**Task-I and Task-III Unit Testing**

**Task-1**

Build and populate necessary tables (30% of course project grade)

• Create the DB table to store the dataset and populate it with data.

• Identify constraints as needed and document them in your Readme.md file.

• Your tables should be created in schema with the name “fifa”.

• In your ReadMe.md, attach a screenshot of your table infrastructure (e.g. DbVisualizer screenshot)

Solution-1:

**Filename: taskOne.py**

The following functions are created along with the structure of the code functionality arrangements for this task-1. i.e.

1. Connect(),

2.create\_schema(),

3.create\_table().

A screenshot of a computer

Description automatically generated with medium confidence

Step-1: \_\_main\_\_ function in this taskOne.py file calls the connect() function.

Description: This function initiates the connection string with the system parameters that existed in my laptop. It’s a “try-except” statement. In the try statement, I called two defined functions i.e. **create\_schema() and create\_table()** independently to create the new schema and new table respectively.

If the try statement fails, the except statement executes to raise the error to let user know that there is an issue with the connection string/while connecting to the database. The following below snippet shows the code for the **“connect()”** function

Text

Description automatically generated

Step-2:

**“connect()” function calls the “create\_schema()”.** Create schema function that tells the user about the database version.It creates the scema with the name **“fifa”** and finally the connection is committed. The below is the snippet of the code.

Text

Description automatically generated

Step-3:

**“connect()”** function calls the **“create\_table()”.** create a table function with the name “**player\_twenty”**. The following code is the snippet of the code for the createtable.

Text

Description automatically generated

**Filename: taskOneUnitTest.py**

The unit test file consists of following functions arranged as follows in the snippet provided below.

A screenshot of a computer

Description automatically generated with medium confidence

It consists of

1.test\_happypath\_dbConnection()

2.test\_sadPath()

Step-1:

test\_happypath\_dbConnection() function consists of the testing the schema and testing the table creation by passing the database parameters.

Text

Description automatically generated

Step-2:

test\_sadPath() function consists of the testing the connect function and testing the overall functionality.I passed the wrong parameters in the connect function to see the exception statement.



**Outcome:**

Graphical user interface, text

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Task-III:**

Conduct analytics on your dataset (40% of course project grade) Develop Python functions to answer the following questions (given that x, y and z) will be user-entered parameters. Each question should be answered by one or more Python functions. Think about how to implement these requirements with the least possible SQL statements/transactions.

1. List the x players who achieved highest improvement across all skillsets.

2. What are the y clubs that have largest number of players with contracts ending in 2021?

3. List the z clubs with largest number of players in the dataset where z >= 5.

4. What is the most popular nation\_position and team\_position in the dataset? (list the most popular for each)

5. What is the most popular nationality for the players in the dataset?

Solution-3.1

**Filename: taskTwoOne.py**

2. 1. List the x players who achieved highest improvement across all skillsets.

The following is code arrangement for the task2.1. I defined 3 dependent functions. “top\_improvements()” is the main function, when it is been called. It performs the desired task to get the top performer list in each individual skillset.

Text

Description automatically generated

The top\_improvements() functions consist of try-except statement to test the database connection if not then raise an error. Here is the code snippet for the same. This function call the skillset() function to get the user input to find how many performers are needed in the each skillset. The second snippet below shows the how the **skillset()** function is being implemented. This function also tests the valid user input, if not **return None** to the main function

Text

Description automatically generated

Text

Description automatically generated

**Filename: taskTwoOneUnitTest.py**

The UnitTest file consists of two functions i.e. test\_happyPath() and test\_sadPath() functions. Each functions tests for the expected and unexpected outcome from the main file.

The following code snippet provides the detailed information for the code implementation of unit testing for this task 2.1.

Text

Description automatically generated

**UniTest Outcome snippets for task 3.1:**

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Solution-3.2

**Filename: taskTwoTwoMain.py**

2. What are the y clubs that have largest number of players with contracts ending in 2021?

“Largest\_number\_of\_players\_with\_contracts\_ends\_in\_twentyone()” is the main function, when it is been called. It performs the desired task.The following snippet shows the arrangement of the functions

A screenshot of a computer

Description automatically generated with medium confidence

The largest\_number\_of\_players\_with\_contracts\_ends\_in\_twentyone() function consist of try-except statement to test the database connection if not then raise an error. Here is the code snippet for the same. This function call the “Largest\_number\_of\_players\_with\_contracts\_ends\_in\_twentyone()” function to get the user input to find how many performers are needed.

Text

Description automatically generated

**Filename: taskTwoTwoUnitTest.py**

The UnitTest file consists of two functions i.e. test\_happyPath() and test\_sadPath() functions. Each functions tests for the expected and unexpected outcome from the main file.

The following code snippet provides the detailed information for the code implementation of unit testing for this task 3.2.

Text

Description automatically generated

**UniTest Outcome snippets for task 3.2:**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Solution-3.3

**Filename: taskTwoThreeMainFile.py**

3. List the z clubs with largest number of players in the dataset where z >= 5.

“clubs\_with\_largest\_number\_of\_players()”is the main function, when it is been called. It performs the desired task.The following snippet shows the arrangement of the functions This functions internally calls the “UserInput()” function to get the “z” value which is eventually used in the query to get the number of players.

Text

Description automatically generated

The main function “clubs\_with\_largest\_number\_of\_players()” consists of try-except function which basically checks for the database connection before running the query, once it successfully connects the query is generated with the help of the user input given from the key board. If the database connection is unsuccessful, then in that case except statement gets executed. And further finally function in the main function closes the db connection.

Text

Description automatically generated

**Filename: taskTwoThreeUnitTest.py**

The UnitTest file consists of two functions i.e. test\_happyPath() and test\_sadPath() functions. Each functions tests for the expected and unexpected outcome from the main file.

The following code snippet provides the detailed information for the code implementation of unit testing for this task 3.3.

Text

Description automatically generated

The following are the output of the html files for both the python files developed i.e. main file and the test file.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Solution-3.4

**Filename: taskTwoFourMainFile.py**

4. What is the most popular nation\_position and team\_position in the dataset? (list the most popular for each)

“most\_popular\_nation\_and\_team\_position()”is the main function, when it is been called. It performs the desired task.The following snippet shows the arrangement of the functions This functions internally calls the “UserInput()” function to get the value which is eventually used in the query to get the number of popular nation and the team position.

Text

Description automatically generated

The main function “most\_popular\_nation\_and\_team\_position()” consists of try-except function which basically checks for the database connection before running the query, once it successfully connects the query is generated with the help of the user input given from the key board. If the database connection is unsuccessful, then in that case except statement gets executed. And further finally function in the main function closes the db connection.

Text

Description automatically generated

**Filename: taskTwoFourUnitTest.py**

The UnitTest file consists of two functions i.e. test\_happyPath() and test\_sadPath() functions. Each functions tests for the expected and unexpected outcome from the main file.

The following code snippet provides the detailed information for the code implementation of unit testing for this task 3.4.

Text

Description automatically generated

The following are the output of the html files for both the python files developed i.e. main file and the test file.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Solution-3.5

**Filename: taskTwoFiveMainFile.py**

5. What is the most popular nationality for the players in the dataset?

“most\_popular\_nationality()”is the main function, when it is been called. It performs the desired task.The following snippet shows the arrangement of the functions This functions internally calls the “UserInput()” function to get the value which is eventually used in the query to get the number of popular nation and the team position.

A screenshot of a computer

Description automatically generated

The main function “most\_popular\_nationality ()” consists of try-except function which basically checks for the database connection before running the query, once it successfully connects the query is generated with the help of the user input given from the key board. If the database connection is unsuccessful, then in that case except statement gets executed. And further finally function in the main function closes the db connection.

Text

Description automatically generated

**Filename: taskTwoFiveTest.py**

The UnitTest file consists of two functions i.e. test\_happyPath() and test\_sadPath() functions. Each functions tests for the expected and unexpected outcome from the main file.

The following code snippet provides the detailed information for the code implementation of unit testing for this task 3.5.

Text

Description automatically generated

The following are the output of the html files for both the python files developed i.e. main file and the test file.

Graphical user interface, application

Description automatically generated