CS 710 - Assignment 1

Dr. Alireza Manashty

September 28, 2021

In this assignment, you are going to practice learning Python programming by creating a simple DBMS in Python. The goal is for you to understand how Python, as a programming language, can be used to provide an interface for a higher-level query language like SQL. You will learn about string manipulation techniques and creating classes in Python.

For this assignment, **(1)**submit a .ipynb file and **(2)**an exported .html file, along with **(3)**required files, that includes all the sections below, clearly identified and **(4)** documented using Markdown cells, written in a Jupyter notebook. Everything should be rigorously tested and the test outputs should be present in the file when creating the output or saving the file. Make sure you identify each heading with Heading 1 in markdown (e.g., '# Assignment 1.1, 1.2, etc.').

1. Implement the following function in Python which receives a simple SQL statement **str** as string and returns the name of the table after the **from** keyword.

   def findTable(str)

For example, *findTable*("select * from customers") will return "customers"

2. Implement the following function which returns the attributes of a select statement which are located between **select** and **from** statement.

   def getAttributes(str)

For example,
getAttributes("select * from customers") will return "*" and
getAttributes("select name, customerID from customers") will return the following list of attributes: ["name", "customerID"]

3. Define a class in Python named **Attribute** with two string fields: **name** and **type** (default to "string").

4. Define a class in Python named **Table** which contains a list of "Attributes" named **attributes** and a list of lists called **rows**. Table also contains a **name** attribute.

5. Create a class named "Database" which can contain a list of Tables called **tables**.

6. Implement **Database.read(tablename, filename)** that reads a CSV file provided in the <filename> which creates a table named <tablename> and adds it to its **tables** list. Assume that CSV file columns contain only "string" data types and has a header row containing the attribute heads.

   For example:

   db = Database()
   db.read('People',"file.csv")

   The file.csv contents could look like the following:

   ID,Name,Age
   1,John,20
   2,Mary,19
   3,Alex,20

7. Implement **Database.select(query, count)** that receives a simple **select** statement as <query> and returns the results as a formatted table up to <count> rows. Use *findTable(str)* and *getAttributes(str)* which you implemented earlier.

   For example:

   db = Database()
   db.read('People',"file.csv")
   db.select('select * from people')

ID	Name	Age
1	John	20
2	Mary	19
3	Alex	20

   db.select('select name from people')

Name
John
Mary
Alex

Notes:

- For this assignment, consider inputs, select statements, and the formatting of the CSV files to be simple and similar to examples provided.

- You should implement most of the functionality as member functions inside of each class. Make sure your classes have necessary member functions, attributes, and constructors to accomplish these tasks.
- You are not allowed to use any third-party libraries or tools (such as NumPy, pandas, etc.), unless absolutely necessary. For example, the CSV file should be read line by line using basic Python, and not using pandas.
- You will require string manipulation functions. Learn about them and apply them as necessary. Stackoverflow and other online references could help you achieve basic tasks if you found yourself struggling.
- Do it yourself
- Good luck!