

北京邮电大学

程序设计实践实验报告



题 目： 学生信息管理系统

姓 名： 张 阳

学 院： 信息与通信工程学院

专 业： 通信工程

班 级： 2016211111

学 号： 2016210318

班内序号： 20

指导教师： 黄平牧老师

2018年6月

目录

摘要与前言	3
关键词	3
一、 程序设计的任务与要求	4
二、 数据库编程与关系型数据库	5
1. 数据库与数据库编程	5
2. 关系型数据库	5
3. MS SQL Server 与 ODBC 数据库连接	6
三、 程序的开发思路与结构框图	6
四、 主要函数功能和算法分析	8
五、 总体程序实现功能说明与展示	16
六、 故障与问题分析	21
七、 总结与心得体会	23
1. 编程的难点	23
2. 我的收获	23
3. 我的体会	23
八、 未来的改进	24
九、 参考资料	25

摘要与前言

本次实验使用 MFC 编写了一个学生信息管理系统的 Windows 客户端，使用本地的 MS SQL Server 作为数据库。使用 MFC ODBC 类连接数据库，访问数据库中的信息，对学生基本信息、课程信息进行显示、浏览、添加、修改和删除和统计学生的成绩计算出平均分和所修学分等功能。该实验的代码与实验报告我同时备份到我的 GitHub 网页上：

<https://github.com/zyzisy/Programming-Practice>

PS：由于最近处于考试周，实验报告现在才写好。因为使用的是我自己电脑本地的 SQL Server 数据库，因此代码在别的电脑上可能无法编译，程序也无法准确运行。

关键词

数据库编程 关系型数据库 MFC Windows 应用开发 C++编程

一、程序设计的任务与要求

1. 功能需求

教学管理人员能够使用 StudentInfo 程序对学生基本信息课程信息进行管理, 包括数据的添加、修改、删除和浏览;能够对学生选课进行管理, 包括添加学生选课信息、录入成绩;能够使用查询功能, 快速查看到指定学生或指定课程的基本信息以及所指定学生的选课信息;能够对学生选课情况进行简单的统计, 包括所选的总的课程数、总学分数及平均成绩。

要注意在添加学生基本信息、课程信息相关数据时, 学号和课程号不能重复;还有在添加学生选课信息时, 要求该学生和课程必须是存在的, 而且不能添加重复的选课信息。

2. 界面要求

应用程序提供的操作界面, 可以方便用户进行功能选择, 实现信息的管理和查询, 并可以清晰地显示相关信息。

3. 成绩评定

占 50 分: 验收+实验报告 (含源代码)

4. 实验报告

开发思路, 主要技术, 相关算法, 主要编程点, 编程心得

5. 案例提交

a) 实验报告 (含源代码) 提交至: homework_bupt@163.com

b) 文件名命名格式: 班级_姓名_小班序号_题目.rar

c) 最后提交时间: 6 月 30 日

注意: 为使压缩文件小, 先要把两个 debug 目录下的文件全删掉。若文件太大, 可存到百度云, 在提交邮件中提供一个文件下载的链接。实验报告不要放在压缩文件中, 要单独提交!

二、数据库编程与关系型数据库

1. 数据库与数据库编程

数据库(Database)是按照数据结构来组织、存储和管理数据的仓库，它产生于距今六十多年前，随着信息技术和市场的发展，特别是二十世纪九十年代以后，数据管理不再仅仅是存储和管理数据，而转变成用户所需要的各种数据管理的方式。数据库有很多种类型，从最简单的存储有各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用。

数据库技术是管理信息系统、办公自动化系统、决策支持系统等各类信息系统的核心部分，是进行科学研究和决策管理的重要技术手段。它存储的是属于某单位的有关数据的集合。使用数据库可以实现数据共享，减少数据冗余和维护废本，支持并发访问。

数据集合具有如下特点：尽可能不重复，以最优方式为某个特定组织的多种应用服务，其数据结构独立于使用它的应用程序，对数据的增、删、改、查由统一软件进行管理和控制。从发展的历史看，数据库是数据管理的高级阶段，它是由文件管理系统发展起来的

一般要为各部门开发一个前台界面，根据所需的功能设计数据库的访问界面。数据库客户端编程主要使用 ODBC API、ADO、ADO.NET、OCI、OTL 等方法；数据库服务端编程主要使用 OLE DB 等方法。数据库编程需要掌握一些访问数据库技术方法，还需要注意怎么设计高效的数据库、数据库管理与运行的优化、数据库语句的优化。

2. 关系型数据库

关系数据库，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示。关系模型是由埃德加·科德于 1970 年首先提出的，并配合“科德十二定律”。现如今虽然对此模型有一些批评意见，但它还是数据存储的传统标准。标准数据查询语言 SQL 就是一种基于关系数据库的语言，这种语言执行对关系数据库中数据的检索和操作。关系模型由关系数据结构、关系操作集合、关系完整性约束三部分组成。

简单说，关系型数据库是由多张能互相联接的二维行列表格组成的数据库。

3. MS SQL Server 与 ODBC 数据库连接

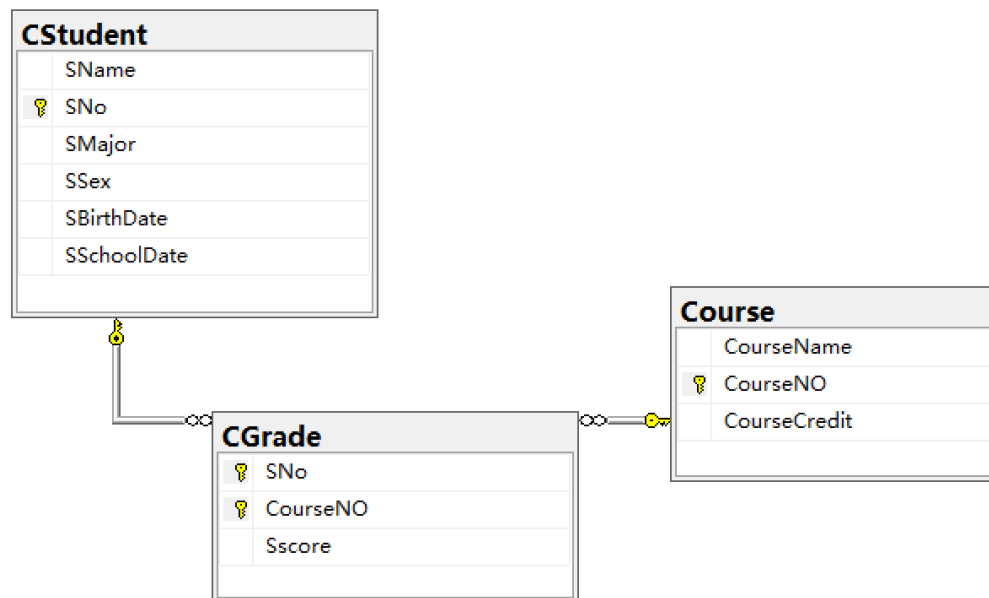
SQL Server 是一个关系数据库管理系统，由微软等公司共同开发，具有使用方便可伸缩性好与相关软件集成程度高等优点，本次实验使用的 SQL Server 2017。

开放数据库连接（Open Database Connectivity，ODBC）是为了解决异构数据库间的数据共享而产生的，是 Windows 环境的一种数据库访问接口标准 ODBC 为异构数据库访问提供统一接口，允许应用程序以 SQL 为数据存取标准，存取不同 DBMS 管理的数据；使应用程序直接操纵 DB 中的数据，免除随 DB 的改变而改变。用 ODBC 可以访问各类计算机上的 DB 文件，甚至访问如 Excel 表和 ASCII 数据文件这类非数据库对象。

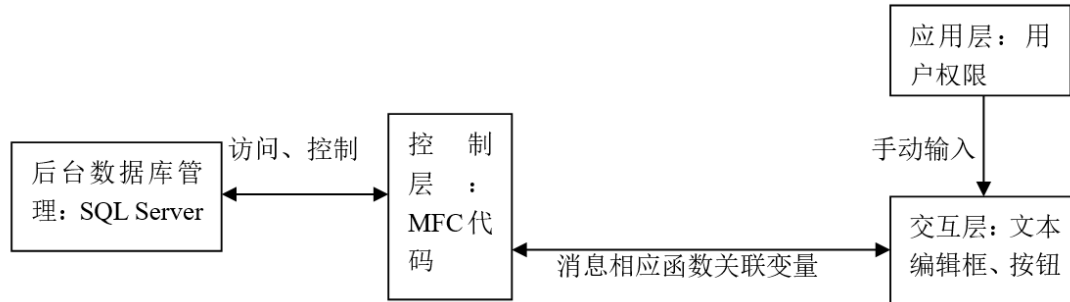
三、程序的开发思路与结构框图

1. 数据库关系表的设计

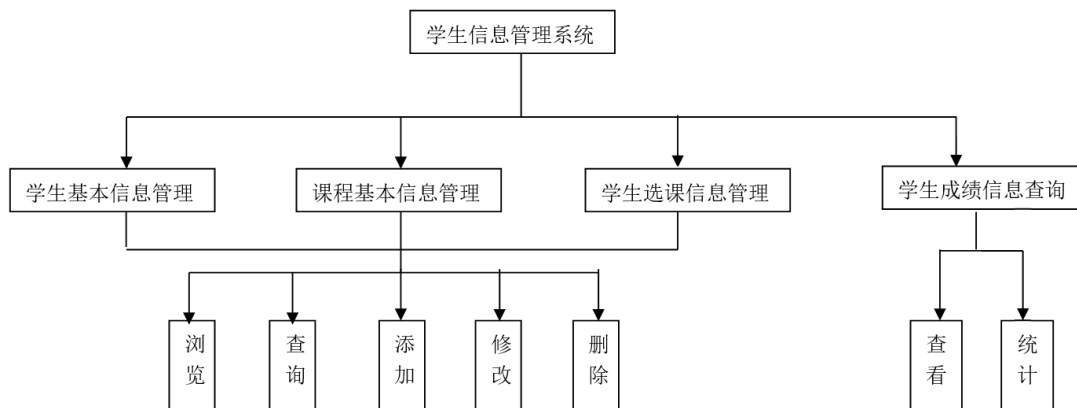
使用 PowerDesigner 软件绘制出数据库的 PDM 和 CDM 图，导出 SQL 脚本，在 SQL Server 中创建关系表，如图所示：



2. 系统框架图



3. 对话框结构设计图



4. 实现思路

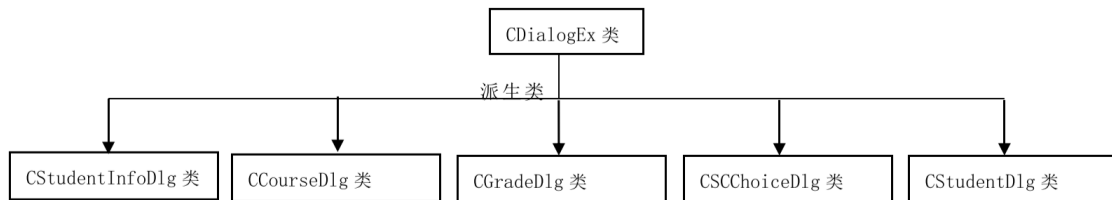
分为 MFC 对话框部分，连接部分和数据库部分。

根据功能直接操纵对话框添加控件，进而设计相应函数，调用相应类来实现功能，然后通过调用 ODBC 库中的函数来建立与 MS SQL Server 数据库的连接。

四、主要函数功能和算法分析

1. 对话框的类关系图

所有的对话框都是 CDialogEx 类的派生



CCourseDlg 对应课程信息的对话框的类。

CGradeDlg 对应着学生成绩查询的对话框的类。

CSCChoiceDlg 对应着选课信息的对话框的类。

CStudentDlg 对应着学生信息的对话框的类。

2. 学生信息的删除操作

a) 算法基本思想

这个操作我认为课本上写的不对。我在原有的基础上，修改了课本上的算法，使程序能够正确编译和运行。应该先查找到要删除学生所选的课程，然后删除他的所有选课信息，最后再删除这个学生的信息。

b) 时间复杂度 $O(n)$

与学生的选课数目有关

c) 代码（注释的代码为我认为书上不正确的地方）

```
void CStudentDlg::OnBnClickedBnDelete()
{
    // TODO: 在此添加控件通知处理程序代码
    //if (AfxMessageBox("震惊！ 是否要删除此条记录？ "), MB_YESNO != IDYES)
        //return;
```

```

        if (AfxMessageBox("震惊! 是否要删除该记录? ", MB_YESNO |
MB_ICONSTOP) == IDYES)
        {
            ChoiceSet.m_strFilter = "SNO='" + m_rsStudentSet.m_SNo + "'";
            ChoiceSet.Requery();
            ChoiceSet.MoveFirst();
            while (!ChoiceSet.IsEOF())
            {
                ChoiceSet.Delete();
                ChoiceSet.MoveNext();
            }

            m_rsStudentSet.Delete();
            m_rsStudentSet.Requery();
            if (m_rsStudentSet.IsBOF())
            {
                AfxMessageBox("什么? 表中居然已经没有记录了! ");
                ClearEditData();
                m_bEmpty = true;
                SetMoveBNState();
            }
            m_bFirst = true;
            m_bLast = false;
            ReadRecord();
            SetMoveBNState();
        }
    }
}

```

3. 查询算法

a) 基本算法思想

点击查询按钮完成查询功能，连接数据库，使用数据库连接类中的 `m_strFilter()` 函数编写 SQL 语句进行查询。

b) 时间复杂度：未知（调用 MFC 中的函数，无法知道他的时间复杂度）

c) 代码

```
void CCourseDlg::OnBnClickedBnQuery()
{
    // TODO: 在此添加控件通知处理程序代码
    UpdateData(TRUE);
    if (CourseNO_Q.IsEmpty() && CourseName_Q.IsEmpty())
    {
        CourseSet.m_strFilter = "";
        CourseSet.Requery();
        if (CourseSet.IsBOF())
        {
            AfxMessageBox("数据集空! ");
            m_bEmpty = true;
            ClearEditData();
            SetMoveBNState();
        }
        else
        {
            m_bEmpty = false;
            CourseSet.MoveFirst();
            m_bFirst = true;
            m_bLast = false;
            ReadRecord();
            SetMoveBNState();
        }
        return;
    }
    BOOL mbSNoQInput = FALSE;
    if (!CourseNO_Q.IsEmpty())
    {
```

```

        CourseSet.m_strFilter = "CourseNO=" + CourseNO_Q;
        CourseSet.m_strFilter = CourseSet.m_strFilter + "";

        mbSNoQInput = TRUE;
    }
    if (!CourseName_Q.IsEmpty())
    {
        if (mbSNoQInput)
        {
            CourseSet.m_strFilter = CourseSet.m_strFilter + " AND
CourseName=";
            CourseSet.m_strFilter = CourseSet.m_strFilter +
CourseName_Q;
            CourseSet.m_strFilter = CourseSet.m_strFilter + "";
        }
        else
        {
            CourseSet.m_strFilter = "CourseName=" + CourseName_Q;
            CourseSet.m_strFilter = CourseSet.m_strFilter + "";
        }
    }
    CourseSet.Requery();
    if (CourseSet.IsEOF())
    {
        AfxMessageBox("没有查到相关记录! ");
        m_bEmpty = true;
        ClearEditData();
        SetMoveBNState();
    }
    else
    {
        m_bEmpty = false;
    }

```

```

        CourseSet.MoveFirst();
        m_bFirst = true;
        ReadRecord();
        SetMoveBNState();
    }
}

```

4. 设置“滚动记录指针”按钮的使用状态

a) 算法基本思想

根据记录集是否为空，记录指针的位置及对数据库的操作类型，来设置“滚动记录指针”按钮的使用状态。

b) 时间复杂度 $O(1)$

c) 代码

```

void CStudentDlg::SetMoveBNState(void)
{
    CWnd *pWnd;
    if (m_bEmpty || m_bEdit)
    {
        pWnd = GetDlgItem(IDC_BN_FIRST);
        pWnd->EnableWindow(FALSE);

        pWnd = GetDlgItem(IDC_BN_PREV);
        pWnd->EnableWindow(FALSE);

        pWnd = GetDlgItem(IDC_BN_NEXT);
        pWnd->EnableWindow(FALSE);
    }
    if (m_bFirst)
    {
        pWnd = GetDlgItem(IDC_BN_FIRST);
    }
}

```

```
pWnd->EnableWindow(FALSE);

pWnd = GetDlgItem(IDC_BN_PREV);
pWnd->EnableWindow(FALSE);

pWnd = GetDlgItem(IDC_BN_LAST);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_BN_NEXT);
pWnd->EnableWindow(TRUE);
}
else if (m_bLast)
{
    pWnd = GetDlgItem(IDC_BN_FIRST);
    pWnd->EnableWindow(TRUE);

    pWnd = GetDlgItem(IDC_BN_PREV);
    pWnd->EnableWindow(TRUE);

    pWnd = GetDlgItem(IDC_BN_LAST);
    pWnd->EnableWindow(FALSE);

    pWnd = GetDlgItem(IDC_BN_NEXT);
    pWnd->EnableWindow(FALSE);
}
else
{
    pWnd = GetDlgItem(IDC_BN_FIRST);
    pWnd->EnableWindow(TRUE);

    pWnd = GetDlgItem(IDC_BN_PREV);
    pWnd->EnableWindow(TRUE);
```

```
pWnd = GetDlgItem(IDC_BN_LAST);  
pWnd->EnableWindow(TRUE);  
  
pWnd = GetDlgItem(IDC_BN_NEXT);  
pWnd->EnableWindow(TRUE);  
}  
}
```

5. 设置编辑框控件的可编辑状态

a) 算法基本思想

根据对数据库的操作类型设置编辑框控件的可编辑状态

b) 时间复杂度分析 $O(1)$

c) 代码

```
void CCourseDlg::SetEditState(void)  
{  
    CWnd *pWnd;  
    if (m_bEdit)  
    {  
        pWnd = GetDlgItem(IDC_ADD);  
        pWnd->EnableWindow(FALSE);  
  
        pWnd = GetDlgItem(IDC_BN_EDIT);  
        pWnd->EnableWindow(FALSE);  
  
        pWnd = GetDlgItem(IDC_BN_DELETE);  
        pWnd->EnableWindow(FALSE);  
  
        pWnd = GetDlgItem(IDC_BN_QUERY);  
        pWnd->EnableWindow(FALSE);  
    }  
}
```

```
pWnd = GetDlgItem(IDC_BN_SAVE);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_BN_CANCEL);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_CNO);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_EDIT_CNAME);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_CREDIT);
pWnd->EnableWindow(TRUE);

}
else
{

pWnd = GetDlgItem(IDC_ADD);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_BN_EDIT);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_BN_DELETE);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_BN_QUERY);
pWnd->EnableWindow(TRUE);

pWnd = GetDlgItem(IDC_BN_QUERY);
pWnd->EnableWindow(TRUE);
```

```
pWnd = GetDlgItem(IDC_BN_SAVE);
pWnd->EnableWindow(FALSE);

pWnd = GetDlgItem(IDC_BN_CANCEL);
pWnd->EnableWindow(FALSE);

pWnd = GetDlgItem(IDC_CNO);
pWnd->EnableWindow(FALSE);

pWnd = GetDlgItem(IDC_EDIT_CNAME);
pWnd->EnableWindow(FALSE);

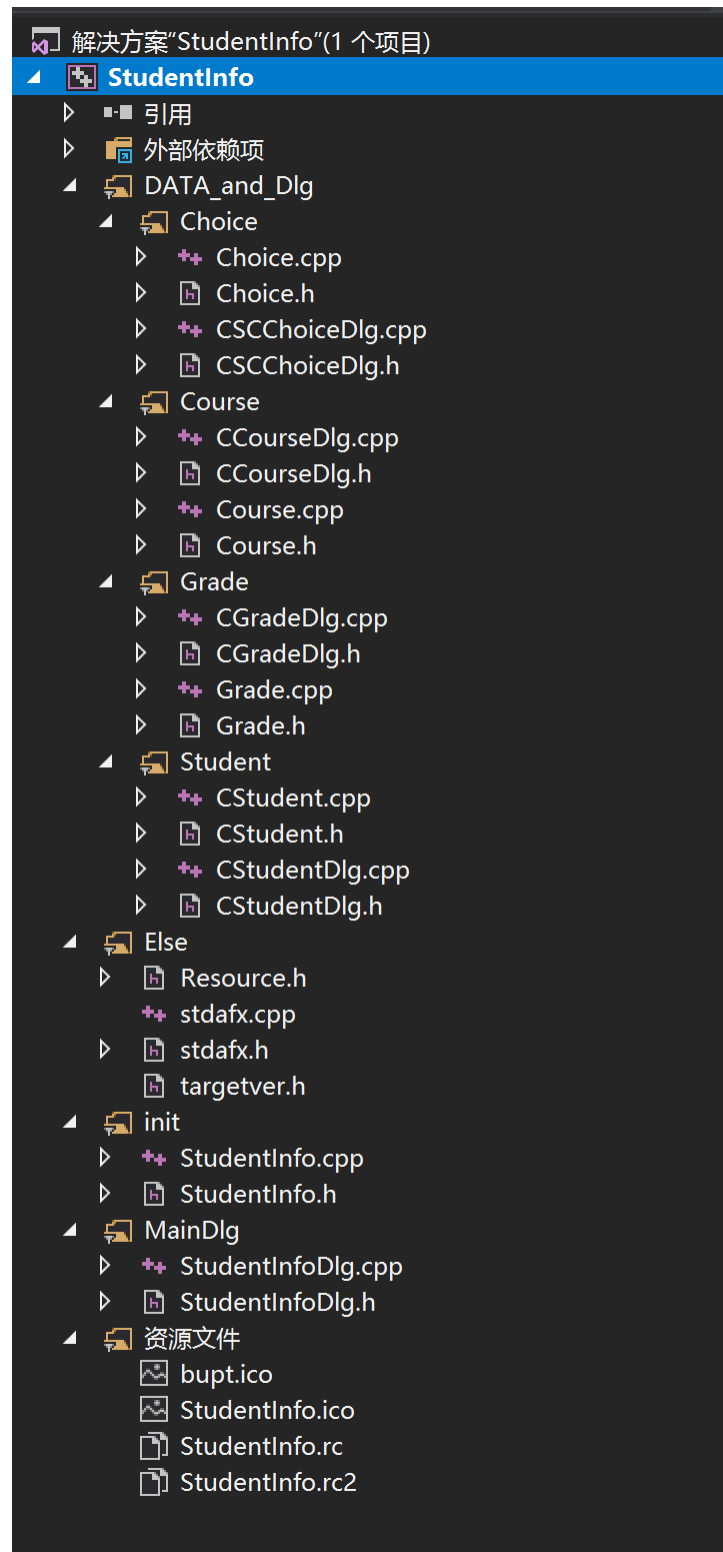
pWnd = GetDlgItem(IDC_CREDIT);
pWnd->EnableWindow(FALSE);

    }
}
```

五、总体程序实现功能说明与展示

1. 程序代码组织结构图

文件筛选器名称	功能
DATA_and_Dlg	用于存 4 个对话框实现的代码与各自数据库连接的代码
Else	存放 MFC 和 windows 应用程序初始化所需要的头文件和代码
Init	存放该 exe 程序初始化的代码
MainDlg	存放最开始的对话框的实现代码
资源文件	存放图标、对话框大小位置等信息等资源文件



2. 主界面

我更改了图标，换成了北邮的 Logo



3. 学生信息

学生基本信息

学号 姓名

学号 姓名

性别 专业

出生日期 入学日期

4. 课程信息

课程基本信息

×

欢迎使用学生信息管理系统

课程号

课程名

查询

课程号

001

第一条

课程名

通信原理

上一条

学分

5

下一条

末一条

添加

修改

删除

保存

取消

退出

5. 选课信息

选课信息

×

欢迎使用学生信息管理系统

学号

课程号

查询

学号

2016210318

第一条

课程号

001

上一条

成绩

100

下一条

末一条

添加

修改

删除

保存

取消

退出

6. 成绩信息

课程基本信息

×

欢迎使用学生信息管理系统

查询输入

请输入学号:

成绩显示

姓名	课程号	课程名	学分	成绩
张阳	003	数据库	2	100
张阳	002	程序设计实践	3	100
张阳	001	通信原理	5	100

统计信息

选课门数: 总学分: 平均成绩:

7. 关于

学生信息管理系统

×

学生信息

课程信息

选课信息


成绩信息

退出

关于

关于 StudentInfo

×

 StudentInfo, 版本 1.0
版权所有 (C) 2018

确定

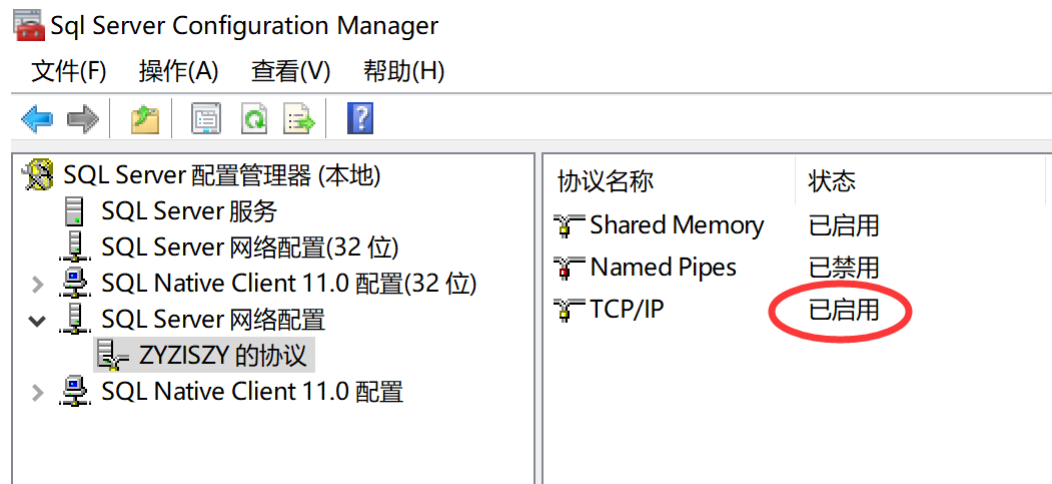
六、故障与问题分析

1. 删除故障

在删除一个学生信息的时候可能会由于数据库表之间的冲突无法删除。因此应该把代码更改一下（我认为书上在删除函数初的代码不正确）我在书上原有代码的基础上，修改了课本上的算法，使程序能够正确编译和运行。应该先查找到要删除学生所选的课程，然后删除他的所有选课信息，最后再删除这个学生的信息。类似的应用与课程信息。

2. 数据库连接问题

数据库 sql server 没有正确设置，在网上查阅资料后，应该在 SQL Server 管理器中启用 TCP/IP 协议，更改 IPALL 端口。





3. 编译和链接出错

没有正确的设置头文件引用，今后应该多了解 makefile 与 cmake 等编译方法，在不同的平台下编写 C++，增加自己对大型工程文件的管理组织能力。

4. 字符编码问题

在编译过程中，`AfxMessageBox("数据打开失败!")`;这样的消息对话框函数可能会导致无法正确编译，在请教老师和查阅资料后，我发现在项目属性中这样设置可以解决问题

启用托管增量生成	否
▼ 项目默认值	
配置类型	应用程序(.exe)
MFC 的使用	在共享 DLL 中使用 MFC
字符集	使用多字节字符集
公共语言运行时支持	无公共语言运行时支持
.NET 目标框架版本	
平台程序集	不同选项

七、总结与心得体会

1. 编程的难点

- 本程序需要掌握 MFC 实现基本电话本功能设计的前端界面，前端与数据库的连接与同步，用 ODBC 库连接 MS SQL Server 实现 MFC 学生管理系统程序导出到数据库，并实现同步的数据添加及删除等的机制，对于只学习了简单数据库只是及基本的 MFC 界面设计的我们来说是难点也是重点，通过此程序的训练，现已进本了解链接的机制；
- 另一难点在于各个类与对话框资源之间的继承，与调用对象关系，通过这个程序很好的锻炼了自己面向对象编程的能力。

2. 我的收获

- 学习了 SQL server 的使用和 MFC 界面的设计。
- 增加了对数据库的理解，加深了对程序设计的设计模式的理解。
- 锻炼了 C++编程能力，对虚函数，类的派生等 C++语言特性有了更多的应用经验。

3. 我的体会

- 数据库的设计和使用是有着很大的学问，如何设计与部署一个安全、稳定、高效的数据库还需要在未来投入大量的时间去学习。通过这个实验我对数据库的设计有了强烈的兴趣，如果有机会，我未来想要从事分布式数据库的开发与部署。
- 本次实验前前后后写了将近快 3500 行代码，充分练习与使用了 C++中的虚函数、重载运算符、面向对象等特性，同时也更加牢固的掌握了 C++这门强大的程序设计语言。
- 我学习并使用了分布式版本控制系统 Git 以应对大量的代码编写，在编程的过程中出现种种问题，比如电脑死机等，在一些时候想恢复到曾经的代码，因此在本次编程

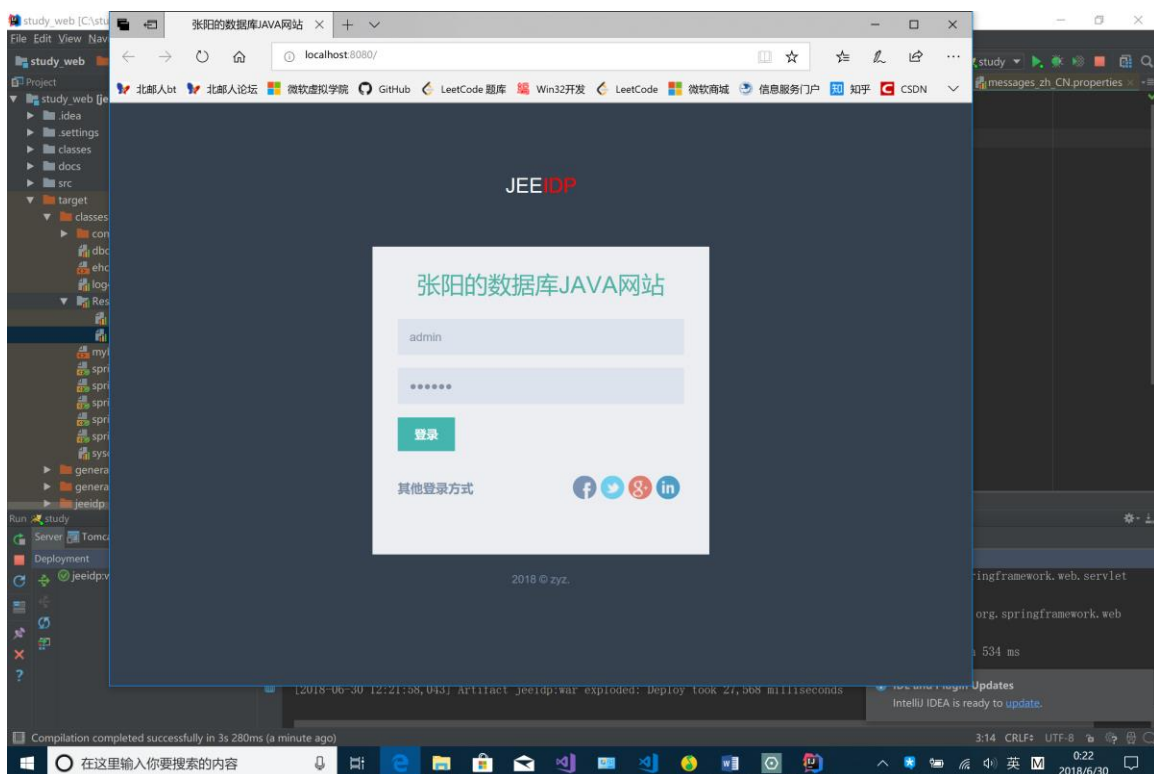
中，在每次上完课时都及时 Git，并把它们备份到我的 GitHub 网页上。

(<https://github.com/zyzisy/Programming-Practice>)

- d) 通过这次开发，使我对 Git 命令更加熟悉，也记住了很多 window 的 CMD 命令，同时更重要的是我了解与学到了很多软件工程的思想。

八、未来的改进

1. 增加登陆功能，不同的管理员有不同的权限
2. Windows 客户端已经不是如今数据库操作界面的主流，C++受限于语言特性，无法快速的完成软件的开发，今后我希望能用 java 写一个动态 web 界面，实现数据库的操作。
3. 将数据库配置到自己的网络服务器中，不再受限于本地，
4. 上述功能我已经使用 java 完成了一部分，如图所示



选修管理

选修管理

+ 新建

查询 重置

#	学生姓名	课程名	成绩	操作
1	关羽	高数	80.0	编辑 删除
2	关羽	英语	64.0	编辑 删除
3	关羽	计算机基础	78.0	编辑 删除
4	黄盖	高数	56.0	编辑 删除
5	黄盖	英语	85.0	编辑 删除
6	黄盖	计算机基础	90.0	编辑 删除
7	张合	高数	68.0	编辑 删除
8	张合	英语	58.0	编辑 删除

共 8 条记录, 当前显示 1 / 1 页

2018 © zyz

九、参考资料

1. 《程序设计实践》北京邮电大学出版社；刘瑞芳等著
2. 《数据库系统概论》高等教育出版社；王珊等著
3. 《C++ primer》
4. 《Effective C++》
5. 百度文库
6. 百度百科