



---

Name: **Fenil Mansara, Om Chalodiya** ID: **202001268, 202001276**

Group: **05** Section: **06**

Date: **November 18, 2022** Project Team ID: **6.16**

---

# **Index**

- 1. Section1: Final version of SRS**
- 2. Section2: Noun Analysis**
- 3. Section3: ER-Diagrams all versions**
- 4. Section4: Mapping of ER-Model to Relational Model**
- 5. Section5: Normalization and Schema Refinement**
- 6. Section6: Final DDL Scripts**
- 7. Section7: 20 SQL Queries**
- 8. Section8: Project Code with output screenshots**

# Section1: Final version of SRS

## Olympics Information

### [1]. Description/Case Study of the Problem Domain

#### 1.1 Purpose

The purpose of the Olympics Database is to provide accurate, transparent, and well-organized information about Olympic sporting events from a variety of angles, including details about an individual athlete's or a country's Olympic accomplishments over time, as well as information about a specific sport or event and its champions and medalists.

On top of that, it holds a growing number of results from Olympic history beyond the top three medalists in each event. The Olympic games database enables us to present accurate medal standings in relation to the number of medals won by each nation, the all-time medal standings in a specific sport or event, the details of the opening ceremony, host venues, chief guests, newly added sports, discontinued sports, etc.

#### 1.2 Intended Audience and Reading Suggestions

The whole audience for the Olympics will be global, diverse, and in simple terms, humongous(enormous).

The Olympics database targets:

- **Athletes:** To review their past performance statistics and make strategies accordingly for upcoming events and competitions.
- **Sports Enthusiasts:** These include audiences of all ages who are enthusiastic about one or more sports.
- **Media Partners:** Media covers broadcast partners, newspapers, magazines, and social media like Twitter, Facebook, etc.
- **Sports Analysts:** These include trainers and coaches who analyze the performance of the athletes and guide them accordingly to improve their performance.

- **Database system developers:** They design, develop, and maintain the overall database system and also look after the technical difficulties faced by the users.
- **Database system managers:** They modify (insert, delete, and update) the data stored in the database system.
- **Sponsors:** These include brands and companies which sponsor the Olympic events and competitions.
- **International Olympic Committee (IOC):** Responsible for selecting the host city, overseeing the planning of the Olympic Games, updating and approving the Olympic sports event, and negotiating sponsorship and broadcasting rights.
- **International Federations (IFs):** These include governing bodies that supervise a sport at an international level.

One can get more idea about the Olympics database model by visiting third-party or official apps/websites like [Link1](#), [Link2](#), etc.

### 1.3 Product Scope

- The scope of the Olympic database can be extended to spread awareness, educate and encourage the youth about the sports practiced without any kind of discrimination and in the Olympic spirit, which requires mutual understanding with a spirit of friendship, solidarity, and fair play.
- The scope of the Olympic database can further be extended to analyze the economic and social impacts on the cities and countries hosting the Olympics. Hosting the Olympics can result in both positive and negative impacts on the host cities and countries.
- Hosting the Olympics can increase the host country's exports and might create opportunities for cities to influence local corporations in ways that benefit the local non-profit sector and civil society. The Olympics can also have significant adverse effects on host communities, for example, the displacement of various disadvantaged and backward-class communities.
- Moreover, the scope of this database can be further expanded by incorporating additional information, like details of sports/events which are going to be introduced in the upcoming Olympics, details of

the venues which are going to host the upcoming Olympics sports/events, etc.

## 1.4 Description

### ❖ Functionalities:

- **Performance analysis:** Efficient analysis of the performance of athletes and nations in past Olympic sports/events.
- **Participation analysis:** Efficient analysis of the participation ratio of males and females in individual sports/events.
- **Sport/Event analysis:** Efficient analysis of the new sports/events being added and sports/events being discontinued.
- **Leaderboard analysis:** Efficient analysis of the leaderboards, which includes the number of medals won by each athlete and the number of medals won by each nation during each Olympic event.
- **Host analysis:** Efficient analysis of the social and economic impacts on the cities and countries hosting the Olympics.

### ❖ Requirements

#### Table Requirements:

The tables that we are planning to implement to provide the functionalities listed above are:

- **Olympics:** This relation keeps track of the information of the venues hosting the Olympics, like Olympics ID, Olympics name (city, year), country, etc.
- **Developer:** This relation stores information of the developers of the database system, like developer ID, developer name, etc.
- **Manager:** This relation stores information of the managers of the database system, like manager ID, manager name, etc.
- **Athlete:** This relation keeps track of the information of the athletes, like athlete ID, athlete name, gender, age, height, weight, country, sport, etc.

- **Country:** This relation stores information about the countries participating in the Olympic events, like country ID, country name, etc.
- **Sport:** This relation stores information about the various kinds of sports, like sport ID, sport name, etc.
- **Event:** This relation stores details about the various kinds of events, like event ID, event name, sport name, etc.
- **Event leaderboard:** This relation keeps track of the winners of the events, like event ID, event name, sport name, gold medalist, silver medalist, bronze medalist, etc.
- **Sport leaderboard:** This relation keeps track of the winners of the sports, like sport ID, sport name, country ranks, etc.
- **Country leaderboard:** This relation keeps track of the number of medals won by the countries, like country ID, country name, gold medals, silver medals, bronze medals, total medals, etc.

## **Function Requirements:**

The functions that we are planning to implement to provide the functionalities listed above are:

- **Insert**

- **Add\_olympics:** This function adds the details of the Olympics into the database system.
- **Add\_developer:** This function adds the details of the developer into the database system.
- **Add\_manager:** This function adds the details of the manager into the database system.
- **Add\_athlete:** This function adds the details of the athlete into the database system.
- **Add\_event:** This function adds the details of the event into the database system.
- **Add\_sport:** This function adds the details of the sport into the database system.
- **Add\_country:** This function adds the details of the country into the database system.

- **Delete**
  - **Drop\_developer:** This function removes the details of the developer from the database system.
  - **Drop\_manager:** This function removes the details of the manager from the database system.
  - **Drop\_athlete:** This function removes the details of the athlete from the database system.
  - **Drop\_event:** This function removes the details of the event from the database system.
  - **Drop\_sport:** This function removes the details of the sport from the database system.
- **Update**
  - **Update\_developer:** This function updates the details of the developer.
  - **Update\_manager:** This function updates the details of the manager.
  - **Update\_athlete:** This function updates the details of the athlete.
  - **Update\_event:** This function updates the details of the event.
  - **Update\_sport:** This function updates the details of the sport.
  - **Update\_event\_leaderboard:** This function updates the details of the event leaderboard.
  - **Update\_sport\_leaderboard:** This function updates the details of the sport leaderboard.
  - **Update\_country\_leaderboard:** This function updates the details of the country leaderboard.
- **Search/View**
  - **View\_athlete:** This function gives the details of the athlete.
  - **View\_event:** This function gives the details of the event.
  - **View\_sport:** This function gives the details of the sport.
  - **View\_event\_leaderboard:** This function gives the details of the event leaderboard.
  - **View\_sport\_leaderboard:** This function gives the details of the sport leaderboard.
  - **View\_country\_leaderboard:** This function gives the details of the country leaderboard.

## [2]. Document the Requirements Collection / Fact-Finding Phase

### 2.1 Reading and Description

By going through the following references and performing background reading on them, we took a brief understanding of how the Olympics data models look like and what functionalities they provide to the different classes of users.

- 1) **Olympian Database-Facts about the Olympics - choose your perspective**

**Reference:** <https://www.olympiandatabase.com/>

- 2) **OLYMPIC RESULTS-Your source for event results and medalists from every Olympic Games.**

**Reference:** <https://olympics.com/en/olympic-games/olympic-results>

From the above two websites, we examined an already existing Olympics database management system. We referred to the services and functionalities that the database system should offer and maintain. We also inferred the functions and roles that we need to design and maintain and also examined the overall relationship between them.

In simpler terms, we referred to the general structure of the database. From these websites, we understood the overall requirements of the Olympics database management system and the points to be considered for the smooth functioning of the system and the satisfaction of the users by focusing on their needs.

- 3) **Database System Concepts, by Abraham Silberschatz, Henry F. Korth, S. Sudarshan**

**Reference:** <https://db-book.com/>

From this article, we got familiar with the various DBMS concepts like E-R model, E-R diagrams, Relational model, Mapping of E-R model to Relational model, Database designing and Database development, etc.

- **Combined Requirements gathered from the Background readings:**
  - A well-functioning Database management system is required to maintain and update all the information about athletes, sports, events, leaderboards, etc.
  - A user interface is required so that the users can easily access the data whenever required, without having to know much about the actual implementation of the system.
  - System administrators (like IOC officials, developers, etc.) and the users should be assigned different roles in order to ensure the integrity and consistency of the database.
  - System structure and functions should be designed in such a way that it ensures the efficient performance of the system.
  - The interface should be clean and without any redundant data.

## 2.2 Interview(s)

### [1]. Interview plan and summary

**System: Olympics Database System**

**Interviewee: Somin Gandhi (Role Play)**

**Designation: Athlete**

**Interviewer:**

1. Om Chalodiya
2. Fenil Mansara

**Designation: Database Developer**

**Designation: Database Manager**

**Date: 25/09/2022**

**Time: 14:00**

**Duration: 45 mins**

**Place: Skype**

**Purpose of interview:**

Preliminary meeting to identify problems and requirements of the athletes regarding the Olympics database management system.

## **Agenda:**

- Requirements analysis of the athletes
- Initial ideas
- Follow-up actions

## **Summary of Results and Requirements:**

- Efficient, optimal, and reliable services.
- Past results of each Olympic sport/event.
- Details about the number of medals won by the athletes and the nation for which they are playing.
- Details about the sports, events, and country leaderboards.
- Details about the format of the sports and events of the upcoming Olympics.
- Efficient statistical analysis of the athlete's performance so that athletes can enhance their strategy towards the sport they play and prepare accordingly in order to perform better in the upcoming Olympic events.

## **[2]. Interview plan and summary**

**System: Olympics Database System**

**Interviewee: Vatsal Gajera (Role Play)**

**Designation: IOC official**

**Interviewer:**

1. Om Chalodiya
2. Fenil Mansara

**Designation: Database Developer**  
**Designation: Database Manager**

**Date: 26/09/2022**

**Time: 14:30**

**Duration: 45 mins**

**Place: Skype**

**Purpose of interview:**

Preliminary meeting to identify problems and requirements of the IOC officials regarding the Olympics database management system.

## **Agenda:**

- Requirements analysis of the IOC officials.
- Initial ideas
- Follow-up actions

## **Summary of Results and Requirements:**

- Efficient, optimal, and reliable services.
- Security, consistency, and integrity of the data stored in the database.
- Analysis of the social and economic impact on the cities or countries hosting the Olympics events.
- Details about the viewership so that they can negotiate for better sponsorship and broadcasting rights.
- Efficient statistical analysis of the performance of each nation in order to get an idea about how favorable it could be for the nation hosting the Olympics over other nations.

## **[3]. Interview plan and summary**

**System: Olympics Database System**

**Interviewee: Shubham Modi (Role Play)**

**Designation: Analyst (Coach/Trainer)**

**Interviewer:**

1. Om Chalodiya
2. Fenil Mansara

**Designation: Database Developer**  
**Designation: Database Manager**

**Date: 27/09/2022**

**Time: 14:00**

**Duration: 45 mins**

**Place: Skype**

**Purpose of interview:**

Preliminary meeting to identify problems and requirements of the analysts (coach/trainer) regarding the Olympics database management system.

## **Agenda:**

- Requirements analysis of the analysts.
- Initial ideas
- Follow-up actions

## **Summary of Results and Requirements:**

- Efficient, optimal, and reliable services.
- Past results of each Olympic sport/event.
- Details about the sports, events, and country leaderboards.
- Details about the number of medals won by the athletes in a particular sport/event.
- Statistical analysis of the athletes' performance in order to guide them in improving their performance.

## **[4]. Interview plan and summary**

**System: Olympics Database System**

**Interviewee: Tirth Vadaria (Role Play)**

**Designation: Audience**

**Interviewer:**

3. Om Chalodiya
4. Fenil Mansara

**Designation: Database Developer**  
**Designation: Database Manager**

**Date: 28/09/2022**

**Time: 14:30**

**Duration: 45 mins**

**Place: Skype**

**Purpose of interview:**

Preliminary meeting to identify problems and requirements of the audience regarding the Olympics database management system.

## **Agenda:**

- Requirements analysis of the audience
- Initial ideas
- Follow-up actions

## **Summary of Results and Requirements:**

- Efficient, optimal, and reliable services.
- Past results of each Olympic sport/event.
- Details about the number of medals won by the athletes and the nations.
- Details about the sports, events, and country leaderboards.
- Details about the format of the sports and events of the upcoming Olympics.
- Details of the venues of the upcoming Olympics events.

## **● Combined Requirements gathered from the Interviews:**

- The database system should be designed in such a way that it provides efficient and reliable services.
- Structure and functionalities of the database system should be designed in such a way that it ensures the optimal performance of the system.
- System administrators and users will be assigned different roles in order to ensure the integrity of the information and consistency of the database system.
- The interface of the database system should be clean and without any redundant data.

## 2.3 Questionnaires

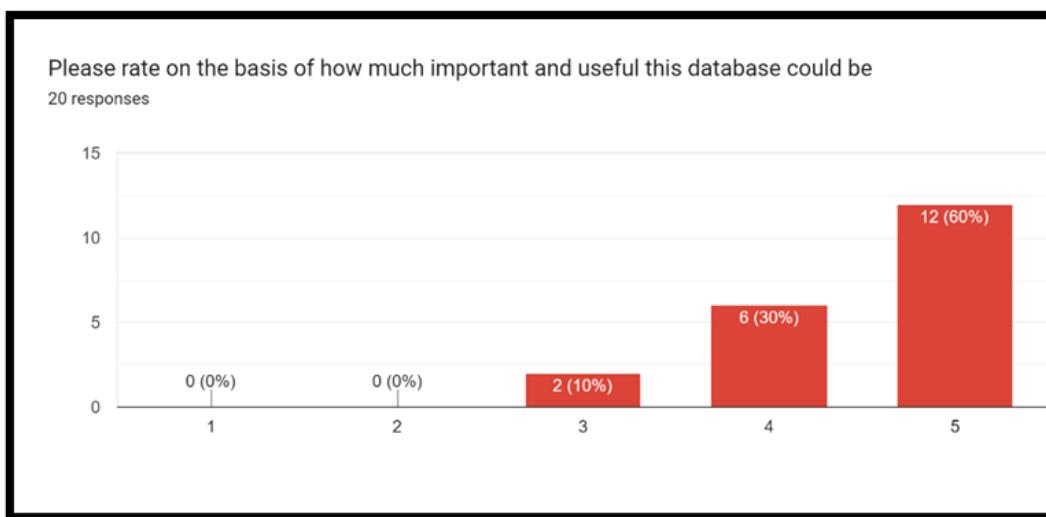
Questions were asked/surveyed via a Google form, and the summary of the responses is shown below:

### 1) Rating about the importance of the database

Please rate on the basis of how important and useful this database could be \*

1      2      3      4      5

Not Important                                    Extremely Important



- **Intent of the question:**

To get an idea about how valuable and functional this database system could be.

- **Observation from the response:**

The majority of responses indicated that such a database would be beneficial and valuable to all types of audiences, including viewers, athletes, analysts, media, IOC officials, etc.

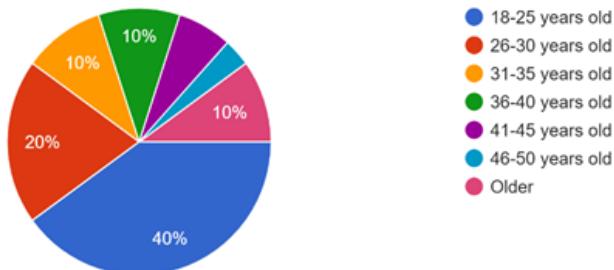
## 2) Age

Age \*

- 18-25 years old
- 26-30 years old
- 31-35 years old
- 36-40 years old
- 41-45 years old
- 46-50 years old
- Older

Age

30 responses

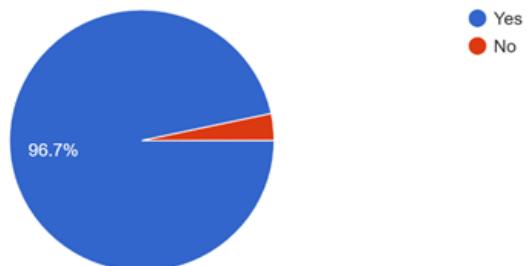


## 3) Interest in playing sports

Do you currently play or have you previously played any sports? \*

- Yes
- No

Do you currently play or have you previously played any sports?  
30 responses

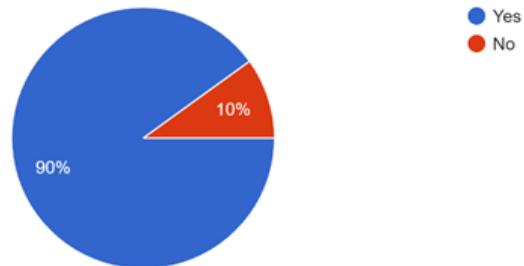


#### 4) Interest in watching the Olympics

Have you watched any Olympic games?\*

- Yes  
 No

Have you watched any Olympic games?  
30 responses



- **Intent of the above three questions:**

To get an idea about the age group of the users of the database and their interests in sports and the Olympics.

- **Observation from the response:**
  - The audience of the Olympics is diverse in nature and covers people from almost all age groups, mainly from the age group of 18-25 years.
  - It is observed that almost all users are interested in sports of one kind or another and are also interested in watching the Olympics.

## 5) Winning factors

Which factor do you think makes a country a better contender for winning medals? \*

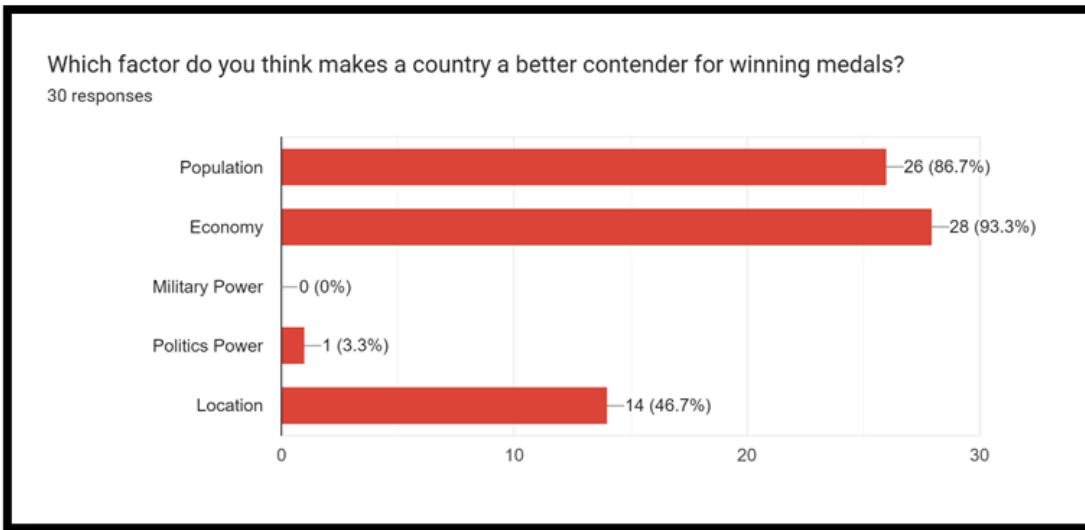
Population

Economy

Military Power

Politics Power

Location



- **Intent of the question:**

To get an idea about what the audience thinks about the factors that make a country a better contender for winning more medals.

- **Observation from the response:**

- From the responses, it is clear that Population and Economy play a vital role in making a country a better contender for winning medals. However, location too plays a significant role in improving the overall competitiveness of the nation.

## 6) Expectations from the database

What do you expect from the database? \*

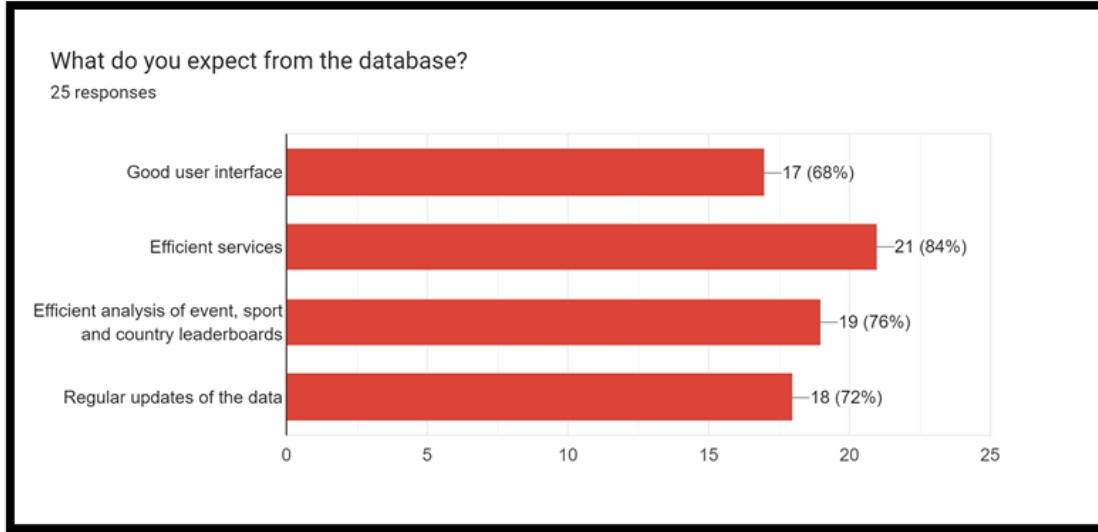
Good user interface

Efficient services

Efficient analysis of event, sport and country leaderboards

Regular updates of the data

Other...



- **Intent of the question:**

To get an idea about what the users expect from the database.

- **Observation from the response:**

- The majority of responses indicated that the database system should provide efficient services along with good UI.
- The database should provide an efficient analysis of the event, sport, and country leaderboards, along with regular updates.

## 7) Additional information that can be incorporated into the database

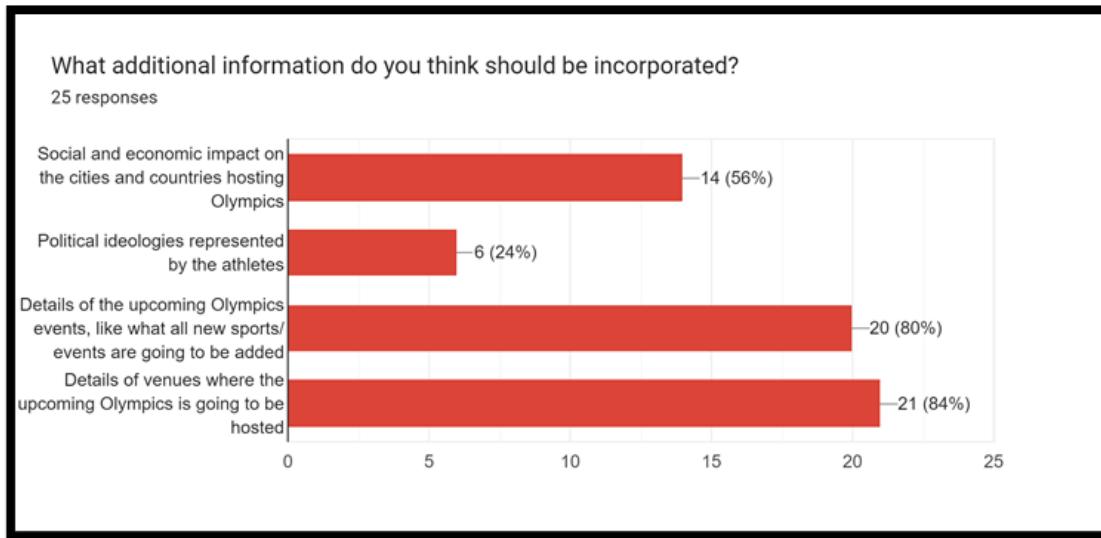
What additional information do you think should be incorporated? \*

Social and economic impact on the cities and countries hosting Olympics

Political ideologies represented by the athletes

Details of the upcoming Olympics events, like what all new sports/events are going to be added

Details of venues where the upcoming Olympics is going to be hosted



- **Intent of the question:**

To get an idea about what additional information should be incorporated into the database.

- **Observation from the response:**

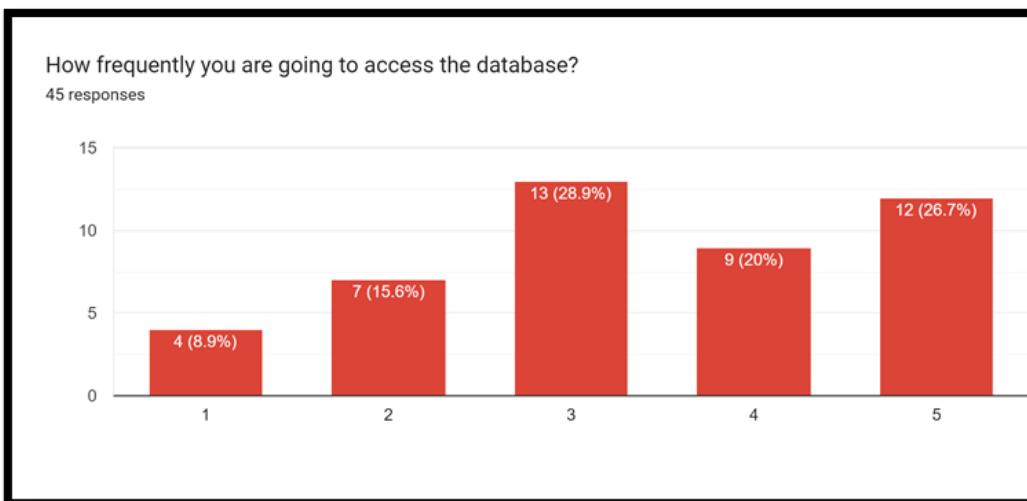
- From the perspective of the audience, many users want to extend the scope of this database by incorporating additional information, like details of the venues of the upcoming Olympic events, new sports/events that are going to be introduced in the upcoming Olympics, etc.
- From the perspective of the officials (like Athletes, Analysts, and IOC), many users want to extend the scope of this database by incorporating additional information, like details of the venues of the upcoming Olympic events, new sports/events that are going to be introduced in the upcoming Olympics, social and political impact on the cities and countries hosting the Olympics, etc.

## 8) Frequency of accessing the database system

How frequently you are going to access the database?\*

1      2      3      4      5

Occasionally                                    Frequently



- **Intent of the question:**

To get an idea about how frequently the users will be accessing the database.

- **Observation from the response:**

- From the perspective of the audience, the database would be accessed Occasionally.
- Whereas on the other hand, from the perspective of officials (like Athletes, Analysts, and IOC), the database would be accessed very frequently.

- **Combined Requirements gathered from the Questionnaire:**

- The database system to be developed should have a good UI and should provide efficient services.
- Regular updates of the data stored in the database system.
- User feedback on a regular basis.
- Incorporation of the additional requirements proposed by the users during the feedback.

## 2.4 Observations

**System:** Official Olympics website [Link](#)

**Observations by:** Om Chalodiya

**Date:** 29/9/2022

**Time:** 14:00

**Duration:** 45 minutes

### Observations and Combined Requirements:

- Efficient and optimal storage of the data in the system.
- Real-time update of all the information stored in the database system, maintaining consistency of the database.
- Efficient services related to the search, insert, delete, and update functionalities.
- Well-designed user interface that should withstand traffic up to a certain level and should provide a quality experience to the users.
- Thorough and efficient analysis of the details about the athlete, sport, event, leaderboard, etc.
- Concurrent access to the database system.
- Security and integrity of the information stored in the database system.
- Internet connection should be stable.

### [3]. Fact Finding Chart

| Objective  | Technique           | Subject  | Time commitment |
|--|---------------------|--|-----------------|
| To gain background knowledge about Olympic sports and events | Background readings | Few relatable official and third-party apps/websites | 1 day           |
| To gain an understanding of the roles of Athletes            | Interview           | Athlete  | 45 mins         |
| To gain an understanding of the roles of IOC officials       | Interview           | IOC official   | 45 mins         |
| To gain an understanding of the roles of Analysts            | Interview           | Coach/Trainer  | 45 mins         |
| To gain an understanding of the roles of audience            | Interview           | Audience   | 45 mins         |
| To gain an understanding of the user's perspective           | Questionnaire       | Users  | 1 day           |
| To gain an understanding of the real world Olympics database | Observation         | Olympics website                                     | 1 hour          |

## [4]. List Requirements

- Details about the date and venue of the Olympics. (**got at Background Reading, Questionnaire**)
- Information about the athletes. (**got at Background Reading, Interview**)
- Details of sports and their respective events. (**got at Background Reading, Interview, Questionnaire, Observation**)
- Sports, events, and country leaderboard details. (**got at Background Reading, Interview, Questionnaire, Observation**)
- Winner of each sport and event. (**got at Background Reading, Interview, Questionnaire, Observation**)
- The number of medals won by each individual athlete. (**got at Background Reading, Interview, Questionnaire, Observation**)
- The number of medals won by each nation in an individual sport. (**got at Background Reading, Interview, Questionnaire, Observation**)
- The software to be developed should have a reliable UI and should provide efficient and optimal services. (**got at Background Reading, Observation**)
- Regular user feedback and suggestions from the various user groups about the functionalities and aspects of the database system. (**got at Questionnaire**)
- System administrators and users should be assigned different roles in order to maintain the security, integrity, and consistency of the database. (**got at Background Reading, Observation**)
- The structure and functions of the system should be designed in such a way that it ensures the optimal performance of the system. (**got at Background Reading, Observation**)
- The interface should be clean and without any redundant data. (**got at Background Reading, Observation**)
- Real-time update of the information stored in the database system. (**got at Background Reading, Interview, Observation**)
- The integrity of data should be maintained in the different parts of the system in real-time. (**got at Background Reading, Observation**)
- Stable internet connectivity. (**got at Background Reading, Observation**)
- Backup of the database. (**got at Background Reading, Observation**)

## [5]. User Categories and Privileges

### List of user categories and their roles

#### 1) Audience

The role of the audience is just to view the statistics of the Olympic events like details of athletes, sports, events, leaderboards, etc. The audience is not granted any privileges related to the insert, delete, and update operations.

#### 2) Athletes and Analysts

The role of the athletes is to analyze their performance and prepare strategies accordingly in order to improve their game. The role of the analysts is to study or analyze the performance of the athletes and guide them accordingly in order to improve their performance for the upcoming Olympics.

#### 3) IOC officials

The role of the IOC officials is to analyze the socio-economic impacts on the cities and countries hosting Olympic events and to decide upon the venues, chief guests, opening ceremonies, media partners, broadcast rights, sponsors, etc.

#### 4) Database system developer

The role of the database system developer is to design, develop and maintain the overall database system, and to look after the cases of technical issues (like system crashes) faced by the users.

#### 5) Database system manager

The role of the database system manager is to modify (insert, delete, and update) the data stored in the database system, maintaining the consistency and integrity of the database system.

## List of privileges/functions that can be accessed by different user classes

### 1) Search

- Audience
- Athletes and Analysts
- IOC officials
- Developer
- Manager

### 2) Insert

- IOC officials
- Manager

### 3) Delete

- IOC officials
- Manager

### 4) Update

- IOC officials
- Manager

## [6]. Product Functions

We plan to implement the functions supported by PostgreSQL, like multi-querying, plotting data related to the properties of the table used, and real-time analysis of the number of people accessing the cloud-integrated DBMS service.

### 1) Insert

- **Add\_olympics:** This function adds the details of the Olympics into the database.
  - **Input:** venue (city, country), year, and other relevant details of the Olympics.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Insertion of the entered record upon the successful verification of the record by the system.

- **Add\_developer:** This function adds the details of the developer into the database.
  - **Input:** ID, name, and other relevant details of the developer.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Insertion of the entered record upon the successful verification of the record by the system.
- **Add\_manager:** This function adds the details of the manager into the database.
  - **Input:** ID, name, and other relevant details of the manager.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Insertion of the entered record upon the successful verification of the record by the system.
- **Add\_athlete:** This function adds the details of the athlete into the database.
  - **Input:** ID, name, and other relevant details of the athlete.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Insertion of the entered record upon the successful verification of the record by the system.
- **Add\_country:** This function adds the details of the country into the database.
  - **Input:** ID, name, and other relevant details of the country.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Insertion of the entered record upon the successful verification of the record by the system.
- **Add\_event:** This function adds the details of the event into the database.
  - **Input:** ID, name, and other relevant details of the event.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Insertion of the entered record upon the successful verification of the record by the system.
- **Add\_sport:** This function adds the details of the sport into the database.

- **Input:** ID, name, and other relevant details of the sport.
- **Processing:** Process and verify the value/record entered by the user.
- **Output:** Insertion of the entered record upon the successful verification of the record by the system.

## 2) Delete

- **Drop\_developer:** This function removes the details of the developer from the database.
  - **Input:** ID of the developer to be removed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Deletion of the entered record upon the successful verification of the record by the system.
- **Drop\_manager:** This function removes the details of the manager from the database.
  - **Input:** ID of the manager to be removed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Deletion of the entered record upon the successful verification of the record by the system.
- **Drop\_athlete:** This function removes the details of the athlete from the database.
  - **Input:** ID of the athlete to be removed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Deletion of the entered record upon the successful verification of the record by the system.
- **Drop\_event:** This function removes the details of the event from the database.
  - **Input:** ID of the event to be removed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Deletion of the entered record upon the successful verification of the record by the system.
- **Drop\_sport:** This function removes the details of the sport from the database.

- **Input:** ID of the sport to be removed.
- **Processing:** Process and verify the value/record entered by the user.
- **Output:** Deletion of the entered record upon the successful verification of the record by the system.

### 3) Update

- **Update\_developer:** This function updates the details of the developer.
  - **Input:** ID of the developer whose details need to be updated.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Updation of the record upon the successful verification of the record by the system.
- **Update\_manager:** This function updates the details of the manager.
  - **Input:** ID of the manager whose details need to be updated.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Updation of the record upon the successful verification of the record by the system.
- **Update\_athlete:** This function updates the details of the athlete.
  - **Input:** ID of the athlete whose details need to be updated.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Updation of the record upon the successful verification of the record by the system.
- **Update\_event:** This function updates the details of the event.
  - **Input:** ID of the event whose details need to be updated.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Updation of the record upon the successful verification of the record by the system.
- **Update\_sport:** This function updates the details of the sport.
  - **Input:** ID of the sport whose details need to be updated.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Updation of the record upon the successful verification of the record by the system.

- **Update\_event\_leaderboard:** This function updates the details of the event leaderboard.
  - **Input:** ID of the event whose leaderboard details need to be updated.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Updation of the event leaderboard upon the successful verification by the system.
- **Update\_sport\_leaderboard:** This function updates the details of the sport leaderboard.
  - **Input:** ID of the sport whose leaderboard details need to be updated.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Updation of the sport leaderboard upon the successful verification by the system.
- **Update\_country\_leaderboard:** This function updates the details of the country leaderboard.
  - **Input:** ID of the country whose leaderboard details need to be updated.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Updation of the country leaderboard upon the successful verification by the system.

#### **4) Search/View**

- **View\_athlete:** This function gives the details of the athlete.
  - **Input:** ID of the athlete whose details need to be viewed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Details of the athlete upon the successful verification by the system.
- **View\_sport:** This function gives the details of the sport.
  - **Input:** ID of the sport whose details need to be viewed.
  - **Processing:** Process and verify the value/record entered by the user.

- **Output:** Details of the sport upon the successful verification by the system.
- **View\_event:** This function gives the details of the event.
  - **Input:** ID of the event whose details need to be viewed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Details of the event upon the successful verification by the system.
- **View\_event\_leaderboard:** This function gives the details of the event leaderboard.
  - **Input:** ID of the event whose leaderboard details need to be viewed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Details of the event leaderboard upon the successful verification by the system.
- **View\_sport\_leaderboard:** This function gives the details of the sport leaderboard.
  - **Input:** ID of the sport whose leaderboard details need to be viewed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Details of the sport leaderboard upon the successful verification by the system.
- **View\_country\_leaderboard:** This function gives the details of the country leaderboard.
  - **Input:** ID of the country whose leaderboard details need to be viewed.
  - **Processing:** Process and verify the value/record entered by the user.
  - **Output:** Details of the country leaderboard upon the successful verification by the system.

## [7]. Assumptions

- It is assumed that the users of this database system will have all the necessary hardware and software required for running this application.
- Moreover, it is assumed that the information stored in the database gets updated in real-time.
- It is assumed that the users of this database system have a stable internet connection.
- The database is consistent.
- The integrity of the data is maintained throughout the database.
- The users will have access to alternative resources in the event of any unprecedented situation.

## [8]. Business Constraints

- The amount of software and hardware required for the database system is limited.
- The amount of computing power available for the database system is also limited.
- The capacity of the database is limited.
- The number of users that can access the database simultaneously is also limited.

## Section2: Noun Analysis

### [1]. Noun and Verb Analysis

| Sr. No. | Nouns               | Verbs        |
|---------|---------------------|--------------|
| 1       | management system   | details      |
| 2       | information         | document     |
| 3       | sporting events     | records      |
| 4       | angles              | keeps        |
| 5       | details             | can          |
| 6       | sports              | create       |
| 7       | events              | will         |
| 8       | champions           | make         |
| 9       | medalists           | participates |
| 10      | results             | tables       |
| 11      | medals              | get          |
| 12      | standing            | references   |
| 13      | venues              | type         |
| 14      | chief guests        | list         |
| 15      | intended audience   | provide      |
| 16      | reading suggestions | access       |
| 17      | users               | models       |

|    |                                       |           |
|----|---------------------------------------|-----------|
| 18 | athletes                              | referred  |
| 19 | competitions                          | offer     |
| 20 | audiences                             | inferred  |
| 21 | sports enthusiasts                    | implement |
| 22 | media partners                        | work      |
| 23 | broadcast partners                    | track     |
| 24 | newspapers                            | sponsor   |
| 25 | magazines                             | takes     |
| 26 | social media                          | planning  |
| 27 | sports analysts                       | stores    |
| 28 | trainers                              | supervise |
| 29 | coaches                               | increase  |
| 30 | developers                            | modify    |
| 31 | technical difficulties                | insert    |
| 32 | Managers                              | delete    |
| 33 | sponsors                              | update    |
| 34 | brands                                | standing  |
| 35 | companies                             | guide     |
| 36 | International Olympic Committee (IOC) | interests |
| 37 | host city                             | like      |

|    |                                 |                     |
|----|---------------------------------|---------------------|
| 38 | games                           | scope               |
| 39 | broadcasting rights             | influence           |
| 40 | International Federations (IFs) | effects             |
| 41 | ages                            | maintain            |
| 42 | strategies                      | examined            |
| 43 | websites                        | needs               |
| 44 | product scope                   | ensures             |
| 45 | awareness                       | allow               |
| 46 | youth                           | view                |
| 47 | discrimination                  | have                |
| 48 | spirit                          | targets             |
| 49 | friendship                      | requires            |
| 50 | solidarity                      | adds                |
| 51 | description                     | removes             |
| 52 | system                          | gives               |
| 53 | structure                       | analyze             |
| 54 | scope                           | Event leaderboard   |
| 55 | cities                          | Sport leaderboard   |
| 56 | countries                       | Country leaderboard |
| 57 | impacts                         | manage              |
| 58 | opportunities                   | case                |

|    |                    |            |
|----|--------------------|------------|
| 59 | exports            | separate   |
| 60 | local corporations | indicate   |
| 61 | non-profit sector  | study      |
| 62 | civil society      | review     |
| 63 | governing bodies   | includes   |
| 64 | communities        | improve    |
| 65 | data               | won        |
| 66 | nations            | holds      |
| 67 | males              | plays      |
| 68 | females            | host       |
| 69 | leaderboards       | impacts    |
| 70 | tables             | represents |
| 71 | functionalities    |            |
| 72 | analysis           |            |
| 73 | Olympics ID        |            |
| 74 | Olympics name      |            |
| 75 | City name          |            |
| 76 | Country name       |            |
| 77 | Year               |            |
| 78 | Developer ID       |            |
| 79 | Developer name     |            |

|     |                     |  |
|-----|---------------------|--|
| 80  | Manager ID          |  |
| 81  | Manager name        |  |
| 82  | Owner ID            |  |
| 83  | Owner name          |  |
| 84  | Athlete ID          |  |
| 85  | Athlete name        |  |
| 86  | Gender              |  |
| 87  | effects             |  |
| 88  | services            |  |
| 89  | functions           |  |
| 90  | roles               |  |
| 91  | Country ID          |  |
| 92  | Country name        |  |
| 93  | Sport ID            |  |
| 94  | Sport name          |  |
| 95  | Event ID            |  |
| 96  | Event name          |  |
| 97  | Event leaderboard   |  |
| 98  | Sport leaderboard   |  |
| 99  | Country leaderboard |  |
| 100 | winners             |  |

|     |                       |  |
|-----|-----------------------|--|
| 101 | gold medalist         |  |
| 102 | silver medalist       |  |
| 103 | bronze medalist       |  |
| 104 | gold medals           |  |
| 105 | silver medals         |  |
| 106 | bronze medals         |  |
| 107 | total medals          |  |
| 108 | First rank            |  |
| 109 | D.O.B                 |  |
| 110 | Age                   |  |
| 111 | Height                |  |
| 112 | Weight                |  |
| 113 | factors               |  |
| 114 | requirements          |  |
| 115 | concepts              |  |
| 116 | diagrams              |  |
| 117 | contender             |  |
| 118 | IOC officials         |  |
| 119 | system administrators |  |
| 120 | problems              |  |
| 121 | actions               |  |

|     |                    |  |
|-----|--------------------|--|
| 122 | ideas              |  |
| 123 | broadcast rights   |  |
| 124 | types              |  |
| 125 | viewers            |  |
| 126 | media              |  |
| 127 | analysts           |  |
| 128 | people             |  |
| 129 | age group          |  |
| 130 | user groups        |  |
| 131 | Second rank        |  |
| 132 | Third rank         |  |
| 133 | responses          |  |
| 134 | expectations       |  |
| 135 | time               |  |
| 136 | date               |  |
| 137 | User Interface     |  |
| 138 | feedback           |  |
| 139 | suggestions        |  |
| 140 | privileges         |  |
| 141 | operations         |  |
| 142 | background reading |  |

|     |             |  |
|-----|-------------|--|
| 143 | interest    |  |
| 144 | application |  |
| 145 | software    |  |
| 146 | hardware    |  |
| 147 | resources   |  |
| 148 | power       |  |
| 149 | situation   |  |
| 150 | capacity    |  |
| 151 | security    |  |
| 152 | integrity   |  |
| 153 | consistency |  |
| 154 | models      |  |

## [2]. Accepted Nouns and Verbs List

| Candidate Entity Set | Candidate Attribute Set  | Candidate Relationship Set  |
|----------------------|--|---|
| Olympics             | <u>Olympics_ID</u> , Olympics_Name<br>(City, Year)                         | Host, Event leaderboard,<br>Sport leaderboard, Country<br>leaderboard |
| Athlete              | <u>Athlete_ID</u> , Athlete_Name,<br>Gender, D.O.B, Age, Height,<br>Weight | Plays, Represents   |
| Developer            | <u>Developer_ID</u> ,<br>Developer_Name                                    |   |
| Manager              | <u>Manager_ID</u> , Manager_Name   |   |
| Owner                | <u>Owner_ID</u> , Owner_Name   |   |
| Country              | <u>Country_ID</u> , Country_Name   | Host, Represents, Country<br>leaderboard, Represents                  |
| Event                | <u>Event_ID</u> , Event_Name,<br>Event_Type                                | Includes, Event leaderboard   |
| Sport                | <u>Sport_ID</u> , Sport_Name   | Plays, Includes, Sport<br>leaderboard                                 |

### [3]. Rejected Nouns and Verbs List

#### Rejected Nouns

| Nouns               | Reject reason |
|---------------------|---------------|
| management system   | General       |
| information         | General       |
| sporting events     | Duplicate     |
| angles              | Vague         |
| details             | Duplicate     |
| champions           | Duplicate     |
| medalists           | Duplicate     |
| results             | Association   |
| medals              | General       |
| standing            | Association   |
| venues              | General       |
| intended audience   | Irrelevant    |
| reading suggestions | Irrelevant    |
| users               | General       |
| competitions        | Duplicate     |
| audiences           | General       |
| ages                | Duplicate     |
| sports enthusiasts  | Duplicate     |

|                                       |            |
|---------------------------------------|------------|
| media partners                        | Irrelevant |
| broadcast partners                    | Irrelevant |
| newspapers                            | Irrelevant |
| magazines                             | Irrelevant |
| social media                          | Irrelevant |
| sports analysts                       | Duplicate  |
| trainers                              | Duplicate  |
| coaches                               | Duplicate  |
| technical difficulties                | Vague      |
| sponsors                              | Irrelevant |
| brands                                | Irrelevant |
| companies                             | Irrelevant |
| International Olympic Committee (IOC) | Irrelevant |
| host city                             | Duplicate  |
| games                                 | General    |
| broadcasting rights                   | Vague      |
| International Federations (IFs)       | Irrelevant |
| governing bodies                      | Irrelevant |
| websites                              | Irrelevant |
| product scope                         | Irrelevant |

|                    |            |
|--------------------|------------|
| awareness          | Vague      |
| youth              | Duplicate  |
| discrimination     | Vague      |
| spirit             | Vague      |
| friendship         | Vague      |
| solidarity         | Vague      |
| system             | General    |
| structure          | General    |
| scope              | Duplicate  |
| cities             | Duplicate  |
| countries          | Duplicate  |
| impacts            | Vague      |
| opportunities      | Vague      |
| exports            | Vague      |
| local corporations | Irrelevant |
| non-profit sector  | Irrelevant |
| civil society      | Irrelevant |
| effects            | Vague      |
| communities        | Irrelevant |
| data               | General    |
| strategies         | Vague      |

|                 |            |
|-----------------|------------|
| nations         | Duplicate  |
| males           | Irrelevant |
| females         | Irrelevant |
| leaderboards    | General    |
| tables          | Irrelevant |
| functionalities | Irrelevant |
| analysis        | General    |
| Olympics ID     | Attribute  |
| Olympics name   | Attribute  |
| city name       | Attribute  |
| country name    | Attribute  |
| year            | Attribute  |
| Developer ID    | Attribute  |
| Developer name  | Attribute  |
| Manager ID      | Attribute  |
| Manager name    | Attribute  |
| Owner ID        | Attribute  |
| Owner name      | Attribute  |
| Athlete ID      | Attribute  |
| Athlete name    | Attribute  |
| gender          | Attribute  |

|                     |             |
|---------------------|-------------|
| D.O.B               | Attribute   |
| age                 | Attribute   |
| height              | Attribute   |
| weight              | Attribute   |
| Country ID          | Attribute   |
| Country name        | Attribute   |
| Sport ID            | Attribute   |
| Sport name          | Attribute   |
| Event ID            | Attribute   |
| Event name          | Attribute   |
| Event leaderboard   | Association |
| Sport leaderboard   | Association |
| Country leaderboard | Association |
| winners             | Association |
| First rank          | Attribute   |
| Second rank         | Attribute   |
| Third rank          | Attribute   |
| gold medalist       | Attribute   |
| silver medalist     | Attribute   |
| bronze medalist     | Attribute   |
| gold medals         | Attribute   |

|                       |            |
|-----------------------|------------|
| silver medals         | Attribute  |
| bronze medals         | Attribute  |
| total medals          | Attribute  |
| background reading    | Irrelevant |
| description           | Vague      |
| models                | Irrelevant |
| services              | Irrelevant |
| functions             | Irrelevant |
| roles                 | Irrelevant |
| requirements          | Irrelevant |
| concepts              | Vague      |
| diagrams              | Vague      |
| contender             | Duplicate  |
| IOC officials         | Irrelevant |
| system administrators | Irrelevant |
| problems              | Vague      |
| actions               | Vague      |
| ideas                 | Vague      |
| broadcast rights      | Vague      |
| types                 | Vague      |
| viewers               | Duplicate  |

|                |            |
|----------------|------------|
| media          | Vague      |
| analysts       | Duplicate  |
| people         | General    |
| age group      | Irrelevant |
| user groups    | Duplicate  |
| interest       | Vague      |
| factors        | Irrelevant |
| responses      | Irrelevant |
| expectations   | Irrelevant |
| time           | Irrelevant |
| date           | Irrelevant |
| User Interface | Vague      |
| feedback       | Irrelevant |
| suggestions    | Irrelevant |
| privileges     | Irrelevant |
| operations     | Irrelevant |
| Irrelevant     | Irrelevant |
| software       | Irrelevant |
| hardware       | Irrelevant |
| resources      | Irrelevant |
| power          | Vague      |

|             |            |
|-------------|------------|
| situation   | Vague      |
| capacity    | Irrelevant |
| security    | Irrelevant |
| integrity   | Irrelevant |
| consistency | Irrelevant |

## Rejected Verbs

| Verbs     | Reject reason |
|-----------|---------------|
| details   | General       |
| document  | General       |
| records   | Duplicate     |
| keeps     | Duplicate     |
| can       | General       |
| create    | Vague         |
| will      | General       |
| make      | General       |
| tables    | Irrelevant    |
| implement | Vague         |
| work      | Vague         |
| track     | Duplicate     |
| sponsor   | Vague         |

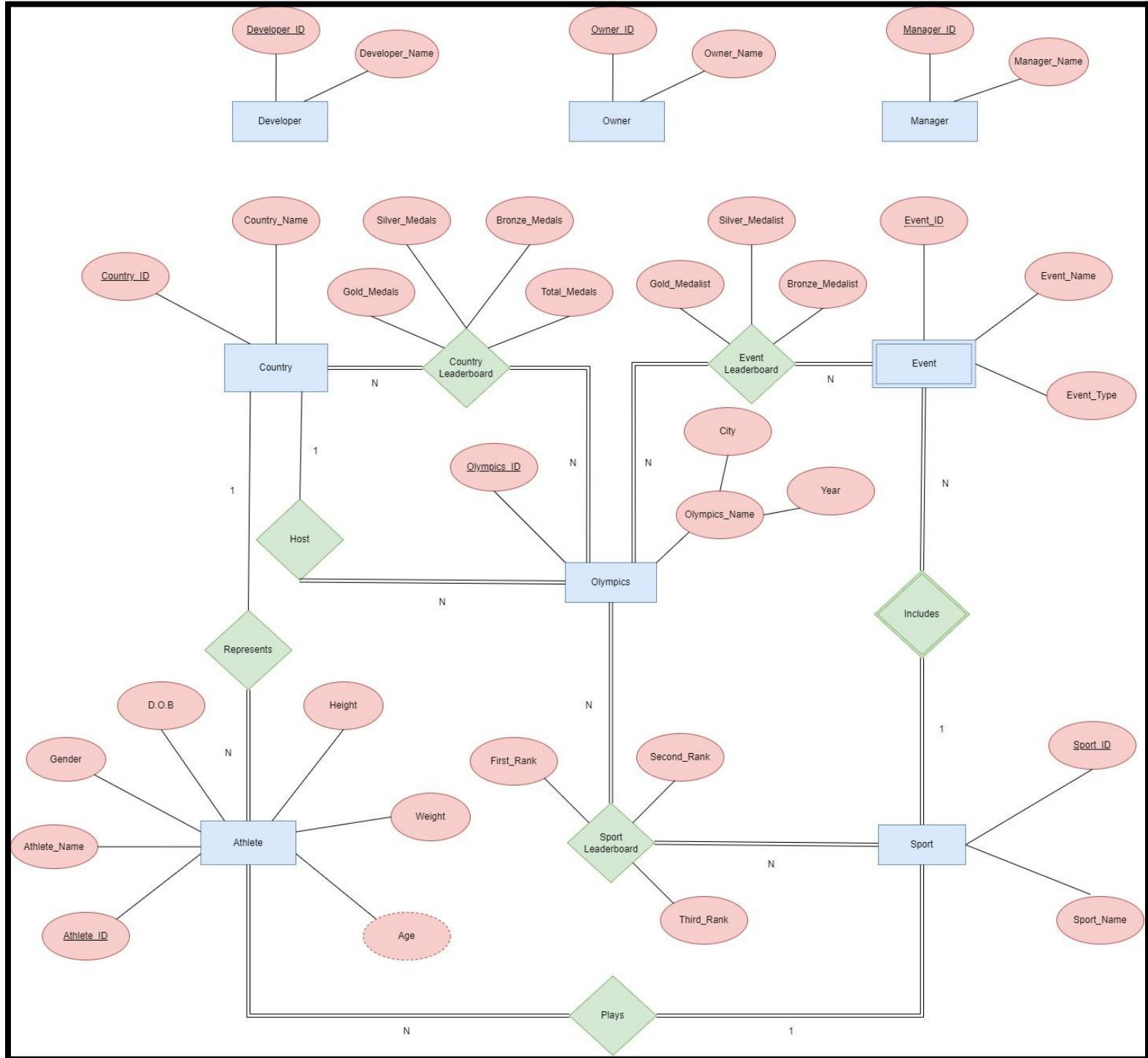
|            |            |
|------------|------------|
| takes      | Irrelevant |
| planning   | Vague      |
| stores     | Vague      |
| supervise  | Vague      |
| increase   | Vague      |
| modify     | Irrelevant |
| get        | Vague      |
| references | Irrelevant |
| type       | Irrelevant |
| list       | Irrelevant |
| provide    | Vague      |
| access     | General    |
| models     | Irrelevant |
| referred   | Vague      |
| offer      | Duplicate  |
| inferred   | Duplicate  |
| maintain   | Vague      |
| examined   | Duplicate  |
| needs      | General    |
| ensures    | Vague      |
| allow      | Duplicate  |

|           |            |
|-----------|------------|
| view      | General    |
| have      | General    |
| requires  | Vague      |
| adds      | Irrelevant |
| removes   | Irrelevant |
| gives     | Duplicate  |
| insert    | Duplicate  |
| delete    | Duplicate  |
| update    | Duplicate  |
| guide     | Irrelevant |
| interests | Irrelevant |
| like      | Vague      |
| scope     | General    |
| influence | Irrelevant |
| effects   | Duplicate  |
| impacts   | Duplicate  |
| manage    | General    |
| case      | General    |
| separate  | Irrelevant |
| indicate  | Irrelevant |
| study     | General    |

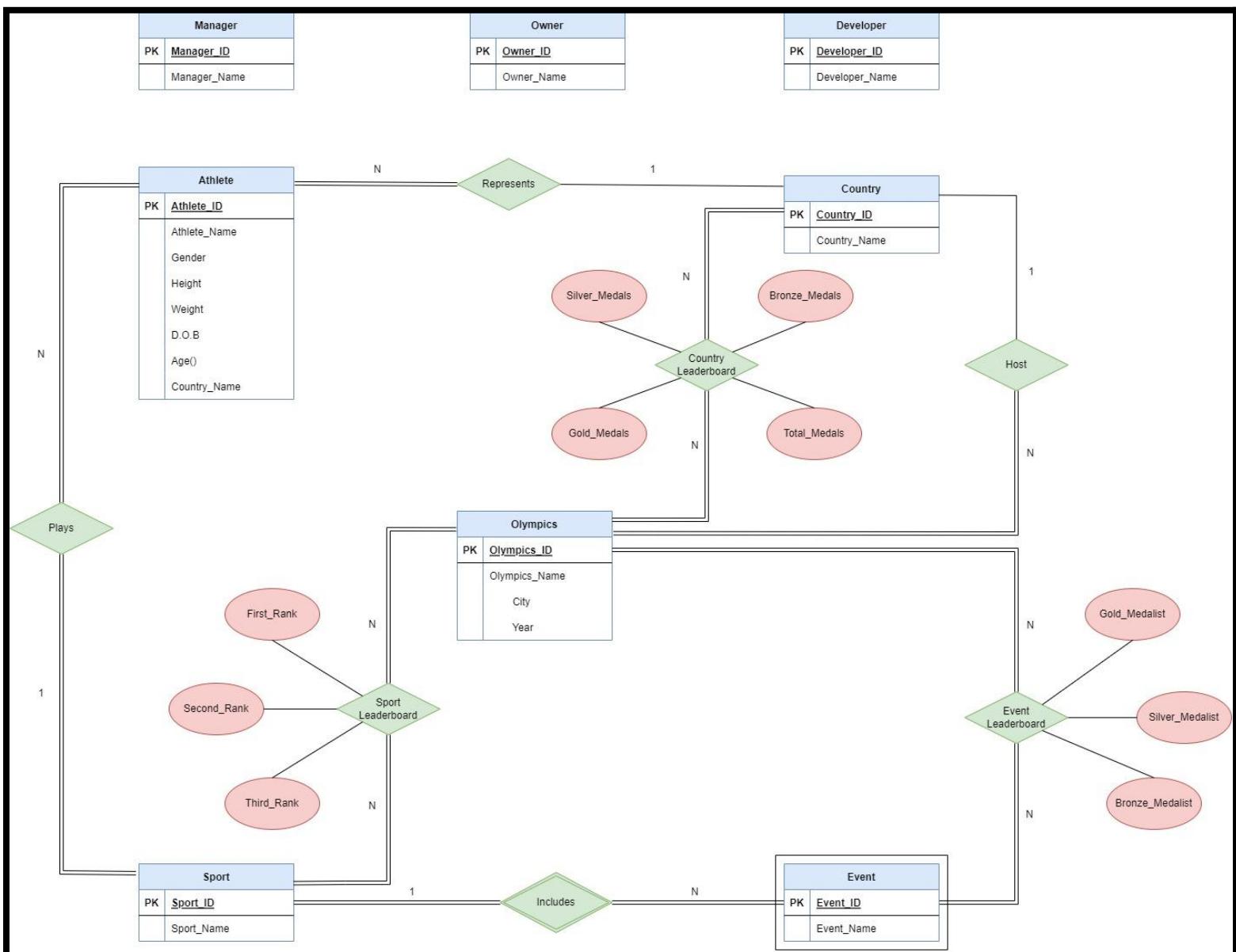
|              |            |
|--------------|------------|
| review       | Irrelevant |
| improve      | Irrelevant |
| analyze      | Irrelevant |
| won          | Duplicate  |
| holds        | Duplicate  |
| participates | Duplicate  |
| standing     | Duplicate  |
| results      | Duplicate  |

## Section3: ER-Diagrams all versions

### [1]. ER diagram Version-1

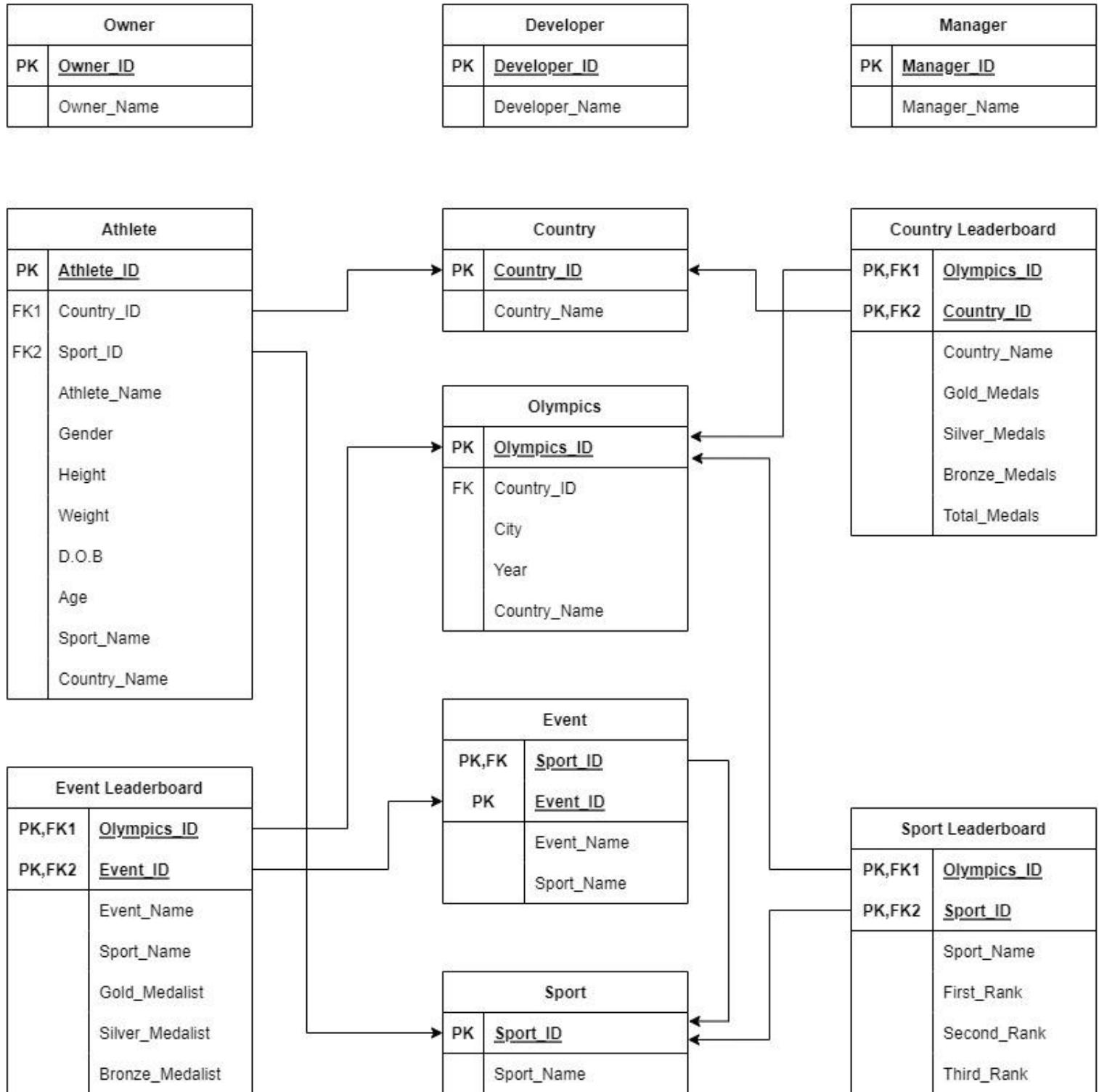


## [2]. ER diagram Version-2 (Final Version)



**NOTE:** Here, 1 and N on the relationship indicate one and many, respectively

## Section4: Mapping of ER-Model to Relational Model



## ❖ Relational Schema

- **Olympics**(Olympics\_ID, Country\_ID, City, Year, Country\_Name)
  - FK Country\_ID references **Country**
- **Athlete**(Athlete\_ID, Country\_ID, Sport\_ID, Athlete\_Name, Gender, Height, Weight, D.O.B, Age, Sport\_Name, Country\_Name)
  - FK Country\_ID references **Country**
  - FK Sport\_ID references **Sport**
- **Country**(Country\_ID, Country\_Name)
- **Sport**(Sport\_ID, Sport\_Name)
- **Event**(Sport\_ID, Event\_ID, Event\_Name, Sport\_Name)
  - FK Sport\_ID references **Sport**
- **Event Leaderboard**(Olympics\_ID, Event\_ID, Event\_Name, Sport\_Name, Gold\_Medalist, Silver\_Medalist, Bronze\_Medalist)
  - FK Olympics\_ID references **Olympics**
  - FK Event\_ID references **Event**
- **Sport Leaderboard**(Olympics\_ID, Sport\_ID, Sport\_Name, First\_Rank, Second\_Rank, Third\_Rank)
  - FK Olympics\_ID references **Olympics**
  - FK Sport\_ID references **Sport**
- **Country Leaderboard**(Olympics\_ID, Country\_ID, Country\_Name, Gold\_Medals, Silver\_Medals, Bronze\_Medals, Total\_Medals)
  - FK Olympics\_ID references **Olympics**
  - FK Country\_ID references **Country**

- **Developer**(Developer\_ID, Developer\_Name)
- **Manager**(Manager\_ID, Manager\_Name)
- **Owner**(Owner\_ID, Owner\_Name)

**❖ Relationship sets removed while mapping ER model to Relational model**

- **Represents**(Athlete\_ID, Country\_ID)  
FK Athlete\_ID references **Athlete**  
FK Country\_ID references **Country**

**Reason:** Athlete-Represents-Country is **many-one** relationship with **total participation on many side (Athlete)**. So, we can remove the **Represents** relationship set by including the **primary key** of the **one side (Country)** in the relational schema arising from the **many side (Athlete)**.

- **Host**(Olympics\_ID, Country\_ID, Year)  
FK Olympics\_ID references **Olympics**  
FK Country\_ID references **Country**

**Reason:** Country-Host-Olympics is **one-many** relationship with **total participation on many side (Olympics)**. So, we can remove the **Host** relationship set by including the **primary key** of the **one side (Country)** in the relational schema arising from the **many side (Olympics)**.

- **Plays**(Athlete\_ID, Sport\_ID)  
FK Athlete\_ID references **Athlete**  
FK Sport\_ID references **Sport**

**Reason:** Athlete-Plays-Sport is many-one relationship with total participation on many side (Athlete). So, we can remove the Plays relationship set by including the primary key of the one side (Sport) in the relational schema arising from the many side (Athlete).

- **Includes**(Sport\_ID, Event\_ID)
  - FK Sport\_ID references Sport
  - FK Event\_ID references Event

**Reason:** Here, the weak entity set (Event) already contains all the attributes of its identifying relationship set (Includes). Hence, Includes relationship set is redundant and can be removed.

## ❖ DDL Scripts

### 1) Country

```
CREATE TABLE IF NOT EXISTS Country
(
    Country_ID bigint NOT NULL,
    Country_Name character varying,
    PRIMARY KEY (Country_ID)
);
```

### 2) Olympics

```
CREATE TABLE IF NOT EXISTS Olympics
(
    Olympics_ID bigint NOT NULL,
    Country_ID bigint NOT NULL,
    City character varying,
    Year bigint,
    Country_Name character varying,
```

```
PRIMARY KEY (Olympics_ID),
FOREIGN KEY (Country_ID) REFERENCES Country MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID
);
```

### 3) Sport

```
CREATE TABLE IF NOT EXISTS Sport
(
Sport_ID bigint NOT NULL,
Sport_Name character varying,
PRIMARY KEY (Sport_ID)
);
```

### 4) Event

```
CREATE TABLE IF NOT EXISTS Event
(
Event_ID bigint NOT NULL,
Sport_ID bigint NOT NULL,
Event_Name character varying,
Sport_Name character varying,
PRIMARY KEY (Sport_ID, Event_ID),
FOREIGN KEY (Sport_ID) REFERENCES Sport MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID
);
```

### 5) Athlete

```
CREATE TABLE IF NOT EXISTS Athlete
(
Athlete_ID bigint NOT NULL,
Country_ID bigint NOT NULL,
Sport_ID bigint NOT NULL,
Athlete_Name character varying,
Gender character varying,
Height bigint,
Weight bigint,
DOB character varying,
Age bigint,
Country_Name character varying,
Sport_Name character varying,
PRIMARY KEY (Athlete_ID),
FOREIGN KEY (Country_ID) REFERENCES Country MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID,
FOREIGN KEY (Sport_ID) REFERENCES Sport MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID
);
```

## 6) Event Leaderboard

```
CREATE TABLE IF NOT EXISTS Event_Leaderboard
(
Olympics_ID bigint NOT NULL,
Event_ID bigint NOT NULL,
Sport_ID bigint NOT NULL,
Event_Name character varying,
Sport_Name character varying,
Gold_Medalist character varying,
```

```
Silver_Medalist character varying,  
Bronze_Medalist character varying,  
  
PRIMARY KEY (Olympics_ID, Event_ID, Sport_ID),  
FOREIGN KEY (Olympics_ID) REFERENCES Olympics MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE  
NOT VALID,  
FOREIGN KEY (Event_ID, Sport_ID) REFERENCES Event MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE  
NOT VALID  
);
```

## 7) Sport Leaderboard

```
CREATE TABLE IF NOT EXISTS Sport Leaderboard  
(  
Olympics_ID bigint NOT NULL,  
Sport_ID bigint NOT NULL,  
Sport_Name character varying,  
First_Rank character varying,  
Second_Rank character varying,  
Third_Rank character varying,  
PRIMARY KEY (Olympics_ID, Sport_ID),  
FOREIGN KEY (Olympics_ID) REFERENCES Olympics MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE  
NOT VALID,  
FOREIGN KEY (Sport_ID) REFERENCES Sport MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE  
NOT VALID  
);
```

## 8) Country Leaderboard

```
CREATE TABLE IF NOT EXISTS Country_Leaderboard
(
    Olympics_ID bigint NOT NULL,
    Country_ID bigint NOT NULL,
    Country_Name character varying,
    Gold_Medals bigint,
    Silver_Medals bigint,
    Bronze_Medals bigint,
    Total_Medals bigint,
    PRIMARY KEY (Olympics_ID, Country_ID),
    FOREIGN KEY (Olympics_ID) REFERENCES Olympics MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    NOT VALID,
    FOREIGN KEY (Country_ID) REFERENCES Country MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    NOT VALID
);
```

## 9) Owner

```
CREATE TABLE IF NOT EXISTS Owner
(
    Owner_ID bigint NOT NULL,
    Owner_Name character varying,
    PRIMARY KEY (Owner_ID)
);
```

## **10) Developer**

```
CREATE TABLE IF NOT EXISTS Developer
(
Developer_ID bigint NOT NULL,
Developer_Name character varying,
PRIMARY KEY (Developer_ID)
);
```

## **11) Manager**

```
CREATE TABLE IF NOT EXISTS Manager
(
Manager_ID bigint NOT NULL,
Manager_Name character varying,
PRIMARY KEY (Manager_ID)
);
```

❖ Relationship sets removed while mapping ER model to Relational model

## **1) Represents**

```
CREATE TABLE IF NOT EXISTS Represents
(
Athlete_ID bigint NOT NULL,
Country_ID bigint NOT NULL,
PRIMARY KEY (Athlete_ID, Country_ID)
FOREIGN KEY (Athlete_ID) REFERENCES Athlete MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID,
FOREIGN KEY (Country_ID) REFERENCES Country MATCH SIMPLE
```

```
ON UPDATE CASCADE  
ON DELETE CASCADE  
NOT VALID  
);
```

**Reason:** Athlete-Represents-Country is **many-one** relationship with **total participation** on **many side (Athlete)**. So, we can remove the **Represents** relationship set by including the **primary key** of the **one side (Country)** in the relational schema arising from the **many side (Athlete)**.

## 2) Host

```
CREATE TABLE IF NOT EXISTS Host  
(  
    Olympics_ID bigint NOT NULL,  
    Country_ID bigint NOT NULL,  
    Year bigint,  
    PRIMARY KEY (Olympics_ID, Country_ID)  
    FOREIGN KEY (Olympics_ID) REFERENCES Olympics MATCH SIMPLE  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
    NOT VALID,  
    FOREIGN KEY (Country_ID) REFERENCES Country MATCH SIMPLE  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
    NOT VALID  
);
```

**Reason:** Country-Host-Olympics is **one-many** relationship with **total participation** on **many side (Olympics)**. So, we can remove the **Host** relationship set by including the **primary key** of the **one side (Country)** in the relational schema arising from the **many side (Olympics)**.

### 3) Plays

```
CREATE TABLE IF NOT EXISTS Plays
(
Athlete_ID bigint NOT NULL,
Sport_ID bigint NOT NULL,
PRIMARY KEY (Athlete_ID, Sport_ID)
FOREIGN KEY (Athlete_ID) REFERENCES Athlete MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID,
FOREIGN KEY (Sport_ID) REFERENCES Sport MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID
);
```

**Reason:** Athlete-Plays-Sport is **many-one** relationship with **total participation** on **many side (Athlete)**. So, we can remove the **Plays** relationship set by including the **primary key** of the **one side (Sport)** in the relational schema arising from the **many side (Athlete)**.

### 4) Includes

```
CREATE TABLE IF NOT EXISTS Includes
(
Sport_ID bigint NOT NULL,
Event_ID bigint NOT NULL,
PRIMARY KEY (Sport_ID, Event_ID),
FOREIGN KEY (Sport_ID) REFERENCES Sport MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID,
FOREIGN KEY (Event_ID) REFERENCES Event MATCH SIMPLE
```

```
ON UPDATE CASCADE  
ON DELETE CASCADE  
NOT VALID  
);
```

**Reason:** Here, the **weak entity set (Event)** already contains all the attributes of its **identifying relationship set (Includes)**. Hence, **Includes** relationship set is **redundant** and can be removed.

## Section5: Normalization and Schema Refinement

### ❖ Relational Schema

- **Olympics**(Olympics\_ID, Country\_ID, City, Year, Country\_Name)
  - FK Country\_ID references **Country**
- **Athlete**(Athlete\_ID, Country\_ID, Sport\_ID, Athlete\_Name, Gender, Height, Weight, D.O.B, Age, Sport\_Name, Country\_Name)
  - FK Country\_ID references **Country**
  - FK Sport\_ID references **Sport**
- **Country**(Country\_ID, Country\_Name)
- **Sport**(Sport\_ID, Sport\_Name)
- **Event**(Event\_ID, Sport\_ID, Event\_Name, Sport\_Name)
  - FK Sport\_ID references **Sport**
- **Event Leaderboard**(Olympics\_ID, Event\_ID, Sport\_ID, Event\_Name, Sport\_Name, Gold\_Medalist, Silver\_Medalist, Bronze\_Medalist)
  - FK Olympics\_ID references **Olympics**
  - FK Event\_ID references **Event**
  - FK Sport\_ID references **Sport**
- **Sport Leaderboard**(Olympics\_ID, Sport\_ID, Sport\_Name, First\_Rank, Second\_Rank, Third\_Rank)
  - FK Olympics\_ID references **Olympics**
  - FK Sport\_ID references **Sport**
- **Country Leaderboard**(Olympics\_ID, Country\_ID, Country\_Name, Gold\_Medals, Silver\_Medals, Bronze\_Medals, Total\_Medals)

- FK Olympics\_ID references **Olympics**
- FK Country\_ID references **Country**
- **Developer**(Developer\_ID, Developer\_Name)
- **Manager**(Manager\_ID, Manager\_Name)
- **Owner**(Owner\_ID, Owner\_Name)

## ❖ Functional Dependencies

- **Olympics**(Olympics\_ID → Country\_ID, Olympics\_ID → City, Olympics\_ID → Year, Olympics\_ID → Country\_Name, Country\_ID → Country\_Name)
- **Athlete**(Athlete\_ID → Country\_ID, Athlete\_ID → Sport\_ID, Athlete\_ID → Athlete\_Name, Athlete\_ID → Gender, Athlete\_ID → Height, Athlete\_ID → Weight, Athlete\_ID → D.O.B, Athlete\_ID → Age, Athlete\_ID → Sport\_Name, Athlete\_ID → Country\_Name, D.O.B → Age, Sport\_ID → Sport\_Name, Country\_ID → Country\_Name)
- **Country**(Country\_ID → Country\_Name)
- **Sport**(Sport\_ID → Sport\_Name)
- **Event**(Event\_ID → Sport\_ID, Event\_ID → Event\_Name, Event\_ID → Sport\_Name, Sport\_ID → Sport\_Name)
- **Event Leaderboard**((Olympics\_ID, Event\_ID) → Event\_Name, (Olympics\_ID, Event\_ID) → Sport\_Name, (Olympics\_ID, Event\_ID) → Gold\_Medalist, (Olympics\_ID, Event\_ID) → Silver\_Medalist, (Olympics\_ID, Event\_ID) → Bronze\_Medalist, Event\_ID → Sport\_ID,

$\text{Event\_ID} \rightarrow \text{Event\_Name}$ ,  $\text{Event\_ID} \rightarrow \text{Sport\_Name}$ ,  $\text{Sport\_ID} \rightarrow \text{Sport\_Name}$ )

- **Sport Leaderboard**((Olympics\_ID, Sport\_ID) → Sport\_Name, ((Olympics\_ID, Sport\_ID) → First\_Rank, ((Olympics\_ID, Sport\_ID) → Second\_Rank, ((Olympics\_ID, Sport\_ID) → Third\_Rank, Sport\_ID → Sport\_Name))
- **Country Leaderboard**((Olympics\_ID, Country\_ID) → Country\_Name, (Olympics\_ID, Country\_ID) → Gold\_Medals, (Olympics\_ID, Country\_ID) → Silver\_Medals, (Olympics\_ID, Country\_ID) → Bronze\_Medals, (Olympics\_ID, Country\_ID) → Total\_Medals, Country\_ID → Country\_Name)
- **Developer**(Developer\_ID → Developer\_Name)
- **Manager**(Manager\_ID → Manager\_Name)
- **Owner**(Owner\_ID → Owner\_Name)

## ❖ Redundancies and Analysis

- **Olympics**(Olympics\_ID, Country\_ID, City, Year, Country\_Name)
  - FK Country\_ID references **Country**
  - **Redundancy:** There can be the same City associated with different Olympics\_ID
  - **Redundancy:** Transitive dependency (Olympics\_ID → Country\_ID and Country\_ID → Country\_Name implies Olympics\_ID → Country\_Name)
  - No partial dependencies
  - No anomalies

- **Athlete**(Athlete\_ID, Country\_ID, Sport\_ID, Athlete\_Name, Gender, Height, Weight, D.O.B, Age, Sport\_Name, Country\_Name)
  - FK Country\_ID references **Country**
  - FK Sport\_ID references **Sport**
  - **Redundancy:** There can be the same Sport\_Name, Athlete\_Name, or Country\_Name associated with different Athlete\_ID
  - **Redundancy:** Transitive dependency ( $\text{Athlete\_ID} \rightarrow \text{Country\_ID}$  and  $\text{Country\_ID} \rightarrow \text{Country\_Name}$  implies  $\text{Athlete\_ID} \rightarrow \text{Country\_Name}$ )
  - **Redundancy:** Transitive dependency ( $\text{Athlete\_ID} \rightarrow \text{Sport\_ID}$  and  $\text{Sport\_ID} \rightarrow \text{Sport\_Name}$  implies  $\text{Athlete\_ID} \rightarrow \text{Sport\_Name}$ )
  - **Redundancy:** Transitive dependency ( $\text{Athlete\_ID} \rightarrow \text{D.O.B}$  and  $\text{D.O.B} \rightarrow \text{Age}$  implies  $\text{Athlete\_ID} \rightarrow \text{Age}$ )
  - No partial dependencies
  - No anomalies
- **Country**(Country\_ID, Country\_Name)
  - No partial dependencies
  - No transitive dependencies
  - No redundancies
  - No anomalies
- **Sport**(Sport\_ID, Sport\_Name)
  - No partial dependencies
  - No transitive dependencies
  - No redundancies
  - No anomalies
- **Event**(Event\_ID, Sport\_ID, Event\_Name, Sport\_Name)
  - FK Sport\_ID references **Sport**
  - **Redundancy:** There can be the same Sport associated with different Events

- **Redundancy:** Transitive dependency ( $\text{Event\_ID} \rightarrow \text{Sport\_ID}$  and  $\text{Sport\_ID} \rightarrow \text{Sport\_Name}$  implies  $\text{Event\_ID} \rightarrow \text{Sport\_Name}$ )
  - No partial dependencies
  - No anomalies
- **Event Leaderboard**(Olympics\_ID, Event\_ID, Sport\_ID, Event\_Name, Sport\_Name, Gold\_Medalist, Silver\_Medalist, Bronze\_Medalist)
  - FK Olympics\_ID references **Olympics**
  - FK Event\_ID references **Event**
  - FK Sport\_ID references **Sport**
  - **Redundancy:** Partial dependency ( $\text{Event\_ID} \rightarrow \text{Sport\_ID}$ )
  - **Redundancy:** Partial dependency ( $\text{Event\_ID} \rightarrow \text{Event\_Name}$ )
  - **Redundancy:** Partial dependency ( $\text{Event\_ID} \rightarrow \text{Sport\_Name}$ )
  - **Redundancy:** Transitive dependency ( $\text{Event\_ID} \rightarrow \text{Sport\_ID}$  and  $\text{Sport\_ID} \rightarrow \text{Sport\_Name}$  implies  $\text{Event\_ID} \rightarrow \text{Sport\_Name}$ )
  - No anomalies
- **Sport Leaderboard**(Olympics\_ID, Sport\_ID, Sport\_Name, First\_Rank, Second\_Rank, Third\_Rank)
  - FK Olympics\_ID references **Olympics**
  - FK Sport\_ID references **Sport**
  - **Redundancy:** Partial dependency ( $\text{Sport\_ID} \rightarrow \text{Sport\_Name}$ )
  - No transitive dependencies
  - No anomalies
- **Country Leaderboard**(Olympics\_ID, Country\_ID, Country\_Name, Gold\_Medals, Silver\_Medals, Bronze\_Medals, Total\_Medals)
  - FK Olympics\_ID references **Olympics**
  - FK Country\_ID references **Country**
  - **Redundancy:** Partial dependency ( $\text{Country\_ID} \rightarrow \text{Country\_Name}$ )

- No transitive dependencies
  - No anomalies
- **Developer**(Developer\_ID, Developer\_Name)
  - **Redundancy:** There can be the same Developer\_Name associated with different Developer\_ID
  - No partial dependencies
  - No transitive dependencies
  - No anomalies
- **Manager**(Manager\_ID, Manager\_Name)
  - **Redundancy:** There can be the same Manager\_Name associated with different Manager\_ID
  - No partial dependencies
  - No transitive dependencies
  - No anomalies
- **Owner**(Owner\_ID, Owner\_Name)
  - **Redundancy:** There can be the same Owner\_Name associated with different Owner\_ID
  - No partial dependencies
  - No transitive dependencies
  - No anomalies

## ❖ Schema Refinement and Normalization up to 3NF/BCNF

- **Olympics**(Olympics\_ID, Country\_ID, City, Year, Country\_Name)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.

- There is only one candidate key (`Olympics_ID`) and only a single attribute in the candidate key. Therefore, there are no partial dependencies. Hence, it is already in 2NF.
  - There is a transitive dependency ( $\text{Olympics\_ID} \rightarrow \text{Country\_ID}$  and  $\text{Country\_ID} \rightarrow \text{Country\_Name}$  implies  $\text{Olympics\_ID} \rightarrow \text{Country\_Name}$ ). Hence, it is not in 3NF. We removed the attribute `Country_Name`, and therefore, there are no transitive dependencies anymore. Hence, it is in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.
- **Athlete**(`Athlete_ID`, `Country_ID`, `Sport_ID`, `Athlete_Name`, `Gender`, `Height`, `Weight`, `D.O.B`, `Age`, `Sport_Name`, `Country_Name`)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (`Athlete_ID`) and only a single attribute in the candidate key. Therefore, there are no partial dependencies. Hence, it is already in 2NF.
  - There are transitive dependencies ( $\text{Athlete\_ID} \rightarrow \text{Country\_ID}$  and  $\text{Country\_ID} \rightarrow \text{Country\_Name}$  implies  $\text{Athlete\_ID} \rightarrow \text{Country\_Name}$ ), ( $\text{Athlete\_ID} \rightarrow \text{Sport\_ID}$  and  $\text{Sport\_ID} \rightarrow \text{Sport\_Name}$  implies  $\text{Athlete\_ID} \rightarrow \text{Sport\_Name}$ ), and ( $\text{Athlete\_ID} \rightarrow \text{D.O.B}$  and  $\text{D.O.B} \rightarrow \text{Age}$  implies  $\text{Athlete\_ID} \rightarrow \text{Age}$ ). Hence, it is not in 3NF. We removed the attributes `Country_Name`, `Sport_Name`, and `D.O.B`. Therefore, there are no transitive dependencies anymore. Hence, it is in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.
- **Country**(`Country_ID`, `Country_Name`)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.

- There is only one candidate key (Country\_ID) and only a single attribute in the candidate key. Therefore, there are no partial dependencies. Hence, it is already in 2NF.
  - There are no transitive dependencies. Hence, it is already in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.
- **Sport**(Sport\_ID, Sport\_Name)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (Sport\_ID) and only a single attribute in the candidate key. Therefore, there are no partial dependencies. Hence, it is already in 2NF.
  - There are no transitive dependencies. Hence, it is already in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.
- **Event**(Event\_ID, Sport\_ID, Event\_Name, Sport\_Name)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (Event\_ID) and only a single attribute in the candidate key. Therefore, there are no partial dependencies. Hence, it is already in 2NF.
  - There is a transitive dependency ( $\text{Event\_ID} \rightarrow \text{Sport\_ID}$  and  $\text{Sport\_ID} \rightarrow \text{Sport\_Name}$  implies  $\text{Event\_ID} \rightarrow \text{Sport\_Name}$ ). Hence, it is not in 3NF. We removed the attribute Sport\_Name, and therefore, there are no transitive dependencies anymore. Hence, it is in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.

- **Event Leaderboard**(Olympics\_ID, Event\_ID, Sport\_ID, Event\_Name, Sport\_Name, Gold\_Medalist, Silver\_Medalist, Bronze\_Medalist)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (Olympics\_ID, Event\_ID), and it is a composite candidate key. There are partial dependencies (Event\_ID → Sport\_ID), (Event\_ID → Event\_Name), and (Event\_ID → Sport\_Name). Hence, it is not in 2NF. We removed the attributes Sport\_ID, Event\_Name, and Sport\_Name. Therefore, there are no partial dependencies anymore. Hence, it is in 2NF.
  - There are no transitive dependencies. Hence, it is already in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.
- **Sport Leaderboard**(Olympics\_ID, Sport\_ID, Sport\_Name, First\_Rank, Second\_Rank, Third\_Rank)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (Olympics\_ID, Sport\_ID), and it is a composite candidate key. There is a partial dependency (Sport\_ID → Sport\_Name). Hence, it is not in 2NF. We removed the attribute Sport\_Name. Therefore, there are no partial dependencies anymore. Hence, it is in 2NF.
  - There are no transitive dependencies. Hence, it is already in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.
- **Country Leaderboard**(Olympics\_ID, Country\_ID, Country\_Name, Gold\_Medals, Silver\_Medals, Bronze\_Medals, Total\_Medals)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (Olympics\_ID, Country\_ID), and it is a composite candidate key. There is partial dependency ( $\text{Country\_ID} \rightarrow \text{Country\_Name}$ ). Hence, it is not in 2NF. We removed the attribute Country\_Name. Therefore, there are no partial dependencies anymore. Hence, it is in 2NF.
  - There are no transitive dependencies. Hence, it is already in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.
- **Developer**(Developer\_ID, Developer\_Name)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (Developer\_ID) and only a single attribute in the candidate key. Therefore, there are no partial dependencies. Hence, it is already in 2NF.
  - There are no transitive dependencies. Hence, it is already in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.
- **Manager**(Manager\_ID, Manager\_Name)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (Manager\_ID) and only a single attribute in the candidate key. Therefore, there are no partial dependencies. Hence, it is already in 2NF.
  - There are no transitive dependencies. Hence, it is already in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.

- **Owner**(Owner\_ID, Owner\_Name)
  - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF.
  - There is only one candidate key (Owner\_ID) and only a single attribute in the candidate key. Therefore, there are no partial dependencies. Hence, it is already in 2NF.
  - There are no transitive dependencies. Hence, it is already in 3NF.
  - All the functional dependencies have a candidate key on the left side. Therefore, it is in BCNF.

## ❖ Final Relational Model

| Owner |                 |
|-------|-----------------|
| PK    | <u>Owner_ID</u> |
|       | Owner_Name      |

| Developer |                     |
|-----------|---------------------|
| PK        | <u>Developer_ID</u> |
|           | Developer_Name      |

| Manager |                   |
|---------|-------------------|
| PK      | <u>Manager_ID</u> |
|         | Manager_Name      |

| Athlete |                   |
|---------|-------------------|
| PK      | <u>Athlete_ID</u> |
| FK1     | Country_ID        |
| FK2     | Sport_ID          |
|         | Athlete_Name      |
|         | Gender            |
|         | Height            |
|         | Weight            |
|         | Age               |

| Country |                   |
|---------|-------------------|
| PK      | <u>Country_ID</u> |
|         | Country_Name      |

| Country Leaderboard |                    |
|---------------------|--------------------|
| PK,FK1              | <u>Olympics_ID</u> |
| PK,FK2              | <u>Country_ID</u>  |
|                     | Gold_Medals        |
|                     | Silver_Medals      |
|                     | Bronze_Medals      |
|                     | Total_Medals       |

| Olympics |                    |
|----------|--------------------|
| PK       | <u>Olympics_ID</u> |
| FK       | Country_ID         |
|          | City               |
|          | Year               |

| Event Leaderboard |                    |
|-------------------|--------------------|
| PK,FK1            | <u>Olympics_ID</u> |
| PK,FK2            | <u>Event_ID</u>    |
|                   | Gold_Medalist      |
|                   | Silver_Medalist    |
|                   | Bronze_Medalist    |

| Event |                 |
|-------|-----------------|
| PK    | <u>Event_ID</u> |
| FK    | Sport_ID        |
|       | Event_Name      |

| Sport Leaderboard |                    |
|-------------------|--------------------|
| PK,FK1            | <u>Olympics_ID</u> |
| PK,FK2            | <u>Sport_ID</u>    |
|                   | First_Rank         |
|                   | Second_Rank        |
|                   | Third_Rank         |

| Sport |                 |
|-------|-----------------|
| PK    | <u>Sport_ID</u> |
|       | Sport_Name      |

## ❖ Final Relational Schema

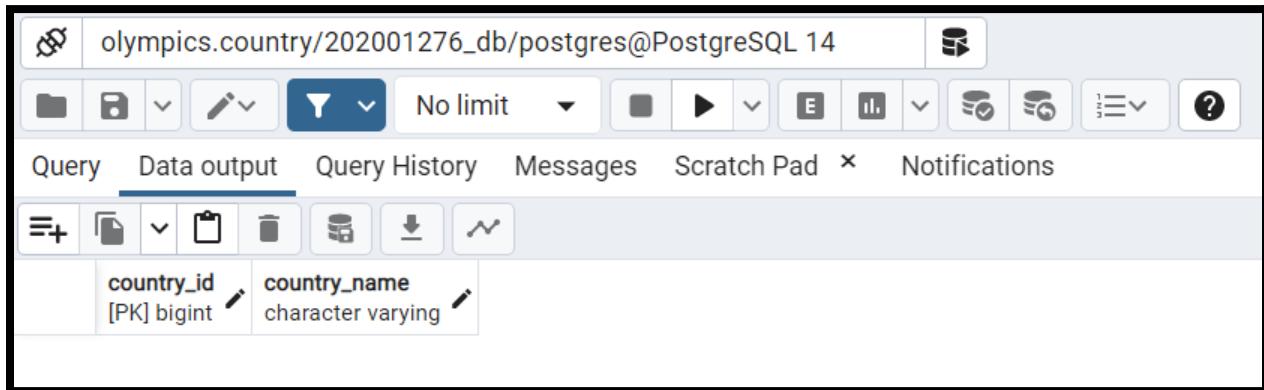
- **Olympics**(Olympics\_ID, Country\_ID, City, Year)
  - FK Country\_ID references **Country**
- **Athlete**(Athlete\_ID, Country\_ID, Sport\_ID, Athlete\_Name, Gender, Height, Weight, Age)
  - FK Country\_ID references **Country**
  - FK Sport\_ID references **Sport**
- **Country**(Country\_ID, Country\_Name)
- **Sport**(Sport\_ID, Sport\_Name)
- **Event**(Event\_ID, Sport\_ID, Event\_Name)
  - FK Sport\_ID references **Sport**
- **Event Leaderboard**(Olympics\_ID, Event\_ID, Gold\_Medalist, Silver\_Medalist, Bronze\_Medalist)
  - FK Olympics\_ID references **Olympics**
  - FK Event\_ID references **Event**
- **Sport Leaderboard**(Olympics\_ID, Sport\_ID, First\_Rank, Second\_Rank, Third\_Rank)
  - FK Olympics\_ID references **Olympics**
  - FK Sport\_ID references **Sport**
- **Country Leaderboard**(Olympics\_ID, Country\_ID, Gold\_Medals, Silver\_Medals, Bronze\_Medals, Total\_Medals)
  - FK Olympics\_ID references **Olympics**
  - FK Country\_ID references **Country**
- **Developer**(Developer\_ID, Developer\_Name)

- **Manager**(Manager\_ID, Manager\_Name)
- **Owner**(Owner\_ID, Owner\_Name)

## Section6: Final DDL Scripts

### 1) Country

```
CREATE TABLE IF NOT EXISTS Country
(
    Country_ID bigint NOT NULL,
    Country_Name character varying,
    PRIMARY KEY (Country_ID)
);
```



### 2) Olympics

```
CREATE TABLE IF NOT EXISTS Olympics
(
    Olympics_ID bigint NOT NULL,
    Country_ID bigint NOT NULL,
    City character varying,
    Year bigint,
    PRIMARY KEY (Olympics_ID),
    FOREIGN KEY (Country_ID) REFERENCES Country MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    NOT VALID
);
```

| olympics_id | [PK] bigint | country_id | bigint | city | character varying | year | bigint |
|-------------|-------------|------------|--------|------|-------------------|------|--------|
|-------------|-------------|------------|--------|------|-------------------|------|--------|

### 3) Sport

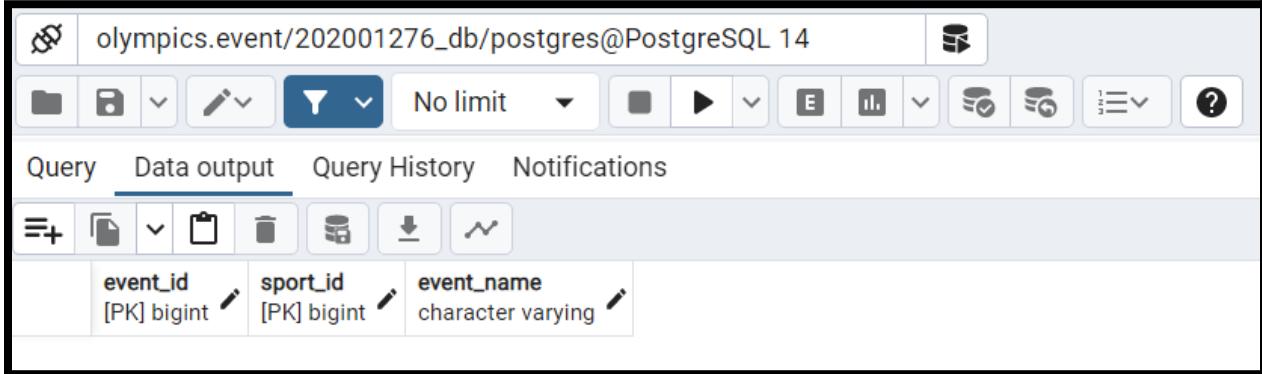
```
CREATE TABLE IF NOT EXISTS Sport
(
Sport_ID bigint NOT NULL,
Sport_Name character varying,
PRIMARY KEY (Sport_ID)
);
```

| sport_id | [PK] bigint | sport_name | character varying |
|----------|-------------|------------|-------------------|
|----------|-------------|------------|-------------------|

### 4) Event

```
CREATE TABLE IF NOT EXISTS Event
(
Event_ID bigint NOT NULL,
Sport_ID bigint NOT NULL,
Event_Name character varying,
PRIMARY KEY (Event_ID),
FOREIGN KEY (Sport_ID) REFERENCES Sport MATCH SIMPLE
ON UPDATE CASCADE
```

```
ON DELETE CASCADE  
NOT VALID  
);
```



## 5) Athlete

```
CREATE TABLE IF NOT EXISTS Athlete  
(  
Athlete_ID bigint NOT NULL,  
Country_ID bigint NOT NULL,  
Sport_ID bigint NOT NULL,  
Athlete_Name character varying,  
Gender character varying,  
Height float,  
Weight bigint,  
Age bigint,  
PRIMARY KEY (Athlete_ID),  
FOREIGN KEY (Country_ID) REFERENCES Country MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE  
NOT VALID,  
FOREIGN KEY (Sport_ID) REFERENCES Sport MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE  
NOT VALID  
);
```

| athlete_id | [PK] bigint | country_id | bigint | sport_id | bigint | athlete_name | character varying | gender | character varying | height | double precision | weight | bigint | age | bigint |
|------------|-------------|------------|--------|----------|--------|--------------|-------------------|--------|-------------------|--------|------------------|--------|--------|-----|--------|
|------------|-------------|------------|--------|----------|--------|--------------|-------------------|--------|-------------------|--------|------------------|--------|--------|-----|--------|

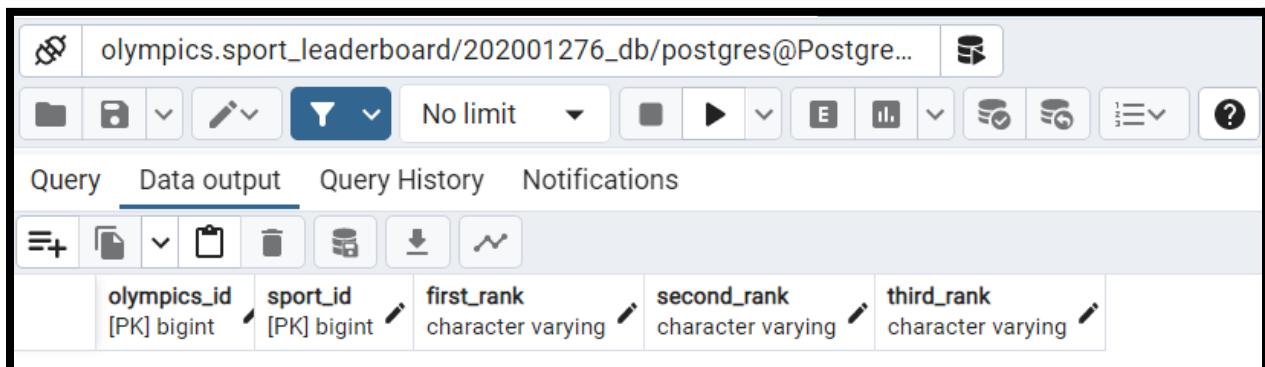
## 6) Event Leaderboard

```
CREATE TABLE IF NOT EXISTS Event_Leaderboard
(
Olympics_ID bigint NOT NULL,
Event_ID bigint NOT NULL,
Gold_Medalist character varying,
Silver_Medalist character varying,
Bronze_Medalist character varying,
PRIMARY KEY (Olympics_ID, Event_ID),
FOREIGN KEY (Olympics_ID) REFERENCES Olympics MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID,
FOREIGN KEY (Event_ID) REFERENCES Event MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID
);
```

| olympics_id | [PK] bigint | event_id | [PK] bigint | sport_id | [PK] bigint | gold_medalist | character varying | silver_medalist | character varying | bronze_medalist | character varying |
|-------------|-------------|----------|-------------|----------|-------------|---------------|-------------------|-----------------|-------------------|-----------------|-------------------|
|-------------|-------------|----------|-------------|----------|-------------|---------------|-------------------|-----------------|-------------------|-----------------|-------------------|

## 7) Sport Leaderboard

```
CREATE TABLE IF NOT EXISTS Sport Leaderboard
(
Olympics_ID bigint NOT NULL,
Sport_ID bigint NOT NULL,
First_Rank character varying,
Second_Rank character varying,
Third_Rank character varying,
PRIMARY KEY (Olympics_ID, Sport_ID),
FOREIGN KEY (Olympics_ID) REFERENCES Olympics MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID,
FOREIGN KEY (Sport_ID) REFERENCES Sport MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID
);
```



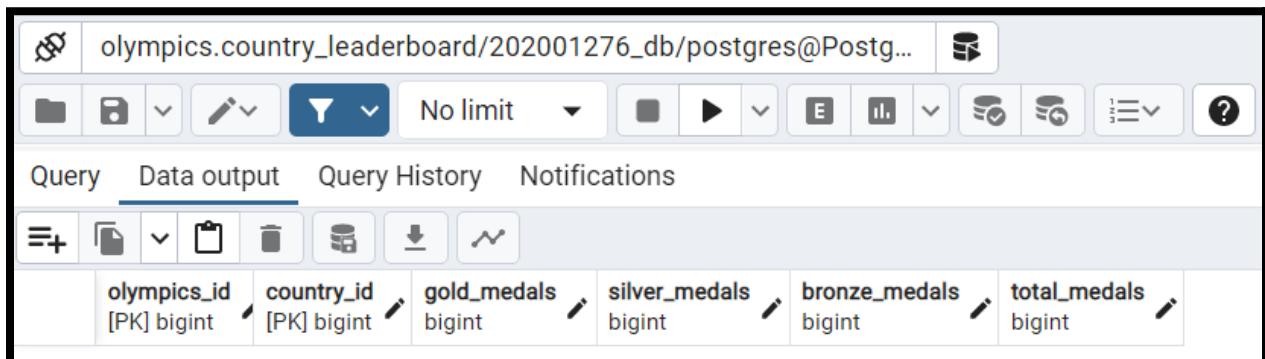
## 8) Country Leaderboard

```
CREATE TABLE IF NOT EXISTS Country Leaderboard
(
Olympics_ID bigint NOT NULL,
Country_ID bigint NOT NULL,
Gold_Medals bigint,
```

```

Silver_Medals bigint,
Bronze_Medals bigint,
Total_Medals bigint,
PRIMARY KEY (Olympics_ID, Country_ID),
FOREIGN KEY (Olympics_ID) REFERENCES Olympics MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID,
FOREIGN KEY (Country_ID) REFERENCES Country MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID
);

```



## 9) Owner

```

CREATE TABLE IF NOT EXISTS Owner
(
  Owner_ID bigint NOT NULL,
  Owner_Name character varying,
  PRIMARY KEY (Owner_ID)
);

```

The screenshot shows the pgAdmin 4 interface with the connection set to 'olympics.owner/202001276\_db/postgres@PostgreSQL 14'. The 'Data output' tab is selected. A table named 'Developer' is being created with the following structure:

|      | owner_id | owner_name        |
|------|----------|-------------------|
| [PK] | bigint   | character varying |

## 10) Developer

```
CREATE TABLE IF NOT EXISTS Developer
(
    Developer_ID bigint NOT NULL,
    Developer_Name character varying,
    PRIMARY KEY (Developer_ID)
);
```

The screenshot shows the pgAdmin 4 interface with the connection set to 'olympics.developer/202001276\_db/postgres@PostgreSQL 14'. The 'Data output' tab is selected. A table named 'Manager' is being created with the following structure:

|      | developer_id | developer_name    |
|------|--------------|-------------------|
| [PK] | bigint       | character varying |

## 11) Manager

```
CREATE TABLE IF NOT EXISTS Manager
(
    Manager_ID bigint NOT NULL,
    Manager_Name character varying,
    PRIMARY KEY (Manager_ID)
);
```

olympics.manager/202001276\_db/postgres@PostgreSQL 14

No limit

Query Data output Query History Notifications

manager\_id [PK] bigint manager\_name character varying

A screenshot of a PostgreSQL database management tool. The top bar shows the connection details: 'olympics.manager/202001276\_db/postgres@PostgreSQL 14'. Below the connection bar are several icons for file operations (New, Open, Save, Print, Find, Copy, Paste, Delete, Import, Export, Help). A dropdown menu for filtering is set to 'No limit'. The main menu includes 'Query' (selected), 'Data output', 'Query History', and 'Notifications'. Below the menu is a toolbar with icons for creating new tables, opening files, saving, printing, deleting, importing, exporting, and help. The central area displays a table structure with two columns: 'manager\_id' (type 'bigint') and 'manager\_name' (type 'character varying'). The 'manager\_id' column is marked as a primary key (PK) with a small icon.

## ❖ Snapshots of tables

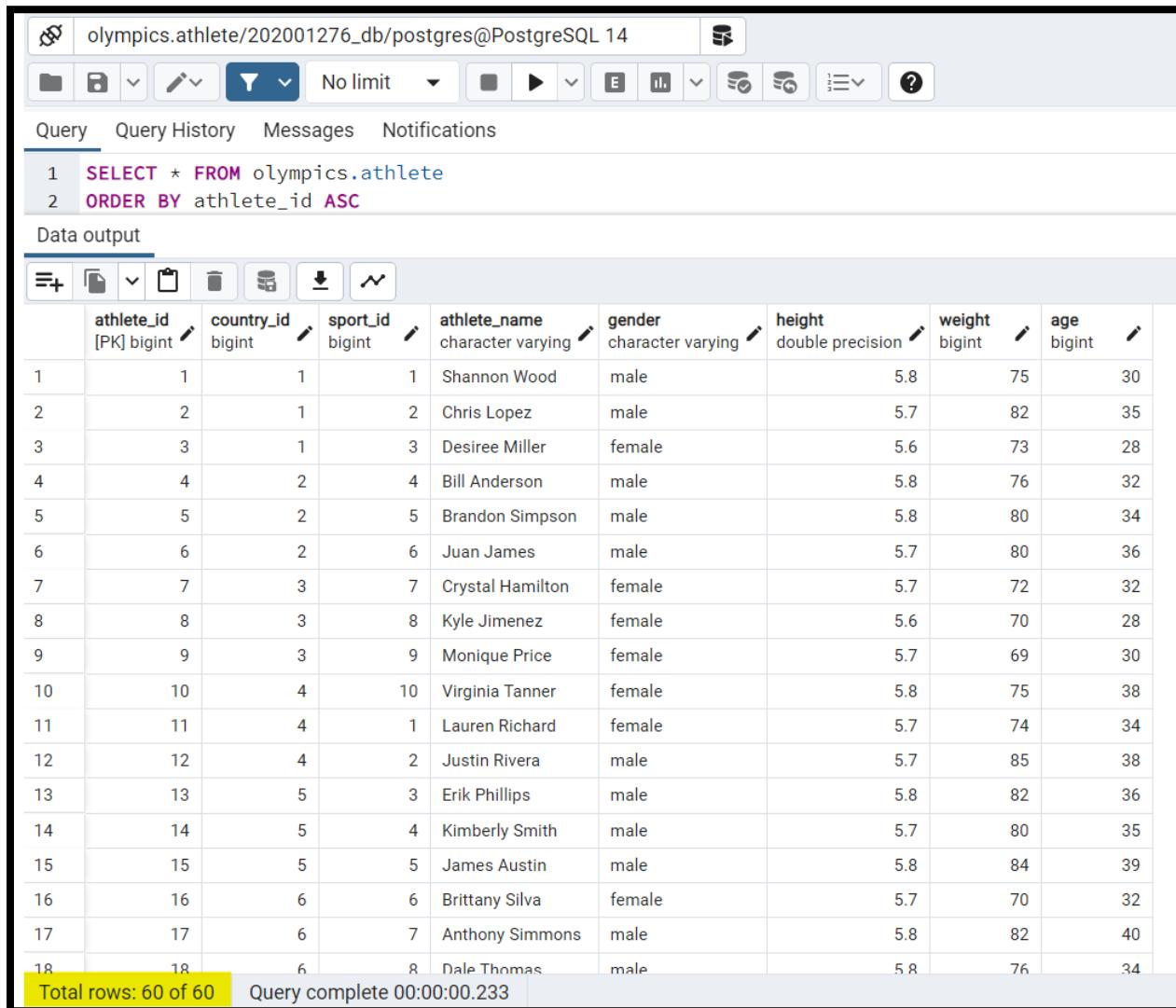
- **Olympics**(Olympics\_ID, Country\_ID, City, Year)

The screenshot shows a PostgreSQL database client interface with the following details:

- Connection:** olympics.olympics/202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations, search, filters, and various database management functions.
- Menu Bar:** Query, Query History, Messages, Notifications.
- Query Editor:** Contains the SQL query:

```
1 SELECT * FROM olympics.olympics
2 ORDER BY olympics_id ASC
```
- Data Output:** A table showing the results of the query. The table has columns: olympics\_id [PK] bigint, country\_id bigint, city character varying, year bigint. The data shows 44 rows of Olympic host cities from 1904 to 2020.
- Table Headers:** olympics\_id [PK] bigint, country\_id bigint, city character varying, year bigint.
- Table Data Rows:** 1 through 44, listing cities like St. Louis, Squaw Valley, Los Angeles, Atlanta, Salt Lake City, Paris, Chamonix, Paris, Cortina d'Ampezzo, Cortina d'Ampezzo, Sapporo, Tokyo, Nagano, Tokyo, Melbourne, Sydney, London, and London.
- Bottom Status:** Total rows: 44 of 44 | Query complete 00:00:00.164

- **Athlete**(Athlete\_ID, Country\_ID, Sport\_ID, Athlete\_Name, Gender, Height, Weight, Age)



The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** olympics.athlete/202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations, search, filter, and various database management functions.
- Menu Bar:** Query, Query History, Messages, Notifications.
- Query Editor:**

```

1  SELECT * FROM olympics.athlete
2  ORDER BY athlete_id ASC

```
- Data Output:** A table showing the results of the query. The table has 18 rows and 9 columns. The columns are:
 

|    | athlete_id<br>[PK] bigint | country_id<br>bigint | sport_id<br>bigint | athlete_name<br>character varying | gender<br>character varying | height<br>double precision | weight<br>bigint | age<br>bigint |
|----|---------------------------|----------------------|--------------------|-----------------------------------|-----------------------------|----------------------------|------------------|---------------|
| 1  | 1                         | 1                    | 1                  | Shannon Wood                      | male                        | 5.8                        | 75               | 30            |
| 2  | 2                         | 1                    | 2                  | Chris Lopez                       | male                        | 5.7                        | 82               | 35            |
| 3  | 3                         | 1                    | 3                  | Desiree Miller                    | female                      | 5.6                        | 73               | 28            |
| 4  | 4                         | 2                    | 4                  | Bill Anderson                     | male                        | 5.8                        | 76               | 32            |
| 5  | 5                         | 2                    | 5                  | Brandon Simpson                   | male                        | 5.8                        | 80               | 34            |
| 6  | 6                         | 2                    | 6                  | Juan James                        | male                        | 5.7                        | 80               | 36            |
| 7  | 7                         | 3                    | 7                  | Crystal Hamilton                  | female                      | 5.7                        | 72               | 32            |
| 8  | 8                         | 3                    | 8                  | Kyle Jimenez                      | female                      | 5.6                        | 70               | 28            |
| 9  | 9                         | 3                    | 9                  | Monique Price                     | female                      | 5.7                        | 69               | 30            |
| 10 | 10                        | 4                    | 10                 | Virginia Tanner                   | female                      | 5.8                        | 75               | 38            |
| 11 | 11                        | 4                    | 1                  | Lauren Richard                    | female                      | 5.7                        | 74               | 34            |
| 12 | 12                        | 4                    | 2                  | Justin Rivera                     | male                        | 5.7                        | 85               | 38            |
| 13 | 13                        | 5                    | 3                  | Erik Phillips                     | male                        | 5.8                        | 82               | 36            |
| 14 | 14                        | 5                    | 4                  | Kimberly Smith                    | male                        | 5.7                        | 80               | 35            |
| 15 | 15                        | 5                    | 5                  | James Austin                      | male                        | 5.8                        | 84               | 39            |
| 16 | 16                        | 6                    | 6                  | Brittany Silva                    | female                      | 5.7                        | 70               | 32            |
| 17 | 17                        | 6                    | 7                  | Anthony Simmons                   | male                        | 5.8                        | 82               | 40            |
| 18 | 18                        | 6                    | 8                  | Dale Thomas                       | male                        | 5.8                        | 76               | 34            |
- Message Bar:** Total rows: 60 of 60 | Query complete 00:00:00.233

- **Country(Country\_ID, Country\_Name)**

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** olympics.country/202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste), search, filter, and various database management functions.
- Menu Bar:** Query, Query History, Messages, Notifications.
- Query Editor:** Displays the SQL query:

```
1 SELECT * FROM olympics.country
2 ORDER BY country_id ASC
```
- Data Output:** A table showing the results of the query. The table has two columns: `country_id` and `country_name`. The data is as follows:

|    | country_id | country_name   |
|----|------------|----------------|
| 1  | 1          | United States  |
| 2  | 2          | France         |
| 3  | 3          | Italy          |
| 4  | 4          | Japan          |
| 5  | 5          | Australia      |
| 6  | 6          | United Kingdom |
| 7  | 7          | Canada         |
| 8  | 8          | Germany        |
| 9  | 9          | China          |
| 10 | 10         | South Korea    |
| 11 | 11         | Russia         |
| 12 | 12         | Greece         |
| 13 | 13         | Norway         |
| 14 | 14         | Austria        |
| 15 | 15         | Switzerland    |
| 16 | 16         | Brazil         |
| 17 | 17         | Spain          |
| 18 | 18         | Yugoslavia     |

**Status Bar:** Total rows: 40 of 40 | Query complete 00:00:00.186

- **Sport(Sport\_ID, Sport\_Name)**

The screenshot shows a PostgreSQL client interface with the following details:

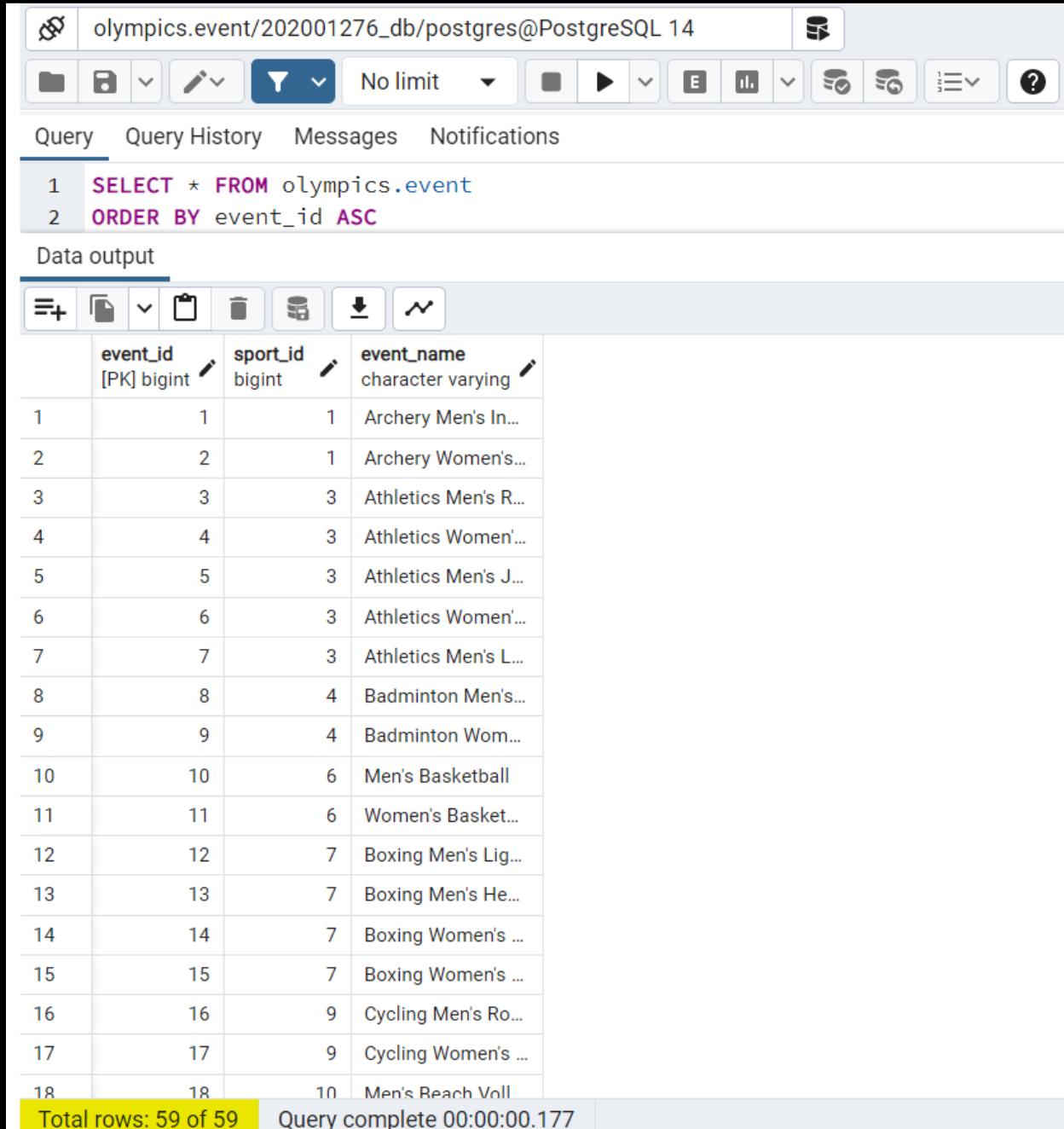
- Connection:** olympics.sport/202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations, search, filters, and various database management functions.
- Menu Bar:** Query, Query History, Messages, Notifications.
- Query Editor:** Displays the SQL query:

```
1 SELECT * FROM olympics.sport
2 ORDER BY sport_id ASC
```
- Data Output:** Shows the results of the query in a table format. The table has two columns: sport\_id (bigint) and sport\_name (character varying). The data consists of 40 rows, numbered 1 to 18, listing various sports.

|    | sport_id<br>[PK] bigint | sport_name<br>character varying |
|----|-------------------------|---------------------------------|
| 1  | 1                       | Archery                         |
| 2  | 2                       | Aquatics                        |
| 3  | 3                       | Athletics                       |
| 4  | 4                       | Badminton                       |
| 5  | 5                       | Baseball                        |
| 6  | 6                       | Basketball                      |
| 7  | 7                       | Boxing                          |
| 8  | 8                       | Canoeing                        |
| 9  | 9                       | Cycling                         |
| 10 | 10                      | Beach Volleyball                |
| 11 | 11                      | Equestrian                      |
| 12 | 12                      | Fencing                         |
| 13 | 13                      | Field Hockey                    |
| 14 | 14                      | Football                        |
| 15 | 15                      | Golf                            |
| 16 | 16                      | Gymnastics                      |
| 17 | 17                      | Handball                        |
| 18 | 18                      | Judo                            |

- Status Bar:** Total rows: 40 of 40 | Query complete 00:00:00.572

- **Event(Event\_ID, Sport\_ID, Event\_Name)**



The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** olympics.event/202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste), search, filter, and various database management functions.
- Menu Bar:** Query, Query History, Messages, Notifications.
- Query Editor:**

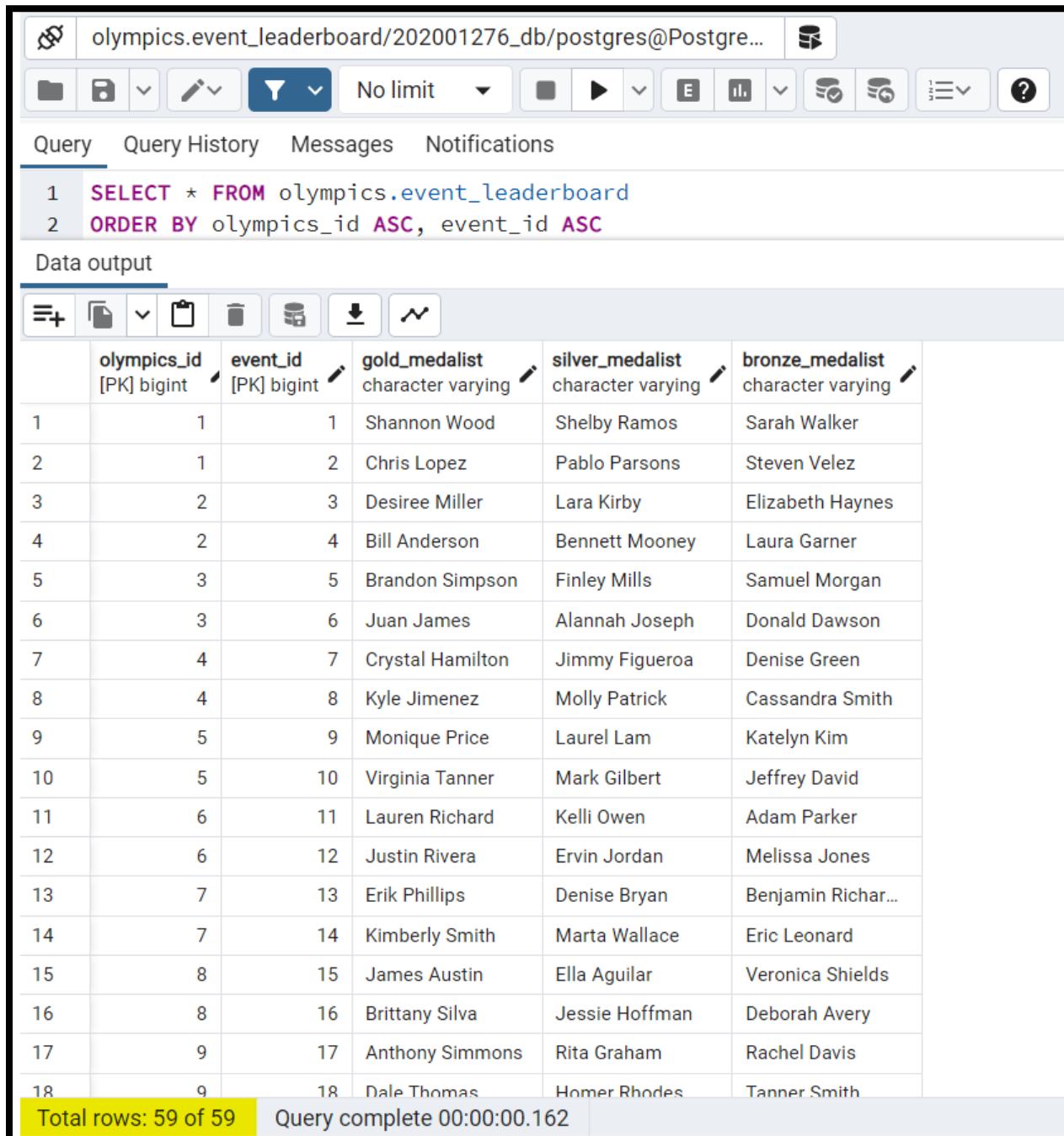
```

1 SELECT * FROM olympics.event
2 ORDER BY event_id ASC

```
- Data Output:** A table showing the results of the query. The columns are:
 

|    | event_id<br>[PK] bigint | sport_id<br>bigint | event_name<br>character varying |
|----|-------------------------|--------------------|---------------------------------|
| 1  | 1                       | 1                  | Archery Men's In...             |
| 2  | 2                       | 1                  | Archery Women's...              |
| 3  | 3                       | 3                  | Athletics Men's R...            |
| 4  | 4                       | 3                  | Athletics Women'...             |
| 5  | 5                       | 3                  | Athletics Men's J...            |
| 6  | 6                       | 3                  | Athletics Women'...             |
| 7  | 7                       | 3                  | Athletics Men's L...            |
| 8  | 8                       | 4                  | Badminton Men's...              |
| 9  | 9                       | 4                  | Badminton Wom...                |
| 10 | 10                      | 6                  | Men's Basketball                |
| 11 | 11                      | 6                  | Women's Basket...               |
| 12 | 12                      | 7                  | Boxing Men's Lig...             |
| 13 | 13                      | 7                  | Boxing Men's He...              |
| 14 | 14                      | 7                  | Boxing Women's ...              |
| 15 | 15                      | 7                  | Boxing Women's ...              |
| 16 | 16                      | 9                  | Cycling Men's Ro...             |
| 17 | 17                      | 9                  | Cycling Women's ...             |
| 18 | 18                      | 10                 | Men's Beach Voll                |
- Status Bar:** Total rows: 59 of 59 | Query complete 00:00:00.177

- **Event Leaderboard**(Olympics\_ID, Event\_ID, Gold\_Medalist, Silver\_Medalist, Bronze\_Medalist)



The screenshot shows a PostgreSQL database client interface with the following details:

- Connection:** olympics.event\_leaderboard/202001276\_db/postgres@Postgre...
- Toolbar:** Includes icons for file operations (New, Open, Save, etc.), search, filter, and various database management functions.
- Menu Bar:** Query, Query History, Messages, Notifications.
- Query Editor:**

```

1 SELECT * FROM olympics.event_leaderboard
2 ORDER BY olympics_id ASC, event_id ASC

```
- Data Output:** A table showing the results of the query. The table has columns: olympics\_id [PK] bigint, event\_id [PK] bigint, gold\_medalist character varying, silver\_medalist character varying, and bronze\_medalist character varying. The data consists of 18 rows, each containing a unique combination of olympics\_id and event\_id, with corresponding gold, silver, and bronze medalists.
- Total Rows:** Total rows: 59 of 59
- Query Status:** Query complete 00:00:00.162

|    | olympics_id<br>[PK] bigint | event_id<br>[PK] bigint | gold_medalist<br>character varying | silver_medalist<br>character varying | bronze_medalist<br>character varying |
|----|----------------------------|-------------------------|------------------------------------|--------------------------------------|--------------------------------------|
| 1  |                            | 1                       | Shannon Wood                       | Shelby Ramos                         | Sarah Walker                         |
| 2  |                            | 1                       | Chris Lopez                        | Pablo Parsons                        | Steven Velez                         |
| 3  |                            | 2                       | Desiree Miller                     | Lara Kirby                           | Elizabeth Haynes                     |
| 4  |                            | 2                       | Bill Anderson                      | Bennett Mooney                       | Laura Garner                         |
| 5  |                            | 3                       | Brandon Simpson                    | Finley Mills                         | Samuel Morgan                        |
| 6  |                            | 3                       | Juan James                         | Alannah Joseph                       | Donald Dawson                        |
| 7  |                            | 4                       | Crystal Hamilton                   | Jimmy Figueroa                       | Denise Green                         |
| 8  |                            | 4                       | Kyle Jimenez                       | Molly Patrick                        | Cassandra Smith                      |
| 9  |                            | 5                       | Monique Price                      | Laurel Lam                           | Katelyn Kim                          |
| 10 |                            | 5                       | Virginia Tanner                    | Mark Gilbert                         | Jeffrey David                        |
| 11 |                            | 6                       | Lauren Richard                     | Kelli Owen                           | Adam Parker                          |
| 12 |                            | 6                       | Justin Rivera                      | Ervin Jordan                         | Melissa Jones                        |
| 13 |                            | 7                       | Erik Phillips                      | Denise Bryan                         | Benjamin Richar...                   |
| 14 |                            | 7                       | Kimberly Smith                     | Marta Wallace                        | Eric Leonard                         |
| 15 |                            | 8                       | James Austin                       | Ella Aguilar                         | Veronica Shields                     |
| 16 |                            | 8                       | Brittany Silva                     | Jessie Hoffman                       | Deborah Avery                        |
| 17 |                            | 9                       | Anthony Simmons                    | Rita Graham                          | Rachel Davis                         |
| 18 |                            | 9                       | Dale Thomas                        | Homer Rhodes                         | Tanner Smith                         |

- **Sport Leaderboard**(Olympics\_ID, Sport\_ID, First\_Rank, Second\_Rank, Third\_Rank)

olympics.sport\_leaderboard/202001276\_db/postgres@Postgre...

No limit

Query History Messages Notifications

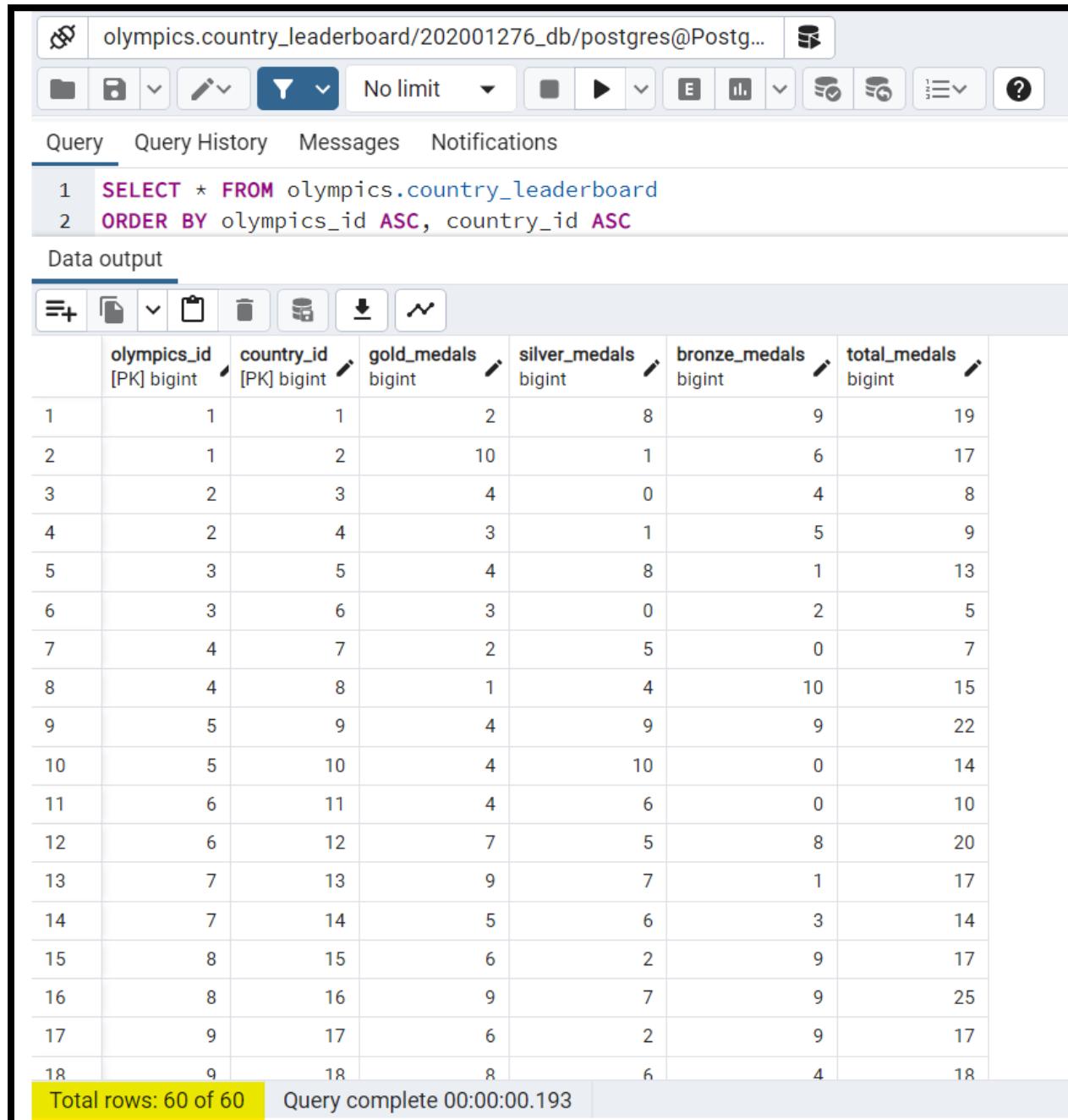
```
1 SELECT * FROM olympics.sport_leaderboard
2 ORDER BY olympics_id ASC, sport_id ASC
```

Data output

|    | olympics_id<br>[PK] bigint | sport_id<br>[PK] bigint | first_rank<br>character varying | second_rank<br>character varying | third_rank<br>character varying |
|----|----------------------------|-------------------------|---------------------------------|----------------------------------|---------------------------------|
| 1  |                            | 1                       | United States                   | Netherlands                      | South Korea                     |
| 2  |                            | 1                       | United States                   | Netherlands                      | South Korea                     |
| 3  |                            | 2                       | France                          | Belgium                          | Russia                          |
| 4  |                            | 2                       | France                          | Belgium                          | Russia                          |
| 5  |                            | 3                       | Italy                           | Sweden                           | Greece                          |
| 6  |                            | 3                       | Italy                           | Sweden                           | Greece                          |
| 7  |                            | 4                       | Japan                           | Georgia                          | Norway                          |
| 8  |                            | 4                       | Japan                           | Georgia                          | Norway                          |
| 9  |                            | 5                       | Australia                       | Oman                             | Austria                         |
| 10 |                            | 5                       | Australia                       | Oman                             | Austria                         |
| 11 |                            | 6                       | United Kingdom                  | Pakistan                         | Switzerland                     |
| 12 |                            | 6                       | United Kingdom                  | Pakistan                         | Switzerland                     |
| 13 |                            | 7                       | Canada                          | Philippines                      | Brazil                          |
| 14 |                            | 7                       | Canada                          | Philippines                      | Brazil                          |
| 15 |                            | 8                       | Germany                         | Poland                           | Spain                           |
| 16 |                            | 8                       | Germany                         | Poland                           | Spain                           |
| 17 |                            | 9                       | China                           | Hong Kong                        | Yugoslavia                      |
| 18 |                            | 9                       | China                           | Hong Kong                        | Yugoslavia                      |

Total rows: 80 of 80      Query complete 00:00:00.490

- **Country Leaderboard**(Olympics\_ID, Country\_ID, Gold\_Medals, Silver\_Medals, Bronze\_Medals, Total\_Medals)



The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** olympics.country\_leaderboard/202001276\_db/postgres@Postg...
- Toolbar:** Includes icons for file operations, search, filters, and various database functions.
- Menu Bar:** Query, Query History, Messages, Notifications.
- Query Editor:**

```

1 SELECT * FROM olympics.country_leaderboard
2 ORDER BY olympics_id ASC, country_id ASC

```
- Data Output:** A table showing the results of the query. The table has 18 rows and 7 columns. The columns are:
 

|    | olympics_id<br>[PK] bigint | country_id<br>[PK] bigint | gold_medals<br>bigint | silver_medals<br>bigint | bronze_medals<br>bigint | total_medals<br>bigint |
|----|----------------------------|---------------------------|-----------------------|-------------------------|-------------------------|------------------------|
| 1  | 1                          | 1                         | 2                     | 8                       | 9                       | 19                     |
| 2  | 1                          | 2                         | 10                    | 1                       | 6                       | 17                     |
| 3  | 2                          | 3                         | 4                     | 0                       | 4                       | 8                      |
| 4  | 2                          | 4                         | 3                     | 1                       | 5                       | 9                      |
| 5  | 3                          | 5                         | 4                     | 8                       | 1                       | 13                     |
| 6  | 3                          | 6                         | 3                     | 0                       | 2                       | 5                      |
| 7  | 4                          | 7                         | 2                     | 5                       | 0                       | 7                      |
| 8  | 4                          | 8                         | 1                     | 4                       | 10                      | 15                     |
| 9  | 5                          | 9                         | 4                     | 9                       | 9                       | 22                     |
| 10 | 5                          | 10                        | 4                     | 10                      | 0                       | 14                     |
| 11 | 6                          | 11                        | 4                     | 6                       | 0                       | 10                     |
| 12 | 6                          | 12                        | 7                     | 5                       | 8                       | 20                     |
| 13 | 7                          | 13                        | 9                     | 7                       | 1                       | 17                     |
| 14 | 7                          | 14                        | 5                     | 6                       | 3                       | 14                     |
| 15 | 8                          | 15                        | 6                     | 2                       | 9                       | 17                     |
| 16 | 8                          | 16                        | 9                     | 7                       | 9                       | 25                     |
| 17 | 9                          | 17                        | 6                     | 2                       | 9                       | 17                     |
| 18 | 9                          | 18                        | 8                     | 6                       | 4                       | 18                     |
- Status Bar:** Total rows: 60 of 60 | Query complete 00:00:00.193

- **Developer(Developer\_ID, Developer\_Name)**

The screenshot shows a PostgreSQL database client interface. The connection is to the 'olympics.developer' database on PostgreSQL 14. The query bar contains the following SQL code:

```
1 SELECT * FROM olympics.developer
2 ORDER BY developer_id ASC
```

The results are displayed in a table titled 'Data output' with two columns: 'developer\_id' and 'developer\_name'. The data consists of 10 rows, each containing a developer's ID and name.

|    | developer_id<br>[PK] bigint | developer_name<br>character varying |
|----|-----------------------------|-------------------------------------|
| 1  | 1                           | Kinley                              |
| 2  | 2                           | Miguel                              |
| 3  | 3                           | Roderick                            |
| 4  | 4                           | Zoe                                 |
| 5  | 5                           | Amiah                               |
| 6  | 6                           | Bryan                               |
| 7  | 7                           | Isai                                |
| 8  | 8                           | Jordan                              |
| 9  | 9                           | Lara                                |
| 10 | 10                          | Sam                                 |

At the bottom, it shows 'Total rows: 10 of 10' and 'Query complete 00:00:00.386'.

- Manager(Manager\_ID, Manager\_Name)

The screenshot shows a PostgreSQL database client interface. The top bar displays the connection information: 'olympics.manager/202001276\_db/postgres@PostgreSQL 14'. Below the connection bar is a toolbar with various icons for database management. The main area has tabs for 'Query', 'Query History', 'Messages', and 'Notifications', with 'Query' currently selected. The query window contains the following SQL code:

```
1 SELECT * FROM olympics.manager
2 ORDER BY manager_id ASC
```

The results are displayed in a table titled 'Data output' with the following schema:

|    | manager_id<br>[PK] bigint | manager_name<br>character varying |
|----|---------------------------|-----------------------------------|
| 1  | 1                         | Mayra                             |
| 2  | 2                         | Janelle                           |
| 3  | 3                         | Naima                             |
| 4  | 4                         | Jack                              |
| 5  | 5                         | Julie                             |
| 6  | 6                         | Sadie                             |
| 7  | 7                         | Mayra                             |
| 8  | 8                         | Jimena                            |
| 9  | 9                         | Jase                              |
| 10 | 10                        | Tyler                             |

At the bottom of the interface, a yellow bar indicates 'Total rows: 10 of 10' and 'Query complete 00:00:00.168'.

- **Owner(Owner\_ID, Owner\_Name)**

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** olympics.owner/202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations, search, filters, and various database management functions.
- Menu Bar:** Query, Query History, Messages, Notifications. "Query" is currently selected.
- Query Editor:** Displays the SQL query:

```
1 SELECT * FROM olympics.owner
2 ORDER BY owner_id ASC
```
- Data Output:** A table showing the results of the query. The table has two rows and two columns:

|   | owner_id<br>[PK] bigint | owner_name<br>character varying |
|---|-------------------------|---------------------------------|
| 1 | 202001268               | Fenil Mansara                   |
| 2 | 202001276               | Om Chalodiya                    |
- Status Bar:** Total rows: 2 of 2 | Query complete 00:00:00.207

## Section7: 20 SQL Queries

1) Display the details of athletes with the maximum height, along with their sport and country

```
select *  
from athlete natural join sport natural join country  
where height = (select max(height)  
                  from athlete  
)  
order by athlete_id asc
```

|    | country_id | sport_id | athlete_id | athlete_name         | gender | height | weight | age | sport_name       | country_name   |
|----|------------|----------|------------|----------------------|--------|--------|--------|-----|------------------|----------------|
| 1  | 1          | 1        | 1          | Shannon Wood         | male   | 5.8    | 75     | 30  | Archery          | United States  |
| 2  | 2          | 4        | 4          | Bill Anderson        | male   | 5.8    | 76     | 32  | Badminton        | France         |
| 3  | 2          | 5        | 5          | Brandon Simpson      | male   | 5.8    | 80     | 34  | Baseball         | France         |
| 4  | 4          | 10       | 10         | Virginia Tanner      | female | 5.8    | 75     | 38  | Beach Volleyball | Japan          |
| 5  | 5          | 3        | 13         | Erik Phillips        | male   | 5.8    | 82     | 36  | Athletics        | Australia      |
| 6  | 5          | 5        | 15         | James Austin         | male   | 5.8    | 84     | 39  | Baseball         | Australia      |
| 7  | 6          | 7        | 17         | Anthony Simmons      | male   | 5.8    | 82     | 40  | Boxing           | United Kingdom |
| 8  | 6          | 8        | 18         | Dale Thomas          | male   | 5.8    | 76     | 34  | Canoeing         | United Kingdom |
| 9  | 7          | 9        | 19         | Debbie Jackson       | male   | 5.8    | 78     | 45  | Cycling          | Canada         |
| 10 | 9          | 5        | 25         | Samuel Morgan        | male   | 5.8    | 80     | 40  | Baseball         | China          |
| 11 | 9          | 6        | 26         | Donald Dawson        | male   | 5.8    | 78     | 46  | Basketball       | China          |
| 12 | 10         | 9        | 29         | Katelyn Kim          | male   | 5.8    | 85     | 35  | Cycling          | South Korea    |
| 13 | 10         | 10       | 30         | Jeffrey David        | male   | 5.8    | 80     | 38  | Beach Volleyball | South Korea    |
| 14 | 11         | 1        | 31         | Adam Parker          | male   | 5.8    | 79     | 34  | Archery          | Russia         |
| 15 | 11         | 3        | 33         | Benjamin Richards... | male   | 5.8    | 75     | 42  | Athletics        | Russia         |
| 16 | 13         | 7        | 37         | Rachel Davis         | male   | 5.8    | 78     | 38  | Boxing           | Norway         |
| 17 | 13         | 8        | 38         | Tanner Smith         | male   | 5.8    | 75     | 32  | Canoeing         | Norway         |
| 18 | 14         | 1        | 41         | Shelby Ramos         | male   | 5.8    | 85     | 32  | Archery          | Austria        |
| 19 | 14         | 2        | 42         | Pablo Parsons        | male   | 5.8    | 82     | 34  | Aquatics         | Austria        |
| 20 | 15         | 5        | 45         | Finley Mills         | male   | 5.8    | 70     | 28  | Baseball         | Switzerland    |
| 21 | 16         | 6        | 46         | Alannah Joseph       | female | 5.8    | 82     | 30  | Basketball       | Brazil         |

**2) Display sport\_id and sport\_name of the sports whose name begins with the character 'B'**

```
select sport_id, sport_name  
from sport  
where sport_name like 'B%'
```

The screenshot shows the pgAdmin 4 interface with a query results window. The title bar indicates the connection is to '202001276\_db/postgres@PostgreSQL 14'. The toolbar has various icons for file operations, search, and navigation. Below the toolbar, tabs for 'Query', 'Data output', 'Query History', 'Messages', and 'Notifications' are present, with 'Data output' being the active tab. The main area displays a table with the following data:

|   | sport_id<br>[PK] bigint | sport_name<br>character varying |
|---|-------------------------|---------------------------------|
| 1 | 4                       | Badminton                       |
| 2 | 5                       | Baseball                        |
| 3 | 6                       | Basketball                      |
| 4 | 7                       | Boxing                          |
| 5 | 10                      | Beach Volleyball                |

At the bottom of the results window, a yellow bar displays the message 'Total rows: 5 of 5' and 'Query complete 00:00:00.113'.

**3) Display the details of the athletes whose age lies in the range 30-35**

select \*

```
from athlete natural join sport natural join country  
where age between 30 and 35  
order by athlete_id asc
```

The screenshot shows a pgAdmin interface with a query results window. The query has been executed successfully, displaying 28 rows of data. The columns are labeled: country\_id, sport\_id, athlete\_id, athlete\_name, gender, height, weight, age, sport\_name, and country\_name. The data includes various athletes from different countries and sports, such as Archery, Aquatics, Badminton, Baseball, Boxing, Cycling, and more.

|    | country_id | sport_id | athlete_id | athlete_name     | gender | height | weight | age | sport_name       | country_name   |
|----|------------|----------|------------|------------------|--------|--------|--------|-----|------------------|----------------|
| 1  |            | 1        | 1          | Shannon Wood     | male   | 5.8    | 75     | 30  | Archery          | United States  |
| 2  |            | 1        | 2          | Chris Lopez      | male   | 5.7    | 82     | 35  | Aquatics         | United States  |
| 3  | 2          | 4        | 4          | Bill Anderson    | male   | 5.8    | 76     | 32  | Badminton        | France         |
| 4  | 2          | 5        | 5          | Brandon Simpson  | male   | 5.8    | 80     | 34  | Baseball         | France         |
| 5  | 3          | 7        | 7          | Crystal Hamilton | female | 5.7    | 72     | 32  | Boxing           | Italy          |
| 6  | 3          | 9        | 9          | Monique Price    | female | 5.7    | 69     | 30  | Cycling          | Italy          |
| 7  | 4          | 1        | 11         | Lauren Richard   | female | 5.7    | 74     | 34  | Archery          | Japan          |
| 8  | 5          | 4        | 14         | Kimberly Smith   | male   | 5.7    | 80     | 35  | Badminton        | Australia      |
| 9  | 6          | 6        | 16         | Brittany Silva   | female | 5.7    | 70     | 32  | Basketball       | United Kingdom |
| 10 | 6          | 8        | 18         | Dale Thomas      | male   | 5.8    | 76     | 34  | Canoeing         | United Kingdom |
| 11 | 8          | 3        | 23         | Elizabeth Haynes | female | 5.7    | 70     | 34  | Athletics        | Germany        |
| 12 | 10         | 8        | 28         | Cassandra Smith  | male   | 5.7    | 75     | 32  | Canoeing         | South Korea    |
| 13 | 10         | 9        | 29         | Katelyn Kim      | male   | 5.8    | 85     | 35  | Cycling          | South Korea    |
| 14 | 11         | 1        | 31         | Adam Parker      | male   | 5.8    | 79     | 34  | Archery          | Russia         |
| 15 | 11         | 2        | 32         | Melissa Jones    | female | 5.6    | 82     | 34  | Aquatics         | Russia         |
| 16 | 12         | 4        | 34         | Eric Leonard     | male   | 5.7    | 76     | 33  | Badminton        | Greece         |
| 17 | 12         | 5        | 35         | Veronica Shields | female | 5.7    | 76     | 33  | Baseball         | Greece         |
| 18 | 13         | 8        | 38         | Tanner Smith     | male   | 5.8    | 75     | 32  | Canoeing         | Norway         |
| 19 | 13         | 9        | 39         | Melissa Miller   | female | 5.7    | 72     | 32  | Cycling          | Norway         |
| 20 | 14         | 10       | 40         | Glenn Bird       | female | 5.7    | 75     | 30  | Beach Volleyball | Austria        |
| 21 | 14         | 1        | 41         | Shelby Ramos     | male   | 5.8    | 85     | 32  | Archery          | Austria        |

**4) Display the country which has a record of winning the highest number of medals in single Olympics**

```
select distinct country_id, country_name, gold_medals, silver_medals,  
bronze_medals, total_medals  
from country natural join country_leaderboard  
where total_medals = (select max(total_medals)  
                      from country_leaderboard  
                     )
```

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** 202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations, search, filters, and various database management functions.
- Menu Bar:** Query, Data output, Query History, Messages, Notifications.
- Table View:** A grid showing the results of the executed query. The columns are: country\_id, country\_name, gold\_medals, silver\_medals, bronze\_medals, and total\_medals. The data row is: 1, Brazil, 9, 7, 9, 25.
- Status Bar:** Total rows: 1 of 1 | Query complete 00:00:00.068

5) Display the events associated with the sport\_id = 40

```
select event_name  
from event  
where sport_id = 40
```

The screenshot shows the pgAdmin 4 interface with a query results window. The title bar indicates the connection is to '202001276\_db/postgres@PostgreSQL 14'. The toolbar has various icons for database management. Below the toolbar, tabs for 'Query', 'Data output' (which is selected), 'Query History', 'Messages', and 'Notifications' are visible. The main area displays a table with one column, 'event\_name', containing six rows of swimming event names. At the bottom, a status bar shows 'Total rows: 6 of 6' and 'Query complete 00:00:00.085'.

|   | event_name                   |
|---|------------------------------|
| 1 | Swimming Men's Freestyle     |
| 2 | Swimming Women's Freestyle   |
| 3 | Swimming Men's Butterfly     |
| 4 | Swimming Women's Butterfly   |
| 5 | Swimming Men's Backstroke    |
| 6 | Swimming Women's Backstro... |

Total rows: 6 of 6    Query complete 00:00:00.085

6) Display all the sports, which are exactly six characters long

```
select sport_name  
from sport  
where sport_name like '_____'
```

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** 202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste), search, filter, and various database management functions.
- Menu Bar:** Query, Data output, Query History, Messages, Notifications.
- Data View:** A table with one column "sport\_name" and six rows, each numbered 1 through 6. The table has a header row: "sport\_name" and "character varying".

|   | sport_name | character varying |
|---|------------|-------------------|
| 1 | Boxing     |                   |
| 2 | Karate     |                   |
| 3 | Rowing     |                   |
| 4 | Tennis     |                   |
| 5 | Futsal     |                   |
| 6 | Hockey     |                   |
- Status Bar:** Total rows: 6 of 6 | Query complete 00:00:00.088

7) Display the total no. of male and female athletes

```
select gender, count(*) as "count"  
from athlete  
group by gender
```

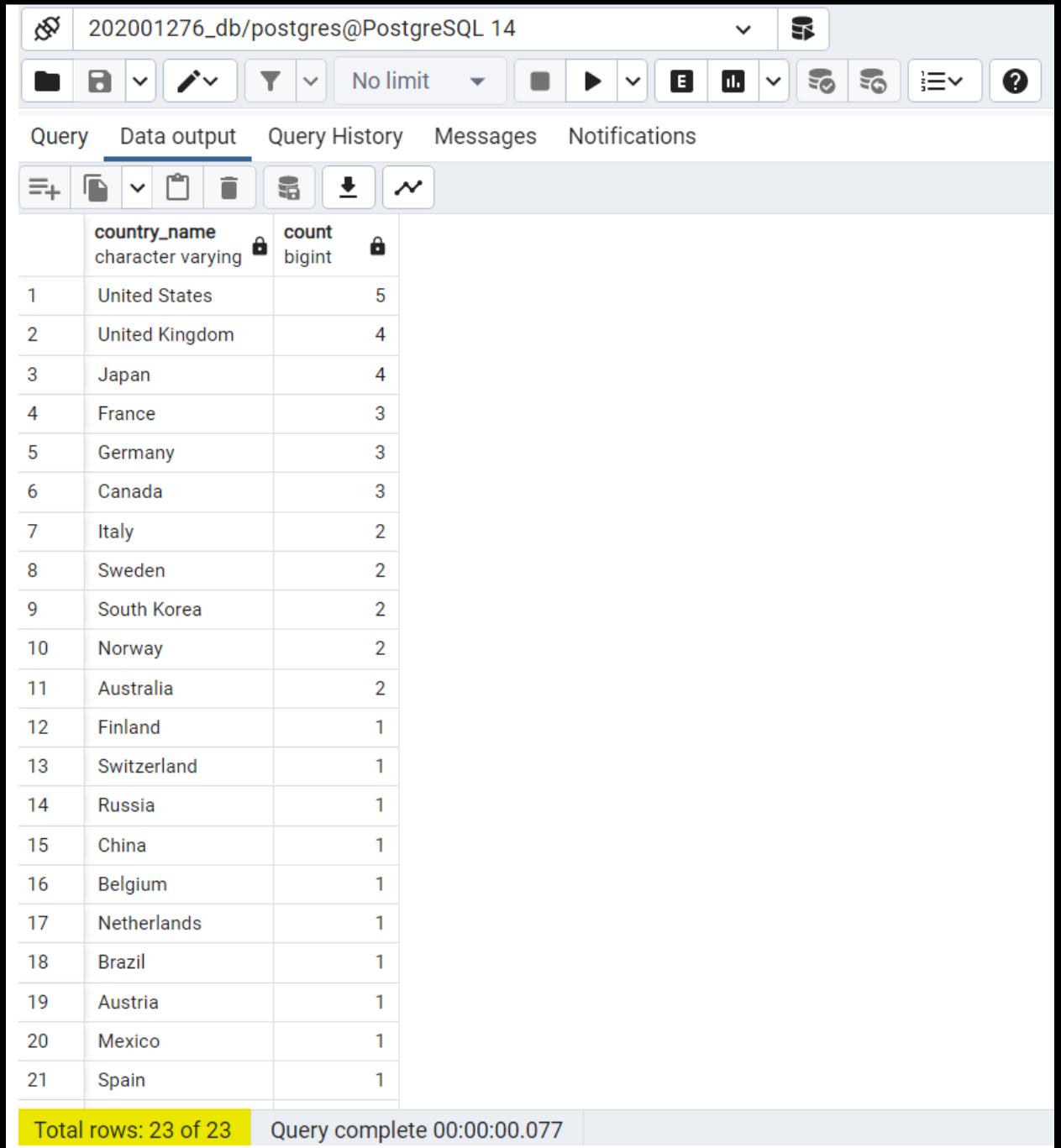
The screenshot shows the pgAdmin 4 interface with a query results window. The title bar indicates the connection is to '202001276\_db/postgres@PostgreSQL 14'. The toolbar has various icons for file operations, search, and navigation. Below the toolbar, tabs for 'Query', 'Data output', 'Query History', 'Messages', and 'Notifications' are present, with 'Data output' being the active tab. The main area displays a table with two rows:

|   | gender            | count  |
|---|-------------------|--------|
|   | character varying | bigint |
| 1 | female            | 23     |
| 2 | male              | 37     |

At the bottom of the results window, a status bar shows 'Total rows: 2 of 2' and 'Query complete 00:00:00.067'.

**8) Display the count of no. of Olympics hosted by each country in descending order**

```
select country_name, count(*) as count
from olympics natural join country
group by country_name
order by count desc
```



The screenshot shows a PostgreSQL client interface with the following details:

- Connection: 202001276\_db/postgres@PostgreSQL 14
- Toolbar: Includes icons for file operations, search, filters, and various database management functions.
- Menu Bar: Query, Data output, Query History, Messages, Notifications.
- Table View: A grid showing the results of the query. The columns are labeled "country\_name" and "count". The data is sorted by "count" in descending order, with the United States at the top (5 hostings).

|    | country_name   | count |
|----|----------------|-------|
| 1  | United States  | 5     |
| 2  | United Kingdom | 4     |
| 3  | Japan          | 4     |
| 4  | France         | 3     |
| 5  | Germany        | 3     |
| 6  | Canada         | 3     |
| 7  | Italy          | 2     |
| 8  | Sweden         | 2     |
| 9  | South Korea    | 2     |
| 10 | Norway         | 2     |
| 11 | Australia      | 2     |
| 12 | Finland        | 1     |
| 13 | Switzerland    | 1     |
| 14 | Russia         | 1     |
| 15 | China          | 1     |
| 16 | Belgium        | 1     |
| 17 | Netherlands    | 1     |
| 18 | Brazil         | 1     |
| 19 | Austria        | 1     |
| 20 | Mexico         | 1     |
| 21 | Spain          | 1     |

Total rows: 23 of 23    Query complete 00:00:00.077

g) Display the details of the athletes who play Basketball sport

```
select athlete_id, athlete_name, gender, height, weight, age,  
country_name  
from athlete natural join country  
where athlete_id in (select athlete_id  
                     from athlete natural join sport  
                     where sport_name = 'Basketball'  
)
```

The screenshot shows the pgAdmin 4 interface with a query results window. The title bar indicates the connection is to '202001276\_db/postgres@PostgreSQL 14'. The toolbar has various icons for file operations, search, and navigation. Below the toolbar, tabs for 'Query', 'Data output', 'Query History', 'Messages', and 'Notifications' are visible, with 'Data output' being the active tab. The main area displays a table with the following data:

|   | athlete_id | athlete_name   | gender | height | weight | age | country_name   |
|---|------------|----------------|--------|--------|--------|-----|----------------|
| 1 | 16         | Brittany Silva | female | 5.7    | 70     | 32  | United Kingdom |
| 2 | 6          | Juan James     | male   | 5.7    | 80     | 36  | France         |
| 3 | 46         | Alannah Joseph | female | 5.8    | 82     | 30  | Brazil         |
| 4 | 26         | Donald Dawson  | male   | 5.8    | 78     | 46  | China          |
| 5 | 36         | Deborah Avery  | female | 5.7    | 70     | 40  | Greece         |
| 6 | 56         | Jessie Hoffman | male   | 5.8    | 75     | 40  | Mexico         |

Total rows: 6 of 6    Query complete 00:00:00.067

**10) Display the count of the no. of events of each sport**

```
select sport_name, count(*) as "no. of events"  
from sport natural join event  
group by sport_name
```

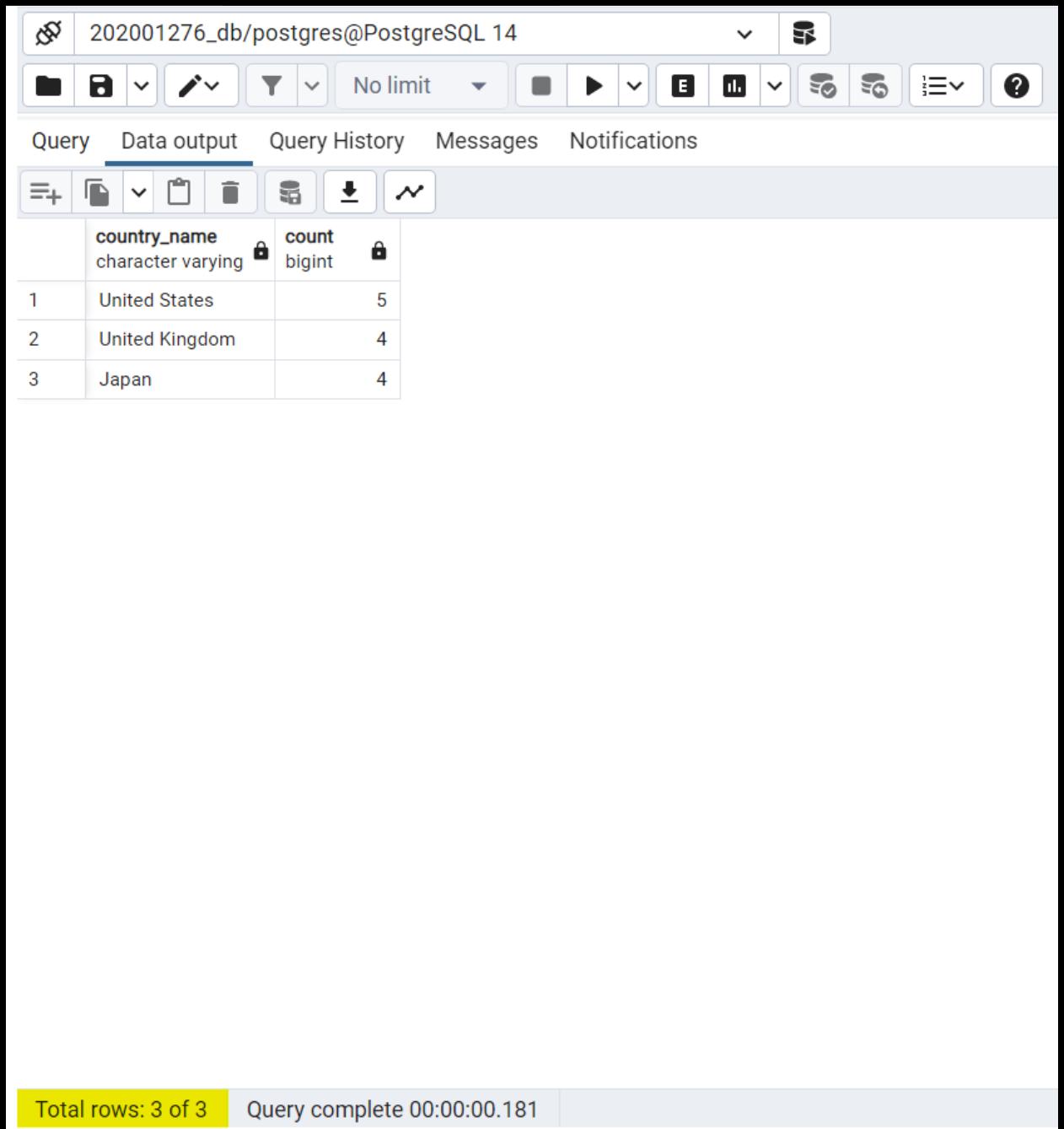
The screenshot shows the pgAdmin 4 interface with a query results window. The title bar indicates the connection is to '202001276\_db/postgres@PostgreSQL 14'. The toolbar has various icons for database management. Below the toolbar, there are tabs: 'Query' (selected), 'Data output' (underlined), 'Query History', 'Messages', and 'Notifications'. The main area displays a table with the following data:

|    | sport_name       | no. of events |
|----|------------------|---------------|
| 1  | Gymnastics       | 2             |
| 2  | Wrestling        | 2             |
| 3  | Judo             | 4             |
| 4  | Athletics        | 5             |
| 5  | Table Tennis     | 2             |
| 6  | Golf             | 2             |
| 7  | Weightlifting    | 2             |
| 8  | Handball         | 2             |
| 9  | Swimming         | 6             |
| 10 | Basketball       | 2             |
| 11 | Boxing           | 4             |
| 12 | Cycling          | 2             |
| 13 | Taekwondo        | 4             |
| 14 | Shooting         | 4             |
| 15 | Football         | 2             |
| 16 | Tennis           | 2             |
| 17 | Hockey           | 2             |
| 18 | Badminton        | 2             |
| 19 | Volleyball       | 2             |
| 20 | Karate           | 2             |
| 21 | Beach Volleyball | 2             |

Total rows: 22 of 22      Query complete 00:00:00.049

**11) Display the top three countries which have hosted the Olympics for the maximum no. of times**

```
select country_name, count(*) as "count"
from olympics natural join country
group by country_name
order by count desc
limit 3
```



The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** 202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste, Find, Filter), a search bar, and various database management tools.
- Menu Bar:** Query, Data output, Query History, Messages, Notifications.
- Table View:** Displays the results of the SQL query. The table has two columns: `country_name` (character varying) and `count` (bigint). The data is as follows:

|   | country_name   | count |
|---|----------------|-------|
| 1 | United States  | 5     |
| 2 | United Kingdom | 4     |
| 3 | Japan          | 4     |

- Status Bar:** Total rows: 3 of 3 | Query complete 00:00:00.181

**12) Display the no. of medals won by Germany in Tokyo 2020 Olympics**

```
select country_name, gold_medals, silver_medals, bronze_medals,  
total_medals  
from country natural join country_leaderboard  
where country_name = 'Germany'  
and olympics_id in (select olympics_id  
                     from olympics  
                     where city = 'Tokyo' and year = 2020  
                )
```

The screenshot shows the pgAdmin 4 interface with a query results window. The connection is set to '202001276\_db/postgres@PostgreSQL 14'. The 'Data output' tab is selected. The results table has columns: country\_name, gold\_medals, silver\_medals, bronze\_medals, and total\_medals. A single row is returned for Germany, with values: 10, 8, 2, and 20 respectively.

|   | country_name | gold_medals | silver_medals | bronze_medals | total_medals |
|---|--------------|-------------|---------------|---------------|--------------|
| 1 | Germany      | 10          | 8             | 2             | 20           |

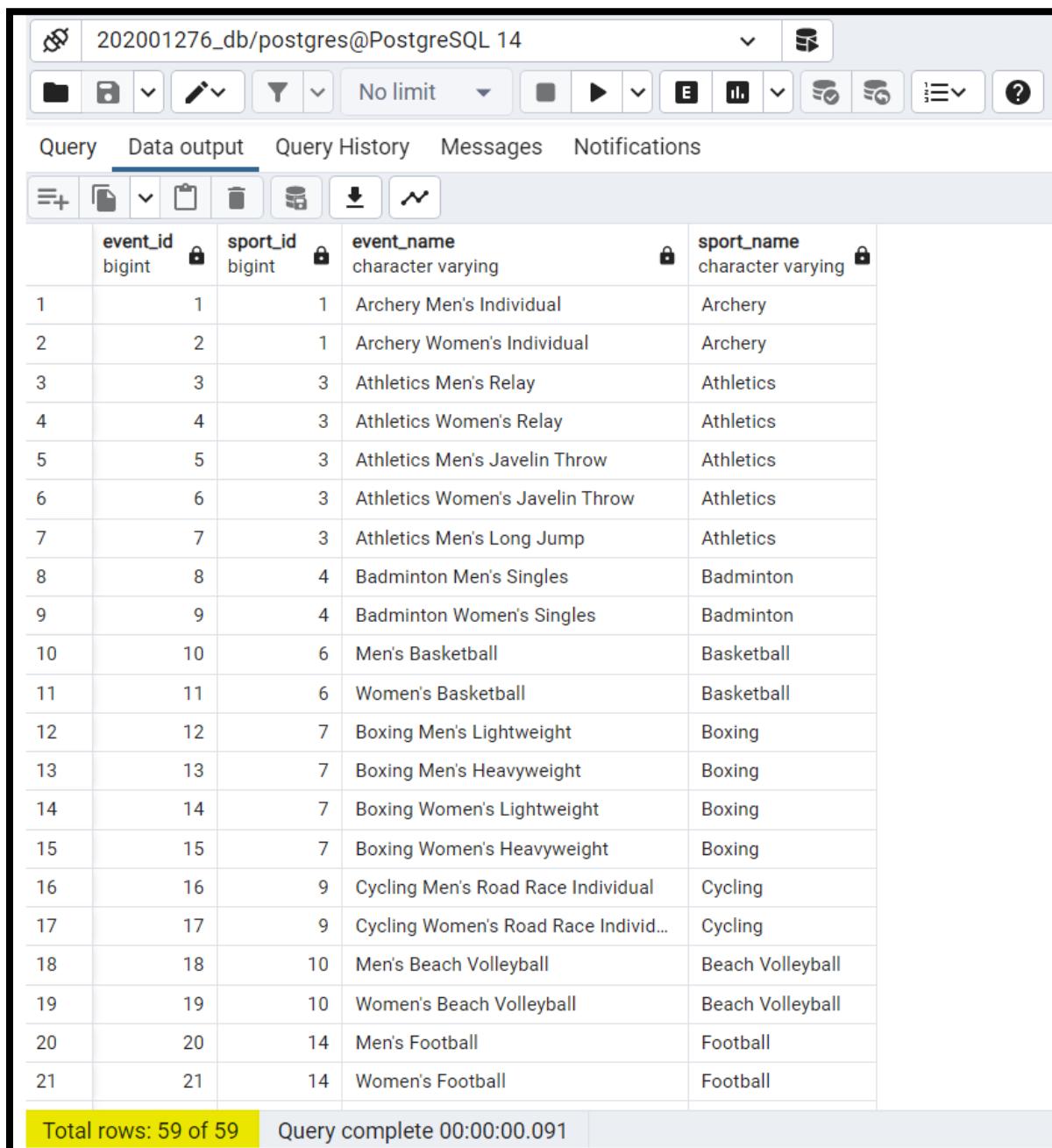
Total rows: 1 of 1    Query complete 00:00:00.066

**13) Display the details of all the events along with the sport associated with them**

```
select event_id, sport.sport_id, event_name, sport_name  
from event join sport  
on event.sport_id = sport.sport_id
```

(or)

```
select event_id, sport_id, event_name, sport_name  
from event natural join sport
```

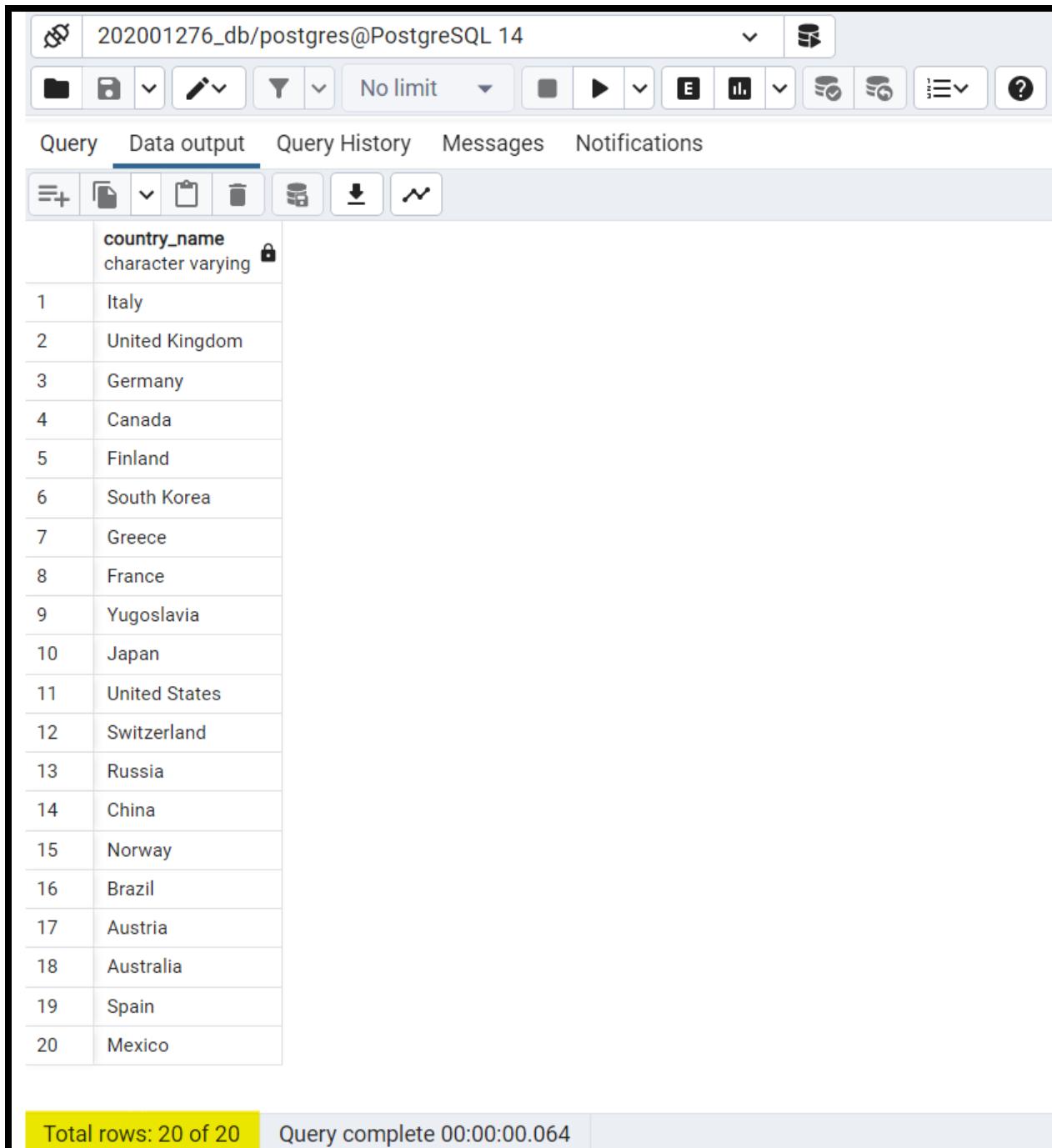


|    | event_id<br>bigint | sport_id<br>bigint | event_name<br>character varying      | sport_name<br>character varying |
|----|--------------------|--------------------|--------------------------------------|---------------------------------|
| 1  | 1                  | 1                  | Archery Men's Individual             | Archery                         |
| 2  | 2                  | 1                  | Archery Women's Individual           | Archery                         |
| 3  | 3                  | 3                  | Athletics Men's Relay                | Athletics                       |
| 4  | 4                  | 3                  | Athletics Women's Relay              | Athletics                       |
| 5  | 5                  | 3                  | Athletics Men's Javelin Throw        | Athletics                       |
| 6  | 6                  | 3                  | Athletics Women's Javelin Throw      | Athletics                       |
| 7  | 7                  | 3                  | Athletics Men's Long Jump            | Athletics                       |
| 8  | 8                  | 4                  | Badminton Men's Singles              | Badminton                       |
| 9  | 9                  | 4                  | Badminton Women's Singles            | Badminton                       |
| 10 | 10                 | 6                  | Men's Basketball                     | Basketball                      |
| 11 | 11                 | 6                  | Women's Basketball                   | Basketball                      |
| 12 | 12                 | 7                  | Boxing Men's Lightweight             | Boxing                          |
| 13 | 13                 | 7                  | Boxing Men's Heavyweight             | Boxing                          |
| 14 | 14                 | 7                  | Boxing Women's Lightweight           | Boxing                          |
| 15 | 15                 | 7                  | Boxing Women's Heavyweight           | Boxing                          |
| 16 | 16                 | 9                  | Cycling Men's Road Race Individual   | Cycling                         |
| 17 | 17                 | 9                  | Cycling Women's Road Race Individ... | Cycling                         |
| 18 | 18                 | 10                 | Men's Beach Volleyball               | Beach Volleyball                |
| 19 | 19                 | 10                 | Women's Beach Volleyball             | Beach Volleyball                |
| 20 | 20                 | 14                 | Men's Football                       | Football                        |
| 21 | 21                 | 14                 | Women's Football                     | Football                        |

Total rows: 59 of 59    Query complete 00:00:00.091

**14) Display the name of countries having exactly three athletes**

```
select country_name  
from country natural join athlete  
group by country_name  
having count(athlete_id) = 3
```

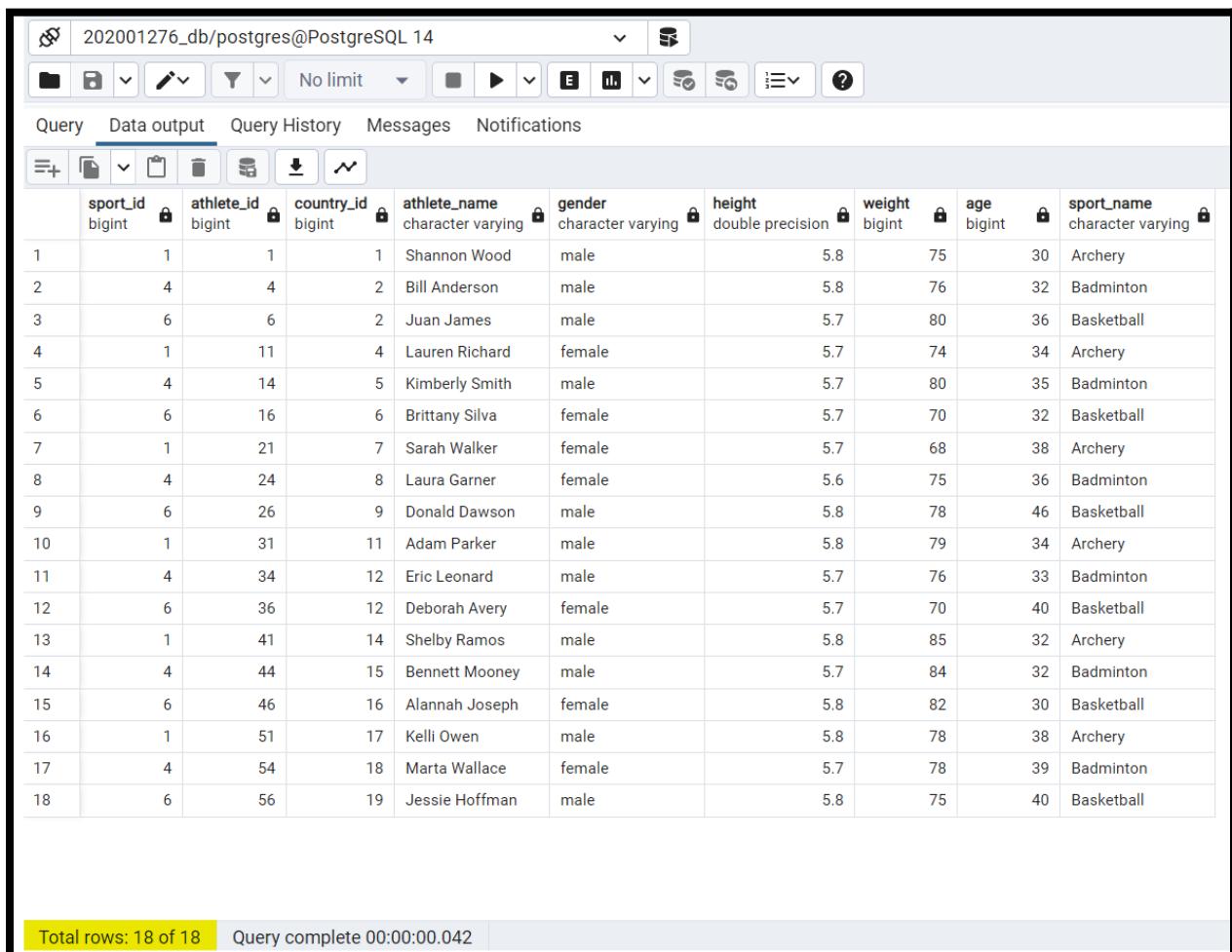


The screenshot shows the pgAdmin 4 interface with a query results window. The title bar indicates the connection is to '202001276\_db/postgres@PostgreSQL 14'. The toolbar has various icons for database management. Below the toolbar, tabs for 'Query', 'Data output' (which is selected), 'Query History', 'Messages', and 'Notifications' are visible. The main area displays a table with 20 rows, each containing a number (1 to 20) and a country name. The column header is 'country\_name character varying'. At the bottom of the results window, a status bar shows 'Total rows: 20 of 20' and 'Query complete 00:00:00.064'.

|    | country_name<br>character varying |
|----|-----------------------------------|
| 1  | Italy                             |
| 2  | United Kingdom                    |
| 3  | Germany                           |
| 4  | Canada                            |
| 5  | Finland                           |
| 6  | South Korea                       |
| 7  | Greece                            |
| 8  | France                            |
| 9  | Yugoslavia                        |
| 10 | Japan                             |
| 11 | United States                     |
| 12 | Switzerland                       |
| 13 | Russia                            |
| 14 | China                             |
| 15 | Norway                            |
| 16 | Brazil                            |
| 17 | Austria                           |
| 18 | Australia                         |
| 19 | Spain                             |
| 20 | Mexico                            |

**15) Display the details of the athletes who play Archery, Badminton, or Basketball**

```
select *
from athlete natural join sport
where sport_name in (select sport_name
                      from sport
                      where sport_name = 'Archery' or sport_name =
                      'Badminton' or sport_name = 'Basketball'
                     )
```



The screenshot shows the pgAdmin 4 interface with a query editor window. The query is executed against the '202001276\_db/postgres@PostgreSQL 14' database. The results are shown in a table with the following columns: sport\_id, athlete\_id, country\_id, athlete\_name, gender, height, weight, age, and sport\_name. The data consists of 18 rows, each representing an athlete's information and their assigned sport.

|    | sport_id<br>bigint | athlete_id<br>bigint | country_id<br>bigint | athlete_name<br>character varying | gender<br>character varying | height<br>double precision | weight<br>bigint | age<br>bigint | sport_name<br>character varying |
|----|--------------------|----------------------|----------------------|-----------------------------------|-----------------------------|----------------------------|------------------|---------------|---------------------------------|
| 1  | 1                  | 1                    | 1                    | Shannon Wood                      | male                        | 5.8                        | 75               | 30            | Archery                         |
| 2  | 4                  | 4                    | 2                    | Bill Anderson                     | male                        | 5.8                        | 76               | 32            | Badminton                       |
| 3  | 6                  | 6                    | 2                    | Juan James                        | male                        | 5.7                        | 80               | 36            | Basketball                      |
| 4  | 1                  | 11                   | 4                    | Lauren Richard                    | female                      | 5.7                        | 74               | 34            | Archery                         |
| 5  | 4                  | 14                   | 5                    | Kimberly Smith                    | male                        | 5.7                        | 80               | 35            | Badminton                       |
| 6  | 6                  | 16                   | 6                    | Brittany Silva                    | female                      | 5.7                        | 70               | 32            | Basketball                      |
| 7  | 1                  | 21                   | 7                    | Sarah Walker                      | female                      | 5.7                        | 68               | 38            | Archery                         |
| 8  | 4                  | 24                   | 8                    | Laura Garner                      | female                      | 5.6                        | 75               | 36            | Badminton                       |
| 9  | 6                  | 26                   | 9                    | Donald Dawson                     | male                        | 5.8                        | 78               | 46            | Basketball                      |
| 10 | 1                  | 31                   | 11                   | Adam Parker                       | male                        | 5.8                        | 79               | 34            | Archery                         |
| 11 | 4                  | 34                   | 12                   | Eric Leonard                      | male                        | 5.7                        | 76               | 33            | Badminton                       |
| 12 | 6                  | 36                   | 12                   | Deborah Avery                     | female                      | 5.7                        | 70               | 40            | Basketball                      |
| 13 | 1                  | 41                   | 14                   | Shelby Ramos                      | male                        | 5.8                        | 85               | 32            | Archery                         |
| 14 | 4                  | 44                   | 15                   | Bennett Mooney                    | male                        | 5.7                        | 84               | 32            | Badminton                       |
| 15 | 6                  | 46                   | 16                   | Alannah Joseph                    | female                      | 5.8                        | 82               | 30            | Basketball                      |
| 16 | 1                  | 51                   | 17                   | Kelli Owen                        | male                        | 5.8                        | 78               | 38            | Archery                         |
| 17 | 4                  | 54                   | 18                   | Marta Wallace                     | female                      | 5.7                        | 78               | 39            | Badminton                       |
| 18 | 6                  | 56                   | 19                   | Jessie Hoffman                    | male                        | 5.8                        | 75               | 40            | Basketball                      |

Total rows: 18 of 18

Query complete 00:00:00.042

**16) Display the sport with the maximum number of events**

```
select sport_name
from sport natural join event
group by sport_name
having count(sport_id) = (select max("events")
                           from (select sport_name, count(*) as "events"
                                 from sport natural join event
                                 group by sport_name
                               ) as r1
                         )
```

The screenshot shows the pgAdmin 4 interface with a query editor window. The title bar indicates the connection is to '202001276\_db/postgres@PostgreSQL 14'. The toolbar has various icons for database management. The menu bar includes 'Query', 'Data output', 'Query History', 'Messages', and 'Notifications'. The main area displays a results table with one row:

|   | sport_name |
|---|------------|
| 1 | Swimming   |

At the bottom of the window, a status bar shows 'Total rows: 1 of 1' and 'Query complete 00:00:00.046'.

**17) Create a view of the total no. of medals won by each country till the present day**

```
create or replace view country_leaderboard_view as
select country_name, sum(gold_medals) as "gold_medals",
sum(silver_medals) as "silver_medals", sum(bronze_medals) as
"bronze_medals", sum(total_medals) as "total_medals"
from country natural join country_leaderboard
group by country_name
```

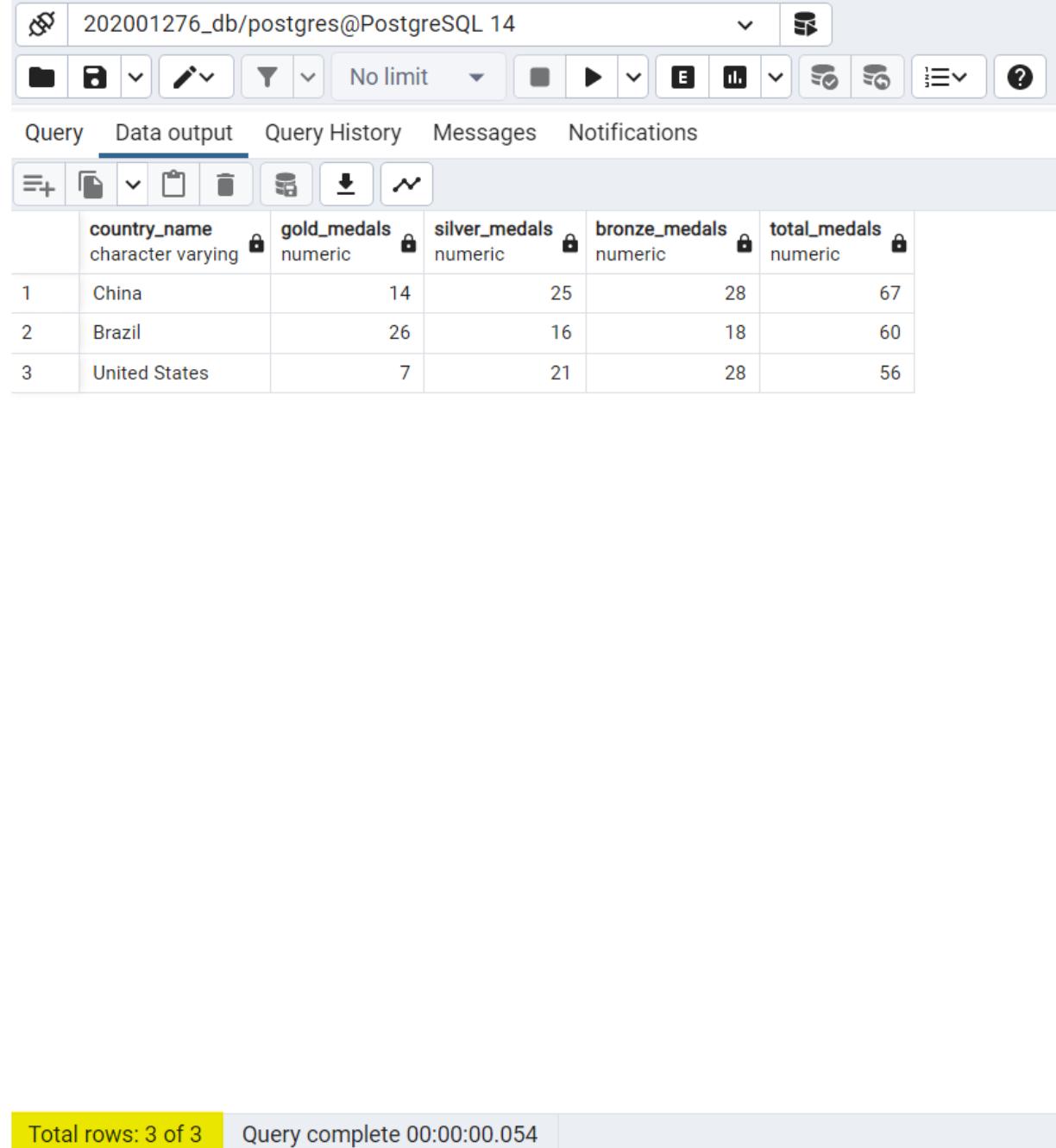
The screenshot shows the pgAdmin 4 interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste, Find, Filter, Refresh, Run, Stop, Exit), a search bar, and various configuration buttons.
- Header:** Displays the connection information: 202001276\_db/postgres@PostgreSQL 14.
- Query Tab:** Active tab, showing the executed SQL query.
- Data Output Tab:** Active tab, displaying the results of the query in a grid format.
- Results Grid:** Shows the data from the query. The columns are:
  - Row number (1 to 20)
  - country\_name (character varying)
  - gold\_medals (numeric)
  - silver\_medals (numeric)
  - bronze\_medals (numeric)
  - total\_medals (numeric)The data rows are:

| row | country_name   | gold_medals | silver_medals | bronze_medals | total_medals |
|-----|----------------|-------------|---------------|---------------|--------------|
| 1   | Italy          | 17          | 8             | 14            | 39           |
| 2   | United Kingdom | 12          | 9             | 11            | 32           |
| 3   | Germany        | 12          | 16            | 22            | 50           |
| 4   | Canada         | 4           | 19            | 6             | 29           |
| 5   | Finland        | 0           | 18            | 17            | 35           |
| 6   | South Korea    | 8           | 21            | 8             | 37           |
| 7   | Greece         | 15          | 16            | 23            | 54           |
| 8   | France         | 24          | 10            | 19            | 53           |
| 9   | Yugoslavia     | 21          | 14            | 8             | 43           |
| 10  | Japan          | 14          | 9             | 10            | 33           |
| 11  | United States  | 7           | 21            | 28            | 56           |
| 12  | Switzerland    | 15          | 9             | 26            | 50           |
| 13  | Russia         | 11          | 13            | 6             | 30           |
| 14  | China          | 14          | 25            | 28            | 67           |
| 15  | Norway         | 22          | 14            | 10            | 46           |
| 16  | Brazil         | 26          | 16            | 18            | 60           |
| 17  | Austria        | 17          | 13            | 15            | 45           |
| 18  | Australia      | 9           | 16            | 7             | 32           |
| 19  | Spain          | 20          | 13            | 18            | 51           |
| 20  | Mexico         | 18          | 11            | 3             | 32           |
- Status Bar:** Shows "Total rows: 20 of 20" and "Query complete 00:00:00.043".

**Display the top three countries of the country leaderboard using the view country\_leaderboard\_view, who has won the highest no. of medals till the present day**

```
select *  
from country_leaderboard_view  
order by total_medals desc  
limit 3
```



The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** 202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste), search, filter, and various execution and monitoring buttons.
- Menu Bar:** Query, Data output, Query History, Messages, Notifications.
- Data Output Tab:** Active tab.
- Table:** Displays the results of the query:

|   | country_name  | gold_medals | silver_medals | bronze_medals | total_medals |
|---|---------------|-------------|---------------|---------------|--------------|
| 1 | China         | 14          | 25            | 28            | 67           |
| 2 | Brazil        | 26          | 16            | 18            | 60           |
| 3 | United States | 7           | 21            | 28            | 56           |

- Status Bar:** Total rows: 3 of 3 | Query complete 00:00:00.054

**18) Create a function that returns the table containing the athlete id, athlete name, and the sport played by each athlete**

```
create or replace function athlete_sport()
returns table(athlete_id bigint, athlete_name character varying,
sport_name character varying)
language 'plpgsql'
as $body$
begin
return query execute format('select athlete_id, athlete_name,
sport_name from athlete natural join sport');
end
$body$
```

```
select *
from athlete_sport();
```

202001276\_db/postgres@PostgreSQL 14

No limit

Query Data output Query History Messages Notifications

athlete\_id athlete\_name sport\_name

|    | athlete_id | athlete_name     | sport_name       |
|----|------------|------------------|------------------|
| 1  | 1          | Shannon Wood     | Archery          |
| 2  | 2          | Chris Lopez      | Aquatics         |
| 3  | 3          | Desiree Miller   | Athletics        |
| 4  | 4          | Bill Anderson    | Badminton        |
| 5  | 5          | Brandon Simpson  | Baseball         |
| 6  | 6          | Juan James       | Basketball       |
| 7  | 7          | Crystal Hamilton | Boxing           |
| 8  | 8          | Kyle Jimenez     | Canoeing         |
| 9  | 9          | Monique Price    | Cycling          |
| 10 | 10         | Virginia Tanner  | Beach Volleyball |
| 11 | 11         | Lauren Richard   | Archery          |
| 12 | 12         | Justin Rivera    | Aquatics         |
| 13 | 13         | Erik Phillips    | Athletics        |
| 14 | 14         | Kimberly Smith   | Badminton        |
| 15 | 15         | James Austin     | Baseball         |
| 16 | 16         | Brittany Silva   | Basketball       |
| 17 | 17         | Anthony Simmons  | Boxing           |
| 18 | 18         | Dale Thomas      | Canoeing         |
| 19 | 19         | Debbie Jackson   | Cycling          |
| 20 | 20         | Joshua Chaney    | Beach Volleyball |
| 21 | 21         | Sarah Walker     | Archery          |

Total rows: 60 of 60    Query complete 00:00:00.059

**19) Create a trigger that checks the primary key constraints before inserting a new record in the athlete table**

```
create or replace function func_1()
returns trigger
language 'plpgsql'
as $body$
begin
if new.athlete_id in (select athlete_id from athlete) then
raise notice 'Violating primary key constraints: athlete_id = % already
exists', new.athlete_id;
return old;
else
raise notice 'Record inserted successfully!';
return new;
end if;
end
$body$
```

```
create or replace trigger trigger_1
before insert
on athlete
for each row
execute procedure func_1();
```

## Before insertion:

The screenshot shows the pgAdmin interface with the 'olympics.athlete' database selected. The 'Data output' tab is active, displaying a table with 60 rows of athlete data. The columns are: athlete\_id, country\_id, sport\_id, athlete\_name, gender, height, weight, and age. The data includes various athletes like Gierin Bird, Shelby Ramos, Pablo Parsons, Lara Kirby, Bennett Mooney, Finley Mills, Alannah Joseph, Jimmy Figueroa, Molly Patrick, Laurel Lam, Mark Gilbert, Kelli Owen, Ervin Jordan, Denise Bryan, Marta Wallace, Ella Aguilar, Jessie Hoffman, Rita Graham, Homer Rhodes, Lloyd Bradley, and Sammy Powell.

|    | athlete_id<br>[PK] bigint | country_id<br>bigint | sport_id<br>bigint | athlete_name<br>character varying | gender<br>character varying | height<br>double precision | weight<br>bigint | age<br>bigint |
|----|---------------------------|----------------------|--------------------|-----------------------------------|-----------------------------|----------------------------|------------------|---------------|
| 40 | 40                        | 14                   | 10                 | Gierin Bird                       | female                      | 5.7                        | 75               | 30            |
| 41 | 41                        | 14                   | 1                  | Shelby Ramos                      | male                        | 5.8                        | 85               | 32            |
| 42 | 42                        | 14                   | 2                  | Pablo Parsons                     | male                        | 5.8                        | 82               | 34            |
| 43 | 43                        | 15                   | 3                  | Lara Kirby                        | female                      | 5.6                        | 80               | 36            |
| 44 | 44                        | 15                   | 4                  | Bennett Mooney                    | male                        | 5.7                        | 84               | 32            |
| 45 | 45                        | 15                   | 5                  | Finley Mills                      | male                        | 5.8                        | 70               | 28            |
| 46 | 46                        | 16                   | 6                  | Alannah Joseph                    | female                      | 5.8                        | 82               | 30            |
| 47 | 47                        | 16                   | 7                  | Jimmy Figueroa                    | male                        | 5.8                        | 76               | 38            |
| 48 | 48                        | 16                   | 8                  | Molly Patrick                     | female                      | 5.6                        | 78               | 34            |
| 49 | 49                        | 17                   | 9                  | Laurel Lam                        | female                      | 5.8                        | 83               | 38            |
| 50 | 50                        | 17                   | 10                 | Mark Gilbert                      | male                        | 5.7                        | 68               | 36            |
| 51 | 51                        | 17                   | 1                  | Kelli Owen                        | male                        | 5.8                        | 78               | 38            |
| 52 | 52                        | 18                   | 2                  | Ervin Jordan                      | male                        | 5.8                        | 83               | 36            |
| 53 | 53                        | 18                   | 3                  | Denise Bryan                      | male                        | 5.6                        | 68               | 35            |
| 54 | 54                        | 18                   | 4                  | Marta Wallace                     | female                      | 5.7                        | 78               | 39            |
| 55 | 55                        | 19                   | 5                  | Ella Aguilar                      | female                      | 5.8                        | 70               | 32            |
| 56 | 56                        | 19                   | 6                  | Jessie Hoffman                    | male                        | 5.8                        | 75               | 40            |
| 57 | 57                        | 19                   | 7                  | Rita Graham                       | female                      | 5.8                        | 80               | 34            |
| 58 | 58                        | 20                   | 8                  | Homer Rhodes                      | male                        | 5.6                        | 78               | 45            |
| 59 | 59                        | 20                   | 9                  | Lloyd Bradley                     | male                        | 5.8                        | 72               | 42            |
| 60 | 60                        | 20                   | 10                 | Sammy Powell                      | male                        | 5.7                        | 75               | 38            |

## Unsuccessful insertion (violating primary key constraints):

```
insert into athlete
values(60, 3, 5, 'Mark Chamberlain', 'male', 5.8, 80, 32)
```

The screenshot shows the pgAdmin interface with the '202001276\_db' database selected. The 'Messages' tab is active, displaying a yellow message box with the text: 'NOTICE: Violating primary key constraints: athlete\_id = 60 already exist'. Below the message, the command 'INSERT 0 0' is shown. At the bottom, the message 'Query returned successfully in 39 msec.' is displayed.

```
NOTICE: Violating primary key constraints: athlete_id = 60 already exist
INSERT 0 0

Query returned successfully in 39 msec.
```

## Successful insertion:

insert into athlete

values(61, 3, 5, 'Mark Chamberlain', 'male', 5.8, 80, 32)

The screenshot shows the pgAdmin interface. At the top, it displays the connection details: '202001276\_db/postgres@PostgreSQL 14'. Below the connection bar are various toolbar icons. The main area has tabs for 'Query', 'Data output', 'Query History', 'Messages', and 'Notifications'. The 'Messages' tab is currently selected, showing a yellow box containing the message 'NOTICE: Record inserted successfully!'. Below this, the query 'INSERT 0 1' is shown. At the bottom, a status message says 'Query returned successfully in 64 msec.'

The screenshot shows the pgAdmin interface with the 'olympics.athlete' schema selected. The 'Data output' tab is active, displaying a table of 61 rows. The table has columns: athlete\_id, country\_id, sport\_id, athlete\_name, gender, height, weight, and age. The last row, which is highlighted in yellow, corresponds to the inserted record: athlete\_id 61, country\_id 3, sport\_id 5, athlete\_name 'Mark Chamberlain', gender 'male', height 5.8, weight 80, and age 32. At the bottom of the table, a message states 'Total rows: 61 of 61' and 'Query complete 00:00:00.137'.

|    | athlete_id<br>[PK] bigint | country_id<br>bigint | sport_id<br>bigint | athlete_name<br>character varying | gender<br>character varying | height<br>double precision | weight<br>bigint | age<br>bigint |
|----|---------------------------|----------------------|--------------------|-----------------------------------|-----------------------------|----------------------------|------------------|---------------|
| 41 | 41                        | 14                   | 1                  | Sneidy Ramos                      | male                        | 5.8                        | 80               | 32            |
| 42 | 42                        | 14                   | 2                  | Pablo Parsons                     | male                        | 5.8                        | 82               | 34            |
| 43 | 43                        | 15                   | 3                  | Lara Kirby                        | female                      | 5.6                        | 80               | 36            |
| 44 | 44                        | 15                   | 4                  | Bennett Mooney                    | male                        | 5.7                        | 84               | 32            |
| 45 | 45                        | 15                   | 5                  | Finley Mills                      | male                        | 5.8                        | 70               | 28            |
| 46 | 46                        | 16                   | 6                  | Alannah Joseph                    | female                      | 5.8                        | 82               | 30            |
| 47 | 47                        | 16                   | 7                  | Jimmy Figueiroa                   | male                        | 5.8                        | 76               | 38            |
| 48 | 48                        | 16                   | 8                  | Molly Patrick                     | female                      | 5.6                        | 78               | 34            |
| 49 | 49                        | 17                   | 9                  | Laurel Lam                        | female                      | 5.8                        | 83               | 38            |
| 50 | 50                        | 17                   | 10                 | Mark Gilbert                      | male                        | 5.7                        | 68               | 36            |
| 51 | 51                        | 17                   | 1                  | Kelli Owen                        | male                        | 5.8                        | 78               | 38            |
| 52 | 52                        | 18                   | 2                  | Ervin Jordan                      | male                        | 5.8                        | 83               | 36            |
| 53 | 53                        | 18                   | 3                  | Denise Bryan                      | male                        | 5.6                        | 68               | 35            |
| 54 | 54                        | 18                   | 4                  | Marta Wallace                     | female                      | 5.7                        | 78               | 39            |
| 55 | 55                        | 19                   | 5                  | Ella Aguilar                      | female                      | 5.8                        | 70               | 32            |
| 56 | 56                        | 19                   | 6                  | Jessie Hoffman                    | male                        | 5.8                        | 75               | 40            |
| 57 | 57                        | 19                   | 7                  | Rita Graham                       | female                      | 5.8                        | 80               | 34            |
| 58 | 58                        | 20                   | 8                  | Homer Rhodes                      | male                        | 5.6                        | 78               | 45            |
| 59 | 59                        | 20                   | 9                  | Lloyd Bradley                     | male                        | 5.8                        | 72               | 42            |
| 60 | 60                        | 20                   | 10                 | Sammy Powell                      | male                        | 5.7                        | 75               | 38            |
| 61 | 61                        | 3                    | 5                  | Mark Chamberlain                  | male                        | 5.8                        | 80               | 32            |

**20) Create a new table, "discontinued\_sports". Create a new trigger and call after the deletion of sport from the sport table. After the removal of each sport, the trigger should enter the sport\_id and sport\_name of the discontinued sport into the discontinued\_sports table.**

```
create table if not exists discontinued_sports(  
sport_id bigint,  
sport_name character varying  
)
```

```
create or replace function func_2()  
returns trigger  
language 'plpgsql'  
as $body$  
begin  
insert into discontinued_sports(sport_id, sport_name)  
values(old.sport_id, old.sport_name);  
return new;  
end  
$body$
```

```
create or replace trigger trigger_2  
after delete  
on sport  
for each row  
execute procedure func_2();
```

## Before Deletion:

### Sport table

The screenshot shows the pgAdmin 4 interface with the following details:

- Database:** olympics.sport/202001276\_db/postgres@PostgreSQL 14
- Toolbar:** Includes icons for file operations (New, Open, Save, Import, Export), search, filter, and various execution and monitoring buttons.
- Menu Bar:** Query, Data output, Query History, Messages, Notifications.
- Table View:** The Sport table is displayed with the following schema and data:
  - Columns:** sport\_id [PK] bigint, sport\_name character varying.
  - Data:** Rows 20 to 40, showing sports like Pentathlon, Rowing, Rugby, Sailing, Shooting, Skateboarding, Sport climbing, Surfing, Table Tennis, Taekwondo, Tennis, Triathlon, Volleyball, Weightlifting, Wrestling, Polo, Trampoline, Cricket, Futsal, Hockey, and Swimming.
- Status Bar:** Total rows: 40 of 40 | Query complete 00:00:00.186

## Discontinued\_sport table

The screenshot shows a PostgreSQL database interface with the following details:

- Connection:** olympics.discontinued\_sports/202001276\_db/postgres@Postg...
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste), search, filter, and various database management functions.
- Menu Bar:** Query, Data output (selected), Query History, Messages, Notifications.
- Tool Buttons:** New query, Open query, Save, Print, Copy, Paste, Download, Refresh.
- Table Structure:** sport\_id (bigint) and sport\_name (character varying).
- Status Bar:** Total rows: 0 of 0 | Query complete 00:00:00.126

## After Deletion:

```
delete from sport  
where sport_id = 40
```

## Sport table

The screenshot shows the DBeaver interface with the 'olympics.sport' database selected. The 'Data output' tab is active, displaying the contents of the 'sport' table. The table has two columns: 'sport\_id' (PK) and 'sport\_name'. The data shows rows from 19 to 39, with the row for sport\_id 40 removed.

|    | sport_id<br>[PK] bigint | sport_name<br>character varying |
|----|-------------------------|---------------------------------|
| 19 | 19                      | Karate                          |
| 20 | 20                      | Pentathlon                      |
| 21 | 21                      | Rowing                          |
| 22 | 22                      | Rugby                           |
| 23 | 23                      | Sailing                         |
| 24 | 24                      | Shooting                        |
| 25 | 25                      | Skateboarding                   |
| 26 | 26                      | Sport climbing                  |
| 27 | 27                      | Surfing                         |
| 28 | 28                      | Table Tennis                    |
| 29 | 29                      | Taekwondo                       |
| 30 | 30                      | Tennis                          |
| 31 | 31                      | Triathlon                       |
| 32 | 32                      | Volleyball                      |
| 33 | 33                      | Weightlifting                   |
| 34 | 34                      | Wrestling                       |
| 35 | 35                      | Polo                            |
| 36 | 36                      | Trampoline                      |
| 37 | 37                      | Cricket                         |
| 38 | 38                      | Futsal                          |
| 39 | 39                      | Hockey                          |

Total rows: 39 of 39    Query complete 00:00:00.105

## Discontinued\_sport table

The screenshot shows a PostgreSQL database interface with the following details:

- Connection:** olympics.discontinued\_sports/202001276\_db/postgres@Postg...
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Import, Export), search, filter, and various database management functions.
- Menu Bar:** Query, Data output, Query History, Messages, Notifications.
- Table View:** A table named "Discontinued\_sport" is displayed with the following schema:

|   | sport_id | sport_name |
|---|----------|------------|
| 1 | 40       | Swimming   |
- Status Bar:** Total rows: 1 of 1 | Query complete 00:00:00.100

## Section8: Project Code with output screenshots

### Backend Codes

#### Connects frontend with backend (Using Django):

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'olympics',  
        'USER': 'postgres',  
        'PASSWORD': 'admin',  
        'HOST': 'localhost',  
        'PORT': '5432',  
    }  
}
```

#### Code for setting up the URL:

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('sortLeaderboard',views.sortLeaderboard,name= "sortLeaderboard"),  
    path('showLeaderboard',views.showLeaderboard,name= "showLeaderboard"),  
    path('insertLeaderboard',views.insertLeaderboard,name= "insertLeaderboard"),  
    path('editLeaderboard/<int:id>',views.editLeaderboard,name= "editLeaderboard"),  
    path('',views.HomePage,name= "HomePage"),  
    path('updateLeaderboard/<int:id>',views.updateLeaderboard,name= "updateLeaderboard"),  
    path('delLeaderboard/<int:id>',views.delLeaderboard,name= "delLeaderboard"),  
    path('deletedLeaderboard/<int:id>',views.deletedLeaderboard,name= "deletedLeaderboard")  
    path('runQueryLeaderboard',views.runQueryLeaderboard,name= "runQueryLeaderboard"),  
  
    path('showAthlete',views.showAthlete,name= "showAthlete"),  
    path('insertAthlete',views.insertAthlete,name= "insertAthlete"),  
    path('sortAthlete',views.sortAthlete,name= "sortAthlete"),  
    path('editAthlete/<int:id>',views.editAthlete,name= "editAthlete"),  
]
```

```

path('updateAthlete/<int:id>',views.updateAthlete,name= "updateAthlete"),
path('delAthlete/<int:id>',views.delAthlete,name= "delAthlete"),
path('deletedAthlete/<int:id>',views.deletedAthlete,name= "deletedAthlete"),
path('runQueryAthlete',views.runQueryAthlete,name= "runQueryAthlete"),
]

```

## **Code for setting up the modules used for frontend:**

```

class Leaderboard(models.Model):
    country_id = models.IntegerField(primary_key=True)
    country_name = models.CharField(max_length=100)
    gold_medals = models.IntegerField()
    silver_medals = models.IntegerField()
    bronze_medals = models.IntegerField()
    total_medals = models.IntegerField()

    class Meta:
        db_table = 'leaderboard'

```

```

class Athlete(models.Model):
    athlete_id = models.IntegerField(primary_key=True)
    athlete_name = models.CharField(max_length=100)
    gender = models.CharField(max_length=100)
    height = models.FloatField()
    weight = models.IntegerField()
    age = models.IntegerField()
    sport = models.CharField(max_length=100)
    country = models.CharField(max_length=100)

    class Meta:
        db_table = 'athlete'

```

## **Code for fetching, editing, deleting and sorting in the database:**

```

def HomePage(request):
    return render(request,'HomePage.html')

```

```
def showLeaderboard(request):
    showall=Leaderboard.objects.all()
    context = {
        'data': showall
    }
    return render(request,'showLeaderboard.html',context)

def sortLeaderboard(request):
    if request.method=="POST":
        if request.POST.get('Sort'):
            type=request.POST.get('Sort')
            sorted=Leaderboard.objects.all().order_by(type)
            context = {
                'data': sorted
            }
            return render(request,'sortLeaderboard.html',context)
    else:
        return render(request,'sortLeaderboard.html')

def insertLeaderboard(request):
    if request.method=="POST":
        if request.POST.get('country_id') and request.POST.get('country_name') and
request.POST.get('gold_medals') and request.POST.get('silver_medals') and
request.POST.get('bronze_medals') and request.POST.get('total_medals'):

            saverecord=Leaderboard()
            saverecord.country_id=request.POST.get('country_id')
            saverecord.country_name=request.POST.get('country_name')
            saverecord.gold_medals=request.POST.get('gold_medals')
            saverecord.silver_medals=request.POST.get('silver_medals')
            saverecord.bronze_medals=request.POST.get('bronze_medals')
            saverecord.total_medals=request.POST.get('total_medals')
            allval=Leaderboard.objects.all()

            for i in allval:
```

```
if int(i.country_id)==int(request.POST.get('country_id')):
    messages.warning(request,'Country already exists....!');
    return render(request,'insertLeaderboard.html')

saverecord.save()
messages.success(request,'Leaderboard '+saverecord.country_name+' is saved
successfully!')
return render(request,'insertLeaderboard.html')

else:
    return render(request,'insertLeaderboard.html')

def editLeaderboard(request,id):
editLeaderboardObj=Leaderboard.objects.get(country_id=id)
context={
    "Leaderboard":editLeaderboardObj
}
return render(request,'editLeaderboard.html',context)

def updateLeaderboard(request,id):
updateLeaderboard=Leaderboard.objects.get(country_id=id)
form=LeaderboardForms(request.POST,instance=updateLeaderboard)
if form.is_valid():
    form.save()
    messages.success(request,'Record updated successfully!')
    return render(request,'editLeaderboard.html',[{"Leaderboard":updateLeaderboard}])

def delLeaderboard(request,id):
delLeaderboardObj=Leaderboard.objects.get(country_id=id)
context={
    "Leaderboard":delLeaderboardObj
}
return render(request,'delLeaderboard.html',context)
```

```

def deletedLeaderboard(request,id):
    delLeaderboardObj=Leaderboard.objects.get(country_id=id)
    delLeaderboardObj.delete()
    showall=Leaderboard.objects.all()
    messages.success(request,'Record deleted successfully!')
    return render(request,'delLeaderboard.html',{'Leaderboard': delLeaderboardObj})

def showAthlete(request):
    showall=Athlete.objects.all()
    context = {
        'data': showall
    }
    return render(request,'showAthlete.html',context)

def insertAthlete(request):
    if request.method== "POST":
        if request.POST.get('athlete_id') and request.POST.get('athlete_name') and
request.POST.get('gender') and request.POST.get('height') and request.POST.get('weight') and
request.POST.get('age') and request.POST.get('sport') and request.POST.get('country'):

            saverecord=Athlete()
            saverecord.athlete_id=request.POST.get('athlete_id')
            saverecord.athlete_name=request.POST.get('athlete_name')
            saverecord.gender=request.POST.get('gender')
            saverecord.height=request.POST.get('height')
            saverecord.weight=request.POST.get('weight')
            saverecord.age=request.POST.get('age')
            saverecord.sport=request.POST.get('sport')
            saverecord.country=request.POST.get('country')

            allval=Athlete.objects.all()

            for i in allval:
                if int(i.athlete_id)==int(request.POST.get('athlete_id')):
                    messages.warning(request,'Athlete already exists....!');

```

```
        return render(request,'insertAthlete.html')

    saverecord.save()
    messages.success(request,'Athlete '+saverecord.athlete_name+' is saved successfully!')
    return render(request,'insertAthlete.html')

else:
    return render(request,'insertAthlete.html')

def sortAthlete(request):
    if request.method=="POST":
        if request.POST.get('Sort'):
            type=request.POST.get('Sort')
            sorted=Athlete.objects.all().order_by(type)
            context = {
                'data': sorted
            }
            return render(request,'sortAthlete.html',context)
    else:
        return render(request,'sortAthlete.html')

def editAthlete(request,id):
    editAthleteObj=Athlete.objects.get(athlete_id=id)
    context={
        "Athlete":editAthleteObj
    }
    return render(request,'editAthlete.html',context)

def updateAthlete(request,id):
    updateAthlete=Athlete.objects.get(athlete_id=id)
    form=AthleteForms(request.POST,instance=updateAthlete)
    if form.is_valid():
        form.save()
        messages.success(request,'Record updated successfully!')
```

```
return render(request,'editAthlete.html',[{"Athlete":updateAthlete})  
  
def delAthlete(request,id):  
    delCusObj=Athlete.objects.get(athlete_id=id)  
    context={  
        "Athlete":delCusObj  
    }  
    return render(request,'delAthlete.html',context)  
  
def deletedAthlete(request,id):  
    delCusObj=Athlete.objects.get(athlete_id=id)  
    delCusObj.delete()  
    showall=Athlete.objects.all()  
    messages.success(request,'Record deleted successfully!')  
    return render(request,'delAthlete.html',[{"Athlete": delCusObj})  
  
def runQueryLeaderboard(request):  
    raw_query = "select * from leaderboard order by total_medals desc limit 3;"  
  
    cursor = connection.cursor()  
    cursor.execute(raw_query)  
    alldata=cursor.fetchall()  
  
    return render(request,'runQueryLeaderboard.html',[{'data':alldata}])  
  
def runQueryAthlete(request):  
    raw_query = "select * from athlete where age between 30 and 35;"  
  
    cursor = connection.cursor()  
    cursor.execute(raw_query)  
    alldata=cursor.fetchall()  
  
    return render(request,'runQueryAthlete.html',[{'data':alldata}])
```

## Screenshots of the Website that connects the Database:

### 1) Login Page:



Username

Password

Remember me

[Forgot password?](#)

### 2) Home Page:



### 3) Athlete Database:

The screenshot shows a web-based application titled "Athlete Records". At the top, there are navigation links: "HomePage", "Athlete Database", and "Insert Athlete". Below the header is a search bar with dropdowns for "Athlete ID" and "Sort", and a "Run Query" button. The main content is a table with 17 rows, each representing an athlete. The columns are: athlete\_id, athlete\_name, gender, height, weight, age, sport, and country. Each row includes "Edit" and "Delete" buttons. The background features the five Olympic rings.

| athlete_id | athlete_name     | gender | height | weight | age | sport            | country        |   |
|------------|------------------|--------|--------|--------|-----|------------------|----------------|---|
| 1          | Shannon Wood     | male   | 5.8    | 75     | 30  | Archery          | United States  | <a href="#">Edit</a> <a href="#">Delete</a> |
| 2          | Chris Lopez      | male   | 5.7    | 82     | 35  | Aquatics         | France         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 3          | Desiree Miller   | female | 5.6    | 73     | 28  | Athletics        | Italy          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 4          | Bill Anderson    | male   | 5.8    | 76     | 32  | Badminton        | Japan          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 5          | Brandon Simpson  | male   | 5.8    | 80     | 34  | Baseball         | Australia      | <a href="#">Edit</a> <a href="#">Delete</a> |
| 6          | Juan James       | male   | 5.7    | 80     | 36  | Basketball       | United Kingdom | <a href="#">Edit</a> <a href="#">Delete</a> |
| 7          | Crystal Hamilton | female | 5.7    | 72     | 32  | Boxing           | Canada         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 8          | Kyle Jimenez     | female | 5.6    | 70     | 28  | Canoeing         | Germany        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 9          | Monique Price    | female | 5.7    | 69     | 30  | Cycling          | China          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 10         | Virginia Tanner  | female | 5.8    | 75     | 38  | Beach Volleyball | South Korea    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 11         | Lauren Richard   | female | 5.7    | 74     | 34  | Equestrian       | Russia         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 12         | Justin Rivera    | male   | 5.7    | 85     | 38  | Fencing          | Greece         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 13         | Erik Phillips    | male   | 5.8    | 82     | 36  | Field Hockey     | Norway         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 14         | Kimberly Smith   | male   | 5.7    | 80     | 35  | Football         | Austria        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 15         | James Austin     | male   | 5.8    | 84     | 39  | Golf             | Switzerland    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 16         | Brittany Silva   | female | 5.7    | 70     | 32  | Gymnastics       | Brazil         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 17         | Anthony Simmons  | male   | 5.8    | 82     | 40  | Handball         | Spain          | <a href="#">Edit</a> <a href="#">Delete</a> |

### 4) Sort Athlete Records:

The screenshot shows the same "Athlete Records" interface as the previous one, but with a different sorting order. The table now lists 17 athletes, but they are sorted by age. The "Age" column is highlighted with a blue background. The sorting dropdown at the top of the table also shows "Age" as the current sort key. The rest of the interface, including the navigation links and the five Olympic rings in the background, remains the same.

| athlete_id | athlete_name     | gender | height | weight | age        | sport        | country        |   |
|------------|------------------|--------|--------|--------|------------|--------------|----------------|---|
| 3          | Desiree Miller   | female | 5.4    | Gender | Athlete ID | Athlete Name |                |   |
| 45         | Finley Mills     | male   | 5.4    | Height |            |              |                |   |
| 8          | Kyle Jimenez     | female | 5.4    | Weight |            |              |                |   |
| 40         | Glenn Bird       | female | 5.4    | Sport  |            |              |                |   |
| 1          | Shannon Wood     | male   | 5.8    | 75     | 30         | Archery      | United States  | <a href="#">Edit</a> <a href="#">Delete</a> |
| 9          | Monique Price    | female | 5.7    | 69     | 30         | Cycling      | China          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 46         | Alannah Joseph   | female | 5.8    | 82     | 30         | Basketball   | United Kingdom | <a href="#">Edit</a> <a href="#">Delete</a> |
| 4          | Bill Anderson    | male   | 5.8    | 76     | 32         | Badminton    | Japan          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 28         | Cassandra Smith  | male   | 5.7    | 75     | 32         | Table Tennis | Poland         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 55         | Ella Aguilar     | female | 5.8    | 70     | 32         | Golf         | Switzerland    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 7          | Crystal Hamilton | female | 5.7    | 72     | 32         | Boxing       | Canada         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 44         | Bennett Mooney   | male   | 5.7    | 84     | 32         | Badminton    | Japan          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 41         | Shelby Ramos     | male   | 5.8    | 85     | 32         | Archery      | United States  | <a href="#">Edit</a> <a href="#">Delete</a> |
| 16         | Brittany Silva   | female | 5.7    | 70     | 32         | Gymnastics   | Brazil         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 38         | Tanner Smith     | male   | 5.8    | 75     | 32         | Futsal       | Mexico         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 39         | Melissa Miller   | female | 5.7    | 72     | 32         | Hockey       | Cuba           | <a href="#">Edit</a> <a href="#">Delete</a> |
| 34         | Eric Leonard     | male   | 5.7    | 76     | 33         | Wrestling    | Indonesia      | <a href="#">Edit</a> <a href="#">Delete</a> |

## 5) Insert Athlete Record:

HomePage | Athlete Database

### Insert Athlete

|              |              |
|--------------|--------------|
| Athlete ID   | 61           |
| Athlete Name | Om Chalodiya |
| Gender       | male         |
| Height       | 5.7          |
| Weight       | 50           |
| Age          | 21           |
| Sport        | Cricket      |
| Country      | India        |
| Insert       |              |

HomePage | Athlete Database

### Insert Athlete

|              |                    |
|--------------|--------------------|
| Athlete ID   | Enter Athlete ID   |
| Athlete Name | Enter Athlete Name |
| Gender       | Enter Gender       |
| Height       | Enter Height       |
| Weight       | Enter Weight       |
| Age          | Enter Age          |
| Sport        | Enter Sport        |
| Country      | Enter Country      |
| Insert       |                    |

Athlete Om Chalodiya is saved successfully!

HomePage | Insert Athlete

### Athlete Records

Athlete ID Sort Run Query

| 44 | Bennett Mooney | male   | 5.7 | 84 | 32 | Badminton        | Japan          | <a href="#">Edit</a> <a href="#">Delete</a> |
|----|----------------|--------|-----|----|----|------------------|----------------|---|
| 45 | Finley Mills   | male   | 5.8 | 70 | 28 | Baseball         | Australia      | <a href="#">Edit</a> <a href="#">Delete</a> |
| 46 | Aiannah Joseph | female | 5.8 | 82 | 30 | Basketball       | United Kingdom | <a href="#">Edit</a> <a href="#">Delete</a> |
| 47 | Jimmy Figueroa | male   | 5.8 | 76 | 38 | Boxing           | Canada         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 48 | Molly Patrick  | female | 5.6 | 78 | 34 | Canoeing         | Germany        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 49 | Laurel Lam     | female | 5.8 | 83 | 38 | Cycling          | China          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 50 | Mark Gilbert   | male   | 5.7 | 68 | 36 | Beach Volleyball | South Korea    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 51 | Kelli Owen     | male   | 5.8 | 78 | 38 | Equestrian       | Russia         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 52 | Ervin Jordan   | male   | 5.8 | 83 | 36 | Fencing          | Greece         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 53 | Denise Bryan   | male   | 5.6 | 68 | 35 | Field Hockey     | Norway         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 54 | Marta Wallace  | female | 5.7 | 78 | 39 | Football         | Austria        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 55 | Ella Aguilar   | female | 5.8 | 70 | 32 | Golf             | Switzerland    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 56 | Jessie Hoffman | male   | 5.8 | 75 | 40 | Gymnastics       | Brazil         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 57 | Rita Graham    | female | 5.8 | 80 | 34 | Handball         | Spain          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 58 | Homer Rhodes   | male   | 5.6 | 78 | 45 | Judo             | Yugoslavia     | <a href="#">Edit</a> <a href="#">Delete</a> |
| 59 | Lloyd Bradley  | male   | 5.8 | 72 | 42 | Karate           | Mexico         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 60 | Sammy Powell   | male   | 5.7 | 75 | 38 | Pentathlon       | Finland        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 61 | Om Chalodiya   | male   | 5.7 | 50 | 21 | Cricket          | India          | <a href="#">Edit</a> <a href="#">Delete</a> |

## 6) Edit Athlete Record:

The screenshot shows a table titled "Athlete Records" with columns for Athlete ID, Name, Gender, Height, Weight, Sport, and Country. The last row, representing Om Chalodiya, has the "Weight" cell highlighted in yellow. There are "Edit" and "Delete" buttons next to each row.

| Athlete ID | Name           | Gender | Height | Weight | Sport | Country          | Edit           | Delete      |
|------------|----------------|--------|--------|--------|-------|------------------|----------------|-------------|
| 44         | Bennett Mooney | male   | 5.7    | 84     | 32    | Badminton        | Japan          | Edit Delete |
| 45         | Finley Mills   | male   | 5.8    | 70     | 28    | Baseball         | Australia      | Edit Delete |
| 46         | Alannah Joseph | female | 5.8    | 82     | 30    | Basketball       | United Kingdom | Edit Delete |
| 47         | Jimmy Figueroa | male   | 5.8    | 76     | 38    | Boxing           | Canada         | Edit Delete |
| 48         | Molly Patrick  | female | 5.6    | 78     | 34    | Canoeing         | Germany        | Edit Delete |
| 49         | Laurel Lam     | female | 5.8    | 83     | 38    | Cycling          | China          | Edit Delete |
| 50         | Mark Gilbert   | male   | 5.7    | 68     | 36    | Beach Volleyball | South Korea    | Edit Delete |
| 51         | Kelli Owen     | male   | 5.8    | 78     | 38    | Equestrian       | Russia         | Edit Delete |
| 52         | Ervin Jordan   | male   | 5.8    | 83     | 36    | Fencing          | Greece         | Edit Delete |
| 53         | Denise Bryan   | male   | 5.6    | 68     | 35    | Field Hockey     | Norway         | Edit Delete |
| 54         | Marta Wallace  | female | 5.7    | 78     | 39    | Football         | Austria        | Edit Delete |
| 55         | Ella Aguilar   | female | 5.8    | 70     | 32    | Golf             | Switzerland    | Edit Delete |
| 56         | Jessie Hoffman | male   | 5.8    | 75     | 40    | Gymnastics       | Brazil         | Edit Delete |
| 57         | Rita Graham    | female | 5.8    | 80     | 34    | Handball         | Spain          | Edit Delete |
| 58         | Homer Rhodes   | male   | 5.6    | 78     | 45    | Judo             | Yugoslavia     | Edit Delete |
| 59         | Lloyd Bradley  | male   | 5.8    | 72     | 42    | Karate           | Mexico         | Edit Delete |
| 60         | Sammy Powell   | male   | 5.7    | 75     | 38    | Pentathlon       | Finland        | Edit Delete |
| 61         | Om Chalodiya   | male   | 5.7    | 50     | 21    | Cricket          | India          | Edit Delete |

The screenshot shows a form titled "Edit Athlete Record" with fields for Athlete ID, Name, Gender, Height, Weight, Age, Sport, and Country. The "Weight" field is highlighted in yellow. A message at the bottom says "Record updated successfully!"

|              |              |
|--------------|--------------|
| Athlete ID   | 61           |
| Athlete Name | Om Chalodiya |
| Gender       | male         |
| Height       | 5.7          |
| Weight       | 60           |
| Age          | 21           |
| Sport        | Cricket      |
| Country      | India        |

The screenshot shows the same "Athlete Records" table as the first screenshot, but the "Weight" cell for Om Chalodiya now contains the value "60", indicating the update was successful.

| Athlete ID | Name           | Gender | Height | Weight | Sport | Country          | Edit           | Delete      |
|------------|----------------|--------|--------|--------|-------|------------------|----------------|-------------|
| 44         | Bennett Mooney | male   | 5.7    | 84     | 32    | Badminton        | Japan          | Edit Delete |
| 45         | Finley Mills   | male   | 5.8    | 70     | 28    | Baseball         | Australia      | Edit Delete |
| 46         | Alannah Joseph | female | 5.8    | 82     | 30    | Basketball       | United Kingdom | Edit Delete |
| 47         | Jimmy Figueroa | male   | 5.8    | 76     | 38    | Boxing           | Canada         | Edit Delete |
| 48         | Molly Patrick  | female | 5.6    | 78     | 34    | Canoeing         | Germany        | Edit Delete |
| 49         | Laurel Lam     | female | 5.8    | 83     | 38    | Cycling          | China          | Edit Delete |
| 50         | Mark Gilbert   | male   | 5.7    | 68     | 36    | Beach Volleyball | South Korea    | Edit Delete |
| 51         | Kelli Owen     | male   | 5.8    | 78     | 38    | Equestrian       | Russia         | Edit Delete |
| 52         | Ervin Jordan   | male   | 5.8    | 83     | 36    | Fencing          | Greece         | Edit Delete |
| 53         | Denise Bryan   | male   | 5.6    | 68     | 35    | Field Hockey     | Norway         | Edit Delete |
| 54         | Marta Wallace  | female | 5.7    | 78     | 39    | Football         | Austria        | Edit Delete |
| 55         | Ella Aguilar   | female | 5.8    | 70     | 32    | Golf             | Switzerland    | Edit Delete |
| 56         | Jessie Hoffman | male   | 5.8    | 75     | 40    | Gymnastics       | Brazil         | Edit Delete |
| 57         | Rita Graham    | female | 5.8    | 80     | 34    | Handball         | Spain          | Edit Delete |
| 58         | Homer Rhodes   | male   | 5.6    | 78     | 45    | Judo             | Yugoslavia     | Edit Delete |
| 59         | Lloyd Bradley  | male   | 5.8    | 72     | 42    | Karate           | Mexico         | Edit Delete |
| 60         | Sammy Powell   | male   | 5.7    | 75     | 38    | Pentathlon       | Finland        | Edit Delete |
| 61         | Om Chalodiya   | male   | 5.7    | 60     | 21    | Cricket          | India          | Edit Delete |

## 7) Delete Athlete Record:

| Athlete Records |                |            |      |    |    |                  |                |   |
|-----------------|----------------|------------|------|----|----|------------------|----------------|---|
|                 |                | Athlete ID | Sort |    |    |                  |                |   |
|                 |                | Run Query  |      |    |    |                  |                |   |
| 44              | Bennett Mooney | male       | 5.7  | 84 | 32 | Badminton        | Japan          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 45              | Finley Mills   | male       | 5.8  | 70 | 28 | Baseball         | Australia      | <a href="#">Edit</a> <a href="#">Delete</a> |
| 46              | Alannah Joseph | female     | 5.8  | 82 | 30 | Basketball       | United Kingdom | <a href="#">Edit</a> <a href="#">Delete</a> |
| 47              | Jimmy Figueroa | male       | 5.8  | 76 | 38 | Boxing           | Canada         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 48              | Molly Patrick  | female     | 5.6  | 78 | 34 | Canoeing         | Germany        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 49              | Laurel Lam     | female     | 5.8  | 83 | 38 | Cycling          | China          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 50              | Mark Gilbert   | male       | 5.7  | 68 | 36 | Beach Volleyball | South Korea    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 51              | Kelli Owen     | male       | 5.8  | 78 | 38 | Equestrian       | Russia         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 52              | Ervin Jordan   | male       | 5.8  | 83 | 36 | Fencing          | Greece         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 53              | Denise Bryan   | male       | 5.6  | 68 | 35 | Field Hockey     | Norway         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 54              | Marta Wallace  | female     | 5.7  | 78 | 39 | Football         | Austria        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 55              | Ella Aguilar   | female     | 5.8  | 70 | 32 | Golf             | Switzerland    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 56              | Jessie Hoffman | male       | 5.8  | 75 | 40 | Gymnastics       | Brazil         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 57              | Rita Graham    | female     | 5.8  | 80 | 34 | Handball         | Spain          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 58              | Homer Rhodes   | male       | 5.6  | 78 | 45 | Judo             | Yugoslavia     | <a href="#">Edit</a> <a href="#">Delete</a> |
| 59              | Lloyd Bradley  | male       | 5.8  | 72 | 42 | Karate           | Mexico         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 60              | Sammy Powell   | male       | 5.7  | 75 | 38 | Pentathlon       | Finland        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 61              | Om Chalodiya   | male       | 5.7  | 50 | 21 | Cricket          | India          | <a href="#">Edit</a> <a href="#">Delete</a> |

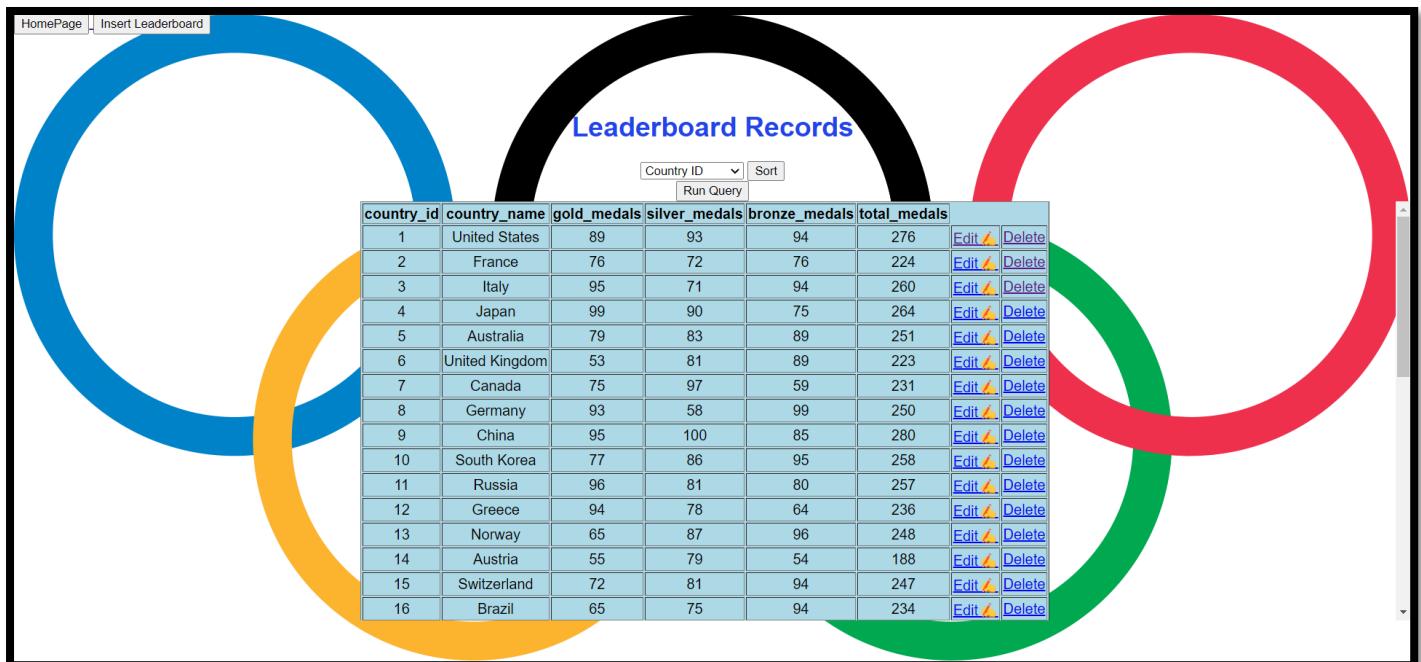
Delete Athlete Record

|              |              |
|--------------|--------------|
| Athlete ID   | None         |
| Athlete Name | Om Chalodiya |
| Gender       | male         |
| Height       | 5.7          |
| Weight       | 60           |
| Age          | 21           |
| Sport        | Cricket      |
| Country      | India        |

Delete Record **Record deleted successfully!**

| Athlete Records |                |            |      |    |    |                  |                |   |
|-----------------|----------------|------------|------|----|----|------------------|----------------|---|
|                 |                | Athlete ID | Sort |    |    |                  |                |   |
|                 |                | Run Query  |      |    |    |                  |                |   |
| 43              | Lara Kirby     | female     | 5.6  | 80 | 36 | Athletics        | Italy          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 44              | Bennett Mooney | male       | 5.7  | 84 | 32 | Badminton        | Japan          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 45              | Finley Mills   | male       | 5.8  | 70 | 28 | Baseball         | Australia      | <a href="#">Edit</a> <a href="#">Delete</a> |
| 46              | Alannah Joseph | female     | 5.8  | 82 | 30 | Basketball       | United Kingdom | <a href="#">Edit</a> <a href="#">Delete</a> |
| 47              | Jimmy Figueroa | male       | 5.8  | 76 | 38 | Boxing           | Canada         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 48              | Molly Patrick  | female     | 5.6  | 78 | 34 | Canoeing         | Germany        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 49              | Laurel Lam     | female     | 5.8  | 83 | 38 | Cycling          | China          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 50              | Mark Gilbert   | male       | 5.7  | 68 | 36 | Beach Volleyball | South Korea    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 51              | Kelli Owen     | male       | 5.8  | 78 | 38 | Equestrian       | Russia         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 52              | Ervin Jordan   | male       | 5.8  | 83 | 36 | Fencing          | Greece         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 53              | Denise Bryan   | male       | 5.6  | 68 | 35 | Field Hockey     | Norway         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 54              | Marta Wallace  | female     | 5.7  | 78 | 39 | Football         | Austria        | <a href="#">Edit</a> <a href="#">Delete</a> |
| 55              | Ella Aguilar   | female     | 5.8  | 70 | 32 | Golf             | Switzerland    | <a href="#">Edit</a> <a href="#">Delete</a> |
| 56              | Jessie Hoffman | male       | 5.8  | 75 | 40 | Gymnastics       | Brazil         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 57              | Rita Graham    | female     | 5.8  | 80 | 34 | Handball         | Spain          | <a href="#">Edit</a> <a href="#">Delete</a> |
| 58              | Homer Rhodes   | male       | 5.6  | 78 | 45 | Judo             | Yugoslavia     | <a href="#">Edit</a> <a href="#">Delete</a> |
| 59              | Lloyd Bradley  | male       | 5.8  | 72 | 42 | Karate           | Mexico         | <a href="#">Edit</a> <a href="#">Delete</a> |
| 60              | Sammy Powell   | male       | 5.7  | 75 | 38 | Pentathlon       | Finland        | <a href="#">Edit</a> <a href="#">Delete</a> |

## 8) Leaderboard Database:



HomePage | Insert Leaderboard

### Leaderboard Records

Country ID Sort Run Query

| country_id | country_name   | gold_medals | silver_medals | bronze_medals | total_medals | Edit | Delete |
|------------|----------------|-------------|---------------|---------------|--------------|------|--------|
| 1          | United States  | 89          | 93            | 94            | 276          | Edit | Delete |
| 2          | France         | 76          | 72            | 76            | 224          | Edit | Delete |
| 3          | Italy          | 95          | 71            | 94            | 260          | Edit | Delete |
| 4          | Japan          | 99          | 90            | 75            | 264          | Edit | Delete |
| 5          | Australia      | 79          | 83            | 89            | 251          | Edit | Delete |
| 6          | United Kingdom | 53          | 81            | 89            | 223          | Edit | Delete |
| 7          | Canada         | 75          | 97            | 59            | 231          | Edit | Delete |
| 8          | Germany        | 93          | 58            | 99            | 250          | Edit | Delete |
| 9          | China          | 95          | 100           | 85            | 280          | Edit | Delete |
| 10         | South Korea    | 77          | 86            | 95            | 258          | Edit | Delete |
| 11         | Russia         | 96          | 81            | 80            | 257          | Edit | Delete |
| 12         | Greece         | 94          | 78            | 64            | 236          | Edit | Delete |
| 13         | Norway         | 65          | 87            | 96            | 248          | Edit | Delete |
| 14         | Austria        | 55          | 79            | 54            | 188          | Edit | Delete |
| 15         | Switzerland    | 72          | 81            | 94            | 247          | Edit | Delete |
| 16         | Brazil         | 65          | 75            | 94            | 234          | Edit | Delete |

## 9) Query-1: Display the details of the athletes who are in the age group of 30-35:

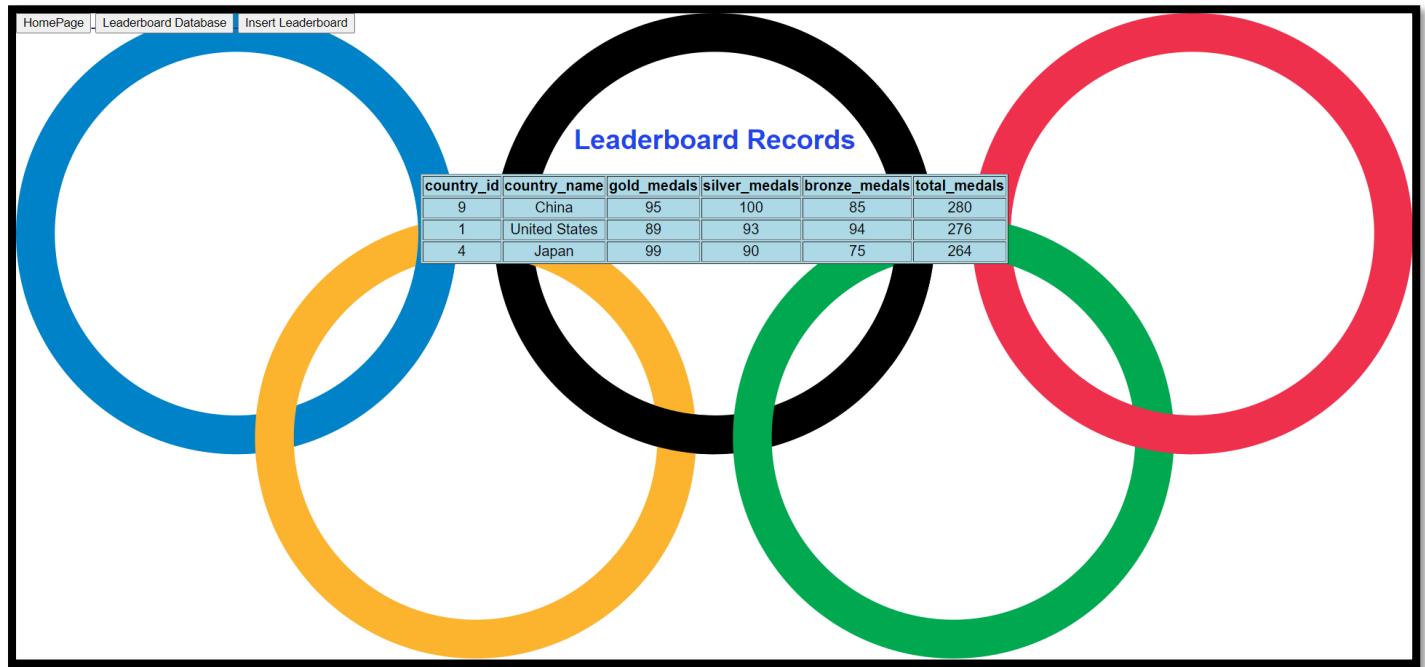


HomePage | Athlete Database | Insert Athlete

### Athlete Records

| athlete_id | athlete_name     | gender | height | weight | age | sport        | country       |
|------------|------------------|--------|--------|--------|-----|--------------|---------------|
| 1          | Shannon Wood     | male   | 5.8    | 75     | 30  | Archery      | United States |
| 2          | Chris Lopez      | male   | 5.7    | 82     | 35  | Aquatics     | France        |
| 4          | Bill Anderson    | male   | 5.8    | 76     | 32  | Badminton    | Japan         |
| 5          | Brandon Simpson  | male   | 5.8    | 80     | 34  | Baseball     | Australia     |
| 7          | Crystal Hamilton | female | 5.7    | 72     | 32  | Boxing       | Canada        |
| 9          | Monique Price    | female | 5.7    | 69     | 30  | Cycling      | China         |
| 11         | Lauren Richard   | female | 5.7    | 74     | 34  | Equestrian   | Russia        |
| 14         | Kimberly Smith   | male   | 5.7    | 80     | 35  | Football     | Austria       |
| 16         | Brittany Silva   | female | 5.7    | 70     | 32  | Gymnastics   | Brazil        |
| 18         | Dale Thomas      | male   | 5.8    | 76     | 34  | Judo         | Yugoslavia    |
| 23         | Elizabeth Haynes | female | 5.7    | 70     | 34  | Sailing      | Sweden        |
| 28         | Cassandra Smith  | male   | 5.7    | 75     | 32  | Table Tennis | Poland        |
| 29         | Katelyn Kim      | male   | 5.8    | 85     | 35  | Taekwondo    | Hong Kong     |
| 31         | Adam Parker      | male   | 5.8    | 79     | 34  | Triathlon    | Romania       |
| 32         | Melissa Jones    | female | 5.6    | 82     | 34  | Volleyball   | Iran          |
| 34         | Eric Leonard     | male   | 5.7    | 76     | 33  | Wrestling    | Indonesia     |
| 35         | Veronica Shields | female | 5.7    | 76     | 33  | Polo         | Zimbabwe      |
| 38         | Tanner Smith     | male   | 5.8    | 75     | 32  | Futsal       | Mexico        |
| 39         | Melissa Miller   | female | 5.7    | 72     | 32  | Hockey       | Cuba          |
| 40         | Glenn Bird       | female | 5.7    | 75     | 30  | Swimming     | Iraq          |
| 41         | Shelby Ramos     | male   | 5.8    | 85     | 32  | Archery      | United States |

**10) Query-2: Display the top three countries of the leaderboard who has won the highest no. of medals till the present day:**



The image shows a screenshot of a web application interface. At the top, there is a navigation bar with three items: "HomePage", "Leaderboard Database", and "Insert Leaderboard". Below the navigation bar, the main content area is titled "Leaderboard Records". Inside this title, there is a table with the following data:

| country_id | country_name  | gold_medals | silver_medals | bronze_medals | total_medals |
|------------|---------------|-------------|---------------|---------------|--------------|
| 9          | China         | 95          | 100           | 85            | 280          |
| 1          | United States | 89          | 93            | 94            | 276          |
| 4          | Japan         | 99          | 90            | 75            | 264          |

**THANK YOU**