
DATABASE REPORT

Olympics Tokyo 2020

NOVEMBER 17, 2024
MUHAMMAD HASHIR SHAKEEL
21634493

Table of Contents

Introduction	2
Design of Database	2
Why these entities?.....	2
Relationships between entities:.....	3
ER-Diagram made:	6
Data Description	7
Assumptions	9
Implementation of Database	10
How I implemented the data?.....	10
Sample Data and Data Insertion	13
Use of Database	13
Level 1	13
Level 2	15
Advanced Features	17
Implementation of Database Connectivity	18
Discussion	19
Challenges Faced	19
Limitations.....	19

Introduction

The Olympic Games Tokyo 2022 was an occasion for all of those nations' athletes, coaches, technical staff and fans to come together, and feel the passion and dedication that exists to perform at their best. Which means that to manage an event of this scale you need a reliable and structured database system to capture in detail information from getting athletes to register and participate to medals tallies to discipline management.

The goal of this project is to develop and deploy a database solution that will manage different pieces of the organization of the Tokyo 2022 Olympics. This system collects key information around athletes, what events they have participated in, their coaches, disciplines, countries, medals, technical staff, and other details. It follows athlete performances, medals and a stakeholder network for each performances offering an overview of the system behind each performance.

Activities include working out an ER diagram to represent the entity relationships and create the relational schema, writing SQL scripts for creating and populating the database and using Python to item the connectivity with the database. Normalization, data integrity and data retrieval are emphasized in this project to facilitate easy accessibility of data in detail about the games by stakeholders.

Design of Database

Why these entities?

Country: It stores country code, name and total medals to represent the participating nations and the total medals earned by each nation.

Discipline: Organizes the Olympics, capturing various sports disciplines, and managing discipline specific details.

Athlete: Stores personal details of competitors, and links them to their country and disciplines in one central entity representing competitors.

Participation: Tracked each event an athlete participated in, across multiple events and within multiple disciplines.

Medals: It keeps track of medals awarded: type, date, event, and which athlete was participating when the medal was won.

Coach: These are coaches training athletes, theirs include personal details as well as even associations with their country and disciplines.

Technical_Staff: It includes people who manage a certain discipline and record the discipline, plus the discipline under which that person manages.

Participation is a weak entity, because it cannot be uniquely identified by its own attributes alone. Instead, its identification depends on the Athlete entity existing. A weak entity, in database design, is one that does not have a primary key that will uniquely identify each record on its own and must rely on a foreign key relationship with another (parent) entity.

Entities and attributes in the database include:

- Country: country_name, country_code
- Discipline: discipline_code, discipline_name
- Athlete: athlete_id, name, gender, birth_date, birth_country, birth_place, sport_class
- Participation (Weak Entity): participation_id, athlete_id, event
- Medals: medal_id, medal_type, medal_date, event, participation_id
- Coach: coach_id, name, gender, birth_date, function, country_code, discipline_code
- Technical Staff: staff_id, name, gender, birth_date, discipline_code

Relationships between entities:

1. Athlete-Country Relationship:

- An athlete represents a country, which is identified by the country code.

2. Athlete-Participation Relationship:

- An athlete participates in multiple events, and each participation is recorded as an event entry.

3. Athlete-Discipline Relationship:

- An athlete competes in one or more disciplines, such as swimming, athletics, etc.

4. Coach-Discipline:

- Coach trains Athletes in for many disciplines.

5. Medals-Discipline Relationship:

- Medals are awarded for different disciplines, which indicates the field in which the medal was earned.

6. Medals-Country Relationship:

- Medals belong to a particular country, representing the achievements of athletes from that country.

7. Technical Staff-Discipline Relationship:

- Technical staff members manage in specific disciplines

8. Medal-Participation:

- When there are participants Medals are earned for performance.

9. Coach-Country:

- Coach belongs to a country, coach represents for their country

Cardinality:

1. Athlete-Country Relationship

- Cardinality: Many-to-One (N:1)

- Notation: Athlete (N) : Country (1)
- Explanation: Each athlete represents one country, but a country can have many athletes representing it. This means multiple athletes (N) are associated with a single country (1).

2. Athlete-Participation Relationship

- Cardinality: One-to-Many (1:N)
- Notation: Athlete (1) : Participation (N)
- Explanation: An event participation can be made for each event so that the athlete can compete in multiple events. In other words, each athlete (1) can be participating in many (N) teams.

3. Athlete-Discipline Relationship

- Cardinality: Many-to-Many (M:N)
- Notation: Athlete (M) : Discipline (N)
- Explanation: Athletes can compete in multiple disciplines, and there can be multiple athletes in any one discipline. Therefore it is that many athletes (M) are related to many disciplines (N).

4. Coach-Discipline Relationship

- Cardinality: Many-to-Many (M:N)
- Notation: Coach (M) : Discipline (N)
- Explanation: Sometimes coaches can coach multiple athletes with the same sport or across multiple sports and sometimes coaches train athletes in multiple disciplines. many coaches (M) who happen to work on many different disciplines (N).

5. Medals-Discipline Relationship

- Cardinality: Many-to-One (N:1)
- Notation: Medals (N) : Discipline (1)
- Explanation: Each medal is awarded for a specific discipline, but a discipline can have many medals awarded in it. So, multiple medals (N) are associated with a single discipline (1).

6. Medals-Country Relationship

- Cardinality: Many-to-One (N:1)

- Notation: Medals (N) : Country (1)
- Explanation: For each medal we have the country of the player who won the medal. For instance, an athlete can win one medal, but many countries I win N medals; thereby N is linked to 1 country.

7. Technical Staff-Discipline Relationship

- Cardinality: Many-to-One (N:1)
- Notation: Technical Staff (N) : Discipline (1)
- Explanation: In one discipline, each technical staff member would manage, but one technical staff member can have more than one discipline. Consequently, one discipline (1) is tied to many technical staff members (N).

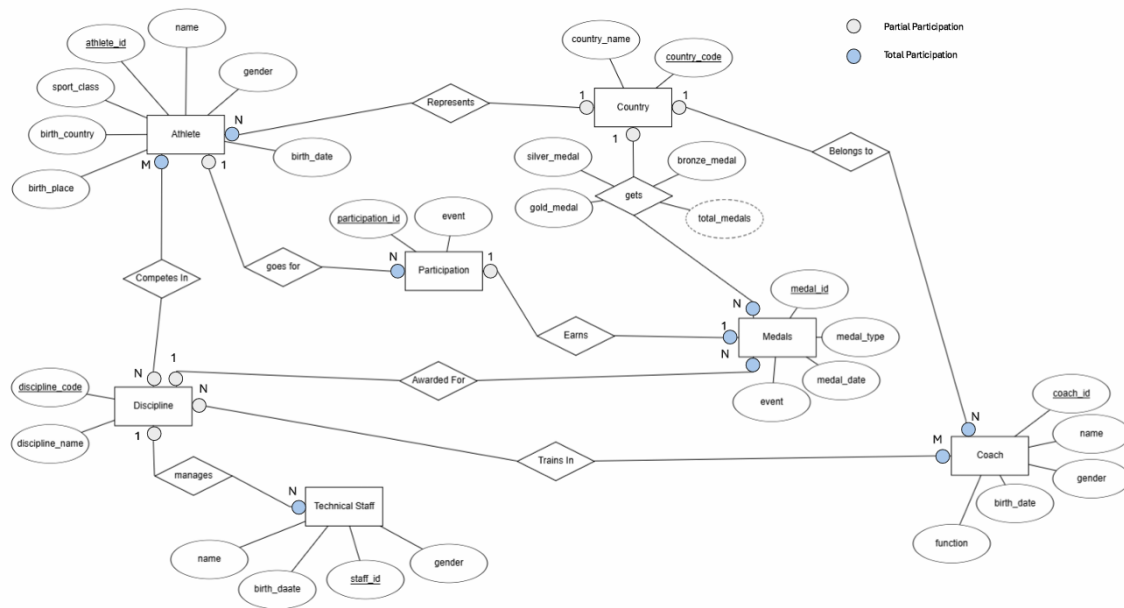
8. Medal-Participation Relationship

- Cardinality: One-to-One (1:1) or Zero-to-One (0:1)
- Notation: Participation (1) : Medal (0 or 1)
- Explanation: Each participation may result in one medal or none at all. A medal is awarded for a specific participation. Therefore, a participation is associated with zero or one medal, and each medal is linked to one participation.

9. Coach-Country Relationship

- Cardinality: Many-to-One (N:1)
- Notation: Coach (N) : Country (1)
- Explanation: Each coach represents one country, but a country can have many coaches representing it. This means multiple coaches (N) are associated with a single country (1).

ER-Diagram made:



Relational Schema:

Country(country_code, country_name, total_medals)

Discipline(discipline_code, discipline_name)

Athlete(athlete_id, name, gender, birth_date, birth_place, birth_country, sport_class, country_code, discipline_code)

FK_country_code REF Country(country_code)

FK_discipline_code REF Discipline(discipline_code)

Participation(participation_id, athlete_id, event)

FK_athlete_id REF Athlete(athlete_id)

Medals(medal_id, medal_type, medal_date, event, participation_id)

FK_participation_id REF Participation(participation_id)

Coach(coach_id, name, gender, birth_date, function, country_code, discipline_code)

FK_country_code REF Country(country_code)

FK_discipline_code REF Discipline(discipline_code)

Technical_Staff(staff_id, name, gender, birth_date, discipline_code)

FK_discipline_code REF Discipline(discipline_code)

Data Description

Country Table

Column Name	Data Type	Constraints	Description
country_code	Char(3)	Primary Key	Unique code identifying each country
country_name	VARCHAR(225)		Name of the country
total_medals	INT		Total number of medals won by the country

Discipline Table

Column Name	Data Type	Constraints	Description
discipline_code	Char(3)	Primary Key	Unique code identifying each discipline
discipline_name	VARCHAR(225)		Name of the discipline

Athlete Table

Column Name	Data Type	Constraints	Description
athlete_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each athlete
name	VARCHAR(255)		Full name of the athlete
gender	ENUM('Male', 'Female')		Gender of the athlete
birth_date	DATE		Birth date of the athlete
birth_place	VARCHAR(255)		Birthplace of the athlete
birth_country	VARCHAR(255)		Birth country of the athlete
sport_class	VARCHAR(255)		Sport class or category of the athlete
country_code	CHAR(3)	FOREIGN KEY (country_code) REFERENCES Country(country_code)	Country the athlete represents
discipline_code	CHAR(3)	FOREIGN KEY (discipline_code)	Discipline the athlete competes in

		REFERENCES Discipline(discipline_code)	
--	--	---	--

Participation Table

Column Name	Data Type	Constraints	Description
participation_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each participation
athlete_id	INT	FOREIGN KEY (athlete_id) REFERENCES Athlete(athlete_id)	Athlete participating in the event
event	VARCHAR(255)		Name of the event the athlete participates in

Medals Table

Column Name	Data Type	Constraints	Description
medal_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each medal
medal_type	ENUM('Gold', 'Silver', 'Bronze')		Type of medal awarded
medal_date	DATE		Date when the medal was awarded
event	VARCHAR(255)		Event for which the medal was awarded
participation_id	INT	FOREIGN KEY (participation_id) REFERENCES Participation(participation_id)	Participation associated with the medal

Coach Table

Column Name	Data Type	Constraints	Description
coach_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each coach
name	VARCHAR(255)		Full name of the coach
gender	ENUM('Male', 'Female')		Gender of the coach
birth_date	DATE		Birth date of the coach
function	VARCHAR(255)		Role or function of the coach
country_code	CHAR(3)	FOREIGN KEY (country_code)	Country the coach represents

		REFERENCES Country(country_code)	
discipline_code	CHAR(3)	FOREIGN KEY (discipline_code) REFERENCES Discipline(discipline_code)	Discipline the coach is associated with

Technical_Staff Table

Column Name	Data Type	Constraints	Description
staff_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each technical staff member
name	VARCHAR(255)		Full name of the staff member
gender	ENUM('Male', 'Female')		Gender of the staff member
birth_date	DATE		Birth date of the staff member
discipline_code	CHAR(3)	FOREIGN KEY (discipline_code) REFERENCES Discipline(discipline_code)	Discipline the staff manages

Assumptions

Completeness of Relationships

For example, in our case we might have assumed that every Athlete is associated with precisely one Country and one Discipline. In other words, did not include athletes for whom I don't know a registered discipline or country if that athlete is not a refugee or independent athlete.

Attributes are Fully Defined

I assumed attributes such as Name, Gender, birth_date, and medal_type were well defined (not open to interpretation for ex: no ambiguity in naming convention like USA vs. United States).

Participation and Medal Relationship

I considered that for every participating Event an Athlete will also receive a medal. As a result, medals cannot exist without participation records.

Tables with Auto-Incremented Primary Keys:

- Athlete
- Participation

- Medals Coach
- Technical_Staff

Reasoning Behind the Assumptions:

In the above tables the primary keys were made auto incremental with the help of AUTO_INCREMENT feature.

This decision was based on the following reasons:

Absence of Natural Primary Keys: The files before joining provided did not contain unique keys or primary keys to uniquely identify records in these tables. Such properties as name and events are not distinctive and can cause troubles with data definition.

Enhancing Data Integrity: Using auto-increment primary key field will assure each record to be unique and clearly recognizable. This helps eliminate unnecessary replication and ensures that the data in the database remains secure and is not contaminated.

Ease of Use: Auto increment identities help to avoid explicit generation of new numbers each time we want to insert records into the table. This makes data insertion easier and cuts the possibility of error incidences. **Facilitating Relationships:** Use of unique primary keys is very important as far as the creation of foreign key relations between any tables is concerned. Since there will be an assurance of uniqueness, referencing records from one table to the other becomes easy something that enhances relational integrity.

Database Credibility: Use of auto-incrementation for developing primary keys will increase the reliability of the database since it will be developed under the recommended standards. It ensures accuracy and quick access to the data and also it standardizes the results in data analysis.

Implementation of Database

How I implemented the data?

To put into practice the database for the Tokyo 2020 Olympics management system, I developed SQL statements to create the database and bring the tables as identified in the aforementioned ER diagram into creation. It used select queries, and among the SQL code created tables were Country, Discipline, Athlete, Participation, Medals, Coach, and Technical Staff. Every table was laid out with correct columns, data types, primary keys as well as foreign keys constraints in order to depict the existing entity relationships in the tables. For this choice, I had to study CSV files received from the conventional Tokyo 2022 Olympic Games data to recognize the notable columns required for each table. This real-world data was then used in defining the mandatory attributes to include in the record and in further developing the database schema to correlate with actual Olympic records. Further, I highlighted and constructed other tables which were absent in the data set but can be useful for enhancing the Databases performance: Country and Participation which were created by extracting original data.

Below is the images of implementing the tables in mysql environment.

```
-- Country Table
CREATE TABLE Country (
  country_code CHAR(3) PRIMARY KEY,
  country_name VARCHAR(255),
  total_medals INT
);

-- Discipline Table
CREATE TABLE Discipline (
  discipline_code CHAR(3) PRIMARY KEY,
  discipline_name VARCHAR(255)
);

-- Athlete Table
CREATE TABLE Athlete (
  athlete_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255),
  gender ENUM('Male', 'Female'),
  birth_date DATE,
  birth_place VARCHAR(255),
  birth_country VARCHAR(255),
  sport_class VARCHAR(50),
  country_code CHAR(3),
  discipline_code CHAR(3),
  FOREIGN KEY (country_code) REFERENCES Country(country_code),
  FOREIGN KEY (discipline_code) REFERENCES Discipline(discipline_code)
);
```

```
-- Participation Table
CREATE TABLE Participation (
  participation_id INT AUTO_INCREMENT PRIMARY KEY,
  athlete_id INT,
  event VARCHAR(255),
  FOREIGN KEY (athlete_id) REFERENCES Athlete(athlete_id)
);

-- Medals Table
CREATE TABLE Medals (
  medal_id INT AUTO_INCREMENT PRIMARY KEY,
  medal_type ENUM('Gold', 'Silver', 'Bronze'),
  medal_date DATE,
  event VARCHAR(255),
  participation_id INT,
  FOREIGN KEY (participation_id) REFERENCES Participation(participation_id)
);

-- Coach Table
CREATE TABLE Coach (
  coach_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255),
  gender ENUM('Male', 'Female'),
  birth_date DATE,
  function VARCHAR(255), -- Using backticks for reserved keyword
  country_code CHAR(3),
  discipline_code CHAR(3),

  -- Foreign Key Constraints
  FOREIGN KEY (country_code) REFERENCES Country(country_code),
  FOREIGN KEY (discipline_code) REFERENCES Discipline(discipline_code)
);
```

```
-- Technical Staff Table
CREATE TABLE Technical_Staff (
  staff_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255),
  gender ENUM('Male', 'Female'),
  birth_date DATE,
  discipline_code CHAR(3),
  FOREIGN KEY (discipline_code) REFERENCES Discipline(discipline_code)
);
```

Expected tables created:

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| country_code | char(3)    | NO   | PRI | NULL    |       |
| country_name | varchar(255) | YES  |     | NULL    |       |
| total_medals | int        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| discipline_code | char(3)    | NO   | PRI | NULL    |       |
| discipline_name | varchar(255) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| athlete_id | int        | NO   | PRI | NULL    | auto_increment |
| name       | varchar(255) | YES  |     | NULL    |               |
| gender     | enum('Male','Female') | YES  |     | NULL    |               |
| birth_date | date       | YES  |     | NULL    |               |
| birth_place | varchar(255) | YES  |     | NULL    |               |
| birth_country | varchar(255) | YES  |     | NULL    |               |
| sport_class | varchar(50) | YES  |     | NULL    |               |
| country_code | char(3)    | YES  | MUL | NULL    |               |
| discipline_code | char(3)    | YES  | MUL | NULL    |               |
+-----+-----+-----+-----+-----+-----+

```

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| participation_id | int        | NO   | PRI | NULL    | auto_increment |
| athlete_id      | int        | YES  | MUL | NULL    |               |
| event           | varchar(255) | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| medal_id | int        | NO   | PRI | NULL    | auto_increment |
| medal_type | enum('Gold','Silver','Bronze') | YES  |     | NULL    |               |
| medal_date | date       | YES  |     | NULL    |               |
| event     | varchar(255) | YES  |     | NULL    |               |
| participation_id | int        | YES  | MUL | NULL    |               |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| coach_id | int        | NO   | PRI | NULL    | auto_increment |
| name     | varchar(255) | YES  |     | NULL    |               |
| gender   | enum('Male','Female') | YES  |     | NULL    |               |
| birth_date | date       | YES  |     | NULL    |               |
| function  | varchar(255) | YES  |     | NULL    |               |
| country_code | char(3)    | YES  | MUL | NULL    |               |
| discipline_code | char(3)    | YES  | MUL | NULL    |               |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| staff_id | int        | NO   | PRI | NULL    | auto_increment |
| name     | varchar(255) | YES  |     | NULL    |               |
| gender   | enum('Male','Female') | YES  |     | NULL    |               |
| birth_date | date       | YES  |     | NULL    |               |
| discipline_code | char(3)    | YES  | MUL | NULL    |               |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

All of the code above is saved in file [create_databases.sql](#)

To use : `_SOURCE create_database.sql`

Sample Data and Data Insertion

For the sample data the database, the information came from the same dataset of Tokyo 2022 Olympic Games. This information gave enough information about the athletes involved, the events, the medals, the coaches and the technical crew which made it easier to simulate the Olympic Games in the database. The data is placed into the SQL tables through INSERT statements for every table I made, following the proper structure of the column for each table and making sure that referential integrity is observed throughout the organization of the database. For instance, this process involved mapping the data obtained from the CSV files sorted based on the proposed database structure with elements like converting the data type from date to string and so on. Olympic data source is valid and helps gaining a better picture of the games through the database.

To insert all the data **SOURCE** `insert_sample_values.sql`

Example of inserts used:

```
INSERT INTO Country (country_name, country_code, total_medals) VALUES
('People's Republic of China', 'CHN', 207),
('Great Britain', 'GBR', 124),
('United States of America', 'USA', 104),

-- Insert statements for each unique discipline with its discipline code
INSERT INTO Discipline (discipline_code, discipline_name) VALUES
('JUD', 'Judo'),
('SWM', 'Swimming'),
('WBK', 'Wheelchair Basketball'),
('VBS', 'Sitting Volleyball'),
('PWL', 'Powerlifting'),
('ATH', 'Athletics'),
```

To view the data after insertion following is already in the .sql file:

```
-- Show's all the inserted values in the table's
SELECT * FROM Country;
SELECT * FROM Discipline;
SELECT * FROM Athlete;
SELECT * FROM Participation;
SELECT * FROM Medals;
SELECT * FROM Coach;
SELECT * FROM Technical_Staff;
```

Use of Database

Level 1

Question 1: Retrieve athletes of a specific gender

- Purpose: Extracts names and sport classes of all male athletes.

- Importance: Facilitates stakeholders to assess the level of participation of different genders in different sport classes.
- Evidence: The query returns table of male athletes which is to be matched with their respective sport class.

athlete_name	sport_class
AAJIM Munkhbat	B3
ABARZA Alberto	S2
ABASLI Namig	B2
ABASSI Mostefa	2.5
ABBAD Abderracuf	1.0
ABDALLA Ashraf	VS1
ABDELHADY Hany	PWL
ABDELLATIF Abdelnaby	VS1
ABDELMAJID Taha	PWL
ABDI Hassan	2.0
ABDULKHABIBOV Azizbek	KL2
ABDULLAKHANLI Kanan	B3
ABDULRAHEEM Kovan	F41
ABIDOGUN Ola	T47
ABLINGER Walter	H3
ABOUELKHIR Metawa	VS1
ABOUELYAZEID Mohamed	VS1
ABRAHAM de DEUS Michel Gustavo	T46
ABRAHAM GEBRU Daniel	C5
ABRAHAM GEBRU Daniel	C5
ABRAHAMS David Henry	S13,SB13,SM13
ABUZARLI Abulfaz	K44
ACAR Deniz	2.5
ACCARDI Federico	B1
ACEITUNO Herbert	PWL
ACHENZA Giovanni	PTWC1
ACOSTA MANJARRES Yamil David	T12
ACOSTA RODRIGUEZ Adolfo	B1
ACOSTA SOTO Omar Jose	T36
ADDESI Pierpaolo	C5
ADESOKAN Yakubu	PWL
ADOLPHE Timothee	T11
AFRI Mahdi	T12
AFROOZ Saeid	F34
AGALAKOV Roman	S13
AGRIPINO dos SANTOS Julio Cesar	T11

Question 2: Retrieve all coaches born after January 1, 1980

- Purpose: Lists names, genders, and roles of coaches born after 1980.
- Importance: Is useful in determining the younger coaches for youth-related programmes and for setting up coaching-enhancement programmes.
- Evidence: The output displays coaches born after January 1, 1980, grouped by their roles and genders.

coach_name	gender	function
ARIMA Masato	Male	Coach
BAN Nicole	Female	Coach
BARGUEIRAS MARTINEZ Jesus David	Male	Coach
CHOUDHRY Gaz	Male	Coach
COTE Patrick	Male	Coach
DUBBERLEY Brad	Male	Coach
DUBROVIN Anton	Male	Coach
GUENTHER Johannes	Male	Coach
KIM Hyun Woo	Male	Coach
KRYVENKO Kseniia	Female	Coach
KUBO Yoshihiro	Male	Coach
KUDO Rikiya	Male	Coach
LAWRENCE Matt	Male	Coach
LAWRENCE Matt	Male	Coach

14 rows in set (0.00 sec)

Question 3: Retrieve events where Gold medals were awarded

- Purpose: Locates all the events where victories in golds for athletes.

- Importance: Helpful for the analysis of high-performance events, and the trends, related to gold medal performance.
- Evidence: Contains information on events linked to gold medals.

```

+-----+
| event |
+-----+
| Men's Judo 60kg |
| Men's Wheelchair Basketball |
| Women's Wheelchair Basketball |
| Women's Judo 48kg |
| Men's Powerlifting -88kg |
| Men's Canoe Sprint KL2 |
| Women's Powerlifting -73kg |
| Women's 200m T35 |
| Men's Volleyball Sitting |
| Men's Road Cycling C5 |
| Women's Table Tennis Class 5 |
| Men's Football 5-a-side |
| Men's Triathlon PTWC1 |
| Men's 200m T36 |
| Men's Powerlifting -49kg |
| Women's Boccia BC4 |
| Men's 1500m T12 |
| Women's Shot Put F20 |
+-----+
18 rows in set (0.01 sec)

```

Question 4: Retrieve disciplines based on a specific discipline code

- Purpose: Extracts discipline names corresponding to the code 'ATH'.
- Importance: Facilitates easy determination of disciplines for certain codes which helps in focused data handling.
- Evidence: Shows the name of a discipline corresponds to the given code.

```

+-----+
| discipline_name |
+-----+
| Athletics       |
+-----+

```

Level 2

Question 5: Retrieve athletes along with their associated countries and disciplines

- Purpose: Links athletes to their countries and disciplines.
- Importance: Enables an understanding of athlete participation in disciplines and coverage of the relationships between countries.
- Evidence: provides a table showing the athlete's name, country and discipline.

athlete_name	country_name	discipline_name
ABDELLI Rima	Tunisia	Athletics
ABDULRAHEEM Kovan	Iraq	Athletics
ABIDOGUN Ola	Great Britain	Athletics
ABILKHASSYMKYZY Saltanat	Kazakhstan	Athletics
ABRAHAM de DEUS Michel Gustavo	Brazil	Athletics
ACOSTA MANJARRES Yamil David	Colombia	Athletics
ACOSTA SOTO Omar Jose	Colombia	Athletics
ADAMS Lisa	New Zealand	Athletics
ADENEGAN Kare	Great Britain	Athletics
ADOLPHE Timothee	France	Athletics
ADZHAMETOVA Leilia	Ukraine	Athletics
AFRI Mahdi	Morocco	Athletics
AFROOZ Saeid	Islamic Republic of Iran	Athletics
AGBLEMAGNON Gloria	France	Athletics
AGRIPINO dos SANTOS Julio Cesar	Brazil	Athletics
ADONINA Daria	RPC	Boccia
ABLINGER Walter	Austria	Cycling Road
ABRAHAM GEBRU Daniel	Netherlands	Cycling Road
ADDESI Pierpaolo	Italy	Cycling Road
AERE Katia	Italy	Cycling Road

Question 6: Retrieve coaches from specific along with their disciplines

- Purpose: Provides lists of specialist coaches from Australia and the speciality of their discipline as example.
- Importance: Helps track coaches from specific countries
- Evidence: The output shows Australian coaches, their disciplines, and associated details.

coach_name	country_name	discipline_name
CORR Peter	Australia	Goalball
DUBBERLEY Brad	Australia	Wheelchair Rugby
FRIDAY Craig	Australia	Wheelchair Basketball
FURNESS Shane	Australia	Wheelchair Basketball

Question 7: Retrieve details of medals along with athlete and country information

- Purpose: Offers medal information related to the athletes as well as their countries of origin.
- Importance: Keeps records of medals per country and per performance standards.
- Evidence: Output contains medal type, event, athlete name, and country.

medal_type	event	athlete_name	country_name
Gold	Men's Judo 60kg	AAJIM Munkhbat	Mongolia
Silver	Men's 100m Freestyle S2	ABARZA Alberto	Chile
Bronze	Men's Judo 66kg	ABASLI Namig	Azerbaijan
Gold	Men's Wheelchair Basketball	ABASSI Mostefa	Algeria
Silver	Men's Wheelchair Basketball	ABBAD Abderracuf	Algeria
Bronze	Men's Volleyball Sitting	ABDALLA Ashraf	Egypt
Gold	Women's Wheelchair Basketball	ABDELLALI Samiha	Algeria
Silver	Women's Powerlifting -49kg	ABDELAZIZ Gihan	Egypt
Bronze	Men's Powerlifting -65kg	ABDELHADY Hany	Egypt
Gold	Women's Judo 48kg	ABDELLAOUI Cherine	Algeria
Silver	Men's Volleyball Sitting	ABDELLATIF Abdelnaby	Egypt
Bronze	Women's Shot Put F40	ABDELLI Rima	Tunisia
Gold	Men's Powerlifting -88kg	ABDELMAJID Taha	Egypt

Question 8: Calculate the total number of medals won by each athlete

- Purpose: It is used to calculate the summary of the medals garnered by each of the athletes.
- Importance: Helps evaluate individual performances.
- Evidence: Outputs a list of athletes with their respective medal counts.

Question 9: Calculate the total number of medals won by each country

- Purpose: Gives general figure of medals received by each country.
- Importance: Reflects overall performance by nation.
- Evidence: Outputs a ranked list of countries and their medal counts.

Question 10: Retrieve all technical staff managing Sitting Volleyball

- Purpose: Identifies technical staff associated with Sitting Volleyball.
- Importance: Manages staff tracking for certain type of sports.
- Evidence: Explains the names of technical staff, their genders, and disciplines connected with the particular type of work.

As images are too large for clear look, Please refer to images folder

Advanced Features

Procedure: AddAthlete

Purpose: Ease the addition of the new athlete into the database by collecting all the necessary details that may include name, gender, country and class of sport in one function.

Importance: Helps minimize the mistakes when entering new data into spread sheets. It guarantees proper and uniform data input for the athletes.

Evidence:

- Example usage: CALL AddAthlete('John Doe', 'Male', '1990-01-15', 'New York', 'USA', 'Class A', 'USA', 'ATH');

```
mysql> CALL AddAthlete('John Doe', 'Male', '1990-01-15', 'New York', 'USA', 'Class A', 'USA', 'ATH');
Query OK, 1 row affected (0.01 sec)
```

Procedure: GetTotalMedalsByCountry

Purpose: Returns the count of medals won in any of the given category by the given country.

Importance: Rapidly accesses a brief overview of the performance of a country. It enables support of decisions and data analysis.

Usage:

- CALL GetTotalMedalsByCountry('USA', @total_medals); SELECT @total_medals; to show the total medals of a specific country

Trigger: PreventDuplicateAthletes

Purpose: To avoid the problem of generating two or more records for different athletes with the same name.

Importance:

- Ensures data integrity by eliminating duplicate records.
- Reduces errors in reporting and analysis caused by duplicate data.

Evidence: Inserting an athlete named "John Doe" when there is an entry with the same name already present. Result: The system returns an error message: Duplicate athlete name not allowed.

Trigger: UpdateTotalMedals

Purpose: Reduces search time for updating medals data:

- It records medal information in the MedalInsertionLog table
- Replaces the total medals earned with new parameters for the country connected by incrementing.

Importance: Saves time to retrieve the medal log and country medal counts than having to update it manually.

Evidence:

- Example scenario: Adding a medal to the Medals table.
- Result:
 - The MedalInsertionLog table logs the new medal details.
 - The Country table's total_medals field is incremented by 1.

Implementation of Database Connectivity

The connection.py contains a detailed implementation to the python implementation for database related activity. Here's how the database connection is established and utilized:

Description of the Implementation

1. Configuration Loading:
 - It maintains the script from being compromised and make the database credentials (host, database, user, password) in a different JSON file (db_config.json)
2. Connection Establishment:
 - The script uses the mysql.connector module to connect to the MySQL database.
 - The credentials loaded from the JSON file are passed to the mysql.connector.connect method.
3. Executing Queries:
 - After establishing a connection, the script demonstrates executing a sample SQL query (SELECT * FROM Athlete LIMIT 5) and fetching the results.
 - The results are printed in the terminal.
4. Error Handling:
 - The script includes error handling to catch and display any connection issues.
 - It ensures the database connection is closed properly in the finally block.

Discussion

In this project you can see how I have managed to design and implement a full database system consisting of data about athletes, events, medals, coaches and disciplines for the Tokyo 2022 Olympics. It comes with ER diagram for relationships between entities, SQL scripts for database creation and populating it, and for integration with Python to secure database interaction. Data management and data integrity was made possible with advanced elements such as stored procedures, triggers and views. Sample queries provided some very useful insights such as Medal tallies, Athlete performances, and Discipline specific data.

Challenges Faced

Some of the challenges the project faced included designing an ER diagram from raw CSV files without pre defined relationships, creating a suitable relational schema with predefined constraints and getting SQL to work along with Python without compromising security. To address these challenges I went through careful data analysis, iterated through and tested, and then controlled credential handling through a configuration file. The learning was made more immersive by the hurdles, and the system became more reliable through each hurdle.

Limitations

However, the project has some limitations. It is specific to MySQL and does not attempt expansions to different database frameworks, for instance, MongoDB or MariaDB, which would give dealing with unexpected information and databases scaling. This also involves lack of user friendly interface, that can cause problems for non technical users trying to perform actions via the system. In addition, the system provides no automation for bulk data operations as data insertion and deletion must be performed manually.

Future Improvements

The system can be enhanced with several improvements, which will better functional and usability. Additional stored procedures and Python automation scripts to handle bulk data handling and simple things like medal tally updates would make for a cleaner type of input. Considering using other database systems, like MongoDB might scale and adapt better to work with unstructured data. The system could be easy to use for non technical users meaning developing one with a graphical or web based interface would be possible. Then performance optimization techniques like indexing would bloom the data retrieval and optimize the system performance.

References

csv information from

[Tokyo 2020 Olympic Summer Games](#)

JSON file usage from python code from

[json — JSON encoder and decoder — Python 3.13.0 documentation](#)