

第 13 章 GSQL 查询分析器基础

13.1 背景

查询、检索是地理信息系统中使用最频繁的功能之一。GIS 用户提出的大部分问题都可以表达为查询形式。例如：“一个给定的地块与危险废弃厂相邻吗？”，“河流洪水泛滥区与提出的高速公路网交叠吗？”或“长江穿过的省区有那些？”。在这些查询中，除了属性条件查询外，更主要的是涉及到空间位置的查询，例如上面的查询所用到的“相邻”、“交叠”、“穿过”等条件。

对空间数据的管理和分析能力成为衡量地理信息系统的一个重要依据。传统的 GIS 大多采用空间和属性分别管理的方法，很难处理用户使用频繁的与地理位置相关的查询和分析功能，因此，迫切需要一个专门的语言来管理地理信息系统中的空间数据。

GIS 数据描述的是空间信息，一般包括位置、属性和时间三个方面。在 GIS 应用中，使用最广泛的是空间数据的查询，有时查询某一个图形，有时是查询空间图形之间的相互关系。目前许多 GIS 软件提供的是常用的关系数据库结构化查询语言(SQL)，而关系数据库结构化查询语言有其固有的缺陷，例如：不支持空间概念特别是空间关系、空间对象的查询结果不能用空间图形的方式有效地显示给用户以及不支持元数据查询、知识查询、定性查询和基于图形对象的查询等。

当前对空间查询语言的研究包括：

(1) 空间结构化查询语言(Geometric SQL)。它是在关系型 SQL 上发展起来的,不仅能完成空间数据的查询,而且能表达查询结果。其形式为：(空间数据库查询语言)SQL + (目标表示语言)GPL = Geometric SQL。

(2) 可视化查询语言。将查询语言的操作对象和过程及其空间关系等，用直观的图形或表格显示给用户，构成可视化查询语言。

(3) 自然查询语言。引入自然语言的概念，使查询语言的描述更接近自然语言，另外用模糊数学方法将模糊概念量化为确定的范围，实现具有能理解模糊概念的查询。

近年来，空间查询语言的研究已引起广泛的关注。由于标准查询语言 SQL 已经非常成熟，具有面向问题和接近自然语言的良好特征，在 SQL 的基础上进行扩展是管理和分析空间数据的一个有利条件，又基于 OGC 的研究成果，我们将要素类、简单要素类和对象类作为空间对象的基本类型，成功的实现基于空间数据的查询和分析语言 GSQL (Geometric Structured Query Language)。而 GSQL 查询分析器就是利用 GSQL 语言进行分析决策的一个极为方便的工具。

在 GSQL 查询分析器中，您可以方便的通过 GSQL 语言，快速的编辑 GSQL 语句脚本来实现查询和分析功能，从而避免了以往反复的切换窗口和记忆文件等操作，大大的提高了工作效率，另外，GSQL 查询分析器中还给您提供了方便的结果查看、修改名称、保存脚本等实用功能，您完全可以在这一个工具中实现您想要的操作。

13.2 基本概念

13.2.1 GSQL

空间数据查询语言（Geometric Structured Query Language），是对数据库标准查询语言 SQL 的扩展，以仿照 SQL 脚本的形式实现空间数据的查询和分析。

13.2.2 属性查询

任何一个 GIS 软件都要管理两类数据：空间数据和属性数据。属性数据是反映事物相关特性的数据，用来描述对象的特征，属性数据通过对象类或要素类的属性表进行表示，以属性字段和属性记录的形式体现，对这些属性数据的查询称为属性查询。

13.2.3 空间查询

空间数据就是在某个空间框架（例如地球表面）中对象的位置数据，是反映事物地理位置有关信息的数据与的属性数据一起共同说明对象的总体特征。空间数据是与实体的地理位置有关的查询称为空间查询。

13.2.4 空间分析

利用计算机对数字地图进行分析。空间分析是 GIS 的核心和灵魂，是 GIS 区别于一般的信息系统、CAD 或者电子地图系统的主要标志之一。空间分析包括以下内容：基于空间关系的查询、空间量算、缓冲区分析、网络分析、叠加分析、空间统计分类分析等。由于网络分析单独成为一个独立的模块，因此，本 GSQL 查询分析器中并不包含网络分析。

13.2.5 语法规则

编辑语句时必须遵守的规范。因为 GSQL 是一种通过解释方式来执行的语言，因此，它必须遵从特定的解释规范才能被正确执行。GSQL 语句的语法规则以 SQL 的语法规则为基础进行扩展，加入了对空间数据类型和空间分析函数的支持。

第 14 章 GSQL 语法规则

14.1 GSQL 语法域

| | |
|---|-----|
| character \in {a, b, ..., z, A, B, ..., Z, @, -, _, #, \$, &, %, *} | 字符 |
| numeric \in {0, 1, ..., 9} | 数字 |
| operator \in {+, -, *, /, %, =, >=, <=, <>, !=, and, or, not, like} | 运算符 |

14.2 GSQL 语句的类型

GSQL 查询分析器主要的语句类型有查询语句和分析语句。

查询语句能够实现从某个类（可以是要素类、简单要素类和对象类）中提取符合条件的记录，结果形成一个新类；

分析语句主要是实现 OGC 规定的常用的空间分析，例如包含、求缓冲区等。

另外，为了体现 GSQL 语句的可读性，编辑器支持注释语句。注释语句的语法规则比较简单，沿用 SQL 的注释形式，以双横线 “--” 开始即为注释，为了输入方便，还支持中文状态下的双横线 “——”。

14.3 GSQL 基本语法结构

查询语句的基本语法结构为：

```
SET @CLS-xxx = SELECT clsName | alaisName | ATT
                FROM  prefix.clsName [as alaisName]
                [WHERE search_condition];
                [ORDER BY fieldName [ASC | DESC]]
```



```
SET @VAL-xxx = SELECT clustering_function
                FROM  prefix.clsName [as alaisName]
                [WHERE search_condition];
```

分析语句的基本语法结构为：

```
SET @CLS-xxx = special_analysis_function(parameter1, parameter2);
```

GSQL 查询分析器最主要的特征是支持多个语句脚本的一次性查询，每个语句都要以分号 “;” 结束，否则语法错误。

14.4 查询语句语法规则

SET: SET 关键字是用来设置查询结果名称的, GSQL 查询分析器中所有的语句都必须设置一个查询结果。

@CLS-: 结果类的名称前缀, 即如果查询结果要保存为一个类, 则类名称必须以@cls-为前缀。

@VAL-: 数值类型结果名称前缀, 即如果查询结果是一个数值, 则数值名称必须以@val-为前缀。

SELECT: SELECT 关键字是用来选择输出结果的, 作为 GSQL 查询分析器的输出结果, 只允许是要素类、简单要素类、对象类和数值。

如果是结果类类型和源类类型相同, 则输出用源类名称或源类的别名, 例如:

```
SET @CLS-xxx = SELECT R FROM FCLS.RIVER AS R WHERE ...
```

如果结果类类型和源类类型不同, 即从要素类或简单要素类中只提取符合条件属性记录, 形成对象类, 则输出需要用 ATT 表示, 例如:

```
SET @CLS-xxx = SELECT ATT FROM FCLS.RIVER AS R WHERE ...
```

如果输出结果是一个数值, 则 SELECT 后必须为聚类函数, 允许使用的聚类函数有 COUNT, SUM, AVG, MIN 和 MAX。聚类函数后的参数用括号括起来, 例如:

```
SET @VAL-COUNT = SELECT COUNT(*) FROM FCLS.RIVER AS R WHERE ...
```

FROM: FROM 关键字是用来确定数据源的, 后面直接跟类的名称, 并且只能是一个类的名称, 即不允许从两个或两个以上的类中进行联合查询, 类名称分两种情况:

第一: 如果要查询的类是在数据库中已经存在的, 则类名称前必须加前缀, 分别是:

要素类: FCLS.

对象类: OCLS.

简单要素类: SFCLS.

类名称中如果包含点号“.”, 则需要将类名称前后加单引号, 为了避免错误, 建议不管类名称是否包含点号, 都将其前后加单引号;

第二: 如果要查询的类是中间结果类, 即对上面的语句生成的结果再查询, 则不需要加前缀。

例如:

```
SET @CLS-HH = SELECT RIVER FROM FCLS.'RIVER.WL' AS RIVER WHERE 水系 LIKE '%黄河%';
```

```
SET @CLS-HH-5km = SELECT HH FROM @CLS-HH AS HH WHERE 长度<=5000;
```

WHERE: WHERE 关键字后是表示查询条件的语句, 这是查询语句中最重要的部分, 分为属性条件查询和空间范围查询, 在下一节分别介绍。

ORDER BY: 对查询结果进行排序，后面直接跟属性字段名称，可以用 ASC 或 DESC 进行升序或降序限制，如果不加限制，默认为升序。

14.4.1 属性查询语句

- (1) 支持简单布尔检验：即单个搜索条件，通常用<, >, <=, >=, =, <>进行比较。并可以通过 NOT 进行逆运算；
- (2) 支持多个搜索条件的组合：通常用 AND, OR 组合查询表达式，并支持括号优先级；
- (3) 支持 NULL 检验：检验列是否为空，用 IS NULL 或 IS NOT NULL 表示；
- (4) 支持 LIKE 检验：常用于字符串类型和数值类型的字段，对于数值中的浮点型和双精度类型，只允许检查小数点前面的数字；
- (5) 常量的表示：数字常量和字段名称直接输入，字符串和日期时间常量必须用单引号括起来，日期的年月日之间用短横杠连接，时间的时分秒之间用冒号连接，邮戳类型由日期和时间用空格连接组成。
- (6) 支持下面的数学函数：

Abs:

语法: Abs (numeric_expression)
功能: 求一个数字表达式的绝对值
参数: 数字类型的表达式
返回: double 类型数字
说明: 总为非负值
举例: Abs(-29.344), Abs(长度-100)

Exp:

语法: Exp (numeric_expression)
功能: 求一个数字表达式的指数
参数: 数字类型的表达式
返回: double 类型数字
说明: 此指数函数是以 2 为底的指数

Floor:

语法: Floor (numeric_expression)
功能: 求小于或等于所给数字表达式的最大整数
参数: 数字类型的表达式
返回: 整型数字

Ceiling:

语法: Ceiling(numeric_expression)
功能: 求大于或等于所给数字表达式的最小整数
参数: 数字类型的表达式

返回：整型数字

Sin:

语法：Sin (numeric_expression)

功能：求一个数学表达式的正弦值

参数：数字类型表达式

返回：double 类型数字

Cos:

语法：Cos (numeric_expression)

功能：求一个数学表达式的余弦值

参数：数字类型表达式

返回：double 类型数字

Tan:

语法：Tan(numeric_expression)

功能：求一个数学表达式的正切值

参数：数字类型表达式

返回：double 类型数字

Cot:

语法：Cot (numeric_expression)

功能：求一个数学表达式的余切值

参数：数字类型表达式

返回：double 类型数字

Log:

语法：Log (numeric_expression)

功能：求一个数学表示式的自然对数

参数：数字类别表达式

返回：double 类型数字

说明：此对数函数是以 e 为底的自然对数

Sqrt:

语法：Sqrt (numeric_expression)

功能：求一个数学表达式的平方根

参数：数字类型表达式

返回：double 类型数字

(7) 支持下面的字符串函数：

Asc:

语法：Asc (str_expression)

功能：求一个字符串的第一个字母的 ASCII 码值

参数：字符串类型表达式

返回：byte 类型数字

Chr:

语法: Chr(numer_expression)
功能: 求给定 ASCII 码对应的字符
参数: 数字类型表达式
返回: 字符
说明: 数字必须是整数类型

Lcase:

语法: Lcase(str_expression)
功能: 将一个字符串转换为小写
参数: 字符串类型表达式
返回: 字符串

Ucase:

语法: Ucase(str_expression)
功能: 将一个字符串转换为大写
参数: 字符串类型表达式
返回: 字符串

Left:

语法: Left(str_expression, numeric num)
功能: 从左边开始截取一定长度的字符串
参数: 第一个参数为字符串类型表达式, 第二个参数为数字类型表达式
返回: 字符串
说明: 第二个参数必须为整数类型

Right:

语法: Right(str_expression, numeric num)
功能: 从右边开始截取一定长度的字符串
参数: 第一个参数为字符串类型表达式, 第二个参数为数字类型表达式
返回: 字符串
说明: 第二个参数必须为整数类型

Mid:

语法: Mid (str_expression, numeric begin, numeric num)
功能: 从 begin 位置开始截取长度为 num 的字符串
参数: 第一个参数为字符串类型, 第二个和第三个为数字表达式类型
返回: 字符串
说明: 第二个和第三个参数必须为整数类型

LTrim:

语法: LTrim(str_expression)
功能: 去掉字符串的前空格
参数: 字符串类型表达式

返回: 字符串

RTrim:

语法: RTrim(str_expression)

功能: 去掉字符串的后空格

参数: 字符串类型表达式

返回: 字符串

Trim:

语法: Trim(str_expression)

功能: 去掉字符串的前后空格

参数: 字符串类型表达式

返回: 字符串

Instr:

语法: Instr(str_expression str1, str_expression str2)

功能: 求字符串 str1 在字符串 str2 中第一次出现的位置

参数: 两个参数都为字符串表达式类型

返回: short 类型数字

Space:

语法: Space (numeric_expression)

功能: 以指定的数字长度构造一个空字符串

参数: 数字表达式类型

返回: 字符串

说明: 数字必须为整型

(8) 支持下面的日期函数:

Date:

语法: Date()

功能: 返回当前日期

参数: 无

返回: 日期类型变量

DateAdd:

语法: DateAdd (datepart, numeric_expression, datetime)

功能: 在指定的日期时间类型的指定部分上加一个数字

参数: 第一个参数为日期部分类型, 第二个参数为数字类型, 第三个参数为日期、时间或邮戳类型

返回: 对应的日期、时间或邮戳类型

说明: 日期部分有:

年 YY/YYYY; 月 M/MM; 日 D/DD; 时 HH; 分 N/MI; 秒 S/SS; 毫秒 MI

DateDiff:

语法: `DateDiff(datepart, numeric_expression, datetime)`

功能: 在指定的日期时间类型的指定部分上减去一个数字

参数: 第一个参数为日期部分类型, 第二个参数为数字类型, 第三个参数为日期、时间或邮戳类型

返回: 对应的日期、时间或邮戳类型

说明: 日期部分有:

年 YY/YYYY; 月 M/MM; 日 D/DD; 时 HH; 分 N/MI; 秒 S/SS; 毫秒 MI

DatePart:

语法: `DatePart(datepart, datetime)`

功能: 求指定日期时间类型的指定部分

参数: 第一个参数为日期部分类型; 第二个参数为日期、时间或邮戳类型

返回: 数字表示的日期部分

Now:

语法: `Now()`

功能: 求当前的日期和时间

参数: 无

返回: 邮戳类型表示的当前日期和时间

14.4.2 空间查询语句

空间查询: 是查询在某个空间范围内的实体, 既可以是矩形范围实体查询, 也可以是点附近实体查询。

矩形范围查询:

语法: `InRect(xmin,ymin,xmax,ymax)` 或 `InterRect(xmin,ymin,xmax,ymax)`

功能: 求矩形范围内的实体, `InRect` 表示只取在矩形范围内部的, `InterRect` 表示包括矩形范围内及与矩形范围相交的。

参数: 四个参数分别表示矩形范围, x 最小值, y 最小值, x 最大值, y 最大值

说明: 调用此函数之前, 必须知道矩形范围

举例: `SET @CLS-RC = SELECT RIVER FROM FCLS.'RIVER.WL' AS RIVER WHERE
INRECT(0.25,10.547,500.254,300.589);`

Near 点查询:

语法: `Near(x,y,dx,dy)`

功能: 求某个点附近的实体

参数: 四个参数分别表示 x 一点的横坐标, y 一点的纵坐标, dx—距离点的横向距离, dy—距离点的纵向距离

说明: x, y 参数必须输入, dx 和 dy 可以省略, 表示不限制范围

举例: `SET @CLS-Near = SELECT RIVER FROM FCLS.'RIVER.WL' AS RIVER WHERE`

Near (500,300);

14.4.3 空间属性联合查询语句

如果要选择的实体既要满足属性条件，又要同时满足空间条件，可以将属性条件和空间条件用 AND 操作符连接起来。

14.5 分析语句语法规则

空间分析语句将分析的结果存入结果类。

“SET @CLS-XXX=” 的语法介绍见查询语句语法规则。

空间分析的类型以 OGC 规范为基础，支持拓扑关系判断函数 EQUAL, DISJOINT, INTERSECT, TOUCH, CROSS, WITHIN, CONTAINS, OVERLAP 和空间分析函数 BUFFER, WITHINBUFF, INTERSECTION, UNION, DIFFERENCE, SYMMMDIFF。

EQUAL:

语法: EQUAL(fcls1,fcls2);

功能: 求两个要素类中相同的要素

参数: fcls1—源要素类, fcls2—目的要素类

返回: 符合条件的要素形成的新要素类

说明: 必须空间和属性信息全部相同要素才相同

DISJOINT:

语法: DISJOINT(fcls1,fcls2);

功能: 求两个要素类中不相交的要素

参数: fcls1—源要素类, fcls2—目的要素类

返回: 源要素类中符合条件的要素形成的新要素类

说明: 两个要素类的顺序不同得到的结果也不同

INTERSECT:

语法: INTERSECT(fcls1,fcls2);

功能: 求两个要素类中相交的要素

参数: fcls1—源要素类, fcls2—目的要素类

返回: 源要素类中符合条件的要素形成的新要素类

说明: 两个要素类的顺序不同得到的结果也不同

TOUCH:

语法: TOUCH(fcls1,fcls2);

功能: 求相邻的要素

参数: fcls1—源要素类, fcls2—目的要素类

返回: 源要素类中符合条件的要素组成的新要素类

说明: 两个要素类的顺序不同得到的结果也不同

CROSS:

语法: CROSS(fcls1,fcls2);

功能: 求穿过目的要素的源要素

参数: fcls1—源要素类, fcls2—目的要素类

返回: 源要素类符合条件的要素组成的新要素类

说明: 一般来说, 源要素类为线要素类, 目的要素类为区要素类

WITHIN:

语法: WITHIN(fcls1,fcls2);

功能: 求包含在目的要素中的源要素

参数: fcls1—源要素类, fcls2—目的要素类

返回: 源要素类中符合条件的要素组成的新要素类

说明: 一般来说, 目的要素类为区要素类

CONTAINS:

语法: CONTAINS(fcls1,fcls2);

功能: 求包含目的要素的源要素

参数: fcls1—源要素类, fcls2—目的要素类

返回: 源要素类中符合条件的要素组成的新要素类

说明: 一般来说, 源要素类为区要素类

OVERLAP:

语法: OVERLAP(fcls1,fcls2);

功能: 求两个要素类中相互叠加的要素

参数: fcls1—源要素类, fcls2—目的要素类

返回: 源要素类中符合条件的要素组成的新要素类

说明: 两个要素类的顺序不同得到的结果也不同

BUFFER:

语法: BUFFER(fcls,bufferLen);

功能: 对要素类中所有要素求缓冲区

参数: fcls—要素类, bufferLen—缓冲区半径

返回: 缓冲区组成的新要素类

说明: 缓冲区函数可以对点要素类、线要素类和区要素类进行

WITHINBUFF:

语法: WITHINBUFF(fcls1,fcls2,bufferLen);

功能: 求在目的要素类缓冲区中的要素

参数: fcls1—源要素类, fcls2—目的要素类, bufferLen—缓冲区半径

返回: 在缓冲区中的要素组成的新要素类

INTERSECTION:

语法: INTERSETION(fcls1,fcls2);

功能: 两个要素求交运算

参数: fcls1—源要素类, fcls2—目的要素类

返回: 两个要素类的交集组成的新要素类

UNION:

语法: UNION(fcls1,fcls2);

功能: 两个要素求并运算

参数: fcls1—源要素类, fcls2—目的要素类

返回: 两个要素类的并集组成的新要素类

DIFFERENCE:

语法: DIFFERENCE(fcls1,fcls2);

功能: 两个要素求差运算

参数: fcls1—源要素类, fcls2—目的要素类

返回: 源要素类中不与目的要素类相交的部分组成的新要素类

SYMMIDIFF:

语法: SYMMIDFF(fcls1,fcls2);

功能: 两个要素求异运算

参数: fcls1—源要素类, fcls2—目的要素类

返回: 两个要素类中互不相交的部分组成的新要素类

第 15 章 GSQL 查询分析器主要功能

15.1 窗口介绍

GSQL 查询分析器主界面由工具栏和对象浏览器窗口组成，新建的 GSQL 查询分析窗口分为脚本编辑窗口和查询结果输出窗口，另外，右键预览显示图形和属性窗口。

如图 54-1 所示：

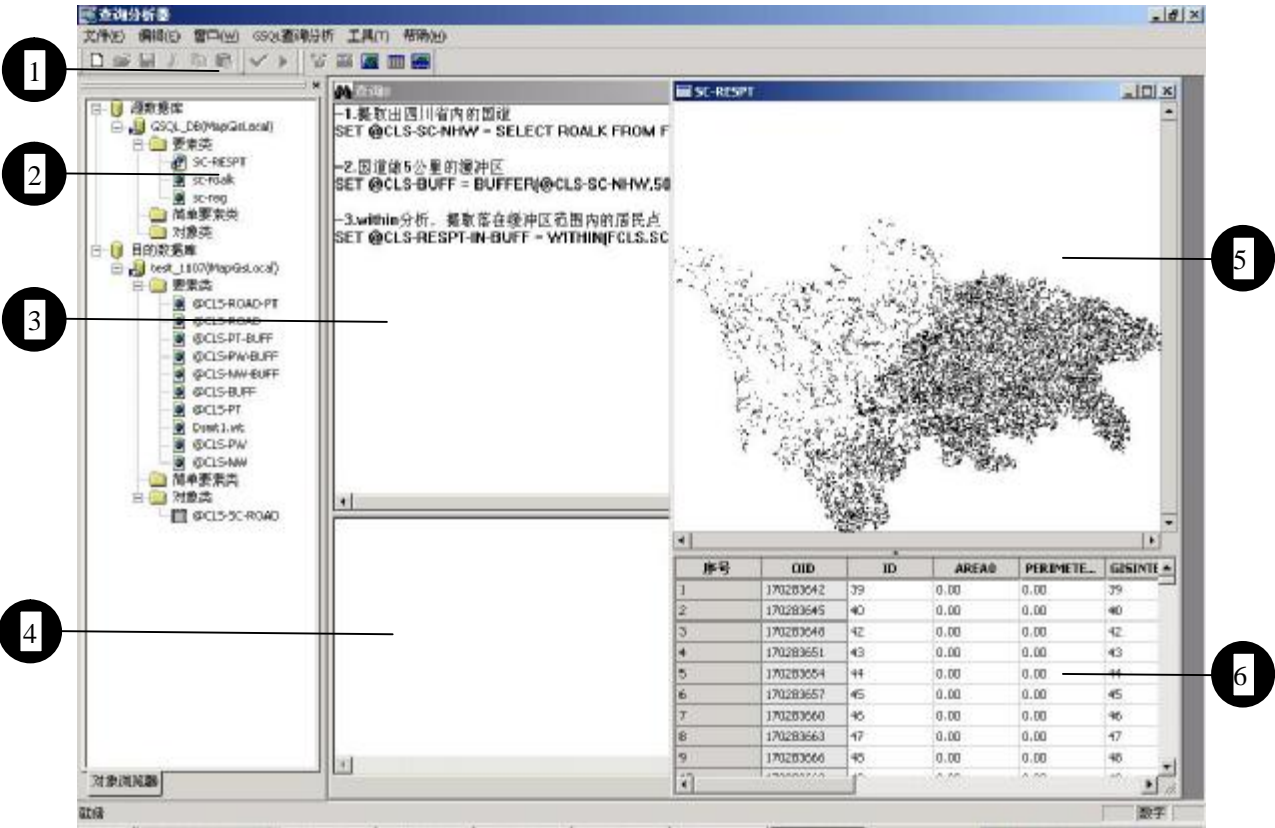


图 15-1 GSQL 查询分析器窗口介绍

15.2 查询分析步骤

1 打开数据库：

如图 15-2 所示，单击右键在弹出菜单上选择打开源数据库选项。



图 15-2 打开源数据库

在弹出的选择数据库对话框中选择源地理数据库。如图 15-3 所示：

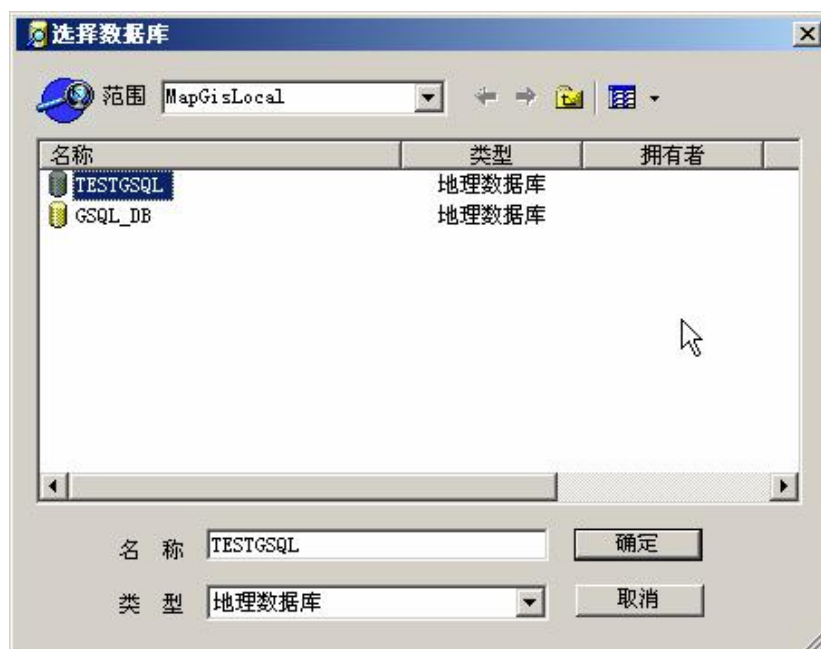


图 15-3 选择数据库

用同样的方法打开目的数据库后，对象浏览器将呈现出两个数据库的目录，如图 54-4 所示：

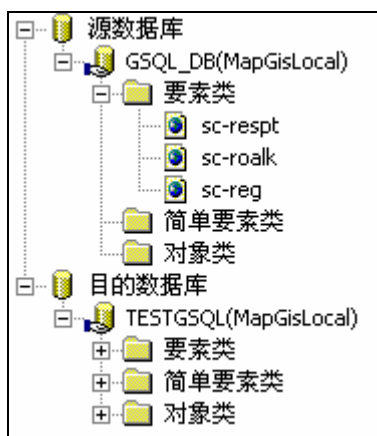


图 15-4 对象浏览器

2 打开或编辑 GSQL 脚本：

您可以在查询窗口中编辑 GSQL 脚本，如图 15-5：

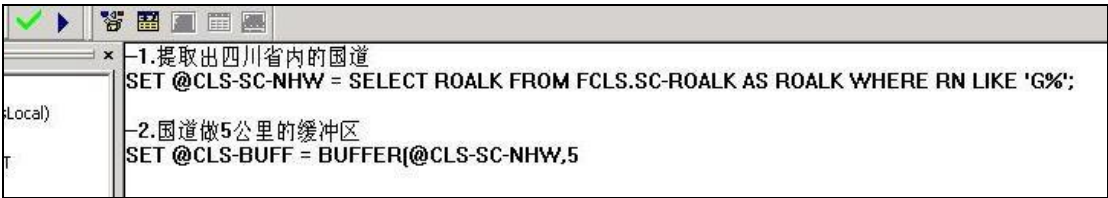


图 15-5 GSQL 脚本编辑器

通常，您也可以通过打开*.gsql 文件的方法获取编辑好的脚本，用图 15-6 所示方法的任意一种打开：



图 15-6 打开 GSQL 脚本

在弹出的对话框中选择*.gsql 文件，如图 15-7 所示：



图 15-7 打开 GSQL 脚本文件

从而获得一样的效果，如图 15-8 所示：



图 15-8 打开后的 GSQL 脚本文件

3 执行查询分析:

为了安全,您您可以先进行分析查询,分析成功后再执行查询。执行查询中包括分析查询功能,因此,您也可以直接执行查询。用图 15-9 所示的任意一种方法可以实现这个操作:



图 15-9 查询分析菜单

每一个 GSQL 语句块的操作结果都将存入目的数据库中,执行操作的同时,查询的状态和时间信息将在在输出窗口中显示如图 15-10。

```

分析查询成功!

正在执行第1个语句块,请等待...
—开始时间: 11:46:34:187
—结束时间: 11:46:49:562

执行查询成功!

正在执行第2个语句块,请等待...
—开始时间: 11:46:49:578
—结束时间: 11:46:55:781|

执行查询成功!

正在执行第3个语句块,请等待...
  
```

图 15-10 输出信息

4 检查结果，保存 GSQL 脚本和输出信息：

执行完毕，在对象浏览器中的目的要素类中各个语句块的执行结果。同时，您还可以把 GSQL 脚本和执行输出的信息保存。保存输出信息通过图 15-11 所示的方法实现：

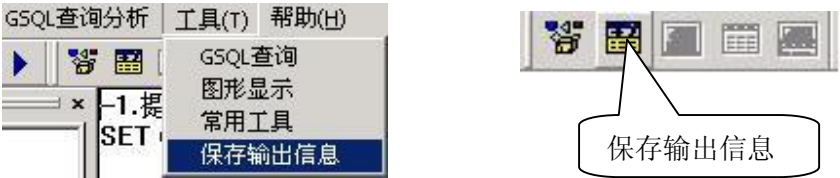


图 15-11 保存输出信息

在弹出的对话框中输入名称，选择路径保存，保存的类型为文本文件，如图 15-12 所示：



图 15-12 保存输出信息

15.3 对象浏览器

对象浏览器是管理源和目的数据库信息的窗口。由于 GSQL 查询分析器只允许要素类、简单要素类和对象类的查询和分析，因此，数据库下也只管理这三个类的信息。

对象浏览器目录分为三个层次：其管理内容如图 15-13 所示：

对于源和目的数据库标签，只允许进行数据库的打开和刷新操作。

对于打开的源和目的数据库，可以浏览其属性并进行刷新。

对于要素类、简单要素类和对象类目录下的各个类，可以查看信息（包括图形和属性）、导出数据、重命名、删除和查看类的属性信息。

下面着重介绍几个基本功能。

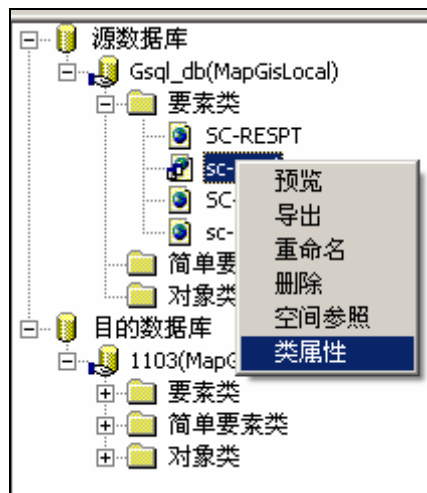


图 15-13 对象浏览器

1 查看数据库属性:

单击数据库名称，点击鼠标右键，在弹出的菜单中选择“属性”，则会显示该数据库的属性信息，如图 54-14 所示：

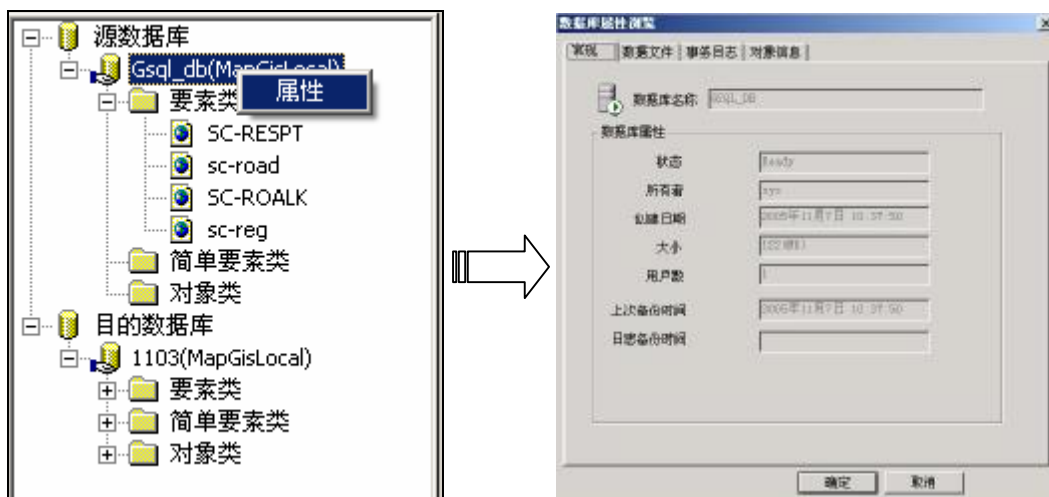


图 15-14 浏览数据库属性

2. 导出:

要素类、简单要素类和对象类都支持数据导出功能，不同的是，对于要素类和简单要素类而言，导出时只对属性数据进行，而丢失了图形数据，因此，数据导出的结果都是属性数据。

(1) 单击要导出地类，在右键菜单中选择“导出”，如图 15-15 所示：



图 15-15 导出数据

(2) 选择导出数据库类型，如图 54-16 所示：



图 15-16 选择数据库类型

(3) 点击 “...” 按钮，在弹出的对话框中选定存放导出表格的数据库，如图 15-17 所示：



图 15-17 选择数据库表格文件

(4) 如有需要, 输入用户名和密码。并选择是否要处理 Unicode, 如图 15-17 所示。

(5) 单击“下一步”按钮。

(6) 选中需要转换的要素类, 点击复选框使之被选中即可。如有需要, 还可以点击“转换”下的按钮, 设置需转换的字段名。如图 15-18 所示:



图 15-18 设置转换字段

(7) 单击“下一步”按钮。

(8) 确认源数据库、目的数据库、要素类名等转换信息, 如果无误, 点击“完成”按钮, 否则

点击“上一步”按钮进行修改，如图 15-19 所示：



图 15-19 检查信息

(9) 单击“完成”按钮，开始导出数据，如图 15-20 所示：



图 15-20 数据导出成功

(10) 点击“成功”按钮退出。

3 查看：

要素类和简单要素类可以同时显示图形和属性信息，而对象类没有图形信息，因此显示属性信息，如图 15-21：

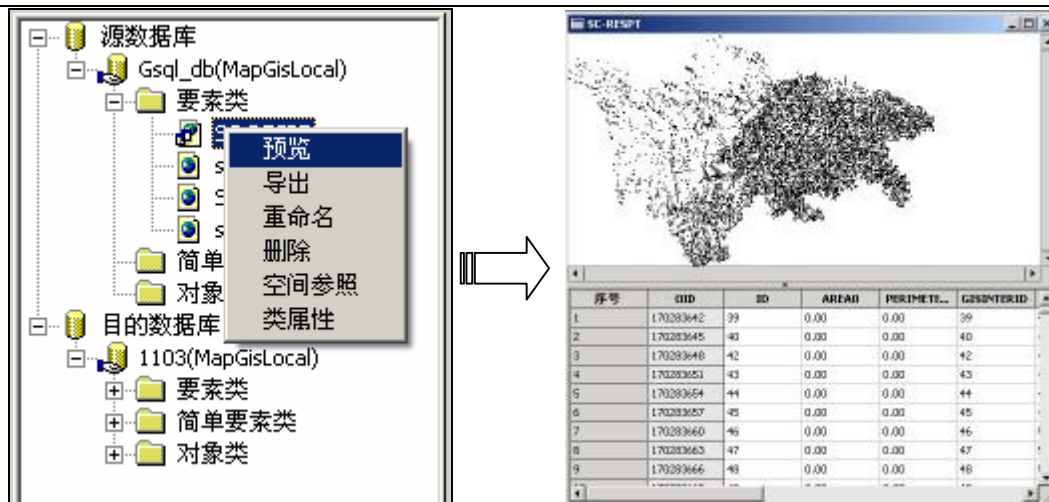


图 15-21 查看图形和属性

4 重命名:

双击类名称，可编辑后输入新的名称即可。同时支持 F2 键的重命名。

5 删除:

删除是很危险的操作，数据一旦删除，将无法再恢复。必须慎用。

6 查看类属性:

点击要查看的类，在鼠标右键菜单中选择“查看类属性”，可以浏览该类总体描述的信息，如图 15-22 所示:



图 15-22 查看类属性

6 查看修改空间参照

点击要操作的类，在鼠标右键菜单中选择“空间参照”，可以浏览该类的空间参照，如图 15-23 所示，还可以修改该类的空间参照。

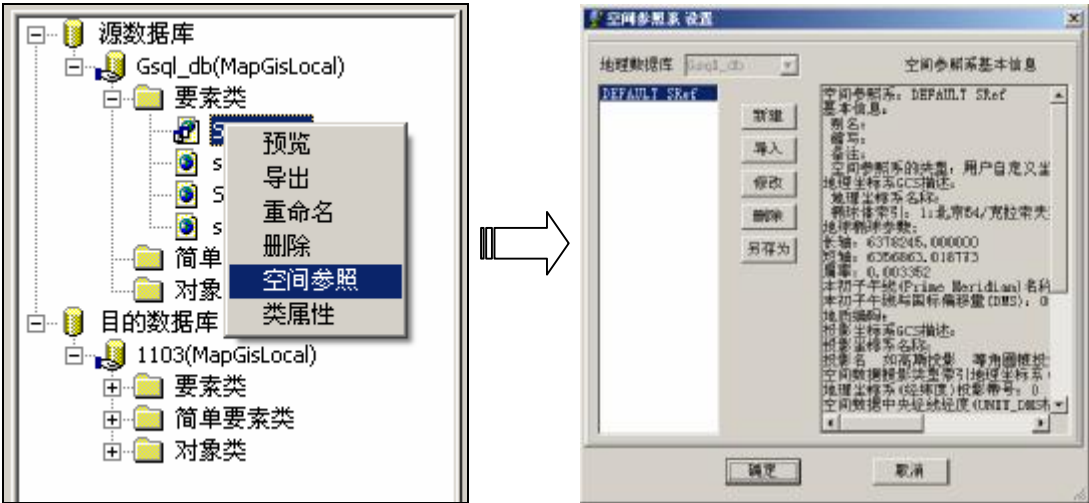


图 15-23 查看修改空间参照

15.4 工具栏

查询分析器工具栏按钮功能介绍如下：

| | |
|--|--|
|  新建查询分析窗口 |  分析查询 |
|  打开 GSQL 脚本文件 |  执行查询 |
|  保存 GSQL 脚本文件 |  对象浏览器 |
|  剪切 |  保存输出文本 |
|  复制 |  显示图形 |
|  粘贴 |  显示属性 |
|  显示所有信息 | |

第 16 章 GSQL 应用案例

案例 1

题目：已知四川省全部居民点（SC-RESPT），求四川省乡（GB=31090）的个数。

分析：此题是要求输出个数，因此不必将所有的乡级居民点提取出来（这样太耗费时间和空间，而且没必要），而是直接使用聚类函数 count，计算乡的个数。

GSQL 脚本：

```
SET @VAL-SC31090 = SELECT COUNT(*) FROM FCLS.SC-RESPT AS SC_DOT  
WHERE GB=31090;
```

执行结果：

查询结果如果是数字，则数字结果不保存，而是在输出信息中显示出来，如图 16-1 所示：

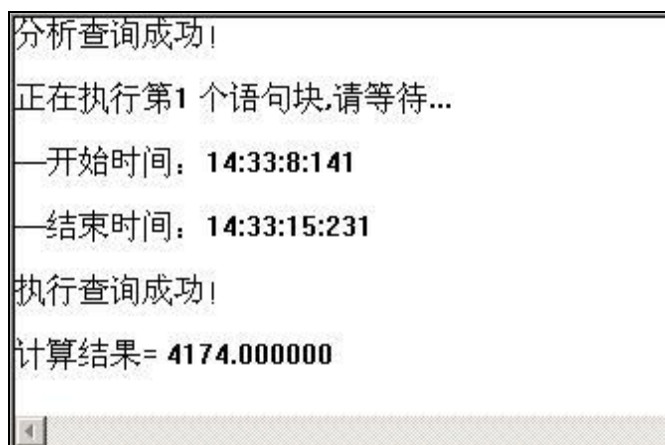


图 16-1 案例 1：输出信息

案例 2

题目：已知四川省的道路（SC-ROALK），查询四川省内的国道（RN="Gxxx"），将其属性信息按照国道号的顺序排列。

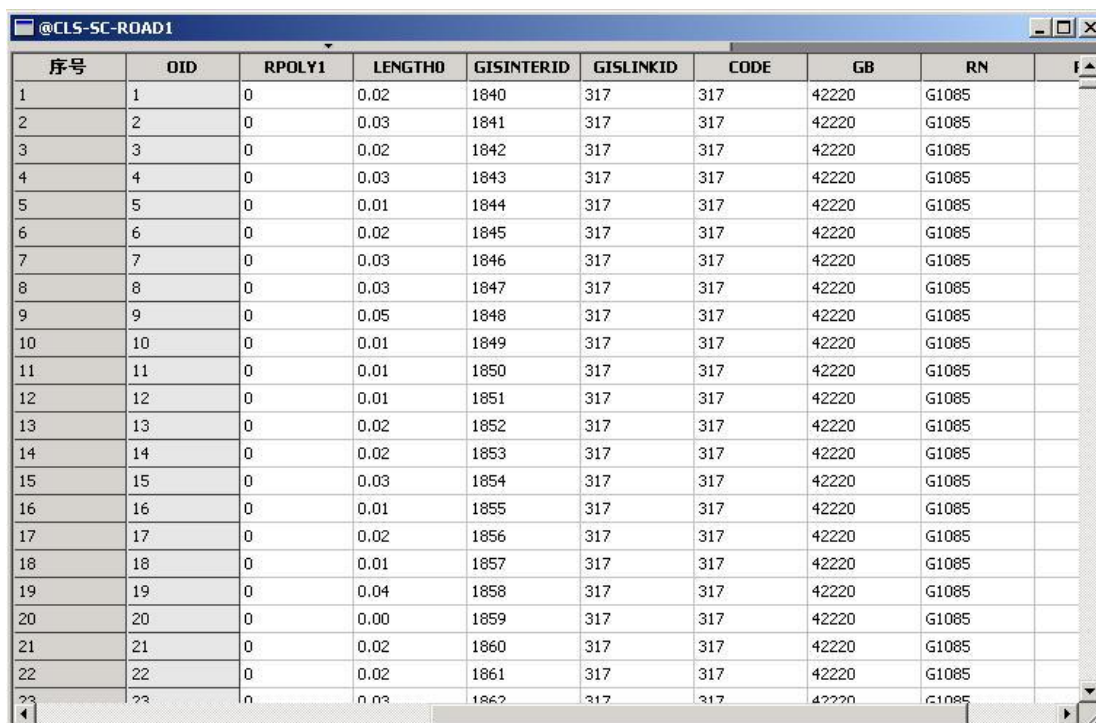
分析：此题是查询国道的属性信息，因此不必提取空间数据，而是用 ATT 实现只提取要素类的属性信息，并用 ORDER BY 现排序功能。国道的提取属于模糊查询，因此使用 LIKE 操作符。

GSQL 脚本：

```
SET @CLS-SC-ROAD = SELECT ATT FROM FCLS.SC-ROALK AS ROALK WHERE  
RN LIKE 'G%' ORDER BY RN;
```

执行结果：

此题的执行结果是对象类，在对象类目录中打开后显示如图 16-2，没有图形信息。



| 序号 | OID | RPOLY1 | LENGTHO | GISINTERID | GISLINKID | CODE | GB | RN | |
|----|-----|--------|---------|------------|-----------|------|-------|-------|--|
| 1 | 1 | 0 | 0.02 | 1840 | 317 | 317 | 42220 | G1085 | |
| 2 | 2 | 0 | 0.03 | 1841 | 317 | 317 | 42220 | G1085 | |
| 3 | 3 | 0 | 0.02 | 1842 | 317 | 317 | 42220 | G1085 | |
| 4 | 4 | 0 | 0.03 | 1843 | 317 | 317 | 42220 | G1085 | |
| 5 | 5 | 0 | 0.01 | 1844 | 317 | 317 | 42220 | G1085 | |
| 6 | 6 | 0 | 0.02 | 1845 | 317 | 317 | 42220 | G1085 | |
| 7 | 7 | 0 | 0.03 | 1846 | 317 | 317 | 42220 | G1085 | |
| 8 | 8 | 0 | 0.03 | 1847 | 317 | 317 | 42220 | G1085 | |
| 9 | 9 | 0 | 0.05 | 1848 | 317 | 317 | 42220 | G1085 | |
| 10 | 10 | 0 | 0.01 | 1849 | 317 | 317 | 42220 | G1085 | |
| 11 | 11 | 0 | 0.01 | 1850 | 317 | 317 | 42220 | G1085 | |
| 12 | 12 | 0 | 0.01 | 1851 | 317 | 317 | 42220 | G1085 | |
| 13 | 13 | 0 | 0.02 | 1852 | 317 | 317 | 42220 | G1085 | |
| 14 | 14 | 0 | 0.02 | 1853 | 317 | 317 | 42220 | G1085 | |
| 15 | 15 | 0 | 0.03 | 1854 | 317 | 317 | 42220 | G1085 | |
| 16 | 16 | 0 | 0.01 | 1855 | 317 | 317 | 42220 | G1085 | |
| 17 | 17 | 0 | 0.02 | 1856 | 317 | 317 | 42220 | G1085 | |
| 18 | 18 | 0 | 0.01 | 1857 | 317 | 317 | 42220 | G1085 | |
| 19 | 19 | 0 | 0.04 | 1858 | 317 | 317 | 42220 | G1085 | |
| 20 | 20 | 0 | 0.00 | 1859 | 317 | 317 | 42220 | G1085 | |
| 21 | 21 | 0 | 0.02 | 1860 | 317 | 317 | 42220 | G1085 | |
| 22 | 22 | 0 | 0.02 | 1861 | 317 | 317 | 42220 | G1085 | |
| 23 | 23 | 0 | 0.03 | 1862 | 317 | 317 | 42220 | G1085 | |

图 16-2 案例 2：执行结果对象类

案例 3

题目：已知四川省的居民点（SC-RESPT）和道路（SC-ROALK），查询四川省内距离国道 5 公里范围内的居民点。

分析：此题是一个综合查询分析题目，也是地理信息系统中使用最频繁的查询分析之一。

首先是要提取出四川省内的国道（属性条件查询），然后对国道做 5 公里的缓冲区（缓冲区 buffer 分析），最后做范围分析，提取出在缓冲区范围内的居民点（within 分析）。

注意：缓冲区分析时必须转换为以米为单位。

GSQL 脚本：

--1.提取出四川省内的国道

```
SET @CLS-SC-NHW = SELECT ROALK FROM FCLS.SC-ROALK AS ROALK
                    WHERE RN LIKE 'G%';
```

--2.国道做 5 公里的缓冲区

```
SET @CLS-BUFF = BUFFER(@CLS-SC-NHW,5000);
```

--3.within 分析，提取落在缓冲区范围内的居民点

```
SET @CLS-RESPT-IN-BUFF = WITHIN(FCLS.SC-RESPT,@CLS-BUFF);
```

执行结果:

图 16-3、图 16-4 和图 16-5 分别是三个步骤的执行结果:

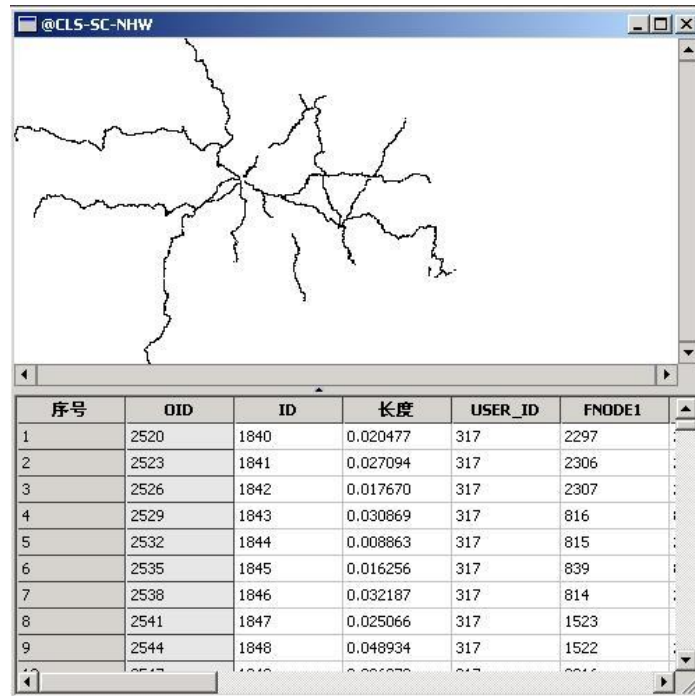


图 16-3 案例 3: 第一步结果

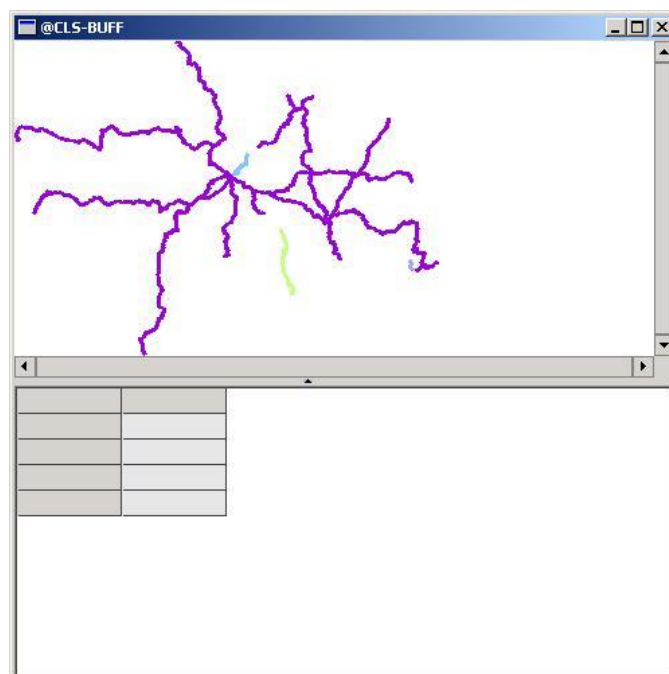


图 16-4 案例 3: 第二步结果

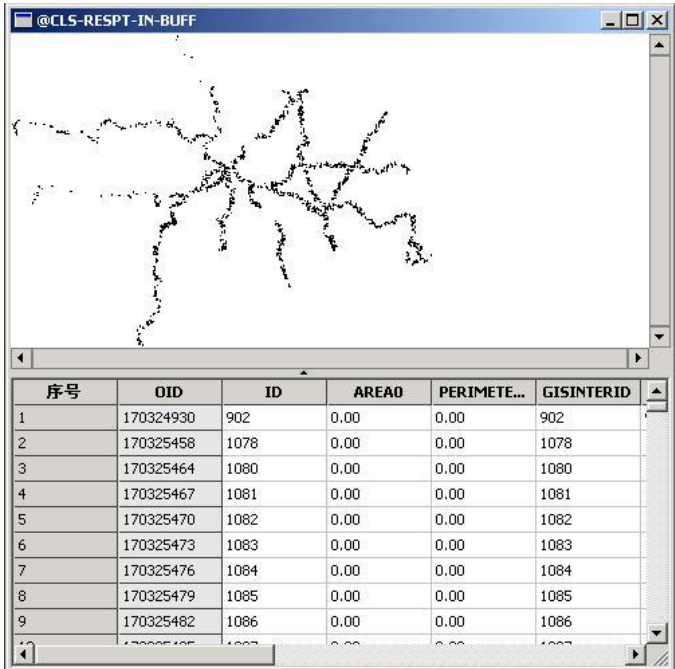


图 16-5 案例 3：第三步结果

案例 4

题目：空间分析实例。

- (1) 提取四川省内的道路（SC-ROALK）中提取国道（RN= “Gxxx”）和省道（代码 RN= “Sxxxxx”），国道两边生成 5KM 的缓冲带，省道两边生成 4KM 的缓冲带；
- (2) 从四川省的居民点（SC-RESPT）中提取镇以上的居民点（GB=31010、GB=31020、GB=31030、GB=31040、GB=31050、GB=31060、GB=31061、GB=31080、GB=31090），以每个居民点为中心生成半径为 8KM 的缓冲区；
- (3) 合并以上生成的缓冲区（空间分析求和）；

分析：此题是一个常用的空间分析题目。题目主要是完成一系列的数据提取和空间分析，题目已经将允许步骤描述的非常清楚，与上面其他题目不同的就是要实现空间分析（即 UNION 操作），见步骤 7、8。

GSQL 脚本：

--1.提取国道

```
SET @CLS-NW = SELECT ROAD FROM FCLS.SC-ROALK AS ROAD WHERE RN LIKE 'G%';
```

--2.提取省道

```
SET @CLS-PW = SELECT ROAD FROM FCLS.SC-ROALK AS ROAD WHERE RN LIKE 'S%';
```

--3.国道两边做 5KM 的缓冲区

```
SET @CLS-NW-BUFF = BUFFER(@CLS-NW,5000);
```

--4.省道两边做 4KM 的缓冲区

```
SET @CLS-PW-BUFF = BUFFER(@CLS-PW,4000);
```

--5.提取镇以上的居民点

```
SET @CLS-PT = SELECT PT FROM FCLS.SC-RESPT AS PT WHERE GB=31010 OR GB=31020 OR GB=31030 OR GB=31040 OR GB=31050 OR GB=31060 OR GB=31061 OR GB=31080 OR GB=31090;
```

--6.居民点周围做 8KM 的缓冲区

```
SET @CLS-PT-BUFF = BUFFER(@CLS-PT,8000);
```

--7.合并国道和省道的缓冲区

```
SET @CLS-ROAD = UNION(@CLS-NW-BUFF,@CLS-PW-BUFF);
```

--8.合并道路和居民点的缓冲区

```
SET @CLS-ROAD-PT = UNION(@CLS-ROAD,@CLS-PT-BUFF);
```

执行结果:

此题运行时间比较长,因此可以清楚地看到各步的执行过程,而且结果比较多,不再一一列出,图 16-6 和图 16-7 显示的是第 7 个语句块和最终的执行结果。

