

**《数据库课程设计》说明书**

**题目：** 学生宿舍管理系统

**院 系： 计算机工程学院**

**专业班级： 软件工程2办**

**学 号：**

**学生姓名： 方银城**

**班内序号： 39**

**指导教师： 秦映波**

**2016年12月12日**

摘要

数据库是数据管理的最新技术，是计算机科学的重要分支。随着计算机科学与技术的迅速发展，计算机应用领域也在不断扩大。今天，信息资源已成为各个部门的重要财富和资源，几乎各行各业都有关于计算机的使用，建立一个满足各级部门信息护理要求的行之有效的信息系统也成为一个企业或组织生存和发展的重要条件。因此，作为信息系统核心和基础的数据库管理系统得到越来越广泛的应用，由于数据库技术的使用，使得人们在于管理、应用及服务等各个领域使用数据方面变得更加简洁，更加方便，大大提高了工作效率，产生了以数据库为主体，以人为辅助的局面。

我的本次课程设计便是以学校信息化管理为目的开发的一套数据库系统——宿舍管理系统。有了宿舍管理系统，更方便于对学生以及宿舍进行分配和管理，而且对于学生的出入监管以及来访人员的管理起到便捷的作用。

目录

**1.** [**概述**](#y4)**4**

**2.** [**课程设计任务的需求分析**](#y4)**4**

**2.1** [**设计任务**](#y4)**4**

**2.2** [**设计要求**](#y4)**4**

**2.3** [**需求描述的规范文档**](#y4)**4**

**3.** [**概念结构设计**](#y4)**4**

**3.1** [**概念结构设计工具(E-R模型)**](#y4)**4**

**3.2** [**管理员子系统**](#y4)**4**

**3.3**[**宿舍子系统**](#y5)**5**

**3.4** [**房间子系统**](#y5)**5**

**3.5** [**学生子系统**](#y6)**6**

**3.6** [**院系子系统**](#y6)**6**

**3.7** [**出入登记子系统**](#y7)**7**

**3.8** [**来访登记子系统**](#y7)**7**

**3.9**[**总体E-R图**](#y7)**7**

**4.** [**逻辑结构设计**](#y8)**8**

**4.1** [**关系设计模式**](#y8)**8**

**4.2** [**视图的设计**](#y9)**9**

**4.3** [**优化**](#y10)**10**

**5.** [**数据库物理设计与实施**](#y10)**10**

**5.1**[**数据库应用的硬件、软件环境介绍**](#y10)**10**

**5.2**[**物理结构设计**](#y10)**10**

**5.3**[**索引的设计**](#y10)**10**

**5.4**[**建立数据库**](#y10)**10**

**5.5**[**加载测试数据**](#y13)1**3**

**6.** [**数据操作要求及实现**](#y14)**14**

**6.1** [**数据查询**](#y14)**14**

**6.2** [**数据更新**](#y15)**15**

**6.3** [**数据删除**](#y15)**15**

**6.4** [**数据添加**](#y15)**15**

**6.5** [**数据维护**](#y16)**16**

**7.** [**源代码及查询截图**](#y16)**16**

**7.1** [**程序流程图**](#y16)**16**

**7.2** [**数据库链接**](#y16)**16**

**7.3** [**重点代码分析**](#y17)**17**

**7.4** [**查询截图**](#y19)**19**

**8.** [**收获、体会和建议**](#y21)**21**

**9.** [**主要参考文献**](#y21)**21**

1. **概述**

本次课程设计是数据库课程设计，我选的是学生宿舍管理系统，功能块主要是宿舍管理、学生管理、出入登记和来访登记，目的在于将学生宿舍的管理实现信息化，监控学生宿舍的情况防止意外发生。

1. **课程设计任务的需求分析**
   1. **设计任务**

设计出一个方便进行宿舍管理的系统的数据库。

* 1. **设计要求**

假定学校有多栋宿舍楼，每栋楼有多层，每层有多个寝室，每个寝室可住多名学生，学生宿舍管理系统对学校的学生宿舍进行规范管理，其管理的对象如下：

．宿舍信息：编号、楼层、床位数、单价等。

．学生：学号、姓名、性别、年龄、所在院系、年级、电话等。

每个宿舍最多可以住4位同学，每个同学只能在一个宿舍，不同宿舍的费用标准可以不同。不同院系、年级的同学可以住同一间宿舍。

系统要能够对宿舍、学生、住宿信息进行登记、调整，并能随时进行各种查询、统计等处理。包括：

．寝室分配：根据院系、年级分配寝室。

．学生管理：实现入住学生信息的登记、维护和查询功能。

．信息查询：按公寓楼号、学生姓名等查询住宿信息。

．出入登记：对学生进出公寓的情况进行登记、实现基本的出、入监控功能。

* 1. **需求描述的规范文档**

1. 宿舍情况查询和清空宿舍；
2. 学生信息查询和修改、删除和添加；
3. 学生出入登记；
4. 来访登记；
5. 权限管理，包括三种权限(宿舍管理员、高级管理员和超级管理员)；
6. 宿舍管理员可以编辑出入登记和来访登记，但是不能操作学生信息和宿舍信息，只能查询；
7. 高级管理员可以操作学生信息和宿舍信息但是不能编辑出入登记和来访登记；
8. 超级管理员除了高级管理员的权限外还能对高级管理员和宿舍管理员的用户进行添加和删除修改。
9. **概念结构设计**
   1. **概念结构设计工具(E-R模型)**

使用E-R图作为概念结构设计工具，构建出数据库E-R模型。

* 1. **管理员子系统**

3.2.1子系统描述

管理员表：id，名字，密码，联系方式，权限。

其中主键为id。

3.2.2分E-R图

管理员表

3.2.3说明

管理员表用于记录管理员信息。

* 1. **宿舍子系统**

3.3.1 子系统描述

宿舍表：id，宿舍管理员id，宿舍号（如C），宿舍楼号，宿舍学生性别。

其中主键为id，外键为宿舍管理员id。

3.3.2 分E-R图

宿舍表

3.3.3 说明

宿舍表用于记录已有宿舍信息。

* 1. **房间子系统**

3.4.1 子系统描述

房间表：id，宿舍id，房间号，价格。

其中主键为id，外键为宿舍id。

3.4.2 分E-R图

房间表

3.4.3 说明

房间表用于记录房间信息。

* 1. **学生子系统**

3.5.1子系统描述

学生表：id，学号，姓名，性别，年龄，联系方式，学院id，专业id，年级，宿舍id，房间id。

其中主键为id，外键为学院id，专业id，宿舍id，房间id。

3.5.2分E-R图

学生表

3.5.3说明

学生表用于记录学生信息。

* 1. **院系子系统**

3.6.1子系统描述

院系专业表：id，院系名字，所属学院id。

其中主键为id，外键为所属学院id。

3.6.2分E-R图

院系专业表

3.6.3说明

院系专业表用于记录院系专业信息。

* 1. **出入登记子系统**

3.7.1 子系统描述

出入登记表：id，宿舍id，学生id，出去时间，返回时间。

其中主键为id，外键为宿舍id，学生id。

3.7.2 分E-R图

出入登记表

3.7.3 说明

出入登记表用于记录学生出入宿舍的信息。

* 1. **来访登记子系统**

3.8.1 子系统描述

来访登记表：id，宿舍id，学生id，来访人名字，来访时间，出去时间。

其中主键为id，外键为宿舍id，学生id。

3.8.2 分E-R图

来访登记表

3.8.3 说明

来访登记表用于宿舍来访登记。

* 1. **总体E-R图**

管理员

1

管理

guanine

n

包含

n

1

房间

宿舍

1

1

1

1

住宿

记录

住宿

n

n

n

记录

1

n

来访登记

学生

记录

1

n

记录

n

n

所属

出入登记

1

院系专业

1. **逻辑结构设计**

**4.1 关系数据模式**

管理员表(admin)：id int 主键

名字(aname) varchar(10) 非空

密码(password) varchar(10) 非空

联系方式(tel) varchar(20) 非空

权限(power) tinyint 非空

宿舍表(dormitory)：id int 主键

管理员id(a\_id) int 非空 外键联系admin表

宿舍号（如C）(ch) char(1) 非空

宿舍编号(num) int 非空

宿舍学生性别(s\_gender) varchar(10) 非空

房间表(room)：id itn 主键

宿舍id(d\_id) int 非空 外键联系dormitory表

宿舍号(num) int 非空

剩余床位(margin) tinyint 非空

价格(price) int 非空

学生表(student)：id int 主键

学号(sno) varchar(20) 非空

姓名(sname) varchar(10) 非空

性别(gender) varchar(10) 非空

年龄(age) int 非空

联系方式(tel) varchar(20) 非空

学院id(u\_id) int 非空 外键联系major表

专业id(m\_id) int 非空 外键联系major表

年级(class) int 非空

宿舍id(d\_id) int 非空 外键联系dormitory表

房间id(r\_id) int 非空 外键联系room表

院系专业表(major)：id int 主键

名字(name) varchar(50) 非空

所属学院id(m\_id) int 非空 外键联系major表

出入登记表(in\_out)：id int 主键

宿舍id(d\_id) int 非空 外键联系dormitory表

学生id(s\_id) int 非空 外键联系student表

出去时间(out\_date) datetime 非空

返回时间(in\_date) datetime

来访登记表(visit)：id int 主键

宿舍id(d\_id) int 非空 外键联系dormitory表

学生id(s\_id) int 非空 外键联系student表

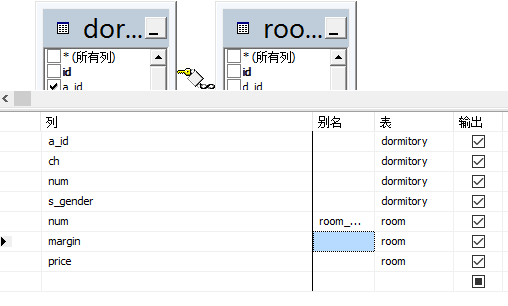
来访人名字(name) varchar(10) 非空

来访时间(in\_date) datetime 非空

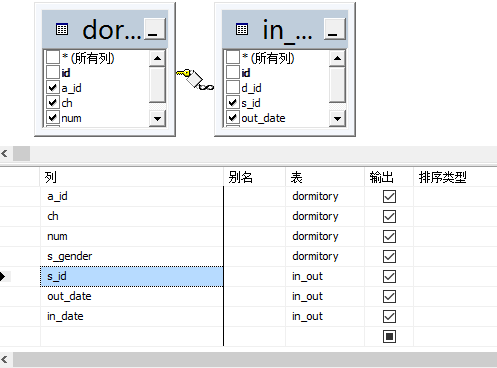
出去时间(out\_date) datetime

**4.2 视图的设计**

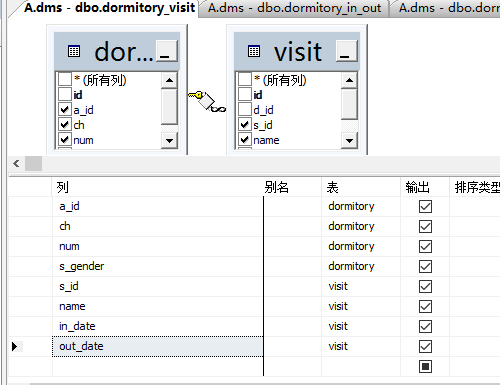
dormitory和room之间的视图：



dormitory和in\_out之间的视图：



dormitory和visit之间的视图：



**4.3 优化**

在宿舍表和房间表、学生表、出入登记表、来访登记表之间建立外键约束，实现删除和更新的级联，并且建立视图。

1. **数据库物理设计与实施**

**5.1 数据库应用的硬件、软件环境介绍**

此次数据库设计使用的硬件包括笔记本电脑一台，装有win10系统，软件环境为VS2012和sql server2008。

**5.2 物理结构设计**

数据库表的设计可见4.1关系数据模式，同时为了提高数据库的查找效率和用户响应速度，数据库应该建立相应的视图和索引提高存取效率，其中视图见4.2。

**5.3 索引的设计**

本次数据库课程设计除了主键索引和非空索引外并没有设置其他索引。

**5.4 建立数据库**

数据库：

CREATE DATABASE [dms] ON PRIMARY

( NAME = N'dms', FILENAME = N'D:\dms.mdf' , SIZE = 3072KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG ON

( NAME = N'dms\_log', FILENAME = N'D:\dms\_log.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

表：

**admin表：**

CREATE TABLE [dbo].[admin](

[id] [int] IDENTITY(1,1) NOT NULL,

[aname] [varchar](10) NOT NULL,

[password] [varchar](10) NOT NULL,

[tel] [varchar](20) NOT NULL,

[power] [tinyint] NOT NULL,

CONSTRAINT [PK\_admin] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

**dormitory表：**

CREATE TABLE [dbo].[dormitory](

[id] [int] IDENTITY(1,1) NOT NULL,

[a\_id] [int] NOT NULL,

[ch] [char](1) NOT NULL,

[num] [int] NOT NULL,

[s\_gender] [varchar](10) NOT NULL,

CONSTRAINT [PK\_dormitory] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

**student表：**

CREATE TABLE [dbo].[student](

[id] [int] IDENTITY(1,1) NOT NULL,

[sno] [varchar](20) NOT NULL,

[sname] [varchar](10) NOT NULL,

[gender] [varchar](10) NOT NULL,

[age] [int] NOT NULL,

[tel] [varchar](20) NOT NULL,

[u\_id] [int] NOT NULL,

[m\_id] [int] NOT NULL,

[class] [int] NOT NULL,

[d\_id] [int] NOT NULL,

[r\_id] [int] NOT NULL,

CONSTRAINT [PK\_student] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

**major表：**

CREATE TABLE [dbo].[major](

[id] [int] IDENTITY(1,1) NOT NULL,

[name] [varchar](50) NOT NULL,

[m\_id] [int] NOT NULL,

CONSTRAINT [PK\_major] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

**visit表：**

CREATE TABLE [dbo].[visit](

[id] [int] IDENTITY(1,1) NOT NULL,

[d\_id] [int] NOT NULL,

[s\_id] [int] NOT NULL,

[name] [varchar](10) NOT NULL,

[in\_date] [datetime] NOT NULL,

[out\_date] [datetime] NULL,

CONSTRAINT [PK\_visit] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

**in\_out表：**

CREATE TABLE [dbo].[in\_out](

[id] [int] IDENTITY(1,1) NOT NULL,

[d\_id] [int] NOT NULL,

[s\_id] [int] NOT NULL,

[out\_date] [datetime] NOT NULL,

[in\_date] [datetime] NULL,

CONSTRAINT [PK\_in\_out] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

**in\_out表：**

CREATE TABLE [dbo].[room](

[id] [int] IDENTITY(1,1) NOT NULL,

[d\_id] [int] NOT NULL,

[num] [int] NOT NULL,

[margin] [tinyint] NOT NULL,

[price] [int] NOT NULL,

CONSTRAINT [PK\_room\_1] PRIMARY KEY CLUSTERED

(

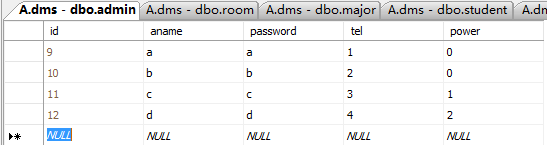
[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

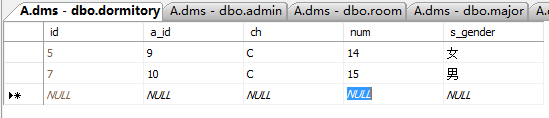
) ON [PRIMARY]

**5.5 加载测试数据**

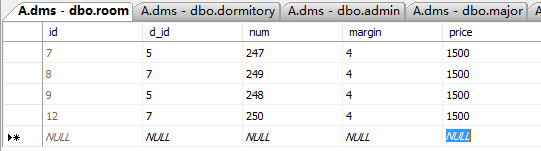
admin表如下：其中0为宿舍管理员权限(此权限无法查询未分配学生)，1为高级管理员，2为超级管理员



dormitory表如下：



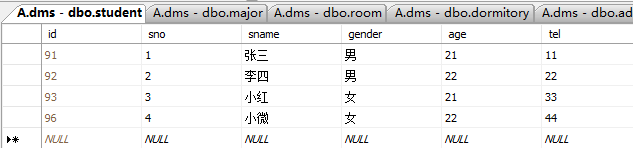
room表如下：

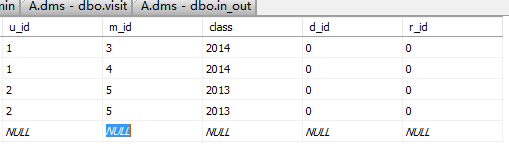


major表如下：



student表如下：





visit表和in\_out表为空。

1. **数据操作要求及实现**
   1. **数据查询**

关于数据查询的代码举例如下：

查询管理员用户名密码：

select id,aname,password,power from admin

查询宿舍楼号：

select ch+cast(num as varchar(2)) d\_id from dormitory

查询未归学生：

select student.sname name from in\_out left join student on in\_out.s\_id=student.id where in\_date is null

查询宿舍：

select room.id, dormitory.ch+cast(dormitory.num as varchar(2)) 楼号,dormitory.s\_gender 性别,room.num 房号,room.margin 剩余床位,room.price 价格 from room left join dormitory on room.d\_id = dormitory.id

“select room.id, dormitory.ch+cast(dormitory.num as varchar(2)) 楼号,dormitory.s\_gender 性别,room.num 房号,room.margin 剩余床位,room.price 价格 from room left join dormitory on room.d\_id = dormitory.id where" + s1 + s2 + s3

查询学生：

"select student.id, student.sno 学号,student.sname 姓名,student.gender 性别,student.age 年龄,student.tel 联系方式,student.class 年级,A.name 学院,B.name 专业,dormitory.ch+cast(dormitory.num as varchar(2)) 楼号,student.r\_id 房间id,room.num 房间" + " from (((student left join major A on student.u\_id=A.id) left join major B on student.m\_id=B.id) left join dormitory on student.d\_id=dormitory.id) left join room on student.r\_id=room.id" + " where"+s1+s2+s3+s4+s5+s6+s7+s8+s9+s10

查询出入登记记录：

"select student.sno 学号,student.sname 姓名,student.tel 联系方式,student.class 年级,A.name 学院,B.name 专业,room.num 房间,in\_out.out\_date 出去时间,in\_out.in\_date 返回时间" +" from ((((in\_out left join student on in\_out.s\_id=student.id) left join major A on student.u\_id=A.id) left join major B on student.m\_id=B.id) left join dormitory on student.d\_id=dormitory.id) left join room on student.r\_id=room.id" +" where" + s1 + s2

查询来访登记记录：

"select visit.name 来访人,student.sno 学号,student.sname 姓名,student.tel 联系方式,student.class 年级,A.name 学院,B.name 专业,room.num 房间,visit.in\_date 访问时间,visit.out\_date 出去时间" +" from ((((visit left join student on visit.s\_id=student.id) left join major A on student.u\_id=A.id) left join major B on student.m\_id=B.id) left join dormitory on student.d\_id=dormitory.id) left join room on student.r\_id=room.id" +" where" + s1 + s2

查询管理员：

select id, aname 名字,tel 联系方式,power 权限 from admin

* 1. **数据更新**

关于数据更新举例如下：

修改学生：

"update student set"+str+"where sno='"+s1+"'"

修改出入登记记录：

"update in\_out set in\_date='"+t.ToString()+"' where d\_id="+s1+" and s\_id="+s2

修改来访登记记录：

"update visit set out\_date='" + t.ToString() + "' where d\_id=" + s1 + " and name='" + s2 + "'"

* 1. **数据删除**

关于数据删除举例如下：

删除学生：

string sqlStr = "delete from student where id=";

删除管理员：

string sqlStr = "delete from admin where id=";

* 1. **数据添加**

关于数据添加举例如下：

添加学生：

"insert into student (sno,sname,gender,age,tel,u\_id,m\_id,class,d\_id,r\_id) values('"+s1+"','"+s2+"','"+s3+"',"+s6+",'"+s5+"',"+s7+","+s8+","+s4+","+s9+","+s10+ ")"

添加出入登记：

"insert into in\_out (d\_id,s\_id,out\_date) values(" + s1 + "," + s2 + ",'" + t.ToString() + "')"

添加来访登记：

"insert into visit (d\_id,s\_id,name,in\_date) values(" +ConnectDB.getInstance().GetDorId(s1)+","+ConnectDB.getInstance().GetStuId(s2)+",'"+s3+"','"+d.ToString()+"')"

* 1. **数据维护**

关于数据维护方面，每隔固定时间就要备份一次数据，并且删除冗余数据。以防止数据丢失和出错。

1. **源代码及查询截图**
   1. **程序流程图**

添加管理员

删除管理员

查询管理员

修改密码

高级管理

登陆

来访登记

学生管理

宿舍管理

出入登记

查询

查询

查询

查询

访

出

删除

查空床位

走

入

插入

清空宿舍

修改

* 1. **数据库链接**

关于数据库链接的代码如下：

string str;

SqlConnection con;

static ConnectDB cdb = new ConnectDB();

private ConnectDB()

{

str = @"Data Source=A;Initial Catalog=dms;Persist Security Info=True;User ID=dms;Password=dms";

con = new SqlConnection(str);

}

数据库连接的打开和关闭：

con.Open();

con.Close();

* 1. **重点代码分析**
     1. 登陆：通过查询数据库中管理员的名字和密码以及权限进行登陆，代码如下，

con.Open();

string cmd = "select id,aname,password,power from admin";

SqlCommand com = new SqlCommand(cmd, con);

SqlDataReader reader = com.ExecuteReader();

while (reader.Read())

{

String user = reader["aname"].ToString();

String pwd = reader["password"].ToString();

Form1.power = Int32.Parse(reader["power"].ToString().Trim());

Form1.admin\_name = "欢迎，"+user;

Form1.admin\_id = Int32.Parse(reader["id"].ToString().Trim());

if (user.Trim() == u && pwd.Trim() == p)

{

reader.Close();

con.Close();

return true; //登陆成功为true

}

}

reader.Close();

con.Close();

return false;

* + 1. 按权限显示内容：通过判断登陆时获取的权限，决定是否显示相应的界面和内容，代码部分举例如下，

if (b == true && power == 0)

{

v\_name\_label1.Visible = b;

v\_name1.Visible = b;

v\_name\_label2.Visible = b;

v\_name\_c.Visible = b;

v\_name\_label3.Visible = b;

v\_name2.Visible = b;

in\_bt.Visible = b;

out\_bt.Visible = b;

}

else

{

v\_name\_label1.Visible = false;

v\_name1.Visible = false;

v\_name\_label2.Visible = false;

v\_name\_c.Visible = false;

v\_name\_label3.Visible = false;

v\_name2.Visible = false;

in\_bt.Visible = false;

out\_bt.Visible = false;

}

* + 1. 清空宿舍：通过将学生的所属宿舍id改为0并且将宿舍剩余床位数量改为相应值来情况宿舍，代码部分举例如下，其中dat为查询出来的结果，最后一句模仿点击查询，将清空之后的结果显示，

if (dat == null)

{

MessageBox.Show("没有选中的内容！", "提示");

return;

}

string sqlStr = "update room set margin=4 where id=";

for (int i = 0; i < dat.Rows.Count; i++)

{

ConnectDB.getInstance().Change(sqlStr+dat.Rows[i]["id"]);

}

sqlStr = "update student set d\_id=0,r\_id=0 where r\_id=";

for (int i = 0; i < dat.Rows.Count; i++)

{

ConnectDB.getInstance().Change(sqlStr + dat.Rows[i]["id"]);

}

select\_Click(sender, e);

* + 1. 按条件查询学生：通过左连接查询相应条件的学生进行显示，代码如下，

string s1 = num.Text.ToString().Trim();

string s2 = name.Text.ToString().Trim();

string s3 = gender.Text.ToString().Trim();

string s4 = class\_t.Text.ToString().Trim();

string s5 = tel.Text.ToString().Trim();

string s6 = age.Text.ToString().Trim();

string s7 = school.Text.ToString().Trim();

string s8 = major.Text.ToString().Trim();

string s9 = do\_c.Text.ToString().Trim();

string s10 = room.Text.ToString().Trim();

try

{

if (s4 != "")

Int32.Parse(s4);

if (s6 != "")

Int32.Parse(s6);

if (s10 != "")

Int32.Parse(s10);

}

catch (Exception a)

{

MessageBox.Show("请输入正确格式的数据！", "提示");

return;

}

DataSet ds;

s1 = s1 != "" ? " student.sno='"+s1+"'" : " student.sno!='null'";

s2 = s2 != "" ? " and student.sname='" + s2 + "'" : " and student.sname!='null'";

s3 = s3 != "" ? " and student.gender='" + s3 + "'" : " and student.gender!='null'";

s4 = s4 != "" ? " and student.class=" + s4 : " and student.class!=0";

s5 = s5 != "" ? " and student.tel='" + s5 +"'" : " and student.tel!='null'";

s6 = s6 != "" ? " and student.age=" + s6 : " and student.age!=0";

s7 = s7 != "" ? " and A.name='" + s7 + "'" : " and A.name!='null'";

s8 = s8 != "" ? " and B.name='" + s8 + "'" : " and B.name!='null'";

s9 = s9 != "全部" ? " and dormitory.ch='" + s9[0] + "' and dormitory.num=" + s9.Substring(1, s9.Length - 1) : "";

s10 = s10 != "" ? " and room.num=" + s10 : "";

ds = ConnectDB.getInstance().GetData("select student.id, student.sno 学号,student.sname 姓名,student.gender 性别,student.age 年龄,student.tel 联系方式,student.class 年级,A.name 学院,B.name 专业,dormitory.ch+cast(dormitory.num as varchar(2)) 楼号,student.r\_id 房间id,room.num 房间" +" from (((student left join major A on student.u\_id=A.id) left join major B on student.m\_id=B.id) left join dormitory on student.d\_id=dormitory.id) left join room on student.r\_id=room.id" + " where"+s1+s2+s3+s4+s5+s6+s7+s8+s9+s10);

dat = ds != null ? ds.Tables[0] : null;

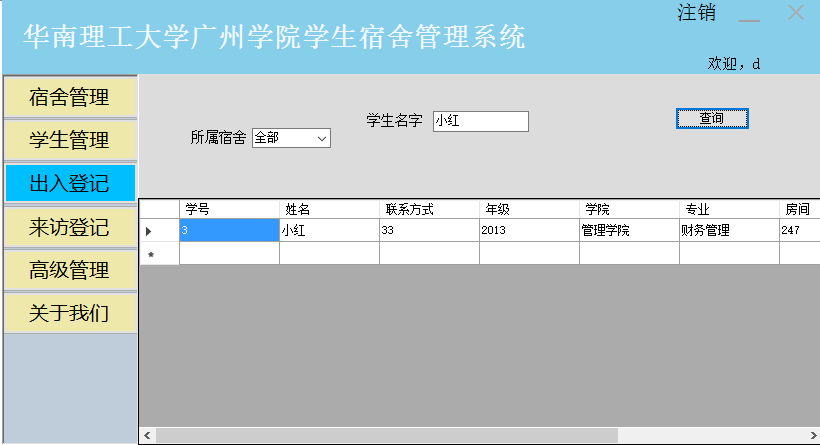
dataGridView0.DataSource = dat;

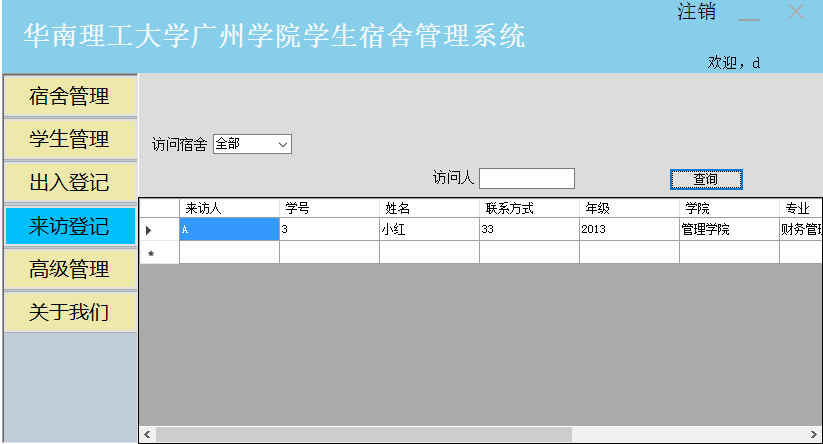
* 1. **查询截图**

以下截图为超级管理员（权限为2）权限下的情况:











1. **收获、体会和建议**

经过这次的数据库课程设计，首先，当然是得到了技能上的提升了，因为在完成这个课程设计的时候必须使用到C#和sql语言，并且C#的陌生和sql语言的复杂都让我有点难以解决，这便让我在这其中得到了技能的提升。

其次，由于本次课程设计的时间只有两周并且是在任务比较多的时间，这让我加快了学习的效率，学会了平心静气不浮躁的学习。

1. **主要参考文献**
2. 《C#程序设计及应用教程》; 马俊、邓居英、杨湖;
3. 《数据库原理及其应用教程(第三版)》; 黄德才主编; 科学出版社.