CPSC1012 – Database Read Demo (Optional)

Introduction

This demo is a quick demo on how to read data into a `List<T>` from a database. The techniques used for the `List<T>` will not be covered, only the means to connect to the database. You will see some similarities between reading from a file and reading from a database. (This demo will not cover how to create a database; that topic is taught in DMIT1508).

For this demo you will need:

1. SQL Express and SQL Server Management Studio
2. The database script, `StudentData.sql`.

Programming Code

The first step is to create a Visual Studio solution called `DatabaseDemoRead`.

Student.cs

In this solution you will need to create a class called `Student` with the following code:

```
namespace DatabaseReadDemo
{
    internal class Student
    {
        private int _studentID;
        private string _lastName;
        private string _firstName;
        private int _grade;

        public int StudentID
        {
            get { return _studentID; }
            set { _studentID = value; }
        }
        public string LastName
        {
            get { return _lastName; }
            set { _lastName = value; }
        }
        public string FirstName
        {
            get { return _firstName; }
            set { _firstName = value; }
        }
        public int Grade
        {
            get { return _grade; }
            set
            {
                if (value >= 0 && value <= 100)
                {
                    _grade = value;
                }
                else
                {
                    throw new Exception("Invalid Grade");
                }
            }
        }
```

```
        public Student(int studentID, string lastName, string firstName, int grade)
        {
            StudentID = studentID;
            LastName = lastName;
            FirstName = firstName;
            Grade = grade;
        }

        public override string ToString()
        {
            return string.Format($"{StudentID,-14}{LastName,-22}{FirstName,-22}{Grade,3}");
        }
    }//eoc
}//eon
```

Program.cs

You will need the following using statement for the database access to work:

```
using System.Data.SqlClient; //needed for database access
```

For ease of passing data to methods, create the following class constants:

```
const string InitialCatalog = "StudentData";
const string connectionString = @"Data Source=.\SQLEXPRESS;Initial Catalog=" + InitialCatalog + ";Integrated Security=True";
```

This class will have the following methods:

```
static void ReadFromDatabase(List<Student> students)
{
    SqlConnection connection = new SqlConnection(connectionString);
    try
    {
        connection.Open();
        Console.WriteLine($"Connection to {InitialCatalog} was successful!");
        string query = @"SELECT StudentID,LastName,FirstName,Grade FROM Student ORDER BY LastName,Firstname";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            //Console.WriteLine($"SQL query: {command.CommandText}");
            SqlDataReader reader = command.ExecuteReader();
            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    Student student = new Student(reader.GetInt32(0),
                                                  reader.GetString(1),
                                                  reader.GetString(2),
                                                  reader.GetInt32(3));
                    students.Add(student);
                }
                Console.WriteLine("Database read completed");
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        connection.Close();
        Console.WriteLine("Database connection closed");
    }//end try-catch-finally
}//end of ReadFromDatabase
```

```csharp
static double CalculateClassAverage(List<Student> students)
{
    double average = 0;
    foreach(Student student in students)
    {
        average += student.Grade;
    }
    return average /= students.Count;
}//end of CalculateClassAverage

static void DisplayStudents(List<Student> students)
{
    Console.WriteLine("\nStudents:");
    Console.WriteLine("{0,-14}{1,-22}{2,-22}{3,3}", "ID", "Lastname", "Firstname", "Grade");
    foreach (Student student in students)
    {
        Console.WriteLine(student);
    }
    Console.WriteLine("\n{0,57}{1,3:0.0}","Class Average: ", CalculateClassAverage(students));
}//end of DisplayStudents

static void Setup()
{
    Console.Title = "Database Demo";
    Console.ForegroundColor = ConsoleColor.Black;
    Console.BackgroundColor = ConsoleColor.White;
    Console.Clear();
}//end of Setup
```

The `Main()` method will have the following code:

```csharp
static void Main(string[] args)
{
    Setup();

    List<Student> students = new List<Student>();
    ReadFromDatabase(students);
    DisplayStudents(students);

    Console.ReadLine();
}//eom
```

StudentData.sql

The code below is needed to create the database:

```sql
--*****************************************************************************
--*************             StudentData Database                   ***********
--*************             Artist: Allan Anderson                 ***********
--*************       Date Created: December 18, 2020              ***********
--*************       Date Modified: June 10, 2022                 ***********
--*****************************************************************************


--*****************************************************************************
--*************       StudentData  CREATE DATABASE statements      ***********
--*****************************************************************************
USE Model
GO
/*
DROP DATABASE StudentData
GO
*/
CREATE DATABASE StudentData
GO
USE StudentData
GO


--*****************************************************************************
--*************        StudentData  CREATE TABLE statements       ***********
```

```sql
--***************************************************************************
CREATE TABLE Student
(
        StudentID INT NOT NULL PRIMARY KEY,
        LastName VARCHAR(20) NOT NULL,
        FirstName VARCHAR(20) NOT NULL,
        Grade INT NOT NULL
)
GO
--***************************************************************************
--************        StudentData  INSERT INTO Student        ************
--***************************************************************************
INSERT INTO Student (StudentID,LastName,FirstName,Grade)
VALUES
        (10004523,'Barns','Russ',89),
        (10003900,'Burton','Kelly',74),
        (10009510,'Collier','Hugh',89),
        (10006345,'Dent','Adam',65),
        (10005904,'Dunstill','Amy',58),
        (10000607,'Frampton','Brian',70),
        (10001424,'Hart','Bary',80),
        (10006579,'Jackson','David',73),
        (10007416,'Mandel','Dale',78),
        (10009911,'Merick','Roger',64),
        (10001187,'Nolan','Tim',63),
        (10001322,'Olczyk','Edward',36),
        (10000320,'Plant','Sammy',69),
        (10007858,'Richter','Mike',61),
        (10005787,'Samson','Lorne',80),
        (10009587,'Smythe','Michael',89),
        (10001707,'Southman','Bridgette',70),
        (10003209,'Sundre','John',96),
        (10008024,'Timmins','Heather',74),
        (10001252,'Trevor','Adrian',58),
        (10005260,'Vale','Rick',29),
        (10006148,'Winn','Doug',68),
        (10001586,'Young','Ryan',62)
GO
```

Program Output

Once you have everything typed in, and the database created, you can execute your code. The result should look like:

```
Database Demo                                                    🗗    —    □    ✕

Connection to StudentData was successful!
Database read completed
Database connection closed

Students:
ID              Lastname            Firstname           Grade
10004523        Barns               Russ                89
10003900        Burton              Kelly               74
10009510        Collier             Hugh                89
10006345        Dent                Adam                65
10005904        Dunstill            Amy                 58
10000607        Frampton            Brian               70
10001424        Hart                Bary                80
10006579        Jackson             David               73
10007416        Mandel              Dale                78
10009911        Merick              Roger               64
10001187        Nolan               Tim                 63
10001322        Olczyk              Edward              36
10000320        Plant               Sammy               69
10007858        Richter             Mike                61
10005787        Samson              Lorne               80
10009587        Smythe              Michael             89
10001707        Southman            Bridgette           70
10003209        Sundre              John                96
10008024        Timmins             Heather             74
10001252        Trevor              Adrian              58
10005260        Vale                Rick                29
10006148        Winn                Doug                68
10001586        Young               Ryan                62

                        Class Average: 69.3
```