Assignment: Report

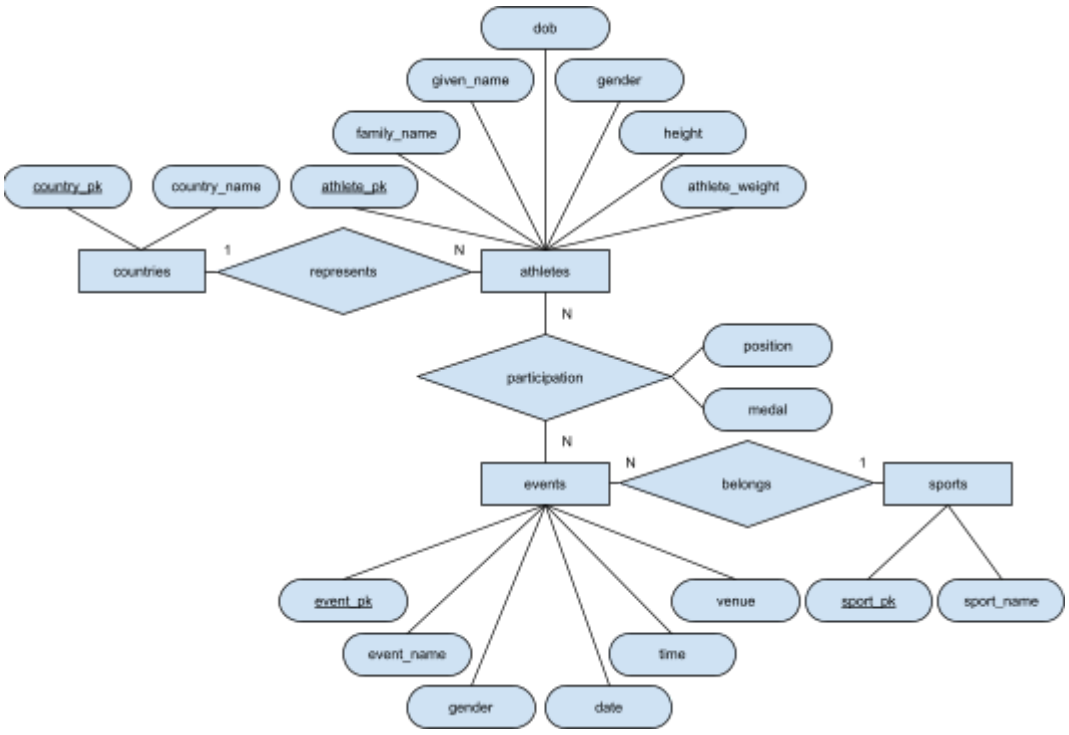
Introduction

Knowledge in regards to the design, implementation and usage of relational databases was utilised to solve the problem of exploring the Tokyo 2020 Summer Olympics. Specifically, gaining knowledge in regards to the details of the game's participating countries and athletes, sports, events and results. In order to further enhance usability, advanced SQL features such as stored procedures and views were designed and implemented. Additionally, connectivity with a programming language interface (Python) was also established. Design and implementation decisions were thoroughly reflected upon, challenges were postulated and documented, and potential improvements were theorised.

Database Design

Illustrative Materials

ER Diagram



Entity Sets

Entity Sets	**Keys**	**Attributes**
countries	country_pk	country_name
sports	sport_pk	sport_name
events	event_pk, sport_fk	gender, event_name, date, time, venue
athletes	athlete_pk, country_fk	family_name, given_name, gender, dob, height, athlete_weight

Relationship Sets

Relationship Sets	**Involvement**	**Attributes**
represents	athletes, countries	
participation	athletes, events	position, medal
belongs	events, sports	

Constraints

Relationships	**Cardinality**	**Participation / Other Constraints***
represents	1:N – An athlete can represent at most one country; A country can be represented by multiple athletes	athletes – total, countries – total (An athlete represents a country; a country must have representing athletes)*
participation	M:N – An athlete can participate in many events; an event can have multiple participating athletes	athletes – total, events – total (An athlete participates in at least one event; An event must have participating athletes)*
belongs	1:N – An event can belong to at most one sport; a sport can have many events belonging to it	sports – total, event – total (A sport has at least one event; An event belongs to a sport)*

** Otherwise, that particular country/athlete/sport would not have been present in the Tokyo 2020 Summer Olympics.*

Data Types

countries				
Attribute	**Type**	**Key**	**Constraints**	**Description**
country_pk	CHAR(3)	Primary	NOT NULL	Country code, unique
country_name	VARCHAR(36)		NOT NULL, UNIQUE	Country name

sports				
Attribute	**Type**	**Key**	**Constraints**	**Description**
sport_pk	CHAR(6)	Primary	NOT NULL	Sport ID, unique
sport_name	VARCHAR(36)		NOT NULL, UNIQUE	Sport name

events				
Attribute	**Type**	**Key**	**Constraints**	**Description**
event_pk	CHAR(6)	Primary	NOT NULL	Event ID, unique
sport_pk	CHAR(6)	Foreign	NOT NULL	Event sport ID, unique
gender	CHAR(1)			Event participants' gender
event_name	VARCHAR(36)		NOT NULL, UNIQUE	Event name
event_datetime	DATETIME		NOT NULL	Event date and time
venue	VARCHAR(36)		NOT NULL	Event venue

athletes				
Attribute	**Type**	**Key**	**Constraints**	**Description**
athlete_pk	CHAR(6)	Primary	NOT NULL	Athlete ID, unique
country_pk	CHAR(3)	Primary	NOT NULL	Athlete country code, unique
family_name	VARCHAR(36)		NOT NULL	Athlete family name
given_name	VARCHAR(36)		NOT NULL	Athlete family name
gender	CHAR(1)		NOT NULL	Athlete gender (‘M’ or ‘F’)
dob	DATE		NOT NULL	Athlete date of birth
height	INTEGER(3)		NOT NULL	Athlete height (cm)
athlete_weight	INTEGER(3)		NOT NULL	Athlete weight (kg)

results				
Attribute	**Type**	**Key**	**Constraints**	**Description**
event_fk	CHAR(6)	Foreign	NOT NULL	Event ID, unique
athlete_fk	CHAR(6)	Foreign	NOT NULL	Athlete ID, unique
position	INTEGER(1)		NOT NULL	Athlete event position (1, 2, 3 etc)
medal	CHAR(1)			Athlete event medal (‘G’, ‘S’, ‘B’ or NULL)

Relational Schema

countries(
  country_pk,
  country_name
)

sports(
  sport_pk,
  sport_name
)

events(
  event_pk,
  sport_fk,
  gender,
  event_name,
  event_datetime,
  venue
)

athletes(
  athlete_pk,
  country_fk,
  family_name,
  given_name,
  gender,
  dob,
  height,
  athlete_weight
)

results(
  event_fk,
  athlete_fk,
  primary_key(event_fk, athlete_fk)
  position,
  medal
)

Primary keys and foreign keys are denoted with underscoring and italicisation, respectively.

Explanations

Entity Sets

countries		
Attribute	**Type**	**Explanation**
country_pk	CHAR(3)	Each country was defined to have a fixed length ID (it's three-letter code as defined by the IOC).
country_name	VARCHAR(36)	Country names vary in length

sports		
Attribute	**Type**	**Explanation**
sport_pk	CHAR(6)	Each sport was defined to have a fixed length ID.
sport_name	VARCHAR(36)	Sport names vary in length

events		
Attribute	**Type**	**Explanation**
event_pk	CHAR(6)	CHAR is an appropriate data type as I defined each event to have a fixed length ID.
sport_fk	CHAR(6)	–
gender	CHAR(1)	Each event was defined to have a fixed length gender designation ('M' or 'F').
event_name	VARCHAR(36)	Event names vary in length
event_datetime	DATETIME	Each event takes place on a particular date at a particular time
venue	VARCHAR(36)	Venue names vary in length

athletes		
Attribute	**Type**	**Explanation**
athlete_pk	CHAR(6)	Each athlete is defined to have a fixed length ID.
country_fk	CHAR(3)	–
family_name	VARCHAR(36)	Family names vary in length
given_name	VARCHAR(36)	Given names vary in length
gender	CHAR(1)	Each athlete was defined to have a fixed length gender designation (‘M’ or ‘F’).
dob	DATE	Each athlete was born on a particular date.
height	INTEGER(3)	Each athlete’s height will be less than 999 cm. So, only need to display three digits
athlete_weight	INTEGER(3)	Each athlete’s weight will be less than 999 kg. So, only need to display three digits

results		
Attribute	**Type**	**Explanation**
event_fk	CHAR(6)	–
athlete_fk	CHAR(6)	–
position	INTEGER(1)	Each athlete position was defined to have a fixed length designation from 1-10.
medal	CHAR(1)	Each athlete was defined to have a fixed length medal designation (‘G’, ‘S’ or ‘B’).

Relationship Sets and Constraints

View *Design* > *Illustrative Materials* > *Relationship Sets* and *Design* > *Illustrative Materials* > *Constraints*.

Assumptions

1. Athletes' are identified by a family name and given name
2. Every attribute of type VARCHAR() is assumed to have a value which will not overflow the prescribed character limit
3. *For the sake of the database*, every athlete's gender is either male or female
4. results entities with position 1, 2 or 3 and NULL medal are results from non-final events (e.g. qualification rounds)

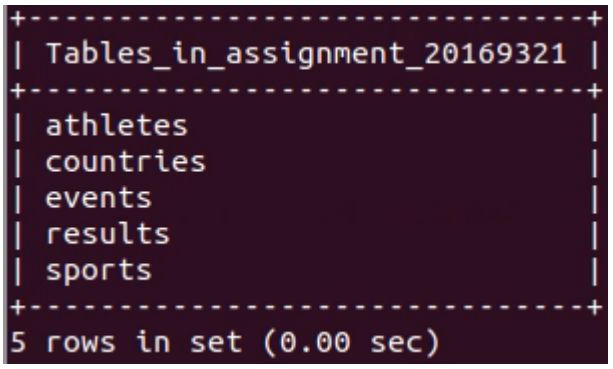
Database Implementation

Login and Database Creation

1. Navigate to the src directory.
2. To enter the MySQL Command-Line Client, at the "$ " prompt, enter:
   mysql -u me -p
   Then, at the "Enter password: " prompt, enter:
   myUserPassword
3. To create the relevant database, at the "mysql>" prompt, enter:
   CREATE DATABASE assignment_20169321;
4. To use the aforementioned database, enter:
   USE assignment_20169321

Table Creation and Value Insertion

1. To create the relevant tables, enter:
   SOURCE assignment_tables.sql
2. To show the relevant tables, enter:
   SHOW TABLES;
   The output should be as follows:



3. To show the description of the athletes table, enter:
   DESC athletes;
   The output should be as follows:

```
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| athlete_pk     | char(6)     | NO   | PRI | NULL    |       |
| country_fk     | char(3)     | YES  | MUL | NULL    |       |
| family_name    | varchar(36) | NO   |     | NULL    |       |
| given_name     | varchar(36) | NO   |     | NULL    |       |
| gender         | char(1)     | NO   |     | NULL    |       |
| dob            | date        | NO   |     | NULL    |       |
| height         | int(3)      | NO   |     | NULL    |       |
| athlete_weight | int(3)      | NO   |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

4. To show the description of the countries table, enter:
   DESC countries;
   The output should be as follows:

```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| country_pk   | char(3)     | NO   | PRI | NULL    |       |
| country_name | varchar(36) | NO   | UNI | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

5. To show the description of the events table, enter:
   DESC events;
   The output should be as follows:

```
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| event_pk        | char(6)     | NO   | PRI | NULL    |       |
| sport_fk        | char(6)     | YES  | MUL | NULL    |       |
| gender          | char(1)     | YES  |     | NULL    |       |
| event_name      | varchar(36) | NO   | UNI | NULL    |       |
| event_datetime  | datetime    | NO   |     | NULL    |       |
| venue           | varchar(36) | NO   |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

6. To show the description of the results table, enter:
   DESC results;
   The output should be as follows:

```
+-------------+---------+------+-----+---------+-------+
| Field       | Type    | Null | Key | Default | Extra |
+-------------+---------+------+-----+---------+-------+
| athlete_fk  | char(6) | NO   | PRI | NULL    |       |
| event_fk    | char(6) | NO   | PRI | NULL    |       |
| position    | int(1)  | NO   |     | NULL    |       |
| medal       | char(1) | YES  |     | NULL    |       |
+-------------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

7. To show the description of the sports table, enter:
   DESC sports;
   The output should be as follows:

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| sport_pk   | char(6)     | NO   | PRI | NULL    |       |
| sport_name | varchar(36) | NO   | UNI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

8. To insert the relevant entities into the relevant tables, enter:
   SOURCE assignment_values.sql
9. To show the entities of the athletes table, enter:
   SELECT * FROM athletes;
   The output should be as follows:

```
+------------+------------+-------------+------------+--------+------------+--------+----------------+
| athlete_pk | country_fk | family_name | given_name | gender | dob        | height | athlete_weight |
+------------+------------+-------------+------------+--------+------------+--------+----------------+
| 029145     | SWE        | AALTONEN    | Paavo      | M      | 1983-12-04 |    180 |             75 |
| 115612     | ZAM        | CHINYEMBA   | Patrick    | M      | 2001-07-28 |    172 |             61 |
| 182321     | SWE        | AARBERG     | Jan-Erik   | M      | 1985-07-09 |    182 |             78 |
| 193021     | USA        | ABBOTT      | Jeremy     | M      | 1991-03-23 |    173 |             72 |
| 193131     | USA        | AALTEN      | Cornelia   | W      | 1999-12-28 |    172 |             48 |
| 193432     | USA        | GRANT       | Rhyan      | M      | 1991-08-01 |    173 |             72 |
| 193821     | JPN        | AOKI        | Kushina    | W      | 1998-06-06 |    173 |             48 |
| 281921     | ZAM        | MAPFUMO     | Lucy       | W      | 2000-04-06 |    170 |             47 |
| 283191     | AUS        | SCOTT       | Lachlan    | M      | 1992-08-21 |    177 |             75 |
| 381022     | AUS        | WINWOOD     | Alex       | M      | 1997-02-02 |    174 |             60 |
| 383131     | AUS        | BEHICH      | Aziz       | M      | 1990-12-16 |    170 |             63 |
| 389121     | USA        | RYAN        | Mat        | M      | 1992-08-04 |    184 |             82 |
| 390112     | USA        | DUKE        | James      | M      | 1992-08-21 |    174 |             74 |
| 398012     | AUS        | WRIGHT      | Bailey     | M      | 1992-08-15 |    186 |             84 |
| 481640     | USA        | MALTO       | Stephanie  | W      | 1997-03-03 |    171 |             47 |
| 489231     | AUS        | CHRIS       | David      | M      | 1990-01-11 |    171 |             73 |
| 547232     | USA        | BOYLE       | Martin     | M      | 1993-04-23 |    191 |             75 |
| 582322     | AUS        | KARACIC     | Fran       | M      | 1996-05-12 |    181 |             75 |
| 839111     | AUS        | MCGOWAN     | Ryan       | M      | 1989-08-15 |    191 |             75 |
| 839198     | USA        | ELDER       | Steve      | M      | 1995-01-01 |    180 |             67 |
+------------+------------+-------------+------------+--------+------------+--------+----------------+
20 rows in set (0.00 sec)
```

10. To show the entities of the countries table, enter:
    SELECT * FROM countries;
    The output should be as follows:

```
+-------------+----------------+
| country_pk | country_name   |
+-------------+----------------+
| AUS         | Australia      |
| JPN         | Japan          |
| SWE         | Sweden         |
| USA         | United States  |
| ZAM         | Zambia         |
+-------------+----------------+
5 rows in set (0.00 sec)
```

11. To show the entities of the events table, enter:
    SELECT * FROM events;
    The output should be as follows:

```
+----------+---------+--------+-----------------------------+---------------------+-----------------------+
| event_pk | sport_fk | gender | event_name                  | event_datetime      | venue                 |
+----------+---------+--------+-----------------------------+---------------------+-----------------------+
| 274100   | 328931   | W      | Women's 100m Butterfly Final | 2021-07-27 11:30:00 | Tokyo Aquatics Centre |
| 318402   | 673480   | M      | Men's Skeet Second Round    | 2021-07-27 15:45:00 | Asaka Shooting Range  |
| 371121   | 938031   | M      | Winwood vs Chinyemba        | 2021-07-28 18:00:00 | Kokugikan Arena       |
| 936103   | 673480   | M      | Men's Skeet Final Round     | 2021-07-28 15:45:00 | Asaka Shooting Range  |
| 937802   | 482301   | M      | Men's Final: Australia vs USA | 2021-07-26 10:00:00 | Miyagi Stadium        |
+----------+---------+--------+-----------------------------+---------------------+-----------------------+
5 rows in set (0.00 sec)
```

12. To show the entities of the results table, enter:
    SELECT * FROM results;
    The output should be as follows:

```
+----------+------------+----------+-------+
| event_fk | athlete_fk | position | medal |
+----------+------------+----------+-------+
| 274100   | 193131     |        1 | G     |
| 274100   | 193821     |        3 | B     |
| 274100   | 281921     |        2 | S     |
| 274100   | 481640     |        4 | NULL  |
| 318402   | 029145     |        1 | NULL  |
| 318402   | 182321     |        2 | NULL  |
| 318402   | 193021     |        3 | NULL  |
| 318402   | 283191     |        4 | NULL  |
| 318402   | 390112     |        5 | NULL  |
| 318402   | 489231     |        6 | NULL  |
| 371121   | 115612     |        1 | G     |
| 371121   | 381022     |        2 | NULL  |
| 936103   | 029145     |        1 | G     |
| 936103   | 182321     |        2 | S     |
| 936103   | 193021     |        3 | B     |
| 937802   | 193021     |        2 | S     |
| 937802   | 383131     |        1 | G     |
| 937802   | 389121     |        2 | S     |
| 937802   | 398012     |        1 | G     |
| 937802   | 547232     |        2 | S     |
| 937802   | 582322     |        1 | G     |
| 937802   | 839111     |        1 | G     |
| 937802   | 839198     |        2 | S     |
+----------+------------+----------+-------+
23 rows in set (0.00 sec)
```

13. To show the entities of the sports table, enter:
    SELECT * FROM sports;
    The output should be as follows:

```
+----------+----------------+
| sport_pk | sport_name     |
+----------+----------------+
| 482301   | 3x3 Basketball |
| 938031   | Boxing         |
| 673480   | Shooting       |
| 328931   | Swimming       |
+----------+----------------+
4 rows in set (0.00 sec)
```

Query Design and Implementation and Database Usage

Query Design and Implementation

To show the results of the relevant queries, enter:
SOURCE assignment_queries.sql
The output should be as follows:

1. Show the number of athletes representing each country.

```
+---------+-------+
| country | count |
+---------+-------+
| AUS     |     7 |
| JPN     |     1 |
| SWE     |     2 |
| USA     |     8 |
| ZAM     |     2 |
+---------+-------+
5 rows in set (0.00 sec)
```

2. Show all participating countries ordered by name.

```
+---------------------+
| country_listing     |
+---------------------+
| Zambia (ZAM)        |
| United States (USA) |
| Sweden (SWE)        |
| Japan (JPN)         |
| Australia (AUS)     |
+---------------------+
5 rows in set (0.00 sec)
```

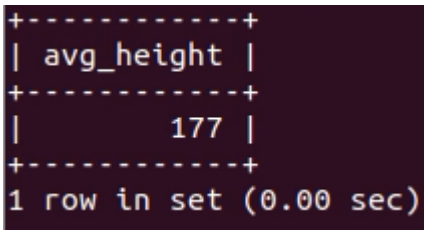
3. Show the age of the average American athlete.

```
+-------------+
| average_age |
+-------------+
|          27 |
+-------------+
1 row in set (0.00 sec)
```

4. Show the names and nationalities of all gold medalists

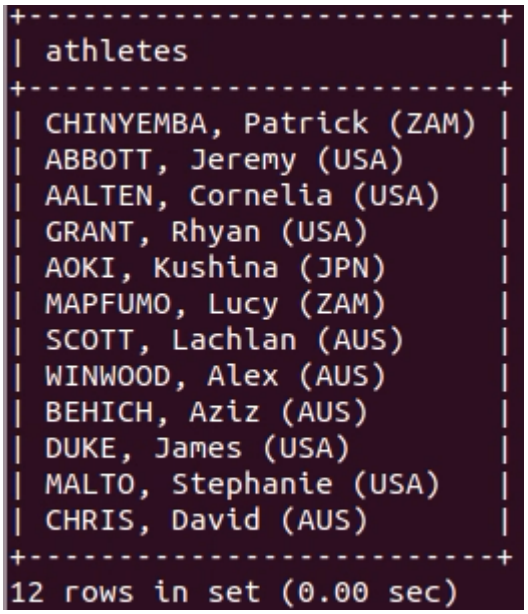
```
+--------------------------+
| gold_medalists           |
+--------------------------+
| AALTEN, Cornelia (USA)   |
| CHINYEMBA, Patrick (ZAM) |
| AALTONEN, Paavo (SWE)    |
| BEHICH, Aziz (AUS)       |
| WRIGHT, Bailey (AUS)     |
| KARACIC, Fran (AUS)      |
| MCGOWAN, Ryan (AUS)      |
+--------------------------+
7 rows in set (0.01 sec)
```

5. Show the height of the average athlete.



```
+------------+
| avg_height |
+------------+
|        177 |
+------------+
1 row in set (0.00 sec)
```

6. Show the names and nationalities of all athletes shorter than the average athlete.



```
+--------------------------+
| athletes                 |
+--------------------------+
| CHINYEMBA, Patrick (ZAM) |
| ABBOTT, Jeremy (USA)     |
| AALTEN, Cornelia (USA)   |
| GRANT, Rhyan (USA)       |
| AOKI, Kushina (JPN)      |
| MAPFUMO, Lucy (ZAM)      |
| SCOTT, Lachlan (AUS)     |
| WINWOOD, Alex (AUS)      |
| BEHICH, Aziz (AUS)       |
| DUKE, James (USA)        |
| MALTO, Stephanie (USA)   |
| CHRIS, David (AUS)       |
+--------------------------+
12 rows in set (0.00 sec)
```

Each of these queries were designed with the goal of answering interesting questions the average person might want to know about the 2020 olympics, even those that don't know how to MySQL (or SQL at all for that matter). For example, *"What is the age of the average American athlete?"*, sounds like an interesting trivia question. From there, I deconstructed each question, focusing on key elements. In regards to this question, these key elements are "age", "average", "athlete" and "American". Thereafter, I designed queries which addressed the key elements of each question, and in turn, answered the question itself. Finally, I implemented those queries and tested them to ensure they worked as expected. As seen above.

Advanced Features

1. To recreate the recreate the relevant tables, enter:
   SOURCE assignment_tables.sql
2. To create the relevant stored procedures, enter:
   SOURCE assignment_procedures.sql
3. To insert the relevant entities into the relevant tables, using the relevant stored procedures, enter:
   SOURCE assignment_values2.sql
4. To show the entities of each of the tables, enter the commands outlined in steps 9-13 of *Implementation* > *Table Creation and Value Insertion*.
5. To implement the relevant views, enter:
   SOURCE assignment_views.sql
6. To show the entities of the representatives view, enter:
   SELECT * FROM representatives;
   The output should be as follows:

```
+---------+-----------------+
| country | representatives |
+---------+-----------------+
| AUS     |               7 |
| JPN     |               1 |
| SWE     |               2 |
| USA     |               8 |
| ZAM     |               2 |
+---------+-----------------+
5 rows in set (0.00 sec)
```

7. To show the entities of the winners view, enter:
   SELECT * FROM winners;
   The output should be as follows:

```
+-------------+------------+---------+
| family_name | given_name | country |
+-------------+------------+---------+
| AALTEN      | Cornelia   | USA     |
| CHINYEMBA   | Patrick    | ZAM     |
| AALTONEN    | Paavo      | SWE     |
| BEHICH      | Aziz       | AUS     |
| WRIGHT      | Bailey     | AUS     |
| KARACIC     | Fran       | AUS     |
| MCGOWAN     | Ryan       | AUS     |
+-------------+------------+---------+
7 rows in set (0.00 sec)
```

Each of these advanced features were designed with the goal of improving workflow. In regards to the stored procedures, database users are then given the ability to perform insertions using the INSERT statement *or* via the procedures I implemented. In regards to the views, rather than constantly having to create a select query with a join in order to view the winning athletes, database users can simply execute the query SELECT * FROM winners.

Python Integration

1. To exit the MySQL Command-Line Client, enter:
   exit
2. To run the relevant python3 file, at the “$ ” prompt, enter:
   python3 assignment.py
   Then, at the “Enter username: ” prompt, enter:
   me
   Lastly, at the “Enter password: ” prompt, enter:
   myUserPassword
   The output should be as follows:

Discussion

Challenges

The most notable challenges included:

1. Appropriately modelling the database. Conceptually, we are all aware that the athletes, countries, sports, events and event results all have some relation to the olympics. However, piecing together how these entities ought to be related to one another was probably the most time-consuming endeavour of the entire assignment. Though, spending a great deal of time designing results in a much less pain-staking implementation process.
2. Successfully navigating MySQL error messages. This required a great deal of patience as the error-messages are often quite cryptic, and in turn, require you to do a lot of research in regards to what the causes of these errors were and their potential remedies.
3. Finding and preparing sample data to use for table insertion. Since the description of the assignment was quite vague (I suspect with purpose), students were essentially left to find and prepare their own data. I had no problems with the idea of doing so, it just required a lot of time and effort to compile.

Limitations

The most notable limitations include:

1. The inability to insert certain entities due to potential overflow. For example, an athlete with a family name greater than 36 characters couldn't be inserted into the athletes table without misrepresenting their name in some way (e.g. shortening it or attempting to use some nickname).
2. The inability to prevent the adding of male athletes to women's events and vice versa.
3. The inability to insert certain entities due to non-conformity with table attributes. For example, an athlete with a three-part name couldn't be added into the athletes table appropriately as it only supports family_name and given_name attributes.

Improvements

Considering the aforementioned limitations, the most notable potential improvements include:

1. Identify all of the extremities in regards to attributes to ensure that no overflow or underflow occurs. For example, if the longest given name for an athlete is 29 characters, we change the data type of given_name attribute of the athletes table from VARCHAR(36) to VARCHAR(29).
2. Implement some form of error-checking to ensure that athletes cannot be mistakenly added to gender-specific events. For example, a man should not be added to a women's event and vice versa.
3. Remove the family_name and given_name attributes of the athletes table and replace them with an athlete_name attribute to further internationalise our database as not all people's names follow this format.

References

Sarkhel, Arjun. 2021. “2021 Olympics in Tokyo”. Kaggle. https://www.kaggle.com/arjunprasadsarkhel/2021-olympics-in-tokyo

The Roar. 2021. “2021 Olympics events schedule”. The Roar. https://www.theroar.com.au/olympics/olympics-events-schedule/