CSE 385 PROJECT INSTRUCTION

Project Title: *Course Information*

Phuong Ho - Ziyang Song

Our project is about building relational data for the information on courses along with students, professors, etc. We aim at creating stored procedures and upload to a Web Server using the script provided. You can also load data through XML files by changing the code at the end of the .sql file uploaded.

1. **Steps to use the system:**
    a. Download and Open the Main Page

- In the ZIP file, we include:
    - project.sql
    - 5 text files:
        - courseAndStudent.txt
        - department.txt
        - course.txt
        - student.txt
        - professor.txt
    - 'project' folder
- To create the database, first download all the .txt files (5) and put them to C:\temp location, so the SQL file can load data in.
- Then, open project.sql and run on (localdb)\MSSQLLocalDB server. The database will be created. You are able to look at stored procedures or views and test.
- Next, you can open 'project.sln' in the project folder file and open API (api.asmx) to see the stored procedures listed.
- Open main.html in Chrome. You are now able to use the website to work with the data that you want. Due to the time limit of the project, we build sample pages of adding, updating, deleting records for the Student table and one to get Course information with a given studentId.
- You can also try using the XML process at the end of the SQL file.

b. Download and Open the Main Page:

- Open the main.html file. You can click on one of the links listed in the Action division.
- **View all the students' information:** click on the first box to see the current list of students
- **Get Course Information By Student ID**: click on the 2nd box and the following information of the student you want to retrieve:
    - Student ID: integer (should be a valid ID)
- **Add Student**: click 3rd box and provide the following information of the student you want to add:

    - First Name: a string with max length of 200
    - Last Name: a string with max length of 200
    - Credit He/She Earned: positive integer
    - Class (the year that the student is in): one of the following: freshman, sophomore, junior, senior
    - Honor Student or Not: boolean (Type True or False to get valid results)
    - Email: String with max length of 50
    - GPA: positive double
- **Update Student**: click 4th box and provide the following information of the student you want to update:
    - Student ID: integer (should be a valid ID)
    - First Name: a string with max length of 200
    - Last Name: a string with max length of 200
    - Credit He/She Earned: positive integer
    - Class (the year that the student is in): one of the following: freshman, sophomore, junior, senior
    - Honor Student or Not: boolean (Type True or False to get valid results)
    - Email: String with max length of 50
    - GPA: positive double
- **Delete Student**: click one the last box and provide the following information of the student you want to delete:

  (please make sure that you want to delete this student's record)

    - Student ID: integer (should be a valid ID and does not have any records in CourseStudent since this is used as a foreign key)

2. List each stored procedure (the parameters and what is expecting to return):

- **spRebuildReorgIndexList:** Rebuild and reorganize table indexes so the lookup time is shorter.
- **spGetStudentByCourseId**: Get Student ID and Fullname By CourseID:
    - Parameter: courseId
    - Return: studentId, FullName
- **spGetCourseByStudentId**: Get Course Information Taken By Student ID:
    - Parameter: studentId
    - Return: courseId, FullName
- **spGetHonorStuden**t: Get Honor Student List:
    - Return: studentId, FullName, GPA, class, email
- **spGetStudentUnderCertainGPA**: Get Student ID, Name, GPA, class Under a Certain GPA:
    - Return: studentId, FullName, GPA, class
- **spGetProfessorNoEmail**: Get a list of professors without emails:
    - Return: professorId, FullName

- **spGetCountByProfessorCours**e: Get Number of courses taught by given professor:
    - Parameter: professorId
    - Return:numberOfCourse
- **spGetNumberofStudentsByCourseId**: Get total number of students in a specific course
    - Return:numberOfStudents
- **spGetHonorStudentByCourseId:** Get the number of honor students in certain course
    - Parameter:courseId
    - Return:courseId, name, studentId, firstName, lastName
- **spGetProfessorNoPhoneByDepartment:** List all the professors in certain department who doesn't have a phone number
    - Parameter:department
    - Return: professorId, firstName, lastName
- **spGetStudentInNoCourse**: Get a list of students that are not registered in any classes
    - Return: studentId, FullName, GPA, email
- **spGetProfessorWithNoCourses**: Get a list of professors that are not teaching any classes
    - Return: professorId, FullName, email
- **spGetStudentByProfessorId**: Get a list of students that are taught by a specific professor.
    - Parameter: professorId
    - Return : studentId, FullName, email
- **spCourseNotFullCpacity:** Get a list of courses where capacity is not full
    - Return: courseId, name, remainedCapacity
- **spGetProfessorHavingDifferentLocation:** Get a list of professors with locations different from their departments' locations
    - Return: professorId, fullName,
- **spGetStudentsTakingMultiCourseSameProfessor:** Get a list of students taking multiple courses of the same professor
    - Parameter: studentId (default 0)
    - Return: studentId, professorId, Number of courses
- **spGetStudentsGpaGreaterSameDepartment:** Get a list of students whose GPA is greater than 3.0 and taking multiple courses of the same department
    - Parameter: studentId (default 0)
    - Return: studentId, gpa, Number of courses
- **spGetStudentScriptTakingMultipleCourses:** Return a List of Students who take multiple courses in multiple departments. Also, if their GPA is below 2.0, return "F", 2.0 ~2.5 return"D", 2.5~3.0 return "C",3.0~3.5 return"B"3.5~4.0 return"A".
    - Return: studentId, Script
- **spGetHighestHonoredStudentsDepartment:** Return the department has the largest number of honor students.

    - Return: departmentId, departmentName, NumOfHonorStudent
- **spGetProfessorCreditHours:** Return all the professors to see how many credit hours they have. If they have more than 6 credit hours course, return "too much course", if they have more than 4 and less than or equal to 6 credit hours, return "enough", if they have less than 4 credit hours, return "add some courses".
    - Return: professorId, Name, totalHours, Comments

3. **ER diagram of the database with an explanation of each entity and relationship for your reference:**

The relations between table is described below (the data is made up):