CHAPTER FOUR – DATA TABLES AND DATA PREPROCESSING

INTRODUCTION

GIS programs link attribute data files to digital maps. The previous Chapter focused on the map side of this equation. Let's focus now on the attribute data files. Like the previous chapter, this chapter examines several key concepts and covers the preprocessing of your GIS data, but it specifically focuses on attributes, data files, and the editing of your attribute data. The concepts focus on attribute data and principles of raster and vector database management. Understanding these concepts will help you to effectively edit and manage your attribute data. The bulk of the chapter focuses on various preprocessing routines including adding and deleting fields, deleting records, joining data files, selecting and sorting records, calculating attributes, and geocoding. The chapter ends with a short discussion regarding attribute verification.

ATTRIBUTE DATA

As described in the previous chapter, spatial data occupies geographic space. It has a specific location that is tied to one of the world's geographic referencing systems (like latitude and longitude). Besides spatial data, GIS files contain non-spatial attributes that describe the spatial features. This section focuses on these non-spatial attributes.

Related to the discussion of "measurements of scale" in Chapter 2, your attributes can be classified as either qualitative or quantitative and actual or derived. Quantitative data focus on numbers and frequencies rather than on subjectivity, meaning, and experience. They are easy to analyze statistically, and their values are often the result of field work and laboratory experiments. Maps exhibiting quantitative data depict differences in magnitude among features.

Qualitative data, by contrast, often provide deeper description and meaning. Maps displaying qualitative data show differences in kind or type. You might subjectively judge whether a quantity is low, medium, or high. You might also classify detailed land uses into broader categories of residential, commercial, and industrial. The statistical options are narrowed too due to the subjectivity of the data and the categorization of data into classes.

Data can also be defined by whether they represent some intrinsic characteristic of the feature being measured (absolute), or whether they are in a sense “created” (derived). Absolute data consists of both the quantitative and qualitative data just described, but it represents phenomena that are measured (like election data or the amount of water stored), the ranking and rating of attributes (even though this process can be subjective), and personal, subjective accounts gained from questionnaires and surveys.

Derived attributes either do not occur naturally, or they cannot be directly gathered; they are the result of statistical manipulation that produces the data. An example is average July temperatures, which is the calculated result of averaging many actual temperature values. Derived data may result from averaging actual values like these, or they represent the relationships between already gathered attribute data, which take three forms: ratio, proportion, and percentage.

Ratio attributes are derived when the value of one attribute is divided by the value of another. Population density is a good example. The total number of people within a particular region is divided by the region's area. Both the population and area attributes may be “actual” values, but the calculated population density attribute is derived.

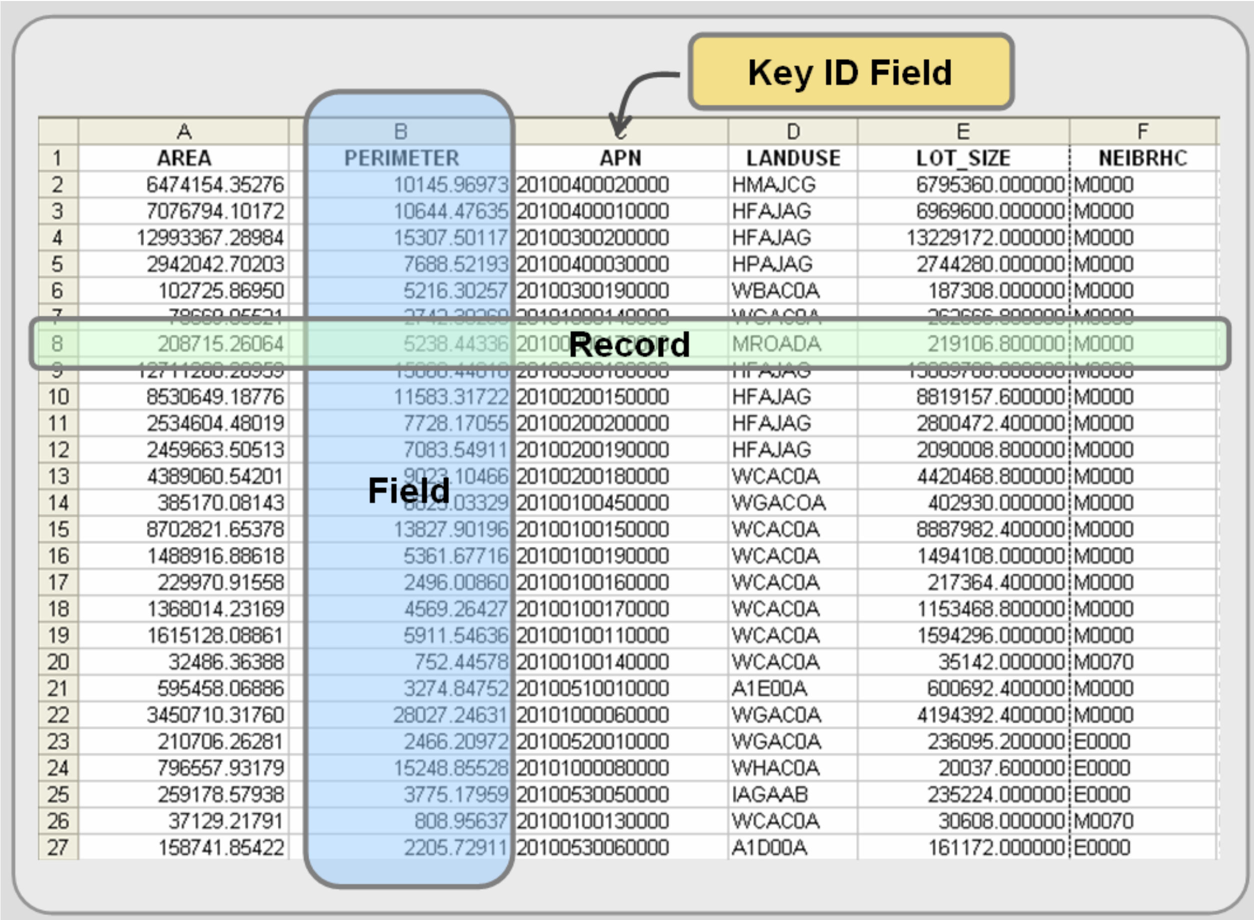
Proportion compares the value of one attribute to the total value of all related attributes. The proportion of all African-Americans to the total population is derived by dividing the number of African-Americans (actual data) by the total number of people (also actual).

Many people think of proportions as *percentages*; they are similar, but percentages multiply proportions by one hundred.

PRINCIPLES OF DATABASE MANAGEMENT - VECTOR

Let's turn our discussion from characteristics of data to how these values are organized within a data file. Data files are the basic "database" for many programs including spreadsheets, statistic programs, and GIS. Within a GIS, there is a data file for each particular type of geographic feature (e.g. streets, street lights, buildings, and parcels of land). They are the database's version of your features. The data files are automatically created when feature layers are defined in your GIS. You place into them the attributes related to the features.

Data files, often called "tables," arrange attributes within a matrix of fields and records. Fields form the columns of a data file (see Figure 4.1), and they contain the values for each specific attribute you are collecting. For example, parcels might include attributes such as area, land use, and Assessor's Parcel Number (APN). In this example, you would have at least three fields: one called area, another titled land use, and one labeled APN.



	A	B	C	D	E	F
1	AREA	PERIMETER	APN	LANDUSE	LOT_SIZE	NEIBRHC
2	6474154.35276	10145.96973	20100400020000	HMAJCG	6795360.000000	M0000
3	7076794.10172	10644.47635	20100400010000	HFAJAG	6969600.000000	M0000
4	12993367.28984	15307.50117	20100300200000	HFAJAG	13229172.000000	M0000
5	2942042.70203	7688.52193	20100400030000	HPAJAG	2744280.000000	M0000
6	102725.86950	5216.30257	20100300190000	WBAC0A	187308.000000	M0000
7	[illegible]	[illegible]	[illegible]	WCAC0A	[illegible]	M0000
8	208715.26064	5238.44336	2010[illegible]	MROADA	219106.800000	M0000
9	[illegible]	[illegible]	[illegible]	HFAJAG	[illegible]	M0000
10	8530649.18776	11583.31722	20100200150000	HFAJAG	8819157.600000	M0000
11	2534604.48019	7728.17055	20100200200000	HFAJAG	2800472.400000	M0000
12	2459663.50513	7083.54911	20100200190000	HFAJAG	2090008.800000	M0000
13	4389060.54201	9023.10466	20100200180000	WCAC0A	4420468.800000	M0000
14	385170.08143	[illegible]	20100100450000	WGACOA	402930.000000	M0000
15	8702821.65378	13827.90196	20100100150000	WCAC0A	8887982.400000	M0000
16	1488916.88618	5361.67716	20100100190000	WCAC0A	1494108.000000	M0000
17	229970.91558	2496.00860	20100100160000	WCAC0A	217364.400000	M0000
18	1368014.23169	4569.26427	20100100170000	WCAC0A	1153468.800000	M0000
19	1615128.08861	5911.54636	20100100110000	WCAC0A	1594296.000000	M0000
20	32486.36388	752.44578	20100100140000	WCAC0A	35142.000000	M0070
21	595458.06886	3274.84752	20100510010000	A1E00A	600692.400000	M0000
22	3450710.31760	28027.24631	20101000060000	WGAC0A	4194392.400000	M0000
23	210706.26281	2466.20972	20100520010000	WGAC0A	236095.200000	E0000
24	796557.93179	15248.85528	20101000080000	WHAC0A	20037.600000	E0000
25	259178.57938	3775.17959	20100530050000	IAGAAB	235224.000000	E0000
26	37129.21791	808.95637	20100100130000	WCAC0A	30608.000000	M0070
27	158741.85422	2205.72911	20100530060000	A1D00A	161172.000000	E0000

Figure 4.1: Key parts of a data file.

Remember from Chapter 2 that each of these fields has a specific "data format" that defines the type and length of the value that can be directly entered into the data file. Frequently attributes are coded as one of the following, but there are many data formats and the specific name of the data format often changes from one software program to another. Broad data format categories include:

Integer	Numeric values consisting of whole numbers. No decimals.
Real	Numbers consisting of integers with decimals.
Byte	Numeric values ranging from 0 to 255.
Character	Alphanumeric values.

Figure 4.2: Data format categories.

A single record, a row in the data file, represents the database's version of a single feature, including all of its specific attribute values (see Figure 4.1). A few of these attributes may be system variables that the GIS needs for data integrity reasons and to link the data file to the feature's spatial files. In addition, some GIS programs automatically generate length calculations for line features and both area and perimeter calculations for polygon features. Each data file should have a key identifier field that uniquely identifies each feature (i.e. each record). The remaining attributes are up to you and the purpose of your study.

Data files are a collection of related records. If you have 25 street lights within your GIS, you will have 25 street light records in its attribute file. As briefly described above, a largely empty data file is created when a new layer is defined within a GIS program. It is your job to add fields and attribute values to the data file. These descriptive attributes can be entered by hand or imported from external sources. It is likely that you will enter some attributes by hand (and it can be time consuming and tedious), but many—if not most—of the attributes you seek will be imported or "joined" from separate, non-GIS data files. This is because many non-spatial data files predate your need for their incorporation into a GIS, but it is deeper than that. Data manipulation within GIS is clumsy, and since most GIS users are familiar with data management programs like Excel and Access, they prefer working with these programs and then exporting their data and "joining" the external data file to the GIS data file. The joining process is described later in this chapter.

These external data files are coded in one of many "file formats". Some file formats are specific to a particular software program while others are somewhat universal. Even those using a program's proprietary format can export the data file into one of many formats that most GIS programs can read. Some of the file formats that can be read by most GIS programs include:

dBase This industry standard format is read by just about every GIS program. Many GIS programs use this format internally rather than creating their own.

Excel and **Access** - Microsoft's file formats for Excel and Access can be read by many GIS programs. If your GIS program does not read these formats, open the data file in Excel or Access and export it into a format that your system reads.

ASCII (American Standard Code for Information Interchange) – Since most computers use ASCII to represent text, it is possible to transfer data from one computer to another in this format. It is also read and written by most GIS programs, but it is rarely used as the primary GIS file format (with the exception of some raster-based GIS programs). Some government data sets are contained in this file format. Text files come in several different "delimited" forms, and all may include numeric or alphanumeric content (see "Joining Data Files" later in this chapter).

Data files contain a matrix of fields and records for each feature layer. A database is a collection of several related data files (like parcels, street lights, and buildings). In other words, databases contain data files for related layers. Accessing these data files are done through either the GIS software or increasingly from external database management systems (DBMS) that are linked to the GIS. DBMS are specialized programs that organize, manipulate, and report non-spatial data and help you store your data more efficiently. They are particularly valuable when working with large data sets because you can select a subset of your records and fields to work with. The entire attribute file does not have to be used. Examples of external DBMS programs include Access, Oracle, Ingres, SQLServer, INFORMIX, and to a lesser degree Excel, which can serve as an elementary database program. Regardless of whether you are accessing the data files within the GIS software or from an external DBMS, all databases have standard operations which include sorting and selecting records, deleting records and fields, and editing fields and attributes.

Different databases have different structures or ways to organize data. The hierarchical and network data models are two examples, but they are rarely used for GIS (and so will be skipped in this section). For vector systems, the relational database model is the most common data model arguably because they are more flexible, the table structure is easy to understand and program, and outside of GIS, data files are commonly held in relational databases.

Linking or joining data files is the relational database model's strength. Key identifiers, found in multiple data files, are used to link records from one data file to another. In other words, you cross reference multiple data files using common attributes and attach (or join) these external data files to your internal GIS data file. This link takes the selected fields in the data file you wish to join and relates them to the appropriate records in the GIS data file. This requires that each data file have at least one common field to perform a join. There are different names for the key identifier including key and primary key. This process is highlighted later in this chapter.

Many, however, think that the relational database model does not adequately represent spatial data. For some, records in a relational data file are too discrete; they do not properly depict the continuous and multi-dimensional nature of the features they are representing. We use relational data models because they are simple and convenient, but we artificially bend geographic features to conform to existing database standards that were created for non-spatial data.

This has led to the development of object-oriented data structures, which are seen as a more sophisticated database model. The database discards many of the foundational concepts that we have applied throughout this book. Features are defined differently; object-oriented features blur the line between points, lines, and polygons. Also, instead of having multiple files for each GIS layer, the geography and attribute data are integrated into a single file. This allows for simultaneous geographic and attribute editing and quicker processing. The more sophisticated model, however, is a more complex model, and that may have slowed its spread even though “object-oriented” databases were one of the hottest topics in GIS in the 1990s. It may still be the touted successor of the relational model, but it seems that the relational model, despite its drawbacks, has significant pluses—including its ease of use—that will help it dominate at least into the near future.

PRINCIPLES OF DATABASE MANAGEMENT - RASTER

As described in Chapter 1, the raster data model aligns the Earth's surface into a grid of columns and rows. Cells, or pixels, the building blocks of the raster data model, form at the intersection of the columns and rows, and each cell contains a single attribute value, representing the condition of a specific portion of the Earth's surface. That means that a single raster layer only contains the values for one specific attribute across space. That last point is important because raster layers fill space. Their attributes occur everywhere in the study area; there are no blank spaces. Empty areas get a “0” value, but every pixel gets a value. If you need more than one attribute, you construct multiple layers, each containing a single specific attribute for the same area. Conceptually, it is a simple model. As in Figure 4.3, your study area is divided into cells, and each cell of each layer has a single attribute that represents that area.

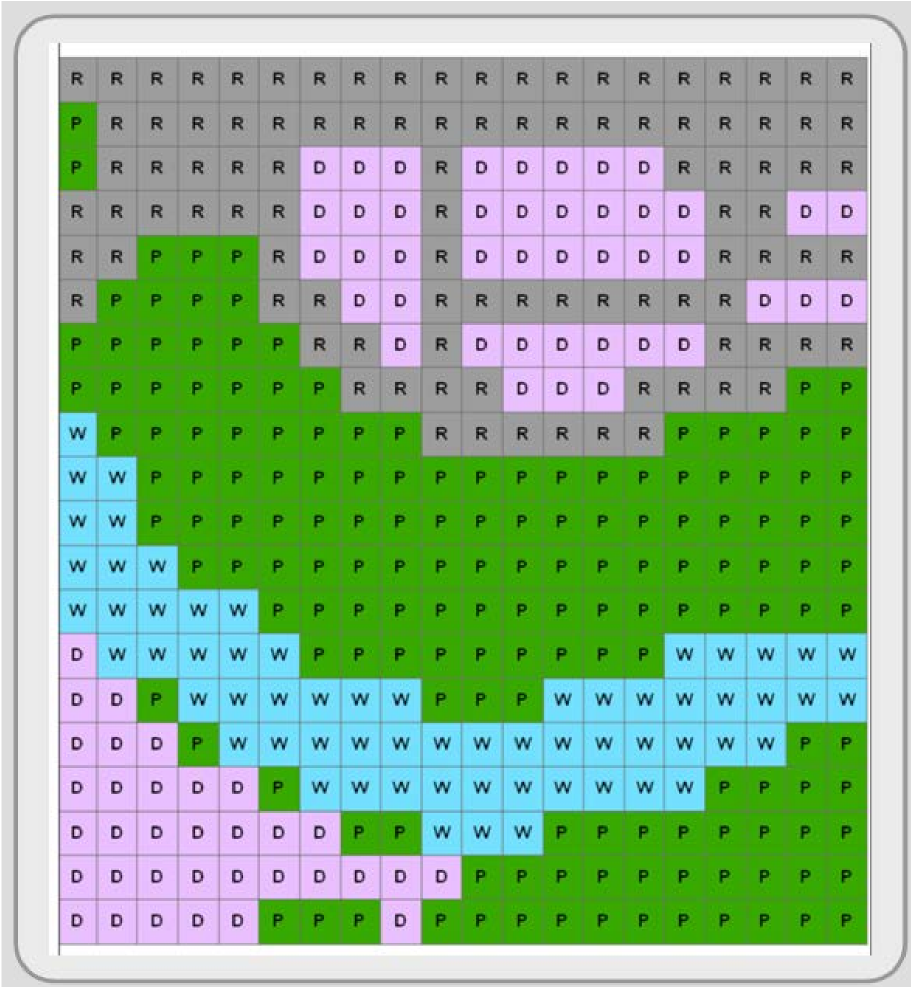
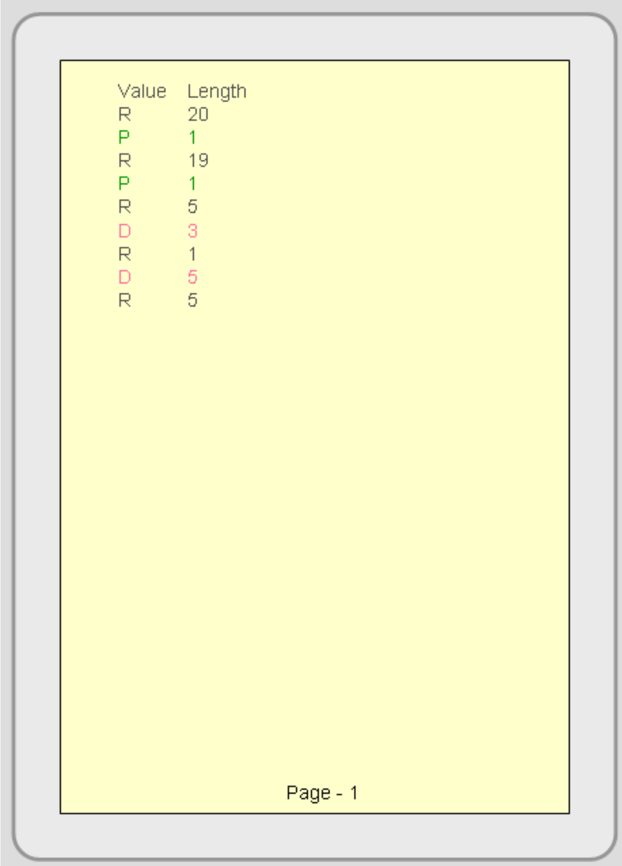


Figure 4.3: Raster image.

There are many ways—some more complex than others—the raster data model may be stored. The two general categories are regular and irregular. The regular structure is conceptually simple, and includes two types: full raster encoding and run-length encoding. Full raster encoding creates a data file that records the attribute value for every pixel. It's as though you read an image's pixels like a book, starting in the upper left corner and reading from left to right and downward row by row. The data file looks a bit different. It records each pixel's attribute value on a separate line, so if you had an image with 640,000 pixels, your data file would have 640,000 lines, making it a very long data file. Figure 4.4 is a simplified example.

Figure 4.4: Full raster encoding. This figure is the beginning—just the first three rows—of the data file for the image in Figure 4.3. Color is added to highlight the different attribute values.

Run-length encoding is more efficient than full raster encoding. Since the same values often occur in runs across several cells, run-length encoding enters the attribute values as pairs: the first number is the run length and the second number is the cell's value. This substantially reduces file size especially if contiguous pixels have the same value. Contrast Figure 4.5 with Figure 4.4.



Value	Length
R	20
P	1
R	19
P	1
R	5
D	3
R	1
D	5
R	5

Page - 1

Figure 4.5: Run-length encoding. This figure also depicts the first three rows of Figure 4.3. Compare run-length encoding with full raster encoding (Figure 4.4). Color is added to highlight the different attribute values.

Irregular raster data structures, like quadtree and others, are more complex, proprietary, and beyond the scope of this e-text. They usually make file size smaller and provide ways to store raster data for quick retrieval.

ATTRIBUTE PREPROCESSING AND EDITING

When you add feature layers, containing both spatial and attribute data, to an active workspace, the attribute data file might not be immediately visible. Opening and editing the attribute files are easy processes, but they are specific to individual programs. Once the attribute table is open, you can enter data by typing attribute values directly into the data file or loading and joining external data files to it. Other processes like editing attributes, adding or deleting fields, deleting records, querying attributes (record selection), calculating fields, and geocoding are completed through the data file interface.

Adding and deleting fields

As described above, fields define feature attributes. Most GIS programs provide a way for you to add or delete fields from within your open data file. The GIS program will instruct you to define a new field. You will give it a name and select from options that determine the data format of the values that will be placed into the field. Deleting a field usually involves selecting the field and deleting it.

Deleting records

You can delete a single record or a group of records in a data file by first selecting them and then deleting them. Since records are the database representation of features, when you delete records in the attribute file, you are also permanently discarding their spatial representation. The entire feature, graphic and record, is deleted.

Generally, you can not add a record through the data file interface because it must also be represented spatially. See Chapter 3 for how to add a feature. Its record is automatically created when the graphic feature is added to the workspace.

Joining Data Files

Once a GIS layer is created, its attribute file can be linked ("joined") to external data files. Joining is one of the most frequently performed data file processes because it brings together feature attributes that are contained in multiple digital data files. To perform a join, a unique matching field, the key identifier, must be observed in both data files. As stated in Chapter 3, the key identifier could be something like a social security number or an assessor parcel number. It is a field that gives the feature a unique identification. Once linked, the join can be temporary or made permanent.

The external files that you load into the GIS to perform a join are typically in file formats such as dBase, ASCII, Microsoft Excel, or Microsoft Access. The precise steps involved in joining together two files are software specific, but it usually involves:

1) loading the external file that you wish to join to the GIS attribute file,

2) selecting the external file and the GIS attribute file that you wish to join,

3) selecting the field (containing the key identifier) in each file, and when joined,

4) making sure that the join was successful.

In the example in Figure 4.6, the parcel layer exists, but it does not include assessed value. It does contain a field named APN (Assessor's Parcel Number) whose values are unique to each record and which could be used to join other data files. A spreadsheet file, with assessed value, also exists, and it must be loaded into the GIS either in its native format (if accepted) or exported from the spreadsheet program to a format that the GIS can read. The spreadsheet has a field named APN_NUM, which, after a visual check, has the same values as those under APN in the parcel layer, and it can be used to perform the join.

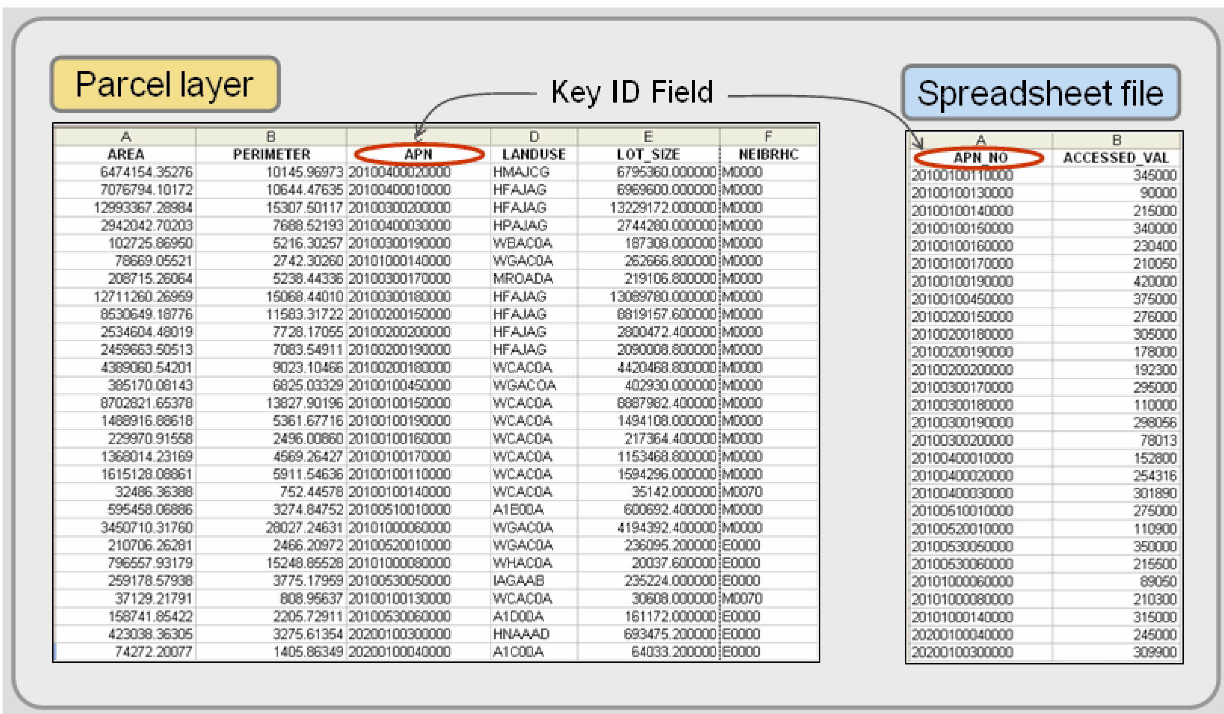
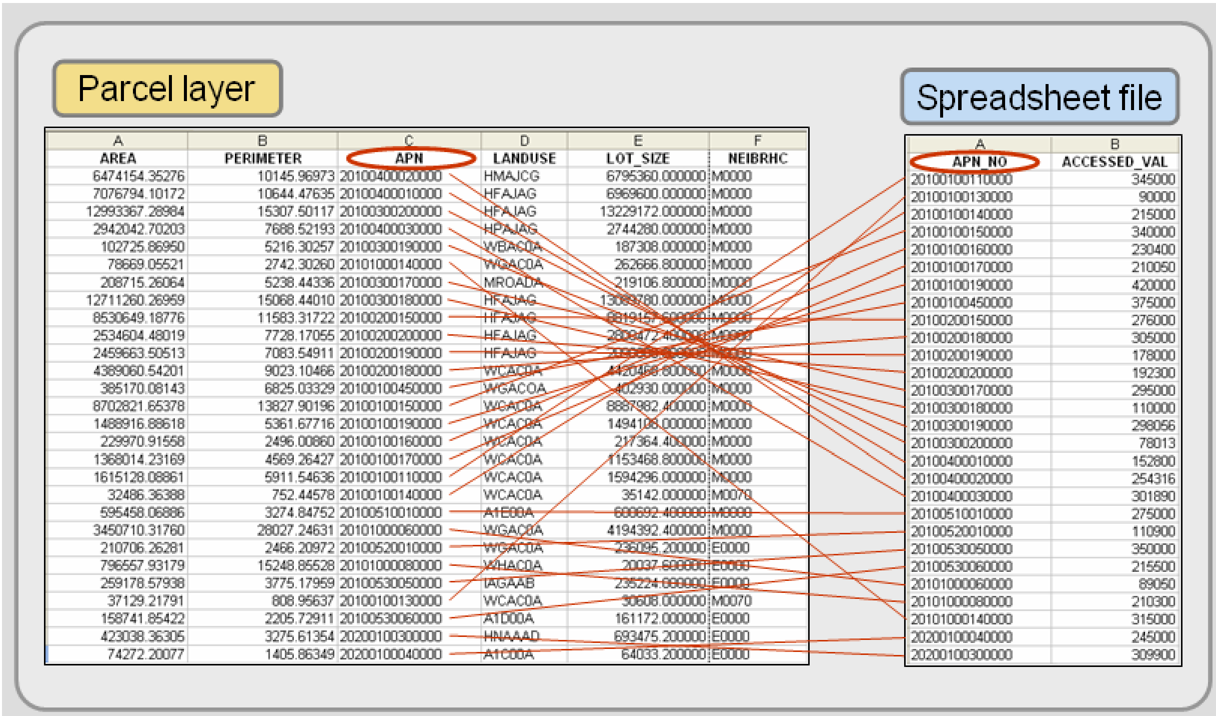


Figure 4.6: Joining two attribute files together requires that the two files each have a common key identifier.

Once the spreadsheet file is loaded, you begin the joining process by specifying the two files (the layer's table and the spreadsheet file) and the two field names that the join will be made on. APN and APN_NUM are the key identifiers of these two files (see Figure 4.7), and even though the field names are not identical, the GIS will be able to join these two files together provided that the values under the two field names match.



AREA	PERIMETER	APN	LANDUSE	LOT_SIZE	NEIBRHC
6474154.35276	10145.96973	20100400020000	HMAJCG	6795360.000000	M0000
7076794.10172	10644.47635	20100400010000	HFAJAG	6969600.000000	M0000
12993367.28984	15307.50117	20100300200000	HFAJAG	13229172.000000	M0000
2942042.70203	7688.52193	20100400030000	HPAJAG	2744280.000000	M0000
102725.86950	5216.30257	20100300190000	WBAC0A	187308.000000	M0000
78669.05521	2742.30260	20101000140000	WGAC0A	262666.800000	M0000
208715.26064	5238.44336	20100300170000	MROADA	219106.800000	M0000
12711260.26959	15068.44010	20100300180000	HFAJAG	13089780.000000	M0000
8530649.18776	11583.31722	20100200150000	HFAJAG	6819157.600000	M0000
2534604.48019	7728.17055	20100200200000	HFAJAG	2808472.400000	M0000
2459663.50513	7083.54911	20100200190000	HFAJAG	[illegible]	M0000
4389060.54201	9023.10466	20100200180000	WCAC0A	4420468.800000	M0000
385170.08143	6825.03329	20100100450000	WGACOA	402930.000000	M0000
8702821.65378	13827.90196	20100100150000	WCAC0A	8887982.400000	M0000
1488916.88618	5361.67716	20100100190000	WCAC0A	1494108.000000	M0000
229970.91558	2496.00860	20100100160000	WCAC0A	217364.400000	M0000
1368014.23169	4569.26427	20100100170000	WCAC0A	1153468.800000	M0000
1615128.08861	5911.54636	20100100110000	WCAC0A	1594296.000000	M0000
32486.36388	752.44578	20100100140000	WCAC0A	35142.000000	M0070
595458.06886	3274.84752	20100510010000	A1E00A	600692.400000	M0000
3450710.31760	28027.24631	20101000060000	WGAC0A	4194392.400000	M0000
210706.26281	2466.20972	20100520010000	WGAC0A	236095.200000	E0000
796557.93179	15248.85528	20101000080000	WHAC0A	20037.600000	E0000
259178.57938	3775.17959	20100530050000	IAGAAB	235224.000000	E0000
37129.21791	808.95637	20100100130000	WCAC0A	30608.000000	M0070
158741.85422	2205.72911	20100530060000	A1D00A	161172.000000	E0000
423038.36305	3275.61354	20200100300000	HNAAAD	693475.200000	E0000
74272.20077	1405.86349	20200100040000	A1C00A	64033.200000	E0000

APN_NO	ACCESSED_VAL
20100100110000	345000
20100100130000	90000
20100100140000	215000
20100100150000	340000
20100100160000	230400
20100100170000	210050
20100100190000	420000
20100100450000	375000
20100200150000	276000
20100200180000	305000
20100200190000	178000
20100200200000	192300
20100300170000	295000
20100300180000	110000
20100300190000	298056
20100300200000	78013
20100400010000	152800
20100400020000	254316
20100400030000	301890
20100510010000	275000
20100520010000	110900
20100530050000	350000
20100530060000	215500
20101000060000	89050
20101000080000	210300
20101000140000	315000
20200100040000	245000
20200100300000	309900

Figure 4.7: Matching key identifiers.

If the match is successful, your two files will be joined together into a single file (see Figure 4.8).

“Joined” parcel layer

A	B	C	D	E	F	G
AREA	PERIMETER	APN	LANDUSE	LOT_SIZE	NEIBRHC	ACCESSED_VAL
6474154.35276	10145.96973	20100400020000	HMAJCG	6795360.000000	M0000	254316
7076794.10172	10644.47635	20100400010000	HFAJAG	6969600.000000	M0000	152800
12993367.28984	15307.50117	20100300200000	HFAJAG	13229172.000000	M0000	78013
2942042.70203	7688.52193	20100400030000	HPAJAG	2744280.000000	M0000	301890
102725.86950	5216.30257	20100300190000	WBAC0A	187308.000000	M0000	298056
78669.05521	2742.30260	20101000140000	WGAC0A	262666.800000	M0000	315000
208715.26064	5238.44336	20100300170000	MROADA	219106.800000	M0000	295000
12711260.26959	15068.44010	20100300180000	HFAJAG	13089780.000000	M0000	110000
8530649.18776	11583.31722	20100200150000	HFAJAG	8819157.600000	M0000	276000
2534604.48019	7728.17055	20100200200000	HFAJAG	2800472.400000	M0000	192300
2459663.50513	7083.54911	20100200190000	HFAJAG	2090008.800000	M0000	178000
4389060.54201	9023.10466	20100200180000	WCAC0A	4420468.800000	M0000	305000
385170.08143	6825.03329	20100100450000	WGACOA	402930.000000	M0000	375000
8702821.65378	13827.90196	20100100150000	WCAC0A	8887982.400000	M0000	340000
1488916.88618	5361.67716	20100100190000	WCAC0A	1494108.000000	M0000	420000
229970.91558	2496.00860	20100100160000	WCAC0A	217364.400000	M0000	230400
1368014.23169	4569.26427	20100100170000	WCAC0A	1153468.800000	M0000	210050
1615128.08861	5911.54636	20100100110000	WCAC0A	1594296.000000	M0000	345000
32486.36388	752.44578	20100100140000	WCAC0A	35142.000000	M0070	215000
595458.06886	3274.84752	20100510010000	A1E00A	600692.400000	M0000	275000
3450710.31760	28027.24631	20101000060000	WGAC0A	4194392.400000	M0000	89050
210706.26281	2466.20972	20100520010000	WGAC0A	236095.200000	E0000	110900
796557.93179	15248.85528	20101000080000	WHAC0A	20037.600000	E0000	210300
259178.57938	3775.17959	20100530050000	IAGAAB	235224.000000	E0000	350000
37129.21791	808.95637	20100100130000	WCAC0A	30608.000000	M0070	90000
158741.85422	2205.72911	20100530060000	A1D00A	161172.000000	E0000	215500
423038.36305	3275.61354	20200100300000	HNAAAD	693475.200000	E0000	309900
74272.20077	1405.86349	20200100040000	A1C00A	64033.200000	E0000	245000

Figure 4.8: A joined file with accessed values a one of the attributes.

Perhaps the most time consuming tasks are the first and fourth steps. Loading an external data file should be easy —and frequently it is—but sometimes the imported data file may be misformatted or unreadable. If it is, return to the host program (your spreadsheet or DBMS programs) and save it in a different format. The probability of your GIS program being able to read the external data file usually improves as you go from more sophisticated file formats (like Excel and Access) to dBase to ASCII (basic formats). Many data files are coded in ASCII because of its almost universal compatibility with computers and software programs, but it does have its complications—it comes in several forms. Below are four of the most used variants of ASCII based on what delimits the file's fields.

Whitespace delimited ASCII files differentiates fields by the use of one or more spaces. Since spaces separate fields, fields that have no value must be represented by a non-blank code and character attributes cannot contain spaces between words (underscores can be used to separate words). You can open ASCII files in any word processer or text editor. A whitespace-delimited ASCII data file with five records might look something like the following:

M1 Betsy_Burns Yes 38.5 0.85
P1 Dan_Arreola No 45.7 0.99
M2 Frank_Aldrich Yes 32.8 0.55
P2 Fritz_Steiner No - -
P3 Ruth_Yabes No 37.72 -

Spacequote delimited ASCII is a variant of whitespace delimitation, but the attributes containing multiple words are enclosed in double quotes, and consequently, they can contain embedded spaces between words. The spacequote delimited ASCII file may look like the following in a text editor:

M1 "Betsy Burns" Yes 38.5 0.85
P1"Dan Arreola" No 45.7 0.99
M2 "Frank Aldrich" Yes 32.8 0.55
P2 "Fritz Steiner" No - -
P3 "Ruth Yabes" No 37.72 -

Tab delimited files separate fields by the use of a single tab. Two tabs in a row signify a blank field. Values within an attribute field cannot contain embedded tabs. A tab delimited ASCII file would look like the following in a text editor.

M1	Betsy Burns	Yes	38.5	0.85
P1	Dan Arreola	No	45.7	0.99
M2	Frank Aldrich	Yes	32.8	0.55
P2	Fritz Steiner	No		
P3	Ruth Yabes	No	37.72	

Comma delimited, also known as comma-quote delimited and CSV, separate fields by commas. Character fields may be enclosed in double quotes, and need to be if they contain an embedded comma. Two commas in a row signify that the field is blank. Usually whitespace is not allowed before or after fields (although this may be tolerated in the CSV form). The comma-delimited ASCII file might look like the following in a text editor:

M1,"Betsy Burns",Yes,38.5,0.85
P1,"Dan Arreola",No,45.7,0.99
M2,"Frank Aldrich",Yes,32.8,0.55

P2,"Fritz Steiner",No,,
P3,"Ruth Yabes",No,37.72,

Sorting records

Sorting temporarily rearranges your data file records, so you can view, select, update, or print them in the new sorted sequence. Although the specifics vary by program, you generally choose the field (or fields) you want to sort by. The first sort field arranges, usually in ascending or descending order, the records based on the field's contents. For example, a class roster might be sorted alphabetically by last name. Some systems allow you to choose a second sort field (or more), which arranges records (in ascending or descending order) when two or more records have the same first field value. In the example above, if your alphabetical list has four students with the last name Smith, those four records could be rearranged in alphabetical order based on their first name.

Record selection/Attribute Query (Boolean Selection)

Selecting specific records is one of the most common database functions. Often called attribute query, it consists of highlighting a subset of the records based on a specific criteria. In other words, you create an expression—a formula—that queries all the records in the data file and the GIS highlights—both in the data file and on the map display—only those features that fit the criteria.

Most GIS programs use a Standard Query Language (SQL) interface to conduct attribute queries. If one is using an external relational DBMS program (like Access or Oracle), SQL makes the call to the external database and isolates only the necessary records that you will use. SQL uses set algebra, Boolean algebra, and arithmetic operators (=, -, *, /) for attribute queries. Set Algebra includes the use of less than (<), greater than (>), equal to (=), and not equal to (<>) operations. You can create an expression like that found below (see Figure 4.9) to isolate only those records that fit your criteria. You can extend or constrain the selected features by using Boolean algebra, which uses the conditions OR (extend), AND (constrain), and NOT to further select or isolate records. Each record is queried and added to the set if it meets the criteria.

AREA > 2000000

APN	AREA	PERIMETER	LANDUSE	ACCESSED_VAL	CITY
20100400020000	6474154.35276	10145.96973	HMAJCG	254316	SACRAMENTO
20100400010000	7076794.10172	10644.47635	HFAJAG	152800	SACRAMENTO
20100300200000	12993367.28984	15307.50117	HFAJAG	78013	ELVERTA
20100400030000	2942042.70203	7688.52193	HPAJAG	301890	SACRAMENTO
20100300190000	102725.86950	5216.30257	WBACOA	298056	SACRAMENTO
20101000140000	78669.05521	2742.30260	WGACOA	315000	ELVERTA
20100300170000	208715.26064	5238.44336	MROADA	295000	SACRAMENTO
20100300180000	12711260.26959	15068.44010	HFAJAG	110000	ELVERTA
20100200150000	8530649.18776	11583.31722	HFAJAG	276000	SACRAMENTO
20100200200000	2534604.48019	7728.17055	HFAJAG	192300	ELVERTA

AREA > 2000000 AND LANDUSE = HFAJAG

APN	AREA	PERIMETER	LANDUSE	ACCESSED_VAL	CITY
20100400020000	6474154.35276	10145.96973	HMAJCG	254316	SACRAMENTO
20100400010000	7076794.10172	10644.47635	HFAJAG	152800	SACRAMENTO
20100300200000	12993367.28984	15307.50117	HFAJAG	78013	ELVERTA
20100400030000	2942042.70203	7688.52193	HPAJAG	301890	SACRAMENTO
20100300190000	102725.86950	5216.30257	WBACOA	298056	SACRAMENTO
20101000140000	78669.05521	2742.30260	WGACOA	315000	ELVERTA
20100300170000	208715.26064	5238.44336	MROADA	295000	SACRAMENTO
20100300180000	12711260.26959	15068.44010	HFAJAG	110000	ELVERTA
20100200150000	8530649.18776	11583.31722	HFAJAG	276000	SACRAMENTO
20100200200000	2534604.48019	7728.17055	HFAJAG	192300	ELVERTA

AREA > 2000000 AND LANDUSE = HFAJAG NOT CITY = ELVERTA

APN	AREA	PERIMETER	LANDUSE	ACCESSED_VAL	CITY
20100400020000	6474154.35276	10145.96973	HMAJCG	254316	SACRAMENTO
20100400010000	7076794.10172	10644.47635	HFAJAG	152800	SACRAMENTO
20100300200000	12993367.28984	15307.50117	HFAJAG	78013	ELVERTA
20100400030000	2942042.70203	7688.52193	HPAJAG	301890	SACRAMENTO
20100300190000	102725.86950	5216.30257	WBACOA	298056	SACRAMENTO
20101000140000	78669.05521	2742.30260	WGACOA	315000	ELVERTA
20100300170000	208715.26064	5238.44336	MROADA	295000	SACRAMENTO
20100300180000	12711260.26959	15068.44010	HFAJAG	110000	ELVERTA
20100200150000	8530649.18776	11583.31722	HFAJAG	276000	SACRAMENTO
20100200200000	2534604.48019	7728.17055	HFAJAG	192300	ELVERTA

Figure 4.9: Select records based on their attributes by using SQL expressions.

Once the records are selected, you can work with just those records. This is helpful for viewing, sorting, editing, calculating fields, generating statistics, using the selected features to select features in another GIS layer, creating a new layer with only the selected features, and isolating specific records to perform analysis functions on (like buffering selected features).

In addition, spatial queries, selecting features based on their geographic location (see Chapter 5), can be combined with attribute queries for more sophisticated queries. There is more on attribute and spatial queries in Chapter 5.

Calculate Attributes

Within an open data file, you can create new attributes by using values in existing fields, mathematical expressions, and text functions (see Figure 4.10). Mathematical operations allow you to add, subtract, multiply, and divide existing fields or values to create new, derived attributes. Text functions allow you to populate fields with data, copy values from one field to another, concatenate fields (and or values), truncate attributes, and convert text to different formats. Before calculating the new field, however, you need to create a new attribute field, which includes defining its field name and its data properties). Calculations can be performed on a single record, several selected records, or on every record in the data file. The calculate function can also be used to copy data from one field to another.

POP00_SQMI = POP2000 / AREA

STATE_NAME	AREA	POP2000	POP00_SQMI
Alabama	51715.786	4447100	0
Arizona	113712.679	5130632	0
Arkansas	52913.232	2673400	0
California	157776.31	33871648	0
Colorado	104101.231	4301261	0
Connecticut	4976.566	3405565	0
Delaware	2054.586	783600	0
District of Columbia	66.063	572059	0
Florida	55814.731	15982378	0
Georgia	58629.222	8186453	0
Idaho	83343.643	1293953	0
Illinois	56299.387	12419293	0
Indiana	36400.304	6080485	0
Iowa	56257.965	2926324	0
Kansas	82196.955	2688418	0
Kentucky	40319.791	4041769	0
Louisiana	45835.844	4468976	0
Maine	32161.925	1274923	0
Maryland	9739.872	5296486	0
Massachusetts	8172.561	6349097	0
Michigan	57899.398	9938444	0
Minnesota	84520.49	4919479	0
Mississippi	47618.965	2844658	0
Missouri	69832.746	5595211	0
Montana	147244.653	902195	0
Nebraska	77330.258	1711263	0
Nevada	110669.975	1998257	0
New Hampshire	9259.527	1235786	0
New Jersey	7507.502	8414350	0
New Mexico	121757.343	1819046	0
New York	48561.751	18976457	0
North Carolina	49048.024	8049313	0
North Dakota	70812.056	642200	0
Ohio	41193.957	11353140	0

STATE_NAME	AREA	POP2000	POP00_SQMI
Alabama	51715.786	4447100	86
Arizona	113712.679	5130632	45
Arkansas	52913.232	2673400	51
California	157776.31	33871648	215
Colorado	104101.231	4301261	41
Connecticut	4976.566	3405565	684
Delaware	2054.586	783600	381
District of Columbia	66.063	572059	8659
Florida	55814.731	15982378	286
Georgia	58629.222	8186453	140
Idaho	83343.643	1293953	16
Illinois	56299.387	12419293	221
Indiana	36400.304	6080485	167
Iowa	56257.965	2926324	52
Kansas	82196.955	2688418	33
Kentucky	40319.791	4041769	100
Louisiana	45835.844	4468976	97
Maine	32161.925	1274923	40
Maryland	9739.872	5296486	544
Massachusetts	8172.561	6349097	777
Michigan	57899.398	9938444	172
Minnesota	84520.49	4919479	58
Mississippi	47618.965	2844658	60
Missouri	69832.746	5595211	80
Montana	147244.653	902195	6
Nebraska	77330.258	1711263	22
Nevada	110669.975	1998257	18
New Hampshire	9259.527	1235786	133
New Jersey	7507.502	8414350	1121
New Mexico	121757.343	1819046	15
New York	48561.751	18976457	391
North Carolina	49048.024	8049313	164
North Dakota	70812.056	642200	9
Ohio	41193.957	11353140	276

Figure 4.10: Calculating fields. In this example, population density is calculated by dividing population by area. First, the field must be added. Then, you calculate the results directly into the new field.

Geocoding

There is a way to create geographic data directly from attribute data. The process, called geocoding, assigns geographic locations to features directly from attribute fields that contain locational information within a data file. This is a popular way to create GIS feature layers; you create or obtain a spreadsheet or data file with location information, open the attribute table in your GIS, and direct the system toward the appropriate attribute fields. There are two types of geocoding: coordinate locations and address matching.

Spatial features can be created from data files containing fields with x,y coordinate values. The coordinates need to be separated into two separate fields: one for the x coordinate and one for the y coordinate. The process is straightforward; you direct the GIS to the data file's appropriate x,y fields, and it creates a spatial layer of point features from the coordinates. One possible complication is that the data file's coordinates are different than the coordinate system you are using. This requires that you open the file in a temporary workspace registered to the data file's coordinate system and then convert the new spatial layer to the desired coordinate system.

Address matching is another type of geocoding. It matches records in two data files—one containing a list of addresses and the other having street network attributes—to create a new layer (see Figure 4.11). In other words, it creates a layer of point features alongside street segments when addresses in the two data files match. It essentially looks up the address in the first record of the external data file and tries to find a match along the street network layer. If multiple possibilities exist, the routine will present them for user input. After the first record is matched or not, it moves to the second record and tries again. The resultant file is assigned the street network's coordinate system.

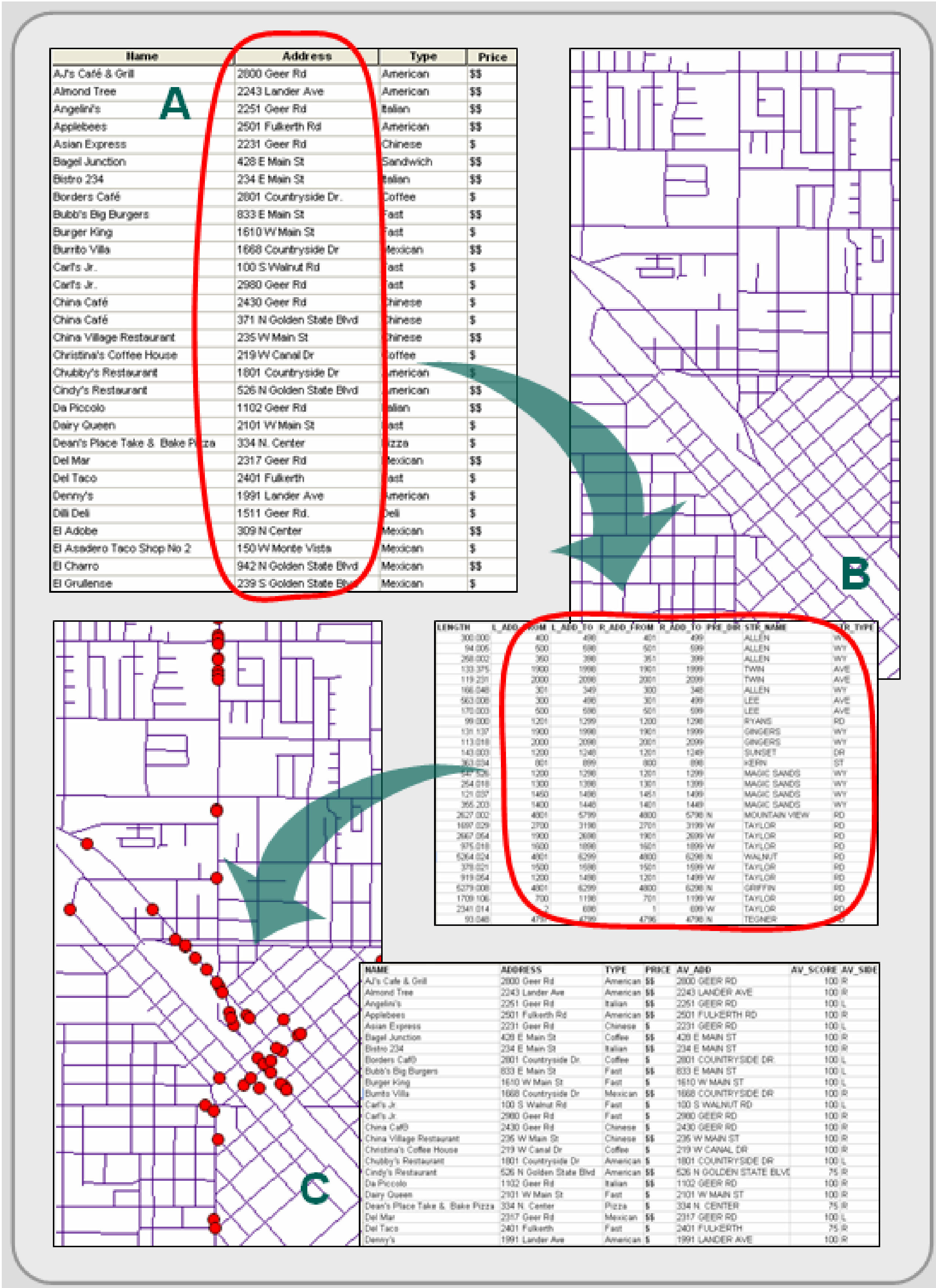

A
Name | Address | Type | Price
AJ's Café & Grill | 2800 Geer Rd | American | $$
Almond Tree | 2243 Lander Ave | American | $$
Angelini's | 2251 Geer Rd | Italian | $$
Applebees | 2501 Fulkerth Rd | American | $$
Asian Express | 2231 Geer Rd | Chinese | $
Bagel Junction | 428 E Main St | Sandwich | $$
Bistro 234 | 234 E Main St | Italian | $$
Borders Café | 2801 Countryside Dr. | Coffee | $
Bubb's Big Burgers | 833 E Main St | Fast | $$
Burger King | 1610 W Main St | Fast | $
Burrito Villa | 1668 Countryside Dr | Mexican | $$
Carl's Jr. | 100 S Walnut Rd | Fast | $
Carl's Jr. | 2980 Geer Rd | Fast | $
China Café | 2430 Geer Rd | Chinese | $
China Café | 371 N Golden State Blvd | Chinese | $
China Village Restaurant | 235 W Main St | Chinese | $$
Christina's Coffee House | 219 W Canal Dr | Coffee | $
Chubby's Restaurant | 1801 Countryside Dr | American | $
Cindy's Restaurant | 526 N Golden State Blvd | American | $$
Da Piccolo | 1102 Geer Rd | Italian | $$
Dairy Queen | 2101 W Main St | Fast | $
Dean's Place Take & Bake Pizza | 334 N. Center | Pizza | $
Del Mar | 2317 Geer Rd | Mexican | $$
Del Taco | 2401 Fulkerth | Fast | $
Denny's | 1991 Lander Ave | American | $
Dilli Deli | 1511 Geer Rd. | Deli | $
El Adobe | 309 N Center | Mexican | $$
El Asadero Taco Shop No 2 | 150 W Monte Vista | Mexican | $
El Charro | 942 N Golden State Blvd | Mexican | $$
El Grullense | 239 S Golden State Blvd | Mexican | $
B
C
NAME | ADDRESS | TYPE | PRICE | AV_ADD | AV_SCORE | AV_SIDE
AJ's Cafe & Grill | 2800 Geer Rd | American | $$ | 2800 GEER RD | 100 | R
Almond Tree | 2243 Lander Ave | American | $$ | 2243 LANDER AVE | 100 | R
Angelini's | 2251 Geer Rd | Italian | $$ | 2251 GEER RD | 100 | L
Applebees | 2501 Fulkerth Rd | American | $$ | 2501 FULKERTH RD | 100 | R
Asian Express | 2231 Geer Rd | Chinese | $ | 2231 GEER RD | 100 | L
Bagel Junction | 428 E Main St | Coffee | $$ | 428 E MAIN ST | 100 | R
Bistro 234 | 234 E Main St | Italian | $$ | 234 E MAIN ST | 100 | R
Borders Caf© | 2801 Countryside Dr | Coffee | $ | 2801 COUNTRYSIDE DR | 100 | L
Bubb's Big Burgers | 833 E Main St | Fast | $$ | 833 E MAIN ST | 100 | L
Burger King | 1610 W Main St | Fast | $ | 1610 W MAIN ST | 100 | L
Burrito Villa | 1668 Countryside Dr | Mexican | $$ | 1668 COUNTRYSIDE DR | 100 | R
Carl's Jr | 100 S Walnut Rd | Fast | $ | 100 S WALNUT RD | 100 | L
Carl's Jr | 2980 Geer Rd | Fast | $ | 2980 GEER RD | 100 | R
China Caf© | 2430 Geer Rd | Chinese | $ | 2430 GEER RD | 100 | R
China Village Restaurant | 235 W Main St | Chinese | $$ | 235 W MAIN ST | 100 | R
Christina's Coffee House | 219 W Canal Dr | Coffee | $ | 219 W CANAL DR | 100 | R
Chubby's Restaurant | 1801 Countryside Dr | American | $ | 1801 COUNTRYSIDE DR | 100 | L
Cindy's Restaurant | 526 N Golden State Blvd | American | $$ | 526 N GOLDEN STATE BLVD | 75 | R
Da Piccolo | 1102 Geer Rd | Italian | $$ | 1102 GEER RD | 100 | R
Dairy Queen | 2101 W Main St | Fast | $ | 2101 W MAIN ST | 100 | R
Dean's Place Take & Bake Pizza | 334 N. Center | Pizza | $ | 334 N. CENTER | 75 | R
Del Mar | 2317 Geer Rd | Mexican | $$ | 2317 GEER RD | 100 | L
Del Taco | 2401 Fulkerth | Fast | $ | 2401 FULKERTH | 75 | R
Denny's | 1991 Lander Ave | American | $ | 1991 LANDER AVE | 100 | R

Figure 4.11: Address matching. The addresses in an external data file (A) are compared to a street network's (B) attribute fields, and if a match is made, the record in the external data file gets a point on the map (C).

Both the street network layer and the external data file need address data (street name, street type, and an address range for start and end of each line segment), and perhaps even more information like city, state, and Zip code attributes to make your address information unique (multiple cities will likely contain streets with the same name). The process works well if the addresses in both the external data file and the street network layer are accurate and complete, but address matching is a time consuming process.

Data Export

Exporting your GIS layers, including their geographic and attribute data files, are covered in Chapter 6. Most GIS programs can export your layer's attribute file in a number of formats including dBase and ASCII. The exported files can then be used in database, statistic, and spreadsheet programs for additional analysis.

ATTRIBUTE VERIFICATION

This section looks at verifying the accuracy of attributes. The verification process looks for both missing attributes and incorrect attribute values. Unlike geographic verification, there are no attribute verification procedures built within the software to verify their accuracy.

Instead, the layer's data file can be displayed and sorted by each attribute in ascending order to identify missing attributes (see Figure 4.12). Map features missing a value for a particular field are revealed at the top of the table. Selecting those features from the data file and highlighting them on the screen can be a handy way to reference those features that have missing attributes. The selected features can then be investigated and updated. You can also sort the attributes alphabetically and glance down the field looking for spelling mistakes.

EMPLOYER_N	EMPLOYER_A	CITY	ZIP
	2100 STANDIFORD AVE SUITE E14	MODESTO	95350
	4801 E. WHITMORE	CERES	95307
	2900 STANDIFORD	MODESTO	
	TONY ROMA'S	MODESTO	95350
	3[illegible]1 STANDIFORD AVENUE #181	MODESTO	
	14[illegible]1 J STREET	MODESTO	
	17[illegible]7 SYVAN AVE.	MODESTO	95356
	19[illegible]0 OAKDALE RD	MODESTO	95355
	3[illegible]01 YOSEMITE BLVD.	MODESTO	95357
	1351 E HATCH ROAD	MODESTO	95351
	804 14TH STREET	MODESTO	95354
	426 14TH ST.	MODESTO	95355
	120 S. EMERALD	MODESTO	95351
	3120 MC HENRY AVE	MODESTO	95350
CONTINENTAL PET TECH INC BARRETT BUSINESS SERVICES, INC.			
"WORKING WITH PETITION" PER HEATHER			
A&L PRODUCTS, INC.	1900 KINSER AVE.	CERES	
A.R.A.P. OF AMERICA "ZAPCO"	413 S. RIVERSIDE DR	MODESTO	95354
ACE LATHING, INC.	1320 ROCKEFELLER	CERES	95307
ADVANCED ROOF SAVERS TECH., INC	4648 SALIDA BL	SALIDA	95356
ALL VALLEY PACKING, INC.	3006 YOSEMITE BLVD	MODESTO	95354
ALPHA POULTRY AND LIVESTOCK EQUIP., INC.	900 W. GLENWOOD	TURLOCK	95380
ALUMA PANEL, INC	2000 MORGAN RD	MODESTO	95358
AM/PM PRESTIGE STATIONS, INC.	1700 HATCH RD	MODESTO	95351
AMERICAN CUSTOM FOAN, INC.	9TH ST.	MODESTO	95350
AMERICAN MEALS, INC.	1991 LANDER AVE	TURLOCK	95380
AMERICAN MEDICAL RESPONSE, INC.	801 10TH ST.	MODESTO	95354
AMERINE SYSTEMS, INC.	10866 CLEVELAND AVE	OAKDALE	95361
ANGEL R MARTINEZ, FARM LABOR	956 KANSAS AVE	MODESTO	95351
ART TO WEAR, INC	1420 GRANITE LN	MODESTO	95351
ASSOC. CLEANING SVC, INC	1312 KANSAS AVE	MODESTO	95351
ATTARCO MOTOR, INC.	1124 KANSAS AVE	MODESTO	
AUTO PARTS DISTRIBUTORS, INC	725 11TH ST	MODESTO	95354
AUTO PARTS DISTRIBUTORS, INC	725 11TH ST	MODESTO	95354

Figure 4.12: Sorting in ascending order can reveal missing data.

More difficult to detect are incorrect attribute values. They require familiarity with the original source maps and an understanding of spatial patterns. For example, if you were working with income data, you should select low income values and display them on a map. Does their spatial location make sense? Do the same with high income values. Nominal data sets can be displayed the same way. For example, select different land use classes, and see if they make geographic sense. Display all heavy industrial sites and look at their locations. If heavy industry appears in the middle of wealthy residential areas or they are not located along highways, railroads, or rivers (which they need for transportation purposes) than these values may be inaccurate. More information may be needed; try looking at web-based aerial photographs or field check odd values.