A black background with a black square

Description automatically generated with medium confidence

CM3010 Database and Advanced Techniques

**Midterms Assessment**

Student ID: 220516639  
Name: Teo Jia En  
Date of Submission: 5th January 2025

Table of Contents

[Introduction 3](#_Toc186805091)

[Stage 1. Dataset Selection and Critique 3](#_Toc186805092)

[1.1 Choosing a source of open data 3](#_Toc186805093)

[1.2 Access the dataset 4](#_Toc186805094)

[1.3 Interest in dataset and questions asked 5](#_Toc186805095)

[Stage 2. Data Model 5](#_Toc186805096)

[2.1 E/R model 6](#_Toc186805097)

[2.2 Cardinality of E/R diagram 8](#_Toc186805098)

[2.3. Database tables and fields 8](#_Toc186805099)

[2.3 BCNF (Boyce-Codd Normal Form) Analysis 8](#_Toc186805100)

[Stage 3. Database Creation 10](#_Toc186805101)

[3.1 Build database structure in MySQL 10](#_Toc186805102)

[3.1.1 Database and User Creation 10](#_Toc186805103)

[3.1.2 Creation of table to store raw data 10](#_Toc186805104)

[3.1.3 Create Tables 11](#_Toc186805105)

[3.2 Enter Instance Data 13](#_Toc186805106)

[3.3 Reflection on how well the database reflects data 14](#_Toc186805107)

[3.4 Listing of SQL commands 14](#_Toc186805108)

[Stage 4: Create a Simple Web Application 14](#_Toc186805109)

[4.2. Research Questions, SQL Statements and Results 15](#_Toc186805110)

[Research Question One: Number of sports 15](#_Toc186805111)

[Research Question Two: Average Age of Olympians 16](#_Toc186805112)

[Research Question Three: An age-old question 17](#_Toc186805113)

[Research Question Four: NOC and accolades 18](#_Toc186805114)

[Research Question Five: Female athletes and gold medals 18](#_Toc186805115)

[Research Question Six: The Young Olympian 19](#_Toc186805116)

[Research Question 7: The Average Olympian 20](#_Toc186805117)

[Research Question Eight: The Cross-season Olympian 21](#_Toc186805118)

[Research Question Nine: Goldies 22](#_Toc186805119)

[Research Question Ten: NOC Participation 24](#_Toc186805120)

[Conclusion 25](#_Toc186805121)

[Works Cited 26](#_Toc186805122)

[Appendix A 28](#_Toc186805123)

[E/R Model 28](#_Toc186805124)

[Application Logic 29](#_Toc186805125)

[Importing libraries and making database connection 29](#_Toc186805126)

[Index Page and Route for GET request to list all Sports 29](#_Toc186805127)

[Route for GET request to list athlete’s participation in all events 30](#_Toc186805128)

[Route for GET request to first research question 30](#_Toc186805129)

[Route for GET request to second research question 30](#_Toc186805130)

[Route for GET request to third research question 31](#_Toc186805131)

[Route for GET request to fourth research question 31](#_Toc186805132)

[Route for GET request to fifth research question 31](#_Toc186805133)

[Route for GET request to sixth research question 32](#_Toc186805134)

[Route for GET request to seventh research question 32](#_Toc186805135)

[Route for GET request to eighth research question 32](#_Toc186805136)

[Route for GET request to ninth research question 33](#_Toc186805137)

[Route for GET request to tenth research question 33](#_Toc186805138)

# Introduction

**Abstract**

This report presents an in-depth analysis of four consecutive Olympic Games (2010-2016) using a relational database to investigate evolving trends in athlete demographics, such as participation rates by gender and nationality representation.

In fulfilment of this coursework, the project demonstrates knowledge and skills learned across the past ten weeks, including modelling of relational databases, execution of complex SQL queries and development of a web application to allow users to analyse data interactively.

**Domain of choice**

From record-breaking feats to historic achievements, the Olympics is regarded as the world’s foremost international sports event. [1] This year, Paris Olympics offered unforgettable moments, with Simon Biles winning gold in the women’s vault after her two-year break [2] , to the introduction of breaking, or break-dancing, as a new sport [3]. What makes the Olympics captivating is that athletes from all around the world congregate together to compete on many different sports. [4] However, cataloguing the events is a complex endeavour, as the Games and athletes evolve over time. The potential of using a relational database to solve such an issue, and a profound fascination with the Olympic Games, serves as the impetus for this project.

# Stage 1. Dataset Selection and Critique

## 1.1 Choosing a source of open data

The primary data source for this project is a publicly available historical dataset that documents 120 years of Olympic games. [5] The original dataset contains 271,116 entries and 15 columns.

|  |  |  |
| --- | --- | --- |
| 1. | **ID** | Unique identifier for every athlete |
| 2. | **Name** | Full name of the athlete |
| 3. | **Sex** | Male or female. Since the dataset covers up to 2016 and the first openly non-binary athlete participated in the 2020 Games, there are only two possible values for this attribute in this project. [6] |
| 4. | **Age** | Age of the athlete at that game |
| 5. | **Height** | Height of the athlete at that game |
| 6. | **Weight** | Weight of the athlete at that game |
| 7. | **Team** | Name of the team the athlete represents |
| 8. | **NOC** | National Olympic Committee 3-letter code |
| 9. | **Games** | The year and season of the game |
| 10. | **Year** | The year the game is held in |
| 11. | **Season** | Winter or Summer |
| 12. | **City** | The city the game is held in |
| 13. | **Sport** | Sport the event is in (Eg. Swimming) |
| 14. | **Event** | Even the athlete is competing in (Eg. Men’s Individual Butterfly) |
| 15. | **Medal** | Medal the athlete achieved |

## 1.2 Access the dataset

The data was acquired as a comma-separated values (CSV) file from Kaggle and fulfils the following criteria:

1. **Quality:** The data is reliable as it originates from Sports Reference, a website that provides robust sports statistics. [7] The data is also validated by cross-checking of records with the Olympics website and other sources, like Dataful. [8]   
  
2. **Detail:** The 15 columns in the dataset provide a holistic view of the athletes and the Games. These attributes can be used to perform analysis, such as identifying participation rate, or comparing performance across games and countries.   
  
3. **Documentation:** The dataset was accompanied by a short blurb and description of each column. Each column in the file has a informative header. While certain attributes like height and weight have missing units of measurement, it can easily be deduced from the range of data.   
  
A black screen with white text

Description automatically generated

Screen Capture 1 Average of 117 suggests height is in cm, while average of 72 suggests that weight is in kg

4. **Interrelation:** An interdisciplinary approach could yield richer insights. For example, incorporating economic data could reveal correlations between a country’s economy and its athletes’ performance. However, not all countries have reliable economic figures. Moreover, an imbalance in the number of athletes each country send, shown below, would create biased results. Hence, this project will focus primarily on the Games and use a single dataset.

|  |  |
| --- | --- |
| A computer screen shot of a number  Description automatically generated  Screen Capture 2 Athlete count by country (Top 5) | A computer screen with white text  Description automatically generated  Screen Capture 3Athlete count by country (Bottom 5) |

5. **Use:** The dataset will be mainly used to answer questions on the demographic of the athletes. Some data missing from this dataset include: details on the coaches and athlete’s performance at other events such as the Commonwealth Games. While these information would augment insights into the Games, they are superfluous and would not have a huge impact on this study.  
  
6. **Discoverability:** It was easy to find open data in the chosen domain of the Olympic Games. However, most data pertain to only one single Game. Other data sources that have been considered, but ultimately eliminated, for this project include:

- Olympics official website: This authoritative website contains substantial information that are dispersed across web pages. [9] Consolidating this information would require the use of web scrapers. However, the site has no clear policy on the use of automated robots. To prevent any legal infractions, the use of this website was not considered.   
  
- Olympic Data Feed (ODF): This repository serves as the Documentation Web site of the International Olympic Committee (IOC). [10] However, the information is stored as PDFs and could not be easily consolidated.

**Terms of Use**   
The owner of the dataset, on Kaggle, lists the data under the CC0 1.0 Universal Deed. [11] Sports Reference, where the data originates, permits the use of its data for personal and non-commercial purposes, provided that the data is 1) not used to create a database that compete with the platform, or 2) used to train generative machine learning algorithms. The use of this dataset, as fulfilment of an academic coursework, complies to the terms of use.

1.3 Interest in dataset and questions asked   
The data set is intriguing, as it provides information of both the athletes, and the games itself. A cursory look indicates that it covers both the Summer and the Winter Olympic Games. This facilitates queries into the difference between the two games. The research questions below investigate factors that influence athlete success and larger emerging trends in the Olympics. This project answers these questions using a relational model and MySQL queries.

*1. How many sports were there at each of the four games?*

*2. What is the average age of athletes that won medals?*

*3. How old is the oldest Olympian?*

*4. Which NOC won the most bronze, silver and gold medals across all four games?*

*5. Which female athletes accumulated the most gold medals?*

*6. How old is the youngest Olympian and what country do they represent?*

*7. What is the average height and weight of athletes in a sport?*

*8. Were there athletes that participated in both Summer and Winter games and won medals?*

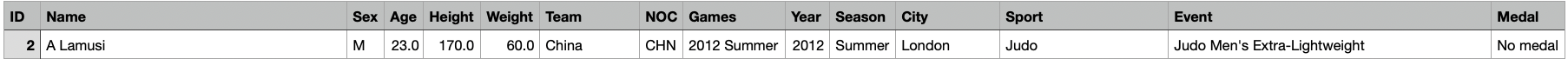
*9. Were there athletes that always ended up on the podium at all their events in London 2012 and won gold medals at both London and Rio 2016?*

*10. Were there any NOCs that participated in the 2010 Vancouver Games but not the 2014 Sochi Games?*

# Stage 2. Data Model

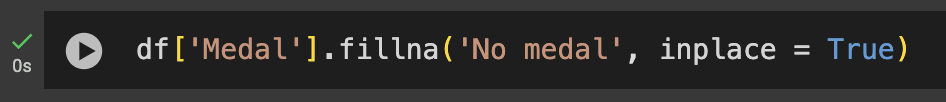
**Dataset pre-processing**

The data is originally in First Normal Form, with single valued attributes. It fails to comply to Second Normal Form as there are partial dependencies.



Screen Capture 4 A single record in the raw data table

For example, the primary key in the table is ‘ID, but ‘City’ is dependent on ‘Year’ and ‘Games’, while ‘Event’ is dependent on ‘Sport’. Data pre-processing for this project is necessary to normalise the table. This project made use of Pandas, ia Python data manipulation and analysis tool, and NumPy, a tool to work with numerical data. [12]   
  
Handling missing values   
The medals column uses null values to represent athletes who did not receive a medal. To accord more meaning to the data, these fields are replaced with a string ‘No medal’.



Screen Capture 5 Pandas fillna method

The dataset contains missing values in 'age' and 'weight' columns, initially represented by NumPy's NaN. Since SQL does not natively support this data type, they were replaced with Python's NoneType object. This facilitates data ingestion within the SQL environment.

A black rectangular with white text

Description automatically generated with medium confidence

Screen Capture 6 Pandas replace method and Numpy NaN data type

The original data contains 271,116 entries. To cater for data processing on Coursera Labs, the data has been limited to include only Games that took place after the year 2010. This reduced the dataset to around 36,000 entries.

A black background with white letters

Description automatically generated

Screen Capture 7 Pandas sub-setting of dataset

The data is then saved as a new CSV file with the Pandas “to\_csv” method.

2.1 E/R model   
To effectively model the relationship between the various entities within the dataset, it is important to construct an accurate Entity-Relationship (E/R) model.   
A diagram of a flowchart

Description automatically generated

Screen Capture 8 ERD diagram (Larger screenshot in landscape format can be found in [Appendix A](#_E/R_Model))

The ERD above uses the conceptual Chen notation. The ERD models a database that captures key entities in the dataset.

|  |  |
| --- | --- |
| Athlete | * Athlete\_id: Unique identifier for the athlete. * Name of the athlete. * Sex of the athlete. |
| Game | * Game\_id: Unique identifier for the Game * Season: Summer or Winter * City: City the game is held (Eg. London) * Year (Eg. 2012) |
| Noc | * Noc\_id: Unique identifier for each National Olympic Committee. * Name: The NOC code. (eg. USA) * Team: The name of the team (eg. United States 1) |
| Sport | * Sport\_id: Unique identifier for each sport. * Name: The name of the sport (eg. “Swimming”) |
| Event | * Event\_id: Unique identifier for each event. * Name: Name of the event (eg. “Women’s 100M Butterfly Event) |
| Medal | * Medal\_type: Gold, Silver, Bronze, No medal |
| Athlete\_at\_game | This entity is an associative entity that resolves many-to-many relationships present in the database. [13]   * The first of such relationships exists between Athletes and Games. One athlete can participate in multiple Games and one Game is participated by many athletes. * The second of such relationship exists between NOC and Games. One NOC can participate in multiple Games and one Game is participated by many athletes.   This entity converts each many-to-many relationship into two one-to-many relationships. |

## 2.2 Cardinality of E/R diagram

|  |  |  |
| --- | --- | --- |
| Noc to Athlete\_game | 1:M | One NOC can be represented by many Athlete games, but one Athlete game can only have one NOC. |
| Game to Athlete\_game | 1:M | One Athlete Game can only belong to one Game, but one Game can have many Athlete Games. |
| Game to Event | 1:M | One Game have many Events. |
| Sport to Event | 1:M | One Sport can have many Events, but one Event can only belong to one Sport. |
| Athlete to Athlete\_game | 1:M | One Athlete can appear at many Athlete\_games but one Athlete\_game can only refer to one Athlete. |
| Athlete to Medal | 1:M | One Athlete can have many Medals, but one Medal only belongs to one Athlete. |
| Medal to Event | 1:M | One Medal belongs to one Event, but one Event can have multiple Medals. |

2.3. Database tables and fields   
A relational database schema is created to serve as a blueprint of how the entities’ relationships will be structured within the database. A diagram of a game

Description automatically generated with medium confidence

Screen Capture 9 Relational diagram of database

The schema consists of seven tables, as translated form the seven entities in the ER diagram, with the primary and foreign keys highlighted in bold.

## 2.3 BCNF (Boyce-Codd Normal Form) Analysis

BCNF is a standard used in database normalisation to reduce redundancy and improve data integrity. In BCNF, functional dependency is defined as (, where should be either the super key or the candidate key. [14] Identification of the primary key and functional dependencies is necessary to check if a table is in this form.

|  |  |  |
| --- | --- | --- |
| **Table** | **Functional Dependency** | **Super key** |
| NOC |  | NOC (Primary key) |
| Athletes |  | athlete\_id (Primary key) |
| Athlete\_Game |  | (athlete\_id, game\_id, noc\_id) (Composite keys) |
| Athlete\_Event\_Medal |  | (athlete\_id, event\_id) (Composite keys) |
| Games |  | Game\_id (Primary key) |
| Events |  | Event\_id (Primary key) |
| Sports |  | Sport\_id (Primary key) |

Evaluation of the functional dependencies and the super keys revealed that all tables satisfy the condition for BCNF.

1. **‘noc’ table**: The ‘noc\_id’ is the primary key that uniquely identifies each Olympic committee.   
  
2. ‘**athletes’ table**: The ‘athlete\_id’ is the primary key and determines the athlete’s name and sex.   
  
3**. ‘sports’ table**: The ‘sport\_id’ field uniquely identifies each sport.   
  
4. **‘events’ table**: The ‘event\_id’ is a primary key that uniquely identifies each event, which includes the event name, the Game it is held in and sport that is played at that event.  
  
5. **‘games’ table**:The ‘games\_id’ is a primary key that uniquely identifies each Game. The season, city and year fields are dependent on this id.

6. **‘athlete\_game’ table**: The composite key of 'athlete\_id', 'game\_id' and ‘noc\_id’ uniquely identifies each athlete's participation in a specific game representing a particular noc, and to associate their age, height, weight, and NOC with that Game.

7. **‘athlete\_event\_medal’**: The composite key of ‘athlete\_id’ and ‘event\_id’ uniquely identifies each athlete’s participation at a specific event.

BCNF ensures that business rules expressed in functional dependencies using keys are correctly followed. [15] It also enforces data integrity and reduces risk of anomalies and inconsistencies by removing partial and transitive dependencies. [16]

The transformation of the table from wide format to long format ensures that the database is scalable and flexible. For example, the ‘sports’ table implemented as a long-table format ensures that the database can scale up and adapt to new sports without a redesign of the database. [17] However, performing analysis of the data, as seen in [Section 3.3](#_3.3_Reflection_on), requires the tables to be de-normalised again and combined.

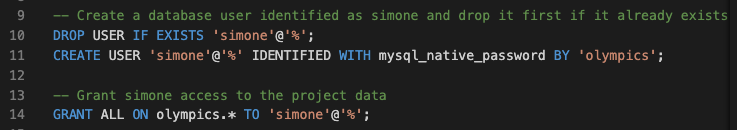
# Stage 3. Database Creation

The database is constructed on a local machine, before it is uploaded onto Coursera Labs. To populate the database with data, four SQL script files were implemented. Extract, transform and load (ETL) is the process of extracting and cleaning data, and loading the prepped data into a target database. The ETL pipeline is used in this project to improve its quality by performing data cleansing before data loading. [18]

## 3.1 Build database structure in MySQL

### 3.1.1 Database and User Creation

The first SQL file, ‘create\_database.sql’, is written to create the Olympics database and add a new user. The command ‘CREATE DATABASE Olympics;’ creates a new database. A fictional user ‘Simone’ with the password ‘olympics’ is then granted all permissions.



Screen Capture 10 Creation of user and granting permission to user

The username and password is saved as a .env file and loaded into the process using the DotEnv module, as seen below.

|  |  |
| --- | --- |
| A screenshot of a computer screen  Description automatically generated  Screen Capture 11.env file for variables | Screen Capture 12 Database conection with DotEnv module |

Storing the credentials as environment variables ensures that sensitive information is kept secure and confidential. This complies with best practices for database management.

***MySQL Command execution of the SQL file****: “SOURCE create\_database SQL”*

### 3.1.2 Creation of table to store raw data

The second SQL file, ‘load\_denormalised.sql’, stores the raw data as a wide format into a table called ‘denormalised’. A CREATE TABLE statement is used to create the table.

|  |
| --- |
| CREATE TABLE denormalised (  id INT,  name VARCHAR(128),  sex VARCHAR(64),  age INT,  height INT NULL,  weight INT NULL,  team VARCHAR(64),  noc VARCHAR(64),  games VARCHAR(128),  year INT,  season VARCHAR(128),  city VARCHAR(64),  sport VARCHAR(64),  event VARCHAR(128),  medal VARCHAR(64)  ); |

Each attribute in the table corresponds to one column in the original CSV file. Using the LOAD DATA SQL statement, the data is loaded into the ‘denormalised’ table.

|  |
| --- |
| LOAD DATA INFILE '/home/coder/project/scripts/athletes\_events\_cleaned.csv'  INTO TABLE denormalised  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n'  IGNORE 1 ROWS  (id, name, sex, @age, @height, @weight, team, noc, games, @year, season, city, sport, event, medal)  SET age = NULLIF(@age,''),  height = CASE  WHEN @height = '' OR @height = 'N' THEN NULL  ELSE @height  END,  weight = NULLIF(@weight, ''),  year = NULLIF(@year, ''); |

The LOAD DATA statement reads the rows from the CSV file, terminating the rows by ‘\n’ and separating each field by a comma. [19] As NULL values are signified by empty strings in CSV files, a CASE statement must be made to handle potential invalid statements. The ‘@’ operator in front of potential invalid field declares these parameters as variables. If the input parameters are empty strings, it will be replaced with NULL using the NULLIF function. [20]

***MySQL Command execution of the SQL file****: “SOURCE load\_denormalised.sql”.*

### 3.1.3 Create Tables

The third SQL file creates the seven tables defined in [Section 2](#_Stage_2._Data).

|  |  |
| --- | --- |
| SQL commands | Description |
| CREATE TABLE noc (  noc\_id int PRIMARY KEY AUTO\_INCREMENT,  noc varchar(16),  team varchar(65)  ); | Creates a table “noc” with attributes noc, team and noc\_id (primary key). |
| CREATE TABLE athletes (  athlete\_id int PRIMARY KEY,  name varchar(255),  sex varchar(16)  ); | Creates a table “athletes” with attributes name and sex and athlete\_id (primary key). |
| CREATE TABLE games (  game\_id int PRIMARY KEY AUTO\_INCREMENT,  season varchar(16),  city varchar(64),  year int  ); | Creates a table “games” with attributes season, city, year and game\_id (primary key). |
| CREATE TABLE sports (  sport\_id int PRIMARY KEY AUTO\_INCREMENT,  name varchar(255)  ); | Creates a table “sports” with attributes name and sport\_id (primary key). |
| CREATE TABLE events (  event\_id int PRIMARY KEY AUTO\_INCREMENT,  event\_name varchar(255),  sport\_id int,  game\_id int,  FOREIGN KEY (sport\_id) REFERENCES sports(sport\_id),  FOREIGN KEY (game\_id) REFERENCES games(game\_id)  ); | Creates a table “events” with attributes event\_name, sport\_id, game\_id and event\_id (Primary key). Sport\_id and game\_id are foreign keys that reference the Sports and Games table respectively. |
| CREATE TABLE athlete\_game (  athlete\_id int,  game\_id int,  age int,  height int,  weight int,  noc\_id int,  PRIMARY KEY (athlete\_id, game\_id, noc\_id),  FOREIGN KEY (noc\_id) REFERENCES noc(noc\_id),  FOREIGN KEY (athlete\_id) REFERENCES athletes(athlete\_id),  FOREIGN KEY (game\_id) REFERENCES games(game\_id)  ); | Creates a junction table “athlete\_game” with attributes athlete\_id, game\_id, age, height, weight and noc\_id. The composite key athlete\_id, game\_id and noc\_id uniquely identifies an athlete’s participation in a game. The keys noc\_id, athlete\_id and game\_id references the Noc, Athletes and Games tables respectively. |
| CREATE TABLE athlete\_event\_medal (  athlete\_id int,  event\_id int,  medal varchar(16),  PRIMARY KEY (athlete\_id, event\_id),  FOREIGN KEY (athlete\_id) REFERENCES athletes(athlete\_id),  FOREIGN KEY (event\_id) REFERENCES events(event\_id)  ); | Creates a junction table “athlete\_event\_medal” with attributes athlete\_id, event\_id, medal. The composite key of athlete\_id and event\_id uniquely identifies an athlete’s participation at a single event. The athlete\_id and event\_id references the Athlete and Events table respectively. |

**Data types** used in this project are:

- INT: Fields that are unique identifiers or numerical.  
- VARCHAR(16): Fields with shorter strings as this length requires less space allocation   
- VARCHARR(64): Fields with strings of medium length  
- VARCHAR(255): Fields with with long strings such as event names.

***MySQL Command execution of the SQL file****: “SOURCE create\_tables.sql”.*

## 3.2 Enter Instance Data

The fourth SQL file, ‘populate\_data.sql’ loads the data from the ‘denormalised’ table into the main tables defined in the previous section.

|  |  |
| --- | --- |
| **Table** | **SQL Statement** |
| Noc | INSERT INTO noc (noc, team)  SELECT DISTINCT noc, team FROM denormalised; |
| Athletes | INSERT INTO athletes (athlete\_id, name, sex)  SELECT DISTINCT id, name, sex FROM denormalised; |
| Games | INSERT INTO games (season, city, year)  SELECT DISTINCT season, city, year FROM denormalised; |
| Sports | INSERT INTO sports (name)  SELECT DISTINCT sport FROM denormalised; |
| Events | INSERT INTO events (event\_name, sport\_id, game\_id)  SELECT DISTINCT d.event, s.sport\_id, g.game\_id  FROM denormalised d  INNER JOIN sports s  ON d.sport = s.name  INNER JOIN games g  ON d.year = g.year AND d.season = g.season AND d.city = g.city; |
| Athlete\_game | INSERT INTO athlete\_game (athlete\_id, game\_id, age, height, weight, noc\_id)  SELECT DISTINCT a.athlete\_id, g.game\_id, d.age, d.height, d.weight, n.noc\_id  FROM denormalised d  INNER JOIN athletes a  ON d.id = a.athlete\_id  INNER JOIN games g  ON d.year = g.year AND d.season = g.season AND d.city = g.city  INNER JOIN noc n  ON d.noc = n.noc AND d.team = n.team; |
| Athlete\_event\_medal | INSERT INTO athlete\_event\_medal (athlete\_id, event\_id, medal)  SELECT DISTINCT a.athlete\_id, e.event\_id, d.medal  FROM denormalised d  INNER JOIN athletes a  ON d.id = a.athlete\_id  INNER JOIN events e  ON e.sport\_id = (SELECT sport\_id FROM sports s WHERE s.name = d.sport)  AND e.game\_id = (SELECT game\_id FROM games g WHERE g.year = d.year AND g.season = d.season AND g.city = d.city)  AND e.event\_name = d.event; |

***MySQL Command execution of the SQL file****: “SOURCE populate\_data.sql”.*

## 3.3 Reflection on how well the database reflects data

The database was able to capture the data well. By using junction tables such as athlete\_game and athlete\_event\_medal, problem many-to-many relationships between the tables – which could introduce data redundancy and performance overhead – are removed. [21]

However, complex queries with multiple joins are resource-intensive, which slows down query execution. Breaking down the data into multiple tables increased storage requirements and resulted in poorer performance.

A computer screen shot of white text

Description automatically generated

Screen Capture 13 Performing analysis with "EXPLAIN FORMAT=TREE" on our last research question

The screen capture above shows that the rows read is astronomical. While results loaded in seconds on the local environment, the query causes a slight lag in Coursera Labs.

## 3.4 Listing of SQL commands

All the questions raised in Stage 1 Step 3 can be answered. The SQL commands and answering these questions will be covered in-depth in section 4.2 of this report.

Stage 4: Create a Simple Web Application  
4.1 Node.js Module

This section covers how a simple web application is created to provide an interface with which users can interact with the data. All routes are defined in the ‘index.js’ file. All the **routes for the endpoints** can be found in [Appendix A](#_Application_Logic).

**A screenshot of a computer

Description automatically generatedIndex Page**

The main page serves as the gateway to the different endpoints displayed in a table, with each row dedicated to an endpoint that answers a specific research question. Each row includes a summary that gives users an understanding of the scope and relevance of the query before initiating it. The ‘Advanced SQL used’ column lists advanced SQL techniques used in this query.A link button executes the query and brings the user to a page that displays the information answering the query.

Screen Capture 14 The index page of the application

The web application uses a three-tier architecture. The presentation tier is developed with HTML and the Bootstrap framework [21]. The application tier is built on Node.JS and Express, which processes requests the client makes and performs business logic. The data tier is implemented with MySQL and the data is stored across seven tables.

**User interaction**

A diagram of a process

Description automatically generated

The flowchart above shows the user interaction flow. When the user enters the command ‘node index.js’, a database connection is made, as seen in [Appendix A](#_Importing_libraries_and). The user makes a database connections with the credentials provided in the ‘.env’ file, as covered in the [previous section](#_3.1.1_Database_and). When a user clicks on a link, a GET request will be triggered for the endpoint. The relevant SQL statement will be executed, and the data will be returned to the user in a web page, with information populated with the EJS templating engine.

## 4.2. Research Questions, SQL Statements and Results

Each research question is addressed through an SQL query. The questions are ordered in increasing complexity, accompanied by more sophisticated SQL statements.

Research Question One: Number of sports  
**Question:** How many sports were there at each of the four games?

**Background:** The recent 2024 Paris Olympics featured a few new sports – such as breaking (street dance), skateboarding and sport climbing. This research question aims to investigate how the number of sports featured has changed over time and whether it is a steady increase.

**SQL used:** Inner Join, Group by and Having clause

|  |
| --- |
| SELECT g.season, g.city, g.year, count(sport\_id) AS sport\_count  FROM games g  INNER JOIN events e  ON g.game\_id = e.game\_id  GROUP BY g.game\_id  ORDER BY g.year; |

This query extracts information about the number of sports at each Olympic Games by joining the Games and Events tables and grouping the resultant table by Game. The results are then ordered chronologically.  
  
**Result page:** The result page shows a table with the count of sports at each game.   
A white paper with black lines

Description automatically generated

Screen Capture 15 Result page for Research Question One

**Analysis:** From the table below, it appears that there is an increase of sports from 2010 to 2014 (86 to 98) and 2012 to 2016 (302 to 306). Evidently, games have been added to newer renditions. This collaborates initial research into this domain, which showed that new sports are introduced and featured as special editions at each Game. [23]

### Research Question Two: Average Age of Olympians

**Question:** What is the average age of athletes that won medals?

**Background:** Since the Olympics test physical prowess, it is plausible that Olympians achieve peak performance in their youth. This research question tests this hypothesis.

**SQL used:** Aggregation with Avg(), Inner Join and filtering with the WHERE clause

|  |
| --- |
| SELECT AVG(ag.age) AS age  FROM athlete\_game ag  INNER JOIN athlete\_event\_medal aem  ON ag.athlete\_id = aem.athlete\_id  WHERE aem.medal != 'No medal'; |

This query joins the athlete\_event\_model and athlete\_game tables, filters for medal-winning athletes, and then calculates the average.

**Result page:** The result page shows the results of the calculation.



Screen Capture 16 Result page for Research Question Two

**Analysis:** Based on the results, Olympians who won medals had an average age of 26.5 years. This implies that younger Olympians may possess a higher probability of winning medals.

Research Question Three: An age-old question  
**Question:** How old is the oldest Olympian?

**Background:** The second query revealed that the average age of an Olympian with a medal is 26.5 years old. This prompts the subsequent research question - who was the oldest Olympian to compete between 2010 and 2016, and which sport did they participate in? This investigation sheds light on the extent to which age may be a less significant factor in certain Olympic sports.

**SQL used**: Left Joins, Aggregation with Max() and filtering with Where clause

|  |
| --- |
| SELECT a.athlete\_id, a.name, a.sex, ag.age, ag.height, e.event\_name  FROM athletes a  INNER JOIN athlete\_game ag  ON a.athlete\_id = ag.athlete\_id  LEFT JOIN athlete\_event\_medal aem  ON a.athlete\_id = aem.athlete\_id  LEFT JOIN events e  ON aem.event\_id = e.event\_id  WHERE age = (SELECT MAX(age) from athlete\_game); |

The query joins three tables – athletes, athlete\_game and athlete\_event\_medal and returns the record that holds the maximum age from the athelete\_game table.

**Result page:**

**A close up of a text

Description automatically generated**

Screen Capture 17 Result page for Research Question Three

**Analysis:** The results showed that Mr Hiroshi Hoketsu was the oldest participant across the four games. He competed in the Equestrianism Mixed Dressage event at 71-years-old. This webpage also includes a link, which brings the user to all the events that the athlete has participated in.

A screenshot of a phone

Description automatically generated

Screen Capture 18 Result page for Olympic Career of Mr Hiroshi Hoketsu

The link brings the user to a second webpage, which discloses the singular event Mr Hokuetsu participated in. He represented Japan at the London Games but did not win a medal. This finding suggests that Japan might be a country that supports and funds older athletes, and that Equestrianism is a sport that is age friendly.

### Research Question Four: NOC and accolades

**Question:** Which NOC won the most bronze, silver and gold medals across all four games?

**Background:** The objective of this research question is to ascertain the countries with the most accolades across all four games. Identifying these nations will be instrumental in conducting a study into the essential infrastructure that a country must establish to foster and nurture a flourishing sports environment.

**SQL used**: Aggregation with Sum, Ordering with Case keyword.

|  |
| --- |
| SELECT n.noc, g.city, g.year, g.season,  SUM(CASE WHEN aem.medal = 'Gold' THEN 1 ELSE 0 END) AS gold,  SUM(CASE WHEN aem.medal = 'Silver' THEN 1 ELSE 0 END) AS silver,  SUM(CASE WHEN aem.medal = 'Bronze' THEN 1 ELSE 0 END) AS bronze,  COUNT(aem.medal) AS total  FROM noc n  JOIN athlete\_game ag ON n.noc\_id = ag.noc\_id  JOIN games g ON ag.game\_id = g.game\_id  LEFT JOIN athlete\_event\_medal aem ON ag.athlete\_id = aem.athlete\_id  GROUP BY n.noc, g.game\_id, g.year, g.season  ORDER BY total DESC; |

The query performs a summation of the medals and group the results by the NOC and the game.

**Result page:**

**A table with numbers and text

Description automatically generated**

Screen Capture 19 Result page for Research Question Four (Truncated)

**Analysis:** This table shows the medals of each NOC aggregated over the four Olympic games. This shows us which country boasts the highest number of medallists and will facilitate an investigation into how a country can best support its athletes.

### Research Question Five: Female athletes and gold medals

**Question:** Which female athletes accumulated the most gold medals?

**Background:** This query aims to identify the most successful female Olympians by finding those that accumulated the most gold medals. This query lays the groundwork for further investigation in any commonalities that these athletes share that aided them in their success.

**SQL used**: Inner Joins, filtering with Where Clause, Group by and Order by

|  |
| --- |
| SELECT a.athlete\_id, a.name, count(m.athlete\_id) AS gold\_medal\_count  FROM athlete\_event\_medal m  INNER JOIN athletes a  ON a.athlete\_id = m.athlete\_id  WHERE (m.medal = 'Gold' AND a.sex = 'F')  GROUP BY a.athlete\_id  ORDER BY gold\_medal\_count DESC  LIMIT 10; |

This query joins two tables, athletes and athlete\_event\_medal to calculate the number of gold medals each female athlete won. The top 10 athletes are then returned.

**Result page:**

**A screenshot of a computer

Description automatically generated**

Screen Capture 20 Result page for Research Question Five

**Analysis:** It is evident that Ms Marit Bjrgen won the most medals across the four games.

### Research Question Six: The Young Olympian

**Question:** How old is the youngest Olympian and what country do they represent?

**Background:** The third query established the oldest age of the Olympian as 71 years old. This research question shifts the focus to the opposite end of the spectrum - to identify the youngest Olympian and the nations they represent. This information can illuminate evolving demographics in participation of youths, as well as any traits that these young athletes share.

**SQL used**: Inner Join, Left Join, Aggregation with Min() and filtering with Where clause.

|  |
| --- |
| SELECT a.athlete\_id, a.name, a.sex, ag.age, ag.height, n.team  FROM athletes a  INNER JOIN athlete\_game ag  ON a.athlete\_id = ag.athlete\_id  LEFT JOIN noc n  ON ag.noc\_id = n.noc\_id  WHERE age = (SELECT MIN(age) from athlete\_game); |

This query combines three tables – athletes, athlete\_game and noc and returns the record with the lowest age.

**Result page:**

**A screenshot of a phone

Description automatically generated**

Screen Capture 21Result page for Research Question Six

**Analysis:**

The table below shows the youngest Olympians across the Games, who are thirteen-year-olds. When delving into their sport (by clicking on the rightmost column), it is evident that all three athletes are swimmers.

A screenshot of a phone

Description automatically generated

Screen Capture 22 Result page of Olympic Career of Ms Ana lulia Dascl

This is interesting as it shows that swimmers typically start competing when they are young.

### Research Question 7: The Average Olympian

**Question:** What is the average height and weight of athletes in a sport?

**Background:** From the previous query, it is evident that the youngest athletes are all 13 years old. One athlete, Ana Inulia Dascl, has a height of 183cm, which is peculiar for her age. This research question asks for the average height and weight of athletes in a sport. Clicking on the query from the index page directs to a page with the sports listed. This page was rendered with the SQL query ‘SELECT \* FROM SPORTS’.

**A screenshot of a sports survey

Description automatically generated**

Screen Capture 23 All Sports (Truncated)

Each sport has a different link that directs to a second page with the appropriate statistic, this is given by the query below.

**SQL used**: Left Join, Right Join, Aggregation with Avg() and filtering with Where clause with different values on the same query.

|  |
| --- |
| SELECT s.name, AVG(ag.height), AVG(ag.weight), AVG(ag.age)  FROM athlete\_game ag  INNER JOIN events e ON ag.game\_id = e.game\_id  INNER JOIN sports s ON e.sport\_id = s.sport\_id  WHERE s.sport\_id = ?; |

This query combines three tables: athlete\_game, events and sports. The placeholder, represented by ‘?’ within the where clause is substituted with the actual sport\_id upon the execution of the query. Finally, the average values for height, weight and age are computed.

**Result page:**

**A screenshot of a sports data

Description automatically generated**

Screen Capture 24 Result page for Research Question Seven

**Analysis:**

Each page describes the typical physical traits of an athlete in that sport.

### Research Question Eight: The Cross-season Olympian

**Question:** Were there athletes that participated in both Summer and Winter games and won medals?

**Background:** Given the distinct nature of the Summer and Winter Games, the conventional assumption is that medalling in both iterations of the Olympic Games is an unattainable feat. This research question challenges this assumption.

**SQL used**: Decorrelated Common Table Expression (CTE) query, Aggregation with COUNT(), Group By, Distinct and Having clause

|  |
| --- |
| WITH pro\_athletes AS (SELECT a.athlete\_id, a.name, a.sex  FROM athletes a  JOIN athlete\_game ag ON a.athlete\_id = ag.athlete\_id  JOIN games g ON ag.game\_id = g.game\_id  WHERE g.season IN ('Summer', 'Winter')  GROUP BY a.athlete\_id, a.name, a.sex  HAVING COUNT(DISTINCT g.season) = 2)  SELECT p.name, p.sex, e.event\_name, g.season, g.city, g.year, aem.medal, p.athlete\_id  FROM athlete\_event\_medal aem  INNER JOIN pro\_athletes p  ON aem.athlete\_id = p.athlete\_id  LEFT JOIN events e  ON aem.event\_id = e.event\_id  LEFT JOIN games g  ON e.game\_id = g.game\_id  WHERE aem.medal != 'No medal'; |

This query retrieves information about the professional athletes that participated in both Summer and Winter Olympics. The ‘WITH’ clause defines a Common Table Expression (CTE) named pro\_athletes, that is a temporary set that contains the id, name and sex of the athletes who have participated in both Summer and Winter Olympics. Then, it retrieves medal information for these athletes, as well as the game and event details.

**Result page:**

**A screenshot of a screen

Description automatically generated**

Screen Capture 25 Result page for Research Question Eight

**Analysis:** Given the distinct nature of the Summer and Winter Games, the conventional assuption is that medalling in both iterations of the Olympic Games is an unattainable feat. This research question overturns this assumption.

### Research Question Nine: Goldies

**Question:** Were there athletes that always ended up on the podium at all their events in London 2012 and won gold medals at both London and Rio 2016?

**Background:** Athletes improve across games as they get faster, stronger or more skilled. This research question seeks to identify a subset of athletes who placed at every event they participated in, in London, and earned a Gold at the Rio Games - a testament to their hard work that their skill remain unmatched across games.

**SQL used**: Correlated queries, Exists and Not Exists clause, filtering for more data with different ids

|  |
| --- |
| WITH london\_athletes AS (  SELECT DISTINCT a.name, ag.athlete\_id, ag.game\_id  FROM athlete\_game ag  INNER JOIN athletes a ON ag.athlete\_id = a.athlete\_id  INNER JOIN athlete\_event\_medal aem ON ag.athlete\_id = aem.athlete\_id  WHERE ag.game\_id = (  SELECT game\_id  FROM games  WHERE year = 2012 AND city = 'London'  )  AND aem.medal = 'Gold'  )  SELECT la.athlete\_id, la.name  FROM london\_athletes la  WHERE NOT EXISTS (  SELECT 1  FROM athlete\_game ag2  INNER JOIN athlete\_event\_medal aem2  ON ag2.athlete\_id = aem2.athlete\_id  WHERE ag2.athlete\_id = la.athlete\_id AND ag2.game\_id = la.game\_id  AND aem2.medal IN ('No medal')  )  AND EXISTS (  SELECT 1  FROM athlete\_game ag3  INNER JOIN athlete\_event\_medal aem3  ON ag3.athlete\_id = aem3.athlete\_id  WHERE ag3.athlete\_id = la.athlete\_id  AND aem3.medal IN ('Gold')  AND ag3.game\_id = (  SELECT game\_id  FROM games  WHERE year = 2016 AND city = 'Rio de Janeiro'  )  ); |

The query identifies athletes who won gold medals in London 2012. Then, the list is filtered to include only those who did always won medals in London and achieved at least one gold medal in Rio. The ‘EXISTS’ condition is used to check the existence returned by a subquery. [24]

**Result page:**

**A screenshot of a group of athletes

Description automatically generated**

Screen Capture 26 Result page for Research Question Nine

**Analysis:** This research question overturns the assumption that medalling in both iterations of the Olympic Games is an unattainable feat.

### Research Question Ten: NOC Participation

**Question:** Were there any NOCs that participated in the 2010 Vancouver games but not the 2014 Sochi games?

**Background:** While the Olympics Games showcase and celebrate athletes from across the globe, it is undeniable that socio-political factors can impact a nation’s decision to send its athletes abroad. [25] Consequently, this research question seeks to determine which countries attended the 2010 Olympic Games in Canada but not the 2014 Olympic Games in Russia.   
  
**SQL used**: Aggregation with Avg(), Inner Join and filtering with the WHERE clause

|  |
| --- |
| SELECT n1.noc\_id, n1.noc, n1.team  FROM noc n1  LEFT JOIN athlete\_game ag1  ON n1.noc\_id = ag1.noc\_id  WHERE ag1.game\_id = (SELECT game\_id FROM games WHERE year = '2010')  AND n1.noc\_id NOT IN (  SELECT n2.noc\_id  FROM noc n2  LEFT JOIN athlete\_game ag2  ON n2.noc\_id = ag2.noc\_id  WHERE ag2.game\_id = (SELECT game\_id FROM games WHERE year = '2014')  ); |

The query identifies all the NOCs and their participation records for the 2014 Olympics. Then, it filters out the NOCs that were present in the 2014 Olympics. This results in a list of NOCs that were absent from the 2014 Games but were present in 2010. Originally, an Except clause was used. However, Coursera Labs did not support the use of Except clause. Hence, the ‘NOT IN’ clause is used to indicate the absence of the NOC instead.

**Result page:**

|  |  |
| --- | --- |
| Screen Capture 27 Result page for Research Question Ten |  |

**Analysis:** The table above lists the NOCs absent from the Sochi Games in 2014. While there are no discernible commonalities, further investigation into these countries may shed light on why they were absent from the 2014 Games, despite their participation in the games four years prior.

# Conclusion

The implementation of a database structure gave valuable insights into the evolution of demographics and trends within the Olympic Games. Through the normalization of the dataset to Boyce-Codd Normal Form (BCNF), redundancies were mitigated, and data integrity was ensured. The formulation and subsequent analysis of ten research questions served to demonstrate the practical applications of a well-structured database. The utilisation of a web application to present the information provided a comprehensive approach to accessible data visualization.

In conclusion, this project represents a significant outcome of the learnings acquired over the past 10 weeks and exemplifies the principles of database.   
  
Future expansion could include incorporating data from other fields, such as the socio-political climate surrounding the games, and broadening the scope of analysis to encompass the full 120-year history of the Olympic Games.

# Works Cited

|  |  |
| --- | --- |
| [1] | D. Ganesan, “Why are the Olympics so special?,” 11 July 2024. [Online]. Available: https://www.straitstimes.com/sport/why-are-the-olympics-so-special. [Accessed 25 December 2024]. |
| [2] | Channel News Asia, “Gymnastics-Biles soars to third gold medal of Paris Olympics with vault triumph,” 3 August 2024. [Online]. Available: https://www.channelnewsasia.com/sport/gymnastics-biles-soars-third-gold-medal-paris-olympics-vault-triumph-4524821. [Accessed 25 December 2024]. |
| [3] | F. Karimi, “This sport is making its Olympics debut in Paris. Just don’t call it breakdancing,” 6 August 2024. [Online]. Available: https://edition.cnn.com/2024/08/06/sport/breaking-olympic-debut-paris-cec/index.html. [Accessed 25 December 2024]. |
| [4] | A. Moore, “5 Reasons Why the Olympics Are Important,” 1 August 2024. [Online]. Available: https://cnr.ncsu.edu/news/2024/08/why-the-olympics-are-important/#:~:text=%E2%80%9CWhat%20makes%20the%20Olympics%20unique,and%20Tourism%20Management%20at%20NC. [Accessed 25 December 2024]. |
| [5] | R. Griffin, “120 years of Olympic history: athletes and results,” 2018. [Online]. Available: https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results?select=athlete\_events.csv. [Accessed 26 December 2024]. |
| [6] | C. Bumbaca, “Nikki Hiltz, US track Olympian, embraces 'superpower' of being queer and running 'free',” 9 August 2024. [Online]. Available: https://www.usatoday.com/story/sports/olympics/2024/08/09/usa-track-athlete-nikki-hiltz-superpower-queer/74733630007/. [Accessed 26 December 2024]. |
| [7] | Sports Reference, “Sports Reference | Sports Stats, fast, easy, and up-to-date,” 2024. [Online]. Available: https://www.sports-reference.com/. [Accessed 26 December 2024]. |
| [8] | Dataful, “Sports - Olympic Games: Year-, Country-, Sport- and Athlete-wise Gold, Silver, and Bronze Medals Won, since 1896,” 2024. [Online]. Available: https://dataful.in/datasets/19672/. [Accessed 26 December 2024]. |
| [9] | International Olympic Committee, “Athletes,” 2024. [Online]. Available: https://olympics.com/en/athletes/. [Accessed 26 December 2024]. |
| [10] | International Olympic Committee, “Archived Documents,” 2024. [Online]. Available: https://odf.olympictech.org/project.htm. [Accessed 26 December 2024]. |
| [11] | Creative Commons, “CCC0 1.0 Universal,” 2024. [Online]. Available: https://creativecommons.org/publicdomain/zero/1.0/. [Accessed 26 December 2024]. |
| [12] | Geek for Geeks, “Difference between Pandas VS NumPy,” 22 July 2024. [Online]. Available: https://www.geeksforgeeks.org/difference-between-pandas-vs-numpy/. [Accessed 26 December 2024]. |
| [13] | Greg, “What is an associative entity in an ERD?,” 28 August 2024. [Online]. Available: https://www.gleek.io/blog/associative-entity-erd. [Accessed 26 December 2024]. |
| [14] | M. Jain, “BCNF in DBMS,” 30 Aoruk 2024. [Online]. Available: https://www.scaler.com/topics/bcnf-in-dbms/. [Accessed 26 December 2024]. |
| [15] | P. Pedamkar, “BCNF,” 24 March 2023. [Online]. Available: https://www.educba.com/bcnf/. [Accessed 26 December 2024]. |
| [16] | LinkedIn community, “What is the Boyce-Codd normal form and how is it used?,” 2024. [Online]. Available: https://www.linkedin.com/advice/0/what-boyce-codd-normal-form-how-used-skills-database-engineering. [Accessed 27 December 2024]. |
| [17] | L. Bennett, “Long vs Wide Data Tables,” 2024. [Online]. Available: https://www.thedataschool.co.uk/luke-bennett/long-vs-wide-data-tables/. [Accessed 27 December 2024]. |
| [18] | IBM, “What is ETL (extract, transform, load)?,” 2024. [Online]. Available: https://www.ibm.com/think/topics/etl. [Accessed 27 December 2025]. |
| [19] | Oracle, “15.2.9 LOAD DATA Statement,” 2024. [Online]. Available: https://dev.mysql.com/doc/refman/8.4/en/load-data.html. [Accessed 28 December 2024]. |
| [20] | Geeks for Geeks, “MySQL NULLIF() Function,” 2024. [Online]. Available: https://www.w3schools.com/sql/func\_mysql\_nullif.asp. [Accessed 28 December 2024]. |
| [21] | Bootstrap, “Build fast, responsive sites with Bootstrap,” 2024. [Online]. Available: https://getbootstrap.com/. [Accessed 29 December 2024]. |
| [22] | M. Eernisse, “Embedded JavaScript templating,” 2024. [Online]. Available: https://ejs.co/. [Accessed 29 December 2024]. |
| [23] | S. Darbhamulla, “Looking back, looking forward: The Olympics of adding new sports,” 12 August 2024. [Online]. Available: https://www.thehindu.com/sport/olympics/new-olympic-sports-and-events-additions/article68505265.ece. [Accessed 30 December 2024]. |
| [24] | Geek for Geeks, “SQL Exists,” 10 December 2024. [Online]. Available: https://www.geeksforgeeks.org/sql-exists/. [Accessed 30 December 2024]. |
| [25] | G. Goorno, “Should the Olympics ignore international conflict?,” 30 September 2024. [Online]. Available: https://www.tuftsdaily.com/article/2024/09/should-the-olympics-ignore-international-conflict. [Accessed 29 December 2024]. |

# Appendix A

## E/R Model

A diagram of a flowchart

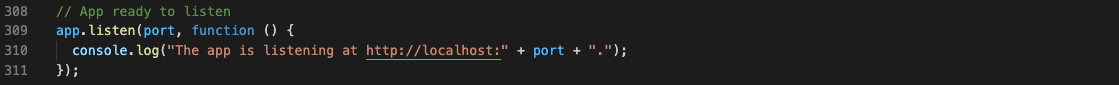
Description automatically generated

## Application Logic

### Importing libraries and making database connection

A computer screen shot of a program

Description automatically generated



### Index Page and Route for GET request to list all Sports

A screen shot of a computer

Description automatically generated

### Route for GET request to list athlete’s participation in all events

A screen shot of a computer

Description automatically generated

### Route for GET request to first research question

A screen shot of a computer

Description automatically generated

### Route for GET request to second research question

A screen shot of a computer

Description automatically generated

### Route for GET request to third research question

A computer screen with text on it

Description automatically generated

### Route for GET request to fourth research question

A computer screen shot of a program

Description automatically generated

### Route for GET request to fifth research question

A screenshot of a computer program

Description automatically generated

### Route for GET request to sixth research question

A screen shot of a computer

Description automatically generated

### Route for GET request to seventh research question

A screen shot of a computer program

Description automatically generated

### Route for GET request to eighth research question

A screen shot of a computer program

Description automatically generated

### Route for GET request to ninth research question

A screenshot of a computer screen

Description automatically generatedA computer screen shot of a black background

Description automatically generated

### Route for GET request to tenth research question

A screenshot of a computer program

Description automatically generated