

Multi-tier Applications Development
Project: 30%
Winter 2023 - group: 07172

Student Identification:

Name : _____

Student number : _____

Date: _____

Attention: At the end, you must create a folder containing your SQL script, the Visual Studio solution folder and your Word / PDF file, zip it up and upload it to LEA (Omnivox). You must use Visual Studio Community 2017, 2019 or 2022.

Attention: Compliance with all instructions is essential. If you deviate from the instructions, your grade will be penalized.

Problem 1 (20 points)

Create a script named Project.sql for SQL Server to create a database named College1en containing four tables described below:

table "Programs" with a column "ProgId" VARCHAR (5) NOT NULL and
a column "ProgName" VARCHAR (50) NOT NULL;

table "Courses" with a column "CId" VARCHAR (7) NOT NULL,
a column "CName" VARCHAR (50) NOT NULL and
a column "ProgId" VARCHAR (5) NOT NULL;

table "Students" with a column "StId" VARCHAR (10) NOT NULL,
a column "StName" VARCHAR (50) NOT NULL and
a column "ProgId" VARCHAR (5) NOT NULL,

table "Enrollments" with a column "StId" VARCHAR (10) NOT NULL,
a column "CId" VARCHAR (7) NOT NULL and
a column "FinalGrade" INT;

Primary keys:

for table "Programs" the primary key is "ProgId";

for table "Courses" the primary key is "CId";

for table "Students" the primary key is "StId";

for table "Enrollments" the primary key is made of two columns: "StId" and "CId".

Foreign Keys:

In table "Courses", "ProgId" is a foreign key referring to "ProgId" in the table "Programs".

In table "Students", "ProgId" is a foreign key referring to "ProgId" in the table "Programs".

In table "Enrollments", "StId" is a foreign key referring to "StId" in the table "Students" and "CId" is a foreign key referring to "CId" in the table "Courses".

You must determine by yourself the "ON DELETE" and "ON UPDATE" conditions for each foreign key, knowing that:

1. If a program is delete, all courses in that program must be deleted.
2. Updating ProgId in the programs table must propagate to the courses table.
3. We cannot delete a program in which there are students.
4. Updating the ProgId in the programs table must propagate to the students table.
5. If a student is deleted, all enrollments of that student must be deleted.
6. Updating the StId in the students table must propagate to the enrollments table.
7. We cannot delete a course to which there are enrolled students.
8. Updating the CId in the courses table must propagate to the enrollments table (if necessary, you may alternatively assume that the CId of a course cannot be updated if there are enrollments to that course).

Create some data to populate the tables. The primary key for Programs is always the letter "P" (uppercase) followed by four digits. The primary key for Courses is always the letter "C" (uppercase) followed by six digits. The primary key for Students is always the letter "S" (uppercase) followed by nine digits.

Problem 2 (80 points)

Create a **3-tiers C#, Windows Form, ADO.NET application** with **four** options at the main menu of the window for, using **one** DataGridView:

1. Option Students: the DataGridView adapted to show the "Students" table, which allows you to see the data in the table, add new rows, modify rows and delete them (deleting several rows). Use SQLAdaper.
2. Option Enrollments: the DataGridView adapted to show enrollments data: StId, StName, CId, CName, FinalGrade, ProgId, ProgName from the tables "Enrollments", "Students" and "Courses". In addition to showing the data, this option must have the sub-options: add, modify, delete and "manage Final Grade". To add or modify an enrollment, we must use an auxiliary form. To add, the auxiliary form must offer a dropbox to choose the StId, a read-only textbox showing the StName, a dropbox to choose the CId, a read-only textbox showing the CName. To modify an enrollment, the StId and the StName must be fixed and we can only choose the CId and the CName will show in read-only textbox. The delete sub-option must delete the selected enrollments

(lines). It must allow deleting multiple enrollments. To "manage Final Grade", another auxiliary form must be used, having four read-only textboxes to show StId, StName, CId and CName and a read-write textbox for the grade. Use SQLAdapter.

3. Option Courses: the DataGridView adapted to show the "Courses" table, which allows you to see the data in the table, add new rows, modify rows and delete them (deleting several rows). Use SQLAdapter.

4. Option Programs: the DataGridView adapted to show the "Programs" table, which allows you to see the data in the table, add new rows, modify rows and delete them (deleting several rows). Use SQLAdapter.

Note: To avoid having foreign key violation exceptions when using the application, the same foreign key must be defined between the DataTables. Therefore, the DataTables must be loaded in memory before using any of them for the first time.

Business Rules:

In addition to the rules listed as foreign keys delete and update rules, there are the following rules:

1. All courses must belong to one and only one program.
2. All students must be in one and only one program.
3. A student can enrol only to courses in the student's program.
4. Valid values of FinalGrade are null (no grade) or an integer between 0 and 100.
5. All enrollments are created with FinalGrade set to null.
6. FinalGrade can be removed (reset to null).
7. Once the FinalGrade is assigned, we cannot delete the enrollment and the only modification possible in the enrollment is to remove the final grade.
8. If the FinalGrade is not assigned, we cannot delete the enrollment and we can also change the course.

Create a Word or PDF file with the screen shots showing that your solution executes correctly.