

[**数 据 库 技 术 及 应 用 B**](javascript:setQuery('QueryName','%E6%95%B0%E6%8D%AE%E5%BA%93%E6%8A%80%E6%9C%AF%E5%8F%8A%E5%BA%94%E7%94%A8B')) **课 程 设 计**

**成都路网信息**

**数据库系统的设计与实现**

学院： 交通运输与物流学院

成员与学号、专业与班级： 刘欣豪 （2020112921） 交通4班

指导老师： 葛乾

|  |  |
| --- | --- |
| 课程  设计  得分 | 刘欣豪  2020112921 |
|  |

目录

[一、简介 2](#_Toc121942026)

[1.1、选题动机 2](#_Toc121942027)

[1.2、系统简介 2](#_Toc121942028)

[1.3、成员构成 2](#_Toc121942029)

[二、需求分析 2](#_Toc121942031)

[2.1、功能需求分析 2](#_Toc121942032)

[2.2、数据流图 3](#_Toc121942033)

[三、概念设计 3](#_Toc121942034)

[3.1、实体属性E-R图 3](#_Toc121942035)

[3.2、实体联系图 5](#_Toc121942036)

[3.3、完整的E-R图 5](#_Toc121942037)

[四、逻辑设计 6](#_Toc121942038)

[4.1、系统总体设计 6](#_Toc121942039)

[4.2、数据库设计 6](#_Toc121942040)

[五、代码实现 7](#_Toc121942041)

[5.1、数据库的建立 7](#_Toc121942042)

[5.2、公共类代码设计 12](#_Toc121942043)

[5.3、登陆界面代码设计 15](#_Toc121942044)

[5.4、主菜单窗体设计 18](#_Toc121942045)

[5.5、路段信息管理功能设计 23](#_Toc121942046)

[5.6、卡口信息管理功能设计 42](#_Toc121942047)

[5.7、栅格数据管理功能设计 52](#_Toc121942048)

[5.8、数据查询与导出功能设计 53](#_Toc121942049)

[5.9、统计信息功能设计 56](#_Toc121942050)

[5.10、帮助信息功能设计 57](#_Toc121942051)

[六、总结 58](#_Toc121942052)

[1、优点 58](#_Toc121942053)

[2、缺点 58](#_Toc121942054)

[七、其它 58](#_Toc121942055)

[参考文献 58](#_Toc121942056)

[致谢 58](#_Toc121942057)

一、简介

1.1、选题动机

本报告设计的数据库系统为成都市主干路路网数据库系统。在专业学习中，时常会遇到对公交车路径进行优化、对公交车充电过程进行优化、对公共交通尾气排放计算，诸如此类需要使用到路网数据的研究。尤其是在进行成都市路网排放计算时，在处理基础数据的过程中，遭遇过最大痛苦的就是不同标准的路网数据对应不上、卡口地理位置数据与路网数据分离、栅格与路网无直接访问联系，需要进行邻域匹配后再一一进行人工校验，十分消耗精神与体力，让人无比恼火；此外，在最终将所有数据匹配完毕后，存储在Excel表格中列数很多，取用也十分不便。联系到本学期的学到的数据库系统设计课程，在编程王国遨游一圈听过TICO的讲解后，顿感数据库之妙用[1]。

当然，大概率在一些科研实验室、政府机关路网数据系统早已存在且十分完善，发挥着作用。故此次设计报告主要的目的还是锻炼自身的数据库设计能力，提高自己的专业素养，并尽力在一定可能性下去方便未来的学习与研究。

1.2、系统简介

该系统主要为主要功能为管理路网结构数据，包括栅格、卡口等附加信息，核心功能为方便数据的筛选、连接与导出。

1.3、成员构成

刘欣豪，承担任务：理念提出、系统设计、代码编程、文稿撰写。

二、需求分析

2.1、功能需求分析

（1）登录功能

对于该数据库系统，本身数据具有一定的保密性，同时为保证数据库内信息不被他人误修改，故内置私人用户，分为管理与普通用户。

（2）数据更新功能

在社会发展与道路网变化的双重背景下，现有数据可能不能满足需求，即可利用该功能进行路网数据、卡口数据、栅格数据的增、删、改。

（3）数据查询功能

对于该系统，核心要素为数据的取用，通过长度、所处区位、道路等级等标准的筛选得到所需数据。

（4）数据导出功能

当数据经筛选查询后，若目标应用无Sql接口，则可选择导出为csv或excel数据表，再读入后使用。

（5）统计功能

记录数据库中各个类型数据的数量。

（6）帮助功能

帮助功能为记录版本信息与作者联系信息。

2.2、数据流图



图2.1数据库系统数据流图

三、概念设计

关联数据主要有三类，路网数据、卡口数据、栅格数据。其中路网数据可细分为线与节点，故共4个实体。

3.1、实体属性E-R图

注：以下的name其中可能包含重复名

（1）线

线即代表线路，主要有属性E\_FID、E\_name、length、highway、maxspeed。E\_FID为主码，不存在传递依赖与部分依赖，属于第三范式(3NF)。



图3.1：LINE属性图

（2）节点

节点的为线的交叉点，主要有N\_FID、longitude、latitude、degree。N\_FID为主码，不存在传递依赖与部分依赖，属于第三范式(3NF)。



图3.2：NODE属性图

（3）卡口

卡口数据为路面卡口信息，包含有C\_FID、device\_name、longitude、latitude、Linked\_id、卡口所摄流量信息。B\_FID为主码，不存在传递依赖与部分依赖，属于第三范式(3NF)。



图3.3：CAMERA属性图

（4）栅格数据

栅格数据为将地图分为长宽等相同的栅格，包含有R\_FID、center\_x、center\_y、s\_length、以及栅格环境信息等。R\_FID为主码，不存在传递依赖与部分依赖，属于第三范式(3NF)。



图3.4：RASTER属性图

3.2、实体联系图

对于各个实体而言，节点与线之间存在连接关系即线有起点与终点；卡口依附于边，一个边可以有多个卡口，一个卡口只能依附于一条边；栅格代表地理信息数据，其中可能存储一些环境信息等，其与各实体存在地理坐标的联系。



图3.5：实体联系图

3.3、完整的E-R图

对整体而言，即：



图3.6：路网信息E-R图

四、逻辑设计

4.1、系统总体设计

该系统主要针对私人用户使用而设计，为方便管理，故在开发中分离为管理员账户（可增删），与普通用户（只读）。

4.2、数据库设计

对于图3.6中的信息，关系模型设计如下(关系的码加粗,外键加下划线)：

NODE（**N\_FID**, longitude, latitude, degree，R\_FID）

EDGE（**E\_FID**, E\_name, length, highway, maxspeed, u, v，R\_FID）

CAMERA(**C\_FID**, device\_name, longitude, latitude，land\_cover, E\_FID)

RASTER(**R\_FID**, center\_x, center\_y, s\_length, humidity)

SUser(**用户名**,密码,级别)

具体属性信息如下：

（1）NODE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段号 | 字段名称 | 字段类型 | 字段大小 | 索引 | 字段说明 |
| 1 | N\_FID | char | 10 | 主键 | 编号 |
| 2 | longitude | double | ~ | ~ | 经度 |
| 3 | latitude | double | ~ | ~ | 纬度 |
| 4 | degree | int | 10 | ~ | 度 |
| 5 | R\_FID | int | 10 | 外键 | 所属栅格 |

表4.1：节点数据库信息表

（2）EDGE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段号 | 字段名称 | 字段类型 | 字段大小 | 索引 | 字段说明 |
| 1 | E\_FID | char | 10 | 主键 | 编号 |
| 2 | E\_name | char | 20 | ~ | 路段名字 |
| 3 | length | double | ~ | ~ | 路段长度 |
| 4 | highway | char | 10 | ~ | 路段类型 |
| 5 | u | char | 10 | 外键 | 起点 |
| 6 | v | char | 10 | 外键 | 终点 |
| 7 | maxspeed | int | ~ | ~ | 最大限速 |
| 8 | R\_FID | char | 10 | 外键 | 对应栅格 |

表4.2：路段数据库信息表

（3）CAMERA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段号 | 字段名称 | 字段类型 | 字段大小 | 索引 | 字段说明 |
| 1 | C\_FID | char | 20 | 主键 | 编号 |
| 2 | device\_name | char | 40 | ~ | 路段名字 |
| 3 | longitude | double | ~ | ~ | 经度 |
| 4 | latitude | double | ~ | ~ | 纬度 |
| 5 | land\_cover | char | 15 | ~ | 覆盖车道编号 |
| 6 | E\_FID | char | 10 | 外键 | 所属路段 |

表4.3：卡口数据库信息表

（4）RASTER

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段号 | 字段名称 | 字段类型 | 字段大小 | 索引 | 字段说明 |
| 1 | R\_FID | char | 10 | 主键 | 编号 |
| 2 | center\_x | double | ~ | ~ | 经度 |
| 3 | center\_y | double | ~ | ~ | 纬度 |
| 4 | s\_length | double | ~ | ~ | 边长 |
| 5 | humidity | double | ~ | ~ | 湿度 |

表4.4：栅格数据库信息表

（5）SUser

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段号 | 字段名称 | 字段类型 | 字段大小 | 索引 | 字段说明 |
| 1 | 用户名 | char | 10 | 主键 | ~ |
| 2 | 密码 | char | 10 | ~ | ~ |
| 3 | 级别 | char | 10 | ~ | ~ |

表4.5：用户数据库信息表

五、代码实现

5.1、数据库的建立

（1）创建数据库

1. **create** **schema** network\_chengdu

（2）建表并添加完整性约束

创建RASTER表：

1. **create** **table** RASTER(
2. R\_FID **char**(10) **primary** **key** ,
3. center\_x **double** ,
4. center\_y **double** ,
5. s\_length **double**,
6. humidity **double**
7. )

创建NODE表:

1. **create** **table** NODE(
2. N\_FID **char**(10) **primary** **key** ,
3. longitude **double** ,
4. latitude **double** ,
5. degree **int** ,
6. R\_FID **char**(10),
7. **constraint** N\_R
8. **foreign** **key**(R\_FID) **references** RASTER(R\_FID)
9. )

创建EDGE表:

1. **create** **table** EDGE
2. (
3. E\_FID   **char**(10) **primary** **key**,
4. E\_NAME  **char**(20),
5. length  **double**,
6. highway **char**(10),
7. u       **char**(10),
8. v       **char**(10),
9. maxspeed **int** ,
10. R\_FID **char**(10),
11. **CONSTRAINT** L\_N\_u
12. **foreign** **key** (u) **references** NODE (N\_FID),
13. **CONSTRAINT** L\_N\_v
14. **foreign** **key** (v) **references** NODE (N\_FID),
15. **CONSTRAINT** L\_R
16. **foreign** **key** (R\_FID) **references** raster (R\_FID)
17. )

创建CAMERA表:

1. **create** **table** CAMERA(
2. C\_FID **char**(20) **primary** **key** ,
3. device\_name **char**(40) ,
4. longitude **double** ,
5. latitude **double** ,
6. land\_cover **char**(15) ,
7. E\_FID **char**(10) ,
8. **constraint** C\_E
9. **foreign** **key** (E\_FID) **references** EDGE(E\_FID)
10. )

同时还应创建SUser存储使用账户信息：

1. **create** **table** SUser(
2. 用户名 **char**(10) **primary** **key** ,
3. 密码 **char**(10),
4. 级别 **char**(10)
5. )

（3）从csv导入基础数据进入各表

导入过程以RASTER为例，RASTER原csv中数据共5688条

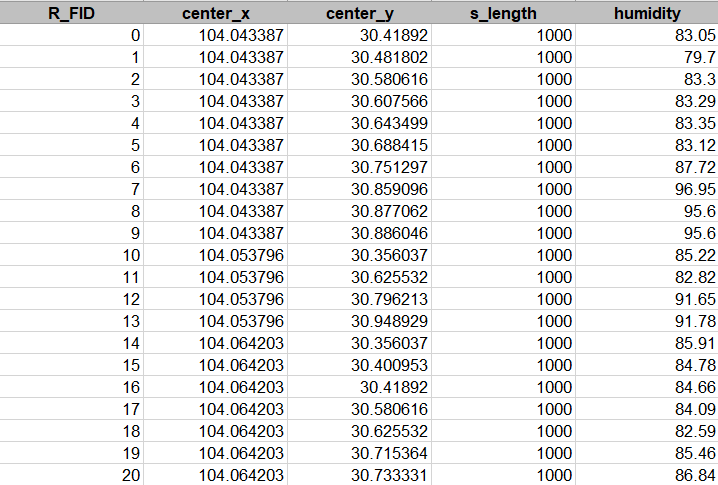


图5.1.1：RASTER部分数据

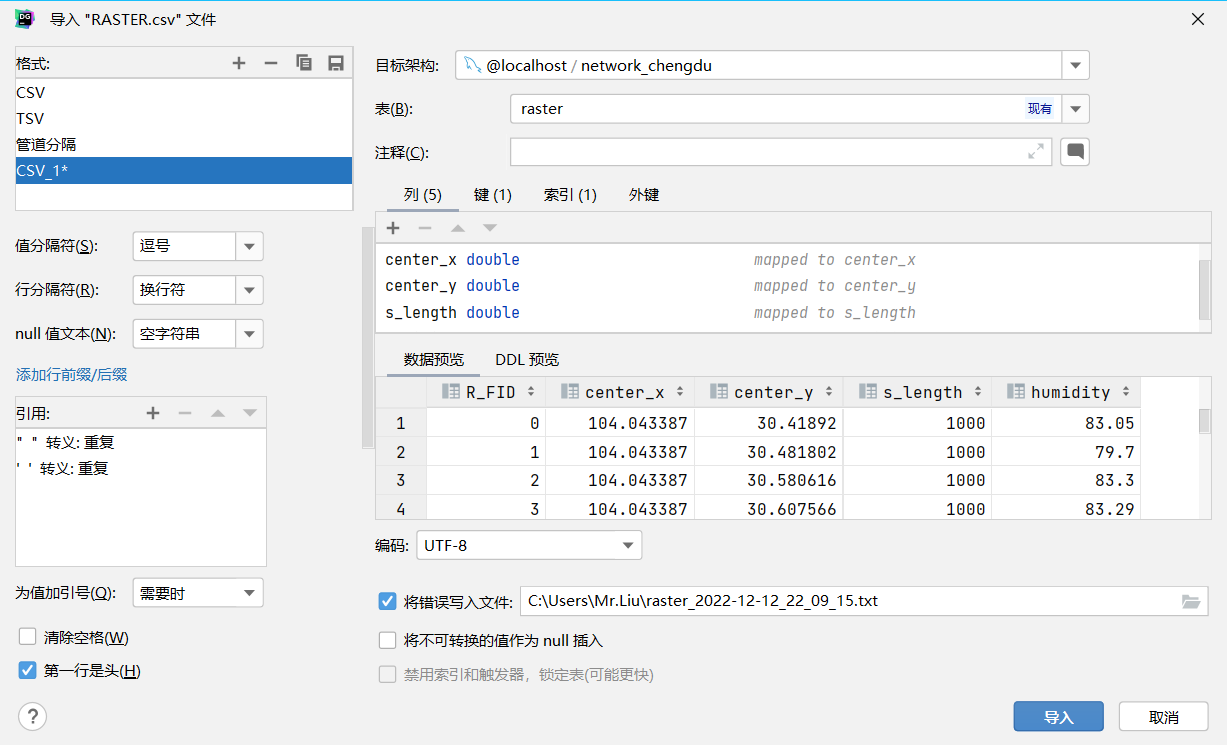


图5.1.2：RASTER导入参数设置

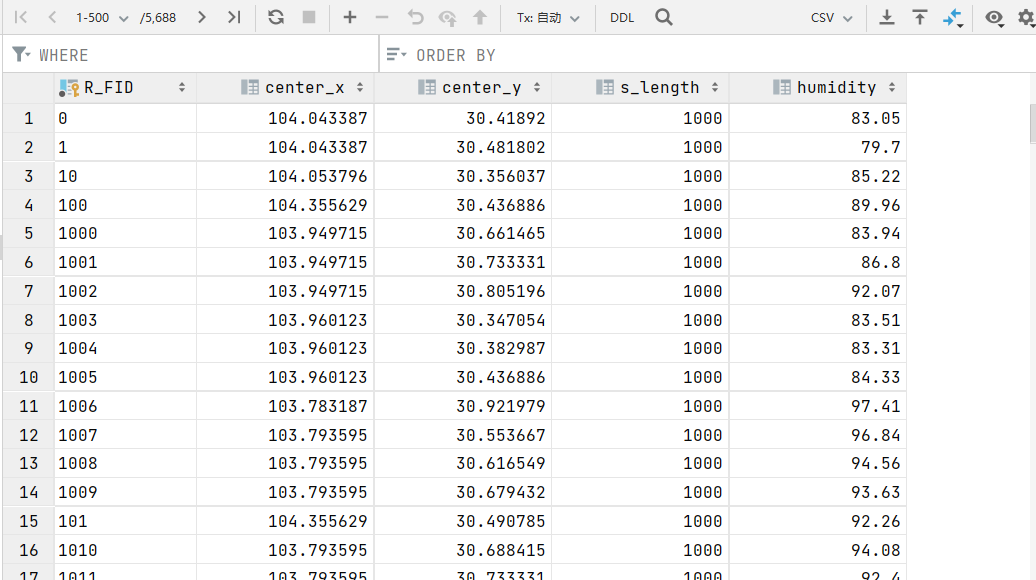


图5.1.3：RASTER数据表

NODE原数据共140条，导入后：

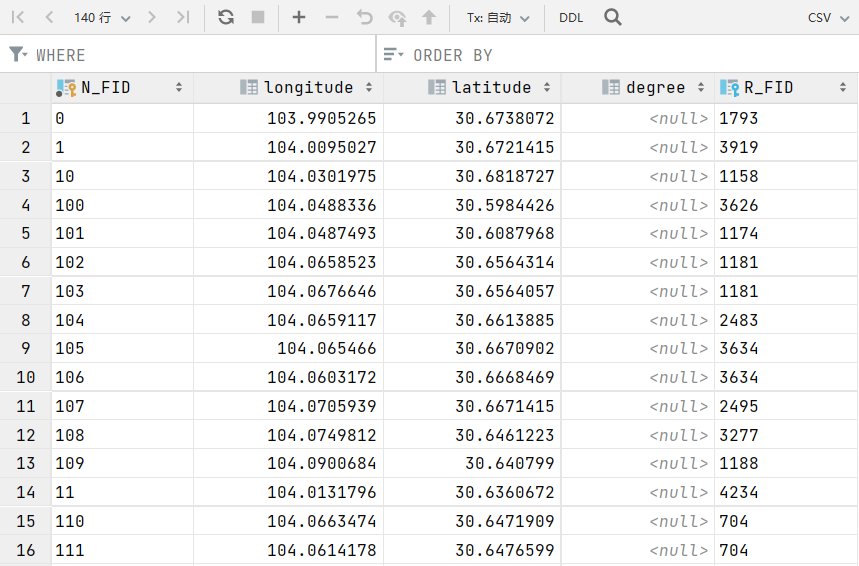


图5.1.4：NODE数据表

EDGE原数据共246条，导入后：

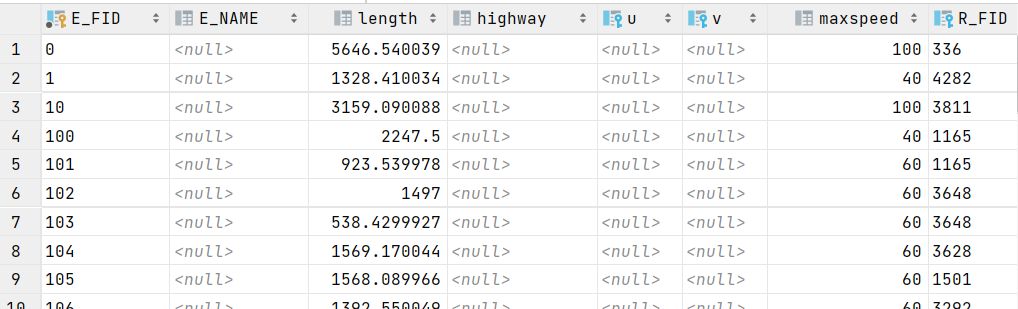


图5.1.5：EDGE数据表

CAMERA原数据5072共条，导入后：

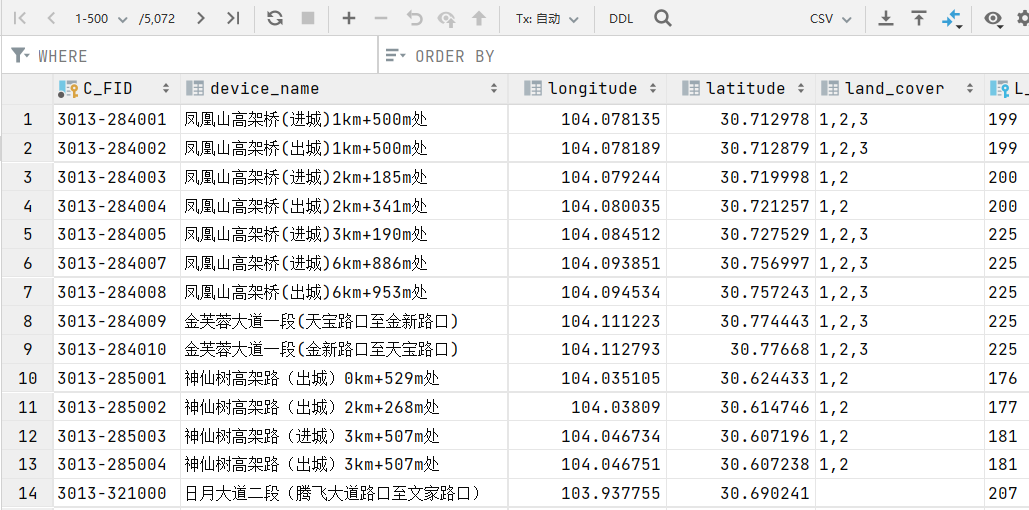


图5.1.6：CAMERA数据表



图5.1.7：Suser数据表

5.2、公共类代码设计

本系统公共类文件为CommDb.cs，其中包含TempData和CommDbOP两个类。其中TempData包含三个静态字段用于在不同窗体间传递数据；CommDb为通用数据库操作的类，包含执行INSERT、UPDATE、DELETE操作等。

App.config：

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<configuration>**
3. **<connectionStrings>**
4. **<add** name="myconnstring" connectionString="server=localhost;user=root;database=network\_chengdu;port=3306;password=123;" providerName=" MySql.Data.MySqlClient " **/>**
5. **<add** name="WindowsFormsTest\_ADO.NET.Properties.Settings.SMKConnectionString" connectionString=" server=localhost;user=root;database=network\_chengdu;port=3306;password=123;" providerName=" MySql.Data.MySqlClient " **/>**
6. **</connectionStrings>**
7. **<startup>**
8. **<supportedRuntime** version="v4.0" sku=".NETFramework,Version=v4.7.2" **/>**
9. **</startup>**
10. **<runtime>**
11. **<assemblyBinding** xmlns="urn:schemas-microsoft-com:asm.v1"**>**
12. **<dependentAssembly>**
13. **<assemblyIdentity** name="System.Runtime.CompilerServices.Unsafe" publicKeyToken="b03f5f7f11d50a3a" culture="neutral" **/>**
14. **<bindingRedirect** oldVersion="0.0.0.0-5.0.0.0" newVersion="5.0.0.0" **/>**
15. **</dependentAssembly>**
16. **</assemblyBinding>**
17. **</runtime>**
18. **</configuration>**

CommDb.cs：

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Linq;
4. **using** System.Text;
5. **using** System.Threading.Tasks;
6. **using** System.Data;
7. **using** MySql.Data;
8. **using** MySql.Data.MySqlClient;
10. **namespace** Network\_Chengdu
11. {
12. **class** CommDB
13. {
14. **public** CommDB() //默认构造函数
15. { }
16. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
17. //返回SELECT语句执行后记录集中的行数
18. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
19. **public** **int** Rownum(**string** sql)
20. { //sql参数指出SQL语句
21. **int** i = 0;
22. **string** mystr = System.Configuration.ConfigurationManager.
23. ConnectionStrings["myconnstring"].ToString();
24. //从App.config文件获取连接字符串
25. MySqlConnection myconn = **new** MySqlConnection();
26. myconn.ConnectionString = mystr;
27. myconn.Open();
28. MySqlCommand mycmd = **new** MySqlCommand(sql, myconn);
29. MySqlDataReader myreader = mycmd.ExecuteReader();
30. **while** (myreader.Read()) //循环读取信息
31. { i++; }
32. myconn.Close();
33. **return** i; //返回读取的行数
34. }
35. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
36. //返回SELECT语句执行后唯一行的唯一字段值
37. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
38. **public** **string** Returnafield(**string** sql)
39. { //sql指出SQL语句
40. **string** fn;
41. **string** mystr = System.Configuration.ConfigurationManager.
42. ConnectionStrings["myconnstring"].ToString();
43. //从App.config文件获取连接字符串
44. MySqlConnection myconn = **new** MySqlConnection();
45. myconn.ConnectionString = mystr;
46. myconn.Open();
47. MySqlCommand mycmd = **new** MySqlCommand(sql, myconn);
48. MySqlDataReader myreader = mycmd.ExecuteReader();
49. myreader.Read();
50. fn = myreader[0].ToString().Trim();
51. myconn.Close();
52. **return** fn; //返回读取的数据
53. }
54. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
55. //执行SQL语句，返回是否成功执行。SQL语句最好是如下：
56. //UPDATE 表名 SET 字段名=value,字段名=value WHERE 字段名=value
57. //DELETE FROM 表名 WHERE 字段名=value
58. //INSERT INTO 表名 (字段名,字段名) values (value,value)
59. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
60. **public** **void** ExecuteNonQuery(**string** sql)
61. {
62. **string** mystr = System.Configuration.ConfigurationManager.
63. ConnectionStrings["myconnstring"].ToString();
64. MySqlConnection myconn = **new** MySqlConnection();
65. myconn.ConnectionString = mystr;
66. myconn.Open();
67. MySqlCommand mycmd = **new** MySqlCommand(sql, myconn);
68. mycmd.ExecuteNonQuery();
69. myconn.Close();
70. }
71. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
72. //执行SELECT语句，返回DataSet对象
73. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
74. **public** DataSet ExecuteQuery(**string** sql, **string** tname)
75. {
76. **string** mystr = System.Configuration.ConfigurationManager.
77. ConnectionStrings["myconnstring"].ToString();
78. MySqlConnection myconn = **new** MySqlConnection();
79. myconn.ConnectionString = mystr;
80. myconn.Open();
81. MySqlDataAdapter myda = **new** MySqlDataAdapter(sql, myconn);
82. DataSet myds = **new** DataSet();
83. myda.Fill(myds, tname);
84. myconn.Close();
85. **return** myds;
86. }
87. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
88. //执行SELECT语句，返回聚合函数结果
89. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
90. **public** **string** ExecuteAggregateQuery(**string** sql)
91. {
92. **string** jg;
93. **string** mystr = System.Configuration.ConfigurationManager.
94. ConnectionStrings["myconnstring"].ToString();
95. MySqlConnection myconn = **new** MySqlConnection();
96. myconn.ConnectionString = mystr;
97. myconn.Open();
98. MySqlCommand mycmd = **new** MySqlCommand();
99. mycmd.CommandText = sql;
100. mycmd.Connection = myconn;
101. jg = mycmd.ExecuteScalar().ToString();
102. myconn.Close();
103. **return** jg;
104. }
105. }
106. **public** **class** TempData
107. {
108. **public** **static** **int** tag; //操作标志: 1：添加,2:修改
109. **public** **static** **string** sql; //窗体之间传递的SQL语句
110. **public** **static** **string** userlevel; //当前用户级别
111. }
112. }

5.3、登陆界面代码设计

登陆界面窗口如图5.3.1所示，主要包含文本框、命令按钮以及一些个性化设置。该窗体会提示用户输入一定的用户名，具有登陆权限的用户可登陆系统。同时登陆系统将不同权限的用户进行区分，操作员只可以使用查询、统计信息和帮助功能，管理员可以使用系统全部功能。



图5.3.1：登陆界面初始图

例如，启动Login窗口后，使用管理员账户进行登陆操作，如图5.3.2所示，点击“登入”命令按钮即可进入本系统。

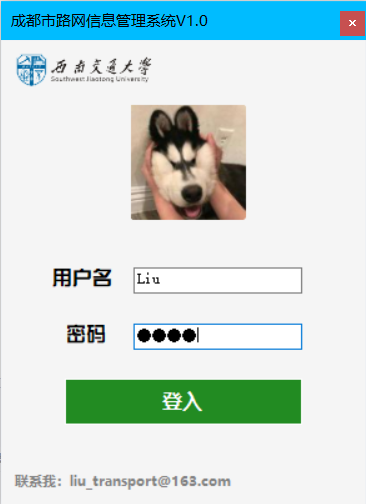


图5.3.2：登陆界面图

Login.cs：

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** Login : Form
14. {
15. **public** Login()
16. {
17. InitializeComponent();
18. }
20. **private** **void** button1\_Click(**object** sender, EventArgs e)
21. {
22. CommDB mydb = **new** CommDB();
23. **string** mystr,mystr\_user;
24. mystr = "SELECT \* FROM SUser WHERE "
25. + "用户名='" + textBox1.Text.Trim()
26. + "' AND 密码='" + textBox2.Text.Trim() + "'";
27. //mystr\_user = "SELECT \* FROM SUser WHERE "+ "用户名='" + textBox1.Text.Trim() + "'";
28. **int** i = mydb.Rownum(mystr);
29. //int j = mydb.Rownum(mystr\_user);
30. **if** (i == 0)
31. {
32. **if** (textBox1.Text == "")
33. MessageBox.Show("请输入用户名");
34. **else**
35. **if** (textBox2.Text == "")
36. MessageBox.Show("请输入密码");
37. **else**
38. MessageBox.Show("用户名或密码错误！", "操作提示");
39. **return**;
40. }
41. **else**
42. {
43. mystr = "SELECT 级别 FROM SUser WHERE "
44. + "用户名='" + textBox1.Text.Trim()
45. + "' AND 密码='" + textBox2.Text.Trim() + "'";
46. TempData.userlevel = mydb.Returnafield(mystr);
47. Form myform = **new** Main();
48. myform.ShowDialog();
49. **this**.Close();
50. }
51. }
52. }
53. }

5.4、主菜单窗体设计

主菜单窗体Main为一个多文档窗体，由menuStrip与statusStrip控件组成，管理员登录界面如图5.4.1所示。

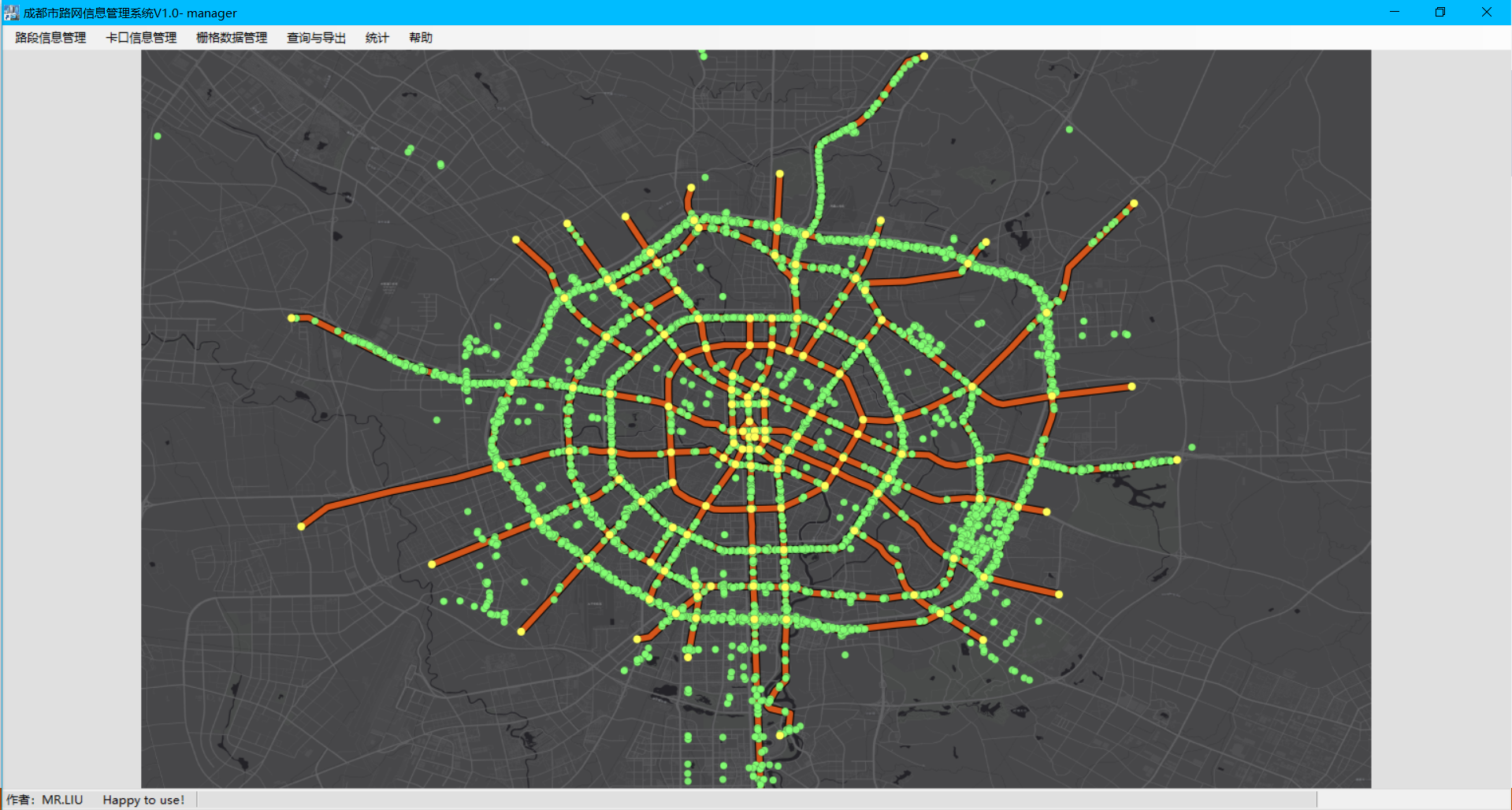


图5.4.1：管理员登录窗口界面

其中menuStrip中分别设置路段信息管理、卡口信息管理、栅格信息管理三大管理功能，查询与导出核心功能、统计与帮助两个附加功能。其中信息管理中，下拉菜单为插入、编辑与删除三种模式，如图5.4.2。



图5.4.2：管理功能下拉菜单

同时对于非管理员用户界面如图5.4.3：



图5.4.3：非管理员主界面窗口

Login.cs：

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** Main : Form
14. {
15. **public** Main()
16. {
17. InitializeComponent();
18. }
19. **private** **void** Main\_Load\_1(**object** sender, EventArgs e)
20. {
22. **if** (TempData.userlevel == "管理员") //管理员限制
23. {
24. menuStrip1.Items["manage\_NETWORK"].Enabled = **true**;
25. menuStrip1.Items["manage\_CAMERA"].Enabled = **true**;
26. menuStrip1.Items["manage\_RASTER"].Enabled = **true**;
27. menuStrip1.Items["find\_output"].Enabled = **true**;
28. menuStrip1.Items["HELP"].Enabled = **true**;
29. **this**.Text += "- manager";
30. }
31. **else** //操作员功能限制
32. {
33. menuStrip1.Items["manage\_NETWORK"].Enabled = **false**;
34. menuStrip1.Items["manage\_CAMERA"].Enabled = **false**;
35. menuStrip1.Items["manage\_RASTER"].Enabled = **false**;
36. menuStrip1.Items["find\_output"].Enabled = **true**;
37. menuStrip1.Items["HELP"].Enabled = **true**;
38. }
39. **this**.toolStripStatusLabel3.Text = "系统当前时间：" + DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss");
41. //对timer1进行相关设置
42. **this**.timer1.Interval = 1000;
43. **this**.timer1.Start();
44. }
46. **private** **void** 添加数据\_Click(**object** sender, EventArgs e)
47. {
48. Form myform = **new** add\_network();
49. myform.TopLevel = **false**;
50. myform.Parent = **this**; //建立父子窗体关系
51. myform.Show();
52. myform.BringToFront();
53. }
55. **private** **void** 更新路段信息\_Click(**object** sender, EventArgs e)
56. {
57. Form myform = **new** update\_network();
58. myform.TopLevel = **false**;
59. myform.Parent = **this**; //建立父子窗体关系
60. myform.Show();
61. myform.BringToFront();
62. }
64. **private** **void** 删除路段信息\_Click(**object** sender, EventArgs e)
65. {
66. Form myform = **new** update\_network();
67. myform.TopLevel = **false**;
68. myform.Parent = **this**; //建立父子窗体关系
69. myform.Show();
70. myform.BringToFront();
71. }
73. **private** **void** 添加卡口信息\_Click(**object** sender, EventArgs e)
74. {
75. Form myform = **new** add\_camera();
76. myform.TopLevel = **false**;
77. myform.Parent = **this**; //建立父子窗体关系
78. myform.Show();
79. myform.BringToFront();
80. }
82. **private** **void** 更新卡口信息\_Click(**object** sender, EventArgs e)
83. {
84. Form myform = **new** update\_camera();
85. myform.TopLevel = **false**;
86. myform.Parent = **this**; //建立父子窗体关系
87. myform.Show();
88. myform.BringToFront();
89. }
91. **private** **void** 删除卡口信息\_Click(**object** sender, EventArgs e)
92. {
93. Form myform = **new** update\_camera();
94. myform.TopLevel = **false**;
95. myform.Parent = **this**; //建立父子窗体关系
96. myform.Show();
97. myform.BringToFront();
98. }
100. **private** **void** 添加栅格数据\_Click(**object** sender, EventArgs e)
101. {
102. Form myform = **new** add\_raster();
103. myform.TopLevel = **false**;
104. myform.Parent = **this**; //建立父子窗体关系
105. myform.Show();
106. myform.BringToFront();
107. }
109. **private** **void** 修改栅格数据\_Click(**object** sender, EventArgs e)
110. {
111. Form myform = **new** update\_raster();
112. myform.TopLevel = **false**;
113. myform.Parent = **this**; //建立父子窗体关系
114. myform.Show();
115. myform.BringToFront();
116. }
118. **private** **void** 删除栅格数据\_Click(**object** sender, EventArgs e)
119. {
120. Form myform = **new** update\_raster();
121. myform.TopLevel = **false**;
122. myform.Parent = **this**; //建立父子窗体关系
123. myform.Show();
124. myform.BringToFront();
125. }
127. **private** **void** 查询\_Click(**object** sender, EventArgs e)
128. {
129. Form myform = **new** find();
130. myform.TopLevel = **false**;
131. myform.Parent = **this**; //建立父子窗体关系
132. myform.Show();
133. myform.BringToFront();
134. }
136. **private** **void** 导出\_Click(**object** sender, EventArgs e)
137. {
138. Form myform = **new** find();
139. myform.TopLevel = **false**;
140. myform.Parent = **this**; //建立父子窗体关系
141. myform.Show();
142. myform.BringToFront();
143. }
145. **private** **void** timer1\_Tick(**object** sender, EventArgs e)
146. {
148. **this**.toolStripStatusLabel3.Text = "当前时间：" + DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss");
150. }
152. **private** **void** 关于ToolStripMenuItem\_Click(**object** sender, EventArgs e)
153. {
154. Form myform = **new** about();
155. myform.TopLevel = **false**;
156. myform.Parent = **this**; //建立父子窗体关系
157. myform.Show();
158. myform.BringToFront();
159. }
161. **private** **void** 联系方式ToolStripMenuItem\_Click(**object** sender, EventArgs e)
162. {
163. Form myform = **new** information();
164. myform.TopLevel = **false**;
165. myform.Parent = **this**; //建立父子窗体关系
166. myform.Show();
167. myform.BringToFront();
168. }

171. **private** **void** 基本统计信息ToolStripMenuItem\_Click(**object** sender, EventArgs e)
172. {
173. Form myform = **new** statistics();
174. myform.TopLevel = **false**;
175. myform.Parent = **this**; //建立父子窗体关系
176. myform.Show();
177. myform.BringToFront();
178. }
179. }
180. }

5.5、路段信息管理功能设计

对于路段数据信息，由添加数据信息、编辑数据信息、删除数据信息三个功能。

首先是添加数据节点功能，设计界面如图5.5.1所示

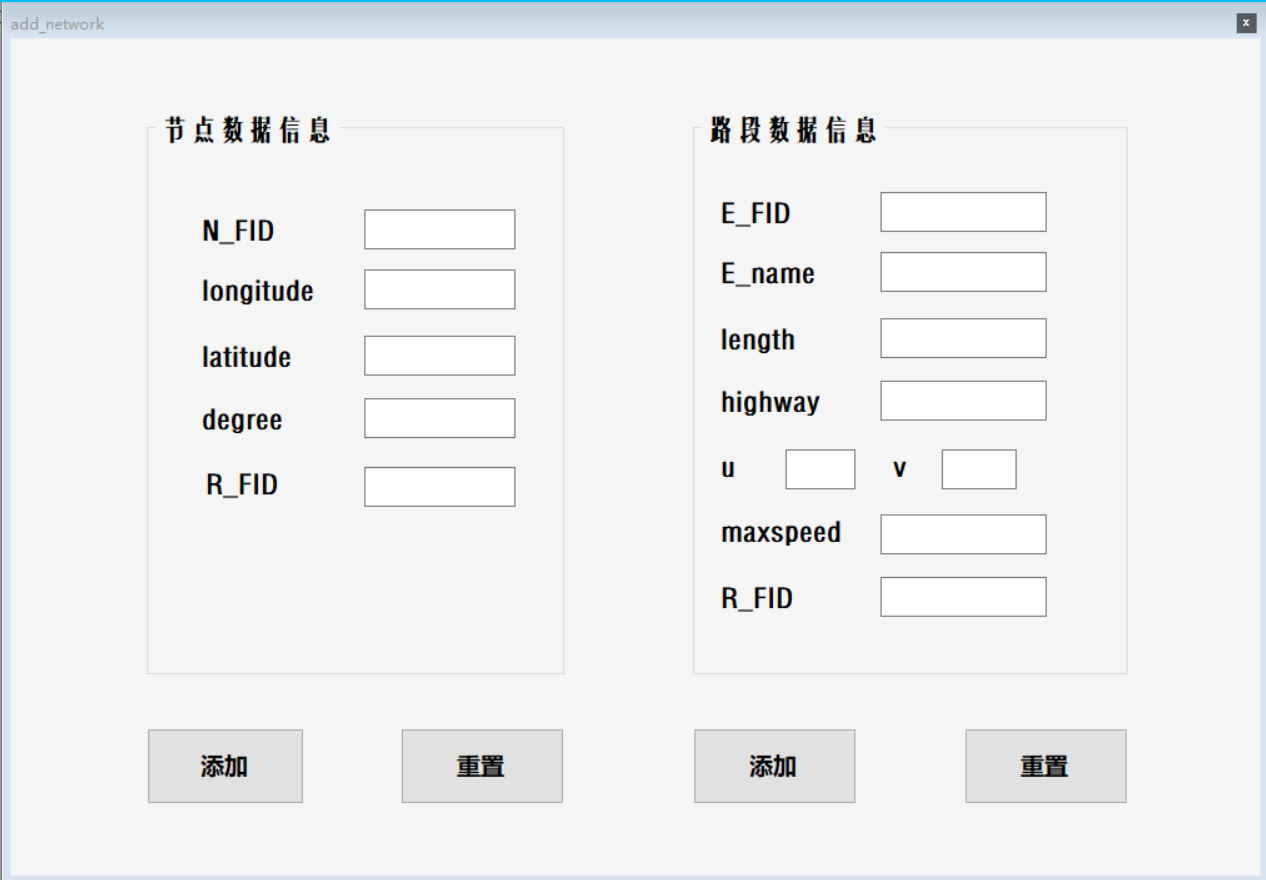


图5.5.1：添加路段信息

信息成功添加后会将出现弹窗提示，如图5.5.2：

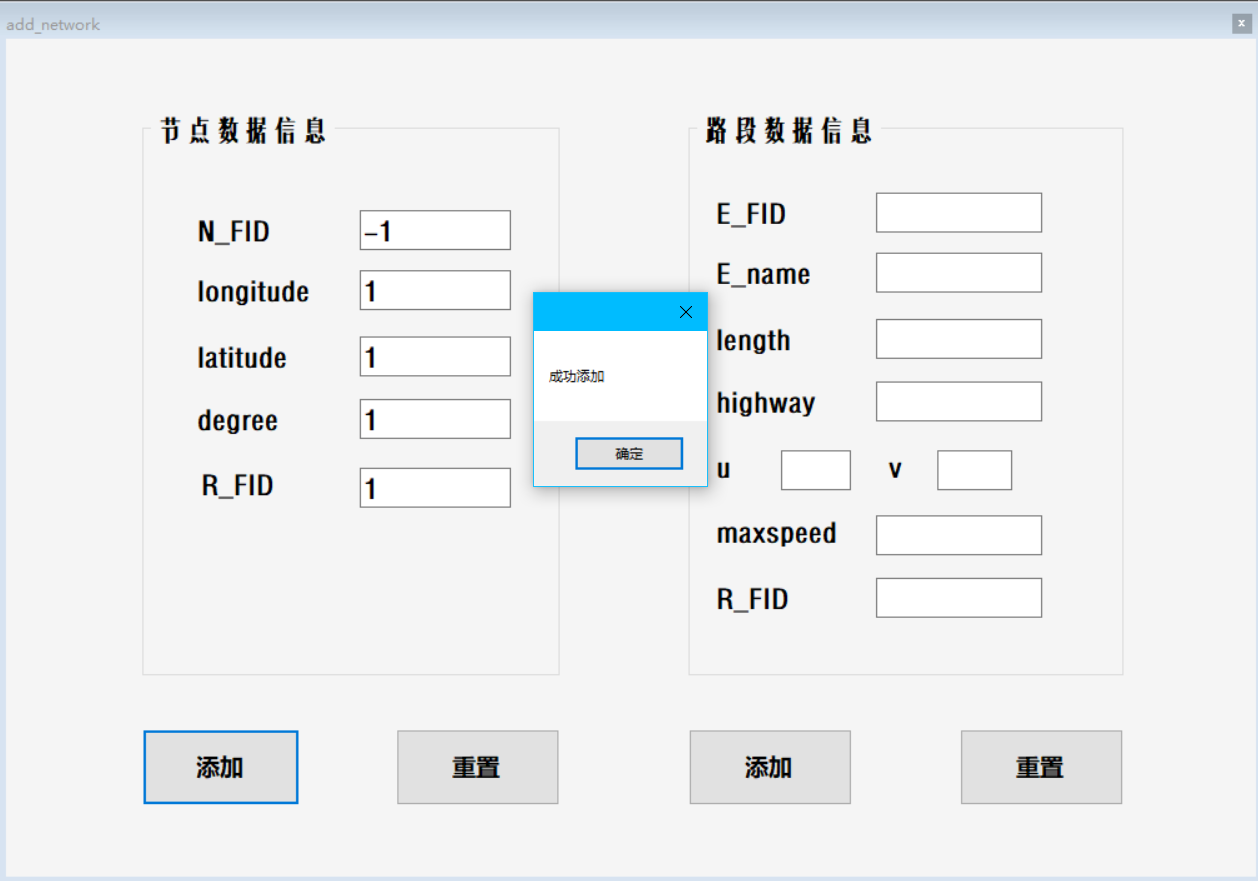


图5.5.2：添加成功后的弹窗界面

更新路段窗口设计如图5.5.3所示：



图5.5.3：更新路段数据窗口

当输入选择数据后会弹窗，如图5.5.4。

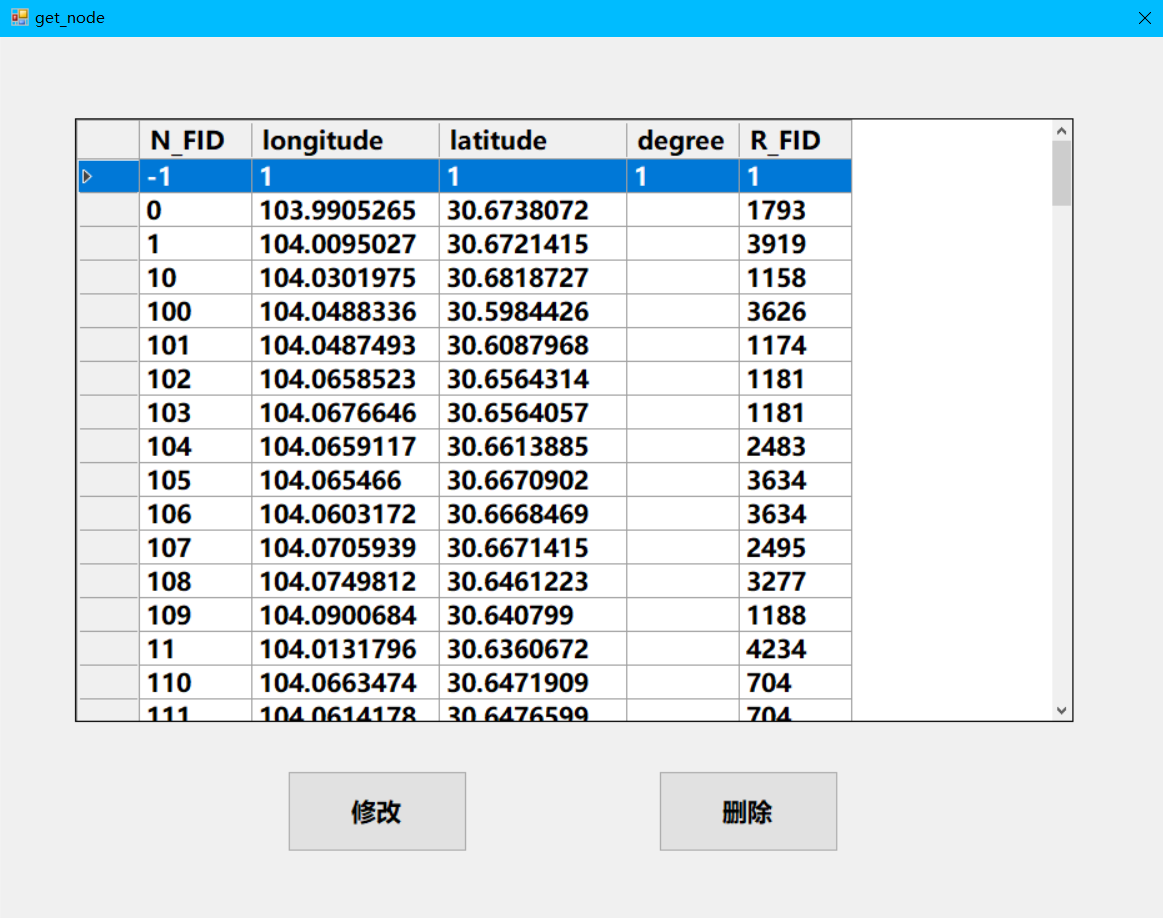


图5.5.4：数据展示设计窗口

选择刚添加过的第一条数据进行修改，如图5.5.5，



图5.5.5：修改节点信息展示设计窗口

单击确认后，数据修改，如图5.5.6：



图5.5.6：修改后节点设计窗口

删除模块即点击删除，如图5.5.7：

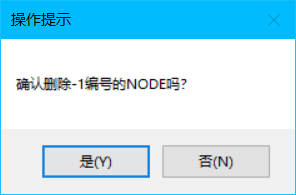


图5.5.7：点击删除或弹窗

单击是，删除后数据展示，如图5.5.8：

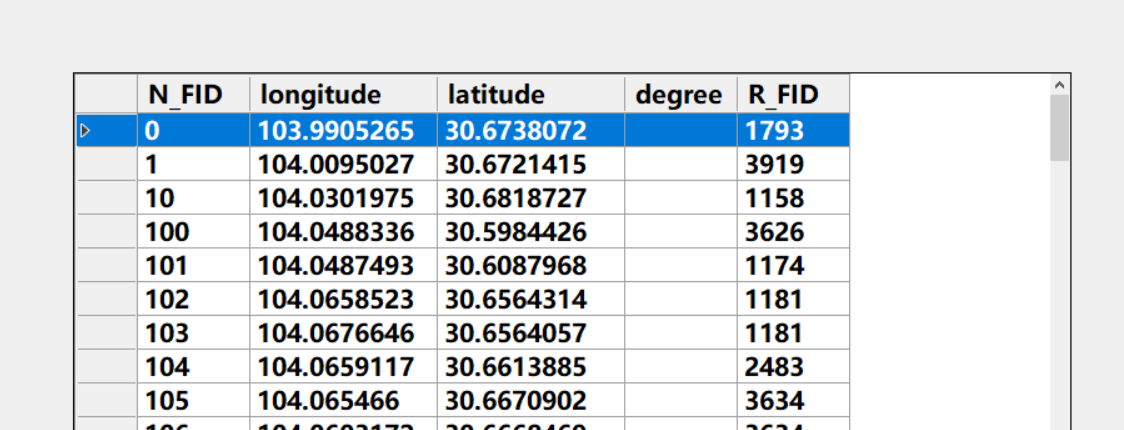


图5.5.8：删除后数据展示

add\_network.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** add\_network : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** add\_network()
17. {
18. InitializeComponent();
19. }
21. **private** **void** button1\_Click\_1(**object** sender, EventArgs e)
22. {
23. **if** (textBox1.Text == "" || textBox2.Text == ""
24. || textBox3.Text == "" || textBox5.Text == "")
25. {
26. MessageBox.Show("除degree外所有数据项必须输入", "操作提示");
27. **return**;
28. }
29. **string** mystr = "SELECT \* FROM node WHERE N\_FID='" + textBox1.Text.Trim() + "'";
30. **int** i = mydb.Rownum(mystr);
31. **if** (i != 0)
32. {
33. MessageBox.Show("输入的NODE编号已存在", "操作提示");
34. **return**;
35. }
37. **try**
38. {
39. mystr = "INSERT INTO node(N\_FID,longitude,latitude,degree,R\_FID) VALUES('"
40. + textBox1.Text.Trim() + "','"
41. + textBox2.Text.Trim() + "','"
42. + textBox3.Text.Trim() + "','"
43. + textBox4.Text.Trim() + "','"
44. + textBox5.Text.Trim() + "')";
45. mydb.ExecuteNonQuery(mystr);
46. MessageBox.Show("成功添加");
47. }
48. **catch** (Exception ex)
49. {
50. MessageBox.Show("输入的NODE数据有错误：" + ex.Message);
51. **return**;
52. }
53. button2\_Click\_1(sender, e);
54. }
56. **private** **void** button3\_Click(**object** sender, EventArgs e)
57. {
58. **if** (textBox6.Text == "" || textBox7.Text == ""
59. || textBox8.Text == "" || textBox9.Text == ""
60. || textBox10.Text == "" || textBox11.Text == ""
61. || textBox12.Text == "" || textBox13.Text == "")
62. {
63. MessageBox.Show("所有数据项必须输入", "操作提示");
64. **return**;
65. }
66. **string** mystr = "SELECT \* FROM node WHERE N\_FID='" + textBox6.Text.Trim() + "'";
67. **int** i = mydb.Rownum(mystr);
68. **if** (i != 0)
69. {
70. MessageBox.Show("输入的EDGE编号已存在", "操作提示");
71. **return**;
72. }
74. **try**
75. {
76. mystr = "INSERT INTO edge(E\_FID,E\_name,length,highway,u,v,maxspeed,R\_FID) VALUES('"
77. + textBox6.Text.Trim() + "','"
78. + textBox7.Text.Trim() + "','"
79. + textBox8.Text.Trim() + "','"
80. + textBox9.Text.Trim() + "','"
81. + textBox10.Text.Trim() + "','"
82. + textBox11.Text.Trim() + "','"
83. + textBox12.Text.Trim() + "','"
84. + textBox13.Text.Trim() + "')";
85. mydb.ExecuteNonQuery(mystr);
86. MessageBox.Show("成功添加");
87. }
88. **catch** (Exception ex)
89. {
90. MessageBox.Show("输入的EDGE数据有错误：" + ex.Message);
91. **return**;
92. }
93. button4\_Click(sender, e);
94. }

97. **private** **void** button2\_Click\_1(**object** sender, EventArgs e)
98. {
99. textBox1.Text = ""; textBox2.Text = "";
100. textBox3.Text = ""; textBox4.Text = "";
101. textBox5.Text = "";
102. textBox1.Focus();
103. }
105. **private** **void** button4\_Click(**object** sender, EventArgs e)
106. {
107. textBox6.Text = ""; textBox7.Text = "";
108. textBox8.Text = ""; textBox9.Text = "";
109. textBox10.Text = ""; textBox11.Text = "";
110. textBox12.Text = ""; textBox13.Text = "";
111. textBox6.Focus();
112. }
113. }
114. }

update\_network.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** update\_network : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** update\_network()
17. {
18. InitializeComponent();
19. }
21. **private** **void** button1\_Click(**object** sender, EventArgs e)
22. {
23. **string** condstr = "";
24. **if** (textBox1.Text != "")
25. condstr += "N\_FID ='" + textBox1.Text.Trim() + "'";
27. **if** (textBox2.Text != "")
28. {
29. **if** (condstr != "")
30. condstr += " and ";
32. condstr += "longitude = " + **double**.Parse(textBox2.Text.Trim()) + "";
33. }
35. **if** (textBox3.Text != "")
36. {
37. **if** (condstr != "")
38. condstr += " and ";
40. condstr += "latitude = " + **double**.Parse(textBox3.Text.Trim()) + "";
41. }
43. **if** (textBox4.Text != "")
44. {
45. **if** (condstr != "")
46. condstr += " and ";
48. condstr += "degree = " + **int**.Parse(textBox4.Text.Trim()) + "";
49. }
51. **if** (textBox5.Text != "")
52. {
53. **if** (condstr != "")
54. condstr += " and ";
56. condstr += "R\_FID = '" + textBox5.Text.Trim() + "'";
57. }
59. **if** (condstr != "")
60. TempData.sql = "SELECT \* FROM node WHERE " + condstr;
61. **else**
62. TempData.sql = "";
63. get\_node myform = **new** get\_node();
64. myform.ShowDialog();
65. }
67. **private** **void** button2\_Click(**object** sender, EventArgs e)
68. {
69. textBox1.Text = ""; textBox2.Text = "";
70. textBox3.Text = ""; textBox4.Text = "";
71. textBox5.Text = "";
72. textBox1.Focus();
73. }
75. **private** **void** button3\_Click(**object** sender, EventArgs e)
76. {
77. **string** condstr = "";
78. **if** (textBox6.Text != "")
79. condstr += "E\_FID = '" + textBox6.Text.Trim() + "'";
81. **if** (textBox7.Text != "")
82. {
83. **if** (condstr != "")
84. condstr += " and ";
86. condstr += "E\_name = '" + textBox7.Text.Trim() + "'";
87. }
89. **if** (textBox8.Text != "")
90. {
91. **if** (condstr != "")
92. condstr += " and ";
94. condstr += "length = " + **double**.Parse(textBox8.Text.Trim()) + "";
95. }
97. **if** (textBox9.Text != "")
98. {
99. **if** (condstr != "")
100. condstr += " and ";
102. condstr += "highway = '" + textBox9.Text.Trim() + "'";
103. }
105. **if** (textBox10.Text != "")
106. {
107. **if** (condstr != "")
108. condstr += " and ";
110. condstr += "u = '" + textBox10.Text.Trim() + "'";
111. }
113. **if** (textBox11.Text != "")
114. {
115. **if** (condstr != "")
116. condstr += " and ";
118. condstr += "v = " + **int**.Parse(textBox11.Text.Trim()) + "";
119. }
121. **if** (textBox12.Text != "")
122. {
123. **if** (condstr != "")
124. condstr += " and ";
126. condstr += "R\_FID = '" + textBox12.Text.Trim() + "'";
127. }
129. **if** (textBox13.Text != "")
130. {
131. **if** (condstr != "")
132. condstr += " and ";
134. condstr += "R\_FID = " + **double**.Parse(textBox13.Text.Trim()) + "";
135. }
136. **if** (condstr != "")
137. TempData.sql = "SELECT \* FROM edge WHERE " + condstr;
138. **else**
139. TempData.sql = "";
140. get\_edge myform = **new** get\_edge();
141. myform.ShowDialog();
142. }
144. **private** **void** button4\_Click(**object** sender, EventArgs e)
145. {
146. textBox6.Text = ""; textBox7.Text = "";
147. textBox8.Text = ""; textBox9.Text = "";
148. textBox10.Text = ""; textBox11.Text = "";
149. textBox12.Text = ""; textBox13.Text = "";
150. textBox6.Focus();
151. }
153. }
154. }

get\_node.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** get\_node : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** get\_node()
17. {
18. InitializeComponent();
19. }
21. **private** **void** get\_node\_Load(**object** sender, EventArgs e)
22. {
23. bind();
24. dataGridView1.EnableHeadersVisualStyles = **false**;
25. dataGridView1.MultiSelect = **false**;
26. dataGridView1.ReadOnly = **true**;
27. dataGridView1.Columns[0].Width = 90;
28. dataGridView1.Columns[1].Width = 150;
29. dataGridView1.Columns[2].Width = 150;
30. dataGridView1.Columns[3].Width = 90;
31. dataGridView1.Columns[4].Width = 90;
32. **if** (dataGridView1.Rows.Count > 0)
33. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
34. }
36. **private** **void** bind() //绑定数据
37. {
38. **string** mystr;
39. **if** (TempData.sql == "")
40. mystr = "SELECT \* FROM node";
41. **else**
42. mystr = TempData.sql;
43. DataSet mydataset = mydb.ExecuteQuery(mystr, "node");
44. dataGridView1.DataSource = mydataset.Tables["node"];
45. }
47. **private** **void** button2\_Click(**object** sender, EventArgs e)
48. {
49. //删除记录
50. **if** (dataGridView1.SelectedRows.Count > 0)
51. {
52. **string** spno = dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim();
53. DialogResult result = MessageBox.Show("确认删除" + spno
54. + "编号的NODE吗？", "操作提示", MessageBoxButtons.YesNo);
55. **if** (result == DialogResult.Yes)
56. {
57. **string** mystr = "DELETE from node WHERE N\_FID='" + spno + "'";
58. mydb.ExecuteNonQuery(mystr);
59. bind();
60. }
61. }
62. }
64. **private** **void** button1\_Click(**object** sender, EventArgs e)
65. {
66. **if** (dataGridView1.SelectedRows.Count > 0)
67. {
68. **string** a = dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim();
69. **string** b = dataGridView1.SelectedRows[0].Cells[1].Value.ToString().Trim();
70. **string** c = dataGridView1.SelectedRows[0].Cells[2].Value.ToString().Trim();
71. **string** d = dataGridView1.SelectedRows[0].Cells[3].Value.ToString().Trim();
72. **string** R = dataGridView1.SelectedRows[0].Cells[4].Value.ToString().Trim();
74. Form myform = **new** edit\_node(a, b, c, d, R); //带传递数据调用edit\_node的构造函数
75. myform.ShowDialog();
76. }
77. }
79. **private** **void** dataGridView1\_CellContentClick(**object** sender, DataGridViewCellEventArgs e)
80. {
81. //单击任何单元时选择该单元所在的行
82. **if** (e.RowIndex >= 0 && e.RowIndex < dataGridView1.Rows.Count)
83. dataGridView1.Rows[e.RowIndex].Selected = **true**;
84. }
86. **private** **void** get\_node\_Activated(**object** sender, EventArgs e)
87. {
88. bind();
89. **if** (dataGridView1.Rows.Count > 0)
90. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
91. }
92. }
93. }

get\_edge.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** get\_edge : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** get\_edge()
17. {
18. InitializeComponent();
19. }
21. **private** **void** get\_edge\_Load(**object** sender, EventArgs e)
22. {
23. bind();
24. dataGridView1.EnableHeadersVisualStyles = **false**;
25. dataGridView1.MultiSelect = **false**;
26. dataGridView1.ReadOnly = **true**;
27. dataGridView1.Columns[0].Width = 80;
28. dataGridView1.Columns[1].Width = 100;
29. dataGridView1.Columns[2].Width = 80;
30. dataGridView1.Columns[3].Width = 90;
31. dataGridView1.Columns[4].Width = 80;
32. dataGridView1.Columns[5].Width = 80;
33. dataGridView1.Columns[6].Width = 80;
34. dataGridView1.Columns[7].Width = 80;
35. **if** (dataGridView1.Rows.Count > 0)
36. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
37. }
38. **private** **void** bind() //绑定数据
39. {
40. **string** mystr;
41. **if** (TempData.sql == "")
42. mystr = "SELECT \* FROM edge";
43. **else**
44. mystr = TempData.sql;
45. DataSet mydataset = mydb.ExecuteQuery(mystr, "edge");
46. dataGridView1.DataSource = mydataset.Tables["edge"];
47. }
49. **private** **void** button2\_Click(**object** sender, EventArgs e)
50. {
51. //删除记录
52. **if** (dataGridView1.SelectedRows.Count > 0)
53. {
54. **string** spno = dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim();
55. DialogResult result = MessageBox.Show("确认删除" + spno
56. + "编号的EDGE吗？", "操作提示", MessageBoxButtons.YesNo);
57. **if** (result == DialogResult.Yes)
58. {
59. **string** mystr = "DELETE from edge WHERE E\_FID='" + spno + "'";
60. mydb.ExecuteNonQuery(mystr);
61. bind();
62. }
63. }
64. }
66. **private** **void** button1\_Click(**object** sender, EventArgs e)
67. {
68. **if** (dataGridView1.SelectedRows.Count > 0)
69. {
70. **string** a = dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim();
71. **string** name = dataGridView1.SelectedRows[0].Cells[1].Value.ToString().Trim();
72. **string** length = dataGridView1.SelectedRows[0].Cells[2].Value.ToString().Trim();
73. **string** h = dataGridView1.SelectedRows[0].Cells[3].Value.ToString().Trim();
74. **string** u = dataGridView1.SelectedRows[0].Cells[4].Value.ToString().Trim();
75. **string** v = dataGridView1.SelectedRows[0].Cells[5].Value.ToString().Trim();
76. **string** speed = dataGridView1.SelectedRows[0].Cells[6].Value.ToString().Trim();
77. **string** R = dataGridView1.SelectedRows[0].Cells[7].Value.ToString().Trim();
79. Form myform = **new** edit\_edge(a, name, length, h, u, v, speed, R); //带传递数据调用edit\_node的构造函数
80. myform.ShowDialog();
81. }
82. }
84. **private** **void** dataGridView1\_CellContentClick(**object** sender, DataGridViewCellEventArgs e)
85. {
86. **if** (e.RowIndex >= 0 && e.RowIndex < dataGridView1.Rows.Count)
87. dataGridView1.Rows[e.RowIndex].Selected = **true**;
88. }
90. **private** **void** get\_edge\_Activated(**object** sender, EventArgs e)
91. {
92. bind();
93. **if** (dataGridView1.Rows.Count > 0)
94. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
95. }
96. }
97. }

edit\_node.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
14. **public** partial **class** edit\_node : Form
15. {
16. CommDB mydb = **new** CommDB();
18. **string** FID1, LON;
19. **string** LAT, DE;
20. **string** FID2;
22. **public** edit\_node(**string** fid1, **string** longitude, **string** latitude, **string** degree, **string** R\_FID)
23. {
24. InitializeComponent();
25. FID1 = fid1;
26. LON = longitude;
27. LAT = latitude;
28. DE = degree;
29. FID2 = R\_FID;
30. }
32. **private** **void** edit\_node\_Load(**object** sender, EventArgs e)
33. {
34. textBox1.Text = FID1;
35. textBox1.Enabled = **false**;
36. textBox2.Text = LON;
37. textBox3.Text = LAT;
38. textBox4.Text = DE;
39. textBox5.Text = FID2;
40. }
41. **private** **void** button1\_Click(**object** sender, EventArgs e)
42. {
43. **string** mystr;
44. **if** (textBox1.Text == "" || textBox2.Text == ""
45. || textBox3.Text == "" || textBox4.Text == ""
46. || textBox5.Text == "")
47. {
48. MessageBox.Show("所有数据项必须输入");
49. **return**;
50. }
51. **try**
52. {
53. mystr = "UPDATE node SET longitude="
54. + **double**.Parse(textBox2.Text.Trim()) + ",latitude="
55. + **double**.Parse(textBox3.Text.Trim()) + ",degree="
56. + **int**.Parse(textBox4.Text.Trim()) + ",R\_FID="
57. + textBox5.Text.Trim() + " WHERE N\_FID='"
58. + FID1 + "'";
59. mydb.ExecuteNonQuery(mystr);
60. }
61. **catch** (Exception ex)
62. {
63. MessageBox.Show("修改的节点数据有错误：" + ex.Message);
64. **return**;
65. }
66. **this**.Close();
67. }
68. **private** **void** button2\_Click(**object** sender, EventArgs e)
69. {
70. **this**.Close();
71. }
73. }
74. }

edit\_edge.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** edit\_edge : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **string** FID1, NAME, LENGTH, HIGHWAY, U, V, MAXSPEED, FID2;
18. **private** **void** edit\_edge\_Load(**object** sender, EventArgs e)
19. {
20. textBox6.Text = FID1;
21. textBox6.Enabled = **false**;
22. textBox7.Text = NAME;
23. textBox8.Text = LENGTH;
24. textBox9.Text = HIGHWAY;
25. textBox10.Text = U;
26. textBox11.Text = V;
27. textBox12.Text = MAXSPEED;
28. textBox13.Text = FID2;
29. }
31. **public** edit\_edge(**string** a, **string** name, **string** length, **string** h, **string** u, **string** v, **string** speed, **string** R)
32. {
33. InitializeComponent();
34. FID1 = a;
35. NAME = name;
36. LENGTH = length;
37. HIGHWAY = h;
38. U = u;
39. V = v;
40. MAXSPEED = speed;
41. FID2 = R;
42. }
44. **private** **void** button4\_Click(**object** sender, EventArgs e)
45. {
46. **this**.Close();
47. }
49. **private** **void** button3\_Click(**object** sender, EventArgs e)
50. {
51. **string** mystr;
52. **if** (textBox6.Text == "" || textBox7.Text == ""
53. || textBox8.Text == "" || textBox9.Text == ""
54. || textBox10.Text == "" || textBox11.Text == ""
55. || textBox12.Text == "" || textBox13.Text == "")
56. {
57. MessageBox.Show("所有数据项必须输入");
58. **return**;
59. }
60. **try**
61. {
62. mystr = "UPDATE edge SET E\_name='"
63. + textBox7.Text.Trim() + "',length="
64. + **double**.Parse(textBox8.Text.Trim()) + ",highway='"
65. + textBox9.Text.Trim() + "',u='"
66. + textBox10.Text.Trim() + "',v='"
67. + textBox11.Text.Trim() + "',highway="
68. + **int**.Parse(textBox12.Text.Trim()) + ",R\_FID="
69. + textBox13.Text.Trim() + " WHERE E\_FID='"
70. + FID1 + "'";
71. mydb.ExecuteNonQuery(mystr);
72. }
73. **catch** (Exception ex)
74. {
75. MessageBox.Show("修改的路段数据有错误：" + ex.Message);
76. **return**;
77. }
78. **this**.Close();
79. }
80. }
81. }

5.6、卡口信息管理功能设计

卡口信息管理功能与路段管理信息功能原理相似，下面只做操作展示与设计源代码。



图5.6.1初始化界面

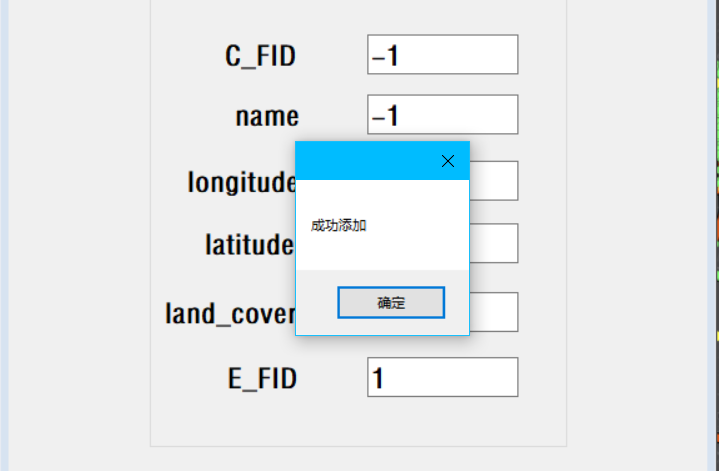


图5.6.2：成功添加弹窗

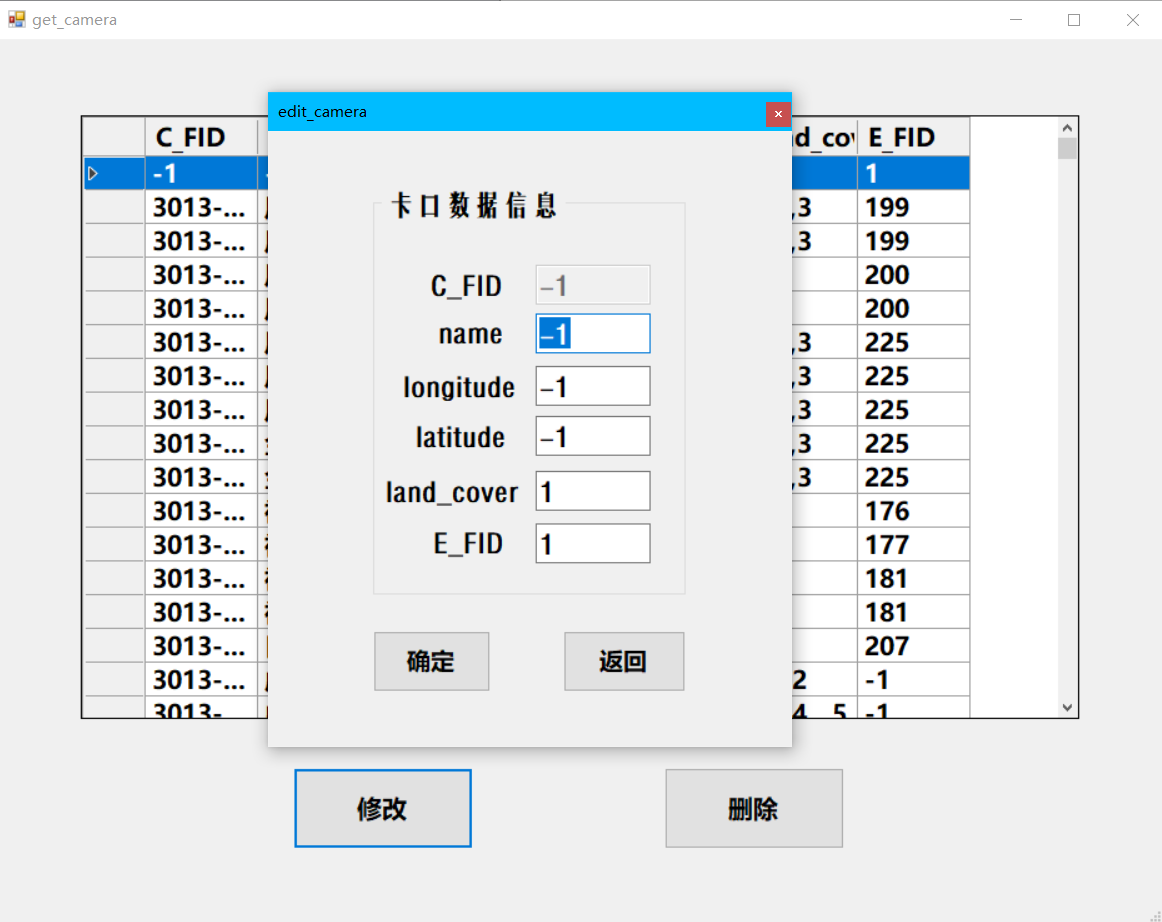


图5.6.3：修改功能演示

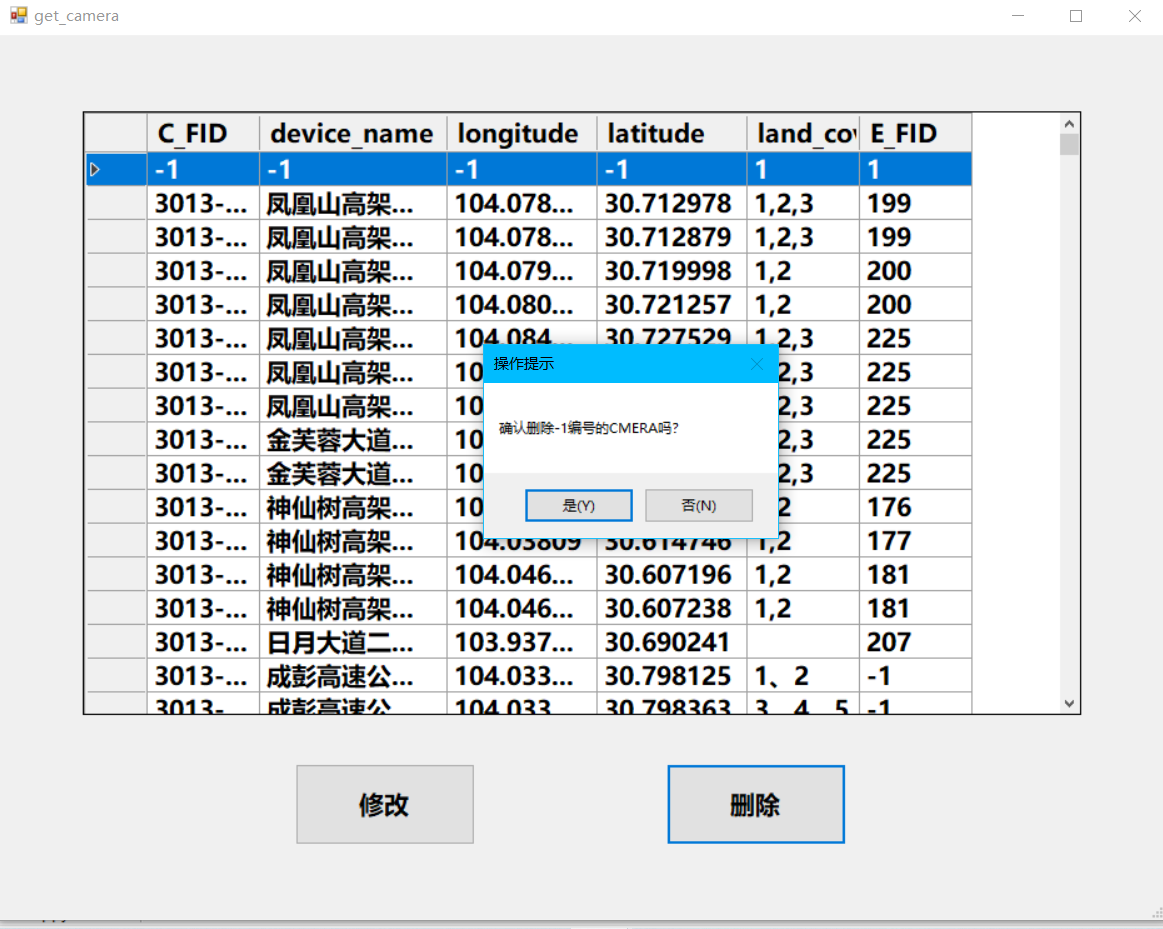


图5.6.4：删除功能演示

add\_camera.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** add\_camera : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** add\_camera()
17. {
18. InitializeComponent();
19. }
21. **private** **void** button1\_Click(**object** sender, EventArgs e)
22. {
23. **if** (textBox1.Text == "" || textBox2.Text == ""|| textBox3.Text == ""
24. || textBox4.Text == "" || textBox5.Text == "" || textBox6.Text == "")
25. {
26. MessageBox.Show("除degree外所有数据项必须输入", "操作提示");
27. **return**;
28. }
29. **string** mystr = "SELECT \* FROM camera WHERE C\_FID='" + textBox1.Text.Trim() + "'";
30. **int** i = mydb.Rownum(mystr);
31. **if** (i != 0)
32. {
33. MessageBox.Show("输入的CAMERA编号已存在", "操作提示");
34. **return**;
35. }
37. **try**
38. {
39. mystr = "INSERT INTO CAMERA(C\_FID,device\_name,longitude,latitude,land\_cover,E\_FID) VALUES('"
40. + textBox1.Text.Trim() + "','"
41. + textBox2.Text.Trim() + "','"
42. + textBox3.Text.Trim() + "','"
43. + textBox4.Text.Trim() + "','"
44. + textBox5.Text.Trim() + "','"
45. + textBox6.Text.Trim() + "')";
46. mydb.ExecuteNonQuery(mystr);
47. MessageBox.Show("成功添加");
48. }
49. **catch** (Exception ex)
50. {
51. MessageBox.Show("输入的CAMERA数据有错误：" + ex.Message);
52. **return**;
53. }
54. button2\_Click(sender, e);
55. }
57. **private** **void** button2\_Click(**object** sender, EventArgs e)
58. {
59. textBox1.Text = ""; textBox2.Text = "";
60. textBox3.Text = ""; textBox4.Text = "";
61. textBox5.Text = ""; textBox6.Text = "";
62. textBox1.Focus();
63. }
64. }
65. }

update\_camera.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** update\_camera : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** update\_camera()
17. {
18. InitializeComponent();
19. }
21. **private** **void** button1\_Click(**object** sender, EventArgs e)
22. {
23. **string** condstr = "";
24. **if** (textBox1.Text != "")
25. condstr += "C\_FID ='" + textBox1.Text.Trim() + "'";
27. **if** (textBox2.Text != "")
28. {
29. **if** (condstr != "")
30. condstr += " and ";
32. condstr += "device\_name = " + textBox2.Text.Trim() + "";
33. }
35. **if** (textBox3.Text != "")
36. {
37. **if** (condstr != "")
38. condstr += " and ";
40. condstr += "longitude = " + **double**.Parse(textBox3.Text.Trim()) + "";
41. }
43. **if** (textBox4.Text != "")
44. {
45. **if** (condstr != "")
46. condstr += " and ";
48. condstr += "latitude = " + **double**.Parse(textBox4.Text.Trim()) + "";
49. }
51. **if** (textBox5.Text != "")
52. {
53. **if** (condstr != "")
54. condstr += " and ";
56. condstr += "land\_cover = '" + textBox5.Text.Trim() + "'";
57. }
59. **if** (textBox6.Text != "")
60. {
61. **if** (condstr != "")
62. condstr += " and ";
64. condstr += "E\_FID = '" + textBox6.Text.Trim() + "'";
65. }
67. **if** (condstr != "")
68. TempData.sql = "SELECT \* FROM camera WHERE " + condstr;
69. **else**
70. TempData.sql = "";
71. get\_camera myform = **new** get\_camera();
72. myform.ShowDialog();
73. }
75. **private** **void** button2\_Click(**object** sender, EventArgs e)
76. {
77. textBox1.Text = ""; textBox2.Text = "";
78. textBox3.Text = ""; textBox4.Text = "";
79. textBox5.Text = "";
80. textBox1.Focus();
81. }
82. }
83. }

get\_camera.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** get\_camera : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** get\_camera()
17. {
18. InitializeComponent();
19. }
21. **private** **void** button1\_Click(**object** sender, EventArgs e)
22. {
23. **if** (dataGridView1.SelectedRows.Count > 0)
24. {
25. **string** a = dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim();
26. **string** b = dataGridView1.SelectedRows[0].Cells[1].Value.ToString().Trim();
27. **string** c = dataGridView1.SelectedRows[0].Cells[2].Value.ToString().Trim();
28. **string** d = dataGridView1.SelectedRows[0].Cells[3].Value.ToString().Trim();
29. **string** R = dataGridView1.SelectedRows[0].Cells[4].Value.ToString().Trim();
30. **string** F = dataGridView1.SelectedRows[0].Cells[4].Value.ToString().Trim();
32. Form myform = **new** edit\_camera(a, b, c, d, R, F); //带传递数据调用edit\_node的构造函数
33. myform.ShowDialog();
34. }
35. }
37. **private** **void** get\_camera\_Load(**object** sender, EventArgs e)
38. {
39. bind();
40. dataGridView1.EnableHeadersVisualStyles = **false**;
41. dataGridView1.MultiSelect = **false**;
42. dataGridView1.ReadOnly = **true**;
43. dataGridView1.Columns[0].Width = 90;
44. dataGridView1.Columns[1].Width = 150;
45. dataGridView1.Columns[2].Width = 120;
46. dataGridView1.Columns[3].Width = 120;
47. dataGridView1.Columns[4].Width = 90;
48. dataGridView1.Columns[5].Width = 90;
49. **if** (dataGridView1.Rows.Count > 0)
50. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
51. }
53. **private** **void** bind() //绑定数据
54. {
55. **string** mystr;
56. **if** (TempData.sql == "")
57. mystr = "SELECT \* FROM camera";
58. **else**
59. mystr = TempData.sql;
60. DataSet mydataset = mydb.ExecuteQuery(mystr, "camera");
61. dataGridView1.DataSource = mydataset.Tables["camera"];
62. }
64. **private** **void** button2\_Click(**object** sender, EventArgs e)
65. {
66. //删除记录
67. **if** (dataGridView1.SelectedRows.Count > 0)
68. {
69. **string** spno = dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim();
70. DialogResult result = MessageBox.Show("确认删除" + spno
71. + "编号的CMERA吗？", "操作提示", MessageBoxButtons.YesNo);
72. **if** (result == DialogResult.Yes)
73. {
74. **string** mystr = "DELETE from camera WHERE C\_FID='" + spno + "'";
75. mydb.ExecuteNonQuery(mystr);
76. bind();
77. }
78. }
79. }
81. **private** **void** get\_camera\_Activated(**object** sender, EventArgs e)
82. {
83. bind();
84. **if** (dataGridView1.Rows.Count > 0)
85. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
86. }
88. **private** **void** dataGridView1\_CellContentClick(**object** sender, DataGridViewCellEventArgs e)
89. {
90. //单击任何单元时选择该单元所在的行
91. **if** (e.RowIndex >= 0 && e.RowIndex < dataGridView1.Rows.Count)
92. dataGridView1.Rows[e.RowIndex].Selected = **true**;
93. }
94. }
95. }

edit.camera.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** edit\_camera : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **string** FID1, LON;
17. **string** LAT, DE;
18. **string** FID2,LA;
20. **private** **void** button2\_Click(**object** sender, EventArgs e)
21. {
22. **this**.Close();
23. }
25. **private** **void** edit\_camera\_Load(**object** sender, EventArgs e)
26. {
27. textBox1.Text = FID1;
28. textBox1.Enabled = **false**;
29. textBox2.Text = DE;
30. textBox3.Text = LON;
31. textBox4.Text = LAT;
32. textBox5.Text = LA;
33. textBox6.Text = FID2;
34. }
36. **public** edit\_camera(**string** fid1, **string** name, **string** longitude, **string** latitude, **string** cover, **string** E\_FID)
37. {
38. InitializeComponent();
39. FID1 = fid1;
40. DE = name;
41. LON = longitude;
42. LAT = latitude;
43. LA = cover;
44. FID2 = E\_FID;
45. }
47. **private** **void** button1\_Click(**object** sender, EventArgs e)
48. {
49. **string** mystr;
50. **if** (textBox1.Text == "" || textBox2.Text == ""
51. || textBox3.Text == "" || textBox4.Text == ""
52. || textBox5.Text == "" || textBox5.Text == "")
53. {
54. MessageBox.Show("所有数据项必须输入");
55. **return**;
56. }
57. **try**
58. {
59. mystr = "UPDATE camera SET device\_name="
60. + textBox2.Text.Trim() + ",longitude="
61. + **double**.Parse(textBox3.Text.Trim()) + ",latitude="
62. + **double**.Parse(textBox4.Text.Trim()) + ",land\_cover="
63. + textBox5.Text.Trim() + ",E\_FID="
64. + textBox6.Text.Trim() + " WHERE C\_FID='"
65. + FID1 + "'";
66. mydb.ExecuteNonQuery(mystr);
67. }
68. **catch** (Exception ex)
69. {
70. MessageBox.Show("修改的节点数据有错误：" + ex.Message);
71. **return**;
72. }
73. **this**.Close();
74. }
75. }
76. }

5.7、栅格数据管理功能设计

栅格管理功能与路段管理信息功能原理相似，下面只做操作展示与设计源代码。



图5.7.1：初始化窗口

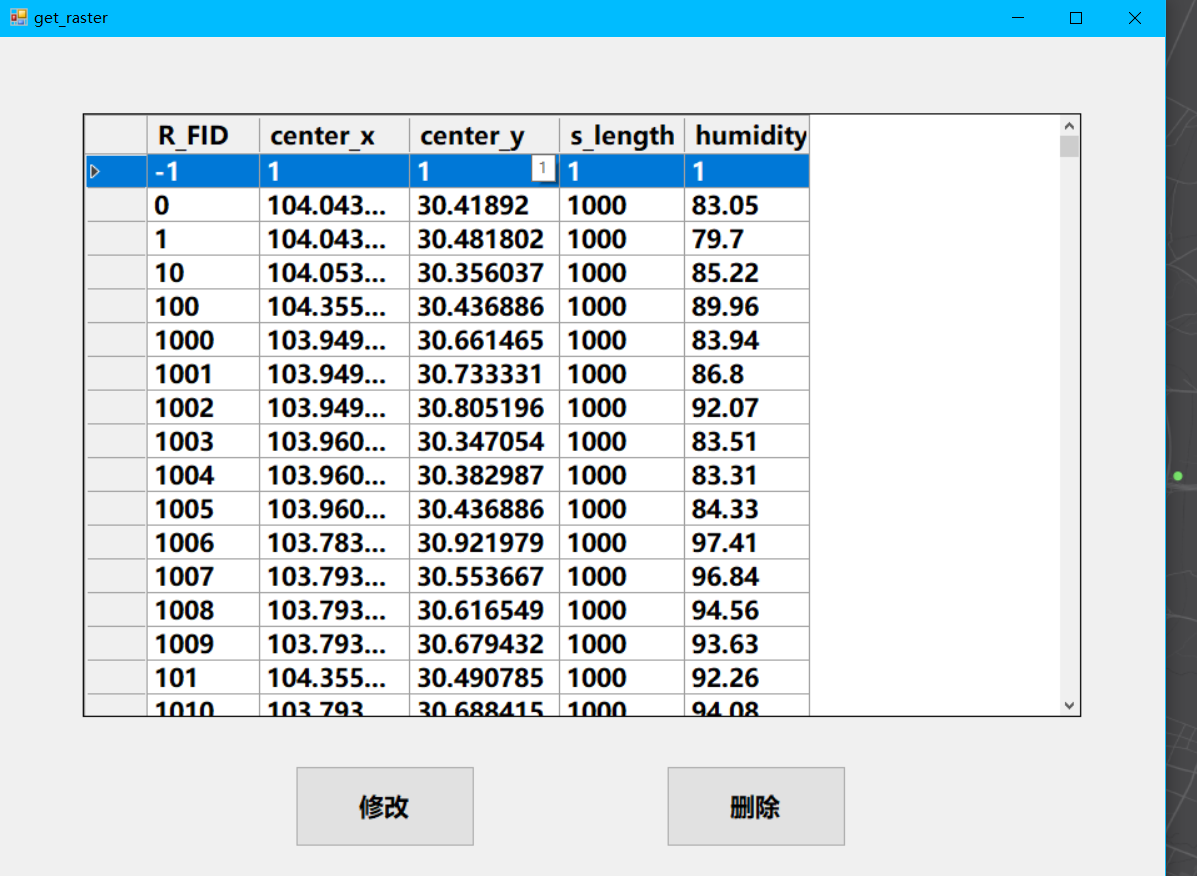


图5.7.2：得到栅格信息后窗口



图5.7.3：栅格信息编辑

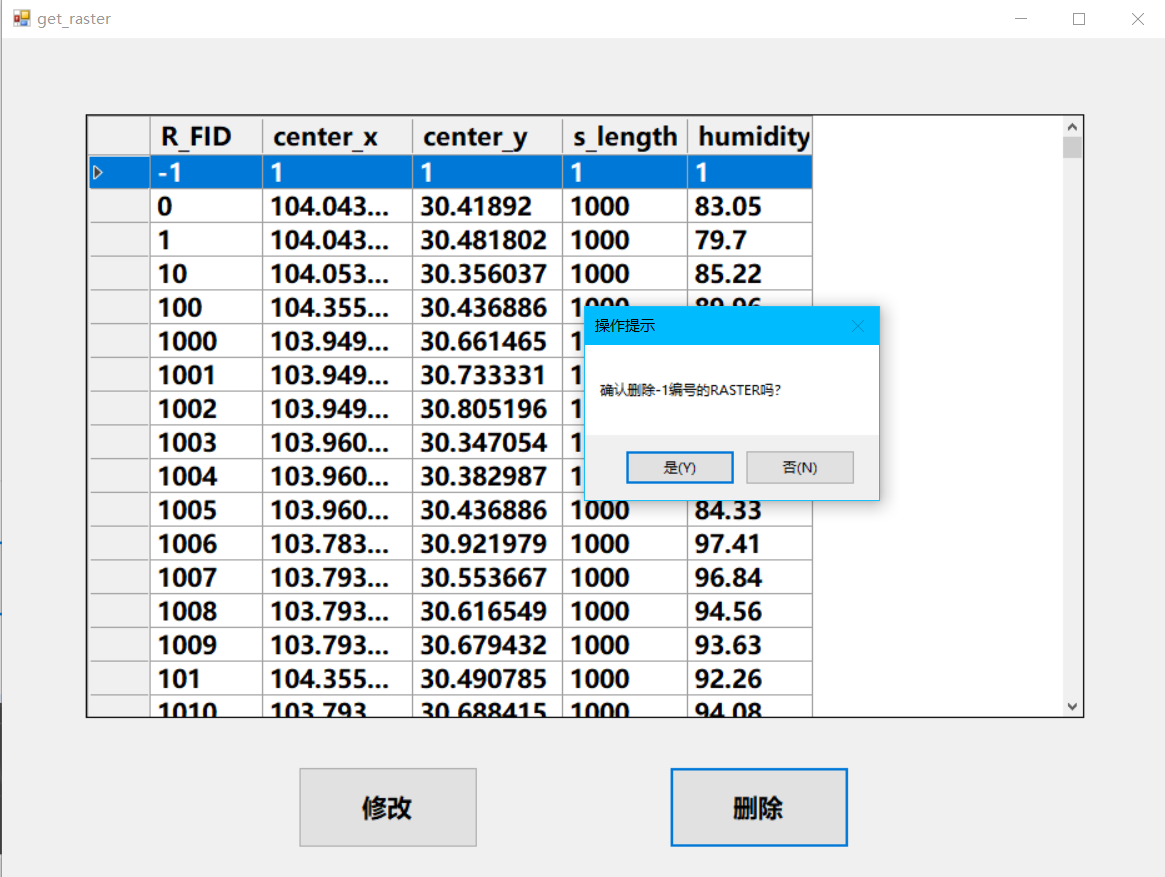


图5.7.4：栅格信息删除

add\_raster.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
14. **public** partial **class** add\_raster : Form
15. {
16. CommDB mydb = **new** CommDB();
17. **public** add\_raster()
18. {
19. InitializeComponent();
20. }
22. **private** **void** button1\_Click(**object** sender, EventArgs e)
23. {
24. **if** (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == ""
25. || textBox4.Text == "" || textBox5.Text == "")
26. {
27. MessageBox.Show("所有数据项必须输入", "操作提示");
28. **return**;
29. }
30. **string** mystr = "SELECT \* FROM raster WHERE R\_FID='" + textBox1.Text.Trim() + "'";
31. **int** i = mydb.Rownum(mystr);
32. **if** (i != 0)
33. {
34. MessageBox.Show("输入的RASTER编号已存在", "操作提示");
35. **return**;
36. }
38. **try**
39. {
40. mystr = "INSERT INTO RASTER(R\_FID,center\_x,center\_y,s\_length,humidity) VALUES('"
41. + textBox1.Text.Trim() + "','"
42. + textBox2.Text.Trim() + "','"
43. + textBox3.Text.Trim() + "','"
44. + textBox4.Text.Trim() + "','"
45. + textBox5.Text.Trim() + "')";
46. mydb.ExecuteNonQuery(mystr);
47. MessageBox.Show("成功添加");
48. }
49. **catch** (Exception ex)
50. {
51. MessageBox.Show("输入的RASTER数据有错误：" + ex.Message);
52. **return**;
53. }
54. button2\_Click(sender, e);
55. }
57. **private** **void** button2\_Click(**object** sender, EventArgs e)
58. {
59. textBox1.Text = ""; textBox2.Text = "";
60. textBox3.Text = ""; textBox4.Text = "";
61. textBox5.Text = "";
62. textBox1.Focus();
63. }
65. }
66. }

update\_raster.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** update\_raster : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** update\_raster()
17. {
18. InitializeComponent();
19. }
21. **private** **void** button1\_Click(**object** sender, EventArgs e)
22. {
23. **string** condstr = "";
24. **if** (textBox1.Text != "")
25. condstr += "R\_FID ='" + textBox1.Text.Trim() + "'";
27. **if** (textBox2.Text != "")
28. {
29. **if** (condstr != "")
30. condstr += " and ";
32. condstr += "center\_x = " + **double**.Parse(textBox2.Text.Trim()) + "";
33. }
35. **if** (textBox3.Text != "")
36. {
37. **if** (condstr != "")
38. condstr += " and ";
40. condstr += "center\_y = " + **double**.Parse(textBox3.Text.Trim()) + "";
41. }
43. **if** (textBox4.Text != "")
44. {
45. **if** (condstr != "")
46. condstr += " and ";
48. condstr += "s\_length = " + **double**.Parse(textBox4.Text.Trim()) + "";
49. }
51. **if** (textBox5.Text != "")
52. {
53. **if** (condstr != "")
54. condstr += " and ";
56. condstr += "humidity = '" + **double**.Parse(textBox5.Text.Trim()) + "'";
57. }
59. **if** (condstr != "")
60. TempData.sql = "SELECT \* FROM raster WHERE " + condstr;
61. **else**
62. TempData.sql = "";
63. get\_raster myform = **new** get\_raster();
64. myform.ShowDialog();
65. }
66. }
67. }

get\_raster.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** get\_raster : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** get\_raster()
17. {
18. InitializeComponent();
19. }
21. **private** **void** button1\_Click(**object** sender, EventArgs e)
22. {
23. **if** (dataGridView1.SelectedRows.Count > 0)
24. {
25. **string** a = dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim();
26. **string** b = dataGridView1.SelectedRows[0].Cells[1].Value.ToString().Trim();
27. **string** c = dataGridView1.SelectedRows[0].Cells[2].Value.ToString().Trim();
28. **string** d = dataGridView1.SelectedRows[0].Cells[3].Value.ToString().Trim();
29. **string** R = dataGridView1.SelectedRows[0].Cells[4].Value.ToString().Trim();
31. Form myform = **new** edit\_raster(a, b, c, d, R); //带传递数据调用edit\_node的构造函数
32. myform.ShowDialog();
33. }
34. }
36. **private** **void** get\_raster\_Load(**object** sender, EventArgs e)
37. {
38. bind();
39. dataGridView1.EnableHeadersVisualStyles = **false**;
40. dataGridView1.MultiSelect = **false**;
41. dataGridView1.ReadOnly = **true**;
42. dataGridView1.Columns[0].Width = 90;
43. dataGridView1.Columns[1].Width = 120;
44. dataGridView1.Columns[2].Width = 120;
45. dataGridView1.Columns[3].Width = 100;
46. dataGridView1.Columns[4].Width = 100;
47. **if** (dataGridView1.Rows.Count > 0)
48. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
49. }
50. **private** **void** bind() //绑定数据
51. {
52. **string** mystr;
53. **if** (TempData.sql == "")
54. mystr = "SELECT \* FROM raster";
55. **else**
56. mystr = TempData.sql;
57. DataSet mydataset = mydb.ExecuteQuery(mystr, "raster");
58. dataGridView1.DataSource = mydataset.Tables["raster"];
59. }
61. **private** **void** button2\_Click(**object** sender, EventArgs e)
62. {
63. **string** spno = dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim();
64. DialogResult result = MessageBox.Show("确认删除" + spno
65. + "编号的RASTER吗？", "操作提示", MessageBoxButtons.YesNo);
66. **if** (result == DialogResult.Yes)
67. {
68. **string** mystr = "DELETE from raster WHERE R\_FID='" + spno + "'";
69. mydb.ExecuteNonQuery(mystr);
70. bind();
71. }
72. }
74. **private** **void** dataGridView1\_CellContentClick(**object** sender, DataGridViewCellEventArgs e)
75. {
76. //单击任何单元时选择该单元所在的行
77. **if** (e.RowIndex >= 0 && e.RowIndex < dataGridView1.Rows.Count)
78. dataGridView1.Rows[e.RowIndex].Selected = **true**;
79. }
81. **private** **void** get\_raster\_Activated(**object** sender, EventArgs e)
82. {
83. bind();
84. **if** (dataGridView1.Rows.Count > 0)
85. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
86. }
87. }
88. }

edit\_raster.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** edit\_raster : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **string** FID, X;
17. **string** Y, LEN;
18. **string** HUM;
20. **private** **void** button2\_Click(**object** sender, EventArgs e)
21. {
22. **this**.Close();
23. }
25. **private** **void** edit\_raster\_Load(**object** sender, EventArgs e)
26. {
27. textBox1.Text = FID;
28. textBox1.Enabled = **false**;
29. textBox2.Text = X;
30. textBox3.Text = Y;
31. textBox4.Text = LEN;
32. textBox5.Text = HUM;
33. }
35. **public** edit\_raster(**string** fid1, **string** x, **string** y, **string** s\_length, **string** humidity)
36. {
37. InitializeComponent();
38. FID = fid1;
39. X = x;
40. Y = y;
41. LEN = s\_length;
42. HUM = humidity;
43. }
45. **private** **void** button1\_Click(**object** sender, EventArgs e)
46. {
47. **string** mystr;
48. **if** (textBox1.Text == "" || textBox2.Text == ""
49. || textBox3.Text == "" || textBox4.Text == ""
50. || textBox5.Text == "")
51. {
52. MessageBox.Show("所有数据项必须输入");
53. **return**;
54. }
55. **try**
56. {
57. mystr = "UPDATE raster SET center\_x="
58. + **double**.Parse(textBox2.Text.Trim()) + ",center\_y="
59. + **double**.Parse(textBox3.Text.Trim()) + ",s\_length="
60. + **double**.Parse(textBox4.Text.Trim()) + ",humidity="
61. + **double**.Parse(textBox5.Text.Trim()) + " WHERE R\_FID='"
62. + FID + "'";
63. mydb.ExecuteNonQuery(mystr);
64. }
65. **catch** (Exception ex)
66. {
67. MessageBox.Show("修改的栅格数据有错误：" + ex.Message);
68. **return**;
69. }
70. **this**.Close();
71. }
72. }
73. }

5.8、数据查询与导出功能设计

数据查询与导出为系统核心功能，利用结构化窗体方便数据的筛选与导出，其中查询功能中主要区分两个模块：需取用数据与筛选条件。筛选条件仅选用一些常用筛选需要取用的数据分为两个层级，选择与不选择两个层级时对窗体各组件的使用有影响。

执行查询操作后可以使用导出功能，导出功能加入组件Excel，将导出为.xlsx类型文件。

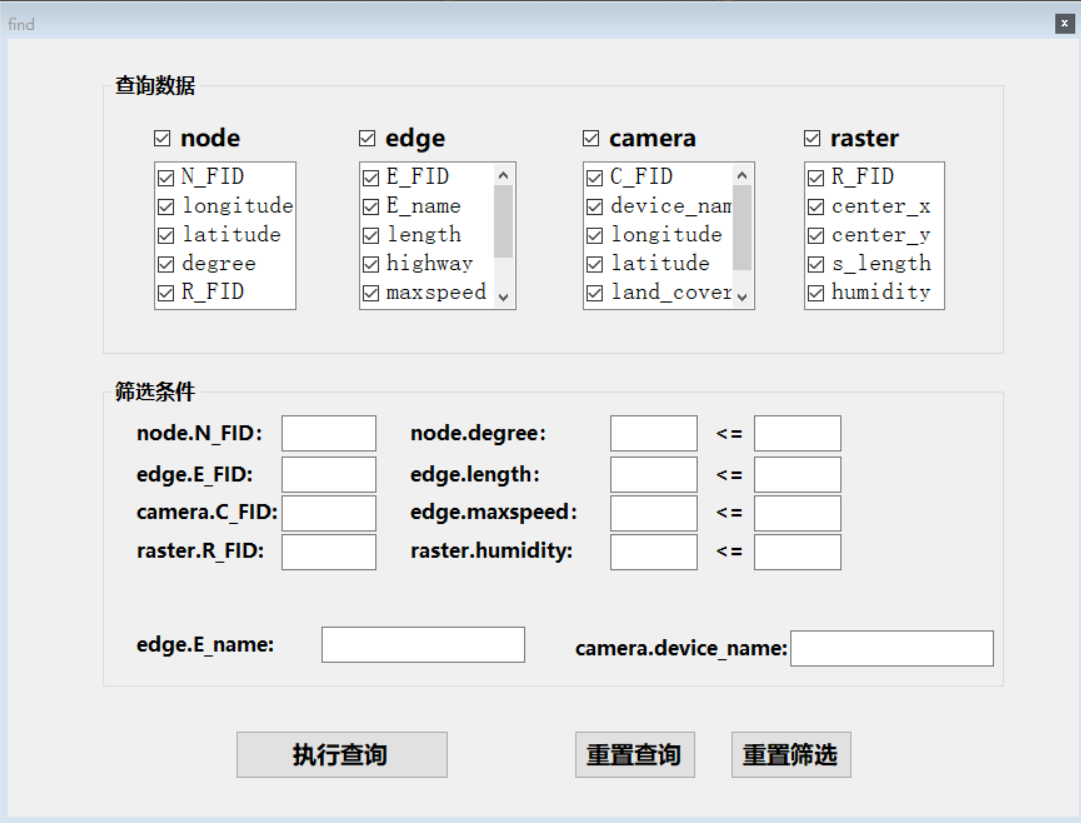


图5.8.1：查询功能初始化界面

下面进行展示对一定湿度条件节点进行筛选的过程。对于湿度值在80到83之间的数据，输入筛选条件，如图5.8.2。



图5.8.2：筛选条件

经筛选后得到数据，如图5.8.3

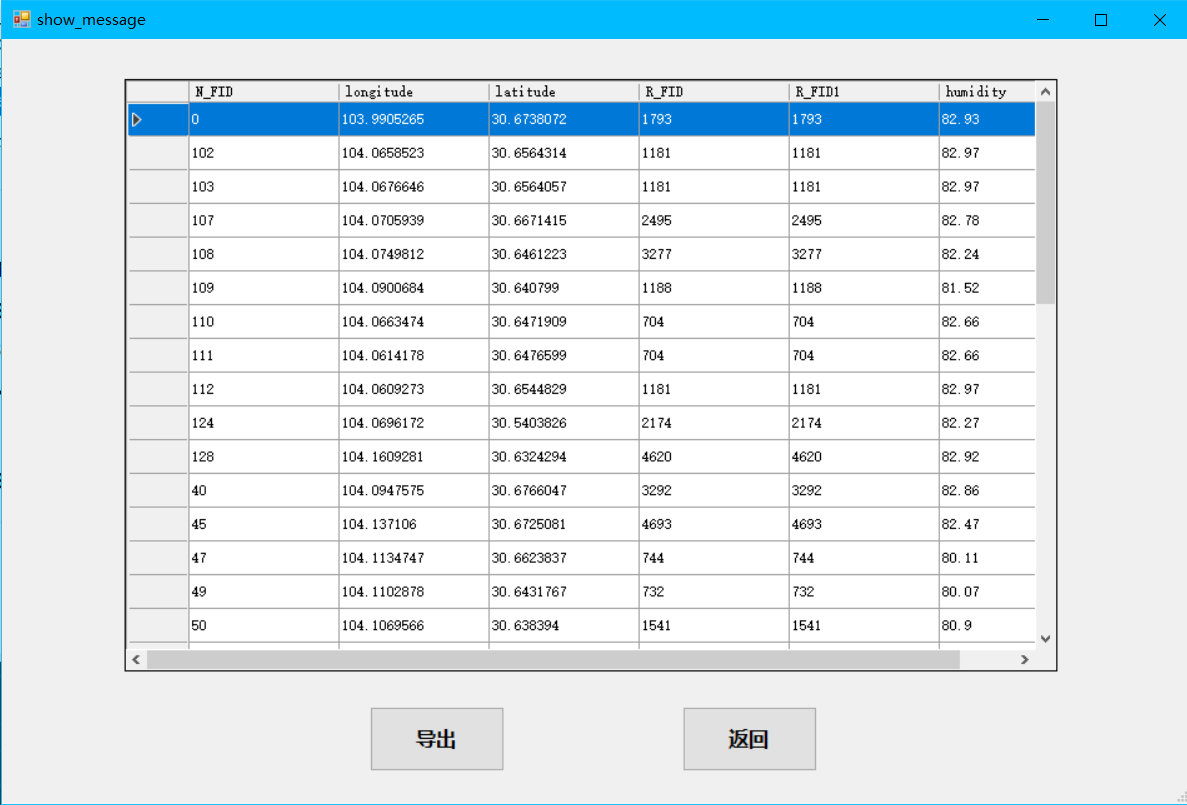


图5.8.3：数据展示

单击导出按钮，弹出导出窗口，如图5.8.4，设置导出路径为桌面，命名为file.xlsx。对导出后数据进行展示，如图5.8.5。

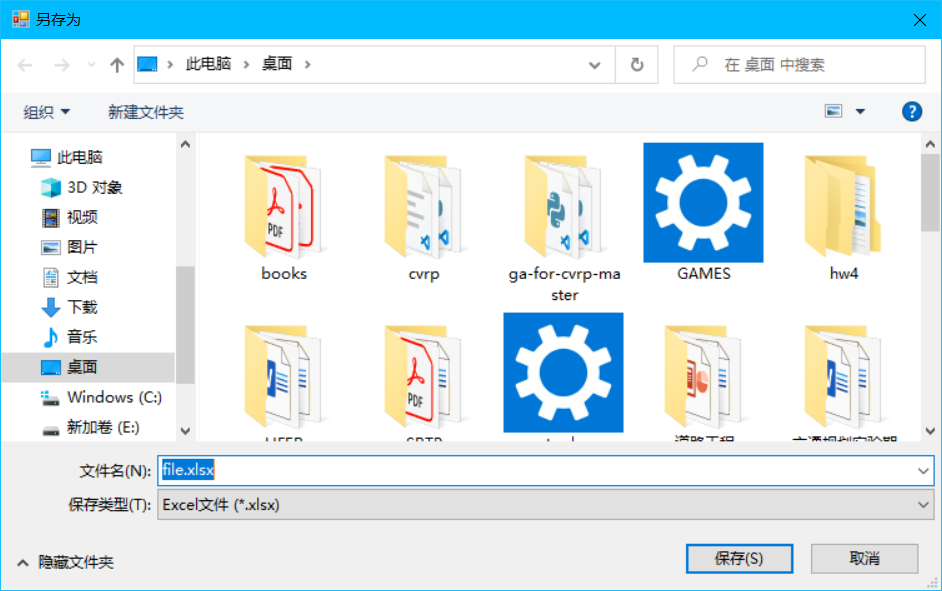


图5.8.4：数据导出



图5.8.5导出数据展示

find.cs：

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** find : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** find()
17. {
18. InitializeComponent();
19. }
21. **private** **void** checkBox1\_CheckedChanged(**object** sender, EventArgs e)
22. {
23. **if** (checkBox1.Checked == **true**)
24. {
25. checkedListBox1.Enabled = **true**;
26. **for** (**int** i = 0; i < checkedListBox1.Items.Count; i++)
27. {
28. checkedListBox1.SetItemChecked(i, **true**);
29. }
30. textBox1.Enabled = **true**;
31. textBox7.Enabled = **true**;
32. textBox8.Enabled = **true**;
33. }
35. **else**
36. {
37. checkedListBox1.Enabled = **false**;
38. **for** (**int** i = 0; i < checkedListBox1.Items.Count; i++)
39. {
40. checkedListBox1.SetItemChecked(i, **false**);
41. }
42. textBox1.Enabled = **false**;
43. textBox7.Enabled = **false**;
44. textBox8.Enabled = **false**;
45. }
46. }
48. **private** **void** find\_Load(**object** sender, EventArgs e)
49. {
50. checkBox1.Checked = **true**;
51. checkBox2.Checked = **true**;
52. checkBox3.Checked = **true**;
53. checkBox4.Checked = **true**;
54. }
56. **private** **void** checkBox2\_CheckedChanged(**object** sender, EventArgs e)
57. {
58. **if** (checkBox2.Checked == **true**)
59. {
60. checkedListBox2.Enabled = **true**;
61. **for** (**int** i = 0; i < checkedListBox2.Items.Count; i++)
62. {
63. checkedListBox2.SetItemChecked(i, **true**);
64. }
65. textBox2.Enabled = **true**;
66. textBox9.Enabled = **true**;
67. textBox10.Enabled = **true**;
68. textBox11.Enabled = **true**;
69. textBox12.Enabled = **true**;
70. textBox5.Enabled = **true**;
71. }
72. **else**
73. {
74. checkedListBox2.Enabled = **false**;
75. **for** (**int** i = 0; i < checkedListBox2.Items.Count; i++)
76. {
77. checkedListBox2.SetItemChecked(i, **false**);
78. }
79. textBox2.Enabled = **false**;
80. textBox9.Enabled = **false**;
81. textBox10.Enabled = **false**;
82. textBox11.Enabled = **false**;
83. textBox12.Enabled = **false**;
84. textBox5.Enabled = **false**;
85. }
86. }
88. **private** **void** checkBox3\_CheckedChanged(**object** sender, EventArgs e)
89. {
90. **if** (checkBox3.Checked == **true**)
91. {
92. checkedListBox3.Enabled = **true**;
93. **for** (**int** i = 0; i < checkedListBox3.Items.Count; i++)
94. {
95. checkedListBox3.SetItemChecked(i, **true**);
96. }
97. textBox3.Enabled = **true**;
98. textBox6.Enabled = **true**;
99. }
100. **else**
101. {
102. checkedListBox3.Enabled = **false**;
103. **for** (**int** i = 0; i < checkedListBox3.Items.Count; i++)
104. {
105. checkedListBox3.SetItemChecked(i, **false**);
106. }
107. textBox3.Enabled = **false**;
108. textBox6.Enabled = **false**;
109. }
110. }
112. **private** **void** checkBox4\_CheckedChanged(**object** sender, EventArgs e)
113. {
114. **if** (checkBox4.Checked == **true**)
115. {
116. checkedListBox4.Enabled = **true**;
117. **for** (**int** i = 0; i < checkedListBox4.Items.Count; i++)
118. {
119. checkedListBox4.SetItemChecked(i, **true**);
120. }
121. textBox4.Enabled = **true**;
122. textBox13.Enabled = **true**;
123. textBox14.Enabled = **true**;
124. }
125. **else**
126. {
127. checkedListBox4.Enabled = **false**;
128. **for** (**int** i = 0; i < checkedListBox4.Items.Count; i++)
129. {
130. checkedListBox4.SetItemChecked(i, **false**);
131. }
132. textBox4.Enabled = **false**;
133. textBox13.Enabled = **false**;
134. textBox14.Enabled = **false**;
135. }
136. }
138. **private** **void** button2\_Click(**object** sender, EventArgs e)
139. {
140. checkBox1.Checked = **false**;
141. checkBox2.Checked = **false**;
142. checkBox3.Checked = **false**;
143. checkBox4.Checked = **false**;
144. **for** (**int** i = 0; i < checkedListBox1.Items.Count; i++)
145. {
146. checkedListBox1.SetItemChecked(i, **false**);
147. }
148. **for** (**int** i = 0; i < checkedListBox2.Items.Count; i++)
149. {
150. checkedListBox2.SetItemChecked(i, **false**);
151. }
152. **for** (**int** i = 0; i < checkedListBox3.Items.Count; i++)
153. {
154. checkedListBox3.SetItemChecked(i, **false**);
155. }
156. **for** (**int** i = 0; i < checkedListBox4.Items.Count; i++)
157. {
158. checkedListBox4.SetItemChecked(i, **false**);
159. }
160. }
162. **private** **void** button3\_Click(**object** sender, EventArgs e)
163. {
164. textBox1.Text = ""; textBox2.Text = "";
165. textBox3.Text = ""; textBox4.Text = "";
166. textBox5.Text = ""; textBox6.Text = "";
167. textBox7.Text = ""; textBox8.Text = "";
168. textBox9.Text = ""; textBox10.Text = "";
169. textBox10.Text = ""; textBox12.Text = "";
170. textBox13.Text = "";
171. }
173. **private** **void** button1\_Click(**object** sender, EventArgs e)
174. {
175. **string** condstr;
176. **string** need\_str = "";
177. **string** limits = "";
178. **string** table = "";
179. **if** (!checkBox1.Checked && !checkBox2.Checked && !checkBox3.Checked && !checkBox4.Checked)
180. {
181. MessageBox.Show("请至少选择一个数据类");
182. **return**;
183. }
184. **else**
185. {
186. **if**(checkBox1.Checked)
187. {
188. **if** (table != "")
189. table = table + ",";
190. table = table + checkBox1.Text;
191. **for** (**int** i = 0; i < checkedListBox1.Items.Count; i++)
192. {
193. **if** (checkedListBox1.GetItemChecked(i))
194. {
195. **if** (need\_str == **string**.Empty)
196. need\_str = "NODE."+checkedListBox1.GetItemText(checkedListBox1.Items[i]);
197. **else**
198. need\_str = need\_str + "," + "NODE." + checkedListBox1.GetItemText(checkedListBox1.Items[i]);
199. }
200. }
201. }
203. **if** (checkBox2.Checked)
204. {
205. **if** (table != "")
206. table = table + ",";
207. table = table + checkBox2.Text;
208. **for** (**int** i = 0; i < checkedListBox2.Items.Count; i++)
209. {
210. **if** (checkedListBox2.GetItemChecked(i))
211. {
212. **if** (need\_str == **string**.Empty)
213. need\_str = "EDGE." + checkedListBox2.GetItemText(checkedListBox2.Items[i]);
214. **else**
215. need\_str = need\_str + "," + "EDGE." + checkedListBox2.GetItemText(checkedListBox2.Items[i]);
216. }
217. }
218. }
219. **if** (checkBox3.Checked)
220. {
221. **if** (table != "")
222. table = table + ",";
223. table = table + checkBox3.Text;
224. **for** (**int** i = 0; i < checkedListBox3.Items.Count; i++)
225. {
226. **if** (checkedListBox3.GetItemChecked(i))
227. {
228. **if** (need\_str == **string**.Empty)
229. need\_str = "CAMERA." + checkedListBox3.GetItemText(checkedListBox3.Items[i]);
230. **else**
231. need\_str = need\_str + "," + "CAMERA." + checkedListBox3.GetItemText(checkedListBox3.Items[i]);
232. }
233. }
234. }
235. **if** (checkBox4.Checked)
236. {
237. **if** (table != "")
238. table = table + ",";
239. table = table + checkBox4.Text;
240. **for** (**int** i = 0; i < checkedListBox4.Items.Count; i++)
241. {
242. **if** (checkedListBox4.GetItemChecked(i))
243. {
244. **if** (need\_str == **string**.Empty)
245. need\_str = "RASTER." + checkedListBox4.GetItemText(checkedListBox4.Items[i]);
246. **else**
247. need\_str = need\_str + "," + "RASTER." + checkedListBox4.GetItemText(checkedListBox4.Items[i]);
248. }
249. }
250. }
251. }
252. **if**(need\_str == "")
253. {
254. MessageBox.Show("请至少选择一个数据项");
255. **return**;
256. }
258. **if**(textBox1.Text!="" && textBox1.Enabled)
259. {
260. **if** (limits != "")
261. limits = limits + " and ";
262. limits = limits + "node.N\_FID = " + textBox1.Text;
263. }
264. **if** (textBox2.Text != "" && textBox2.Enabled)
265. {
266. **if** (limits != "")
267. limits = limits + " and ";
268. limits = limits + "edge.E\_FID = " + textBox2.Text;
269. }
270. **if** (textBox3.Text != "" && textBox3.Enabled)
271. {
272. **if** (limits != "")
273. limits = limits + " and ";
274. limits = limits + "camera.C\_FID = " + textBox3.Text;
275. }
276. **if** (textBox4.Text != "" && textBox4.Enabled)
277. {
278. **if** (limits != "")
279. limits = limits + " and ";
280. limits = limits + "raster.R\_FID = " + textBox4.Text;
281. }
283. **if** (textBox5.Text != "" && textBox5.Enabled)
284. {
285. **if** (limits != "")
286. limits = limits + " and ";
287. limits = limits + "edge.E\_name LIKE '%" + textBox5.Text +"%' ";
288. }
289. **if** (textBox6.Text != "" && textBox6.Enabled)
290. {
291. **if** (limits != "")
292. limits = limits + " and ";
293. limits = limits + "camera.device\_name LIKE %" + textBox6.Text + "%' ";
294. }
296. **if** ((textBox7.Text != ""|| textBox8.Text != "") && textBox7.Enabled && textBox8.Enabled)
297. {
298. **int** lmin, lmax;
299. **if** (limits != "")
300. limits = limits + " and ";
301. **if** (textBox7.Text != "")
302. {
303. **try**
304. {
305. lmin = **int**.Parse(textBox7.Text);
306. }
307. **catch**
308. {
309. MessageBox.Show("node.degree最小值有错误：");
310. **return**;
311. }
312. limits = limits + "node.degree >= " + lmin;
313. }
314. **if** (limits != "")
315. limits = limits + " and ";
316. **if** (textBox8.Text != "")
317. {
318. **try**
319. {
320. lmax = **int**.Parse(textBox8.Text);
321. }
322. **catch**
323. {
324. MessageBox.Show("node.degree最大值有错误：");
325. **return**;
326. }
327. limits = limits + "node.degree <= " + lmax;
328. }
329. }
331. **if** ((textBox9.Text != "" || textBox10.Text != "") && textBox9.Enabled && textBox10.Enabled)
332. {
333. **double** lmin, lmax;
334. **if** (limits != "")
335. limits = limits + " and ";
336. **if** (textBox9.Text != "")
337. {
338. **try**
339. {
340. lmin = **double**.Parse(textBox9.Text);
341. }
342. **catch**
343. {
344. MessageBox.Show("edge.length最小值有错误：");
345. **return**;
346. }
347. limits = limits + "edge.length >= " + lmin;
348. }
349. **if** (limits != "")
350. limits = limits + " and ";
351. **if** (textBox10.Text != "")
352. {
353. **try**
354. {
355. lmax = **int**.Parse(textBox10.Text);
356. }
357. **catch**
358. {
359. MessageBox.Show("edge.length最大值有错误：");
360. **return**;
361. }
362. limits = limits + "edge.length <= " + lmax;
363. }
364. }
366. **if** ((textBox11.Text != "" || textBox12.Text != "") && textBox11.Enabled && textBox12.Enabled)
367. {
368. **int** lmin, lmax;
369. **if** (limits != "")
370. limits = limits + " and ";
371. **if** (textBox11.Text != "")
372. {
373. **try**
374. {
375. lmin = **int**.Parse(textBox11.Text);
376. }
377. **catch**
378. {
379. MessageBox.Show("edge.maxspeed最小值有错误：");
380. **return**;
381. }
382. limits = limits + "edge.maxspeed >= " + lmin;
383. }
384. **if** (limits != "")
385. limits = limits + " and ";
386. **if** (textBox12.Text != "")
387. {
388. **try**
389. {
390. lmax = **int**.Parse(textBox12.Text);
391. }
392. **catch**
393. {
394. MessageBox.Show("edge.maxspeed最大值有错误：");
395. **return**;
396. }
397. limits = limits + "edge.maxspeed <= " + lmax;
398. }
399. }
401. **if** ((textBox13.Text != "" || textBox14.Text != "") && textBox13.Enabled && textBox14.Enabled)
402. {
403. **double** lmin, lmax;
404. **if** (limits != "")
405. limits = limits + " and ";
406. **if** (textBox13.Text != "")
407. {
408. **try**
409. {
410. lmin = **double**.Parse(textBox13.Text);
411. }
412. **catch**
413. {
414. MessageBox.Show("raster.humidity最小值有错误：");
415. **return**;
416. }
417. limits = limits + "raster.humidity >= " + lmin;
418. }
419. **if** (limits != "")
420. limits = limits + " and ";
421. **if** (textBox14.Text != "")
422. {
423. **try**
424. {
425. lmax = **int**.Parse(textBox14.Text);
426. }
427. **catch**
428. {
429. MessageBox.Show("raster.humidity最大值有错误：");
430. **return**;
431. }
432. limits = limits + "raster.humidity <= " + lmax;
433. }
434. }
436. **if** (checkBox1.Checked && checkBox4.Checked)
437. {
438. **if** (limits != "")
439. limits = limits + " and ";
440. limits = limits + "node.R\_FID = raster.R\_FID";
441. }
443. **if** (checkBox2.Checked && checkBox4.Checked)
444. {
445. **if** (limits != "")
446. limits = limits + " and ";
447. limits = limits + "edge.R\_FID = raster.R\_FID";
448. }
450. **if** (checkBox3.Checked && checkBox2.Checked)
451. {
452. **if** (limits != "")
453. limits = limits + " and ";
454. limits = limits + "edge.E\_FID = camera.E\_FID";
456. }
458. condstr = "SELECT " + need\_str + " FROM " + table;
459. **if**(limits != "")
460. condstr = condstr + " WHERE " + limits;
461. TempData.sql = condstr;
462. **try**
463. {
464. mydb.ExecuteNonQuery(TempData.sql);
465. }
466. **catch** (Exception ex)
467. {
468. MessageBox.Show(ex.Message);
469. **return**;
470. }
471. show\_message myform = **new** show\_message();
472. myform.ShowDialog();
473. }
475. }
476. }

show\_message.cs:

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
10. **using** Excel = Microsoft.Office.Interop.Excel;
12. **namespace** Network\_Chengdu
13. {
14. **public** partial **class** show\_message : Form
15. {
16. CommDB mydb = **new** CommDB();
17. **public** show\_message()
18. {
19. InitializeComponent();
20. }
22. **private** **void** bind() //绑定数据
23. {
24. **string** mystr;
25. **if** (TempData.sql == "")
26. **return**;
27. **else**
28. mystr = TempData.sql;
29. DataSet mydataset = mydb.ExecuteQuery(mystr, "mixed");
30. dataGridView1.DataSource = mydataset.Tables["mixed"];
31. }
33. **private** **void** button2\_Click(**object** sender, EventArgs e)
34. {
35. **this**.Close();
36. }
38. **private** **void** button1\_Click(**object** sender, EventArgs e)
39. {
40. **string** fileName = "file";//可以在这里设置默认文件名
41. **string** saveFileName = "";//文件保存名
42. SaveFileDialog saveDialog = **new** SaveFileDialog();//实例化文件对象
43. saveDialog.DefaultExt = "xlsx";//文件默认扩展名
44. saveDialog.Filter = "Excel文件|\*.xlsx";//获取或设置当前文件名筛选器字符串，该字符串决定对话框的“另存为文件类型”或“文件类型”框中出现的选择内容。
45. saveDialog.FileName = fileName;
46. saveDialog.ShowDialog();//打开保存窗口给你选择路径和设置文件名
47. saveFileName = saveDialog.FileName;
48. **if** (saveFileName.IndexOf(":") < 0)
49. **return**; //被点了取消
50. Microsoft.Office.Interop.Excel.Application xlApp = **new** Microsoft.Office.Interop.Excel.Application();
51. **if** (xlApp == **null**)
52. {
53. MessageBox.Show("无法创建Excel对象，您的电脑可能未安装Excel");
54. **return**;
55. }
56. Microsoft.Office.Interop.Excel.Workbooks workbooks = xlApp.Workbooks;//Workbooks代表一个 Microsoft Excel 工作簿
57. Microsoft.Office.Interop.Excel.Workbook workbook = workbooks.Add(Microsoft.Office.Interop.Excel.XlWBATemplate.xlWBATWorksheet);//新建一个工作表。 新工作表将成为活动工作表。
58. Microsoft.Office.Interop.Excel.Worksheet worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.Worksheets[1];//取得sheet1
59. //写入标题
60. **for** (**int** i = 0; i < dataGridView1.ColumnCount; i++)//遍历循环获取DataGridView标题
61. {
62. worksheet.Cells[1, i + 1] = dataGridView1.Columns[i].HeaderText;
63. }// worksheet.Cells[1, i + 1]表示工作簿第一行第i+1列，Columns[i].HeaderText表示第i列的表头
64. //写入数值
65. **for** (**int** r = 0; r < dataGridView1.Rows.Count; r++)//这里表示数据的行标,dataGridView1.Rows.Count表示行数
66. {
67. **for** (**int** i = 0; i < dataGridView1.ColumnCount; i++)//遍历r行的列数
68. {
69. worksheet.Cells[r + 2, i + 1] = dataGridView1.Rows[r].Cells[i].Value;//Cells[r + 2, i + 1]表示工作簿从第二行开始第一行保存为表头了，dataGridView1.Rows[r].Cells[i].Value获取列的r行i值
70. }
71. System.Windows.Forms.Application.DoEvents();//实时更新表格
72. }
73. worksheet.Columns.EntireColumn.AutoFit();//列宽自适应
74. MessageBox.Show(fileName + "资料保存成功", "提示", MessageBoxButtons.OK);//提示保存成功
75. **if** (saveFileName != "")//saveFileName保存文件名不为空
76. {
77. **try**
78. {
79. workbook.Saved = **true**;//获取或设置一个值，该值指示工作簿自上次保存以来是否进行了更改
80. workbook.SaveCopyAs(saveFileName);  //fileSaved = true;将工作簿副本保存到文件中，但不修改内存中打开的工作簿
81. }
82. **catch** (Exception ex)
83. {//fileSaved = false;
84. MessageBox.Show("导出文件时出错,文件可能正被打开！\n" + ex.Message);
85. }
86. }
87. xlApp.Quit();
88. GC.Collect();//强行销毁
89. }
91. **private** **void** show\_message\_Load(**object** sender, EventArgs e)
92. {
93. bind();
94. **int** num;
95. num = dataGridView1.ColumnCount;
96. dataGridView1.EnableHeadersVisualStyles = **false**;
97. dataGridView1.MultiSelect = **false**;
98. dataGridView1.ReadOnly = **true**;
99. **for**(**int** i = 0;i<num;i++)
100. {
101. dataGridView1.Columns[i].Width = 120;
102. }
103. **if** (dataGridView1.Rows.Count > 0)
104. dataGridView1.Rows[0].Selected = **true**; //默认选择第1行
105. }
107. **private** **void** dataGridView1\_CellContentClick(**object** sender, DataGridViewCellEventArgs e)
108. {
109. **if** (e.RowIndex >= 0 && e.RowIndex < dataGridView1.Rows.Count)
110. dataGridView1.Rows[e.RowIndex].Selected = **true**;
111. }
112. }
113. }

5.9、统计信息功能设计

统计信息功能可实时动态查看当下数据库中各类实体的数量，如图5.9.1：



图5.9.1：统计信息窗口

statistic.cs：

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Linq;
7. **using** System.Text;
8. **using** System.Threading.Tasks;
9. **using** System.Windows.Forms;
11. **namespace** Network\_Chengdu
12. {
13. **public** partial **class** statistics : Form
14. {
15. CommDB mydb = **new** CommDB();
16. **public** statistics()
17. {
18. InitializeComponent();
19. }
21. **private** **void** statistics\_Load(**object** sender, EventArgs e)
22. {
23. **string** mystr;
24. mystr = "select Count(\*) from NODE";
25. DataSet mydataset = mydb.ExecuteQuery(mystr, "node\_sum");
26. label5.Text = mydataset.Tables[0].Rows[0][0].ToString();
28. mystr = "select Count(\*) from EDGE";
29. DataSet mydataset2 = mydb.ExecuteQuery(mystr, "edge\_sum");
30. label6.Text = mydataset2.Tables[0].Rows[0][0].ToString();
32. mystr = "select Count(\*) from CAMERA";
33. DataSet mydataset3 = mydb.ExecuteQuery(mystr, "camera\_sum");
34. label7.Text = mydataset3.Tables[0].Rows[0][0].ToString();
36. mystr = "select Count(\*) from RASTER";
37. DataSet mydataset4 = mydb.ExecuteQuery(mystr, "raster\_sum");
38. label8.Text = mydataset4.Tables[0].Rows[0][0].ToString();
39. }
40. }
41. }

5.10、帮助信息功能设计

帮助信息中包含关于与作者信息，关于中包含版本信息与声明，作者信息中包含作者联系方式与一些diy。

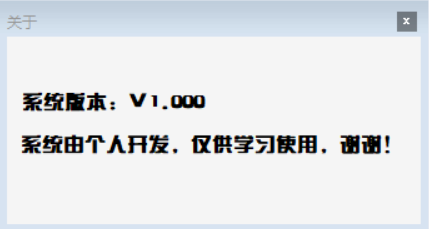


图5.10.1：关于窗口界面



图5.10.2：作者信息窗口界面

六、总结

1、优点

该系统基于真实脱敏数据，立足现实需求，能够基本完成既定数据查询与导出的目标。该系统添加了各项功能，使用到了C#中多组控件，进行对Sql表的操作，达到课程设计的标准与要求，锻炼了数据库编程开发的能力。

2、缺点

由于时间问题，UI设计仍有极大的进步空间，部分功能存在冗余重叠问题，代码量可通过设置通用函数进行缩减。

七、其它

后续更新计划（虽说遥遥无期，但总不能弃如敝履）：

V1.0后续版本进行代码简化、UI设计、功能整合，同时计划更新批量导入数据功能。

V2.0以及后续版本开发路网数据可视化功能以及内嵌gis\_t模块。

V3.0以及后续版本设计系统内嵌算法

V4.0以及后续版本内嵌编程模块

参考文献

[1].Mana Takahashi, Shoko Azuma. The Manga guide to databases. No StarchPress. 2009 （高橋麻奈，あづま笙子. 漫画数据库. 北京：科学出版社. 2010）

[2].王珊, 萨师煊. 数据库系统概论(第5 版). 北京: 高等教育出版社. 2014

[3].李春葆等. C7 语言与数据库基础教程. 北京: 清华大学出版社. 2016

[4].西南交通大学交通运输与物流学院.《数据库系统及应用》课程实验指导书. 2021

[5].安筱. [C# DataGridView数据导出Excel文件 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/438660896).2022.12.14

致谢

很感谢老师告诉我们，去写一个致谢吧。常常辗转忙碌于左左右右、边边角角，一直是走在路上，学习过很多不知来源的代码，受过很多人无私的帮助，但很少能够静下心来细数过。

我想我首先应拜谢的是我亲爱的家人们，我的父母、我的大哥，他们从始至终关心我，给予我无私的支持。感谢过去曾陪伴过我的同学、朋友，蓦然回首，那些经历已经融入灵魂。感谢老师无私的教诲，学校给予的资源帮助。

谈一谈正事，TICO让我生动看到数据库之妙用，影响了我的热情，老师提供的学习资料与实验指导书帮助我了解了知识。该系统config引用了实验指导书中的代码，窗体设计为原创。数据查询部分代码对其有所参照，核心部分查找为纯原创，导出功能参照知乎安筱用户的代码，对设置属性值进行改变。