**软件项目案例设计**

1. 数据库设计：
2. **课程信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程信息:course | | | | |
| 序号 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | id | int | 10 | 流水号（自动增长），主键 |
| 2 | number | char | 10 | 课程号，唯一 |
| 3 | cname | varchar | 50 | 课程中文名，唯一 |
| 4 | score | int | 3 | 学分 |
| 5 | chour | int | 3 | 周理论学时 |
| 6 | lhour | int | 3 | 周实验学时 |
| 7 | tchour | int | 3 | 理论总学时 |
| 8 | tlhour | int | 5 | 实验总学时 |
| 9 | teacherid | int | 10 | 外键（teacherid [id]） |

1. **班级信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 班级信息:class | | | | |
| 序号 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | id | int | 10 | 流水号（自动增长），主键 |
| 2 | name | varchar | 50 | 班级名称，唯一 |
| 3 | Maior | varchar | 50 | 专业名称 |

1. **学生信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生信息:student | | | | |
| 序号 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | id | int | 10 | 流水号（自动增长），主键 |
| 2 | scode | char | 10 | 学号，唯一 |
| 3 | name | varchar | 50 | 姓名 |
| 4 | gender | char | 2 | 性别，取值{男，女} |
| 5 | zhuangtai | Int | 1 | 账号状态，默认为0 |
| 6 | class\_id | Int |  | 外键（class [id]） |
| 7 | Pwd | vachar | 50 | 密码，默认为123 |

1. **教师信息**

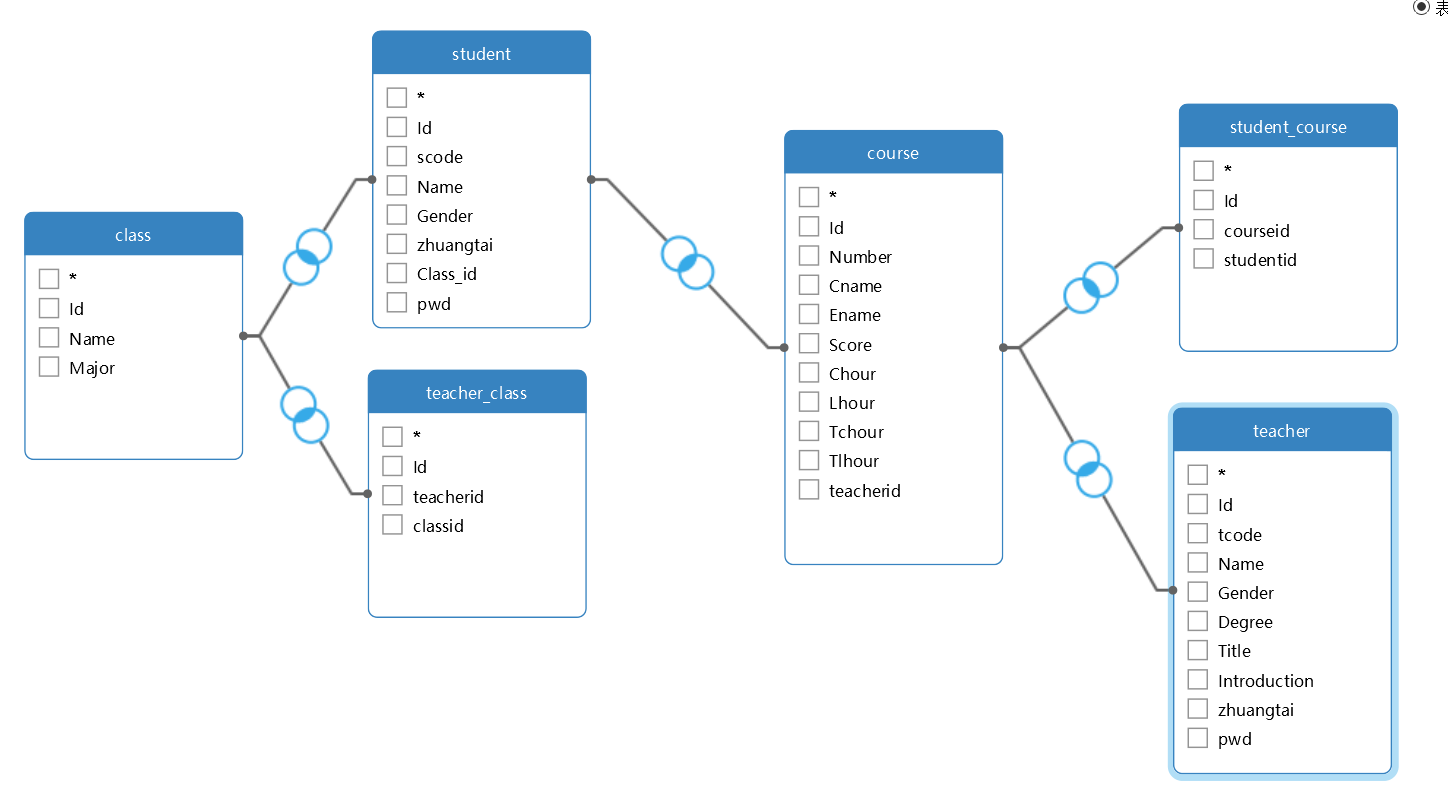
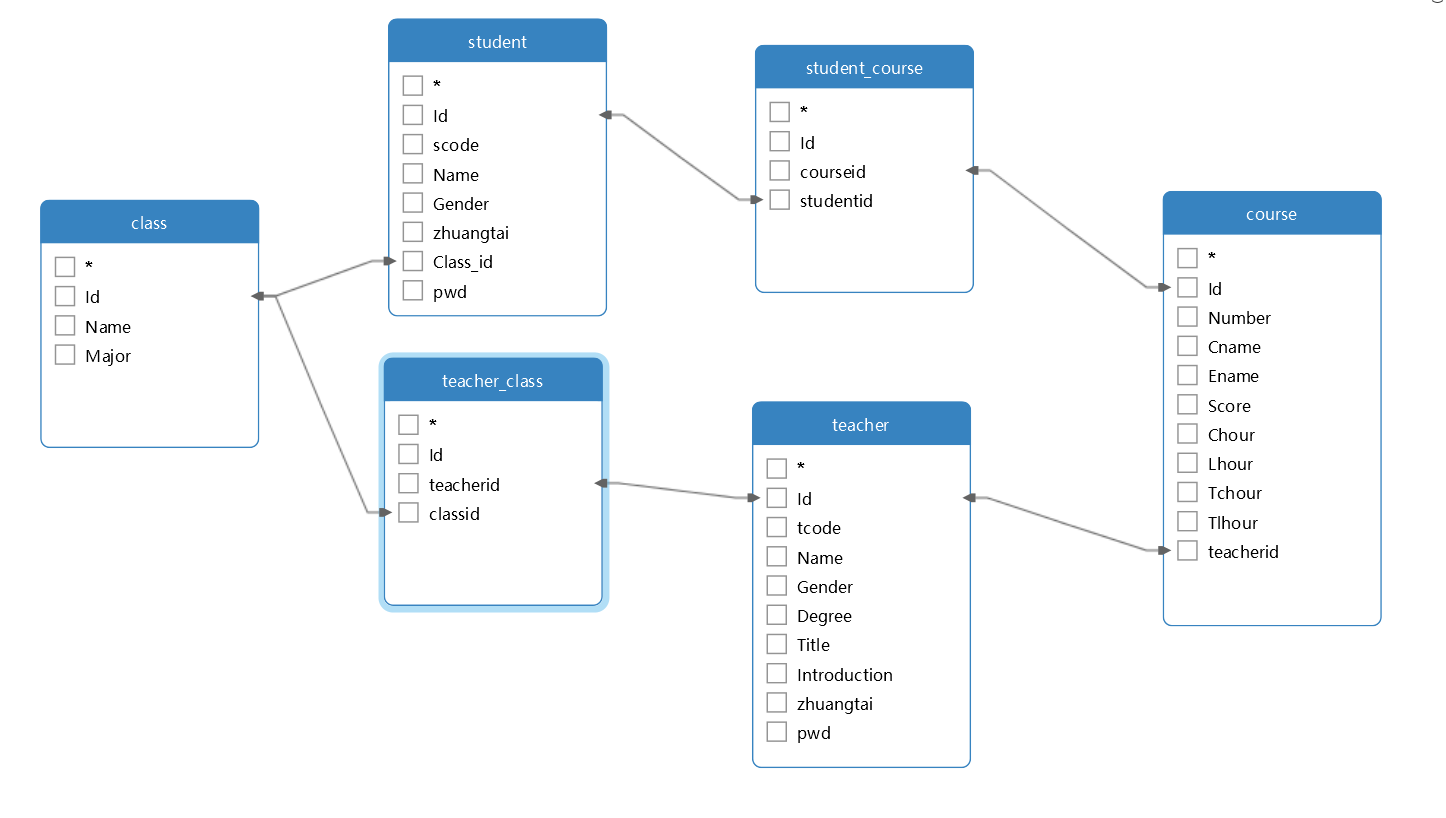
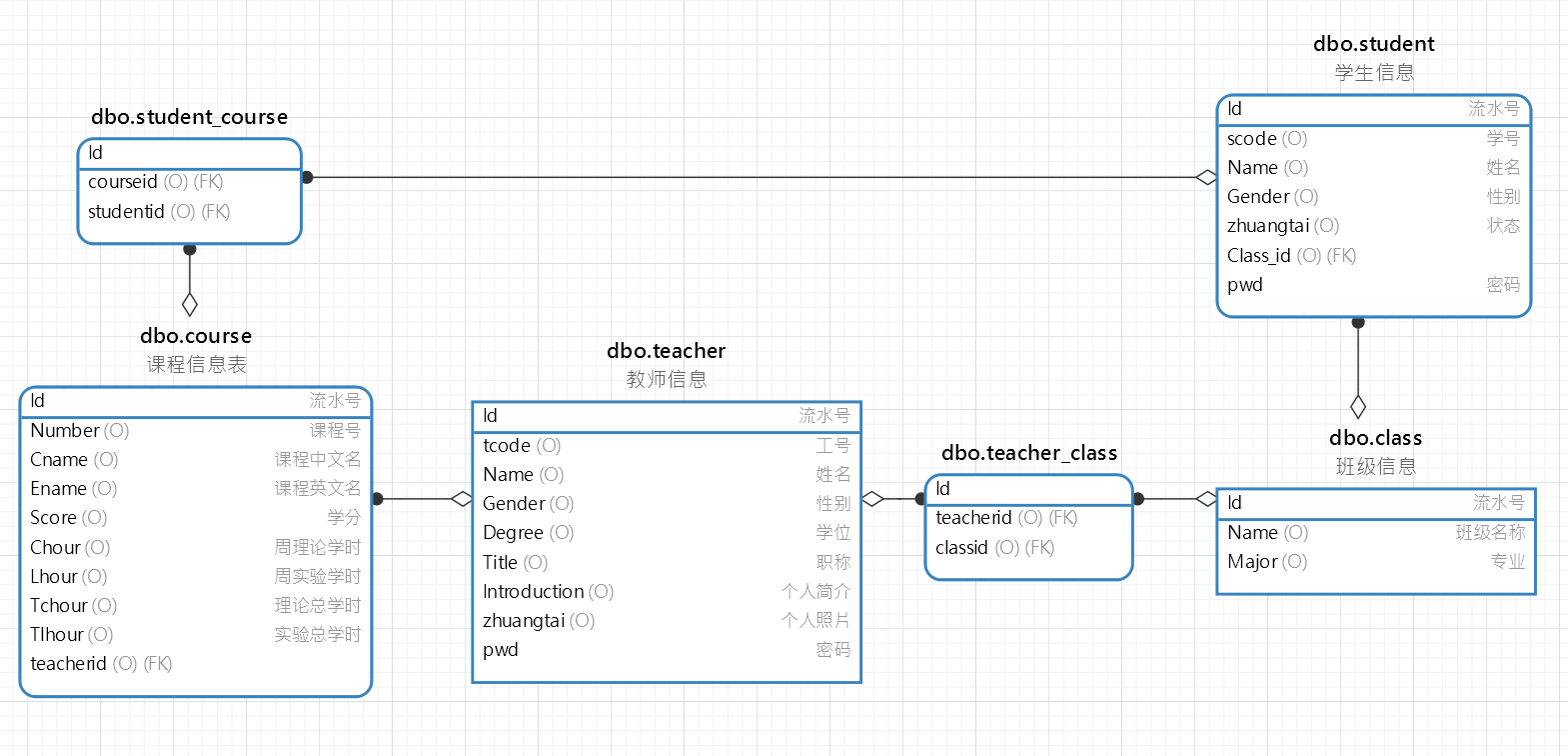
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 教师信息:teacher | | | | |
| 序号 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | id | int | 10 | 流水号（自动增长），主键 |
| 2 | tcode | char | 10 | 工号 |
| 3 | name | varchar | 50 | 姓名 |
| 4 | gender | char | 2 | 性别 |
| 5 | degree | varchar | 10 | 学位 |
| 6 | title | varchar | 10 | 职称 |
| 7 | introduction | varchar |  | 个人简介 |
| 8 | Zhuangtai | Int | 1 | 账号状态，默认为0 |
| 9 | Pwd | vachar | 50 | 密码，默认为123 |

1. **学生选课+成绩信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生选课信息:student\_course | | | | |
| 序号 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | id | int | 10 | 流水号（自动增长），主键 |
| 2 | Courseid | Int | 10 | 外键（course [id]） |
| 3 | studentid | varchar | 50 | 外键（student [id]） |
| 4 | Grade | varchar | 100 | 记录学生成绩 |

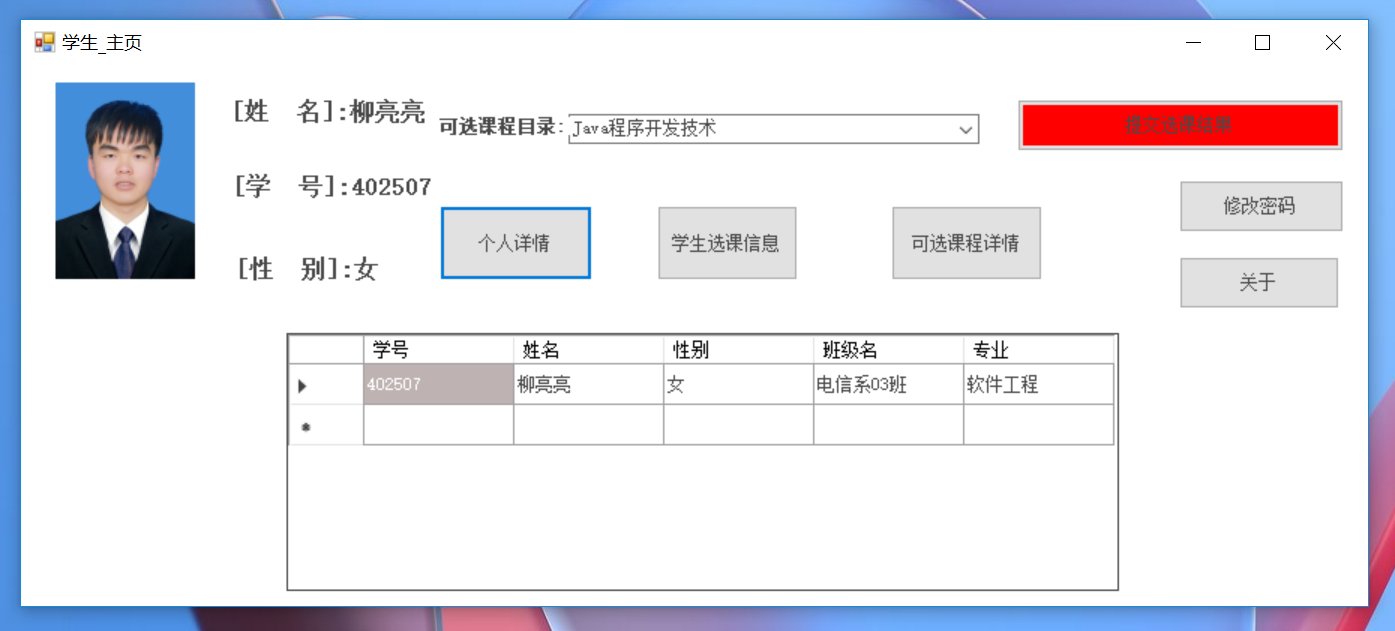
1. **教师班级信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 教师班级信息:teacher\_class | | | | |
| 序号 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | id | int | 10 | 流水号（自动增长），主键 |
| 2 | Teacherid | Int | 10 | 外键（teacher[id]） |
| 3 | Classid | varchar | 50 | 外键（class [id]） |

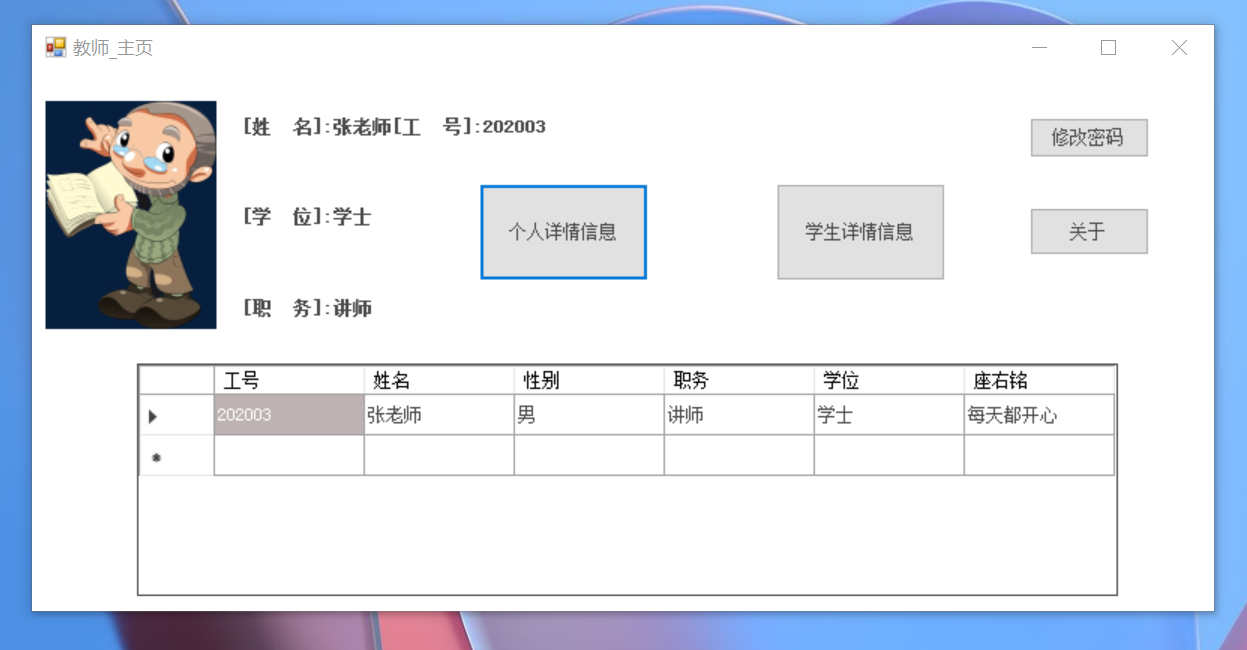
1. **数据库关系图**
2. **数据库关系图**
3. **字段关系图**
4. **数据库设计E-R图**
5. 程序设计
6. **主要界面展示**
7. **主界面**



1. **学生首页**



1. **教师首页**



1. **主要代码**
2. **主界面(摘要)**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**new teacherindex().Show();**

**this.Hide();**

**}**

**private void button2\_Click(object sender, EventArgs e)**

**{**

**new studentindex().Show();**

**this.Hide();**

**}**

1. **学生首页(摘要)**

**private void QueryAllCourse1()**

**{**

**string sqlStr = "SELECT DISTINCT [student].scode AS '学号',[student].Name AS '姓名',[student].Gender AS '性别',[class].Name AS '班级名',[class].Major AS '专业',teacher.Name AS '教师',course.Cname AS '课程' FROM [student],[CLASS],[course],[student\_course],[teacher],[teacher\_class] WHERE [student].Class\_id=[class].Id AND course.teacherid=teacher\_class.teacherid AND teacher.ID=teacher\_class.teacherid AND student\_course.studentid=student.Id AND student\_course.courseid=course.id AND student.Scode=" + student.SCode;**

**DataTable dataTable = BaseDao<object>.FindDataTable(sqlStr);**

**dataGridView1.DataSource = null;**

**dataGridView1.Columns.Clear();**

**if (dataTable.Rows.Count > 0){**

**//dataTable.Rows[1].Delete();**

**//将数据集合的首张表绑定到dataGridView1的数据源**

**this.dataGridView1.DataSource = dataTable;}}**

**private void QueryAllCourse3(){**

**string sqlStr = "SELECT Cname AS 课程名称,Ename AS 语言,score AS 学分,chour AS周理论学时,Lhour AS 周实验学时,Tchour AS 理论总学时,Tchour AS 实验总学时,teacher.Name AS 教师 FROM [dbo].[course],teacher WHERE course.teacherid=teacher.id";**

**DataTable dataTable = BaseDao<object>.FindDataTable(sqlStr);**

**if (dataTable != null){**

**//dataTable.Rows[1].Delete();**

**//将数据集合的首张表绑定到dataGridView1的数据源**

**this.dataGridView3.DataSource = dataTable;}**

**else if (dataTable == null){**

**System.Windows.Forms.MessageBox.Show("首次使用");}}**

**private void QueryAllCourse2(){**

**string sqlStr = "SELECT DISTINCT [student].scode AS '学号',[student].Name AS '姓名',[student].Gender AS '性别',[class].Name AS '班级名',[class].Major AS '专业' FROM [student],[CLASS],[course],[student\_course],[teacher],[teacher\_class] WHERE [student].Class\_id=[class].Id AND student.Scode=" + student.SCode;**

**DataTable dataTable = BaseDao<object>.FindDataTable(sqlStr);**

1. **数据库连接关键部分(摘要)**

**protected static String connStr = "Data Source=47.110.139.115;Initial Catalog=s\_t;Persist Security Info=True;User ID=sa;Password=19990206lll".ToString();**

**protected static SqlConnection conn = null;**

1. **数据库查询方法(摘要)**

**public StudentBean FindBySCode(string sCode,string pwd){**

**StudentBean studentBean = null;**

**try{**

**//连接数据库**

**conn = new SqlConnection(connStr); //数据库连接对象**

**conn.Open();**

**//SQL**

**string sqlStr = "SELECT \* FROM [student] WHERE sCode=@sCode and pwd=@pwd";//占位符**

**SqlCommand cmd = new SqlCommand(sqlStr, conn);**

**cmd.Parameters.AddWithValue("@sCode", sCode);**

**cmd.Parameters.AddWithValue("@pwd", pwd);**

**DataTable dt = new DataTable();**

**new SqlDataAdapter(cmd).Fill(dt);**

**//dataTable转模型**

**studentBean = ConvertToModel(dt);**

**return studentBean;}**

**catch (Exception ex){**

**Console.WriteLine("【Execption】" + ex.Message.ToString());**

**return studentBean;}**

1. **主要数据库查询语句**
2. **查询学生选课信息**

**语句：**

**SELECT DISTINCT**

**[student].scode AS '学号',**

**[student].Name AS '姓名',**

**[student].Gender AS '性别',**

**[class].Name AS '班级名',**

**[class].Major AS '专业',**

**teacher.Name AS '教师',**

**course.Cname AS '课程'**

**FROM**

**[student],**

**[CLASS],**

**[course],**

**[student\_course],**

**[teacher],**

**[teacher\_class]**

**WHERE**

**[student].Class\_id= [class].Id**

**AND course.teacherid= teacher\_class.teacherid**

**AND teacher.ID= teacher\_class.teacherid**

**AND student\_course.studentid= student.Id**

**AND student\_course.courseid= course.id**

**AND student.Scode=student.Scode**

**结果：**

1. **查看课程被选结果**

**语句：**

**SELECT DISTINCT**

**class.name AS 班级,**

**class.Major AS 专业,**

**student.name AS 学生姓名,**

**student.scode AS 学生学号,**

**student.Gender AS 学生性别**

**FROM**

**[dbo].[teacher\_class],**

**class,**

**teacher,**

**student**

**WHERE**

**teacher\_class.classid= class.id**

**AND class.id= student.Class\_id**

**AND teacher\_class.teacherid= teacher.id**

**AND teacher.id= student.Id**

**结果：**

1. 软件测试

**3.1测试项目名称及测试内容**

|  |  |
| --- | --- |
| **测试项目名称** | **测试内容** |
| **登录功能** | **用户能否正常输入用户名、密码和正常登录** |
| **选课功能** | **用户能否正常选课** |
| **查看功能** | **用户能否查看已选课程** |

**3.2测试用例**

1. **用户登陆模块测试**

|  |  |
| --- | --- |
| **测试用例编号** | **Test\_001** |
| **测试项目** | **用户登陆** |
| **测试标题** | **输入已有的注册账号** |
| **重要级别** | **高** |
| **预置条件** | **进入系统** |
| **输入** | **1234（密码）** |
| **操作步骤** | **1.输入空的账号，2.输入密码1234** |
| **预期输出** | **系统提示注册学号和密码不能为空** |

1. **用户退出模块测试**

|  |  |
| --- | --- |
| **测试用例编号** | **Test\_0022** |
| **测试标题** | **用户退出系统** |
| **重要级别** | **高** |
| **预置条件** | **进入系统** |
| **输入** |  |
| **操作步骤** | **点击退出按钮** |
| **预期输出** | **退出系统** |

1. **用户查看课程信息模块测试**

|  |  |
| --- | --- |
| **测试用例编号** | **Test\_003** |
| **测试项目** | **用户查看所有课程信息** |
| **测试标题** | **查看课程信息** |
| **重要级别** | **高** |
| **预置条件** | **进入系统** |
| **输入** |  |
| **操作步骤** | **点击查看课程按钮** |
| **预期输出** | **系统显示所有课程信息** |

1. **用户查看课程模块测试**

|  |  |
| --- | --- |
| **测试用例编号** | **Test004** |
| **测试项目** | **查询** |
| **测试标题** | **查询已选课程** |
| **重要级别** | **高** |
| **预置条件** | **进入系统** |
| **输入** | **无** |
| **操作步骤** | **点击查询** |
| **预期输出** | **显示已选择课程** |

四,总结

对于一个系统，数据库的设计是非常重要的，数据库设计决定了以后数据好不好维护。后期需求好不好展。同时也决定了系统的性能。一个坏的数据库设计一个功能点的改动可能会设计多张表的改动。一不小心可能就会引起数据的不一致。为了解决这些问题。在数据库设计之初就要考虑这些问题。

**一、范式与反范式的设计**

范式设计的目的是为了减少数据冗余从而节约存储空间提高查询效率，同时也使得数据一致性容易得到维护。建立学生信息表与课程信息表。当学生进行选课操作时。建立选课表使得学生与课程建立关系。建立关系只需要建立主键ID的关系。

**二、数据库约束**

数据约束最好是在数据设计之初就定义好。包括非空约束数据唯一性约束。这样可以解决很多系统空异常，并且使得数据更好维护。

**四、数据库访问权限界定**

数据库访问权限一定要控制在自己的系统范围内，严禁其他系统直接操作数据库。系统需要与其他系统进行信息交换，或者数据变更，应该采用本系统开发接口给外部系统进行调用并且对接口调用进行日志记录，而不是直接开放数据库给对方。