**PB**

|  |  |
| --- | --- |
| 首先是建立数据库部分，我用的是SQL SERVER,代码如下：   |  | | --- | | -- 创建选课系统数据库  CREATE DATABASE 选课系统数据库;  USE 选课系统数据库;  CREATE TABLE Departments (  DepartmentID INT PRIMARY KEY,  DepartmentName VARCHAR(255) NOT NULL  );  CREATE TABLE Members (  MemberID INT PRIMARY KEY,  Name VARCHAR(255) NOT NULL,  Code VARCHAR(50) NOT NULL,  DepartmentID INT,  Permission INT,  FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)  );  CREATE TABLE Courses (  CourseID INT PRIMARY KEY,  CourseName VARCHAR(255) NOT NULL,  DepartmentID INT,  FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)  );  CREATE TABLE Semesters (  SemesterID INT PRIMARY KEY,  SemesterName VARCHAR(50) NOT NULL  );  CREATE TABLE Classes (  ClassID INT PRIMARY KEY,  SemesterID INT,  CourseID INT,  TeacherID INT,  ClassTime VARCHAR(255),  IsSelectable BIT,  FOREIGN KEY (SemesterID) REFERENCES Semesters(SemesterID),  FOREIGN KEY (CourseID) REFERENCES Courses(CourseID),  FOREIGN KEY (TeacherID) REFERENCES Members(MemberID)  );  CREATE TABLE Enrollments (  ClassID INT,  StudentID INT,  Grade DECIMAL(5,2),  IsEditable BIT,  PRIMARY KEY (ClassID, StudentID),  FOREIGN KEY (ClassID) REFERENCES Classes(ClassID),  FOREIGN KEY (StudentID) REFERENCES Members(MemberID)  );  -- 更新开课表，添加课程容量和已选人数字段  ALTER TABLE Classes  ADD Capacity INT,  EnrolledCount INT;  ALTER TABLE Courses  ADD Credits DECIMAL(5, 2);  -- 更新课程表，添加学分字段  ALTER TABLE Courses  ADD Credits DECIMAL(5, 2);  -- 更新课程数据，为新增字段赋初值  UPDATE Courses  SET Credits = 3.0; -- 假设学分为3.0，你可以根据实际情况调整  -- 更新开课数据，为新增字段赋初值  UPDATE Classes  SET Capacity = 30, -- 假设课程容量为30人  EnrolledCount = 0; -- 初始已选人数为0 | |
| 连接数据库：  导入数据库部分当时没截图，就放一下完成图：  因为我用的是机器身份认证，没用账号密码，所以代码修改如下：   |  | | --- | | // 假设你已经配置了数据库配置文件或有相关的连接参数  string ls\_dbms, ls\_server, ls\_database  boolean ll\_database  // 从数据库配置文件中获取连接参数，或手动设置参数  // 连接数据库  //MessageBox("数据库连接错误", "见鬼" )  //Connect Using SQLCA;  SQLCA.DBMS="ODBC"  SQLCA.AutoCommit = False  SQLCA.DBParm ="ConnectString='DSN=CourseSelect;'"  Connect;  // 检查连接是否成功  if SQLCA.SQLCode <> 0 then  MessageBox("数据库连接错误", "无法连接到数据库: " + SQLCA.SQLErrText)  // 在此处理连接失败的情况，可能需要关闭应用程序或提供用户重新尝试的选项  else  MessageBox("数据库连接成功", "成功连接到数据库")  s\_info = 'root'  sid=1  // 为调试程序，先初始化全局变量 s\_info 为 "S1"  open(w\_login)  // 在此可以执行其他初始化操作  // 添加以下代码    end if | |
| 随后是date文件,我共建立了如下内容：  开课表：    选课表：  开课全表：  开课选择表：  课程全表：  人员全表：  成绩表：  学期表：  学生选择信息表：    学生选择选课信息表：  教师选择选课学生表： |
| 窗口及代码： 开课管理窗口：   |  | | --- | | Open: int sum,hascol  dw\_1.settransobject(sqlca);  dw\_1.retrieve();  hascol=dw\_1.retrieve();  //pb\_2.enabled=false  //st\_1.text=string(hascol) | | // "新建" 按钮(pb\_new) 的 Clicked() 事件脚本  string classIDString, semesterIDString, courseIDString, teacherIDString, departmentIDString, classTime, enrolledCountString, capacityString, isSelectableString  // 从输入框中获取选课信息  classIDString = sle\_1.Text  semesterIDString = sle\_2.Text  courseIDString = sle\_3.Text  teacherIDString = sle\_4.Text  //departmentIDString = sle\_5.Text  classTime = sle\_6.Text  enrolledCountString = sle\_7.Text  capacityString = sle\_8.Text  isSelectableString = sle\_9.Text  // 转换字符串为整数  int classID, semesterID, courseID, teacherID, departmentID, enrolledCount, capacity  boolean isSelectable  classID = Integer(classIDString)  semesterID = Integer(semesterIDString)  courseID = Integer(courseIDString)  teacherID = Integer(teacherIDString)  //departmentID = Integer(departmentIDString)  enrolledCount = Integer(enrolledCountString)  capacity = Integer(capacityString)  isSelectable = Integer(isSelectableString) <> 0  // 检查选课ID是否重复  int ClassCount = 0  SELECT COUNT(\*) INTO :ClassCount FROM Classes WHERE ClassID = :classID;  IF ClassCount > 0 THEN  MessageBox("错误", "已存在相同选课ID，请选择其他选课ID。")  RETURN  END IF  // 插入选课数据  INSERT INTO Classes (ClassID, SemesterID, CourseID, TeacherID, ClassTime, EnrolledCount, Capacity, IsSelectable) VALUES  (:classID, :semesterID, :courseID, :teacherID, :classTime, :enrolledCount, :capacity, :isSelectable);  // 清空输入框  sle\_1.Text = ""  sle\_2.Text = ""  sle\_3.Text = ""  sle\_4.Text = ""  //sle\_5.Text = ""  sle\_6.Text = ""  sle\_8.Text = ""  sle\_9.Text = ""  sle\_7.Text = ""  dw\_1.Reset()  dw\_1.SetTransObject(sqlca)  dw\_1.Retrieve()  // 提示用户操作成功  MessageBox("成功", "选课数据已成功插入数据库。"); | | // "修改" 按钮(pb\_update) 的 Clicked() 事件脚本  string classIDString, semesterIDString, courseIDString, teacherIDString, departmentIDString, classTime, enrolledCountString, capacityString, isSelectableString  // 获取要修改的选课ID和新的选课信息  classIDString = sle\_1.Text  semesterIDString = sle\_2.Text  courseIDString = sle\_3.Text  teacherIDString = sle\_4.Text  //departmentIDString = sle\_5.Text  classTime = sle\_6.Text  enrolledCountString=sle\_7.text  capacityString=sle\_8.text  isSelectableString = sle\_9.Text  // 转换字符串为整数  int classID, semesterID, courseID, teacherID, departmentID, enrolledCount, capacity  boolean isSelectable  classID = Integer(classIDString)  semesterID = Integer(semesterIDString)  courseID = Integer(courseIDString)  teacherID = Integer(teacherIDString)  //departmentID = Integer(departmentIDString)  enrolledCount = Integer(enrolledCountString)  capacity = Integer(capacityString)  isSelectable = Integer(isSelectableString) <> 0  // 检查选课ID是否存在  int ClassCount = 0  SELECT COUNT(\*) INTO :ClassCount FROM Classes WHERE ClassID = :classID;  IF ClassCount = 0 THEN  MessageBox("错误", "指定的选课ID不存在，请选择有效的选课ID。")  RETURN  END IF  // 执行修改选课数据的 SQL 语句  UPDATE Classes SET SemesterID = :semesterID, CourseID = :courseID, TeacherID = :teacherID,Capacity=:capacity,EnrolledCount=:enrolledCount, ClassTime = :classTime, IsSelectable = :isSelectable WHERE ClassID = :classID;  // 清空输入框  sle\_1.Text = ""  sle\_2.Text = ""  sle\_3.Text = ""  sle\_4.Text = ""  //sle\_5.Text = ""  sle\_6.Text = ""  sle\_9.Text = ""  dw\_1.Reset()  dw\_1.SetTransObject(sqlca)  dw\_1.Retrieve()  // 提示用户操作成功  MessageBox("成功", "选课数据已成功修改。"); | | // "删除" 按钮(pb\_delete) 的 Clicked() 事件脚本  string classIDString  // 获取要删除的选课ID  classIDString = sle\_1.Text  // 转换字符串为整数  int classID  classID = Integer(classIDString)  // 检查选课ID是否存在  int ClassCount = 0  SELECT COUNT(\*) INTO :ClassCount FROM Classes WHERE ClassID = :classID;  IF ClassCount = 0 THEN  MessageBox("错误", "指定的选课ID不存在，请选择有效的选课ID。")  RETURN  END IF  // 删除选课数据  DELETE FROM Classes WHERE ClassID = :classID;  // 清空输入框  sle\_1.Text = ""  dw\_1.Reset()  dw\_1.SetTransObject(sqlca)  dw\_1.Retrieve()  // 提示用户操作成功  MessageBox("成功", "选课数据已成功删除。"); | | PB4:  close(parent) |   课程管理窗口：  代码与前者类似，就不贴了。  学期管理系统：  成员管理系统：  以上是管理原能看的；  然后是学生成绩管理系统：   |  | | --- | | dw\_1.SetTransObject(SQLCA)  MessageBox("数据库连接错误", "sid=: " + string(sid))  dw\_1.Retrieve(sid)  dw\_1.Object.DataWindow.ReadOnly = "yes" | | // "修改" 按钮(pb\_update) 的 Clicked() 事件脚本  string classIDString, studentIDString, gradeString  // 获取要修改的成绩信息  classIDString = sle\_1.Text  studentIDString = sle\_2.Text  gradeString = sle\_3.Text  // 转换字符串为整数  int classID, studentID  decimal grade  classID = Integer(classIDString)  studentID = Integer(studentIDString)  grade = Dec(gradeString)  // 检查成绩是否可编辑  int EditableCount = 0  SELECT COUNT(\*) INTO :EditableCount FROM Enrollments WHERE ClassID = :classID AND StudentID = :studentID AND IsEditable = 1;  IF EditableCount = 0 THEN  MessageBox("错误", "成绩不可编辑，无法进行修改。")  RETURN  END IF  // 检查当前用户是否为相应课程的任课老师  int TeacherCount = 0  SELECT COUNT(\*) INTO :TeacherCount FROM Classes WHERE ClassID = :classID AND TeacherID = :sid;  IF TeacherCount = 0 THEN  MessageBox("错误", "您不是该课程的任课老师，无法修改成绩。")  RETURN  END IF  // 更新成绩数据  UPDATE Enrollments SET Grade = :grade WHERE ClassID = :classID AND StudentID = :studentID;  // 清空输入框  sle\_1.Text = ""  sle\_2.Text = ""  sle\_3.Text = ""  dw\_1.Reset()  dw\_1.SetTransObject(sqlca)  dw\_1.Retrieve(sid)  // 提示用户操作成功  MessageBox("成功", "成绩数据已成功修改。"); | | close(parent) |   然后是学生选课部分：   |  | | --- | | dw\_1.SetTransObject(SQLCA)  MessageBox("数据库连接错误", "sid=: " + string(sid))  dw\_1.Retrieve(sid)  dw\_1.Object.DataWindow.ReadOnly = "yes"  dw\_2.SetTransObject(SQLCA)  dw\_2.Retrieve(sid)  dw\_2.Object.DataWindow.ReadOnly = "yes"  dw\_3.SetTransObject(SQLCA)  dw\_3.Retrieve()  dw\_3.Object.DataWindow.ReadOnly = "yes"  dw\_4.SetTransObject(SQLCA)  dw\_4.Retrieve(sid)  dw\_4.Object.DataWindow.ReadOnly = "yes"  sle\_1.SetFocus() | | string ccname,cnn0  int courseid=0,classID = 0  int stuid=0  c\_info = Upper(sle\_1.Text)  classID = Integer(c\_info)  if IsNull(c\_info) then  MessageBox("出错", "请输入课程号!")  else  // 将字符串类型的课程号转为整数    //MessageBox("调试信息", "DBParm: " + SQLCA.DBParm)  //MessageBox("调试信息", "ClassID: " + string(classID) + ", CourseID: " + string(courseid))    // 查询课程名称ClassID  courseid = 0  SELECT Classes.CourseID INTO :courseid FROM Classes WHERE Classes.ClassID = :classID;    int numb=0,capt=0,f=0;  SELECT Classes.EnrolledCount INTO :numb FROM Classes WHERE Classes.ClassID = :classID;  SELECT Classes.Capacity INTO :capt FROM Classes WHERE Classes.ClassID = :classID;  SELECT Classes.IsSelectable INTO :f FROM Classes WHERE Classes.ClassID = :classID;  MessageBox("调试信息", " CourseID: " + string(courseid))  if courseid = 0 then  MessageBox("出错", "此课程号不存在!")  else  // 查询学生是否已选该课程  stuid=0  SELECT StudentID INTO :stuid FROM Enrollments WHERE ClassID =:classID AND StudentID = :sid;  if stuid <> 0 then  MessageBox("出错", "此课程已选!")  else    if numb<capt or f=1 then      //INSERT INTO Enrollments (ClassID, StudentID) VALUES (:classID, :s\_info);  INSERT INTO Enrollments (ClassID, StudentID, Grade, IsEditable) VALUES (:classID, :sid, NULL, 1);  // 更新课程表的已选人数  MessageBox("调试信息", "ClassID: " + string(classID))  UPDATE Classes SET Classes.EnrolledCount = EnrolledCount+1 WHERE ClassID = :classID;  if SQLCA.SQLCode <> 0 then  MessageBox("更新已选人数错误", SQLCA.SQLErrText)  else  MessageBox("更新已选人数成功", "已成功更新已选人数")  end if;    dw\_4.Reset()  dw\_3.Reset()  sle\_1.Text = ""  MessageBox("success", "success")  // 使用正确的 TransObject  dw\_4.SetTransObject(SQLCA)  // 查询学生的选课信息  dw\_4.Retrieve(sid)    dw\_3.SetTransObject(SQLCA)  // 查询学生的选课信息  dw\_3.Retrieve()  else  MessageBox("出错", "此课程已满或不可选课!")  end if  end if  end if  end if | | int ccno=0, ggrade  int classID = 0  c\_info = Upper(sle\_1.Text)  if IsNull(c\_info) then  MessageBox("出错", "请输入课程号!")  else  // 将字符串类型的课程号转为整数  classID = Integer(c\_info)  ccno=0  // 查询学生是否已选该课程  SELECT StudentID, Grade INTO :ccno, :ggrade FROM Enrollments WHERE ClassID = :classID AND StudentID = :sid;  if ccno=0 then  MessageBox("出错", "此课程未选!")  else  if IsNull(ggrade) or ggrade=0 then  // 删除选课记录  DELETE FROM Enrollments WHERE StudentID = :sid AND ClassID = :classID;  // 更新课程表的已选人数  UPDATE Classes SET Classes.EnrolledCount = Classes.EnrolledCount - 1 WHERE ClassID = :classID;  dw\_4.Reset()  dw\_3.Reset()  sle\_1.Text = ""  MessageBox("出错", "success")  // 使用正确的 TransObject  dw\_4.SetTransObject(SQLCA)  // 查询学生的选课信息  dw\_4.Retrieve(sid)    dw\_3.SetTransObject(SQLCA)  // 查询学生的选课信息  dw\_3.Retrieve()  else  MessageBox("出错", "此课程已登分!")  end if  end if  end if | | close(parent) |   最后是登录系统     |  | | --- | | // "登录" 按钮(pb\_login) 的 Clicked() 事件脚本  string memberIDString, code  // 获取用户输入的 MemberID 和 Code  memberIDString = sle\_1.Text  code = sle\_2.Text  // 转换字符串为整数  int memberID  memberID = Integer(memberIDString)  // 验证用户信息  int UserCount = 0  SELECT COUNT(\*) INTO :UserCount FROM Members WHERE MemberID = :memberID AND Code = :code;  IF UserCount = 0 THEN  MessageBox("登录失败", "用户名或密码错误，请重新输入。")  RETURN  END IF  // 获取用户信息和权限  string userName, userCode  int departmentID, permission  SELECT Name, Code, DepartmentID, Permission INTO :userName, :userCode, :departmentID, :permission FROM Members WHERE MemberID = :memberID;  // 保存当前用户信息和权限  currentUserID = memberID  currentUserName = userName  currentUserCode = userCode  currentUserDepartmentID = departmentID  currentUserPermission = permission  sid=currentUserID  // 跳转到主界面或执行其他操作  //open(w\_test)  IF currentUserPermission = 1 THEN  // 执行学生权限的操作，比如打开学生管理窗口  Open(w\_mainselect)  ELSEIF currentUserPermission = 2 THEN  // 执行教师权限的操作，比如打开教师管理窗口  Open(w\_grademanagement)  ELSEIF currentUserPermission = 3 THEN  // 执行管理员权限的操作，比如打开管理员管理窗口  Open(w\_studentmanagement)  Open(w\_semestermanagement)  Open(w\_coursemanagement)  Open(w\_classmanagement)  //Open(w\_grademanagement)  //Open(m\_student)  END IF  // 清空输入框  sle\_1.Text = ""  sle\_2.Text = "" | | close(parent) | |