

# Curtin University – Department of Computing

## 1. Introduction

The design and execution of a MySQL relational database system created for the 2024 Olympic Games are described in this report. This database provides a foundation for querying data in a variety of formats and supports advanced database operations including stored procedures and views. Its goal is to handle information on athletes, events, medallist, teams, total medals, and schedules, coach. In the user guide it is mentioned that there are **5 SQL files** (**CreateTable.sql**, **LoadData.sql**, **Query.sql**, **advanceQuery.sql**, **DeleteTable.sql**) in order to efficiently use the Olympic game database. Again, there are 7 csv files that contains information to loading data into respective table like (**athlete.csv**, **coach.csv**, **team.csv**, **schedule.csv**, **medallist.csv**, **medal\_total.csv** and **events.csv**). Additionally, it also contains **3 python file** to be connected the MYSQL Server in a python environment and do some operation such as (**pythonConnect.py**, **eventdata.py**, **scheduleData.py**).

With an emphasis on efficiency and flexibility, the project included data modelling, database implementation, data loading, and query creation. Moreover, MySQL 8.0.39 has been used to implement the database on a Linux host running Ubuntu that can be accessed via the VMware Horizontal View Client. The design, implementation, data loading procedures, query usage, and complex database capabilities like stored procedures and views will all be covered in this study. The study will also examine the difficulties encountered and make recommendations for modifications for upcoming system versions.

## 2. Design of database

### 2.1 Entity Sets

Entity Set	Key	Other Attributes
COACH	<u>coach_code</u>	coach_name, gender, coach_function, country_code, country_long, disciplines, events
ATHLETE	<u>athlete_code</u>	name, name_short, name_tv, gender, country_code, country_long, disciplines, events
TEAM	<u>code</u>	team_gender, country_code, country, country_long, disciplines, disciplines_code, events, num_athletes
TOTAL_MEDAL	<u>medal_id</u>	country_code, country_long, Gold_Medal, Silver_Medal, Bronze_medal, Total
EVENT	<u>event_id</u>	event, sport, sport_code
MEDALLIST	<u>medallist_id</u> , REF KEY <u>code_atheltes</u> , <u>code_team</u>	medal_date, medal_type, medal_code, name, gender, country_code, country_long, team_gender, discipline, events, event_type, birth_date, is_medallist,
SCHEDULE	<u>schedule_id</u>	start_date, end_date, status, discipline, event, event_medal, phrase, gender, venue, venue_code

*Relationship Sets*

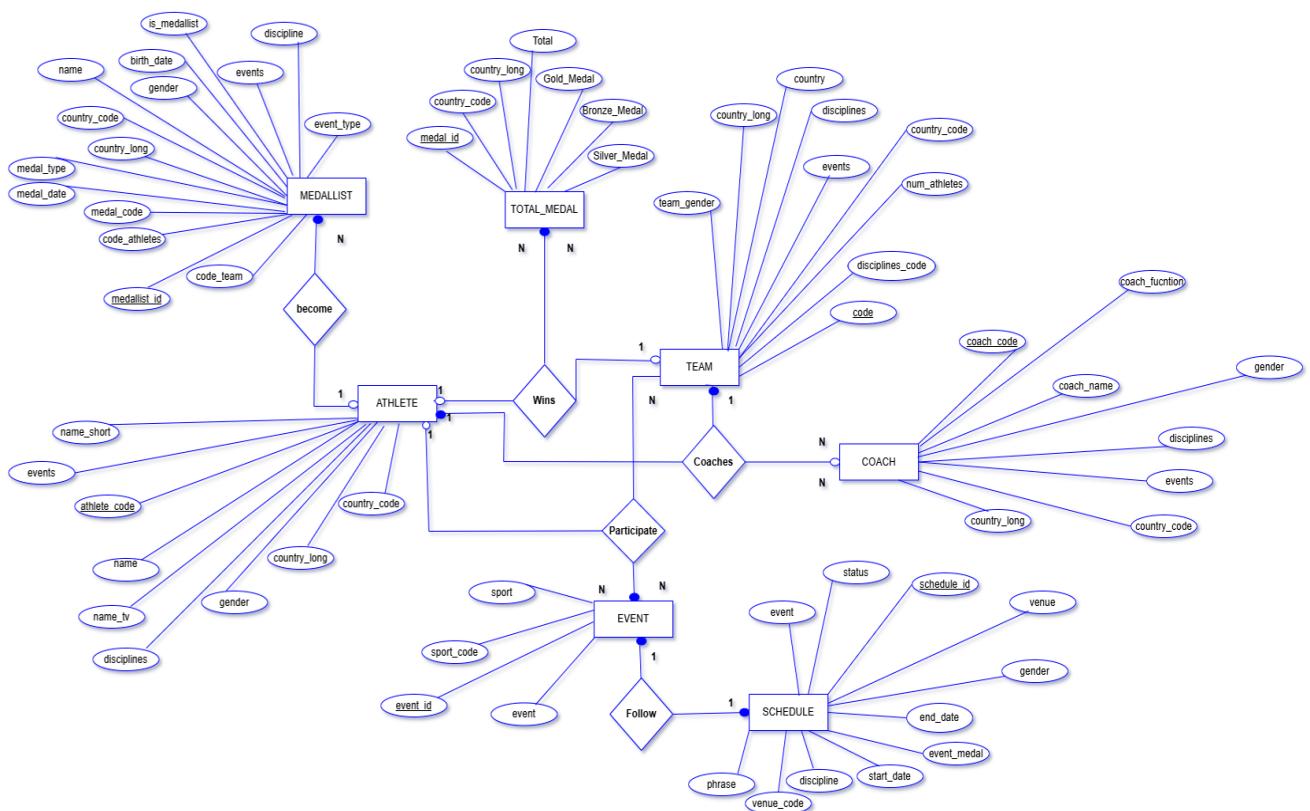
Relationship set	Between Entity Sets	Attributes
<b>become</b>	ATHLETE, MEDALLIST	
<b>wins</b>	ATHLET, TOTAL_MEDAL	
<b>wins</b>	TEAM, TOTAL_MEDAL	
<b>coaches</b>	TEAM, COACH	
<b>coaches</b>	ATHLETE, COACH	
<b>participate</b>	TEAM, EVENT	
<b>participate</b>	ATHELET, EVENT	
<b>follow</b>	EVENT, SCHEDULE	

*Constraints*

Between Entity Sets	Relationship set	Cardinality	Participation
ATHLETE, MEDALLIST	<b>become</b>	<p>One – Many</p> <p>Each athlete can win multiple medals (gold, silver, bronze) in various events. But each record refers to one athlete who won a medal.</p>	<b>ATHLETE - Partial</b> <b>MEDALLIST – Total</b> An athlete can exist without being the medallist. But every medallist must be Athletes.
ATHLETE, TOTAL_MEDAL	<b>wins</b>	<p>One – Many</p> <p>Each athlete can win multiple medals in various events.</p> <p>But each total medal record is associated with one athlete.</p>	<b>ATHLETE – Partial</b> <b>TOTAL_MEDAL - Total</b> Not every athlete will win medals, so an athletes may exist without a medal. Every medal must be linked to an athlete who has won medals
TEAM, TOTAL_MEDAL	<b>wins</b>	<p>One – Many</p> <p>A team can win multiple total medals in various events.</p> <p>But each total medal record is associated with one specific team.</p>	<b>TEAM – Partial,</b> <b>TOTAL_MEDAL - Total</b> A team can exist without winning any medals. Every medal must be belong to a team that won those medals.
TEAM, COACH	<b>coaches</b>	<p>One -Many</p> <p>A team can have one coach.</p> <p>But a coach can coaches many team.</p>	<b>COACH – Partial,</b> <b>TEAM - Total</b> Every team must have one coach. A coach can exist without being assigned to any team. But a team can not be exit without being trained from Coach.
COACH, ATHLETE,	<b>coaches</b>	<p>One -Many</p> <p>A coach can be trains/ coaches multiple athletes.</p> <p>Each athlete is associated with one coach at a time.</p>	<b>COACH – Partial,</b> <b>ATHLETE - Total</b> A coach can exist without training to athletes. Every athletes must be assign to one coach for training.
TEAM, EVENT	<b>participate</b>	<p>Many-Many</p> <p>One team can participant in multiple events</p> <p>But one events can have multiple teams</p>	<b>TEAM – Partial,</b> <b>EVENT - Partial</b> A team can exit participating in an event An event( individual/ team) can exit involving any teams.
ATHELET, EVENT	<b>participate</b>	One -Many	<b>ATHELTE – Partial</b>

		An athlete can participate many events, But each event participation linked to Many athlete	<b>EVENT- Partial</b> Not all athletes participate in every event. An event can occur without the participation of athletes.
EVENT, SCHEDULE	<b>follow</b>	One- One Each event must have one schedule. Each schedule is connected to only one event	<b>EVENT – Total</b> <b>SCHEDULE – Total</b> An event can not exit without having a schedule to take place.. Also, an schedule can not be exit without an events.

## 2.2 ER diagram



## 2.3 Data Description

### ATHLETE\_TABLE

Field	Type	Null	Key	Default	Extra
athlete_code	varchar(100)	NO	PRI	NULL	
name	char(70)	NO		NULL	
name_short	varchar(100)	NO		NULL	
name_tv	varchar(90)	NO		NULL	
gender	char(1)	NO		NULL	
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
disciplines	varchar(200)	NO		NULL	
events	varchar(100)	NO		NULL	

### COACH\_TABLE

Field	Type	Null	Key	Default	Extra
coach_code	varchar(100)	NO	PRI	NULL	
coach_name	char(70)	NO		NULL	
gender	char(3)	NO		NULL	
coach_function	varchar(50)	NO		NULL	
country_code	char(3)	NO		NULL	
country_long	varchar(100)	YES		NULL	
disciplines	varchar(200)	NO		NULL	
events	varchar(100)	YES		NULL	

### TEAM\_TABLE

Field	Type	Null	Key	Default	Extra
code	varchar(100)	NO	PRI	NULL	
team_gender	char(1)	NO		NULL	
country_code	char(3)	NO		NULL	
country	varchar(50)	NO		NULL	
country_long	varchar(100)	NO		NULL	
discipline	varchar(200)	NO		NULL	
disciplines_code	varchar(6)	NO		NULL	
events	varchar(100)	YES		NULL	
num_athletes	int	YES		NULL	

### MEDALLIST\_TABLE

Field	Type	Null	Key	Default	Extra
medal_date	date	YES		NULL	
medal_type	varchar(20)	YES		NULL	
medal_code	int	YES		NULL	
name	varchar(70)	YES		NULL	
gender	char(1)	YES		NULL	
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
team_gender	char(1)	YES		NULL	
discipline	varchar(200)	YES		NULL	
events	varchar(100)	YES		NULL	
event_type	varchar(20)	YES		NULL	
birth_date	date	YES		NULL	
code_athlete	varchar(100)	YES	MUL	NULL	
code_team	varchar(100)	YES	MUL	NULL	
is_medallist	tinyint(1)	YES		NULL	
medalist_id	int	NO	PRI	NULL	auto_increment

## **EVENT\_TABLE**

Field	Type	Null	Key	Default	Extra
event	varchar(100)	NO		NULL	
sport	varchar(200)	NO		NULL	
sport_code	varchar(10)	NO		NULL	
event_id	int	NO	PRI	NULL	auto_increment

## **SCHEDULE\_TABLE**

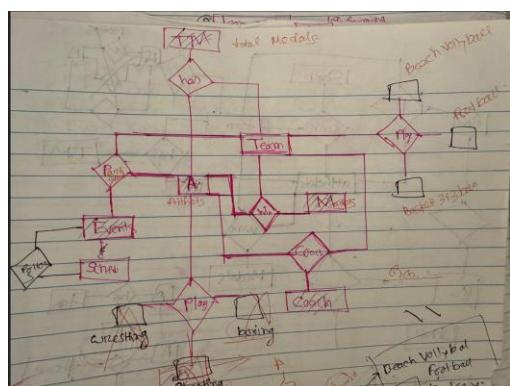
Field	Type	Null	Key	Default	Extra
start_date	timestamp	YES		NULL	
end_date	timestamp	YES		NULL	
status	varchar(20)	NO		NULL	
discipline	varchar(200)	NO		NULL	
discipline_code	varchar(60)	NO		NULL	
event	varchar(100)	YES		NULL	
event_medal	int	YES		NULL	
phase	varchar(100)	YES		NULL	
gender	char(1)	YES		NULL	
event_type	varchar(20)	YES		NULL	
venue	varchar(100)	YES		NULL	
venue_code	varchar(60)	YES		NULL	
schedule_id	int	NO	PRI	NULL	auto_increment

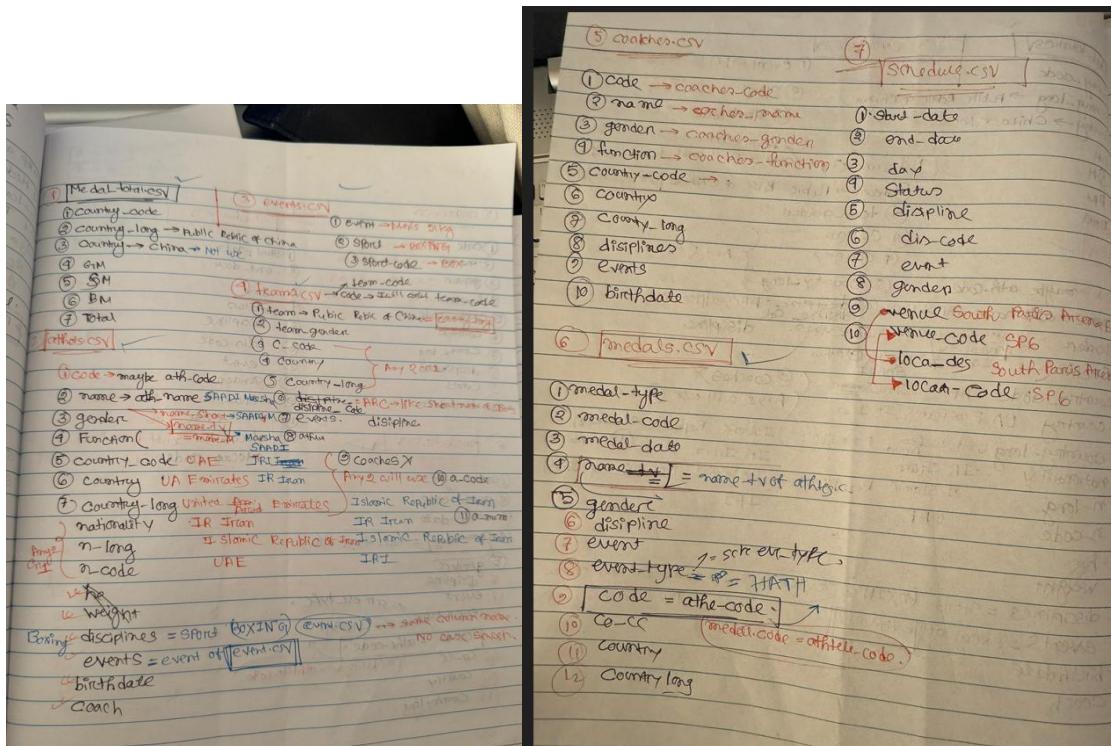
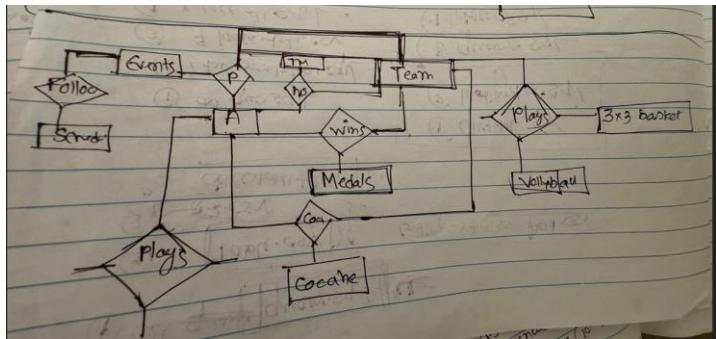
## **TOTAL MEDAL TABLE**

Field	Type	Null	Key	Default	Extra
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
Gold_Medal	int	YES		NULL	
Silver_Medal	int	YES		NULL	
Bronze_Medal	int	YES		NULL	
Total	int	YES		NULL	
medal_id	int	NO	PRI	NULL	auto_increment

#### ***2.4 Assumption I made during the design of the database.***

I did a draft from the csv file that which attribute will I use to creating the tables and also try to draw the ER diagram first, after finishing the draft. I changed it so many times, to get my desire ER diagram, and attributes. Here is a sample from one of it.





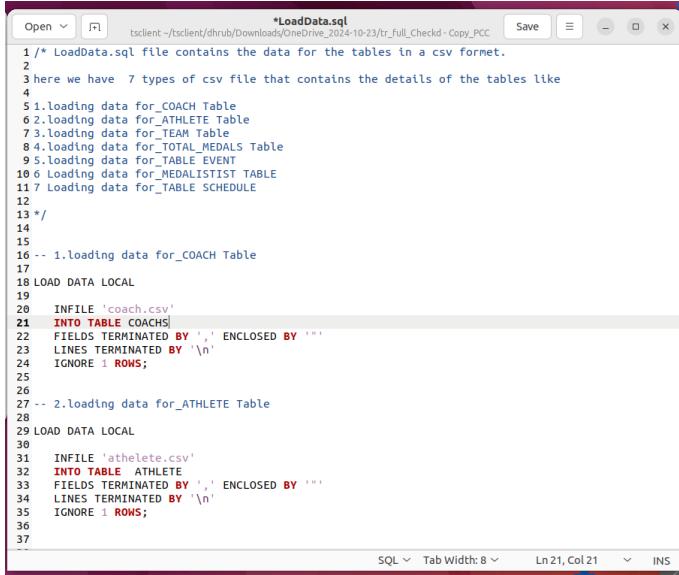
### **3. Implementation of the database and adding sample data**

#### *i. Table creation and constraints:*

The tables were created in the `CretaeTable.sql` scripts. There are constraints ,such as primary key foreign key, NOT NULL, NULL based on the data. Here is the file looks like:

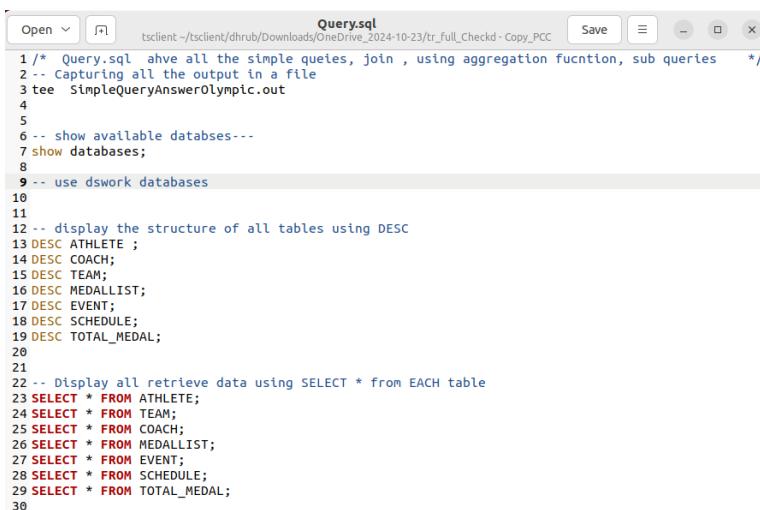
## ii. Loading Data:

After creating the tables to load the data , there is a file named **LoadData.sql** which have all the data of csv files to loading data into tables. Here is the **LoadData.sql** looks like:



```
*LoadData.sql
1 /* LoadData.sql file contains the data for the tables in a csv format.
2
3 here we have 7 types of csv file that contains the details of the tables like
4
5 1.loading data for_COACH Table
6 2.loading data for_ATHLETE Table
7 3.loading data for_TEAM Table
8 4.loading data for_TOTAL_MEDALS Table
9 5.loading data for_TABLE EVENT
10 6 Loading data for_MEDALLISTIST TABLE
11 7 Loading data for_TABLE SCHEDULE
12
13 */
14
15
16 -- 1.loading data for_COACH Table
17
18 LOAD DATA LOCAL
19
20 INFILE 'coach.csv'
21 INTO TABLE COACH$|
22 FIELDS TERMINATED BY ',' ENCLOSED BY ''
23 LINES TERMINATED BY '\n'
24 IGNORE 1 ROWS;
25
26
27 -- 2.loading data for_ATHLETE Table
28
29 LOAD DATA LOCAL
30
31 INFILE 'athelete.csv'
32 INTO TABLE ATHLETE$|
33 FIELDS TERMINATED BY ',' ENCLOSED BY ''
34 LINES TERMINATED BY '\n'
35 IGNORE 1 ROWS;
36
37
```

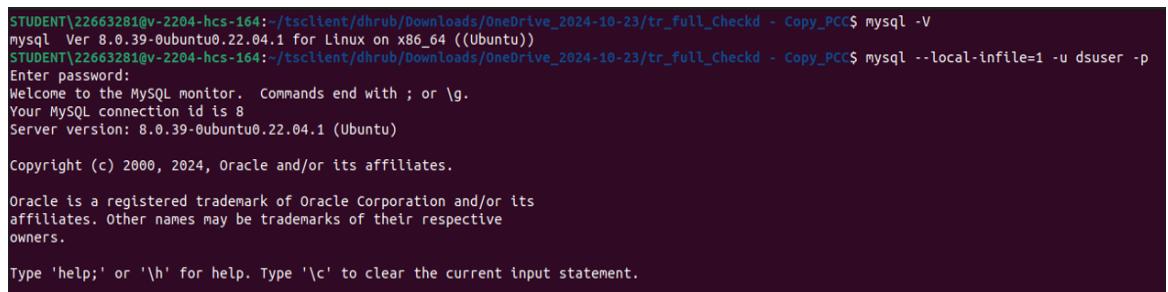
Each files have the explanation about the file description. Like in **Query.sql** have the description of about the file . like this SQL file is about creating queries suing join, subqueries, aggregation function and so on.



```
Query.sql
1 /* Query.sql ahve all the simple quetes, join , using aggregation fucntion, sub queries   */
2 -- Capturing all the output in a file
3 tee SimpleQueryAnswerOlympic.out
4
5
6 -- show available databases--
7 show databases;
8
9 -- use dwork databases
10
11
12 -- display the structure of all tables using DESC
13 DESC ATHLETE ;
14 DESC COACH;
15 DESC TEAM;
16 DESC MEDALLIST;
17 DESC EVENT;
18 DESC SCHEDULE;
19 DESC TOTAL_MEDAL;
20
21
22 -- Display all retrieve data using SELECT * from EACH table
23 SELECT * FROM ATHLETE;
24 SELECT * FROM TEAM;
25 SELECT * FROM COACH;
26 SELECT * FROM MEDALLIST;
27 SELECT * FROM EVENT;
28 SELECT * FROM SCHEDULE;
29 SELECT * FROM TOTAL_MEDAL;
```

## 4. Use of the database

### i. First, we need to login in VMware horizon client like this



```
STUDENT\22663281@v-2204-hcs-164:~/tsclient/dhrub/Downloads/OneDrive_2024-10-23/tr_full_Checkd - Copy_PCC$ mysql -v
mysql Ver 8.0.39-Ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
STUDENT\22663281@v-2204-hcs-164:~/tsclient/dhrub/Downloads/OneDrive_2024-10-23/tr_full_Checkd - Copy_PCC$ mysql --local-infile=1 -u dsuser -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.39-Ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Then see the existence DATABASE like this

```

mysql> show databases;
+-----+
| Database |
+-----+
| dswork   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.00 sec)

```

Now need to create our own database which is;

```

mysql> CREATE DATABASE IF NOT EXISTS Olympic_Game_2024_22663281;
Query OK, 1 row affected (0.05 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| Olympic_Game_2024_22663281 |
| dswork   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
6 rows in set (0.00 sec)

mysql> use Olympic_Game_2024_22663281;
Database changed

```

Now create all the tables using **CreateTable.sql**.

```

mysql> \. CreateTable.sql
Query OK, 0 rows affected, 1 warning (0.01 sec)
Query OK, 0 rows affected (0.13 sec)
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected (0.07 sec)
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected (0.08 sec)
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected, 4 warnings (0.09 sec)
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected (0.11 sec)
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected (0.17 sec)
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected (0.08 sec)
mysql>

```

After this, now its time to loading data using **LoadData.sql** file

```

mysql> \. LoadData.sql
Query OK, 974 rows affected, 974 warnings (0.22 sec)
Records: 974 Deleted: 0 Skipped: 0 Warnings: 974
Query OK, 11113 rows affected, 11346 warnings (0.97 sec)
Records: 11113 Deleted: 0 Skipped: 0 Warnings: 11346
Query OK, 1698 rows affected, 3 warnings (0.31 sec)
Records: 1698 Deleted: 0 Skipped: 0 Warnings: 3
Query OK, 92 rows affected, 92 warnings (0.15 sec)
Records: 92 Deleted: 0 Skipped: 0 Warnings: 92
Query OK, 329 rows affected, 329 warnings (0.17 sec)
Records: 329 Deleted: 0 Skipped: 0 Warnings: 329
Query OK, 1555 rows affected, 7706 warnings (0.56 sec)
Records: 2315 Deleted: 0 Skipped: 760 Warnings: 7706
Query OK, 3895 rows affected, 3897 warnings (0.65 sec)
Records: 3895 Deleted: 0 Skipped: 0 Warnings: 3897
mysql>

```

In this step, we will try some queries using **Queries.sql** file. As **Queries.sql** file contains almost 20 queries. If I source the file it will show me all result together. So here, I am trying to attempt every query one by one.

This shows me the **ATHLETE TABLE** description. I have 7 tables like this.

Field	Type	Null	Key	Default	Extra
athlete_code	varchar(100)	NO	PRI	NULL	
name	char(70)	NO		NULL	
name_short	varchar(100)	NO		NULL	
name_tv	varchar(90)	NO		NULL	
gender	char(1)	NO		NULL	
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
disciplines	varchar(200)	NO		NULL	
events	varchar(100)	NO		NULL	

This one shows the **TOTAL\_MEDAL TABLE** Description.

Field	Type	Null	Key	Default	Extra
country_code	char(3)	NO		NULL	
country_long	varchar(100)	NO		NULL	
Gold_Medal	int	YES		NULL	
Silver_Medal	int	YES		NULL	
Bronze_Medal	int	YES		NULL	
Total	int	YES		NULL	
medal_id	int	NO	PRI	NULL	auto_increment

Now I am trying to retrieve all the data from **TOTAL\_MEDAL TABLE** Like this:

country_code	country_long	Gold_Medal	Silver_Medal	Bronze_Medal	Total	medal_id
USA	United States of America	40	44	42	126	1
CHN	People's Republic of China	40	27	24	91	2
JPN	Japan	20	12	13	45	3
AUS	Australia	18	19	16	53	4
FRA	France	16	26	22	64	5
NED	Netherlands	15	7	12	34	6
GBR	Great Britain	14	22	29	65	7
KOR	Republic of Korea	13	9	10	32	8
ITA	Italy	12	13	15	40	9
GER	Germany	12	13	8	33	10
NZL	New Zealand	10	7	3	20	11
CAN	Canada	9	7	11	27	12
UZB	Uzbekistan	8	2	3	13	13
HUN	Hungary	6	7	6	19	14
ESP	Spain	5	4	9	18	15
SWE	Sweden	4	4	3	11	16
KEN	Kenya	4	2	5	11	17
NOR	Norway	4	1	3	8	18
IRL	Ireland	4	0	3	7	19
BRA	Brazil	3	7	10	20	20
IRI	Islamic Republic of Iran	3	6	3	12	21
UKR	Ukraine	3	5	4	12	22
ROU	Romania	3	4	2	9	23
GEO	Georgia	3	3	1	7	24
PER	Peru	2	4	6	12	25

It has 92 Rows .

Now I am trying to retrieve all the data from **EVENT TABLE** Like this:

```
mysql> SELECT * FROM EVENT;
+-----+-----+-----+-----+
| event | sport | sport_code | event_id |
+-----+-----+-----+-----+
| Men's Individual | Archery | ARC | 1 |
| Women's Individual | Archery | ARC | 2 |
| Men's Team | Archery | ARC | 3 |
| Women's Team | Archery | ARC | 4 |
| Mixed Team | Archery | ARC | 5 |
| Men's Team | Artistic Gymnastics | GAR | 6 |
| Men's All-Around | Artistic Gymnastics | GAR | 7 |
| Men's Floor Exercise | Artistic Gymnastics | GAR | 8 |
| Men's Pommel Horse | Artistic Gymnastics | GAR | 9 |
| Men's Rings | Artistic Gymnastics | GAR | 10 |
| Men's Vault | Artistic Gymnastics | GAR | 11 |
| Men's Parallel Bars | Artistic Gymnastics | GAR | 12 |
| Men's Horizontal Bar | Artistic Gymnastics | GAR | 13 |
| Women's Team | Artistic Gymnastics | GAR | 14 |
| Women's All-Around | Artistic Gymnastics | GAR | 15 |
| Women's Vault | Artistic Gymnastics | GAR | 16 |
| Women's Uneven Bars | Artistic Gymnastics | GAR | 17 |
| Women's Balance Beam | Artistic Gymnastics | GAR | 18 |
| Women's Floor Exercise | Artistic Gymnastics | GAR | 19 |
| Duet | Artistic Swimming | SWA | 20 |
| Team | Artistic Swimming | SWA | 21 |
| Men's 100m | Athletics | ATH | 22 |
| Men's 200m | Athletics | ATH | 23 |
| Men's 400m | Athletics | ATH | 24 |
| Men's 800m | Athletics | ATH | 25 |
| Men's 1500m | Athletics | ATH | 26 |
+-----+-----+-----+-----+
```

it has 329 rows

Now, retrieving data from **COACH TABLE**. It has rows of 974 .

```
mysql> SELECT * FROM COACH;
+-----+-----+-----+-----+-----+-----+-----+-----+
| coach_code | coach_name | gender | coach_function | country_code | country_long | disciplines | events |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1533246 | PEDRERO ofelia | Fem | Coach | MEX | Mexico | Artistic Swimming | Team |
| 1535775 | RADHI SHENAISHIL | Mal | Head Coach | IRQ | Iraq | Football | Men |
| 1536055 | AFLAKIKHAMEH Majid | Mal | Coach | IRI | Islamic Republic of Iran | Taekwondo | Men |
| 1536059 | YOUSEFY Mehrdad | Mal | Coach | IRI | Islamic Republic of Iran | Taekwondo | Men |
| 1536060 | MADDAH Minoo | Fem | Coach | IRI | Islamic Republic of Iran | Taekwondo | Men |
| 1536328 | LOFTUS Adriana | Fem | Coach | MEX | Mexico | Artistic Swimming | Team |
| 1538313 | FERRARA Fernando | Mal | Head Coach | ARG | Argentina | Hockey | Women |
| 1538315 | GULLA Alejandra | Fem | Assistant Coach | ARG | Argentina | Hockey | Women |
| 1538317 | CAPURRO Santiago | Mal | Assistant Coach | ARG | Argentina | Hockey | Women |
| 1538745 | RONCONI Mariano | Mal | Head Coach | ARG | Argentina | Hockey | Women |
| 1538748 | PAULON Ezequiel | Mal | Assistant Coach | ARG | Argentina | Hockey | Women |
| 1538751 | VILA Matias | Mal | Assistant Coach | ARG | Argentina | Hockey | Women |
| 1539598 | KHEIRKHAN HAJIRASOULIAN Mohammadre | Mal | Coach | IRI | Islamic Republic of Iran | Artistic Gymnastics | Team |
| 1540258 | DAVIS DIAZ David | Mal | Coach | MEX | Mexico | Taekwondo | Men |
| 1540259 | MENDOZA MORA Abel | Mal | Coach | MEX | Mexico | Taekwondo | Men |
| 1540260 | VICTORIA ESPINOSA de los Alfonso | Mal | Coach | MEX | Mexico | Taekwondo | Men |
| 1540522 | MILANO Guillermo | Mal | Head Coach | ARG | Argentina | Handball | Men |
| 1540638 | GOMEZ CORA Santiago | Mal | Head Coach | ARG | Argentina | Rugby Sevens | Men |
| 1540639 | GRAVANO Leonardo | Mal | Assistant Coach | ARG | Argentina | Rugby Sevens | Men |
| 1540840 | FILIORIANU Ana Luiza | Fem | Coach | ROU | Romania | Rhythmic Gymnastics | Team |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Retrieving data from **TEAM TABLE**

```
mysql> SELECT * FROM TEAM;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| code | team_gender | country_code | country | country_long | discipline | disciplines_code | events | num_athletes |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ARCTEAM3---CHN01 | M | CHN | China | People's Republic of China | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---COL01 | M | COL | Colombia | Colombia | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---FRA01 | M | FRA | France | France | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---GBR01 | M | GBR | Great Britain | Great Britain | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---IND01 | M | IND | India | India | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---ITA01 | M | ITA | Italy | Italy | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---JPN01 | M | JPN | Japan | Japan | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---KAZ01 | M | KAZ | Kazakhstan | Kazakhstan | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---KOR01 | M | KOR | Korea | Republic of Korea | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---MEX01 | M | MEX | Mexico | Mexico | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---TPE01 | M | TPE | Chinese Taipei | Chinese Taipei | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---TUR01 | M | TUR | Turkeye | Turkeye | Archery | ARC | Men's Team | 3 |
| ARCTEAM3---CHN01 | W | CHN | China | People's Republic of China | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---FRA01 | W | FRA | France | France | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---GBR01 | W | GBR | Great Britain | Great Britain | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---GER01 | W | GER | Germany | Germany | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---INA01 | W | INA | Indonesia | Indonesia | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---IND01 | W | IND | India | India | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---KOR01 | W | KOR | Korea | Republic of Korea | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---MAS01 | W | MAS | Malaysia | Malaysia | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---MEX01 | W | MEX | Mexico | Mexico | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---NED01 | W | NED | Netherlands | Netherlands | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---TPE01 | W | TPE | Chinese Taipei | Chinese Taipei | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---USA01 | W | USA | United States | United States of America | Archery | ARC | Women's Team | 3 |
| ARCTEAM3---AUS01 | X | AUS | Australia | Australia | Archery | ARC | Mixed Team | 2 |
| ARCTEAM2---BRA01 | X | BRA | Brazil | Brazil | Archery | ARC | Mixed Team | 2 |
| ARCTEAM2---CAN01 | X | CAN | Canada | Canada | Archery | ARC | Mixed Team | 2 |
| ARCTEAM2---CHN01 | X | CHN | China | People's Republic of China | Archery | ARC | Mixed Team | 2 |
| ARCTEAM2---COL01 | X | COL | Colombia | Colombia | Archery | ARC | Mixed Team | 2 |
| ARCTEAM2---CZE01 | X | CZE | Czechia | Czechia | Archery | ARC | Mixed Team | 2 |
| ARCTEAM2---EGY01 | X | EGY | Egypt | Egypt | Archery | ARC | Mixed Team | 2 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

It has rows of 1698.

## 5. Implement queries

1. By this queries we can get the **discipline name** which has occur venue name ‘**champ-de-Mars arena**’

```
mysql> SELECT
->     DISTINCT discipline
->     FROM
->     SCHEDULE
->     WHERE
->         venue = 'Champ-de-Mars Arena';
+-----+
| discipline |
+-----+
| Judo      |
| Wrestling |
+-----+
2 rows in set (0.01 sec)
```

2. By this query we can get out the total athletes number and renamed the table as **TOTAL\_ATHLETES FROM TEAM TABLE**.

```
mysql> SELECT  SUM(num_athletes) AS TOTAL_ATHLETES  FROM TEAM;
+-----+
| TOTAL_ATHLETES |
+-----+
|      7808      |
+-----+
1 row in set (0.00 sec)

mysql>
```

3. This query help us to get all the team code from TEAM TABLE where discipline is ‘%Archery%’;

```
mysql> SELECT code AS TEAM_code  FROM TEAM where discipline like '%Archery%' ;
+-----+
| TEAM_code   |
+-----+
| ARCMTEAM3---CHN01|
| ARCMTEAM3---COL01|
| ARCMTEAM3---FRA01|
| ARCMTEAM3---GBR01|
| ARCMTEAM3---IND01|
| ARCMTEAM3---ITA01|
| ARCMTEAM3---JPN01|
| ARCMTEAM3---KAZ01|
| ARCMTEAM3---KOR01|
| ARCMTEAM3---MEX01|
| ARCMTEAM3---TPE01|
| ARCMTEAM3---TUR01|
| ARCHTEAM3---CHN01|
| ARCWTEAM3---FRA01|
| ARCWTEAM3---GBR01|
| ARCHTEAM3---GER01|
| ARCWTEAM3---INA01|
| ARCWTEAM3---IND01|
| ARCWTEAM3---KOR01|
| ARCWTEAM3---MAS01|
| ARCWTEAM3---MEX01|
| ARCHTEAM3---NEP01|
+-----+
```

It display 57 rows.

4. From **ATHLETE TABLE** get out the country name, count athlete based on the country. renamed the table as Total participant

```

mysql> SELECT
    ->     country_long AS Name_of_Country,
    ->     COUNT(athlete_code) AS Total_Participants
    -> FROM
    ->     ATHLETE
    -> GROUP BY
    ->     country_long
    -> ORDER BY
    ->     country_long ;

```

Name_of_Country	Total_Participants
Afghanistan	6
AIN	32
Albania	8
Algeria	46
American Samoa	2
Andorra	2
Angola	25
Antigua and Barbuda	5
Argentina	143
Armenia	15
Aruba	6
Australia	475
Austria	84
Azerbaijan	48
Bahamas	19
Bahrain	14
Bangladesh	5
Barbados	4
...	...

it displays 206 Rows

5. From **COACH TABLE** get out the country name, count coach based on the country. renamed the table as **Total\_coches\_num\_by\_country**

```

mysql> SELECT
    ->     country_long,
    ->     COUNT(coach_name) AS Total_Coaches_NUM_BY_Country
    -> FROM
    ->     COACH
    -> GROUP BY
    ->     country_long
    -> ORDER BY
    ->     country_long;

```

country_long	Total_Coaches_NUM_BY_Country
AIN	3
Algeria	3
Angola	3
Argentina	19
Armenia	2
Australia	40
Austria	3
Azerbaijan	6
Belgium	17
Brazil	35
Bulgaria	9
Burkina Faso	3
Canada	34
Chile	1
Chinese Taipei	3
Colombia	5
Côte d'Ivoire	2
Croatia	10
...	...

displays around 98 rows.

6. This query to get out the maximum event name from TEAM using aggregation function

```

mysql> Select MAX(events) AS EVENT_MAX from TEAM
    -> ;

```

EVENT_MAX
Women's Team Sprint

1 row in set (0.00 sec)

7. Athletes and coach ratio from **ATHLTES AND CAOCH TABLE**

```

mysql> SELECT
->     c.country_code,
->     COUNT(DISTINCT a.athlete_code) AS Total_Athletes,
->     COUNT(DISTINCT c.coach_code) AS Total_Coaches,
->     ROUND(COUNT(DISTINCT a.athlete_code) / COUNT(DISTINCT c.coach_code), 2) AS Athletes_vs_Coach_Ratio
-> FROM
->     COACH c
-> JOIN
->     ATHLETE a ON c.country_code = a.country_code          -- joining based on country_code
-> GROUP BY
->     c.country_code
-> ORDER BY
->     Total_Athletes DESC , Total_Coaches DESC;
+-----+-----+-----+-----+
| country_code | Total_Athletes | Total_Coaches | Athletes_vs_Coach_Ratio |
+-----+-----+-----+-----+
| USA           |      619       |       56       |        11.05        |
| FRA           |      601       |       55       |        10.93        |
| AUS           |      475       |       48       |        11.88        |
| GER           |      457       |       37       |        12.35        |
| JPN           |      431       |       43       |        10.62        |
| ESP           |      401       |       68       |         5.90        |
| CHN           |      398       |       45       |         8.84        |
| ITA           |      397       |       23       |        17.26        |
| GBR           |      343       |       32       |        10.72        |
| CAN           |      332       |       34       |         9.76        |
| BRA           |      299       |       35       |         8.29        |
| NED           |      299       |       20       |        14.50        |
| POL           |      226       |        7       |        32.29        |
| NZL           |      208       |       22       |         9.45        |
+-----+-----+-----+-----+

```

it displays 98 rows.

#### 8. List All Coaches Who Have Coached Athletes in More Than 30 Discipline

```

mysql> SELECT
->     c.coach_name,
->     COUNT(DISTINCT a.disciplines) AS DisciplineCount   -- Count distinct disciplines for each coach
-> FROM
->     COACH c
-> JOIN
->     ATHLETE a ON c.country_code = a.country_code
-> GROUP BY
->     c.coach_name                                     -- Group by coach_name
-> HAVING
->     COUNT(DISTINCT a.disciplines) > 30            -- display coaches involved more than 30 discipline
-> ORDER BY
->     c.coach_name ASC;                            -- ordering them based on coaches name alphabetically
+-----+-----+
| coach_name | DisciplineCount |
+-----+-----+
| ABARCA Megan |        47 |
| AGUILAR ESTIADA Eduardo |        39 |
| ALLENBURG Esteban |        39 |
| ALCANTARA Edson |        38 |
| ALLER CARBALLO Manuel Angel |        39 |
| ALONSO BERNARDO Jessica |        39 |
| ALONSO TAPIA Gulsah |        47 |
| ALONSO TAPIA Rosendo |        47 |
| ALONSO TAPIA Valentin |        42 |
| ANDERSON MUNIZ Daniel Edward |        42 |
| ANDERSON MUNIZ Las |        42 |
| ANDRE Felix |        47 |
+-----+-----+

```

it displays 488 rows

#### 9. Get the most medals won by each country, using limit get the top 10

```

mysql> select country_long as Country_Won_Most_Medals, Total
-> from TOTAL_MEDAL
-> order by
->     Total DESC
-> limit 10;
+-----+-----+
| Country_Won_Most_Medals | Total |
+-----+-----+
| United States of America | 126 |
| People's Republic of China | 91 |
| Great Britain | 65 |
| France | 64 |
| Australia | 53 |
| Japan | 45 |
| Italy | 40 |
| Netherlands | 34 |
| Germany | 33 |
| Republic of Korea | 32 |
+-----+-----+
10 rows in set (0.00 sec)

```

#### 10. get total\_events, total\_atheltes, total\_country and total medals select queries.

```

mysql> SELECT
->     (SELECT COUNT(DISTINCT event_id) FROM EVENT) AS Total_Events,
->     (SELECT COUNT(DISTINCT country_code) FROM ATHLETE) AS Total_Countries,
->     (SELECT COUNT(DISTINCT athlete_code) FROM ATHLETE) AS Total_Athletes,
->     (SELECT SUM(Gold_Medal + Silver_Medal + Bronze_Medal) FROM TOTAL_MEDAL) AS Total_Medals;
+-----+-----+-----+-----+
| Total_Events | Total_Countries | Total_Athletes | Total_Medals |
+-----+-----+-----+-----+
|      329      |       206       |     11113      |      1043      |
+-----+-----+-----+-----+
1 row in set (0.03 sec)

```

#### 11. calculates the male participants percentages in each discipline Using ATHLETE table

```

mysql> SELECT
->     a.disciplines AS Games_participated_by_male,
->     SUM(a.gender = 'M') AS Male,          -- counts male participants using Sum aggrigation function
->     COUNT(a.athlete_code) AS Total_athlete_num,      -- Total number of participants
->     ROUND((SUM(a.gender = 'M') / COUNT(a.athlete_code)) * 100, 2) AS Male_Participation -- Male participation percentage
->   FROM
->     ATHLETE a
->   GROUP BY
->     a.disciplines
->   ORDER BY
->     Male_Participation DESC;           -- ordering by descending of male participation

```

Games_participated_by_male	Male	Total_athlete_num	Male_Participation
[Athletics]	2	2	100.00
[Equestrian]	1	1	100.00
['Marathon Swimming', 'Swimming']	13	17	76.47
['Wrestling']	195	291	67.01
['Equestrian']	149	242	61.57
['Football']	313	553	56.00
['Handball']	156	286	54.55
['Swimming']	450	836	53.83
['Cycling Road']	90	174	51.72
['Taekwondo']	69	134	51.49
['Cycling Mountain Bike']	35	68	51.47
['Triathlon']	57	111	51.35
['Volleyball']	183	261	51.11
['Canoe Slalom']	49	84	51.19
['Hockey']	212	415	51.08
['Cycling Track']	114	224	50.89

it displays 52 rows.

## 12. calculates the Female participants percentages in each discipline Using ATHLETE table

```

mysql> SELECT
->     a.disciplines AS Games_participated_by_female,
->     SUM(a.gender = 'F') AS Female,
->     COUNT(a.athlete_code) AS Total_athlete_num,
->     ROUND((SUM(a.gender = 'F') / COUNT(a.athlete_code)) * 100, 2) AS Female_Participation
->   FROM
->     ATHLETE a
->   GROUP BY
->     a.disciplines
->   ORDER BY
->     Female_Participation DESC;

```

Games_participated_by_female	Female	Total_athlete_num	Female_Participation
['3x3 Basketball', 'Basketball']	1	1	100.00
['Cycling Road', 'Triathlon']	1	1	100.00
['Rhythmic Gymnastics']	94	94	100.00
['Artistic Swimming']	106	186	100.00
['Cycling Road', 'Cycling Track']	8	11	72.73
['Marathon Swimming']	20	38	52.63
['Handball']	202	386	52.33
['Breaking']	17	33	51.52
['Diving']	68	135	50.37
['Table Tennis']	88	175	50.29
['Canoe Sprint']	120	239	50.21
['Cycling BMX Freestyle']	12	24	50.00
['Trampoline Gymnastics']	16	32	50.00
['Golf']	60	120	50.00
['Skateboarding']	44	88	50.00
['Beach Volleyball']	48	96	50.00

it displays 52 rows.

## 13. get the coach name ASSOCIATED WITH COUNTRY AND won least 10 gold medals , used join and sub-query

```

mysql> SELECT
->     c.coach_name,
->     c.country_code AS COUNTRY,
->     c.disciplines AS GAMES
->   FROM
->     COACH c
->   INNER JOIN (
->     SELECT
->       country_code,
->       SUM(Gold_Medal) AS GOLD_TOTAL
->     FROM
->       TOTAL_MEDAL
->     GROUP BY
->       country_code
->     HAVING
->       SUM(Gold_Medal) >= 10
->   ) AS WINNER_OF_MEDAL
->
->   ON c.country_code = WINNER_OF_MEDAL.country_code;

```

coach_name	COUNTRY	GAMES
THORPE Karen	GBR	Artistic Swimming
TOMOMATSU Yumiko	GBR	Artistic Gymnastics
BUSNARI Alberto	ITA	Artistic Gymnastics
COCCIA RO Giuseppe	ITA	Artistic Gymnastics
FORTUNA Marco	ITA	Artistic Gymnastics
BERGAMELLI Monica Roberta	ITA	Artistic Gymnastics
ALTENBURG Valentin	GER	Hockey
HENNING Andre	GER	Hockey

it displays 388 rows.

## 14. retrieve the current age of the athlete using DATEDIFF format and renamed the table CURRENT\_AGE\_ATHELETS, name, gender, medal type and discipline from MEDALLIST TABLE

```

mysql> SELECT
->   name,
->   birth_date,
->   FLOOR(DATEDIFF(CURDATE(), birth_date) / 365.25) AS CURRENT_AGE_ATHLETS,
->   gender,
->   medal_type,
->   discipline
-> FROM
->   MEDALLIST;
+-----+-----+-----+-----+-----+
| name | birth_date | CURRENT_AGE_ATHLETS | gender | medal_type | discipline
+-----+-----+-----+-----+-----+
| CHANG Yani | 2001-12-07 | 22 | F | Gold Medal | Diving
| CHEN Yiwen | 1999-06-15 | 25 | F | Gold Medal | Diving
| BACON Sarah | 1996-09-20 | 28 | F | Silver Medal | Diving
| COOK Kasstdy | 1995-05-09 | 29 | F | Silver Medal | Diving
| HARPER Yasmin | 2000-07-28 | 24 | F | Bronze Medal | Diving
| MEW JENSEN Scarlett | 2001-12-31 | 22 | F | Bronze Medal | Diving
| PASQUET Varian | 1999-07-29 | 25 | M | Gold Medal | Rugby Sevens
| TIMO Andy | 2004-05-28 | 20 | M | Gold Medal | Rugby Sevens
| REBBADJ Rayan | 1999-08-15 | 25 | M | Gold Medal | Rugby Sevens
| FORNER Theo | 2001-10-17 | 23 | M | Gold Medal | Rugby Sevens
| PAREZ EDO MARTIN Stephen | 1994-08-01 | 30 | M | Gold Medal | Rugby Sevens
| RIVA Paulin | 1994-04-20 | 30 | M | Gold Medal | Rugby Sevens
| JOSEPH Jefferson-Lee | 2002-08-29 | 22 | M | Gold Medal | Rugby Sevens
| ZEGHDAR Antoine | 1999-05-22 | 25 | M | Gold Medal | Rugby Sevens
| GRANDIOTER Nkanang Aaron | 2000-05-18 | 24 | M | Gold Medal | Rugby Sevens
| BARRAQUE Jean Pascal | 1991-04-24 | 33 | M | Gold Medal | Rugby Sevens
| DUPONT Antoine | 1996-11-15 | 27 | M | Gold Medal | Rugby Sevens
| SEPHO Jordan | 1998-12-08 | 25 | M | Gold Medal | Rugby Sevens
+-----+-----+-----+-----+-----+

```

it displays 1555 rows

### 15. Compare medals between Female & Male

```

mysql> SELECT
->   gender,
->   discipline,
->   COUNT(*) AS total_medals,
->   SUM(CASE WHEN medal_type = 'Gold Medal' THEN 1 ELSE 0 END) AS Gold_Medals,
->   SUM(CASE WHEN medal_type = 'Silver Medal' THEN 1 ELSE 0 END) AS Silver_Medals,
->   SUM(CASE WHEN medal_type = 'Bronze Medal' THEN 1 ELSE 0 END) AS Bronze_Medals
-> FROM
->   MEDALLIST
-> GROUP BY
->   discipline, gender
-> ORDER BY
->   discipline;
+-----+-----+-----+-----+-----+
| gender | discipline | total_medals | Gold_Medals | Silver_Medals | Bronze_Medals |
+-----+-----+-----+-----+-----+
| F | 3x3 Basketball | 12 | 4 | 4 | 4 |
| M | 3x3 Basketball | 12 | 4 | 4 | 4 |
| F | Archery | 12 | 4 | 4 | 4 |
| M | Archery | 12 | 4 | 4 | 4 |
| F | Artistic Gymnastics | 15 | 5 | 5 | 5 |
| M | Artistic Gymnastics | 15 | 5 | 5 | 5 |
| F | Artistic Swimming | 33 | 11 | 11 | 11 |
| F | Athletics | 54 | 18 | 17 | 19 |
| M | Athletics | 50 | 18 | 15 | 17 |
| F | Badminton | 9 | 3 | 3 | 3 |
| M | Badminton | 9 | 3 | 3 | 3 |
+-----+-----+-----+-----+-----+

```

it display 54 rows

### 16. find out the Medals by discipline

```

mysql> SELECT discipline, COUNT(*) AS total_medals
-> FROM MEDALLIST
-> GROUP BY discipline
-> ORDER BY total_medals DESC;
+-----+-----+
| discipline | total_medals |
+-----+-----+
| Rowing | 138 |
| Swimming | 135 |
| Football | 124 |
| Athletics | 104 |
| Hockey | 102 |
| Handball | 94 |
| Rugby Sevens | 78 |
| Water Polo | 78 |
| Volleyball | 78 |
| Basketball | 72 |
| Fencing | 72 |
| Cycling Track | 69 |
| Canoe Sprint | 50 |
| Judo | 49 |
| Equestrian | 36 |
| Artistic Swimming | 33 |
| Artistic Gymnastics | 30 |
| Diving | 24 |
| 3x3 Basketball | 24 |
| Sailing | 24 |
| Table Tennis | 24 |
| Archery | 24 |
| Tennis | 18 |
| Badminton | 18 |
| Shooting | 10 |
+-----+-----+

```

shows 28 rows

17. list down the athletes age group and compare how many medals( medal\_type) they have win based on age group

```
mysql> SELECT
->     CASE
->         WHEN YEAR(CURDATE()) - YEAR(birth_date) < 20 THEN 'UNDER 20'
->         WHEN YEAR(CURDATE()) - YEAR(birth_date) BETWEEN 20 AND 29 THEN 'AGE between 20 to 29'
->         WHEN YEAR(CURDATE()) - YEAR(birth_date) BETWEEN 30 AND 39 THEN 'AGE between 30 to 39'
->         ELSE 'AGE OVER 40'
->     END AS ATHLETES_AGE_GROUP,
->     gender,
->     COUNT(*) AS MEDAL_TOTAL,
->     SUM(CASE WHEN medal_type = 'Gold Medal' THEN 1 ELSE 0 END) AS Gold_Medals,
->     SUM(CASE WHEN medal_type = 'Silver Medal' THEN 1 ELSE 0 END) AS Silver_Medals,
->     SUM(CASE WHEN medal_type = 'Bronze Medal' THEN 1 ELSE 0 END) AS Bronze_Medals
->   FROM
->     MEDALLIST
->   GROUP BY
->     ATHLETES_AGE_GROUP, gender
->   ORDER BY
->     MEDAL_TOTAL DESC, ATHLETES_AGE_GROUP, gender;
+-----+-----+-----+-----+-----+
| ATHLETES_AGE_GROUP | gender | MEDAL_TOTAL | Gold_Medals | Silver_Medals | Bronze_Medals |
+-----+-----+-----+-----+-----+
| AGE between 20 to 29 | F     |      547 |      173 |       194 |        180 |
| AGE between 20 to 29 | M     |      502 |      171 |       158 |        173 |
| AGE between 30 to 39 | M     |      221 |       75 |        79 |        67  |
| AGE between 30 to 39 | F     |      207 |       78 |        60 |        69  |
| UNDER 20            | F     |       35 |       10 |        13 |        12  |
| AGE OVER 40          | M     |       19 |        4 |        6 |        9   |
| UNDER 20            | M     |       15 |        2 |        8 |        5   |
| AGE OVER 40          | F     |       9 |        4 |        2 |        3   |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

18. find out the youngest athlete using birthdate from MEDALLIST TABLE, also name, code of the athlete, gender, discipline, events, medal\_type ,

```
mysql>
mysql> SELECT
->     name ,
->     code_athlete,
->     gender,
->     birth_date,
->     discipline,
->     events,
->     medal_type,
->     medal_date
->   FROM
->     MEDALLIST
-> WHERE birth_date = (SELECT MIN(birth_date) FROM MEDALLIST);
+-----+-----+-----+-----+-----+-----+-----+
| name      | code_athlete | gender | birth_date | discipline | events      | medal_type | medal_date |
+-----+-----+-----+-----+-----+-----+-----+
| KRAUT Laura | 1951840    | F     | 1965-11-14 | Equestrian | Jumping Team | Silver Medal | 2024-08-02 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

19: from MEDALLIST find out top Athletes by counting medal , use limit to get only top 10

```
mysql> SELECT name, country_code, COUNT(*) AS medal_count
->   FROM MEDALLIST
->   GROUP BY name, country_code
->   ORDER BY medal_count DESC
->   LIMIT 10;
+-----+-----+-----+
| name           | country_code | medal_count |
+-----+-----+-----+
| O'CALLAGHAN Mollie | AUS          |      4 |
| YANG Junxuan   | CHN          |      4 |
| HUSKE Torri    | USA          |      3 |
| McKEON Emma    | AUS          |      3 |
| WALSH Gretchen | USA          |      3 |
| ZHANG Yufei    | CHN          |      3 |
| DRESSEL Caeleb | USA          |      3 |
| JACK Shayna    | AUS          |      2 |
| SOUTHAM Flynn  | AUS          |      2 |
| HARRIS Meg     | AUS          |      2 |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

20. from SCGEDULE TABLE get the start\_date

```

mysql> SET time_zone = '+00:00' ;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @@session.time_zone;
+-----+
| @@session.time_zone |
+-----+
| +00:00 |
+-----+
1 row in set (0.00 sec)

mysql> select start_date from SCHEDULE;
+-----+
| start_date      |
+-----+
| 2024-07-24 13:00:00 |
| 2024-07-24 13:00:00 |
| 2024-07-24 13:30:00 |
| 2024-07-24 14:00:00 |
| 2024-07-24 14:30:00 |
| 2024-07-24 15:00:00 |
| 2024-07-24 15:00:00 |
| 2024-07-24 15:00:00 |
| 2024-07-24 15:30:00 |
| 2024-07-24 16:00:00 |
| 2024-07-24 17:00:00 |
| 2024-07-24 17:00:00 |
| 2024-07-24 17:00:00 |
| 2024-07-24 17:30:00 |
| 2024-07-24 18:00:00 |
| 2024-07-24 18:30:00 |
| 2024-07-24 19:00:00 |
| 2024-07-24 19:00:00 |
| 2024-07-24 19:00:00 |

```

returns 3895 rows displaying time in UTC

**21. from SCHEDULE TABLE get the start\_date, end\_date , vanue\_code when status is cancelled**

```

mysql> SELECT start_date, end_date, venue_code FROM SCHEDULE WHERE status = 'CANCELLED';
+-----+-----+-----+
| start_date | end_date | venue_code |
+-----+-----+-----+
| 2024-07-29 06:30:00 | 2024-07-29 06:30:00 | CPL
| 2024-07-29 12:00:00 | 2024-07-29 12:00:00 | CPL
| 2024-07-30 12:58:00 | 2024-07-30 12:58:00 | CPL
| 2024-07-31 07:28:00 | 2024-07-31 07:28:00 | CPL
| 2024-08-01 11:36:00 | 2024-08-01 11:57:00 | MAM
| 2024-08-01 11:56:00 | 2024-08-01 12:17:00 | MAM
| 2024-08-01 12:07:00 | 2024-08-01 12:28:00 | MAM
| 2024-08-01 12:27:00 | 2024-08-01 12:48:00 | MAM
| 2024-08-01 15:38:00 | 2024-08-01 15:59:00 | MAM
| 2024-08-01 15:38:00 | 2024-08-01 15:59:00 | MAM
| 2024-08-01 16:09:00 | 2024-08-01 16:30:00 | MAM
| 2024-08-01 16:28:00 | 2024-08-01 16:49:00 | MAM
| 2024-08-01 16:28:00 | 2024-08-01 16:49:00 | MAM
| 2024-08-01 16:48:00 | 2024-08-01 17:01:00 | MAM
| 2024-08-01 16:59:00 | 2024-08-01 17:20:00 | MAM
| 2024-08-01 17:30:00 | 2024-08-01 17:51:00 | MAM
| 2024-08-01 18:01:00 | 2024-08-01 18:22:00 | MAM
| 2024-08-05 11:23:00 | 2024-08-05 12:14:00 | MAM
| 2024-08-05 12:45:00 | 2024-08-05 13:36:00 | MAM
| 2024-08-05 13:58:00 | 2024-08-05 14:41:00 | MAM
| 2024-08-06 12:29:00 | 2024-08-06 13:19:00 | MAM
| 2024-08-06 13:36:00 | 2024-08-06 14:26:00 | MAM
| 2024-08-07 10:03:00 | 2024-08-07 10:28:00 | MAM
| 2024-08-07 10:23:00 | 2024-08-07 10:40:00 | MAM
| 2024-08-07 10:43:00 | 2024-08-07 11:00:00 | MAM
| 2024-08-07 10:48:00 | 2024-08-07 11:05:00 | MAM
| 2024-08-07 11:13:00 | 2024-08-07 11:30:00 | MAM
| 2024-08-07 11:23:00 | 2024-08-07 11:40:00 | MAM
| 2024-08-07 11:38:00 | 2024-08-07 11:55:00 | MAM
| 2024-08-07 12:03:00 | 2024-08-07 12:20:00 | MAM

```

it displays 70 rows

However, by sourcing **Query.sql** we can implement all these queries at a time and also get all the result in a output file named “**SimpleQueryAnswerOlympic.out**“

## **6.implementation of advanced features like PROCEDURE , VIEW**

in order to implement the advance features of queries by using Procedure like user variable, if-else, cursor, store procedure In, out parameter one can source the file name “**advanceQuery.sql**” which also have the output file named “**advanceQueryAnswerOlympic.out**”. now we will do the queries one by one.

### **Procedure \_TASK1 :**

store procedure to calculate the **total medals** on a given game and gender .

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE CAL_TOTAL_MEDALS(
->     IN PRO_discipline VARCHAR(200),
->     IN PRO_gender CHAR(1),
->     OUT TotalMedals INT    -- this will return the total medal number
->
-> )
-> COMMENT 'calucaltion of total media fro a dicipline and gender'
-> BEGIN
->     -- declaring variable (Existsdiscipline,TotalCount )
->     DECLARE Existsdiscipline INT DEFAULT 0;  -- checks if discipline exits or not
->     DECLARE TotalCount INT DEFAULT 0;   -- this var hold the total count
->
->     SET TotalMedals = 0;
->
->     -- checkings the discilines exixtence
->     SELECT COUNT(*) INTO Existsdiscipline
->     FROM MEDALLIST
->     WHERE discipline = PRO_discipline;
->
->     -- dicipline does not exit, so display msg
->     IF Existsdiscipline = 0 THEN
->         -- SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Discipline does not exist.';
->         SELECT CONCAT('Sorry!!! Discipline you have entered :', PRO_discipline, ' does not exist.') AS Total_Medal_CALCUALTION ;
->     ELSE
->
->         -- calculates total medal number
->         SELECT COUNT(*) INTO TotalCount
->         FROM MEDALLIST
->         WHERE discipline = PRO_discipline AND gender = PRO_gender;
->
->         -- now, the result(TotalCount) are assigning to out parameter which is  TotalMedals
->         SET TotalMedals = TotalCount;
->
->     END IF;
-> END //
Query OK, 0 rows affected (0.02 sec)
```

1.Then calling the procedure for **total of Male gender in Football** and displaying the total using user variable

```
mysql> CALL CAL_TOTAL_MEDALS('Football', 'M', @total);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @total AS TOTAL_MEDALS;
+-----+
| TOTAL_MEDALS |
+-----+
|      62      |
+-----+
1 row in set (0.00 sec)
```

2. Now, calling a PROCEDURE for a discipline that does not exit and return 0 as a total medal

```

mysql> CALL CAL_TOTAL_MEDALS('Ludo', 'M', @total);
+-----+
| Total_Medal_CALCUALTION |
+-----+
| Sorry!!! Discipline you have entered :Ludo does not exist. |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @total AS TOTAL_MEDALS;
+-----+
| TOTAL_MEDALS |
+-----+
|          0 |
+-----+
1 row in set (0.00 sec)

```

**3. For “water Polo” discipline get out the total medals in male gender**

```

mysql> CALL CAL_TOTAL_MEDALS('Water Polo', 'M', @total);
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> SELECT @total AS TOTAL_MEDALS;
+-----+
| TOTAL_MEDALS |
+-----+
|          39 |
+-----+
1 row in set (0.00 sec)

```

[Procedure \\_TASK2](#):

using procedure user variable that **counting total medals** that have been won in a **discipline**

```

mysql> DROP PROCEDURE IF EXISTS Counting_Medal_By_discipline;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE Counting_Medal_By_discipline(
    >     IN discipline_name VARCHAR(200), -- input for discipline name
    >     OUT medal_count INT           -- out parameter for counting medals
    > )
    >
    > COMMENT ' COUNTING total medal won on a specific discipline .'
    > BEGIN
    >     SELECT COUNT(*) INTO medal_count
    >     FROM MEDALLIST
    >     WHERE discipline = discipline_name;
    > END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL Counting_Medal_By_discipline('Handball', @totalMedals);
Query OK, 1 row affected (0.01 sec)

```

**1. Calling procedure for handball to get the total medals for ti**

```

mysql> CALL Counting_Medal_By_discipline('Handball', @totalMedals);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @totalMedals AS Total_Medals;
+-----+
| Total_Medals |
+-----+
|          94 |
+-----+
1 row in set (0.00 sec)

```

**2. Calling procedure to get the total medals for table tennis**

```

mysql> CALL Counting_Medal_By_discipline('Table Tennis', @totalMedals);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @totalMedals AS Total_Medals;
+-----+
| Total_Medals |
+-----+
|          24 |
+-----+
1 row in set (0.00 sec)

```

### *Procedure \_TASK3*

*store procedure using Cursor and LOOP to count medals ( gold medals, silzer emdals, bronze medals) on a discipline*

```

mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE Count_Discipline(
->     IN discipline_name VARCHAR(200),
->     OUT count_gold INT,           -- out parameter for gold medal counting
->     OUT count_silver INT,        -- out parameter for silver medal counting
->     OUT count_bronze INT
-> )
-> BEGIN
->     -- declarign var (var_medal_type, done)
->     DECLARE var_medal_type VARCHAR(20);
->     DECLARE done INT DEFAULT 0; -- To control loop exit
->
->     -- declarign the cursor
->     DECLARE cursor_medal CURSOR FOR
->         SELECT medal_type
->             FROM MEDALLIST
->             WHERE discipline = discipline_name;
->
->
->     -- declar not found handler
->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
->
-> /*
/*> -- initializing counts
/*>     SET count_gold = 0;
/*>     SET count_silver = 0;
/*>     SET count_bronze = 0;
/*>
/*> */
->     -- open cursor(cursor_medal)
->     OPEN cursor_medal;
->
-> /*
->     -- start loop
->     StartLoop: LOOP
->
->         -- fetch the cursor(cursor_medal) into variable( var_medal_type)
->
->         FETCH cursor_medal INTO var_medal_type;
->
->         -- checks cursor is done
->         IF done = 1 THEN
->             LEAVE StartLoop;
->         END IF;
->
->
->         -- checks the medal type by using if else
->         IF var_medal_type = 'Gold' THEN
->             SET count_gold = count_gold + 1;
->         ELSEIF var_medal_type = 'Silver' THEN
->             SET count_silver = count_silver + 1;
->         ELSEIF var_medal_type = 'Bronze' THEN
->             SET count_bronze = count_bronze + 1;
->         END IF;
->     END LOOP;
->
->     -- closing cursor
->     CLOSE cursor_medal;
-> END //
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;

```

**1.** Calling the procedure for Water Polo to count the Gold Medal, silver medal, bronze medal

```

mysql> CALL Count_Discipline ('Water Polo', @count_gold, @count_silver, @count_bronze);
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @count_gold AS Gold_Medals, @count_silver AS Silver_Medals, @count_bronze AS Bronze_Medals;
+-----+-----+-----+
| Gold_Medals | Silver_Medals | Bronze_Medals |
+-----+-----+-----+
|      NULL |        NULL |        NULL |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> 

```

Some reason, it is not showing me the proper value. Showing me NULL.

**2.** In order to DELETE all the procedure we can use this command like this

```
mysql> DROP PROCEDURE IF EXISTS CAL_TOTAL_MEDALS;
Query OK, 0 rows affected (0.03 sec)

mysql> DROP PROCEDURE IF EXISTS Counting_Medal_By_discipline;
Query OK, 0 rows affected (0.01 sec)

mysql> █
```

3. SHOW PROCEDURE STATUS WHERE Db='Olympic\_Game\_2024\_22663281';

```
mysql> SHOW PROCEDURE STATUS WHERE Db='Olympic_Game_2024_22663281'; -- display all procedure in a specific database
+-----+-----+-----+-----+-----+-----+-----+-----+
| Db      | Name        | Type    | Definer   | Modified   | Created   | Security_type | Comment          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| character_set_client | collation_connection | Database Collation |           |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Olympic_Game_2024_22663281 | CAL_TOTAL_MEDALS | PROCEDURE | dsuser@localhost | 2024-10-24 17:23:12 | 2024-10-24 17:23:12 | DEFINER       | calcualtion of total media fro a discipline and gender |
utf8mb4
| utf8mb4_0900_ai_ci | utf8mb4_0900_ai_ci |           |
| Olympic_Game_2024_22663281 | Counting_Medal_By_discipline | PROCEDURE | dsuser@localhost | 2024-10-24 17:24:09 | 2024-10-24 17:24:09 | DEFINER       | COUNTING total medal won on a specific discipline . |
utf8mb4
| utf8mb4_0900_ai_ci | utf8mb4_0900_ai_ci |           |
| Olympic_Game_2024_22663281 | Count_Discipline | PROCEDURE | dsuser@localhost | 2024-10-24 17:14:32 | 2024-10-24 17:14:32 | DEFINER       |           |
utf8mb4
| utf8mb4_0900_ai_ci | utf8mb4_0900_ai_ci |           |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

### **VIEW TASK1 :**

Create VIEW name "MEDALLIST\_TEAM" by joining MEDALLIST and TEAM table

where MEDALLIST medal date medal type, medalist id, code athlete , birth date, name and

*FROM team discipline and events attribute will be included.*

```
mysql> CREATE VIEW MEDALLIST_TEAM AS
-> SELECT
->     m.medal_date AS DATE_OF_MEDAL,
->     m.medal_type AS TYPE_OF_MEDAL,
->     m.medalist_id AS MEDALLIST_ID,
->     m.code_athlete ATHELTE_CODE_M,
->     m.birth_date AS BIRTHDATE,
->     m.name AS NAME,
->     e.code AS TEAM_CODE,
->     e.discipline AS GAMES,
->     e.events AS EVENT_LIST
->
-> FROM
->     MEDALLIST m
-> JOIN
->     TEAM e
-> ON
->     m.code_team = e.code;
Query OK, 0 rows affected (0.02 sec)
```

*By this command we can retrieve data from view MEDALLIST TEAM*

Medalist Team Data										
DATE_OF_MEDAL	TYPE_OF_MEDAL	MEDALLIST_ID	ATHLETE_CODE_M	BIRTHDATE	NAME	TEAM_CODE	GAMES	EVENT_LIST	REMARKS	
2024-07-27	Gold Medal	1	1901539	2001-12-07	CHANG Yani	DIV3MTEAM2-CHN01	Diving	Women's Synchronised 3m Springboard		
2024-07-27	Gold Medal	2	1901538	1999-06-15	CHEN Yilwen	DIV3MTEAM2-CHN01	Diving	Women's Synchronised 3m Springboard		
2024-07-27	Silver Medal	3	1954159	1996-09-20	BACON Sarah	DIV3MTEAM2-USA01	Diving	Women's Synchronised 3m Springboard		
2024-07-27	Silver Medal	4	1954168	1995-05-09	COOK Cassidy	DIV3MTEAM2-USA01	Diving	Women's Synchronised 3m Springboard		
2024-07-27	Bronze Medal	5	1900841	2000-07-28	HARPER Yasmin	DIV3MTEAM2-GBR01	Diving	Women's Synchronised 3m Springboard		
2024-07-27	Bronze Medal	6	1973355	2001-12-31	MEN JENSEN Scarlett	DIV3MTEAM2-GBR01	Diving	Women's Synchronised 3m Springboard		
2024-07-27	Gold Medal	7	1789782	1999-07-29	PASQUET Varian	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	8	1789607	2000-05-26	PEYRE Arnaud	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	9	1789683	1999-07-29	ROBADIO Rayan	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	10	1789761	2001-10-17	FORNER Theo	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	11	1789681	1994-08-01	PAREZ EDO MARTIN Stephen	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	12	1789684	1994-04-20	RIVA Paulin	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	13	1789675	2002-08-29	JOSEPH Jefferson-Lee	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	14	1789692	1999-05-22	ZEGHDAR Antoine	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	15	1789672	2000-05-18	GRANDIDIER NIKANANG Aaron	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	16	1789661	1991-04-24	BARRAQUE Jean Pascal	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	17	1789669	1996-11-15	DUPONT Antoine	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	18	1789685	1998-12-08	SEPHO Jordan	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Gold Medal	19	1789679	2000-05-20	ELIAS Nilsen	RUT7MTEAM---FRA01	Rugby Sevens	Men		
2024-07-27	Silver Medal	20	1789697	2000-06-01	MASOVAI Paul	RUT7MTEAM---FIJ01	Rugby Sevens	Men		
2024-07-27	Silver Medal	21	1988638	1997-04-01	TALACOLO Joseva	RUT7MTEAM---FIJ01	Rugby Sevens	Men		
2024-07-27	Silver Medal	22	1913346	1998-07-14	MATANA Jeremala	RUT7MTEAM---FIJ01	Rugby Sevens	Men		
2024-07-27	Silver Medal	23	1977633	1990-06-29	MOCENAGAT Sevuloni	RUT7MTEAM---FIJ01	Rugby Sevens	Men		
2024-07-27	Copper Medal	24	1913253	1998-05-01	BALEWATUMA Teofilo	RUT7MTEAM---ETI01	Rugby Sevens	Men		

## VIEW\_TASK2:

update name in **MEDALLIST TABLE** and checked on **VIEW MEDALLIST\_TEAM** if the update was successful or not by calling procedure

let's say **medallist\_id 1159 changed the name and checked**

```
mysql> SELECT NAME FROM MEDALLIST_TEAM where MEDALLIST_ID = '1159';
+-----+
| NAME      |
+-----+
| van de WIEL Anne |
+-----+
```

1<sup>st</sup> checking the name in **VIEW MEDALLIST\_TEAM** before updating

```
mysql> SELECT name FROM MEDALLIST where medalist_id = '1159';
+-----+
| name      |
+-----+
| van de WIEL Anne |
+-----+
1 row in set (0.01 sec)
```

now checking on **MEDALIST TABLE**

```
mysql> UPDATE MEDALLIST
    -> SET name = 'dhrubo das'
    -> WHERE medalist_id = '1159' ;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Updating the name IN **MEDALLIST TABLE**

```
mysql> SELECT NAME FROM MEDALLIST_TEAM where MEDALLIST_ID = '1159';
+-----+
| NAME      |
+-----+
| dhrubo das |
+-----+
1 row in set (0.00 sec)

mysql> SELECT name FROM MEDALLIST where medalist_id = '1159';
+-----+
| name      |
+-----+
| dhrubo das |
+-----+
1 row in set (0.00 sec)
```

Again checking in **MEDALLIST Table & MEDALLIST\_TEAM VIEW** if the update was successful or not

This query shows all the **VIEWS in the DATABASE**

```
mysql> SHOW FULL TABLES IN Olympic_Game_2024_22663281 WHERE TABLE_TYPE = 'VIEW';
+-----+-----+
| Tables_in_Olympic_Game_2024_22663281 | Table_type |
+-----+-----+
| MEDALLIST_TEAM                      | VIEW       |
+-----+-----+
1 row in set (0.01 sec)
```

This query shows the Specific **VIEW in Details**

```
mysql> SHOW CREATE VIEW MEDALLIST_TEAM ;
+-----+-----+
| View      | Create View
+-----+-----+
| MEDALLIST_TEAM | CREATE ALGORITHM=UNDEFINED DEFINER='dsuser'@'localhost' SQL SECURITY DEFINER VIEW `MEDALLIST_TEAM` AS select `m`.`medal_date` AS `DATE_OF_MEDAL`, `m`.`medal_type` AS `TYPE_OF_MEDAL`, `m`.`medallist_id` AS `MEDALLIST_ID`, `m`.`code_athlete` AS `ATHLETE_CODE_M`, `m`.`birth_date` AS `BIRTHDATE`, `m`.`name` AS `NAME`, `e`.`code` AS `TEAM_CODE`, `e`.`discipline` AS `GAMES`, `e`.`events` AS `EVENT_LIST` from (`MEDALLIST` `m` join `TEAM` `e` on(`m.code_team` = `e.code`)) | utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)
```

Query to drop the **VIEW**

```
mysql> DROP VIEW MEDALLIST_TEAM;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

## 7. Connecting Python to MYSQL DATABASE

I have this python files to do some operation .

- *python3 eventdata.py*
- *python3 scheduleData.py*
- *python3 pythonConnect.py*
- *pyhton3 insert.py*

7.1 To connect MYSQL Server with python environment need to do this at first like this:

```
STUDENT\22663281@v-2204-hcs-164:/tsclient/dhrub/Downloads/OneDrive_2024-10-23/tr_full_Checkd - Copy_PCC$ mysql -V
mysql Ver 8.0.39-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
STUDENT\22663281@v-2204-hcs-164:/tsclient/dhrub/Downloads/OneDrive_2024-10-23/tr_full_Checkd - Copy_PCC$ pip3 install mysql-connector-python
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.1.0-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (6.0 kB)
  Downloading mysql_connector_python-9.1.0-cp310-cp310-manylinux_2_28_x86_64.whl (34.4 MB)
           34.4/34.4 MB 54.2 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.1.0
STUDENT\22663281@v-2204-hcs-164:/tsclient/dhrub/Downloads/OneDrive_2024-10-23/tr_full_Checkd - Copy_PCC$
```

7.2 *pythonConnect.py* this file shows that we can connect the MYSQL Server with python environment.

```

1 # this file contains the details of how to connect python to SQL SERVER
2
3 import mysql.connector
4 import getpass
5
6
7 # Read username and password from the user
8 username = input("Enter MySQL username: ")
9 password = getpass.getpass("Enter MySQL password: ")
10
11 try:
12     # Create a connection to the database
13     conn = mysql.connector.connect(
14         host='localhost',
15         user=username,
16         password=password,
17         database='Olympic_Game_2024_22663281'
18     )
19     if conn.is_connected():
20         db_info = conn.get_server_info()
21         print(f"Connected to MySQL Server version {db_info}")
22         cursor = conn.cursor()
23         cursor.execute("SELECT DATABASE();")
24         record = cursor.fetchall()
25         print(f"You're connected to database: {record[0]}")
26
27
28
29 except mysql.connector.Error as err:
30     print(f"Error: {err}")
31 finally:
32     if conn.is_connected():
33         cursor.close()
34         conn.close()
35     print("MySQL connection is closed.")

```

```

STUDENT\22663281@v-2204-hcs-164:~/tsclient/dhrub/Downloads/OneDrive_2024-10-23/tr_full_Checkd - Copy_PCC$ python3 pythonConnect.py
Enter MySQL username: dsuser
Enter MySQL password:
Connected to MySQL Server version 8.0.39-0ubuntu0.22.04.1
You're connected to database: ('Olympic_Game_2024_22663281',)
MySQL connection is closed.

```

### 7.3 evendata.py -- to retrieve all rows of all EVENT table and display using fetchone() command..

```

#evendata.py -- Add code to retrieve all rows of all EVENT table and display using fetchone() command..
import mysql.connector
import getpass

# Read username and password from the user
username = input("Enter MySQL username: ")
password = getpass.getpass("Enter MySQL password: ")

try:
    # Create a connection to the database
    conn = mysql.connector.connect(
        host='localhost',
        user=username,
        password=password,
        database='Olympic_Game_2024_22663281'
    )
    if conn.is_connected():
        db_info = conn.get_server_info()
        print(f"Connected to MySQL Server version {db_info}")
        cursor = conn.cursor()
        cursor.execute("SELECT DATABASE();")
        record = cursor.fetchone()
        print(f"You're connected to database: {record[0]}")

        # Add the code to retrieve and display rows from the Emp table
        select_query1 = "SELECT * FROM EVENT"
        cursor.execute(select_query1)

        # Get all rows
        rows = cursor.fetchall()

        # Print the first four columns of all rows
        for row in rows:
            print(row[0], row[1], ", ", row[2], ", ", row[3]) # Adjusted to match the first four columns of the EVENT table

except mysql.connector.Error as err:
    print(f"Error: {err}")
finally:
    if conn.is_connected():
        cursor.close()
        conn.close()
    print("MySQL connection is closed.")

```

*The above code we imported library MySQL connector. to interacts with SQL server.*

The fetchall() statement fetch all the rows from result table . The exception given prints the error if occurred. Then finally close the cursor and then the connection.

```
NameError: name 'conn' is not defined
STUDENT[22663281@v-2204-hcs-164:~/tsclient/dhrub/Downloads/OneDrive_2024-10-23/tr_full_Checkd - Copy_PCK$ python3 eventdata.py
Enter MySQL username: dsuser
Enter MySQL password:
Connected to MySQL Server version 8.0.39-Ubuntu0.22.04.1
You're connected to database: Olympic_Game_2024_22663281
Men's Individual Archery , ARC , 1
Women's Individual Archery , ARC , 2
Men's Team Archery , ARC , 3
Women's Team Archery , ARC , 4
Mixed Team Archery , ARC , 5
Men's Team Artistic Gymnastics , GAR , 6
Men's All-Around Artistic Gymnastics , GAR , 7
Men's Floor Exercise Artistic Gymnastics , GAR , 8
Men's Pommel Horse Artistic Gymnastics , GAR , 9
Men's Rings Artistic Gymnastics , GAR , 10
Men's Vault Artistic Gymnastics , GAR , 11
Men's Parallel Bars Artistic Gymnastics , GAR , 12
Men's Horizontal Bar Artistic Gymnastics , GAR , 13
Women's Team Artistic Gymnastics , GAR , 14
Women's All-Around Artistic Gymnastics , GAR , 15
Women's Vault Artistic Gymnastics , GAR , 16
Women's Uneven Bars Artistic Gymnastics , GAR , 17
Women's Balance Beam Artistic Gymnastics , GAR , 18
Women's Floor Exercise Artistic Gymnastics , GAR , 19
Duet Artistic Swimming , SWA , 20
Team Artistic Swimming , SWA , 21
Men's 100m Athletics , ATH , 22
Men's 200m Athletics , ATH , 23
Men's 400m Athletics , ATH , 24
Men's 800m Athletics , ATH , 25
Men's 1500m Athletics , ATH , 26
Men's 5000m Athletics , ATH , 27
Men's 10,000m Athletics , ATH , 28
Men's Marathon Athletics , ATH , 29
Men's 3000m Steeplechase Athletics , ATH , 30
Men's 110m Hurdles Athletics , ATH , 31
Men's 400m Hurdles Athletics , ATH , 32
Men's High Jump Athletics , ATH , 33
Men's Pole Vault Athletics , ATH , 34
Men's Long Jump Athletics , ATH , 35
Men's Triple Jump Athletics , ATH , 36
Men's Shot Put Athletics , ATH , 37
```

#### **7.4 schdeule.py this pyhton file retrive the data of SCEHEDULE table such as end\_date, start\_date, vanue\_code using where conditon status = CANCELLED**

```
# schdeule.py this pyhton file retrive the data of SCEHEDULE table such as end_date, start_date, vanue_code using where conditon status = CANCELLED
import mysql.connector
import getpass

# Read username and password from the user
username = input("Enter MySQL username: ")
password = getpass.getpass("Enter MySQL password: ")

try:
    # Create a connection to the database
    conn = mysql.connector.connect(
        host='localhost',
        user=username,
        password=password,
        database='Olympic_Game_2024_22663281'
    )
    if conn.is_connected():
        db_info = conn.get_server_info()
        print(f"Connected to MySQL Server version {db_info}")
        cursor = conn.cursor()
        cursor.execute("SELECT DATABASE();")
        record = cursor.fetchall()
        print(f"You're connected to database: {record[0]}")

    # Retrieve and display the employee's details using the hardcoded employee number
    select_query = """SELECT start_date, end_date, venue_code
                      FROM SCHEDULE WHERE
                      status = 'CANCELLED' """

    cursor.execute(select_query)

    # Fetch and display the result
    rows = cursor.fetchall()
    if rows:
        for row in rows:
            print(f" start date: {row[0]}, end date: {row[1]}, venue code: {row[2]}")

    else:
        print("No schedule events with status found.")

except mysql.connector.Error as err:
    print(f"Error: {err}")
finally:
    if conn.is_connected():
        cursor.close()
        conn.close()
        print("MySQL connection is closed.")
```

*Shows me the result of start\_date, end date and venue code that have been cancelled*

```
STUDENT\22663281@v-2204-hcs-164:~/tsclient/dhrub/Downloads/OneDrive_2024-10-23/tr_full_Checkd - Copy_PCC$ python3 scheduleData.py
Enter MySQL username: dsuser
Enter MySQL password:
Connected to MySQL Server version 8.0.39-0ubuntu0.22.04.1
You're connected to database: ('Olympic_Game_2024_22663281',)
start date: 2024-07-29 14:30:00, end date: 2024-07-29 14:30:00, venue code: CPL
start date: 2024-07-29 20:00:00, end date: 2024-07-29 20:00:00, venue code: CPL
start date: 2024-07-30 20:50:00, end date: 2024-07-30 20:50:00, venue code: CPL
start date: 2024-07-31 15:20:00, end date: 2024-07-31 15:20:00, venue code: CPL
start date: 2024-08-01 19:36:00, end date: 2024-08-01 19:57:00, venue code: MAM
start date: 2024-08-01 19:56:00, end date: 2024-08-01 20:17:00, venue code: MAM
start date: 2024-08-01 20:07:00, end date: 2024-08-01 20:28:00, venue code: MAM
start date: 2024-08-01 20:27:00, end date: 2024-08-01 20:48:00, venue code: MAM
start date: 2024-08-01 23:38:00, end date: 2024-08-01 23:59:00, venue code: MAM
start date: 2024-08-01 23:38:00, end date: 2024-08-01 23:59:00, venue code: MAM
start date: 2024-08-02 00:09:00, end date: 2024-08-02 00:30:00, venue code: MAM
start date: 2024-08-02 00:28:00, end date: 2024-08-02 00:49:00, venue code: MAM
start date: 2024-08-02 00:28:00, end date: 2024-08-02 00:49:00, venue code: MAM
start date: 2024-08-02 00:40:00, end date: 2024-08-02 01:01:00, venue code: MAM
start date: 2024-08-02 00:59:00, end date: 2024-08-02 01:20:00, venue code: MAM
start date: 2024-08-02 01:30:00, end date: 2024-08-02 01:51:00, venue code: MAM
start date: 2024-08-02 02:01:00, end date: 2024-08-02 02:22:00, venue code: MAM
start date: 2024-08-05 19:23:00, end date: 2024-08-05 20:14:00, venue code: MAM
start date: 2024-08-05 20:45:00, end date: 2024-08-05 21:36:00, venue code: MAM
start date: 2024-08-05 21:50:00, end date: 2024-08-05 22:41:00, venue code: MAM
start date: 2024-08-06 20:29:00, end date: 2024-08-06 21:19:00, venue code: MAM
start date: 2024-08-06 21:36:00, end date: 2024-08-06 22:26:00, venue code: MAM
start date: 2024-08-07 18:03:00, end date: 2024-08-07 18:20:00, venue code: MAM
start date: 2024-08-07 18:23:00, end date: 2024-08-07 18:40:00, venue code: MAM
start date: 2024-08-07 18:43:00, end date: 2024-08-07 19:00:00, venue code: MAM
start date: 2024-08-07 18:48:00, end date: 2024-08-07 19:05:00, venue code: MAM
start date: 2024-08-07 19:13:00, end date: 2024-08-07 19:30:00, venue code: MAM
start date: 2024-08-07 19:23:00, end date: 2024-08-07 19:40:00, venue code: MAM
start date: 2024-08-07 19:38:00, end date: 2024-08-07 19:55:00, venue code: MAM
start date: 2024-08-07 20:03:00, end date: 2024-08-07 20:20:00, venue code: MAM
start date: 2024-08-07 20:03:00, end date: 2024-08-07 20:20:00, venue code: MAM
start date: 2024-08-07 20:43:00, end date: 2024-08-07 21:00:00, venue code: MAM
start date: 2024-08-07 22:00:00, end date: 2024-08-07 22:17:00, venue code: MAM
start date: 2024-08-07 22:00:00, end date: 2024-08-07 22:17:00, venue code: MAM
start date: 2024-08-07 22:00:00, end date: 2024-08-07 22:17:00, venue code: MAM
```

#### 7.4 :

#### **Discussion**

[ 5 marks] Reflect on your own work, including a summary of what you have achieved, challenges you have faced, limitations and ways to improve your work with other features you have not considered, and any other information you wish to present.