Helping Organizations Succeed with GIS

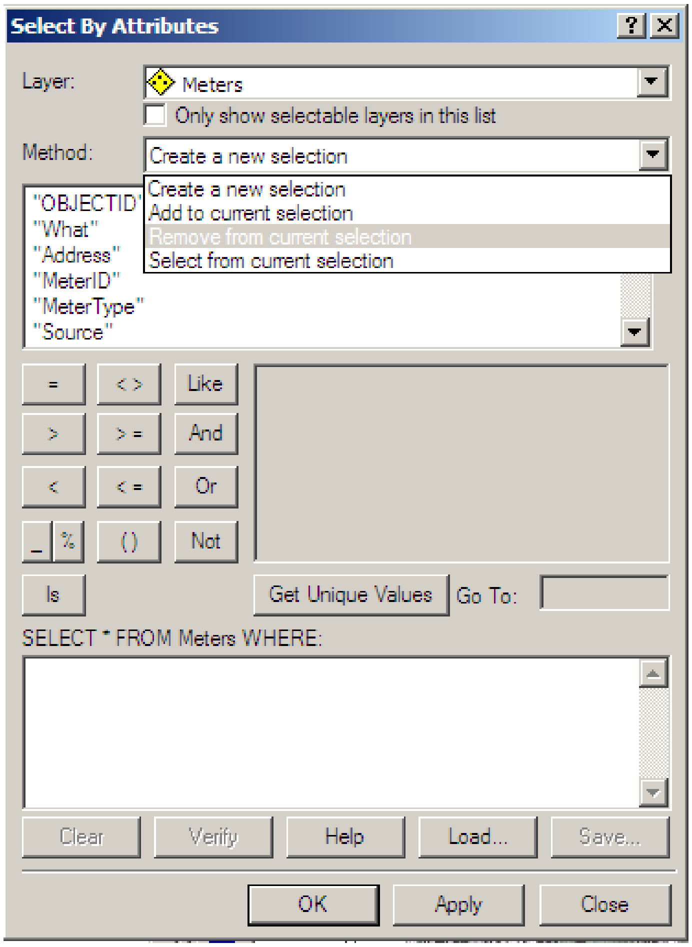
www.juniperGIS.com
Bend, OR 97702
Ph: 541-389-6225
Fax: 541-389-6263

Selecting Features by Attributes in ArcGIS – Using the Query Builder

ESRI provides an easy to use dialog box, which I call the "Query Builder", for selecting or querying features based on the data in the attribute table. This is a very powerful tool, but it helps to know how all the pieces work. Versions of the Query Builder are used in several tools in ArcMap where queries are useful, but the most common way to query is by selecting "Select by Attributes" from the Selection menu in the Standard Toolbar.

Choosing what to query and the query method.

- First specify the layer or table you want to query.
- Next Specify the Selection Method:
  - Create a new selection – is the default and you should start there unless you have already selected features from that layer or table.
  - Add to current selection – us this to add features selected by this query to features you have already selected. Don't forget that these previously selected features could have been selected in any number of ways, such as using Select by Location.
  - Remove from current selection – removes features selected by this query from features you have already selected. This is like saying "Keep everything except these features."


Select By Attributes

Layer: Meters

□ Only show selectable layers in this list

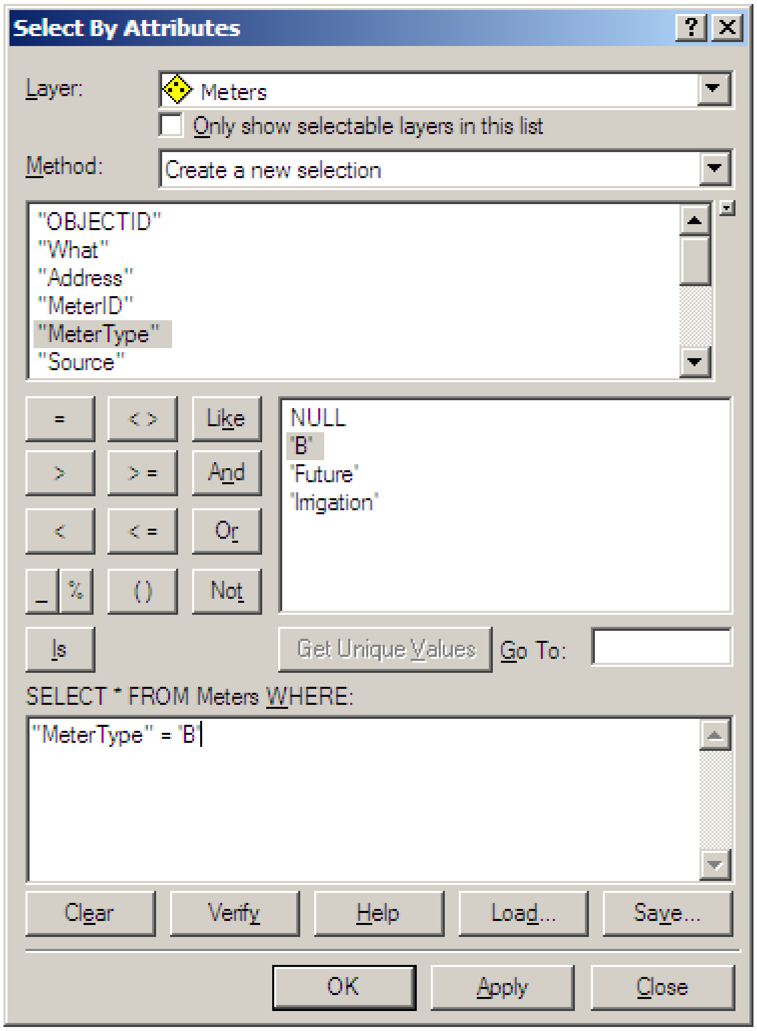
Method: Create a new selection

SELECT * FROM Meters WHERE:

  - Select from current selection - keeps only those features you select in this query. This is like saying "out of all the features selected, I only want to keep these."
- These selection methods are a very powerful way to fine-tune your selection set.

Crating the query expression, or how to tell ArcGIS what you want. In the example here we will query on a specific type of water meter.

- Double-click on the field on which you want to query.



Select By Attributes

Layer: Meters

☐ Only show selectable layers in this list

Method: Create a new selection

"OBJECTID"
"What"
"Address"
"MeterID"
"MeterType"
"Source"

NULL
'B'
'Future'
'Irrigation'

SELECT * FROM Meters WHERE:

"MeterType" = 'B'

- Click once on “Get Unique Values”. This will create a list of all the unique values in that field, which could take a few seconds for large databases. Typing part of the word in the Go To: box will jump to that part of the values list. You should always use Get Unique Values to show the values, because if the value doesn’t show up in this list, it isn’t in the database and you won’t be able to query on it. The only time you should not use Get Unique Values is when there area lot of numbers, like elevation values of population.

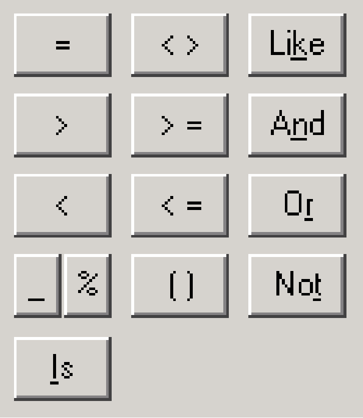
- Single-click on the operator “=, <>, Like, etc” (There’s more on operators on the next page.)

- Then double-click on the value you want from the list of values.

- As you click different parts of the dialog box to create the query, make sure you see the expression being built in the expression box at the bottom of the dialog box. It is very easy to move quickly and not quite get in that double-click or single-click correctly.

- Then click Verify to check that the query is working and then click OK, or Apply to run the query. Verify checks the syntax of the query, but remember that even if the query is written with the correct syntax, it may not be valid query in that it might not give you the information you really want.

Using the **Database Operators** – these specify the relationships between **Fields** and **Values** in queries and provide a lot of power in queries.



- = Equals

- **Like** Similar to “Equal” but used for characters or string data and allows for wildcards and more variations.

- > Greater than < Less than – these can be also used with string data and will then use alphabetical order. Example: “NAME”>’M%’ will give you all the words values starting with M through Z.

- <> Not equal to

- >= Greater than or equal to <= Less than or equal to

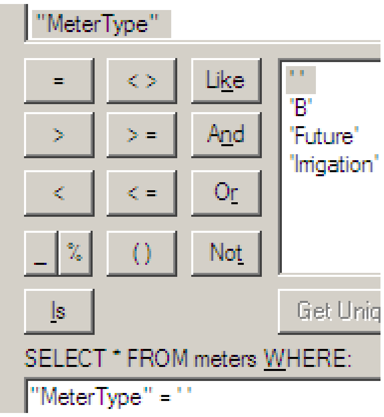
- **And** Both expressions are true; Example: [acres]>= 10 And [property] = “Undeveloped”
  - “And” is <u>exclusive</u> and limits the number of features selected.

- **Or** At least one expression is true;.Example: [acres]>= 10 Or [property] = “Undeveloped”
  - “Or” is <u>inclusive</u> and increases the number of features selected. If you need to select 2 or more values from the same field, you need to use **Or**.
    Example: "PipeClass" = 'C9' OR "PipeClass" = 'C16'

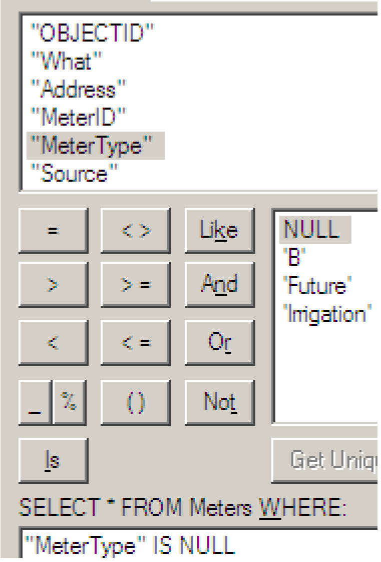
- **Not** excludes values - usually used with “And”
  Example: STRM_CLASS" <= 3 AND NOT "STRM_CLASS" = 0 – this would give you stream classes 1,2 ND 3, and not stream class 0. You could also get the same results by using: STRM_CLASS" = 1 OR STRM_CLASS" = 2 OR STRM_CLASS" = 3

- **Is** – Used with databases that use the NULL keyword as a value for empty records. Example: Is Null, Is Not Null

  If you are working with tables that don’t use NULL, you can use the example here to query for empty records.



"MeterType" = ''



"MeterType" IS NULL

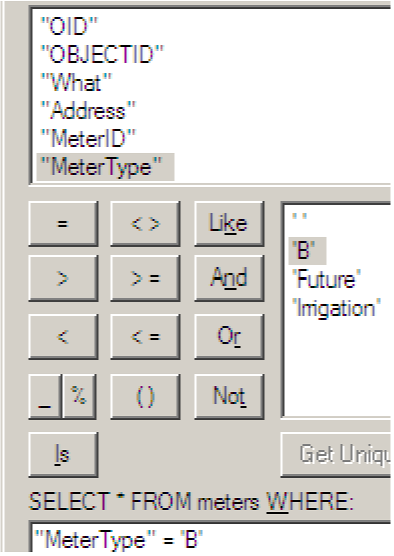
- **()** expressions within parentheses are evaluated first.

- Mathematical operators such as +, -, *, / can also be used in queries

- Wildcards such as _ or **?** (spaceholder) and **%** or ***** (all) can also be used to further refine selections.

Using different types of tables

ArcGIS is smart enough to know that different types of tables require different syntax. If you are querying a file geodatabase or a shapefile the field name is in quotes.



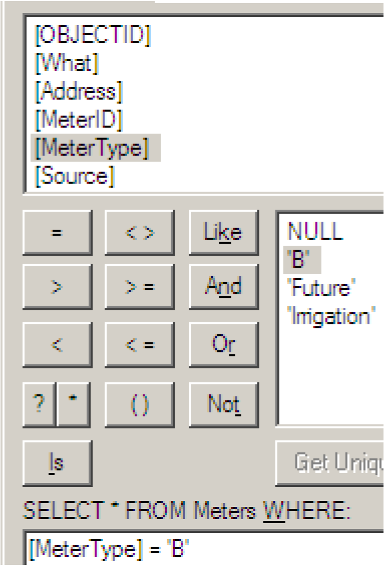
SELECT * FROM meters WHERE:

"MeterType" = 'B'

But if you are querying a personal geodatabase or an Access table, the field name is in brackets. [ ]

Also notice that the characters for wildcards are different. In the picture in the upper right the wild cards are an underscore _ and the **%** sign, while in the picture to the right, the wild cards are the **?** and the ***** characters.

I don’t even pay attention to these difference any more because I never try to type; I always select from the dialog box and it will almost always create the correct syntax.



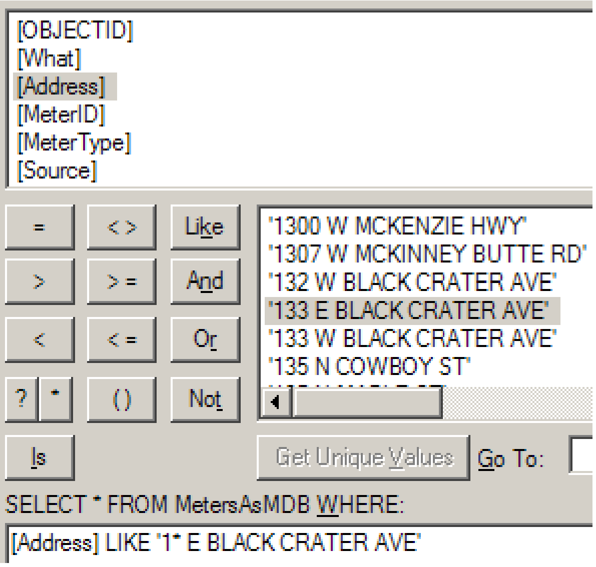
SELECT * FROM Meters WHERE:

[MeterType] = 'B'

Using Wildcards in the Query Builder

Wildcards are used to select subsets of a text string. The two wildcards are the “space holder”, which is either an underscore _ or a **?** and the “everything” which is either the **%** or the ***** depending on the type of table being queried. The wildcards are usually used with **Like** as the operator. Here’s a few examples of how these would be used, showing the ? and the * as the wildcard characters.

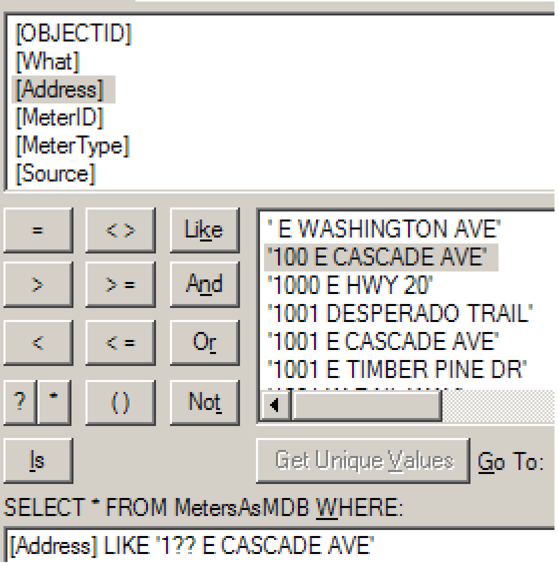
In this example I want to select all the water meters in the 100 block of East Black Crater Avenue. Double-click on **Address**, click on **Get Unique Values** to see all the addresses, and then double click on any address in the 100 block such as 133 E BLACK CRATER AVE. Then just delete the “33” and replace that with the ***** wildcard. The result is all the meters in that block where the address starts with **1** are selected. In this case this works because there is no 1000 block, but there is a way to select only from the 100 block even if there was a 100 block and a 1000 block.



SELECT * FROM MetersAsMDB WHERE:

[Address] LIKE '1* E BLACK CRATER AVE'

In this example, you can see that we have addresses for 100 East Cascade Avenue and 1001 East Cascade Avenue. If I used the expression above, I would get all the addresses in both blocks. But here is how to use the space holder wildcard to get only those meters in the 100 block. I start the expression above, but tell ArcGIS I only want addresses that start with 1 and have two more characters by putting two **??** after the 1.



SELECT * FROM MetersAsMDB WHERE:

[Address] LIKE '1?? E CASCADE AVE'

Let's try one more example. What if I want all the meters for the 100 block of East Cascade and West Cascade avenues? (If you ever drive through Sisters, Oregon these are the two blocks on the main street in the center of town.) I could just take the query above and change it to: **[Address] LIKE '1?? * CASCADE AVE'**

Understanding how to use these wildcards will make it a lot easier to query pieces of information out of any table.

Compound and nested Queries

Compound queries are used when you need to get selected features based on multiple criteria, and use connectors such as **AND** or **OR**. Remember that when using AND in a query, you will get fewer results because all parts of the expression need to be true; and when you use OR, you will get more results because either part of the expression can be true.

For example, I am searching for large undeveloped properties. I would then use this expression that would find large properties and then restrict those to only properties that are undeveloped.

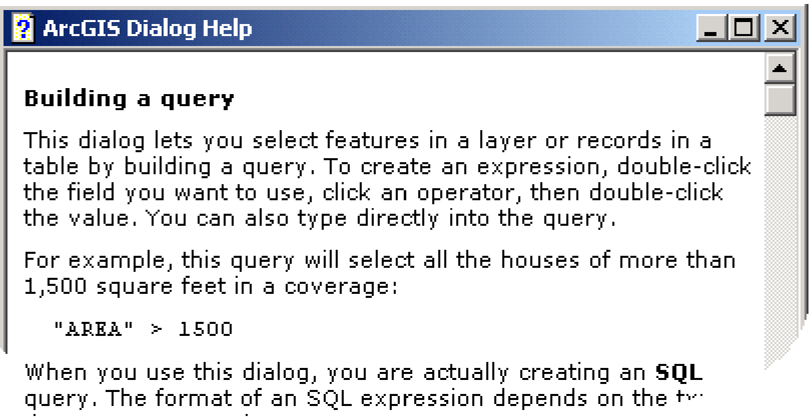
[acres]>= 10 AND [property] = "Undeveloped"

But if I changed the expression to *[acres]>= 10 OR [property] = "Undeveloped"* I could end up with large, developed properties or small undeveloped properties.

If I am searching for large undeveloped or agricultural properties I could use a nested query as part of a compound query. *[acres]>= 10 AND ([property] = "Undeveloped" OR [property] = "Residential"*. I used the **( )** to have ArcGIS read the part inside the parentheses as a sub query. Alos notice that when I queried for Residential and Undeveloped I could not use the expression *([property] = "Undeveloped" OR "Residential"*. When using OR like this, you need to repeat the full phrase with the field name for each value on which you are querying.

Other options in the Query Builder dialog box

- Queries are temporarily saved in dialog box; next time you open the dialog box, the last query will be there – usually.
    - **Clear** – clears existing query
    - **Verify** – tells you if a query is valid, but doesn't tell you what the problem is if it isn't valid
    - **Load** – loads a saved query
    - **Save** – saves a query, can be reused in other maps.
    - **Help** provides help on query options



Building a query

This dialog lets you select features in a layer or records in a table by building a query. To create an expression, double-click the field you want to use, click an operator, then double-click the value. You can also type directly into the query.

For example, this query will select all the houses of more than 1,500 square feet in a coverage:

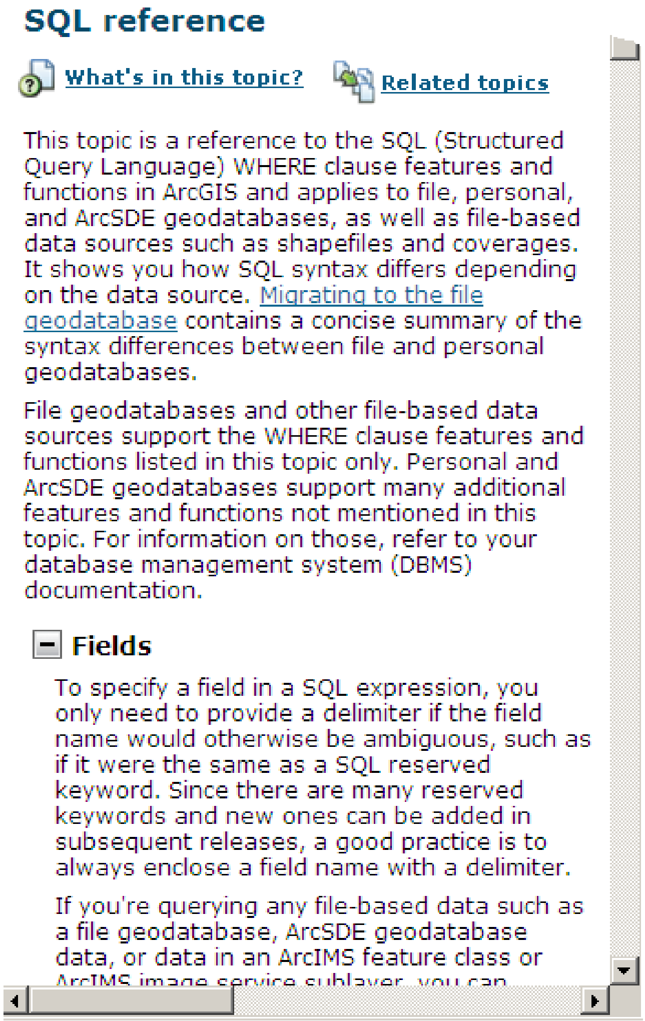
```
"AREA" > 1500
```

When you use this dialog, you are actually creating an **SQL** query. The format of an SQL expression depends on the

Other Query Operators

ArcGIS only shows a small set of query operators you can use in the Query Builder.

- Check Help under "SQL reference" for help on creating expressions and additional operators.
    - **In** includes, e.g. "Park" in (4,5) or Park includes 4 and 5
    - **Between** [Strm_Class] Between 1 and 3 (works with Geodatabases)



SQL reference

This topic is a reference to the SQL (Structured Query Language) WHERE clause features and functions in ArcGIS and applies to file, personal, and ArcSDE geodatabases, as well as file-based data sources such as shapefiles and coverages. It shows you how SQL syntax differs depending on the data source. Migrating to the file geodatabase contains a concise summary of the syntax differences between file and personal geodatabases.

File geodatabases and other file-based data sources support the WHERE clause features and functions listed in this topic only. Personal and ArcSDE geodatabases support many additional features and functions not mentioned in this topic. For information on those, refer to your database management system (DBMS) documentation.

Fields

To specify a field in a SQL expression, you only need to provide a delimiter if the field name would otherwise be ambiguous, such as if it were the same as a SQL reserved keyword. Since there are many reserved keywords and new ones can be added in subsequent releases, a good practice is to always enclose a field name with a delimiter.

If you're querying any file-based data such as a file geodatabase, ArcSDE geodatabase data, or data in an ArcIMS feature class or ArcIMS image service sublayer, you can