USER GUIDE

Olympic 2020 Tokyo DB

Muhammad Hashir
2163443

Muhammad Hashir
21634493

USER GUIDE

Folder includes:

- commands(directory)
- images(directory)
- References(directory)
- HashirMuhammad_21634493_DbSReport2024.pdf (file)
- UserGuide.pdf (file)

Please make sure you are in a Linux environment

Information of folder and files

commands (directory):

- **create_databases.sql :** Used to create the database tables
- **olympics_2020_queries.sql :** Contains the level 1 and level 2 queries
- **insert_sample_values.sql :** Inserts the sample data into the tables
- **advanced_queries.sql :** Contains advanced features
- **db_config.json :** JSON file used for securing credentials
- **connection.py :** python code to access the database using connection

images (directory):

- **Evidence(directory) :** Contains all evidence images

References (directory):

- **References.docx :** Contains references to online sources used

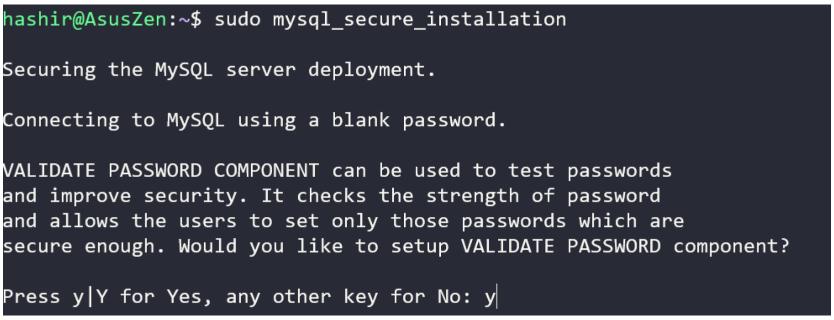
Steps to install mysql:

Run the following commands in your terminal

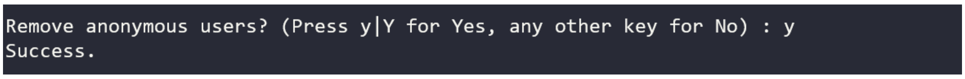
1. sudo apt update && sudo apt upgrade -y
2. sudo apt install mysql-server -y
3. sudo systemctl start mysql
4. sudo systemctl enable mysql
5. sudo mysql_secure_installation

Prompt this for permissions for mysql when you run sudo mysql_secure_installation

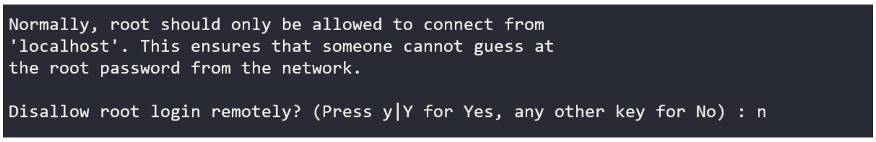
Muhammad Hashir
21634493



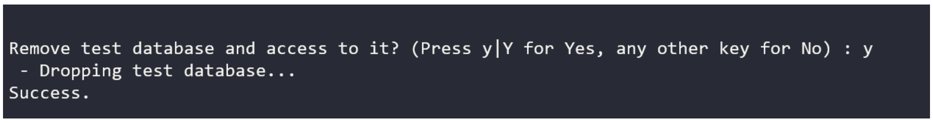
```
hashir@AsusZen:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y
```



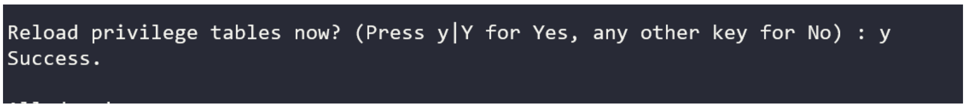
```
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.
```



```
Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n
```



```
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
 - Dropping test database...
Success.
```



```
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.
```

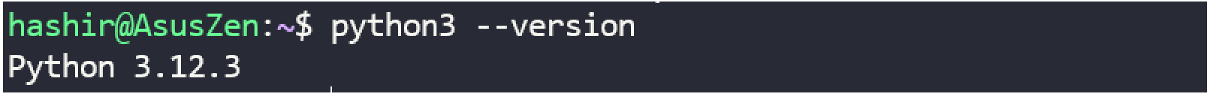
6. sudo mysql
    a. ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '<YourPassword>';
    b. FLUSH PRIVILEGES;
    c. exit

mysql will be successfully installed if all the commands are followed, preferred password to keep would be 123456789.

Steps to intsall Python and its dependencies for the Assignment

Run the following commands in terminal:

1. sudo apt install python3
2. python3 --version (shows version of python)



```
hashir@AsusZen:~$ python3 --version
Python 3.12.3
```

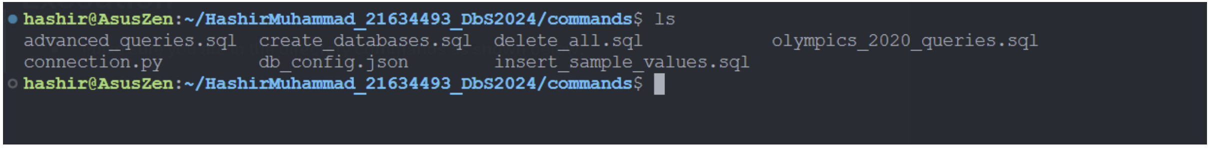
3. sudo apt install python3-pip
4. pip install mysql.connector

the commands above makes sure python and its dependencies are installed.

After installing all the dependencies for the assignment, its time to execute and create the tables.

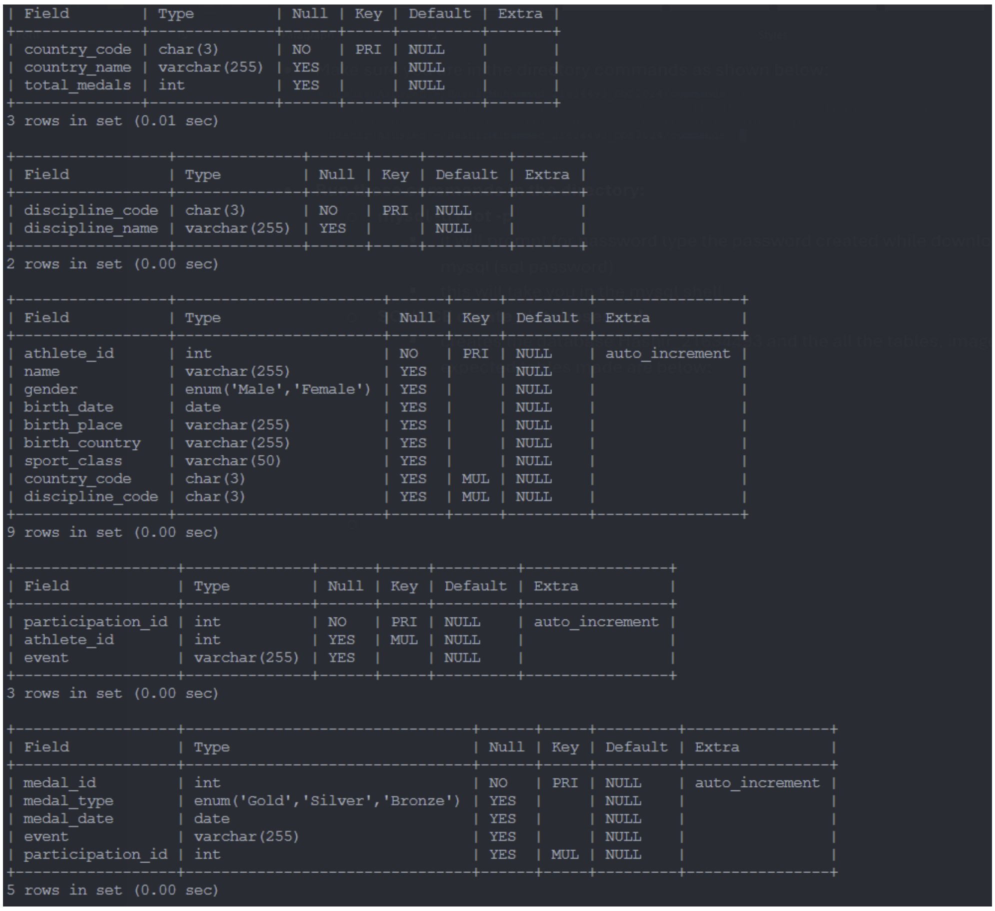
Execution

- Make sure you are in the directory commands as shown below:

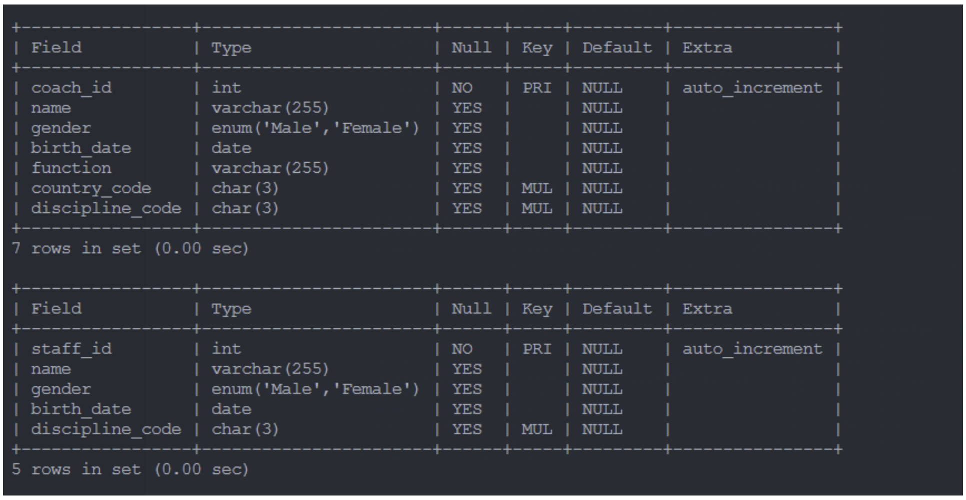


```
hashir@AsusZen:~/HashirMuhammad_21634493_DbS2024/commands$ ls
advanced_queries.sql  create_databases.sql  delete_all.sql          olympics_2020_queries.sql
connection.py         db_config.json        insert_sample_values.sql
hashir@AsusZen:~/HashirMuhammad_21634493_DbS2024/commands$ 
```

- **Run these commands in the directory:**
  - **mysql -u root -p**
    - it will prompt for a password, type the password created while downloading mysql (sql password)
    - this will take you in the mysql shell
  - **SOURCE create_databases.sql**
    - creates the database Hashir_21634493 and the all the tables, images expected tables made are below:



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

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

+-----------------+------------------------+------+-----+---------+----------------+
| Field           | Type                   | Null | Key | Default | Extra          |
+-----------------+------------------------+------+-----+---------+----------------+
| athlete_id      | int                    | NO   | PRI | NULL    | auto_increment |
| name            | varchar(255)           | YES  |     | NULL    |                |
| gender          | enum('Male','Female')  | YES  |     | NULL    |                |
| birth_date      | date                   | YES  |     | NULL    |                |
| birth_place     | varchar(255)           | YES  |     | NULL    |                |
| birth_country   | varchar(255)           | YES  |     | NULL    |                |
| sport_class     | varchar(50)            | YES  |     | NULL    |                |
| country_code    | char(3)                | YES  | MUL | NULL    |                |
| discipline_code | char(3)                | YES  | MUL | NULL    |                |
+-----------------+------------------------+------+-----+---------+----------------+
9 rows in set (0.00 sec)

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

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



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

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

- **SOURCE insert_sample_values.sql**
  - this will enter the sample data into the created tables.
- **SOURCE olympics_2020_queries.sql**
  - this would run the queries made for level 1 and level 2 and outputs them on theterminal
- **SOURCE advanced_queries.sql**
  - would run the queries and create advanced procedures and triggers

TO USE THE ADVANCED FUNCTIONS :

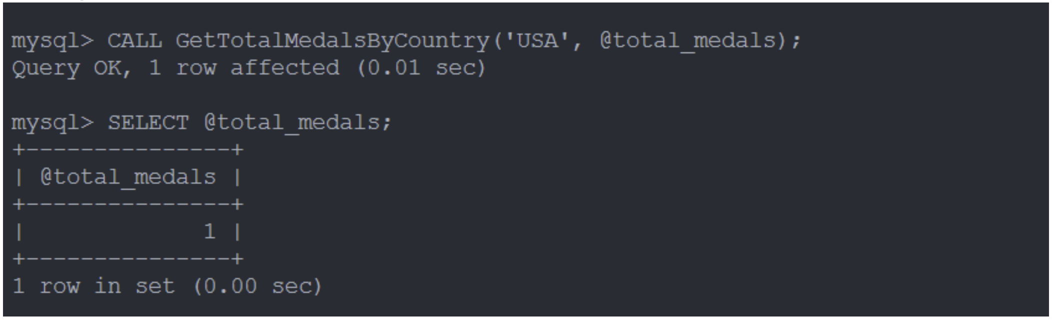
CALL AddAthlete('John Doe', 'Male', '1990-01-15', 'New York', 'USA', 'Class A', 'USA', 'ATH');

- this CALL allows us to ass more athlete's

CALL GetTotalMedalsByCountry(‘<country_code>’, @total_medals);

SELECT @total_medals;

- this code will show us the total_medals of the specific country_code given
- Example :

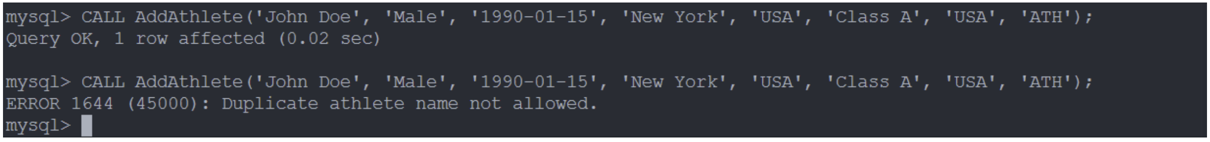


```
mysql> CALL GetTotalMedalsByCountry('USA', @total_medals);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @total_medals;
+---------------+
| @total_medals |
+---------------+
|             1 |
+---------------+
1 row in set (0.00 sec)
```

This one is a trigger not a function- ***TRIGGER: PreventDuplicateAthletes***

- If you try to enter an athlete which already exists an error will be thrown

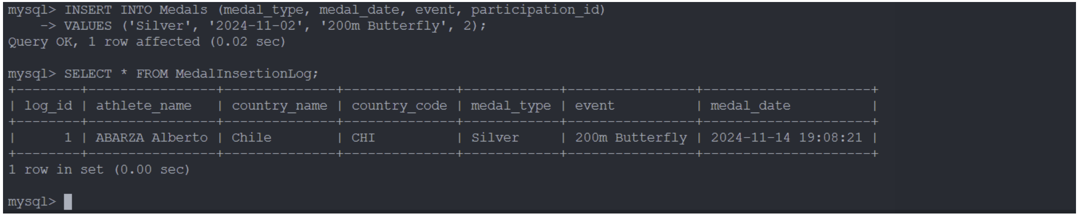
- Example:



```
mysql> CALL AddAthlete('John Doe', 'Male', '1990-01-15', 'New York', 'USA', 'Class A', 'USA', 'ATH');
Query OK, 1 row affected (0.02 sec)

mysql> CALL AddAthlete('John Doe', 'Male', '1990-01-15', 'New York', 'USA', 'Class A', 'USA', 'ATH');
ERROR 1644 (45000): Duplicate athlete name not allowed.
mysql> 
```

This one is a trigger not a function- *UpdateTotalMedals*

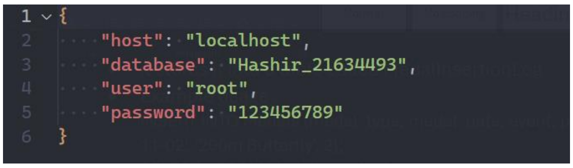
- when a new medal is added this trigger automates medal tracking and updates for better data management
- it stores it into a table called MedalInsertionLog
- Example usage :
  INSERT INTO Medals (medal_type, medal_date, event, participation_id) VALUES ('Silver', '2024-11-02', '200m Butterfly', 2);
  SELECT * FROM MedalInsertionLog;



```
mysql> INSERT INTO Medals (medal_type, medal_date, event, participation_id)
    -> VALUES ('Silver', '2024-11-02', '200m Butterfly', 2);
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM MedalInsertionLog;
+--------+----------------+--------------+--------------+------------+----------------+---------------------+
| log_id | athlete_name   | country_name | country_code | medal_type | event          | medal_date          |
+--------+----------------+--------------+--------------+------------+----------------+---------------------+
|      1 | ABARZA Alberto | Chile        | CHI          | Silver     | 200m Butterfly | 2024-11-14 19:08:21 |
+--------+----------------+--------------+--------------+------------+----------------+---------------------+
1 row in set (0.00 sec)

mysql> 
```

Python Integration:

- configure the db_config.json to creditentials

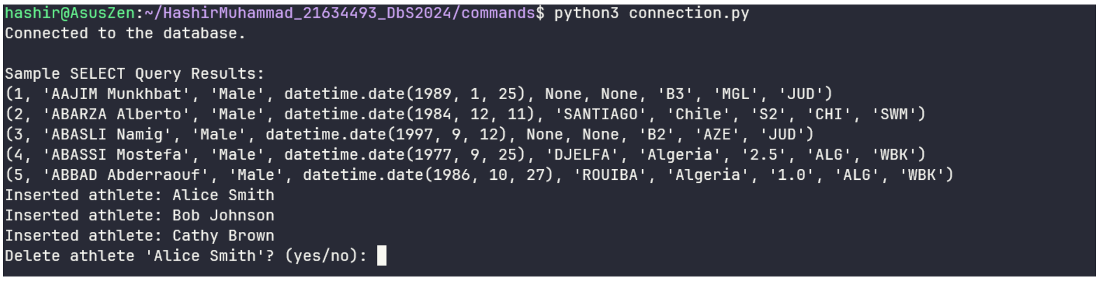


```
{
    "host": "localhost",
    "database": "Hashir_21634493",
    "user": "root",
    "password": "123456789"
}
```

- make sure mysql is running and enabled.

run python file using command:

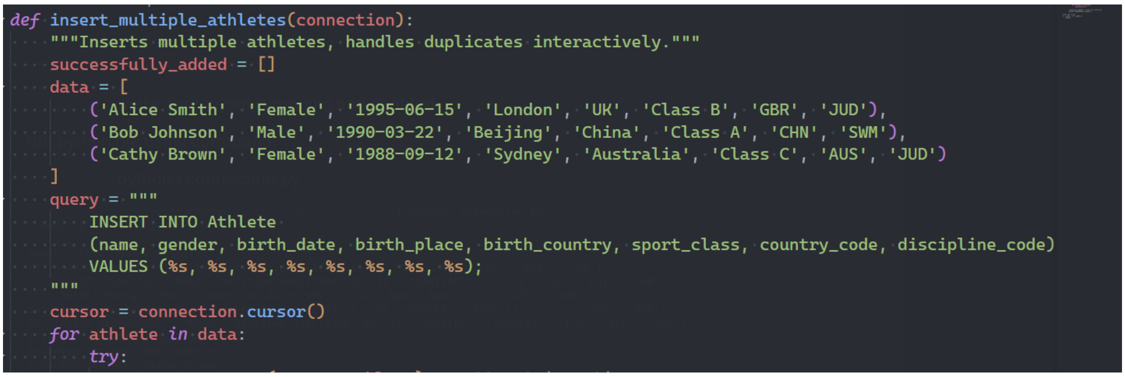
- python3 connection.py



```
hashir@AsusZen:~/HashirMuhammad_21634493_DbS2024/commands$ python3 connection.py
Connected to the database.

Sample SELECT Query Results:
(1, 'AAJIM Munkhbat', 'Male', datetime.date(1989, 1, 25), None, None, 'B3', 'MGL', 'JUD')
(2, 'ABARZA Alberto', 'Male', datetime.date(1984, 12, 11), 'SANTIAGO', 'Chile', 'S2', 'CHI', 'SWM')
(3, 'ABASLI Namig', 'Male', datetime.date(1997, 9, 12), None, None, 'B2', 'AZE', 'JUD')
(4, 'ABASSI Mostefa', 'Male', datetime.date(1977, 9, 25), 'DJELFA', 'Algeria', '2.5', 'ALG', 'WBK')
(5, 'ABBAD Abderraouf', 'Male', datetime.date(1986, 10, 27), 'ROUIBA', 'Algeria', '1.0', 'ALG', 'WBK')
Inserted athlete: Alice Smith
Inserted athlete: Bob Johnson
Inserted athlete: Cathy Brown
Delete athlete 'Alice Smith'? (yes/no): 
```

- Add's the data in the code and then asks if you want to keep it or delete it
- To add more data refer to the code function insert_multiple_athletes(connection)

Muhammad Hashir
21634493



```python
def insert_multiple_athletes(connection):
    """Inserts multiple athletes, handles duplicates interactively."""
    successfully_added = []
    data = [
        ('Alice Smith', 'Female', '1995-06-15', 'London', 'UK', 'Class B', 'GBR', 'JUD'),
        ('Bob Johnson', 'Male', '1990-03-22', 'Beijing', 'China', 'Class A', 'CHN', 'SWM'),
        ('Cathy Brown', 'Female', '1988-09-12', 'Sydney', 'Australia', 'Class C', 'AUS', 'JUD')
    ]
    query = """
        INSERT INTO Athlete
        (name, gender, birth_date, birth_place, birth_country, sport_class, country_code, discipline_code)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s);
    """
    cursor = connection.cursor()
    for athlete in data:
        try:
```