Chapter 8. Queries

Objectives

- Understanding queries and how they are used
- Selecting features based on attributes using SQL and Boolean operators
- Selecting features based on their spatial location with respect to other features
- Applying selection options, including the selectable layers and the selection method

Mastering the Concepts

GIS Concepts

About queries

Queries are a fundamental tool in GIS, used in nearly every analysis. They serve a critical function in data exploration or in picking one's way through a data set looking for patterns. They can also serve as the preparatory step of extracting certain features prior to performing another function. Some common applications involving queries include the following:

- ***Selecting features of interest.*** Queries can be used to search a table and to find which features meet certain criteria. How many houses currently for sale fall into my price range? Combined with the statistics function, this tool becomes powerful. What is the average cost of three-bedroom homes in this town?
- ***Exploring spatial patterns.*** Creating a map from selected features and examining their spatial distribution can be illuminating. Where do the wells with high values of contaminants occur? Do they show a pattern relative to known point source pollution emissions? Are they widely spaced or clustered together?
- ***Isolating features for more analysis.*** Select the largest earthquakes and find out how many of them lie close to volcanoes. Choose the states that have lost population and subject them to a multivariate statistical analysis of demographic data to find out why.
- ***Exploring spatial relationships.*** Which parcels lie in the floodplain? Which cities are close to a volcano? Where do roads cross unstable shale units?
- ***Creating raster queries.*** Although we have only discussed vector queries, cells of a raster can be queried also. Which cells have a slope greater than 10 degrees? Which cells show an increase in vegetative greenness as measured by satellite images taken in 1998 and 1999? Where did land use change between 1970 and 2000?

A query extracts features or records from a feature class or from a table and isolates them for further use, such as printing them, calculating statistics about them, editing them, graphing them, creating new files from them, or doing more queries on them. In the simplest kind of query, you look at a map or a table and use the mouse to select the desired record(s). An **attribute query** uses records in the attribute table to test a condition, such as finding all cities with population greater than 1 million people. Attribute queries are sometimes called aspatial queries because they do not require any information about location. A **spatial query** uses information about how features from two different layers are located with respect to each other. A spatial query might choose cities within a certain county, rivers that are within a state, or hospitals within 20 miles of

an airport. Attribute queries can be performed on either attribute tables or standalone tables. A spatial query requires a spatial data layer. Spatial and attribute queries may be combined, such as selecting cities with more than 50,000 people who are within 50 miles of an airport. The spatial and attribute queries must be performed separately but can be done in any order.

Attribute queries

In this method, the user specifies a certain condition based on fields in the attribute table and selects the records that meet the criteria, such as choosing all states with a population greater than 5 million. We already introduced selecting by attributes from tables in Chapter 4. In this section, we extend selection to features and discuss more advanced selection options.

Attribute queries test conditions between two or more attributes using **logical operators**, which include the familiar <, >, and = operators. The condition is written as an **expression** in a format called **Structured Query Language** (SQL), which is used by most major databases to perform selections. SQL expressions can be complex and sophisticated, or fairly simple.

SELECT *FROM cities WHERE [POP1990] >= 500000

SELECT *FROM counties WHERE [BEEFCOW_92] < [BEEFCOW_87]

The SELECT *FROM and WHERE key words must be present as dictated by the syntax rules of SQL. The lowercase words (e.g., *cities*) give the name of the table from which to select, and the terms in double quotes or brackets are the field names. This syntax would be used in any SQL-compliant database, so SQL is a transportable language. However, minor differences between database SQL varieties can be found, such as whether field names are enclosed in brackets or quotes, or nothing.

Care must be taken when performing queries on numeric values depending on the type of field used to store the data. The = operator can only be used on integers, because a floating-point value is never exactly equal to an integer or specific decimal value. The expression NUM = 3 will not work on a floating-point field, and the expression NUM = 3.6 would never work at all. Queries on floating-point fields must use only the < or > operators.

Boolean operators

When a query includes more than one condition, the additional operators AND, OR, XOR, and NOT are needed. Consider the following queries with two conditions:

SELECT *FROM accounts WHERE Cust = 'COM' AND Balance > 500

SELECT *FROM accounts WHERE Cust = 'COM' OR Cust = 'GOV'

SELECT *FROM accounts WHERE Balance > 500 AND Balance < 1500

The first expression selects records from a customer accounts table that are held by commercial customers and have a balance over $500. The second expression finds the accounts held by commercial and governmental customers. The third expression finds accounts with balances between $500 and $1500.

TIP: SQL requires that a field name appear in every condition of the expression, even if it is the same field. SQL will not correctly evaluate the expression Cust = 'COM' AND 'GOV'.

AND, OR, XOR, and NOT are called **Boolean operators**, and they evaluate two input conditions, which may be either true or false, and return a set of records meeting those conditions. Figure 8.1 represents these operators graphically using **Venn diagrams**. The large ellipse labeled T represents all the records in the table. Circle A represents the subset of records meeting condition A, circle B represents the subset meeting condition B, and the blue region indicates the records selected by the different Boolean operators.

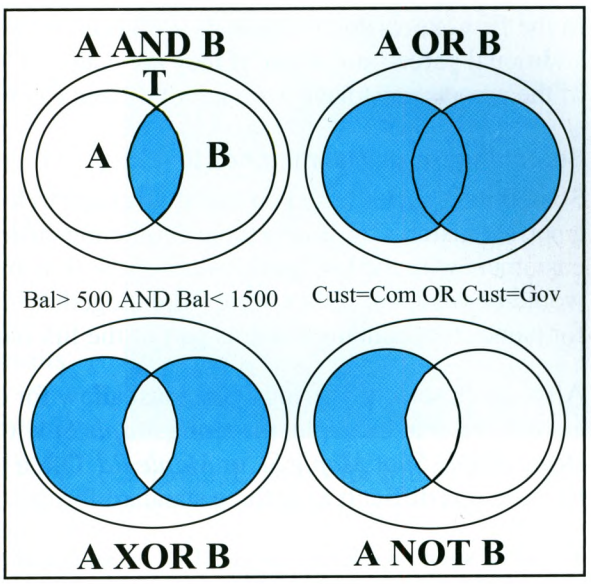


Fig. 8.1. Boolean operators

Examine the first expression in the preceding examples of Boolean operators. The first condition, Cust = 'COM', constitutes condition A; condition B would be Balance > 500. A AND B finds the accounts for which both input conditions are true (commercial customers and balance over $500). If either condition is false, the record is not selected.

The expression A OR B selects the records for which either condition is true, finding all the commercial customers and all the accounts with balances over $500 (commercial or not). A NOT B finds all the commercial accounts but excludes the ones with balances over $500, instead finding the commercial customers with balances under $500. The "exclusive or" operator XOR is not available for attribute queries in ArcGIS but is used in raster analysis. XOR finds instances when one or the other condition is true, but not both.

The AND and OR operators are sometimes confused. In English we would say, "select the commercial and the governmental accounts" if we want both, and it seems natural to use the expression Cust = 'COM' AND Cust = 'GOV'. However, this expression yields no records because a customer account cannot be both commercial and governmental at the same time. The OR operator should be used.

The AND operator is used to find records within a numeric range. Consider the third expression, Balance > 500 AND Balance < 1500. If a record with Balance = 1000 is being considered, it would test true for both conditions and be selected. Using OR is incorrect; it would return ALL of the values instead of those inside the range. A record with Balance = 300 would test true for the second condition and so be included. A record with Balance = 3000 would test true for the first condition and be included.

SQL can accept three or more Boolean conditions. Typically, parentheses must be used to enforce the correct order of evaluation. Compare these two expressions for selecting parcels:

(LUCODE = 42 AND VALUE > 50000) OR SIZE > 50

LUCODE = 42 AND (VALUE > 50000 OR SIZE > 50)

In the first expression, all parcels > 50 hectares are chosen regardless of their zoning or value, and additional parcels are added if they are code = 42 and high in value. In the second expression, all of the parcels must have a zoning of 42 and can be either high in value or large in size.

Searching for partial matches in text

Sometimes queries based on text fields need to look for partial matches to a condition rather than a perfect match. Consider a table containing customer names. A user might wish to find all customers with the last name *Smith*. If the first and last names were stored in separate fields it would be easy, but if they were stored together in a single field it would be necessary to search for names that contain *Smith* as part of the full name.

Also, when selecting strings, one must allow for extraneous spaces or inconsistencies in how data are entered. For example, the string *Maple St* with a space following *St* would not match the string *Maple St* or *Maple St.* or *Maple Rd*. Often databases have inconsistencies like this one, because it is difficult to enforce standard formats perfectly during data entry.

To solve such problems, most GIS systems (and databases) provide an operator, such as LIKE or CONTAINS, that allows searches for a particular substring within another string. This operator is helpful in searching for *Smith* within a field containing both first and last names or *Maple* within a field of street addresses. LIKE searches for the specified set of characters within the field and returns any record that contains those characters. When using LIKE, a wild card character (% for shapefiles; * for geodatabases) is used to stand for other letters that may appear. Consider selecting cities based on their NAME field. The two queries

NAME LIKE 'New %' *(shapefile)*

NAME LIKE 'New *' *(geodatabase)*

would select all cities beginning with the word *New*, such as New London and New Haven, but would not include Newcastle because there is no space in the latter. The expression

NAME LIKE '%Smith%'

would return the customers with the name *Smith*. Notice the second % sign used in the expression. One might expect it to be superfluous, and often it would be, but it does ensure that an extraneous space character at the end of the name would not prevent it from being selected.

Spatial queries

Spatial queries are a powerful tool unique to GIS because they select based on spatial relationships, such as finding wells within five miles of a river or finding parcels inside a floodplain. A spatial selection uses two layers and one spatial condition. The features of the layer being selected are compared spatially with the features of the second layer to see which ones meet the criteria, and features meeting the criteria are selected.

Panning and zooming techniques, already familiar to the reader, are actually a type of spatial query. Pans and zooms define a rectangle of interest based on *x*-*y* coordinates and ask that the software return all features that fall inside it.

Because feature classes vary in precision and geometric accuracy, it can often happen that two objects that coincide in the real world will not match when their GIS coordinates are compared. It is helpful to be able to specify a search radius when evaluating a spatial condition so that the features need not exactly match.

Consider the problem of selecting cities intersecting rivers. A city at a national scale is represented by a point, and a river by a generalized line. Even if the real-life city intersects a river, it would be only luck if the point occurred on the line feature representing the river (Fig. 8.2). Using the search radius provides some slack. If one assumes that a typical large city might be represented by a square 10 kilometers on a side, then setting the search radius to 5 kilometers might produce an acceptable result. Even so, one would probably encounter some false positive or false negative results.

Fig. 8.2. Features for Minneapolis–St. Paul and the Mississippi River

Users must always remember the difference between real-world entities and how they are represented by a GIS, and be prepared to mitigate problems that arise because of scale or accuracy issues. In this case, for example, the user might go ahead with the query, but then check each selected city using the online imagery to ensure that the correct relationship exists.

Attribute queries test relationships between attributes using logical operators, such as <, >, or =. Spatial queries test for relationships using **spatial operators**, including **containment**, **intersection**, and **proximity**. Consider two feature classes, A and B. The spatial operators shown in Figure 8.3 test for the relationship of each feature in A to each feature in B and return the features from A that meet the criteria with respect to B. A complete description of each operator can be found in the ArcGIS Help.

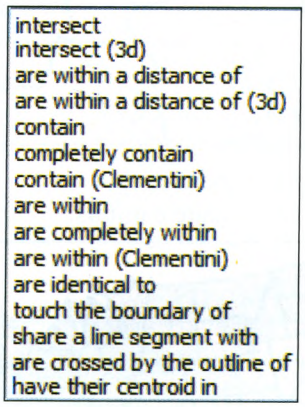


Fig. 8.3. Spatial operators

Containment tests whether one feature includes another. The test A *completely contains* B returns all features in A that fully surround the features in B. A *contains* B if B is inside A but they share a boundary. For example, both the yellow and blue counties in Figure 8.4 are contained by the state of Oregon, but only the blue counties are completely contained by the state. The *are within* and *are completely within* operators are the inverse of *contain* and *completely contain*. The *have their centroid in* operator tests whether the center of a feature in A lies inside a feature in B.

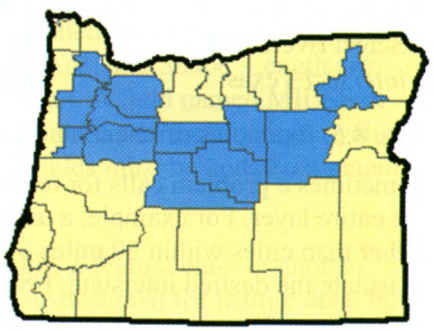
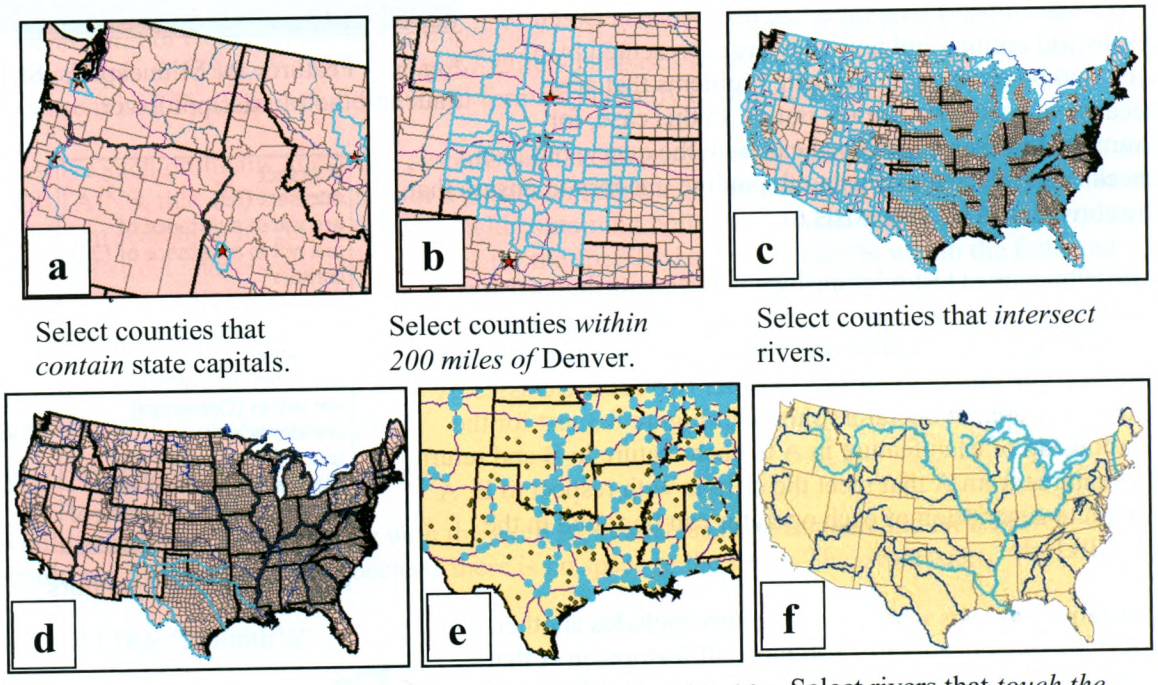


Fig. 8.4. The blue counties are completely contained by Oregon.

Intersection is the most generic operator and returns any feature in A that touches, crosses, or overlaps any part of a feature in B. The *are crossed by the boundary of* operator is a special case of intersection that returns features in A that are crossed only by the boundary of features in B, such as the parcels that are crossed by the city limits. The most stringent condition is that the feature from A equals the feature from B exactly (i.e., they have precisely the same geometry). This corresponds to the operator *are identical to*.

Proximity tests how close features in A are to features in B. The most generic test, *are within a distance of*, selects features in A that are within a certain distance of B, such as returning all parcels within two miles of a school. Adjacency is a special case of proximity where the distance goes to zero and the boundaries of the features actually touch each other. These conditions are covered by the *share a line segment with* and *touch the boundary of* operators (which might also be considered to be intersection operators because the features touch). Figure 8.5 shows several examples of applying various operators to two different feature classes.

TIP: A complete description of the different spatial operators with graphic examples can be found in the ArcGIS Help by searching in the Index for *Select By Location, described*.



Select counties that *contain* state capitals.

Select counties *within 200 miles of* Denver.

Select counties that *intersect* rivers.

Select rivers that *intersect* Texas.

Select cities that *are within 20 miles* of an interstate.

Select rivers that *touch the boundary of* states.

Fig. 8.5. Examples of selecting a layer based on its spatial relationship to another layer

Sometimes a problem calls for testing spatial relationships using a *subset* of features rather than the entire layer. For example, a user might want to select cities within 50 miles of Interstate 80, rather than cities within 50 miles of any interstate. This procedure begins with an attribute query to isolate the desired interstate, I-80, and then a spatial query to select the cities.

It must be kept in mind that spatial queries can only select or not select an entire feature. Consider Figure 8.5d, which shows the rivers that intersect Texas. Most of the rivers extend beyond the Texas border. If the within operator is used, then only rivers inside the state are selected (in this case, none of them). A different type of function is needed to extract only the portions of rivers that fall inside Texas.

Extraction functions

Extraction functions separate features of interest from a larger group. Some consider queries as one type of extraction. The functions **clip** and **erase** are able to extract both whole features and

portions of features based on a spatial boundary supplied by another feature class and save them in a new feature class. Clip and erase have the ability to truncate features when they cross a boundary, for example, a road crossing a state line. This ability differs from the Select By Location function, which must select (or not select) the entire feature.

A **clip** works like a cookie cutter to truncate the features of one file based on the outline of another. In Figure 8.6, the county roads layer (gray lines) has been clipped by the land use layer (beige) to produce a shapefile of the roads that fall inside the city planning boundaries (red). The features being extracted come from the input layer. The feature class providing the boundary is called the clip layer. The clip layer must always be a polygon feature class, but the input layer may be points, lines, or polygons. Only the outside boundary of the clip layer is used for clipping; internal boundaries, if present, have no effect on the output.

SKILL TIP: To temporarily clip features for display only, use the Data Frame tab of the Data Frame Properties menu (Layouts and Data Frames).

Erase works in the opposite sense of a clip, keeping features that fall outside the erase layer and eliminating those inside. Given the same input and clip layers in Figure 8.6, an erase would keep the roads outside the city (gray) and eliminate the roads inside the city (red). Erase is only available to users with an ArcGIS Advanced license.

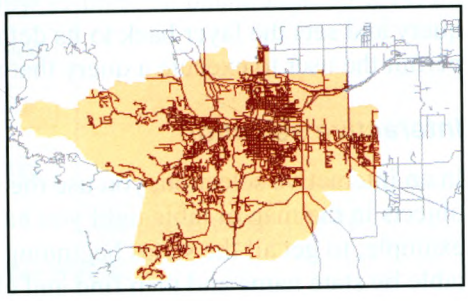


Fig. 8.6. A clip truncates the features of a layer by the boundary of another.

Attributes from both truncated and nontruncated features in the input layer are preserved intact during a clip or an erase operation. However, attributes from the clip or erase layer are ignored. The result of a clip or erase is a copy of the input layer that now occupies a smaller region of interest.

In clip and erase, any features crossing the boundary will be truncated, potentially changing areas, perimeters, or lengths of features. The geodatabase fields Shape_Area and Shape_Length are automatically updated, but user-defined area or length fields are not. A field named MILES or ACRES containing highway miles or tract acres, for example, would contain an incorrect value for any truncated feature. The lengths and areas of user-defined fields must be updated manually using the Calculate Geometry function.

Rasters may also be clipped to a smaller area of interest defined by a feature class or another raster. The tools used for raster extraction are different from those designed for feature data, but they work in much the same fashion.

About ArcGIS

ArcMap offers three ways to select features. **Interactive selection** uses a pointer to select features on the screen. Attribute queries are performed using **Select By Attributes**. Spatial queries are executed using **Select By Location**. After a query, the selected features are highlighted in the map and in the table.

Processing layers with selections

ArcGIS follows an important rule when processing features or records in any of its functions or tools: ***after a query is performed and a subset of features or records is selected, any subsequent operation on that layer or table honors the selected set.*** You have already seen this rule in action. If you were to select the New England states and calculate statistics, only the New England states would be included. If you selected parcels in a floodplain and then exported the features to a new feature class, only the parcels in the floodplain would be exported. Once a query has been executed on a layer, the layer behaves as if the unselected features do not exist.

By default, a layer has no query applied. In this state, all of the features are available for processing by a command. If a Statistics command were requested in this state, all of the features would be included in the calculations. Executing a query on a layer pushes it into a new state. The layer has a selection, and any function applied in ArcMap or in ArcToolbox will restrict the operation of the function to the selected features or records.

Sometimes a user may execute a query that has proper syntax but yields no matching records. For example, it is possible to query for cities above 10,000 feet in elevation, but no cities actually meet the condition, and no features would be returned. When this happens, ArcGIS removes the query and sets the layer back to its default state with all features available. Essentially, it does not permit the user to execute a query that returns no records or features.

Interactive selection

In an interactive selection, you use the mouse to click on one or more objects in the map or table until you have all the desired records. For example, to get all the states beginning with A, you could sort the table by state name and then find and click on each record containing states starting with A. Or, if you wanted the states that have a Pacific Ocean coastline, you could look on the map, hold down the Shift key, and choose California, Oregon, Washington, and Alaska.

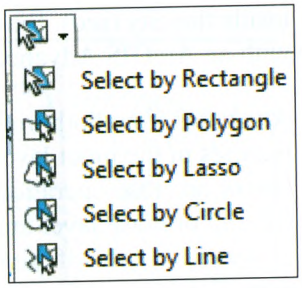


Fig. 8.7. Selecting using shapes

You can use graphics to select features, such as a box, a polygon, a circle, or a line. The Standard toolbar has several tools that select features by constructing a shape on-the-fly (Fig. 8.7). You can also select using an existing graphic, as shown in Figure 8.8. When selecting using a shape or graphic object, features only need to touch the object. Notice that three states are selected in Figure 8.8b, even though the rectangle does not completely enclose any of them. This default setting can be changed in the Selection options menu, however.

The user can also control which layers may be selected using an interactive query. By default, all layers are selectable, so that the rectangle in Figure 8.8a has selected states, counties, rivers, roads, and more. The Selectable Layers option can turn off selection for one or more layers. In Figure 8.8b, the same rectangle has been used as before, but the user has set the selectable layers to states only.

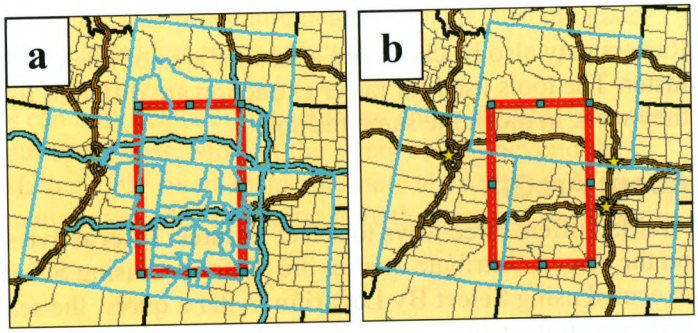


Fig. 8.8. (a) Selecting with all layers selectable; (b) selecting with only states selectable

TIP: The Selectable Layers option applies only to interactive selection with the mouse. It does not affect selecting by attributes or selecting by location using the menus.

Selecting by attributes

SQL expressions in ArcMap are constructed using the fields and the buttons in the Select By Attributes window (Fig. 8.9). The window automatically enters the first part of the expression (SELECT *FROM *layer* WHERE), so in subsequent examples we will drop this part and focus on the text to be entered. Notice a few additional options that can be used in this window.

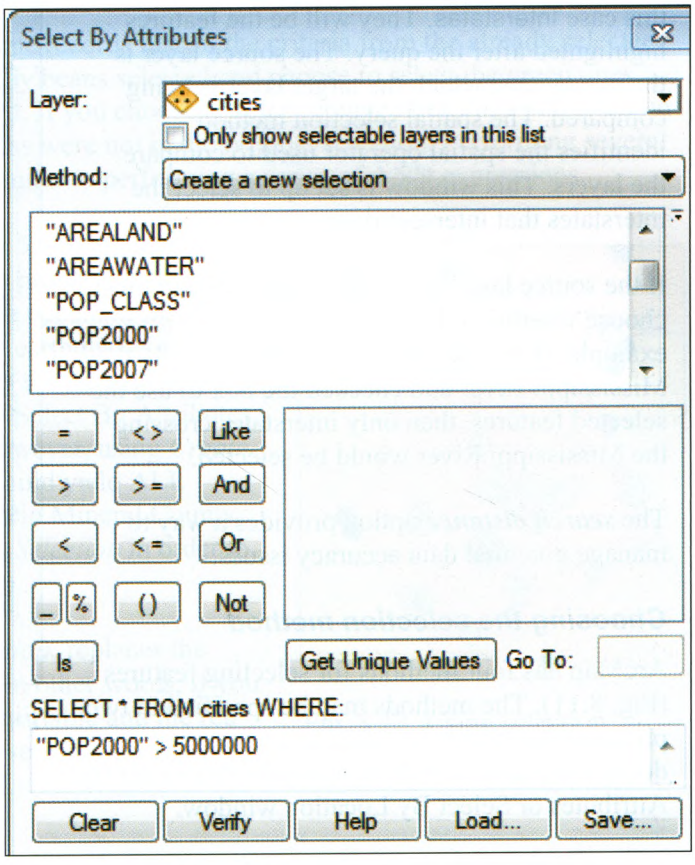


Fig. 8.9. The Select By Attributes window

The check box allows the user to restrict the available layers to include only the ones set as selectable. By default this box is unchecked.

The ***Method*** option controls what happens to previous selections when this one is executed. By default, a previous selection is discarded before the new one is applied. For information about the other options, see the later section "Choosing the selection method."

The ***Get Unique Values*** button adds values from the table to the window where the user may select them instead of typing them. This button is useful for text and categorical data when you don't know the values to use.

The ***Verify*** button checks an SQL expression before executing it.

The ***Save and Load*** buttons store queries for use again later or to record what criteria were used, especially for long, complex queries with many conditions. Queries are saved in text files with the extension .exp and can be edited with a simple text editor.

The ***Clear*** button erases old or invalid queries and starts again.

TIP: Field names may have quotation marks, brackets, or nothing around them, depending on the type of data format and which version of ArcGIS is in use. The Select By Attributes window will always contain the proper syntax when a query is performed.

Selecting by location

The Select By Location window provides the means to set up a spatial query (Fig. 8.10). The target layer refers to the layer being selected, in this case Interstates. They will be the features highlighted after the query. The source layer is the one against which the target layer is being compared. The spatial selection method identifies the spatial operator used to compare the layers. This window is set up to select the interstates that intersect rivers.

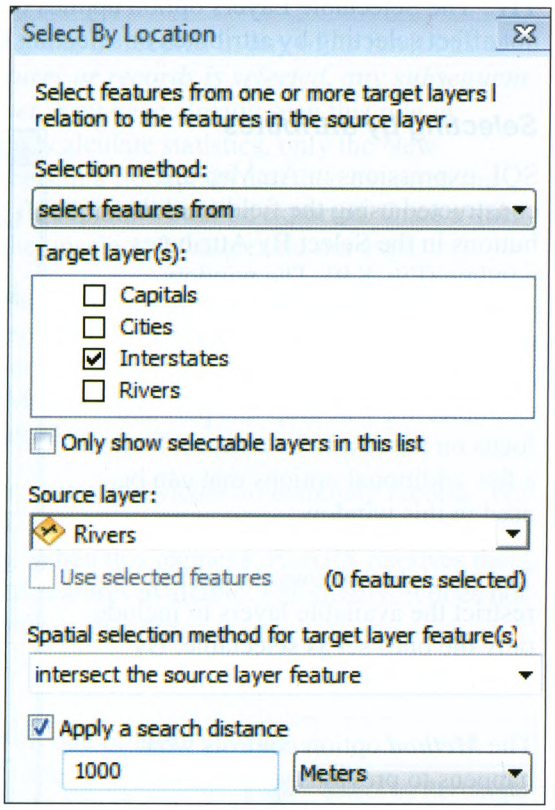


Fig. 8.10. The Select By Location window

If the source layer has a query applied, you must choose whether to honor the selection. For example, if the user previously selected only the Mississippi River and checked the box to use the selected features, then only interstates crossing the Mississippi River would be selected.

The ***search distance*** option provides a way to manage potential data accuracy issues.

Choosing the selection method

ArcMap has four methods for selecting features (Fig. 8.11). The methods may be specified in two places, from the main Selection menu when doing interactive selection or within the Select By Attributes or Select By Location window.

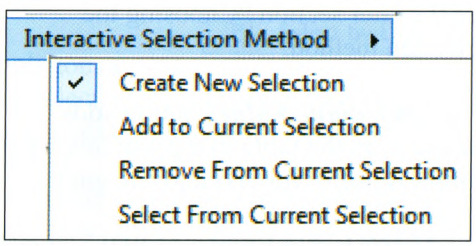


Fig. 8.11. Selection methods

The **selection method** applies to all three types of selections (interactive, selecting by attribute, and selecting by location). These methods give greater flexibility in defining selection sets and enable selection using multiple steps. When performed sequentially, complex expressions involving many criteria are often less confusing than when executed by entering a long expression full of ANDs and ORs and trying to figure out where to put the parentheses.

Create New Selection selects the specified records regardless of what else is already selected, and it replaces any previous selection with the new selected records. For example, if you have selected green jelly beans and then choose to select red jelly beans, then you will only have red jelly beans selected. This method is the default.

Add to Current Selection yields the records that fit the query in addition to any that are currently selected. Thus, if you have green jelly beans already selected and then select the red ones, then you will have both red and green jelly beans selected. Using this method is equivalent to doing a single selection step with a double expression using the OR operator, as when selecting beans where "color" = red OR "color" = green.

Remove From Current Selection takes away the records that meet the selection criteria. If you already have red, green, and yellow jelly beans selected and you "select" green jelly beans, then you will have red and yellow jelly beans left. In this option, what you choose gets deleted from your selected set, so you must think backward on this one. It is equivalent to the NOT operator.

Select From Current Selection chooses the records that fit the criteria from the already selected set. If you have red, green, and yellow jelly beans selected and choose to select the green ones, then you will have the green ones selected. If you choose to select orange ones, then you will end up with *nothing* because orange jelly beans were not in the original selection. Performing several selections in a row with this method is similar to performing a series of AND expressions.

Imagine selecting earthquakes larger than 5.0 magnitude in Mineral County, Nevada (Fig. 8.12). This problem involves selecting by attribute *and* by location, so it occurs in two steps. First, use Select By Location to select the earthquakes within the county, using *Create New Selection* as the method (the green earthquakes in Fig. 8.12). Then use Select By Attributes to choose earthquakes with MAG > 5. However, using *Create New Selection* again in the second step would yield ALL earthquakes with MAG > 5, not just those in Mineral County. You need to specify *Select From Current Selection* in order to obtain the right earthquakes (circled ones).

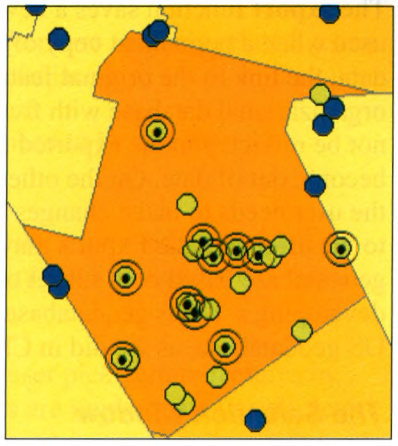


Fig. 8.12. Using selection methods

One other selection option, ***Switch Selection,*** replaces the selected set with the unselected records. In other words, if you have red, yellow, and green jelly beans available and the red ones are selected, and if you do a switch selection, then you will have green and yellow ones selected.

Managing results from queries

After a query, it is often useful to preserve the selected features for future use. Several options are available, both temporary to permanent.

Creating a selection layer

In Figure 8.13, the user selected the southeastern states and created a layer from the selected features, resulting in the new states selection layer shown above the original states layer.

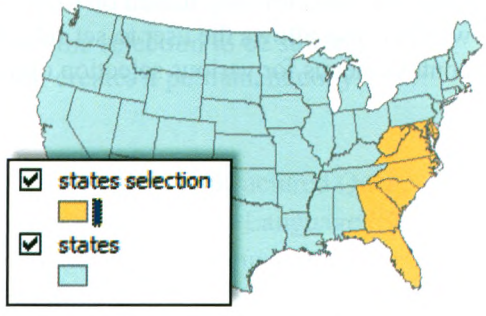


Fig. 8.13. The South Atlantic states were selected and used to create a layer.

Creating a selection layer has several advantages. The layer can be given its own symbols and can be displayed separately from the original layer. The layer preserves the selected features for future reference and eliminates the risk of accidentally clearing the selection and having to redo it. Selection layers can be used to input the same set of features to several tools. They are helpful for viewing and recording intermediate results in a complex series of queries. The layer can also be saved as a layer file for use in other map documents.

Creating a layer is the best solution when a selected set is intended for use within a single map document (although it can also be saved as a layer file for use in other map documents). It does

not create multiple copies of the data, and edits to the source data will be updated automatically. It is the best solution when you need a temporary duplicate of features from your own data or from an organizational database shared by multiple users.

Definition queries

A **definition query** is a property of a layer. It is similar to a selection layer, except that instead of being created from a selected set, the query is performed in the layer properties window. Like selection layers, definition queries refer back to the original database rather than creating a copy of it, and they share the same advantages and disadvantages.

Exporting data

The **export** function saves a new feature class that contains only the selected features, and it is used when a permanent copy of the selection is needed. Because it creates a physical copy of the data, the link to the original feature class is broken. This can be a disadvantage, such as when an organizational database with frequent updates is the source. Changes made to the database will not be reflected in the exported file, which may eventually become out of date. On the other hand, it can be an advantage if the user needs to make changes but does not have write access to the original data. Exports may also be used to develop a new geodatabase based on a subset of existing data, such as developing a Texas geodatabase using feature classes from a US geodatabase, as we did in Chapter 2.

The Selection window

The Table of Contents has a viewing window that facilitates working with queries, the List By Selection window (Fig. 8.14). It reports which layers currently have selections applied and how many features are selected (Counties and Capitals), which layers are selectable but have no current selection applied (Rivers and Cities), and which layers are not available to interactive selection (States and Interstates). The dim layers are not currently turned on to be visible on the map. This window also allows the user to set selection options and access context menus for various selection options and functions.

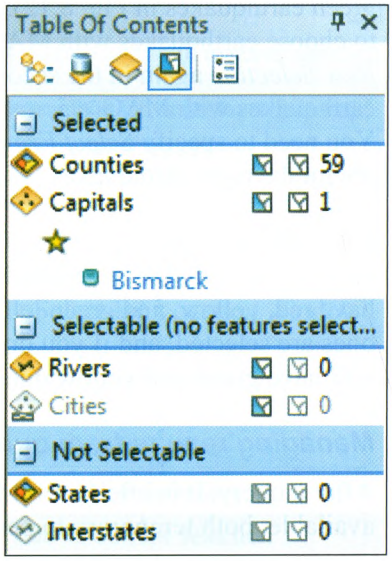


Fig. 8.14. The List By Selection view

Summary

- Queries extract a subset of records or features from a data set based on a set of one or more conditions.
- Extraction functions, such as clip and erase, separate features of interest based on a "cookie cutter" provided by a polygon layer and can truncate features at the cutter boundaries. Attributes are carried through without change.
- Attribute queries extract features based on conditions from fields in the attribute table. They are written in SQL and use operators such as =, >, <, AND, and OR. Partial matches to text strings are evaluated with the LIKE operator.
- Spatial queries extract features based on criteria of how two layers are spatially related, using spatial operators to test containment, intersection, or proximity/adjacency.
- Data accuracy and scale issues may affect spatial queries and need to be considered in evaluating the methods and results.
- ArcGIS uses only the selected features in a layer when a function, command, or tool is applied to the layer. If no query has been applied, all features are used. If a query has been applied, only the selected features are used.
- An interactive query uses the Select Features tool and lets the user pick certain features by finding them on the screen and clicking them. Attribute queries are implemented using Select By Attributes, and spatial queries use Select By Location.
- The Selectable Layers option restricts the operation of the Select Features tool to the specified layers. By default all layers are selectable.
- Four selection methods add greater flexibility to queries. These methods include creating a new selection, adding to the current selection, removing from the current selection, and selecting from the current selection.
- Creating a new layer from a set of selected features allows the selection to be stored, displayed, and passed on to tools or commands. Exporting creates a permanent copy.

Important Terms

attribute query	expression	Select By Attributes	Structured Query
Boolean operators	extraction	Select By Location	Language (SQL)
clip	interactive selection	selection method	Venn diagram
containment	intersection	spatial operators	
definition query	logical operators	spatial query	
erase	proximity		
export			