**实 验 报 告**

课程名称： web应用开发

实验名称： 基于数据库、控制台，完成学生选课系统

院 （系）： 信控学院

专业班级： 计算机2104

2024年 5月 28 日

# 实验目的

使用C#基础语法并且掌握使用.NET的ORM框架Entity Framework完成对线上数据库的连接操作，并编写一个控制台程序，实现学生管理系统的功能，要求对学生信息的任何操作都能体现在程序连接的远程数据库里。

# 二．实验基本要求和内容

在学生管理系统中要设计学生类（包括学号、姓名、班级、初始密码）、课程类（包括课程代码、课程名称、课程学分）、已选课程类（包括课程代码、学号）；设计数据库表；增删改查学生信息；学生选课、查看学生已选课程；增删改查课程信息等功能。

# 三．实验环境

Visio Studio 2022 .NET 8.0 datagrip Mircrosoft.EntityFrameworkCore

# 四．软件设计（包括软件功能设计和数据库设计）

(一)软件功能设计

在该学生信息管理系统中，可以看到主要需要对三种类型的信息进行管理，分别是：学生信息、学生选课信息、课程信息。所以在软件功能设计中，先在项目源代码路径下新建三个类，分别将不同功能代码写到不同类中，以便进行后期迭代和代码功能测试。

1. StudentManagement 类
   1. ShowStudentMenu()：显示学生管理系统二级菜单,将会读取用户输入信息,并调用下列对学生信息进行修改的方法。
   2. AddNewStudent()：添加学生的方法，将会提示用户输入学生所有信息，并进行信息检查，对于不合法的信息将输出错误信息。
   3. ShowAllStudents()：通过循环输出数据库中所有学生信息，格式化打印在控制台。
   4. SearchStudent()：通过用户输入的学号搜索学生信息，首先会对学号进行合法性检查，然后检索学生姓名以及班级并输出在控制台上。
   5. UpdateStudentInfo()：首先通过用户输入的学号搜索原学生的信息，并将其输出在控制台上，提示用户输入该学号学生的新信息，输入完成后将修改后的学生信息输出在控制台上。
   6. DelStudentInfo()：首先检查用户输入的学号是否合法，若是，则删除该学号对应的所有学生信息，并输出成功删除的提示信息。
2. StudentCourseManagement类
   1. ShowStudentCourseMenu()：控制台输出一个学生选课信息的二级菜单，将会读取用户输入信息并调用下列方法。
   2. SelectCourse()：给学生选课，该方法首先会检查用户输入的学号，然后检查用户输入的密码是否是学号对应学生的密码，然后检查用户输入的课程号是否合法，如果以上所有检查通过，将会在对应学生选课表中添加一行数据，表示给该学生选课成功。
   3. ShowSelectedCourses()：显示某个学生所有选课信息，输入学生学号以后，控制台输出该学号选择的所有课程，包括课程名称、课程号等信息，该查询需要三张表联合查询。
3. CourseManagement 类
   1. ShowCourseMenu()：显示课程信息管理的二级菜单，将会读取用户输入的信息并调用下列函数。
   2. AddNewCourse()：添加一个新的课程，需要用户在控制台输入课程所有信息，检查合法性以后将会在数据库的课程信息表中添加一行。
   3. ShowAllCourses()：显示所有课程信息，将信息输出到控制台上。
   4. SearchCourse()：查询课程信息，通过输入的课程号即课程信息表的主键，然后调用Find函数查找主键对应的所有信息，并将其输出到控制台上。
   5. UpdateCourseInfo()：更新课程信息，通过输入的课程号，先查询到具体的课程，然后用户输入该课程号对应的新的课程信息，并将其输出到控制台上。
   6. DelCourseInfo()：删除课程信息，通过用户输入的课程号查询对应的表项，然后将所有的信息进行删除，删除成功后将提示信息输出在控制台上。
4. 数据库设计
5. 学生信息表设计
6. StudentID 字段:varchar(20) 类型，主键，并且限制不为空，表示学生学号。
7. StudentName 字段：varchar（10）类型，表示学生姓名。
8. StudentClass 字段：varchar（10）类型，表示学生班级。
9. InitPassword 字段：varchar（10）类型，表示学生初始密码。
10. 学生选课信息表设计
11. CourseID 字段：varchar（20）类型，表示课程号。
12. StudentID 字段：varchar（20）类型，表示学号，两个字段一起作为学生选课信息表的主键。
13. 课程信息表设计
14. CourseID 字段：varchar（20）类型，主键并且限制不为空，表示课程号。
15. CourseaName 字段：varchar（10）类型，表示课程名称。
16. CourseCredit 字段：int 类型，表示课程的学分。

# 五．关键源代码与说明

//class CourseManagement

public class CourseManagement

{

public void ShowCourseMenu()

{

bool isContinue = true;

while (isContinue)

{

Console.Clear();

Console.WriteLine("1. 添加一门课程");

Console.WriteLine("2. 显示所有课程");

Console.WriteLine("3. 搜索课程");

Console.WriteLine("4. 更新课程信息");

Console.WriteLine("5. 删除课程信息");

Console.WriteLine("6. 返回上一级");

Console.Write("Enter your choice: ");

var choice = Console.ReadLine();

switch (choice)

{

case "1":

AddNewCourse();

break;

case "2":

ShowAllCourses();

break;

case "3":

SearchCourse();

break;

case "4":

UpdateCourseInfo();

break;

case "5":

DelCourseInfo();

break;

case "6":

isContinue = false;

break;

}

}

}

// 有关课程的信息

void AddNewCourse()

{

Console.WriteLine("请输入课程代码:");

var str = Console.ReadLine();

using (var context = new \_2114010313DbContext())

{

var checkCourse = context.Courses.Find(str);

if (!str.StartsWith('C'))

{

Console.WriteLine("课程号不合法,请重新输入");

printMessage();

return;

}

if (checkCourse != null && checkCourse.CourseName != "")

{

Console.WriteLine($"课程号为: {str} , 已经被 {checkCourse.CourseName}占用, 请重试");

printMessage();

return;

}

else

{

Console.WriteLine("请输入课程名称:");

var courseName = Console.ReadLine();

Console.WriteLine("请输入课程学分:");

int credit = int.Parse(Console.ReadLine());

var course = new Course();

course.CourseName = courseName;

course.CourseCredit = credit;

course.CourseId = str;

context.Courses.Add(course);

context.SaveChanges();

}

printMessage();

}

}

void ShowAllCourses()

{

using (var context = new \_2114010313DbContext())

{

var courses = context.Courses.ToList();

foreach (var course in courses)

{

Console.WriteLine($"课程号: {course.CourseId}, 课程名称: {course.CourseName},课程学分: {course.CourseCredit}");

}

printMessage();

}

}

void SearchCourse()

{

Console.WriteLine("请输入要查找的课程号:");

var str = Console.ReadLine();

using (var context = new \_2114010313DbContext())

{

var checkCourse = context.Courses.Find(str);

if (checkCourse == null)

{

Console.WriteLine($"不存在课程号为:{str}的课程,请重试");

printMessage();

return;

}

else

{

Console.WriteLine($"课程号为:{str} 的课程名称为:{checkCourse.CourseName} ,课程学分为:{checkCourse.CourseCredit}");

printMessage();

}

}

}

void UpdateCourseInfo()

{

Console.WriteLine("请输入想要修改的课程的课程号:");

var str = Console.ReadLine();

using (var context = new \_2114010313DbContext())

{

//tell StudentID is valid or not

var checkCourse = context.Courses.Find(str);

//判断逻辑

if (checkCourse == null)

{

Console.WriteLine($"不存在学号为:{str}的,请重试");

printMessage();

}

else

{

Console.WriteLine($"该课程的原课程名为:{checkCourse.CourseName},原课程学分为:{checkCourse.CourseCredit}");

Console.WriteLine("请输入新的课程姓名:");

var courseName = Console.ReadLine();

Console.WriteLine("请输入新的课程学分:");

int credit = int.Parse(Console.ReadLine());

checkCourse.CourseName = courseName;

checkCourse.CourseCredit = credit;

context.SaveChanges();

Console.WriteLine("课程信息修改成功,该课程的新信息如下:");

Console.WriteLine($"课程号为:{str} 的课程名修改为:{checkCourse.CourseName} ,课程学分修改为:{checkCourse.CourseCredit}");

printMessage();

}

}

}

void DelCourseInfo()

{

Console.WriteLine("请输入想要删除的课程号:");

var str = Console.ReadLine();

using (var context = new \_2114010313DbContext())

{

var checkCourse = context.Courses.Find(str);

//判断逻辑

if (checkCourse == null)

{

Console.WriteLine($"不存在课程号为：{str} 的课程,请重试");

printMessage();

}

else

{

context.Courses.Remove(checkCourse);

context.SaveChanges();

Console.WriteLine("该课程信息已经成功删除...");

printMessage();

}

}

}

//输出提示信息

void printMessage()

{

Console.WriteLine("successfully, press any key to continue...");

Console.ReadKey();

}

}

//class StudentManagement

public class StudentManagement

{

public void ShowStudentMenu()

{

bool isContinue = true;

while (isContinue)

{

Console.Clear();

Console.WriteLine("1. 新增一个学生");

Console.WriteLine("2. 显示所有学生信息");

Console.WriteLine("3. 搜索一个学生");

Console.WriteLine("4. 更新学生信息");

Console.WriteLine("5. 删除学生信息");

Console.WriteLine("6. 返回上级菜单");

Console.Write("Enter your choice: ");

var choice = Console.ReadLine();

switch (choice)

{

case "1":

AddNewStudent();

break;

case "2":

ShowAllStudents();

break;

case "3":

SearchStudent();

break;

case "4":

UpdateStudentInfo();

break;

case "5":

DelStudentInfo();

break;

case "6":

isContinue = false;

break;

default:

break;

}

}

}

public void AddNewStudent()

{

Console.WriteLine("请输入新添加的学生学号:");

var studentId = Console.ReadLine();

if(!studentId.StartsWith('S'))

{

Console.WriteLine("学号不合法,请重新输入");

printMessage();

return;

}

Console.WriteLine("请输入新添加的学生姓名:");

var studentName = Console.ReadLine();

Console.WriteLine("请输入学生的班级: ");

var studentClass = Console.ReadLine();

Console.WriteLine("输入学生密码:");

var studentPassword = Console.ReadLine();

using (var context = new \_2114010313DbContext())

{

var checkStu = context.Students.Find(studentId);

if (checkStu != null && checkStu.StudentName != "")

{

Console.WriteLine($"学号：{studentId},已经被 {checkStu.StudentName} 占用");

printMessage();

return;

}

var student = new Student();

student.StudentName = studentName;

student.StudentId = studentId;

student.StudentClass = studentClass;

student.InitPassword = studentPassword;

context.Students.Add(student);

context.SaveChanges();

}

printMessage();

}

public void ShowAllStudents()

{

using (var context = new \_2114010313DbContext())

{

var students = context.Students.ToList();

foreach (var student in students)

{

Console.WriteLine($"学号: {student.StudentId}, 学生姓名: {student.StudentName},学生班级: {student.StudentClass}");

}

}

printMessage();

}

public void SearchStudent()

{

Console.WriteLine("请输入学生学号:");

var str = Console.ReadLine();

if (!str.StartsWith('S'))

{

Console.WriteLine("学号不合法,请重新输入");

printMessage();

return;

}

using (var context = new \_2114010313DbContext())

{

//tell StudentID is valid or not

var checkStudent = context.Students.Find(str);

//判断逻辑

if (checkStudent == null)

{

Console.WriteLine($"不存在学号为:{str}的学生,请重试");

printMessage();

}

else

{

Console.WriteLine($"学号为:{str} 的学生姓名为:{checkStudent.StudentName} ,班级为:{checkStudent.StudentClass}");

printMessage();

}

}

}

public void UpdateStudentInfo()

{

Console.WriteLine("请输入想要修改的学生的学号:");

var str = Console.ReadLine();

if (!str.StartsWith('S'))

{

Console.WriteLine("学号不合法,请重新输入");

printMessage();

return;

}

using (var context = new \_2114010313DbContext())

{

//tell StudentID is valid or not

var checkStudent = context.Students.Find(str);

//判断逻辑

if (checkStudent == null)

{

Console.WriteLine($"不存在学号为:{str}的学生,请重试");

printMessage();

}

else

{

Console.WriteLine($"该学生的原信息为:姓名 {checkStudent.StudentName},班级{checkStudent.StudentClass}");

Console.WriteLine("请输入新的学生姓名:");

var studentName = Console.ReadLine();

Console.WriteLine("请输入新的学生班级:");

var studentClass = Console.ReadLine();

Console.WriteLine("请输入新的学生密码:");

var studentPassword = Console.ReadLine();

checkStudent.StudentName = studentName;

checkStudent.StudentClass = studentClass;

checkStudent.InitPassword = studentPassword;

context.SaveChanges();

Console.WriteLine("学生信息修改成功,该学生的新信息如下:");

Console.WriteLine($"学号为:{str} 的学生姓名修改为:{checkStudent.StudentName} ,班级修改为:{checkStudent.StudentClass}");

printMessage();

}

}

}

// 删除学生信息逻辑

public void DelStudentInfo()

{

Console.WriteLine("请输入想要删除的学生学号:");

var str = Console.ReadLine();

if (!str.StartsWith('S'))

{

Console.WriteLine("学号不合法,请重新输入");

printMessage();

return;

}

using (var context = new \_2114010313DbContext())

{

//tell StudentID is valid or not

var checkStudent = context.Students.Find(str);

//判断逻辑

if (checkStudent == null)

{

Console.WriteLine($"不存在学号为:{str}的学生,请重试");

printMessage();

}

else

{

context.Students.Remove(checkStudent);

context.SaveChanges();

Console.WriteLine("该名学生的信息已经成功删除...");

printMessage();

}

}

}

public void printMessage()

{

Console.WriteLine("successfully, press any key to continue...");

Console.ReadKey();

}

}

//class StudentCourseManagement

public class StudentCourseManagement

{

public void ShowStudentCourseMenu()

{

bool isContinue = true;

while (isContinue)

{

Console.Clear();

Console.WriteLine("1. 给学生选课");

Console.WriteLine("2. 显示学生所有已选课程");

Console.WriteLine("3. 返回上级菜单");

Console.Write("Enter your choice: ");

var choice = Console.ReadLine();

switch (choice)

{

case "1":

SelectCourse();

break;

case "2":

ShowSelectedCourses();

break;

case "3":

isContinue = false;

break;

default:

Console.WriteLine("输入错误,请重新输入");

break;

}

}

}

void SelectCourse()

{

Console.WriteLine("请输入学生的学号:");

var str = Console.ReadLine();

using (var context = new \_2114010313DbContext())

{

var checkStudent = context.Students.Find(str);

if (checkStudent == null)

{

Console.WriteLine($"不存在学号为:{str}的学生,请重试");

printMessage();

}

else

{

Console.WriteLine("请输入该学生的初始密码:");

var password = Console.ReadLine();

if (password == checkStudent.InitPassword)

{

Console.WriteLine("密码正确,请输入学生选课的课程号:");

var CourseID = Console.ReadLine();

var checkCourse = context.Courses.Find(CourseID);

if (checkCourse == null)

{

Console.WriteLine($"不存在课程号为:{CourseID} 的课程, 请重试");

printMessage();

return;

}

else

{

var stuCourse = new Stucourse1();

stuCourse.StudentId = str;

stuCourse.CourseId = CourseID;

context.Stucourse1s.Add(stuCourse);

context.SaveChanges();

Console.WriteLine("为该名学生选课成功...");

printMessage();

}

}

else

{

Console.WriteLine("密码错误,请重试");

printMessage();

return;

}

}

}

}

void ShowSelectedCourses()

{

Console.WriteLine("请输入想要查看选课信息的学生学号:");

var str = Console.ReadLine();

using (var context = new \_2114010313DbContext())

{

var checkStudent = context.Students.Find(str);

if (checkStudent == null)

{

Console.WriteLine($"不存在学号为:{str}的学生,请重试");

printMessage();

return;

}

var stuCourses = context.Stucourse1s.Where(Stucourse1 => Stucourse1.StudentId == str).ToList();

if (stuCourses.Count == 0)

{

Console.WriteLine("该学生没有选课");

printMessage(); return;

}

var Student = context.Students.Find(str);

foreach (var stuCourse in stuCourses)

{

var Course = context.Courses.Find(stuCourse.CourseId);

Console.WriteLine($"学号为{str}的学生{Student.StudentName}选了{stuCourse.CourseId}课程,课程名称为:{Course.CourseName}");

}

printMessage();

}

}

//输出提示信息

public void printMessage()

{

Console.WriteLine("successfully, press any key to continue...");

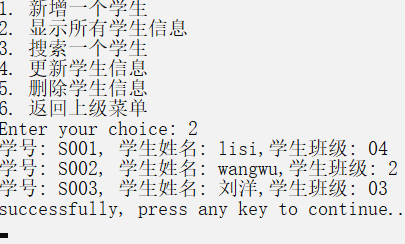
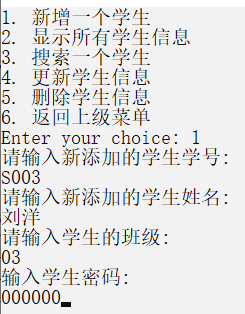
Console.ReadKey();

}

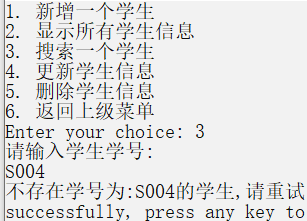
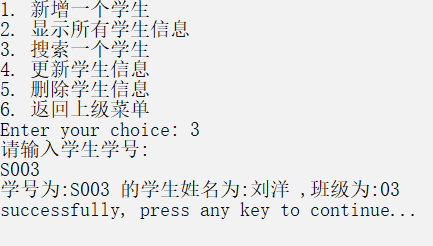
}

# 六．实验结果（包含产生结果用到的测试用例和过程说明）

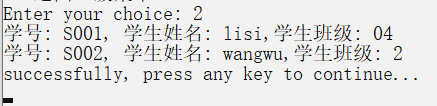
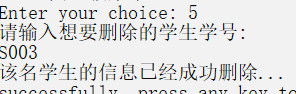
(一) 测试学生信息管理功能



新增一个学生信息的时候，需要保证学生学号不与已存在的学生学号冲突并且以S开头，输入学生所有信息以后，按下回车键，学生信息就会添加到数据库中。输入2显示所有学生信息可以看到新增加的学生信息，如图所示。

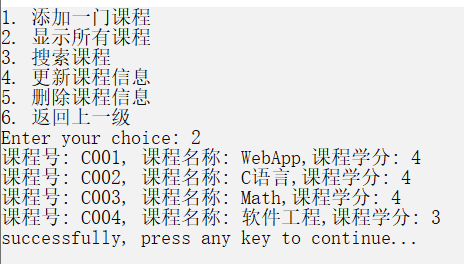
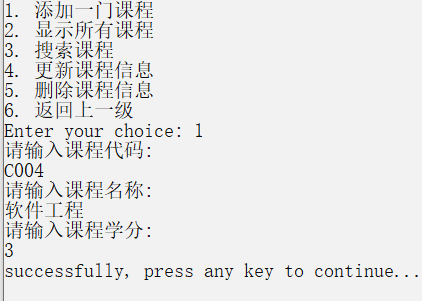


搜索学生信息时，如果输入一个不存在的学号信息，将会输出错误信息；输入一个合法的学号，就将学生信息打印在控制台上。

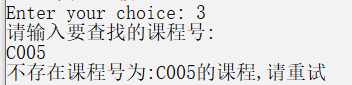
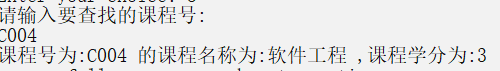


删除学生信息，输入学号即可删除学生信息，删除后再显示学生信息时，可以发现S003学号对应的学生信息已经删除。

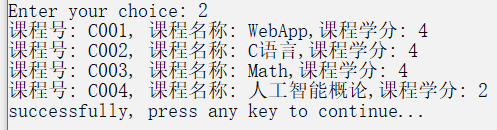
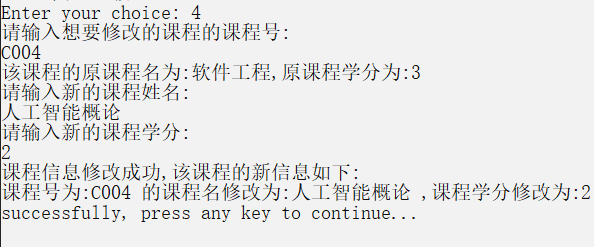
（二）测试课程信息管理功能



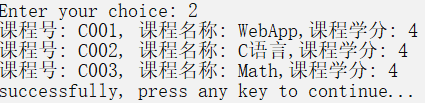
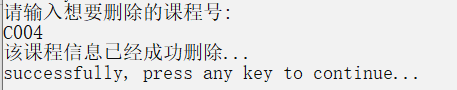
添加一门新的课程，添加时输入的课程代码必须正确，否则会输出错误信息，在正确输入所有课程信息按下回车键以后，然后输入2显示所有课程信息，可以看出课程代码为C004的课程已经被加入到了数据库中。



搜索课程信息和搜索学生信息相同，只有输入课程号正确的情况下会返回课程相关信息。

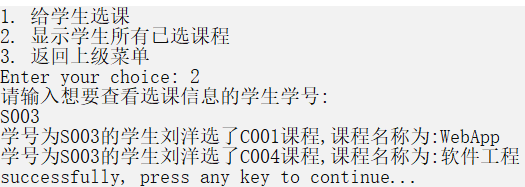
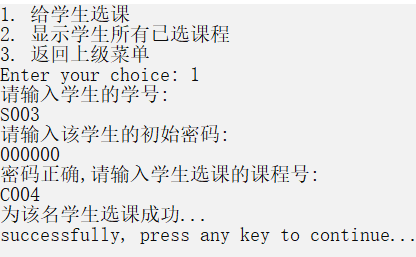


更新课程信息，如图所示，在正确输入课程号以后，输入新的课程信息，即可对课程信息进行更改，然后显示所有课程信息，发现课程号为C004的课程已经更新。



删除课程信息，输入合法的课程号以后，程序删除数据库中课程号对应的表项内容，然后再显示所有课程信息，发现课程号为C004的课程已经成功删除。

（三）学生选课管理系统



测试给学生选课的功能，首先需要输入学生的学号以及该学生的初始密码，如果两个信息错误将会返回错误信息，然后需要输入正确的课程号，所有信息正确的情况下就能选课成功，在控制台输入2，查看学生所有已选课程信息，控制台输出学生的选课信息。

# 七．存在的问题与收获

（一）存在的问题

在测试控制台程序中，在删除一个学生信息或者一门课程信息的时候，无法在学生选课信息表中将对应的表项也进行删除，即级联删除。如果不实现级联删除，当尝试删除一个已经选过课程的学生信息的时候，在学生信息表中能够正确删除该表项，但是在学生选课信息表中仍然存在这个已删除学生的学号信息，当尝试对学生选课信息进行查询的时候，由于学生表中已经不存在这个学号信息，所以会出现null异常。同理，删除被学生选过的课程信息也会抛出该异常。

1. 收获

通过本次实验，成功完成了一个简单的学生管理系统，实现了对学生信息、课程信息、学生选课信息的增、删、改、查操作，并将这些操作反映在远程数据库中。通过使用Entity Framework，简化了数据库操作的实现，使得代码更加简洁易读。