# 考务管理系统

## 4.1系统功能模块设计

根据需求分析画出如下的功能设计图：

考务管理系统

后台管理

学生模块

教师模块

教学信息

考试公告管理

成绩管理

个人中心

个人中心

查看公告

考试安排查询

成绩查询

给教师留言

教师监考查询

教师学生信息

考务相关

查看留言

查询考试安排

资源状态查询

考场安排

图2

## 4.2数据库表设计

结合功能设计图，根据数据库设计的实体完整性，参照完整性，用户定义完整性原则，从数据库运行效率出发，建立表之间的关联，提高查询速率。 将数据库各表设计如下：

学生表：student

该表设置学生编号sno为主键，方便与班级表class，成绩表sc，角色表role建立关联。具体字段如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| sno | int | 不允许 | 学号，主键 |
| ssex | char(2) | 允许 | 性别 |
| stell | varchar(10) | 允许 | 电话 |
| saddress | varchar(50) | 允许 | 地址 |
| classno | int | 允许 | 班级编号，外键 |
| roleno | int | 允许 | 角色编号 ，外键 |
| spass | varchar(10) | 允许 | 密码 |

课程表：course

该表以课程编号cno作为主键。

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| cno | Int | 不允许 | 课程编号、主键 |
| cname | Varchar(50) | 允许 | 课程名称 |

班级表：class

该表以班级标号classno作为主键，学院编号dno作为外键与学院表进行关联。

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| classno | int | 不允许 | 班级编号、主键 |
| classname | varchar(50) | 允许 | 班级名称 |
| ccount | int | 允许 | 班级人数 |
| dno | int | 允许 | 学院编号，外键 |

课程安排表：planning

该表以班级编号、考场编号作为主键。

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| classno | int | 不允许 | 班级编号、主键之一 |
| cno | int | 不允许 | 课程编号、主键之一 |
| studytime | numeric(4,0) | 允许 | 学时 |

教室表：room

该表以教室编号roomno为key，与考场安排test表关联

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| roomno | int | 不允许 | 教室编号，主键 |
| roomname | varchar(50) | 允许 | 教室名称 |
| seats | numeric(4,0) | 允许 | 座位数 |

教师表：teacher

该表以教师编号tno为key，以学院编号dno、角色编号roleno为外键，与学院表、角色表存在关联。

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| tno | int | 不允许 | 教师编号，主键 |
| tname | varchar(10) | 允许 | 教师名称 |
| tprof | varchar(10) | 允许 | 职称 |
| ttell | varchar(10) | 允许 | 联系电话 |
| tpass | varchar(10) | 允许 | 密码 |
| dno | int | 允许 | 学院编号，外键 |
| roleno | int | 允许 | 角色编号，外键 |

教程安排表：teaching

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| cno | int | 不允许 | 课程编号，主键之一 |
| tno | int | 不允许 | 教师编号，主键之一 |
| createtime | datetime | 允许 | 创建时间 |

成绩表：sc

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| sno | int | 不允许 | 学生编号，主键之一 |
| cno | int | 不允许 | 课程编号，主键之一 |
| score | decimal(4,0) | 允许 | 分数 |

学院表：department

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| dno | int | 不允许 | 学院编号，主键 |
| dleader | varchar(10) | 允许 | 领导名称 |
| dtell | nchar(10) | 允许 | 学院电话 |
| dname | varchar(20) | 允许 | 学院名称 |

学生留言表：message

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| messageno | int | 不允许 | 留言编号，主键 |
| mcontent | nvarchar(50) | 允许 | 留言内容 |
| ntime | datetime | 允许 | 留言时间 |
| tno | int | 允许 | 教师编号，留言时间 |

角色表：role

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| roleno | int | 不允许 | 角色编号，主键 |
| rolecontent | nvarchar | 允许 | 角色说明 |

管理员表：manager

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| mno | int | 不允许 | 管理员编号，主键 |
| mname | varchar(10) | 允许 | 管理员姓名 |
| mpass | varchar(10) | 允许 | 管理员密码 |
| roleno | int | 不允许 | 角色编号，外键 |

公告表：news

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许为空 | 说明 |
| nno | int | 不允许 | 公告编号，主键 |
| ncontent | nvarchar(50) | 允许 | 公告内容 |
| ntitle | varchar(20) | 允许 | 公告标题 |

实体类 ：学生，课程，角色，管理员，学生留言，教师

边界类 ：教室，班级，成绩，学院，公告表

控制类 ：课程安排，教程安排

课程安排

角色

课程

班级

学生留言

学生

管理员

教室

教师

公告表

成绩

学院

教师安排

## 4.3系统主要模块设计

### 4.3.1考场安排模块

该模块主要根据任务陈述中考场安排约束条件进行考场安排。用户只需选择学院，课程及指定一个考试时间，系统便可根据选择的学院，课程，时间进行自动考场安排。具体安排步骤如下：

1. 用户选择一个学院，选择一门课程，指定一个时间（如，数计学院，C#程序设计，2011-4-30上午）；
2. 系统根据指定的课程，查找选修了该门课程的班级，并给将数据显示予用户（可以是单个班级信息，也可能是多个班级信息）；
3. 用户点击安排按钮，系统根据提供的信息，开始查询符合条件的资源进行安排；
4. 系统检测冲突，首先检测时间是否相同，根据时间检测结果进行下一步操作，e或者f；
5. 若时间相同检测班级，根据班级检测结果进行下一步g或h，
6. 若时间不同，则显示安排结果，给出成功提示；
7. 若班级冲突，则提示冲突原因，及操作提示；
8. 若班级不冲突，检测考场是否冲突，根据检测结果进行下一步i或j；
9. 若考场冲突，则提示冲突原因，及操作提示。
10. 若考场不冲突，则检测监考老师是否冲突，根据结果进行下一步k或l；
11. 若监考老师冲突，则提示冲突原因，及操作提示；
12. 若监考老师不冲突，则提示考场安排成功。

该过程的流程图如下：



此外系统还支持，资源查询，如剩余考场数，未安排班级等。

### 4.3.2考场安排查询模块

该模块为用户提供多种查询方式进行考场安排查询，用户可以根据需要，进行选择查询。提供的选择条件有：学院、班级、课程、时间、考场、监考老师，提供的排序方式有根据按时间，按班级编号，按考场编号。用户可以根据列出的条件进行组合查询，当系统有满足条件的记录时，给出相应记录，若是用户没有选择任何条件，那么系统则把所有记录数呈现。

# 5系统的实现

## 5.1系统部署图



图4

## 5.2系统人机交互设计

系统考场查询模块人机交互设计：

|  |
| --- |
| 输入本次考场查询命令：  输出提示信息；  for 检测用户输入的查询条件；  if 满足考场的条件存在；  then 打印符合条件的考场信息；  else 打印所有考场信息；  end for；  打印查询结果，及提示信息（结果是否为输入条件的结果） |

## 5.3 SQL Server 2005 服务器中使用存储过程

存储过程不仅可以重复使用，同时还可以提高查询速率，提高性能。系统考场查询模块的设计就是用存储过程处理的；代码如下：

CREATE PROCEDURE ProSearchTest

@dno int, --学院编号

@classno int,--班级编号

@testtime varchar(50),--考试时间

@roomno int,--考场编号

@tno int,--监考老师编号

@cno int--课程编号

AS

begin

select \* from test1

where (dno=@dno or @dno is null)

and (classno=@classno or @classno is null)

and (testtime=@testtime or @testtime='')

and (roomno=@roomno or @roomno is null)

and (tno1=@tno or @tno is null or tno2=@tno)

and (cno=@cno or @cno is null)

end

GO

# 6 系统功能测试

主要对系统各功能模块进行黑盒测试，例如：

1．登录模块，当用输入错误的用户名和密码是，禁止登入，同时能够给出相关提示信息；

2．剩余资源查询模块，选择录入搜索信息，能提供正确的资源剩余情况。

3．系统稳定性测试，当系统访问量达到一定量时，是否能运行正常，及运行速度是否减慢。

# 7总结

考务管理系统的开发，使得高校考务管理工作逐渐从手动劳动的环境中脱离出来，方便高校考务管理的同时提高了工作效率，实现了考场的智能化安排。然而受个人技术的局限性，系统在某些方面（如画面美化设计、效率问题考虑）还有待改进。通过这个系统的开发让我学到了很多以前没有用过的知识，如一些联动效果，无刷新效果，存储过程的技术等等,同时，锻炼了自己的独立思考能力和实际操作能力。

# 参考文献

[1] 庞哑娟 孙明丽 吕继迪 ASP技术方案宝典，北京：人民邮电出版社 2008.

[2] 关涛 CSS从入门到精通，北京：化学工业出版社 2009.

[3] 陈伟 编 SQL Server 2005数据库应用开发教程 北京：清华大学出版社，2005.

[4] 肖刚 古辉 程振波 张元鸣 实用软件文档写作，北京：清华大学出版社，2009.