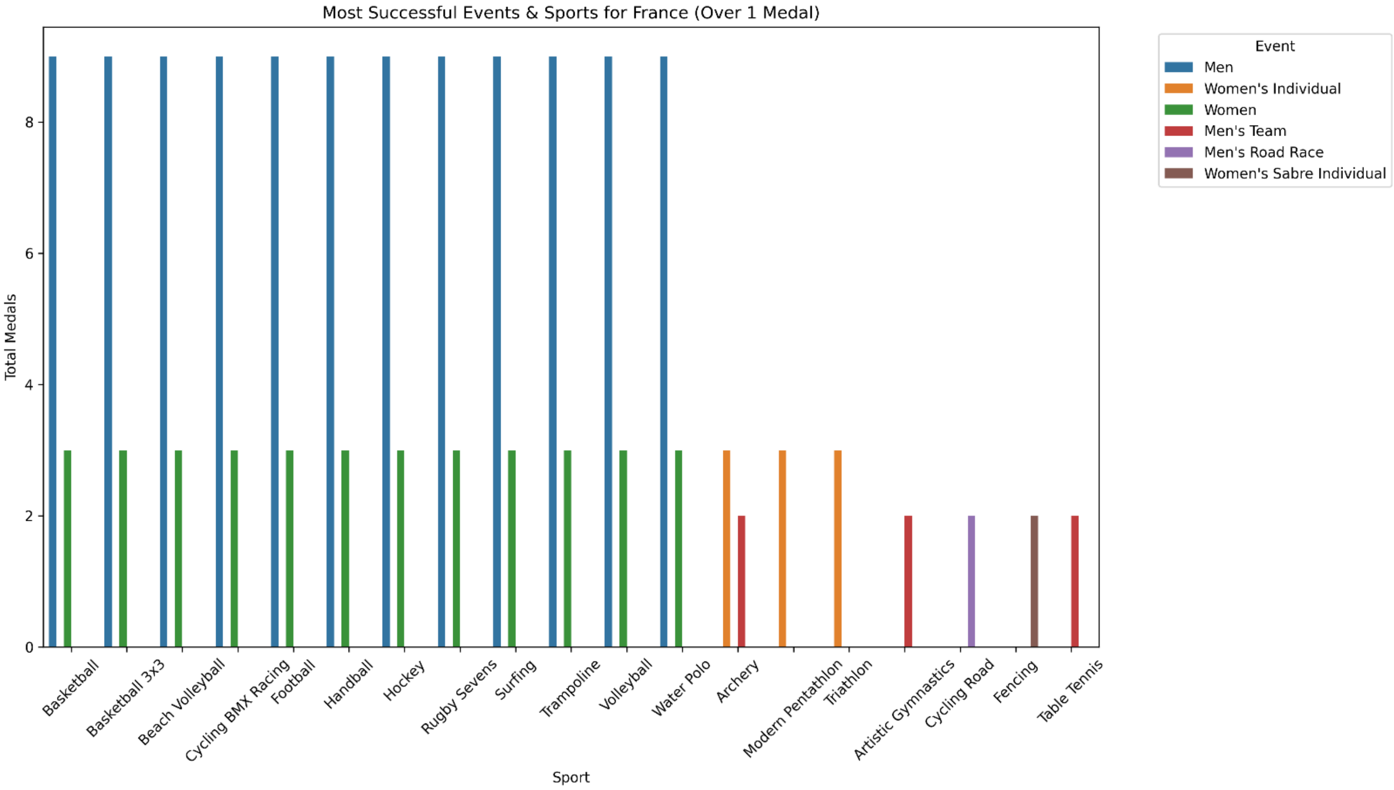
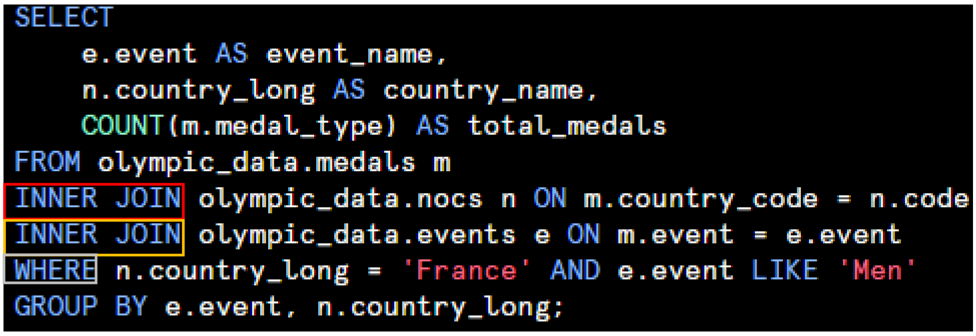


Figure 20 – Visual Representation of the Query

IV. Total Medals of a Country in an Event (France in Men's Events)

- This query retrieves the total number of medals won by France in men’s events.

  It joins medals with nocs for country details, and events to filter by event type. It includes a WHERE clause to focus on France and men’s events, then groups by event and country, counting medals.



```
SELECT
    e.event AS event_name,
    n.country_long AS country_name,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.medals m
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
INNER JOIN olympic_data.events e ON m.event = e.event
WHERE n.country_long = 'France' AND e.event LIKE 'Men'
GROUP BY e.event, n.country_long;
```

Figure 21 – Query

- The INNER JOIN between medals and nocs connects medals to the countries. This is important for focusing on France and aggregating medal counts.
- The INNER JOIN with events links medals to specific events, allowing filtering and grouping based on event type (men’s, in this case).

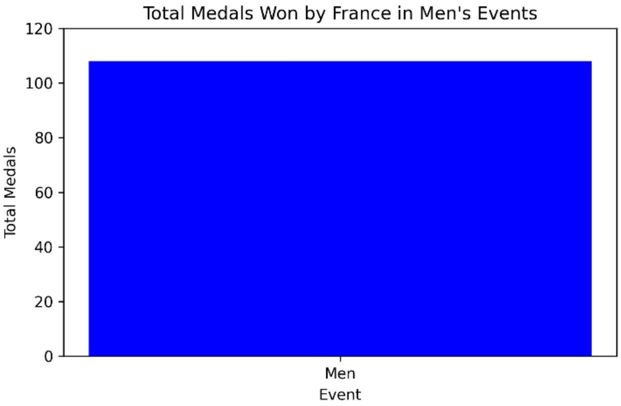


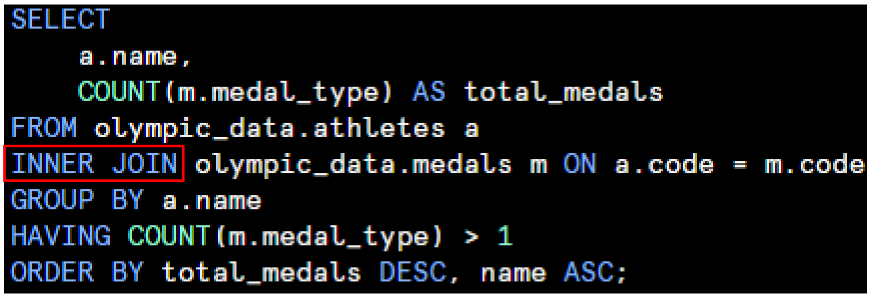
Figure 22 – Visual Representation of the Query

5.4 Athlete Performance Analysis

I. **Multi-Medal Athletes**

- This query identifies athletes who have won more than one medal.

  It joins athletes with medals to count medals per athlete. The HAVING clause is put to include only those with more than one medal, and results are first ordered by total medals in descending order, then by athlete name.



```
SELECT
    a.name,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.athletes a
INNER JOIN olympic_data.medals m ON a.code = m.code
GROUP BY a.name
HAVING COUNT(m.medal_type) > 1
ORDER BY total_medals DESC, name ASC;
```

Figure 23 – Query

- The INNER JOIN between athletes and medals is used to associate athletes with their medals. This join is necessary to count the number of medals each athlete has won and filter out those with more than one medal.

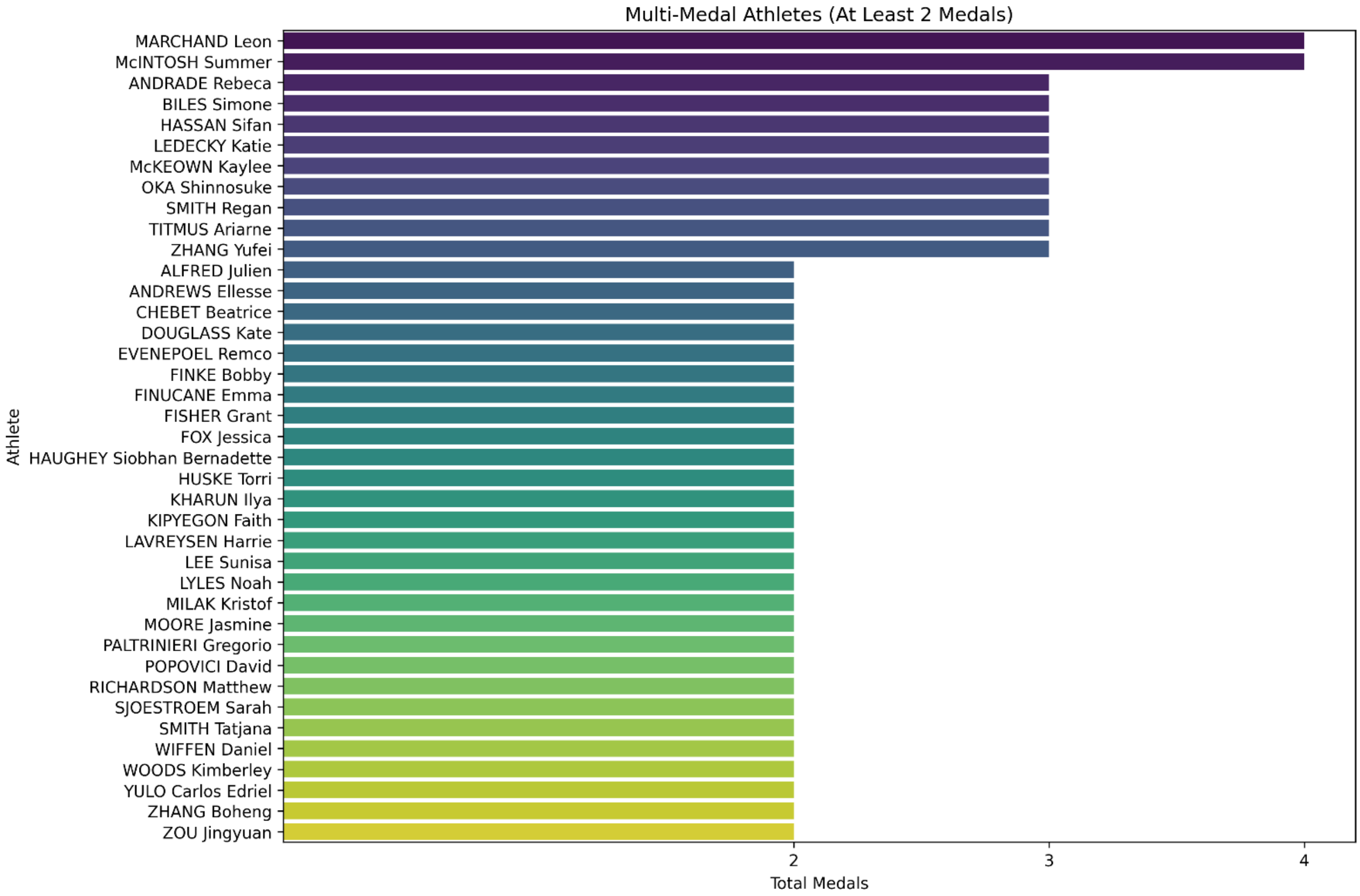
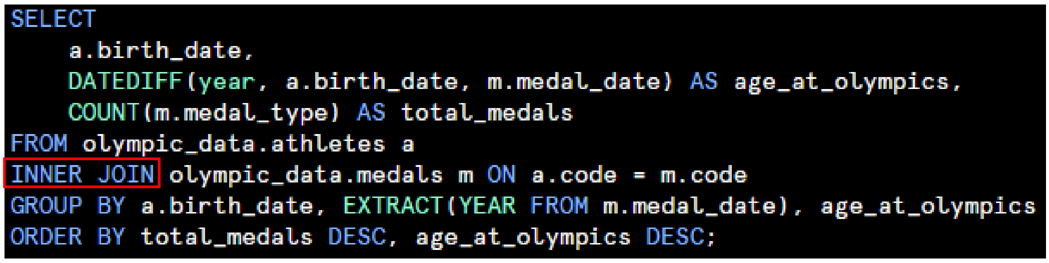


Figure 24 – Visual Representation of the Query

II. Age of Medal Winners

- This query calculates the age of athletes at the time of winning medals.

  It joins athletes with medals and calculates the age at the time of the medal using DATEDIFF. Results are grouped by birth date and medal year, with the total number of medals counted and ordered by the number of medals and age.



```sql
SELECT
    a.birth_date,
    DATEDIFF(year, a.birth_date, m.medal_date) AS age_at_olympics,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.athletes a
INNER JOIN olympic_data.medals m ON a.code = m.code
GROUP BY a.birth_date, EXTRACT(YEAR FROM m.medal_date), age_at_olympics
ORDER BY total_medals DESC, age_at_olympics DESC;
```

Figure 25 – Query

- The INNER JOIN between athletes and medals links each athlete to their medals, allowing the calculation of their age at the time of winning medals.

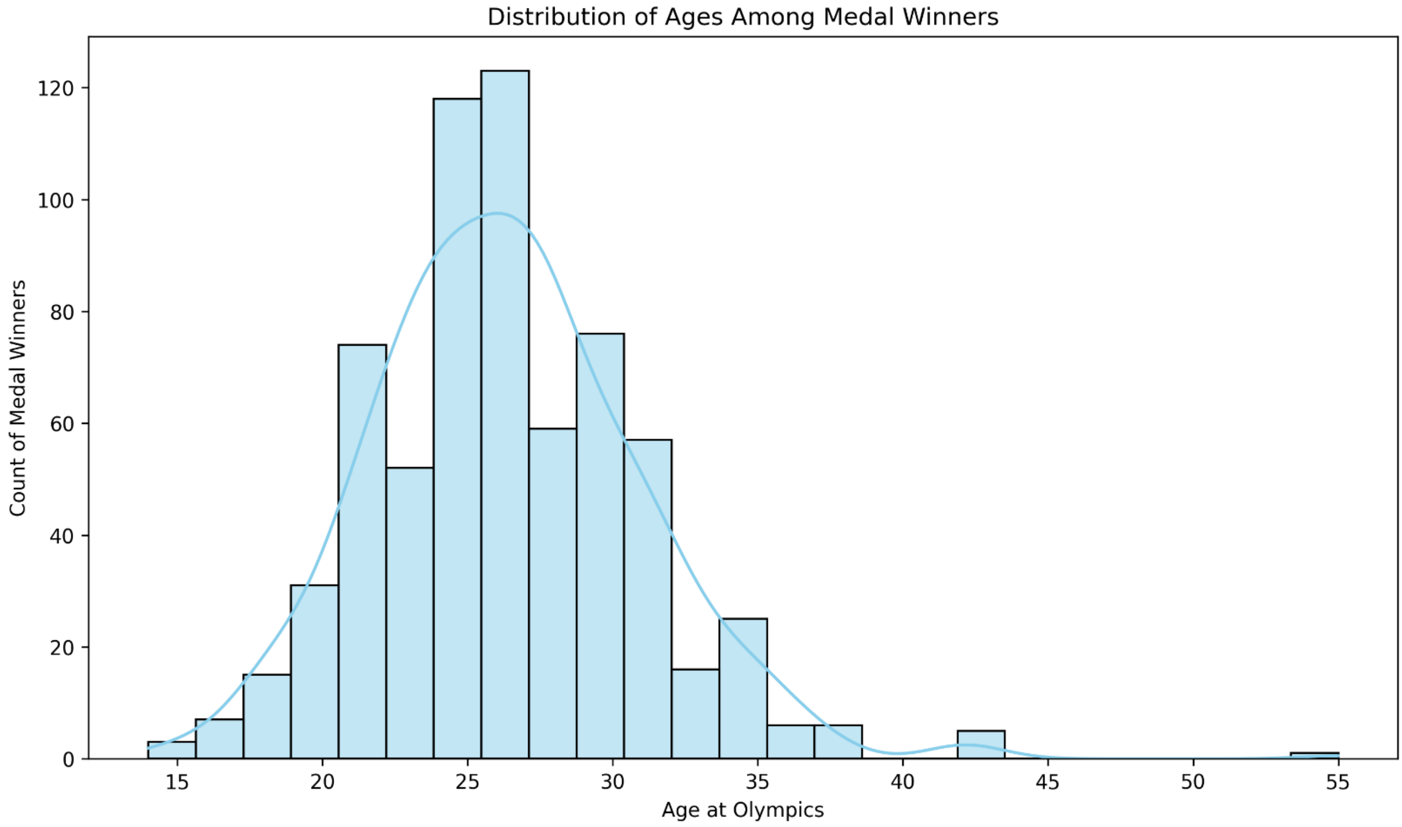


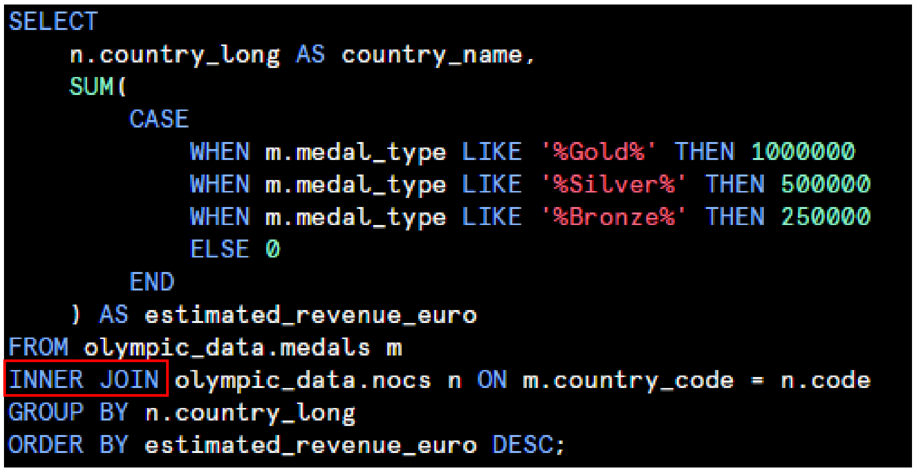
Figure 26 – Visual Representation of the Query

5.5 Business-Oriented Analysis

I. **Medal Revenue Estimation for Sponsors**

- This query estimates the revenue potential for sponsors based on the type of medals won.

  It joins medals with nocs to associate medals with countries. A CASE statement assigns revenue values to each type of medal, and the total estimated revenue is calculated and ordered by the estimated amount in descending order.



```
SELECT
    n.country_long AS country_name,
    SUM(
        CASE
            WHEN m.medal_type LIKE '%Gold%' THEN 1000000
            WHEN m.medal_type LIKE '%Silver%' THEN 500000
            WHEN m.medal_type LIKE '%Bronze%' THEN 250000
            ELSE 0
        END
    ) AS estimated_revenue_euro
FROM olympic_data.medals m
INNER JOIN olympic_data.nocs n ON m.country_code = n.code
GROUP BY n.country_long
ORDER BY estimated_revenue_euro DESC;
```

Figure 27 – Query

- The INNER JOIN between medals and nocs links medals to their countries, allowing for revenue estimation based on the number and type of medals won by each country.

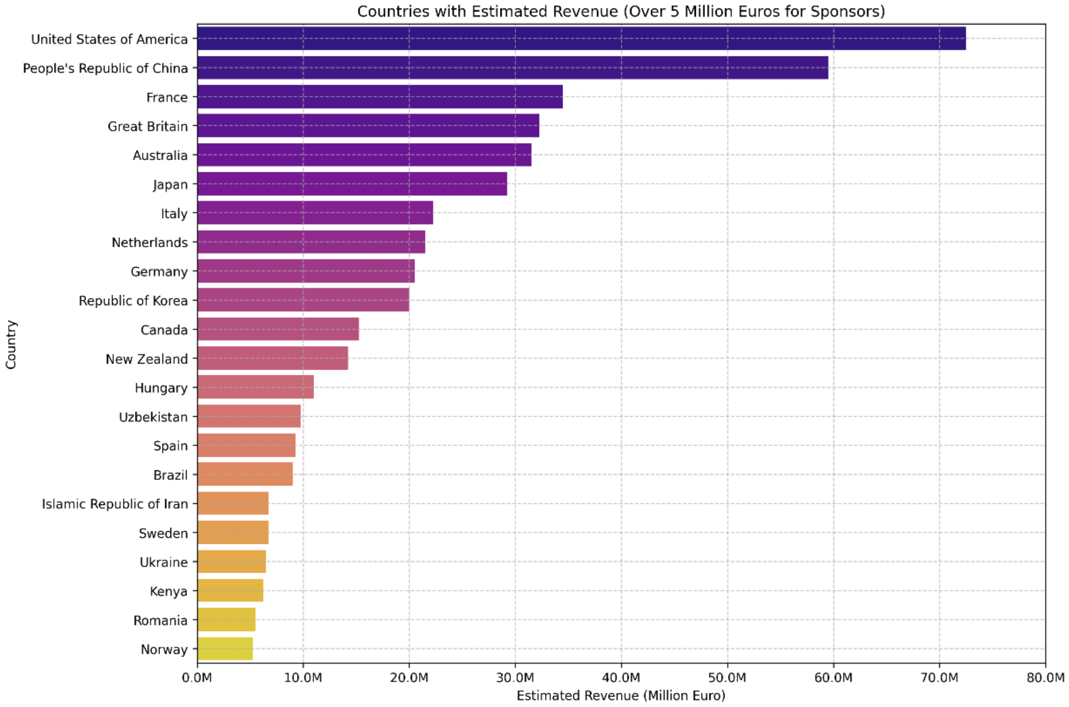
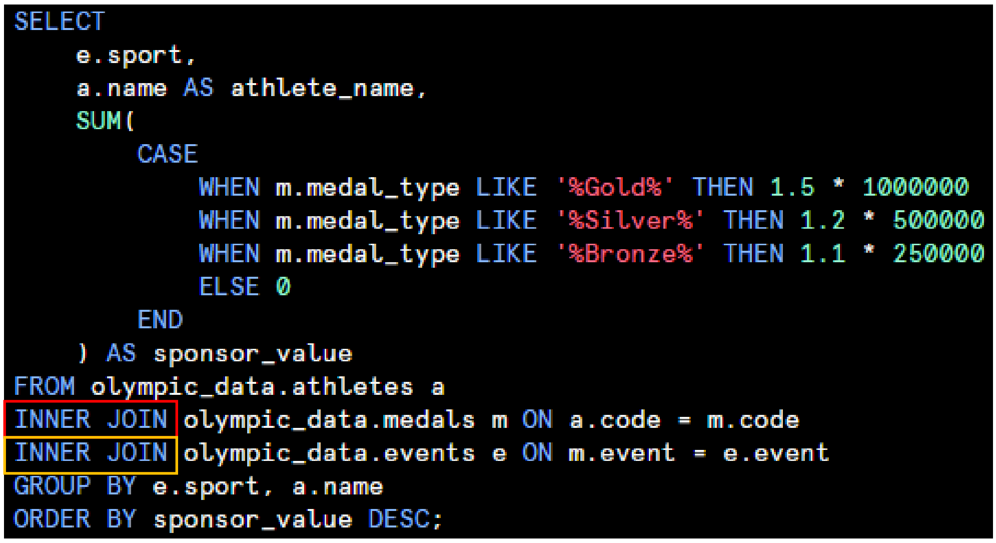


Figure 28 – Visual Representation of the Query

II. Medal Revenue Estimation by Sport and Athlete

- This query estimates the sponsor value of medals won by athletes, considering the sport and the type of medal.

  It joins athletes with medals and events to get sport details. A CASE statement calculates the sponsor value for each medal type, and the results are grouped by sport and athlete, ordered by sponsor value in descending order.



```
SELECT
    e.sport,
    a.name AS athlete_name,
    SUM(
        CASE
            WHEN m.medal_type LIKE '%Gold%' THEN 1.5 * 1000000
            WHEN m.medal_type LIKE '%Silver%' THEN 1.2 * 500000
            WHEN m.medal_type LIKE '%Bronze%' THEN 1.1 * 250000
            ELSE 0
        END
    ) AS sponsor_value
FROM olympic_data.athletes a
INNER JOIN olympic_data.medals m ON a.code = m.code
INNER JOIN olympic_data.events e ON m.event = e.event
GROUP BY e.sport, a.name
ORDER BY sponsor_value DESC;
```

Figure 29 – Query

- The INNER JOIN between athletes and medals is necessary to link athletes with their medals, for the subsequent calculation of the sponsor value.

- The INNER JOIN with events connects medals to their sports, allowing the revenue estimation based on both the sport and the athlete.

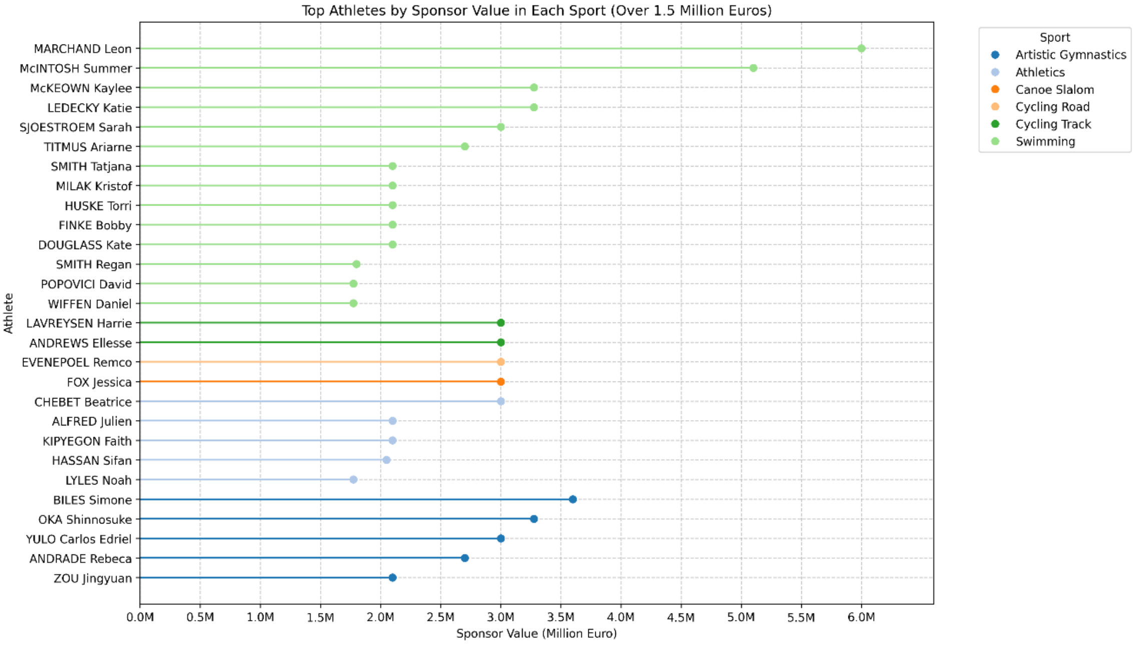
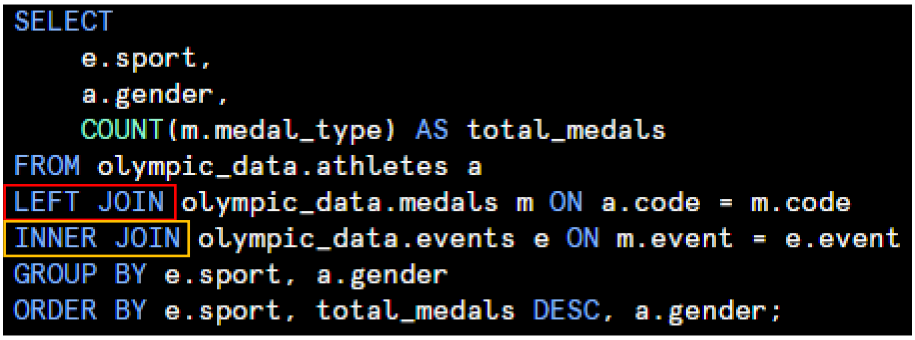


Figure 30 – Visual Representation of the Query

III. Gender Disparity in Medal Distribution

- This query examines the disparity in medal distribution between genders across different sports.

  It performs a LEFT JOIN between athletes and medals to include all athletes, even those without medals, and an INNER JOIN with events to get sport information. The results are grouped by sport and gender, and ordered by sport and total medals in descending order.



```
SELECT
    e.sport,
    a.gender,
    COUNT(m.medal_type) AS total_medals
FROM olympic_data.athletes a
LEFT JOIN olympic_data.medals m ON a.code = m.code
INNER JOIN olympic_data.events e ON m.event = e.event
GROUP BY e.sport, a.gender
ORDER BY e.sport, total_medals DESC, a.gender;
```

Figure 31 – Query

- The LEFT JOIN between athletes and medals ensures the inclusion of all athletes, even those without medals.
- The INNER JOIN with events links medals to their sports, allowing for grouping and counting medals by sport and gender.

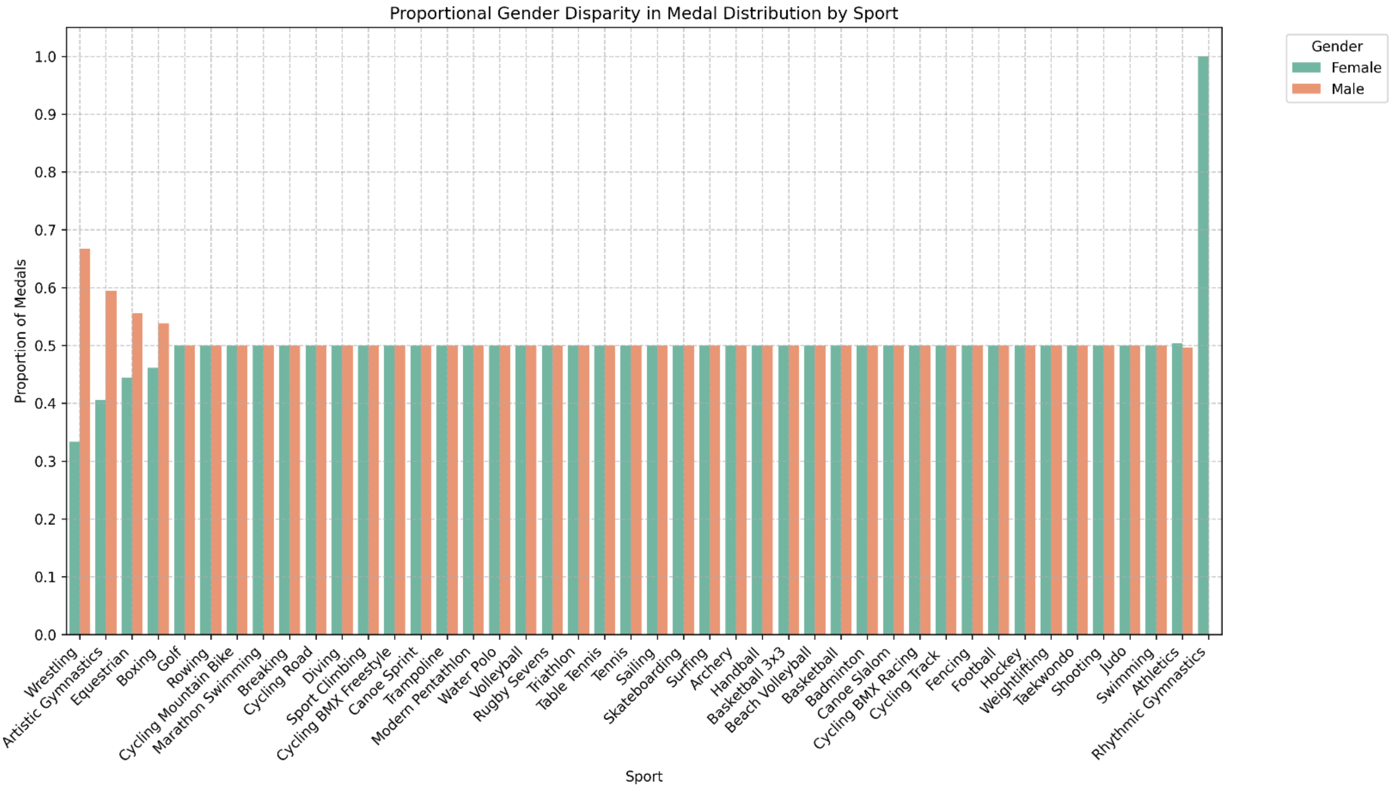


Figure 32 – Visual Representation of the Query

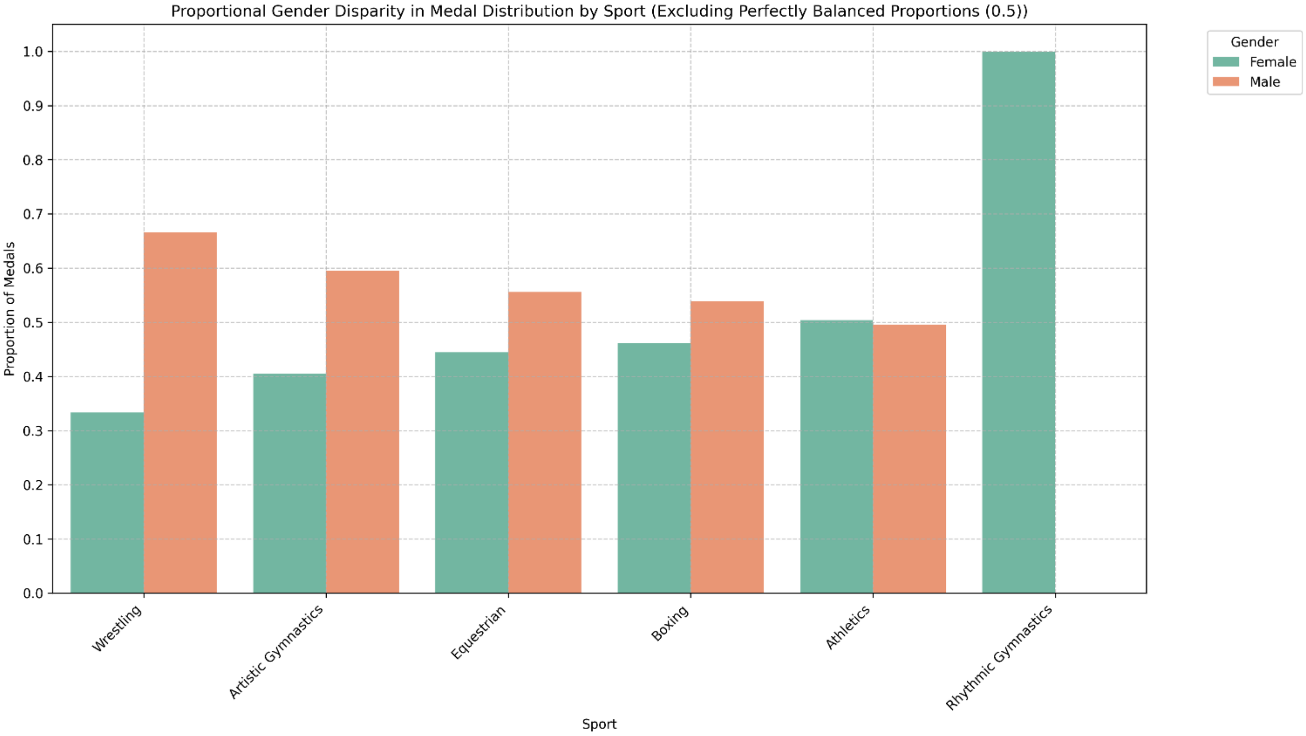


Figure 33 – Visual Representation of the Query: Zoom in on the non-equal distributions

6. Business Case Applications

This case study provides insights that can be applied in several business contexts:

I. Gender Equality in Sports

Organizations focused on promoting gender equality can utilize these analyses to:

- Assess and address gender disparities in participation and performance across different sports and countries.
- Develop initiatives to promote greater gender equality, ensuring fair representation and success in future sporting events.

II. Strategic Decision-Making and Sponsorship ROI

National Olympic committees, sports organizations, and sponsors can use these analyses to:

- Allocate resources more effectively by focusing on high-potential sports, athletes, or countries that are likely to yield higher returns regarding medal wins and associated revenues.
- Optimize athlete participation strategies and sponsorship investments based on performance data and trends.
- Estimate the potential return on investment from sponsoring specific athletes or teams, and develop marketing strategies that highlight their strengths, particularly in medal-winning sports.

III. Talent Scouting, Athlete Development

National sports organizations and sports analysts can use these analyses to:

- Identify countries or athletes with high participation rates but low medal counts, pinpointing areas for improvement and potential growth.
- Enhance training programs and talent development initiatives to improve performance in future competitions.

IV. Event and Venue Optimization

Event organizers and planners can use these analyses to:

- Assess which events and venues generate the most successful outcomes, helping to plan future competitions more effectively.
- Optimize the scheduling, marketing, and logistical planning of events to maximize success, athlete performance, and profitability.

This case study has shown important insights into Olympic performance, highlighting the dominance of particular countries in certain sports and the effectiveness of their athletes. This analysis provides practical strategies for national teams to improve training and resource allocation.

In conclusion, this study establishes a solid starting point for a data-driven structure that can support a sponsor's decision-making process on sponsorship, sustainable mobility, and brand positioning.