Lab Report

CSE - 4308

Database Management Systems Lab

Name : Mukit Mahdin
ID : 200042170
Department : CSE
Program: SWE
Group: B

Task: 1:

Find the average of the Student's GPA from the grades.txt

Analysis :

There are 3 columns defining Student ID, GPA, and Semester separated by semicolons. We need to take the GPA from each tuple and find the average of every value representing the GPA.

Solution :

```
with open('F:\CodeBuddy\Database-Lab\Lab-1\grades.txt') as f:

                contents = f.read()
                new_list = contents.split("\n")
                sum = 0
                for i in range(0,len(new_list)-1):
                        temp_list = new_list[i].split(";")
                        sum = sum + float(temp_list[1])

                f.close()
                print(sum/len(new_list))
```

Explanation :

I solved the first problem using Python.
In line - 1, I started the task by accessing the file from its respective directory as f. Then read the file with f.read() and stored it in contents and slightly modified it in the new_list.
Since we will need the sum to get the average so I initialized the sum as 0. To access the values separated by semicolons I ran a for loop and stored the accessed values in temp_list. In the temp_list the GPA is situated in the 2nd column which means temp_list[1].
After traversing all GPA values, I closed the file with f.close(). Finally, we are getting the average GPA by dividing the sum by the length of the list.
In this particular problem, I didn't face any difficulties.

Task: 2 :

Take Student ID, GPA, and Semester as input. Then after validating the input, insert the information as a new row in the grades.txt file. If the information is invalid, discard the input and show an error message.

Analysis :

As this is basically an appending task, so my plan was to take input from the user and merge that into a semicolon-separated string and add that string with a preceding new line in the existing grades.txt file.

Solution :

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Append_Values
{
    internal class Program
    {
        public class Student
        {
            public string StudentID;
            public string GPA;
            public string Sem;

            public Student(string StudentID, string GPA, string Sem)
            {
                this.StudentID = StudentID;
                this.GPA = GPA;
                this.Sem = Sem;
            }
        }
```

```
  static void Main(string[] args){
      List<Student> StudentList = new List<Student>();
      string StudentID = Console.ReadLine();
      string GPA = Console.ReadLine();
      string Semester = Console.ReadLine();

      double dGPA = Convert.ToDouble(GPA);
      int dSemester = Convert.ToInt32(Semester);

      if ((dGPA > 4.00 && dGPA < 0.00) && (dSemester > 8 &&
dSemester < 1)){
            Console.WriteLine("The Given input is invalid");
      }

string newFileName = @"F:\CodeBuddy\Database-Lab\Lab-1\grades.txt";
string StudentDetails = "\n" + StudentID + ";" + GPA + ";" +
Semester;

if (!File.Exists(newFileName)){
      File.WriteAllText(newFileName,StudentDetails);
}

File.AppendAllText(newFileName, StudentDetails);
Console.WriteLine("Information is appended to the grades.txt");

         }
     }
}
```

Explanation :

I solved this particular problem using C#. Because it seems comparatively easier to me to work with file read and write in C#.

Here I’m using System.IO to read and write files. I also used a bit of OOP to make things easier. So, there is a Student Class and constructor containing the attributes StudentID, GPA, and Semester.

In the Main() method there is a List called StudentList which allows the Student type to store. So I’m taking Student ID, GPA, and Semester as input one after another as strings.

Then I converted the GPA and Semester to float and int respectively to check validity further.
If the (GPA > 4.00 and GPA < 0.00) and (Semester > 8 and Semester < 1) then the Given input will be considered invalid at the new value won't be appended to the grades.txt file.
We read the file in the variable named "newFileName" to add the newly generated input.
After getting input from the user I merged all inputs in "StudentDetails" as "\n" + StudentID + ";" + GPA + ";" + Semester;

Now the details are semicolon-separated values with a preceding new line(\n) like the other tuples in grades.txt.

After checking the file's existence, the newly generated string will be appended to the grades.txt.

Task : 3:

Take Student ID as input and show his/her name and the semester in which he/she got the lowest GPA. Print an error message if the Student ID does not exist in your database.

Analysis :

In this particular task, we need to merge two text files named "grades.txt" and "studentInfo.txt". The new combined will contain all the unique columns of both text files.
From the newly generated combined file, we have to perform our query. In the query, our key is the Student ID and we'll have to return the lowest GPA and Semester of the respective ID. Also, we have to show an error message if there is no such Student ID exists in the file.

Solution :

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lowest_GPA
{
    internal class Program
    {
        public class Student
        {
            public string StudentID;
            public string GPA;
            public string Sem;

            public Student(string StudentID, string GPA, string Sem)
            {
                this.StudentID = StudentID;
                this.GPA = GPA;
                this.Sem = Sem;
            }
            public string getStudentID()
            {
                return this.StudentID;
            }
        }

static void Main(string[] args){

    List<Student> StudentList = new List<Student>();
    using (var reader =
    new StreamReader(@"F:\CodeBuddy\Database-Lab\Lab-1\grades.txt"))
    {
       while (!reader.EndOfStream){

            var line = reader.ReadLine();
            var values = line.Split(';');
Student dummy_student = new Student(values[0], values[1],
values[2]);
                   StudentList.Add(dummy_student);
                }
            }

            string CheckID = Console.ReadLine();
            double GPA;
            double CGPA = 4.0;
            string Sem;


            foreach (Student temp in StudentList)
            {
                if (temp.getStudentID() == CheckID)
                {
                    GPA = Convert.ToDouble(temp.GPA);

                    if (GPA < CGPA)
                    {
                        CGPA = GPA;
                        Sem = temp.Sem;
                    }
                    Console.WriteLine("Lowest GPA :" + CGPA);
                    Console.WriteLine("Semester :"+temp.Sem);
                }
            }
        }
    }
}
```

Explanation :

I perform this particular problem using C#. Here I used OOP also like the previous one to make it easier. Also, I maintained a List<Student> here.
I used StreamReader to read the text file till the End of the Stream. And values separated by semicolons are handled by line.Split(';').

The 3 values of each tuple is denoted by values[0], values[1] and values [2] respectively. Which also maintains the sequence - StudentID, GPA, Semester. After reading every value by dummy_student I added them to the List<Student> StudentList.

Then took a variable named "CheckID" to take the input of the query. With this key, I traversed the whole StudentList to find the match of the Query ID.
When it found the ID it started to find the lowest GPA of that specific ID gradually. After getting the lowest the method will return the lowest GPA along with the particular semester.

The main problem I faced in this task during the lab is to merge two files properly. I got the wrong merge which was inconsistent so that's why I finished the task with just grades.txt. So returning the Name wasn't possible. Other than that every functionality is performing well.