# 自动排课系统

目录

[**自动排课系统** 1](#_Toc22311614)

[**数据库设计** 2](#_Toc22311615)

[班级信息表Z\_TMP\_CLASSINFO 2](#_Toc22311616)

[课程信息表 Z\_TMP\_COURSESUM 2](#_Toc22311617)

[教师休假信息表 Z\_TMP\_TIMEOFF 3](#_Toc22311618)

[教师信息表 Z\_TMP\_TEACHERINFO 3](#_Toc22311619)

[课程表Z\_TMP\_TIMETABLE 4](#_Toc22311620)

[**用户界面** 5](#_Toc22311621)

[**主要逻辑算法** 7](#_Toc22311622)

[GenerateTiemTable(int n) 8](#_Toc22311623)

[getString(int b) 13](#_Toc22311624)

[TimeOffConstraint(int x, int y, int CourseNumber, int Class) 14](#_Toc22311625)

[TeacherFinderSEQ(string Course,int Class)存储过程版本 16](#_Toc22311626)

[TeacherFinder(string Course,int Class)一般版本 18](#_Toc22311627)

[Converter(int x,int y) 18](#_Toc22311628)

[Check() 20](#_Toc22311629)

[**遇到的问题以及解决的办法** 21](#_Toc22311630)

[一个表格完成增和改的操作 21](#_Toc22311631)

[模糊查询的优化办法 23](#_Toc22311632)

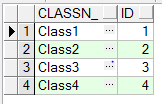
[如何在课程信息更改的情况下教师信息负责课程一栏名称跟着一起变化 24](#_Toc22311633)

[多名教师负责一个班级的同一节课的问题 26](#_Toc22311634)

[如何在最后将教师的信息与负责的课程一起显示在课程表上 30](#_Toc22311635)

# 数据库设计

### 班级信息表Z\_TMP\_CLASSINFO



这里主要显示需要排课的班级的信息





ClassN\_表示班级名字 字符类型为varchar2

ID表示班级的ID信息 字符类型为number作为班级信息表的主键

### 课程信息表 Z\_TMP\_COURSESUM



这里记录需要随机排课的课程





CID\_表示课程名字 字符类型为number 作为课程信息表的主键

CNAME\_表示课程的ID名称 字符类型为varchar2

### 教师休假信息表 Z\_TMP\_TIMEOFF

****

这里记录教师休假的时间，在休假的时间不再给这位教师安排课程

****

****

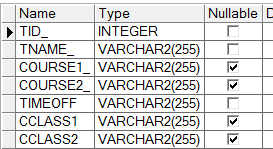
ID\_表示休假时间段的id，字符类型为integer

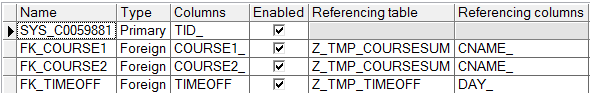
DAY\_表示休假时间，字符类型为VARCHAR2，unique类型

### 教师信息表 Z\_TMP\_TEACHERINFO



这里记录老师的信息以及需要负责的课程，休假的信息以及所需负责的班级





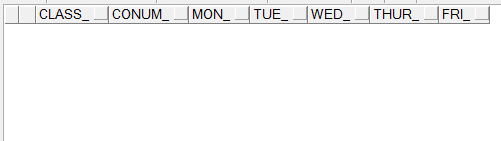
TID\_表示教师的ID信息 字符类型为number 作为教师信息表的主键

TNAME\_表示教师的姓名 字符类型为varchar2

COURSE1\_与COURSE2\_表示该名教师所需负责的两个课程，字符类型为varchar，作为上面课程信息表的外键

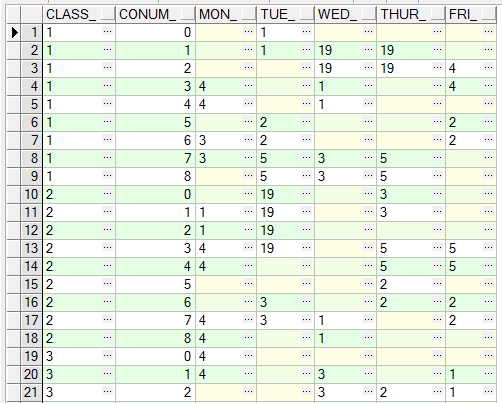
TIMEOFF 为教师休假的时间段，字符类型为varchar，作为教师休假时间段表的外键

### 课程表Z\_TMP\_TIMETABLE

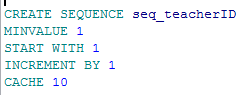


此表用来存放所有课程安排好之后的课程信息，故而一开始为空

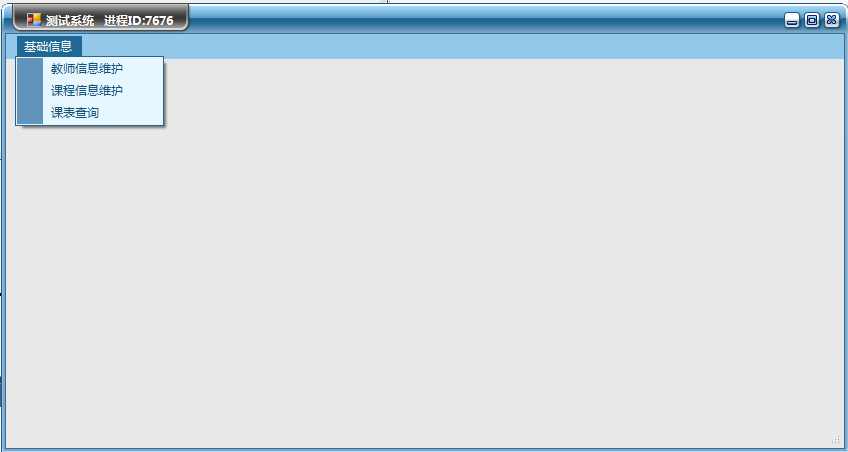
在随机生成课表之后，部分如下：



以上所有表的ID字段均为自增，因为Oracle中没有MySQL中的自增函数auto\_increment,这里采用sequence代替：

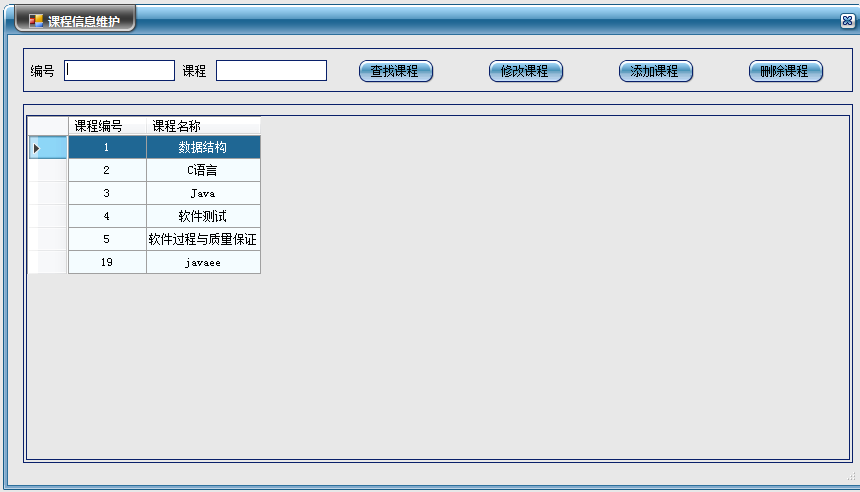


# 用户界面

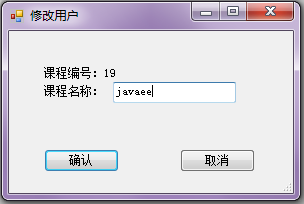


Main.cs

这里是三个基本的维护界面，分别表示教师的信息维护，课程信息维护，以及最终生成课表的界面



CourseInfo.cs

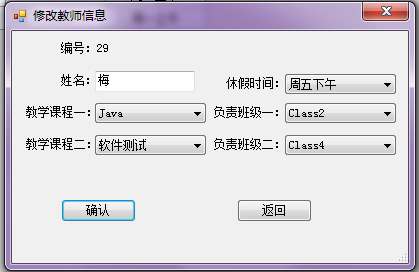


SubCourse.cs

这个是课程信息的维护界面，包括增删改查的功能，查询采用模糊查询



TeacherInfo.cs



SubTeacher.cs

这个是教师信息的维护界面，同样包括增删改查，并且模糊查询



TimeTable.cs

这里是课表生成的界面，包括查询课表和删除当前课表。并且采用模糊查询

# 主要逻辑算法

1. 随机生成课表并插入在课程信息表中维护的课程，规则为相隔两个班级一起排课即一个老师负责两个班级，同时一节课自动生成两节。
2. 更改老师信息以及课程信息时课表中的信息也会跟着一起改变
3. 老师在休假时间不得安排课程，不得在同一天安排两个相冲突的课程（java与c语言）
4. 上午最后一节课和下午第一节课不能安排连续的课程
5. 最后在课表上显示出课程名称以及教师的姓名信息
6. 若有班级的课程没有老师负责，则提示该课程信息并提醒安排课程信息

我的解决办法大致是写了一个GenerateTiemTable函数通过随机数将Z\_TMP\_timetable这个空白表格用数字填上课程所在位置，再将这些数字与课程的ID相对应，并在每一次加数字时用TimeOffConstraint函数来判断是否满足限制条件，最后再结合教师信息表一起打印出来。

### GenerateTiemTable(int n)

主要逻辑，通过随机数生成课表

private void GenerateTiemTable(int n)

{

int i = 0;

int m = n + 2;

int count = 0;

long tick = DateTime.Now.Ticks;

Random ran = new Random((int)(tick & 0xffffffffL) | (int)(tick >> 32));//指定当前时间作为随机数种子可以确保Random类的对象来产生随机数不同

int[,] course1 = new int[5, 9];

string sqltext = "select \* from z\_tmp\_CourseSum";

string sqlCount = "select count(\*)from z\_tmp\_CourseSum";

int Num = Convert.ToInt32(dao.ExecuteScale(sqlCount).ToString());

DataTable dtGrid = dao.GetDataTable(sqltext);

int[] Name = new int[Num];//定义一个Name数组长度等于z\_tmp\_CourseSum（课程信息表）中字段的个数Num

foreach (DataRow dr in dtGrid.Rows)

{

Name[count] = int.Parse(dr["CID\_"].ToString());

count++;

} //将课程信息表中的每一个课程名称写入Name数组当中

while (i < Num \* 2)//两个相邻班级一起安排课程，当一个课程被安排完之后i数值加1，当两个班课程都安排结束，课程数量i即等于课程总数Num\*2;

{

int x = ran.Next(0, 5);//随机数x表示星期数（0—4分别表示周一到周五）

int y = ran.Next(0, 9);//随机数y表示第几节课（0—9分别表示第一节课到9节课）

if (i < Num)//开始排第一个班级

{

if (y == 8)//当随机到当天最后一节课

{

if (course1[x, y] == 0 && course1[x, y - 1] == 0)//先判断当前和上一节课是否已经被安排课程，若已经被占用则跳出重新抽取随机数

{

if (TimeOffConstraint(x, y, Name[i], n) == 1)//TimeOffConstraint函数，判断当前教师在这一时间段是否为休假时间，当天是否有相互冲突的两节课，若有则跳出重新抽取随机数

{

string strSql = "insert into z\_tmp\_timetable(class\_,conum\_," + getString(x) + ")values('" + n + "','7','" + Name[i] + "')";//通过getString（x）函数将随机生成的数字变成相对应的星期数

string strSql1 = "insert into z\_tmp\_timetable(class\_,conum\_," + getString(x) + ")values('" + n + "','8','" + Name[i] + "')";

object obj = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '7'and class\_='" + n + "'");

object obj1 = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '8'and class\_='" + n + "'");

if (obj == null)

{

dao.ExecuteNonQuery(strSql); //若该天没有已经安排的课程则执行插入操作

}

else if (obj != null)

{

dao.ExecuteNonQuery("update z\_tmp\_timetable set " + getString(x) + "='" + Name[i] + "'where conum\_='7'and class\_='" + n + "'");//若该天已经有安排的课程则执行更新操作（要不然就就会出现很多个同样的星期因为不断insert）

}

if (obj1 == null)//因为两节课连续所有在执行一次该操作

{

dao.ExecuteNonQuery(strSql1);

}

else if (obj1 != null)

{

dao.ExecuteNonQuery("update z\_tmp\_timetable set " + getString(x) + "='" + Name[i] + "'where conum\_='8'and class\_='" + n + "'");

}

course1[x, y] = 1;

course1[x, y - 1] = 1;

i++;

}

else

{

continue;

}

}

else

{

continue;

}

}

else//若没有随机到最后一节课则正常执行与上相同

{

if (course1[x, y] == 0 && course1[x, y + 1] == 0)

{

if (TimeOffConstraint(x, y, Name[i], n) == 1)

{

int t = y + 1;

string strSql = "insert into z\_tmp\_timetable(class\_,conum\_," + getString(x) + ")values('" + n + "','" + y + "','" + Name[i] + "')";

string strSql1 = "insert into z\_tmp\_timetable(class\_,conum\_," + getString(x) + ")values('" + n + "','" + t + "','" + Name[i] + "')";

object obj = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '" + y + "'and class\_='" + n + "'");

object obj1 = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '" + t + "'and class\_='" + n + "'");

if (obj == null)

{

dao.ExecuteNonQuery(strSql);

}

else if (obj != null)

{

dao.ExecuteNonQuery("update z\_tmp\_timetable set " + getString(x) + "='" + Name[i] + "'where conum\_='" + y + "'and class\_='" + n + "'");

}

if (obj1 == null)

{

dao.ExecuteNonQuery(strSql1);

}

else if (obj1 != null)

{

dao.ExecuteNonQuery("update z\_tmp\_timetable set " + getString(x) + "='" + Name[i] + "'where conum\_='" + t + "'and class\_='" + n + "'");

}

course1[x, y] = 1;

course1[x, y + 1] = 1;

i++;

}

else

{

continue;

}

}

}

} else//第二个班级，这时i>Num，重复上操作

{

if (y == 8)

{

if (course1[x, y] == 0 && course1[x, y - 1] == 0)

{

if (TimeOffConstraint(x, y, Name[i - Num], n) == 1)

{

string strSql = "insert into z\_tmp\_timetable(class\_,conum\_," + getString(x) + ")values('" + n + "','7','" + Name[i - Num] + "')";

string strSql1 = "insert into z\_tmp\_timetable(class\_,conum\_," + getString(x) + ")values('" + n + "','8','" + Name[i - Num] + "')";

object obj = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '7'and class\_='" + n + "'");

object obj1 = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '8'and class\_='" + n + "'");

if (obj == null)

{

dao.ExecuteNonQuery(strSql);

}

else if (obj != null)

{

dao.ExecuteNonQuery("update z\_tmp\_timetable set " + getString(x) + "='" + Name[i - Num] + "'where conum\_='7'and class\_='" + n + "' ");

}

if (obj1 == null)

{

dao.ExecuteNonQuery(strSql1);

}

else if (obj1 != null)

{

dao.ExecuteNonQuery("update z\_tmp\_timetable set " + getString(x) + "='" + Name[i - Num] + "'where conum\_='8'and class\_='" + n + "'");

}

course1[x, y] = 1;

course1[x, y - 1] = 1;

i++;

}

else

{

continue;

}

}

else

{

continue;

}

}

else

{

if (course1[x, y] == 0 && course1[x, y + 1] == 0)

{

if (TimeOffConstraint(x, y, Name[i - Num], n) == 1)

{

int t = y + 1;

string strSql = "insert into z\_tmp\_timetable(class\_,conum\_," + getString(x) + ")values('" + n + "','" + y + "','" + Name[i - Num] + "')";

string strSql1 = "insert into z\_tmp\_timetable(class\_,conum\_," + getString(x) + ")values('" + n + "','" + t + "','" + Name[i - Num] + "')";

object obj = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '" + y + "'and class\_='" + n + "'");

object obj1 = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '" + t + "'and class\_='" + n + "'");

if (obj == null)

{

dao.ExecuteNonQuery(strSql);

}

else if (obj != null)

{

dao.ExecuteNonQuery("update z\_tmp\_timetable set " + getString(x) + "='" + Name[i - Num] + "'where conum\_='" + y + "'and class\_='" + n + "'");

}

if (obj1 == null)

{

dao.ExecuteNonQuery(strSql1);

}

else if (obj1 != null)

{

dao.ExecuteNonQuery("update z\_tmp\_timetable set " + getString(x) + "='" + Name[i - Num] + "'where conum\_='" + t + "'and class\_='" + n + "'");

}

course1[x, y] = 1;

course1[x, y + 1] = 1;

i++;

}

else

{

continue;

}

}

}

}

}

for (int q = 0; q < 9; q++)//最后将所有没有随机到的节数加上空白

{

object objM1 = dao.ExecuteScale("select \* from z\_tmp\_timetable where conum\_ = '" + q + "'and class\_='" + n + "'");

if (objM1 == null)

{

string sql = "insert into z\_tmp\_timetable(class\_,conum\_,mon\_,tue\_,wed\_,thur\_,fri\_)values('" + n + "','" + q + "','','','','','')";

dao.ExecuteNonQuery(sql);

}

}

}

### getString(int b)

通过getString（x）函数将随机生成的数字变成相对应的星期数

static string getString(int b) //变量b为随机数x

{

string str;

switch (b)

{

case 0:

str = "MON\_";

break;

case 1:

str = "TUE\_";

break;

case 2:

str = "WED\_";

break;

case 3:

str = "THUR\_";

break;

case 4:

str = "FRI\_";

break;

default:

str = "";

break;

}

return str;

}

### TimeOffConstraint(int x, int y, int CourseNumber, int Class)

判断当前教师在这一时间段是否为休假时间，当天是否有相互冲突的两节课，若有则返回

private int TimeOffConstraint(int x, int y, int CourseNumber, int Class)

{

string TimeOffB = Converter(x, y);//通过Convert函数获得当前的时间段TimeOffB

string sqlNum = "select Cname\_ from z\_tmp\_CourseSum where CID\_='" + CourseNumber + "'";

string Course = dao.ExecuteScale(sqlNum).ToString();//获得当前的课程名称

string TName = TeacherFinderSEQ(Course, Class);//通过存储过程简化逻辑，找出这门课负责的老师

string sqlnum = "select count(\*)from z\_tmp\_timetable where class\_ = '" + Class + "'";

int num = Convert.ToInt32(dao.ExecuteScale(sqlnum).ToString());

string Sql = "select \* from Z\_tmp\_timetable";

DataTable dtGrid = dao.GetDataTable(Sql);

int state = 0;

if (TName == "Missing")//找不到负责该课程的老师，返回-1

{

return -1;

}

else//若有老师则查找这名老师的休假时间

{

string sql = "select TimeOff from z\_tmp\_teacherinfo where TName\_ ='" + TName + "' ";

string TimeOff = dao.ExecuteScale(sql).ToString();

if (TimeOffB == "NO")//表明这节课为第五节课，不能安排课程，否则违反了第五节课第六节课不能安排连续课程的规律，返回0

{

return 0;

}

if (TimeOff == TimeOffB)//若该时间段与该老师休假时间段相同则同样无法安排课程，返回0

{

return 0;

}

else//若同时满足上述条件则观察是否存在冲突课程，在这里课程编号3与6相对应的java与c语言冲突

{

if (num == 0 ||( CourseNumber != 3&&CourseNumber != 6))//如果当天没有出现这两节课或者还未给当天排课则返回1

{

return 1;

}

if (CourseNumber == 3)//若出现其中一节则查找是否有另外一节课

{

DataRow[] drList = dtGrid.Select(""+ getString(x)+"= '6'");//在datarow中搜索指定的值

if (drList.Length > 0)//有则return-1

{

state = -1;

}

if (state == 0)

{

return 1;

}

else

{

return 0;

}

}

if (CourseNumber == 6)//与上相同

{

DataRow[] drList = dtGrid.Select("" + getString(x) + "= '3'");

if (drList.Length > 0)

{

state = -1;

}

if (state == 0)

{

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

}

}

### TeacherFinderSEQ(string Course,int Class)存储过程版本

寻找该课程的负责老师信息

private string TeacherFinderSEQ(string Course,int Class)

{

string[] returnparm = new string[3];

string errinfo = "";

string sysIxdex = "";

string sysMessage = "";

dao.RunProcedure("SP\_TMP\_TeacherFinder", new OracleParameter[5] {

new OracleParameter("v\_Course", Course) { OracleType = OracleType.VarChar, Size=12 },

new OracleParameter("v\_Class", Class) { OracleType = OracleType.Int32, Size=12 },

new OracleParameter("o\_retcode", OracleType.VarChar,10) { Direction = ParameterDirection.Output, OracleType = OracleType.VarChar },

new OracleParameter("o\_retmsg", OracleType.VarChar,10) { Direction = ParameterDirection.Output, OracleType = OracleType.VarChar },

new OracleParameter("o\_retmsg1", OracleType.VarChar,10) { Direction = ParameterDirection.Output, OracleType = OracleType.VarChar }

}, out returnparm, out errinfo);//在存储过程中执行查找逻辑

sysIxdex = returnparm[0].Trim();

sysMessage = returnparm[1].Trim();

if (sysIxdex == "1")

{

return sysMessage;

}

else

{

string WAR = string.Format("未安排班级：Class{0},{1}的课程；", Class, Course);

warning.Append(WAR);

return sysMessage;//若有班级的课程未安排老师则将该信息加到字符串中提示用户

}

}

存储过程中的代码：

create or replace procedure SP\_TMP\_TeacherFinder

(v\_Course in varchar2,

v\_Class in number,

o\_retcode out varchar,

o\_retmsg out varchar2,

o\_retmsg1 out varchar2

) is

Cou varchar2(6);

begin

o\_retcode := '0';

o\_retmsg := '-';

o\_retmsg1 := '-';

select TName\_ into Cou from z\_tmp\_TeacherInfo where CClass1 =(select classn\_ from z\_Tmp\_Classinfo z where z.id=v\_Class) and course1\_ = v\_Course or CClass1 =(select classn\_ from z\_Tmp\_Classinfo z where z.id=v\_Class) and course2\_ = v\_Course or CClass2 =(select classn\_ from z\_Tmp\_Classinfo z where z.id=v\_Class) and course1\_ =v\_Course or CClass2 =(select classn\_ from z\_Tmp\_Classinfo z where z.id=v\_Class) and course2\_ = v\_Course;

o\_retcode := '1';

o\_retmsg := Cou;

exception

when NO\_DATA\_FOUND then

o\_retcode := '0';

o\_retmsg := 'Missing';

o\_retmsg1 := v\_Course;

end SP\_TMP\_TeacherFinder;

这里其实不用存储过程好像更简单。。

### TeacherFinder(string Course,int Class)一般版本

private string TeacherFinder(string Course,int Class)

{

string strSql = "select TName\_ from z\_tmp\_TeacherInfo where CClass1 ='Class" + Class + "' and course1\_ ='" + Course + "' or CClass1 ='Class" + Class + "' and course2\_ ='" + Course + "' or CClass2 ='Class" + Class + "' and course1\_ ='" + Course + "' or CClass2 ='Class" + Class + "' and course2\_ ='" + Course + "'";

object obj = dao.ExecuteScale(strSql);

if (obj==null)

{

string WAR = string.Format("未安排班级：Class{0},{1}的课程；", Class, Course);

warning.Append(WAR);

return "Missing";

}

else

{

return dao.ExecuteScale(strSql).ToString();

}

}

### Converter(int x,int y)

将数字转化成该天对应的休假时间段

private string Converter(int x,int y)

{

string strInfo="";

if (y == 4)

{

strInfo = "NO";

}

else

{

switch (x)

{

case 0:

if (y < 5)

strInfo = "周一上午";

else

strInfo = "周一下午";

break;

case 1:

if (y < 5)

strInfo = "周二上午";

else

strInfo = "周二下午";

break;

case 2:

if (y < 5)

strInfo = "周三上午";

else

strInfo = "周三下午";

break;

case 3:

if (y < 5)

strInfo = "周四上午";

else

strInfo = "周四下午";

break;

case 4:

if (y < 5)

strInfo = "周五上午";

else

strInfo = "周五下午";

break;

default:

strInfo = "";

break;

}

}

return strInfo;

}

### Check()

将每一节课带入TeacherFinder函数当中查看是否有课程没有安排老师

private bool Check()

{

int count = 0;

int state = 0;

string strSql = "select \* from z\_tmp\_CourseSum";

string sqlCount = "select count(\*)from z\_tmp\_CourseSum";

int Num = Convert.ToInt32(dao.ExecuteScale(sqlCount).ToString());

string sql = "select count(\*)from z\_tmp\_classinfo";

int n = Convert.ToInt32(dao.ExecuteScale(sql).ToString());

DataTable dtGrid = dao.GetDataTable(strSql);

string[] Name = new string[Num];

foreach (DataRow dr in dtGrid.Rows)

{

Name[count] = dr["CName\_"].ToString();

count++;

}

for(int j = 1; j < n + 1; j++)

{

for (int i = 0; i < Num; i++)

{

if (TeacherFinderSEQ(Name[i], j) == "Missing")

{

state = -1;

}

}

}

if (state == 0)

{

return true;

}

else

{

return false;

}

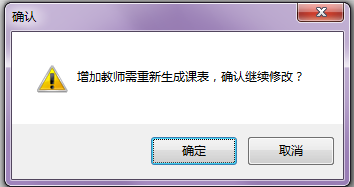
}

# 遇到的问题以及解决的办法

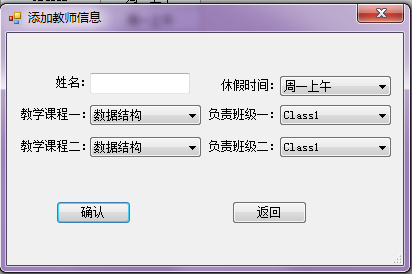
### 一个表格完成增和改的操作

在我最开始的版本当中，增和改都是另外建了一个窗口，比较花费时间，于是我便再想如何和能够将增和改这两个操作放在一个窗口中去，这里以教师信息的维护为例子。

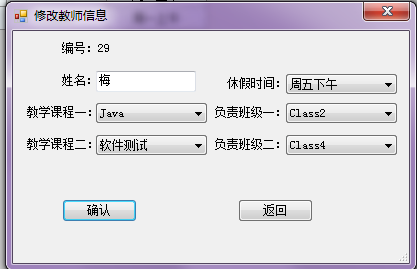
当用户点了增加课程的按钮时，给出一个首先给出提示



当确定之后给出表格，这时通过函数传入一个参数state为1表示当前状态为增加字段，同理若为修改字段时传入参数state=2



State=1时



State=2时

private void btnadd\_Click(object sender, EventArgs e)

{

string Confirm = "增加教师需重新生成课表，确认继续修改？";

DialogResult dr = MessageBox.Show(Confirm, "确认", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

deleteTable();

SubTeacher addform = new SubTeacher();

addform.getState(1);//传入参数1，表示增加

addform.ShowDialog();

}

Query();

}

if (states == 1)//当传入的参数等于1的时候，即代表增加字段

{

Text = "添加教师信息";//修改标题

arr[0] = "1";

}

if (states == 2)//当传入的参数等于1的时候，即代表修改字段

{

Text = "修改教师信息";

if (arr[2] == "")

{

arr[2] = "无";

}

if (arr[3] == "")

{

arr[3] = "无";

}

if (arr[5] == "")

{

arr[5] = "无";

}

if (arr[6] == "")

{

arr[6] = "无";

}

lbID.Text = "编号：" + arr[0];

txtName.Text = arr[1];

CoBoxC1.Text = arr[2];

CoBoxC2.Text = arr[3];

CoBoxDay.Text = arr[4];

CoBoxCL1.Text = arr[5];

CoBoxCL2.Text=arr[6];//读取当前字段并填入

最后通过oracle中存储过程来实现数据库更新。

### 模糊查询的优化办法

由于字段很多，若一个个sql语句写过来就会有一大堆，于是我想了一个可以优化一点的办法

private void btnSearch\_Click(object sender, EventArgs e)

{

string strID = txtID.Text.Trim();

string strName = txtName.Text.Trim();

string strCourse = txtCourse.Text.Trim();

string strDay = txtTimeOff.Text.Trim();

StringBuilder strSql = new StringBuilder();//新建的StringBuild可以随时向其中添加sql语句

if (strID.Length != 0)//若查找ID框体不为空，添加查找IDsql语句

{

strSql.Append(" TID\_ like '%"+ strID + "%'");

}

if (strName.Length != 0)//若查找名字框不为空

{

if (strSql.Length == 0)

{

strSql.Append(" Tname\_ like '%" + strName + "%'");//若这个strSql为空则可以直接添加查找语句

}

else

{

strSql.Append(" and Tname\_ like '%" + strName + "%'");//若不为空则需要加上链接词and

}

}

if (strDay.Length != 0)//以下同理

{

if (strSql.Length == 0)

{

strSql.Append(" timeoff like '%" + strDay + "%'");

}

else

{

strSql.Append(" and timeoff like '%" + strDay + "%'");

}

}

if (strCourse.Length != 0)

{

if (strSql.Length == 0)

{

strSql.Append(" course1\_ like '%" + strCourse + "%' or "+strSql.Replace("where","")+"course2\_ like '%" + strCourse + "%'");

}

else

{

strSql.Append(" and course1\_ like '%" + strCourse + "%' or "+ strSql.Replace("where","") + "and course2\_ like '%" + strCourse + "%'");

}

}

string str = "select \*from Z\_TMP\_TeacherInfo where" + strSql + " order by tid\_";//最后加上select开头就可以了

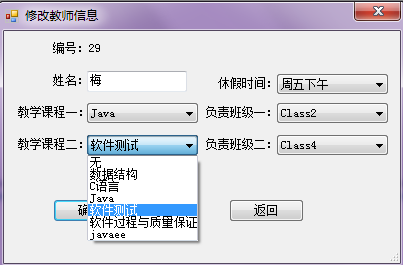
DataTable dtUser = dao.GetDataTable(str);

grdview.DataSource = dtUser;//打印表

}

### 如何在课程信息更改的情况下教师信息负责课程一栏名称跟着一起变化

之前最开始的版本中下拉框是手动输入的，使用导致在课程信息栏更改信息之后这个下拉栏信息未改变。



string strsql = ("select \* from z\_tmp\_COURSESUM order by CID\_");

string strDay = ("select \* from z\_tmp\_TimeOff");

string strClass = ("select \* from z\_tmp\_Classinfo");

DataTable dt1 = dao.GetDataTable(strsql);

DataTable dt2 = dao.GetDataTable(strsql);

DataTable dt3 = dao.GetDataTable(strDay);

DataTable dt4 = dao.GetDataTable(strClass);

DataTable dt5 = dao.GetDataTable(strClass);

CoBoxC1.DataSource = dt1;

DataRow dr1 = dt1.NewRow();

dr1["CName\_"] = "无";

dt1.Rows.InsertAt(dr1, 0);

CoBoxC1.DisplayMember = "CName\_";

CoBoxC1.ValueMember = "CID\_";

CoBoxC2.DataSource = dt2;

DataRow dr2 = dt2.NewRow();

dr2["CName\_"] = "无";

dt2.Rows.InsertAt(dr2, 0);

CoBoxC2.DisplayMember = "CName\_";

CoBoxC2.ValueMember = "CID\_";

CoBoxDay.DataSource = dt3;

CoBoxDay.DisplayMember = "Day\_";

CoBoxDay.ValueMember = "id\_";

CoBoxCL1.DataSource = dt4;

DataRow dr4 = dt4.NewRow();

dr4["ClassN\_"] = "无";

dt4.Rows.InsertAt(dr4, 0);

CoBoxCL1.DisplayMember = "ClassN\_";

CoBoxCL1.ValueMember = "ClassN\_";

CoBoxCL2.DataSource = dt5;

DataRow dr5 = dt5.NewRow();

dt5.Rows.InsertAt(dr5, 0);

dr5["ClassN\_"] = "无";

CoBoxCL2.DisplayMember = "ClassN\_";

CoBoxCL2.ValueMember = "ClassN\_";

通过将下拉框的内容和课程信息表的课程名称字段相连接，就可以在维护课程信息表的时候一起修改下拉框中的内容，并且手动输入“无”，增加没有负责课程的选项

### 多名教师负责一个班级的同一节课的问题

这个问题相对于其他几个问题而言花的时间最多，因为若老师负责一个班级一节课还行，但是涉及到了一个老师负责两个班级的两节课，并且有一门课程可以为空，就相对而言比较复杂，因为逻辑比较多，我将其写在了存储过程当中：SP\_TMP\_OccupyCheck

#### SP\_TMP\_OccupyCheck

create or replace procedure SP\_TMP\_OccupyCheck (

v\_state in number,

v\_tid in varchar2,

v\_course1 in varchar2,

v\_course2 in varchar2,

v\_class1 in varchar2,

v\_class2 in varchar2,

v\_tname in varchar2,

v\_timeoff in varchar2,

o\_retcode out varchar2,

o\_retmsg out varchar2

) */\*输入与输出的参数\*/*

is

v\_cou number(2) ;

course1 varchar2(20);

course2 varchar2(20);

class1 varchar2(20);

class2 varchar2(20);*/\*全局变量\*/*

begin

o\_retcode := '0';

o\_retmsg := '-';*/\*初始化参数\*/*

course1 :=v\_course1;

course2 :=v\_course2;

class1 :=v\_class1;

class2 :=v\_class2;

if(v\_course1 = '无')then

course1:='';

v\_cou := FN\_TMP\_OccupyCheck(v\_class1,v\_class2,v\_course2,v\_course2,0); */\*通过函数FN\_TMP\_OccupyCheck来查课程是否冲突，当课程1为空时，原本课程1的位置也变成课程2，最后传入参数0代表存在“无”课程或者班级\*/*

if(v\_cou = 0)then

o\_retmsg := '已存在课程';

o\_retcode := '0';

return;

end if;

elsif(v\_course2 = '无')then

course2:='';

v\_cou := FN\_TMP\_OccupyCheck(v\_class1,v\_class2,v\_course1,v\_course1,0);

if(v\_cou = 0)then

o\_retmsg := '已存在课程';

o\_retcode := '0';

return;

end if;

elsif(v\_class1 = '无')then

class1:='';

v\_cou := FN\_TMP\_OccupyCheck(v\_class2,v\_class2,v\_course1,v\_course2,0); */\*通过函数FN\_TMP\_OccupyCheck来查课程是否冲突，当课程1为空时，原本课程1的位置也变成课程2，最后传入参数0代表存在“无”课程或者班级\*/*

if(v\_cou = 0)then

o\_retmsg := '已存在课程';

o\_retcode := '0';

return;

end if;

elsif(v\_class2 = '无')then

course2:='';

v\_cou := FN\_TMP\_OccupyCheck(v\_class1,v\_class1,v\_course1,v\_course2,0);

if(v\_cou = 0)then

o\_retmsg := '已存在课程';

o\_retcode := '0';

return;

end if;

else

v\_cou := FN\_TMP\_OccupyCheck(v\_class1,v\_class2,v\_course1,v\_course2,1); */\*传入参数1代表不存在“无”的课程\*/*

if(v\_cou = 0)then

o\_retmsg := '已存在课程';

o\_retcode := '0';

return;

end if;

end if;

if(v\_state = '1')then

insert into z\_tmp\_teacherinfo (tid\_,tname\_,course1\_,course2\_,timeoff,CClass1,CClass2) values(seq\_teacher.nextval,v\_tname,course1,course2,v\_timeoff,class1,class2);

commit;

o\_retmsg := '保存成功';

o\_retcode := '1';

return;

end if ;

if(v\_state = '2')then

update z\_tmp\_teacherinfo set tname\_=v\_tname ,course1\_=course1, course2\_=course2, timeoff=v\_timeoff ,CClass1=class1 , CClass2=class2 where tid\_=v\_tid;

commit;

o\_retmsg := '保存成功';

o\_retcode := '2';

return;

end if;

end SP\_TMP\_OccupyCheck ;

#### FN\_TMP\_OccupyCheck

判断是否有重复负责的课程

CREATE OR REPLACE FUNCTION FN\_TMP\_OccupyCheck( v\_class1 in varchar2,

v\_class2 in varchar,

v\_course1 in varchar2,

v\_course2 in varchar2,

v\_Type in number

) return number is

v\_cou1 number(4);

v\_cou2 number(4);

v\_cou3 number(4);

v\_cou4 number(4);

begin

select count(1) into v\_cou1 from z\_tmp\_teacherinfo where CClass1=v\_class1 and Course1\_=v\_course1 or CClass1=v\_class2 and Course2\_=v\_course1;

select count(1) into v\_cou2 from z\_tmp\_teacherinfo where CClass2=v\_class1 and Course1\_=v\_course2 or CClass2=v\_class2 and Course2\_=v\_course2;

if(v\_cou1>0 or v\_cou2>0) then

return 0 ;

end if;*/\*若存在“无”则可以判断出是否存在重复班级\*/*

if(v\_Type=1)then*/\*当不存在“无”时，则再加入两个判断\*/*

select count(1) into v\_cou3 from z\_tmp\_teacherinfo where CClass1=v\_class2 and Course1\_=v\_course1 or CClass1= v\_class1 and Course2\_=v\_course1;

select count(1) into v\_cou4 from z\_tmp\_teacherinfo where CClass2=v\_class2 and Course1\_=v\_course2 or CClass2= v\_class1 and Course2\_=v\_course2;

if( v\_cou3>0 or v\_cou4>0) then

return 0 ;

end if ;

end if ;

return 1;

exception

when others then

return 0;

end FN\_TMP\_OccupyCheck;

### 如何在最后将教师的信息与负责的课程一起显示在课程表上

这个问题尝试了很多办法最后想出了这一个但是还是十分的复杂：

StringBuilder str = new StringBuilder();

str.Append("select z.classn\_,t.conum\_,");//选取班级信息表中的班级名次，以及课程表中的课程数

str.Append("(select c.cname\_ || '(' || (select distinct e.tname\_ from z\_Tmp\_Teacherinfo e where(e.cclass1 = z.classn\_ and e.course1\_ = c.cname\_) or(e.course2\_ = c.cname\_ and e.cclass2 = z.classn\_) or(e.cclass1 = z.classn\_ and e.course2\_ = c.cname\_)or(e.cclass2 = z.classn\_ and e.course1\_ = c.cname\_) ) || ')' from z\_tmp\_coursesum c where c.cid\_ = t.mon\_) as mon\_,");//选取周一的课程和周一课程所对应的教师姓名

str.Append("(select c.cname\_ || '(' || (select distinct e.tname\_ from z\_Tmp\_Teacherinfo e where(e.cclass1 = z.classn\_ and e.course1\_ = c.cname\_) or(e.course2\_ = c.cname\_ and e.cclass2 = z.classn\_) or(e.cclass1 = z.classn\_ and e.course2\_ = c.cname\_)or(e.cclass2 = z.classn\_ and e.course1\_ = c.cname\_) ) || ')' from z\_tmp\_coursesum c where c.cid\_ = t.tue\_) as tue\_,");//选取周二的课程和周一课程所对应的教师姓名

str.Append("(select c.cname\_ || '(' || (select distinct e.tname\_ from z\_Tmp\_Teacherinfo e where(e.cclass1 = z.classn\_ and e.course1\_ = c.cname\_) or(e.course2\_ = c.cname\_ and e.cclass2 = z.classn\_) or(e.cclass1 = z.classn\_ and e.course2\_ = c.cname\_)or(e.cclass2 = z.classn\_ and e.course1\_ = c.cname\_) ) || ')' from z\_tmp\_coursesum c where c.cid\_ = t.wed\_) as wed\_,");//选取周三的课程和周一课程所对应的教师姓名

str.Append("(select c.cname\_ || '(' || (select distinct e.tname\_ from z\_Tmp\_Teacherinfo e where(e.cclass1 = z.classn\_ and e.course1\_ = c.cname\_) or(e.course2\_ = c.cname\_ and e.cclass2 = z.classn\_) or(e.cclass1 = z.classn\_ and e.course2\_ = c.cname\_)or(e.cclass2 = z.classn\_ and e.course1\_ = c.cname\_) ) || ')' from z\_tmp\_coursesum c where c.cid\_ = t.thur\_) as thur\_,");//选取周四的课程和周一课程所对应的教师姓名

str.Append("(select c.cname\_ || '(' || (select distinct e.tname\_ from z\_Tmp\_Teacherinfo e where(e.cclass1 = z.classn\_ and e.course1\_ = c.cname\_) or(e.course2\_ = c.cname\_ and e.cclass2 = z.classn\_) or(e.cclass1 = z.classn\_ and e.course2\_ = c.cname\_)or(e.cclass2 = z.classn\_ and e.course1\_ = c.cname\_) ) || ')' from z\_tmp\_coursesum c where c.cid\_ = t.fri\_) as fri\_");//选取周五的课程和周一课程所对应的教师姓名

str.Append(" from z\_tmp\_timetable t, z\_Tmp\_Classinfo z where t.class\_ = z.id order by Class\_, conum\_");

DataTable dtUser = dao.GetDataTable(str.ToString());

grdView.DataSource = dtUser;//最后再打印表