Data Structures and Personal Geodatabases in GIS

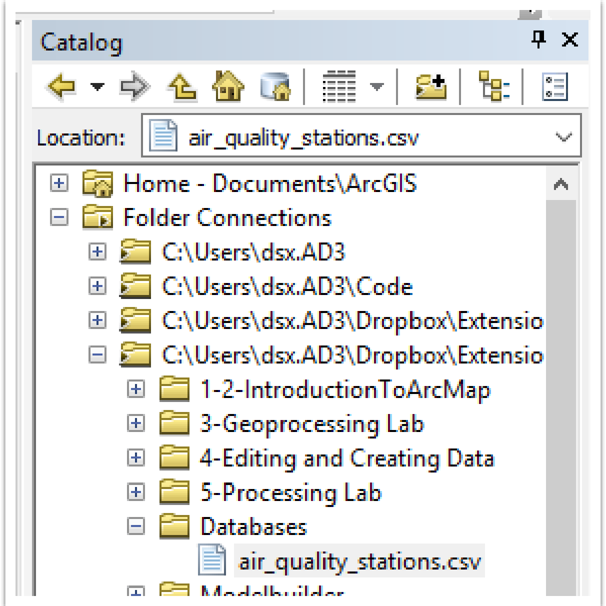
UC DAVIS EXTENSION

In this tutorial, you'll learn workflows and data structures common in GIS environments. In this tutorial, you'll make a database of air quality monitoring data for California, by first creating the monitoring location features from a data table, then creating the necessary supporting data in a personal geodatabase.

We'll start with a comma separated values (CSV) data table of the locations of air quality monitoring stations in California. You will load that into ArcGIS and turn it into point features. Then, you will create a new data table of air quality observations and define relationships to connect it to the monitoring stations.

1. Getting set up

To get started, open up a new, blank document in ArcMap and make sure you have a folder connection that can access the folder containing the data for this lab tutorial (air_quality_stations.csv).

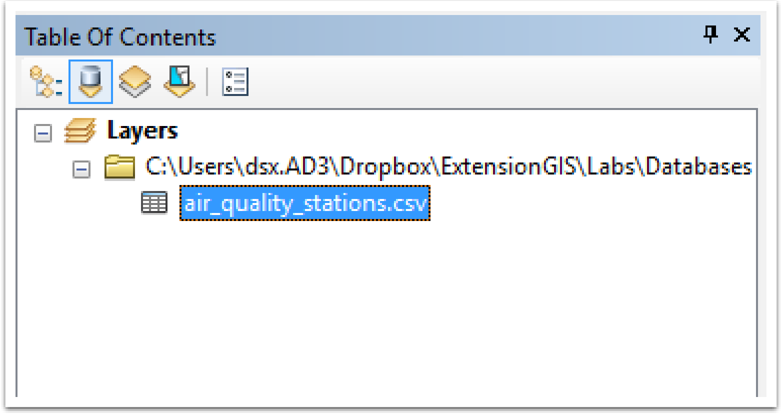


Location: air_quality_stations.csv

1.1 Add data to your map

Next, add the data table *air_quality_stations.csv* to your map document. This data table includes the locations of all of the air quality monitoring stations managed by the California Air Resources Board, and it's what we'll work with for the rest of this tutorial - we'll turn it into features, then enter data corresponding to a few of the stations in southern California (in Long Beach and Thousand Oaks). How you add the data is up to you. You can either do it from the *Add Data* dialog, or by dragging and dropping it from the Catalog window.

Note how once you do, there is still nothing in your map. We've only added a table, and it doesn't have a representation in the map because it's not feature data - *yet.*



1.2 Open the attribute table

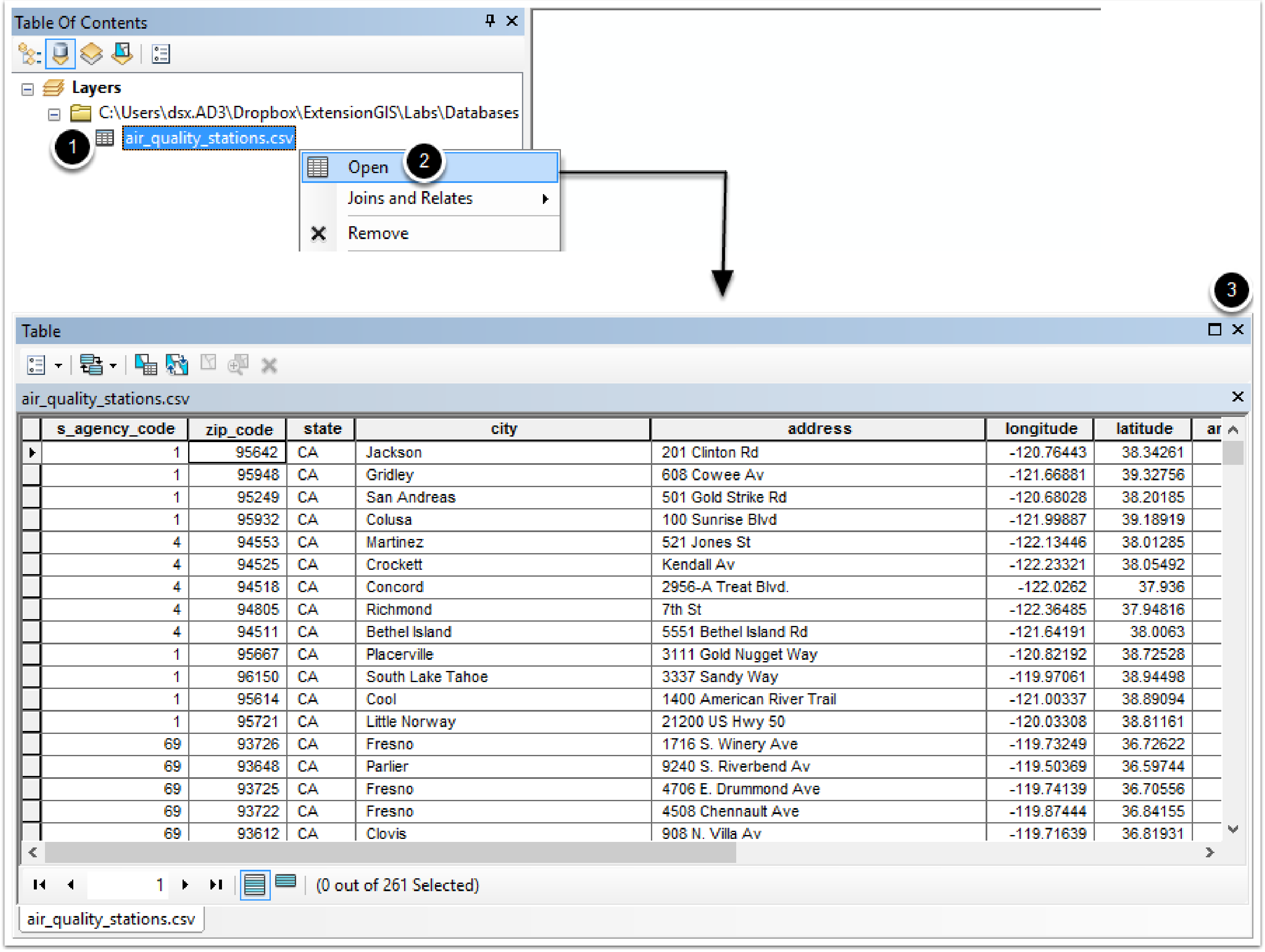
1. Right click on the data table and
2. Click *Open* to open the data table for viewing

The attribute table will open - take a look through it for a moment. Notice anything interesting about it?

This table has two things worth pointing out. First, it doesn't have a primary key column - in most cases a field named *OBJECTID*. This table has no field with a unique identifier for each record. That's a bad thing, and we'll rectify it later. Second, it has two fields with coordinate information - *longitude* and *latitude*. But ArcMap still doesn't display it in the map. Why might that be?

First, ArcGIS can't be positive these are coordinates until we tell it that those fields contain them, so it stays out of the way and leaves the table alone. Second, unless we provide the information, it can't be sure of what coordinate system or projection those coordinates are in. There are thousands of possible coordinate systems, so we need to tell it which one those data use in order to display the data. So let's do that now.

When you're ready to proceed, close the attribute table (3).

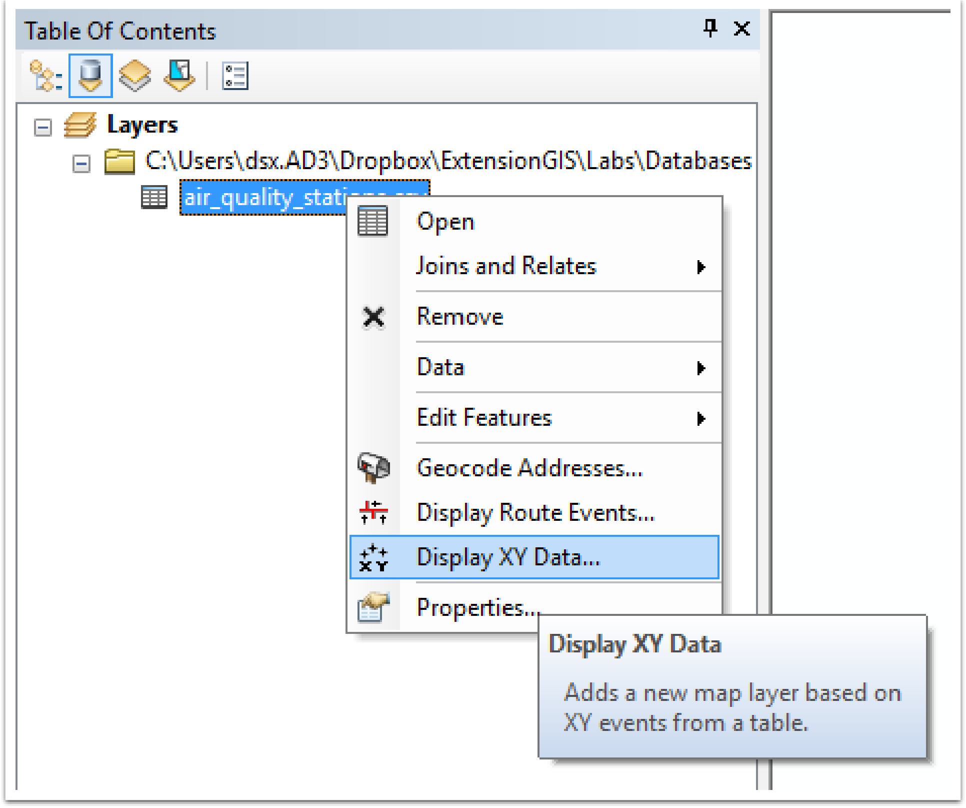


s_agency_code	zip_code	state	city	address	longitude	latitude	ar
1	95642	CA	Jackson	201 Clinton Rd	-120.76443	38.34261	
1	95948	CA	Gridley	608 Cowee Av	-121.66881	39.32756	
1	95249	CA	San Andreas	501 Gold Strike Rd	-120.68028	38.20185	
1	95932	CA	Colusa	100 Sunrise Blvd	-121.99887	39.18919	
4	94553	CA	Martinez	521 Jones St	-122.13446	38.01285	
4	94525	CA	Crockett	Kendall Av	-122.23321	38.05492	
4	94518	CA	Concord	2956-A Treat Blvd.	-122.0262	37.936	
4	94805	CA	Richmond	7th St	-122.36485	37.94816	
4	94511	CA	Bethel Island	5551 Bethel Island Rd	-121.64191	38.0063	
1	95667	CA	Placerville	3111 Gold Nugget Way	-120.82192	38.72528	
1	96150	CA	South Lake Tahoe	3337 Sandy Way	-119.97061	38.94498	
1	95614	CA	Cool	1400 American River Trail	-121.00337	38.89094	
1	95721	CA	Little Norway	21200 US Hwy 50	-120.03308	38.81161	
69	93726	CA	Fresno	1716 S. Winery Ave	-119.73249	36.72622	
69	93648	CA	Parlier	9240 S. Riverbend Av	-119.50369	36.59744	
69	93725	CA	Fresno	4706 E. Drummond Ave	-119.74139	36.70556	
69	93722	CA	Fresno	4508 Chennault Ave	-119.87444	36.84155	
69	93612	CA	Clovis	908 N. Villa Av	-119.71639	36.81931	

2. Turning our data into points

Let's take those coordinates and tell ArcGIS how to use them as points. This will spatially enable our layer. Right now, it has spatial data, but it's not able to be analyzed that way.

1. Right click on the *air_quality_stations.csv* table
2. Click on *Display XY Data* in the menu.



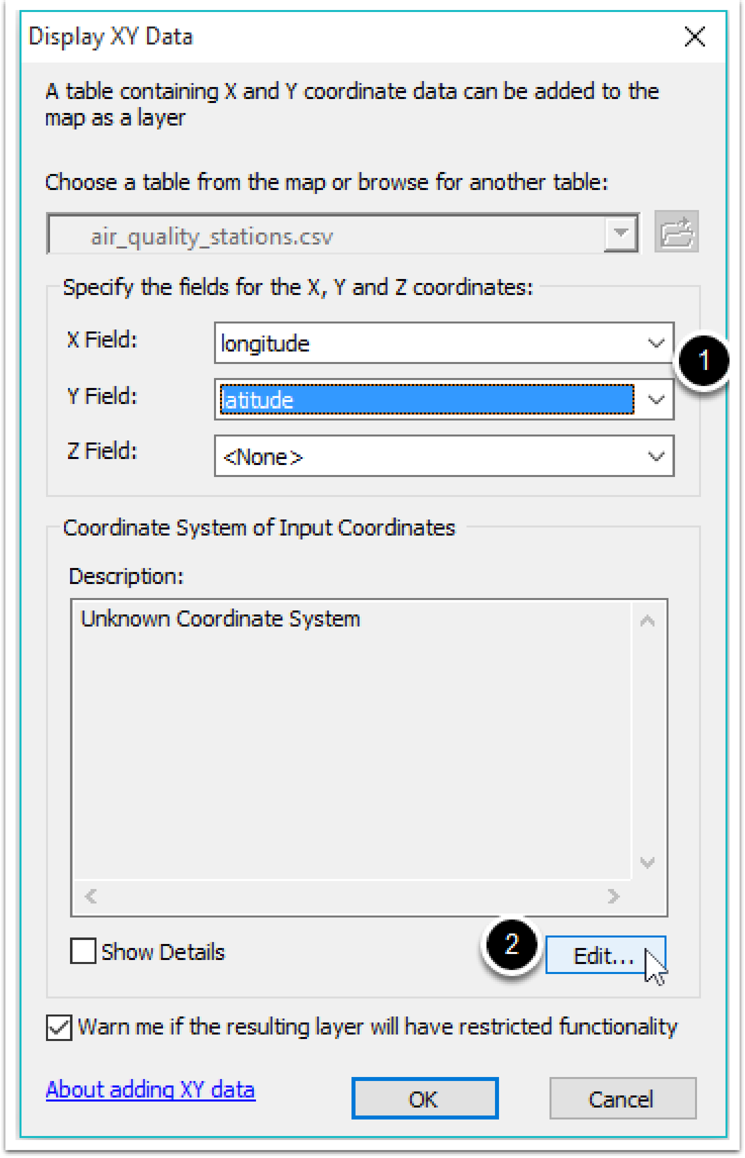
2.1 The Display XY Data dialog

A new dialog will pop up, where we need to tell ArcMap 2 primary pieces of information in order to display the data as points. First, we need to tell it which fields provide the coordinates for each record. And second, we need to tell it what coordinate system the coordinates in those fields are in. Consider that for a moment - since there are many different ways we can reference data to the Earth, we need to tell ArcGIS which way, which coordinate system, we are using.

ArcMap should automatically populate the *X Field* and *Y Field* options with longitude for the X field and latitude for the Y field. Make sure to check that the field selections are correct before you proceed (1).

Sometimes matching longitude to the X coordinate and latitude to the Y coordinate can confuse people because we often say "X and Y" as well as "latitude and longitude", but if you consider that latitude moves upward from the equator (in a Y direction) and longitude moves away from the Prime Meridian (in an X direction), then it should make more sense.

Now, let's select the coordinate system. Click the *Edit* button within the *Coordinate System of Input Coordinates* section to bring up the coordinate system selection dialog.



Display XY Data

A table containing X and Y coordinate data can be added to the map as a layer

Choose a table from the map or browse for another table:

air_quality_stations.csv

Specify the fields for the X, Y and Z coordinates:

X Field: longitude

Y Field: latitude

Z Field: <None>

Coordinate System of Input Coordinates

Description:

Unknown Coordinate System

☐ Show Details

☑ Warn me if the resulting layer will have restricted functionality

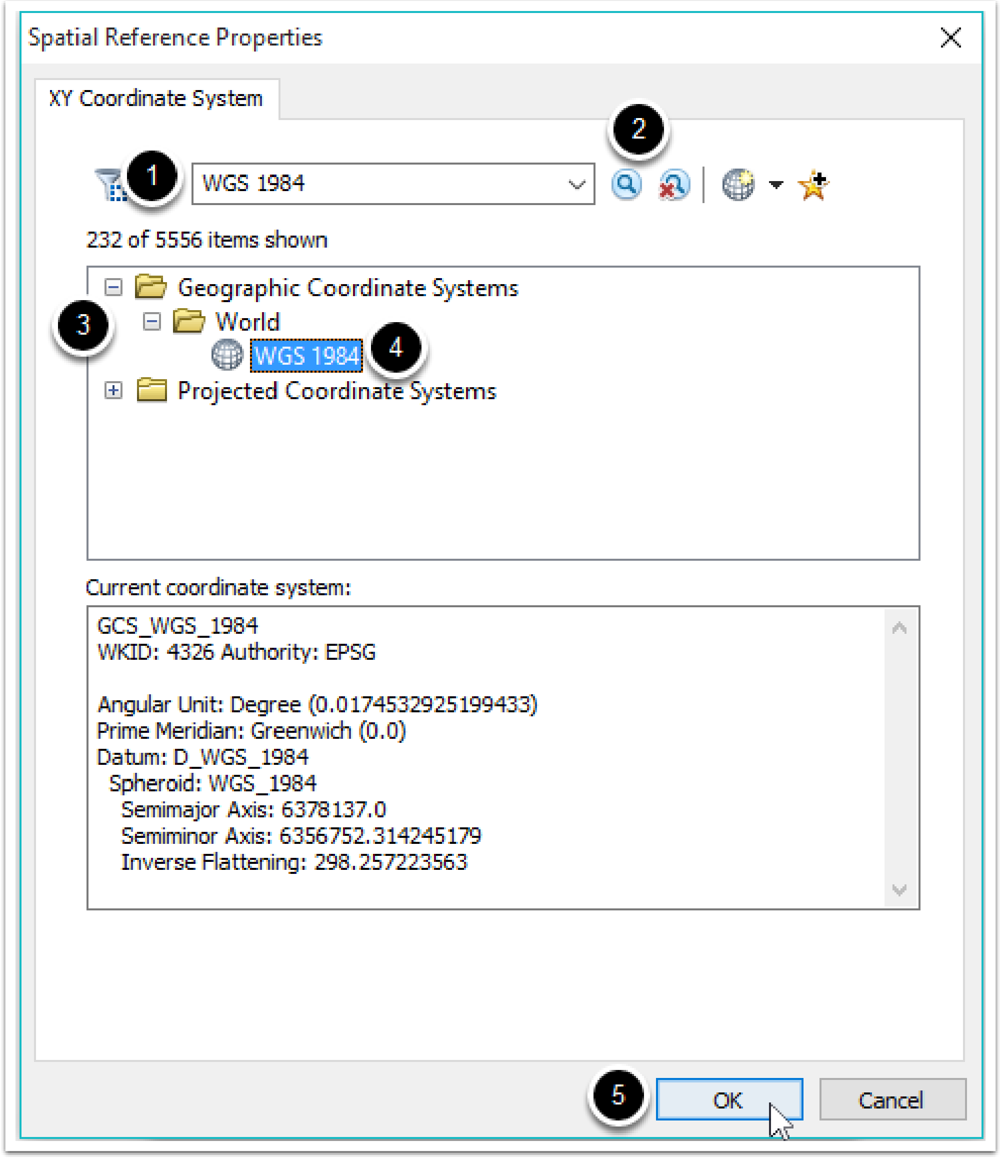
2.2 Select the Coordinate System

This box should probably be starting to feel familiar if you took the previous course, but just as a review, we use it to select a coordinate system that will be used for the dataaset. This isn't something you guess - it's something you *kno*w based on where you received the data from. It's an inherent property of the data, built into the coordinates themselves, but this information isn't included with the data because we brought in a nonspatial format - the comma separated values file. So that's why we need to tell ArcMap how to interpret the coordinates.

In this case, the coordinate system is the Geographic Coordinate System named *WGS 1984.*

1. Type *WGS 1984* in the search box
2. Click the search button or hit the *Enter* key to search
3. Expand out the controls for *Geographic Coordinate Systems* and *World.*
4. Click on WGS 1984 to select it
5. Click *OK* to save the setting and close the box.

*Bonus:*As a note for those curious about how to determine the coordinate system, it depends where the data came from. In this case, I adapted this data from a Google Map on the website of the California Air Resources Board. Google Maps uses the WGS 1984 coordinate system for specifying locations (even though the map is displayed in a different coordinate system!), so I knew what coordinate system these data are in. That said, there are tools for determining coordinate systems when you don't know, but it's complicated and error prone. It's often best to provide someone a sample of your coordinates in an online forum if you aren't sure what coordinate system they are in, to see if anyone knows. In the long run, you'll get a sense for the format of many common coordinate systems.



WGS 1984

232 of 5556 items shown

Geographic Coordinate Systems
World
WGS 1984
Projected Coordinate Systems

Current coordinate system:

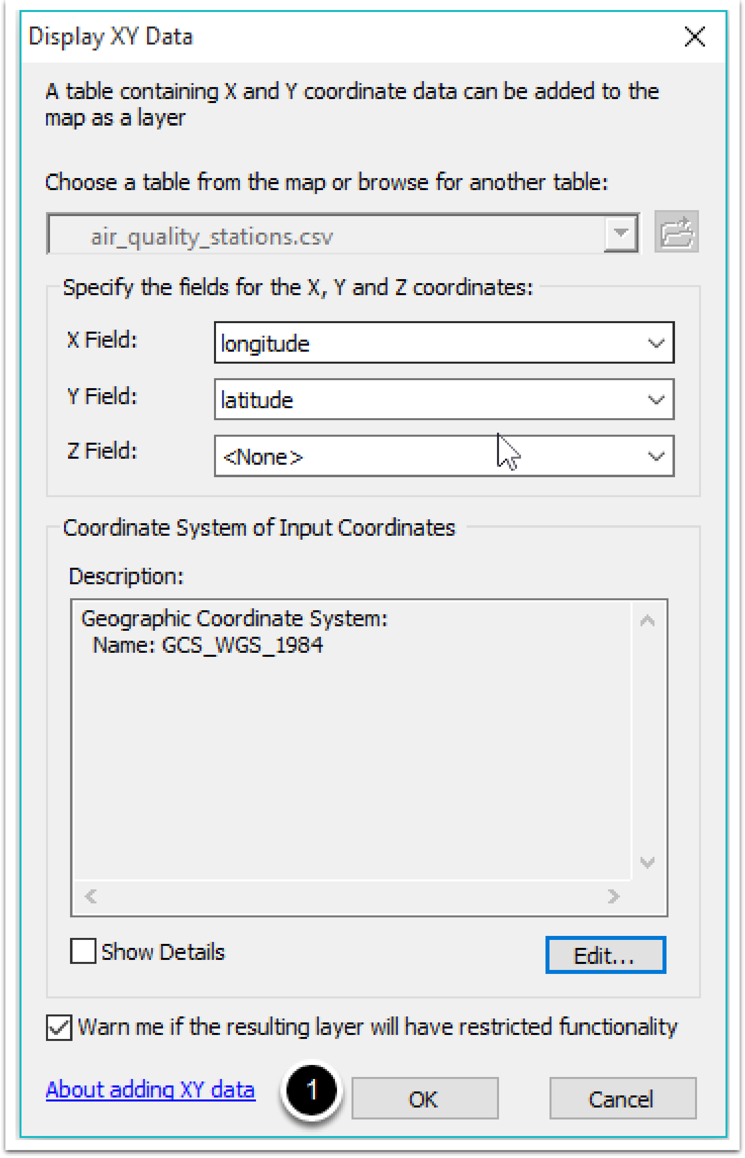
GCS_WGS_1984
WKID: 4326 Authority: EPSG

Angular Unit: Degree (0.0174532925199433)
Prime Meridian: Greenwich (0.0)
Datum: D_WGS_1984
Spheroid: WGS_1984
Semimajor Axis: 6378137.0
Semiminor Axis: 6356752.314245179
Inverse Flattening: 298.257223563

2.3 Close the Display XY Data Dialog

You should now be back at the Display XY Data dialog.

1. We're ready to continue, so just click the *OK* button to create the new layer



Display XY Data

A table containing X and Y coordinate data can be added to the map as a layer

Choose a table from the map or browse for another table:

air_quality_stations.csv

Specify the fields for the X, Y and Z coordinates:

X Field: longitude

Y Field: latitude

Z Field: <None>

Coordinate System of Input Coordinates

Description:

Geographic Coordinate System:
Name: GCS_WGS_1984

□ Show Details

☑ Warn me if the resulting layer will have restricted functionality

2.4 A warning!

ArcGIS pops up a warning letting us know that our functionality on this layer will be limited since it doesn't have a primary key field (OBJECTID). This is ok - we already knew this data didn't have it - we'll deal with it soon. In fact, even if this data did have a primary key field, ArcGIS wouldn't understand it since we're bringing this data in from a CSV - it could see the primary key data as integers, but wouldn't know that it was the key field, so we would still get this warning. More structured data tables don't have that problem in this tool - it's linked to our use of the text-based CSV.

1. Click OK to continue after reading the warning.

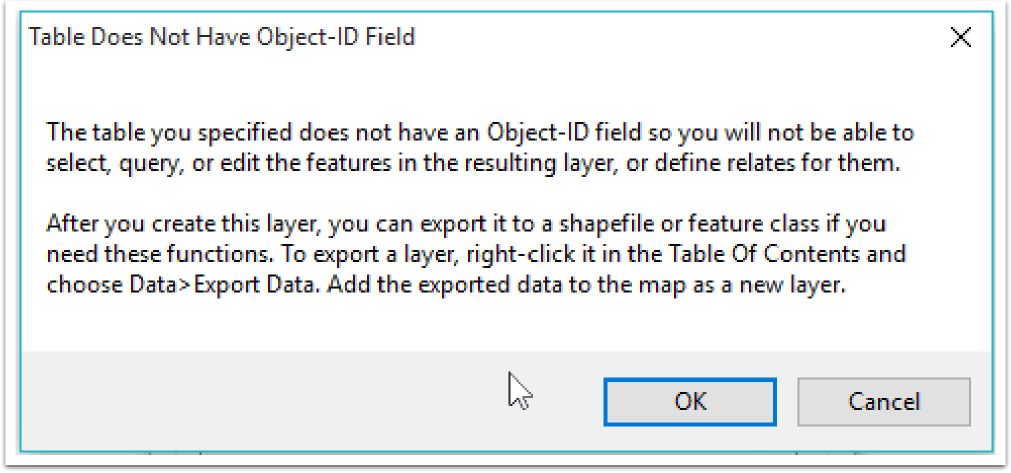


Table Does Not Have Object-ID Field

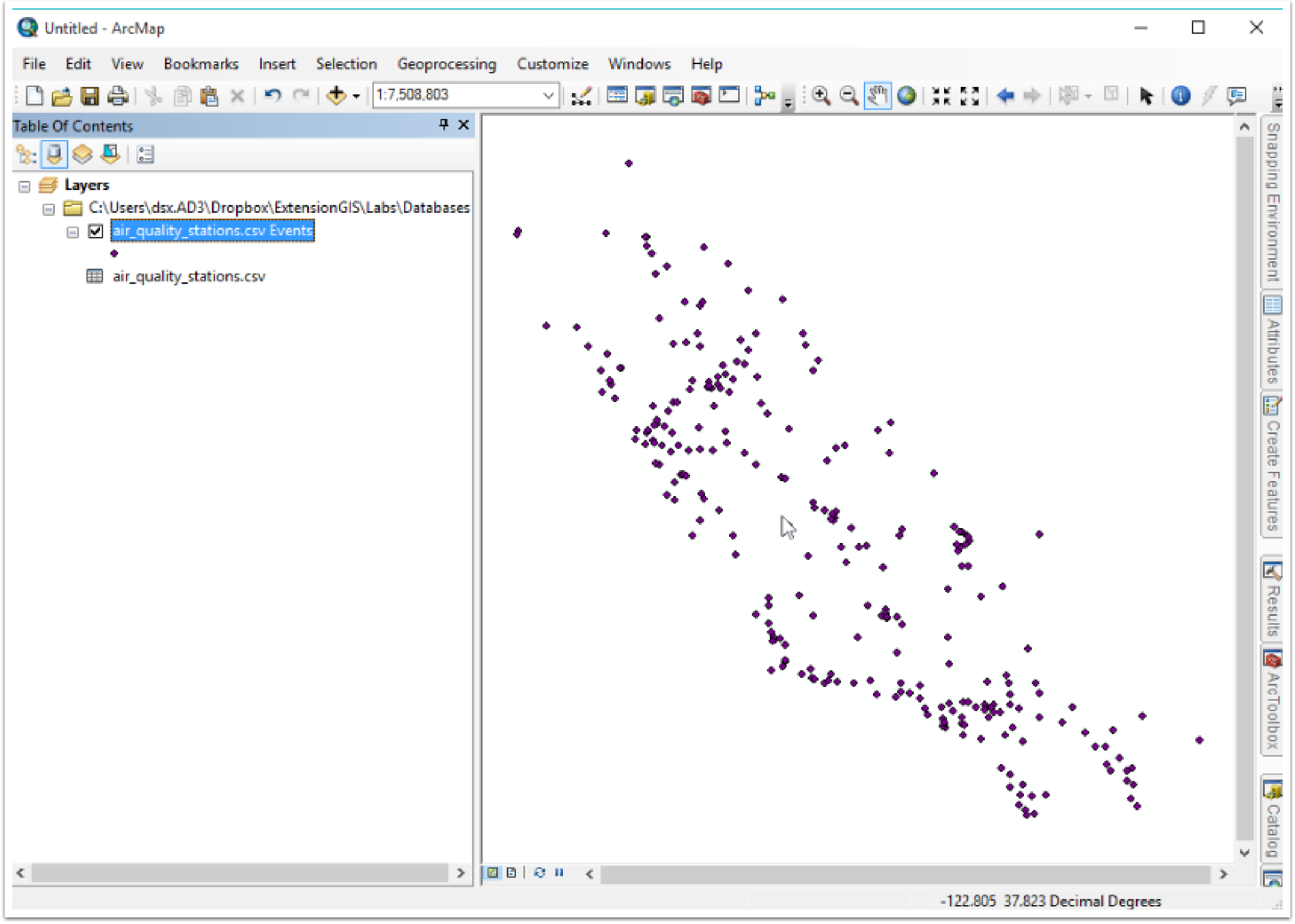
The table you specified does not have an Object-ID field so you will not be able to select, query, or edit the features in the resulting layer, or define relates for them.

After you create this layer, you can export it to a shapefile or feature class if you need these functions. To export a layer, right-click it in the Table Of Contents and choose Data>Export Data. Add the exported data to the map as a new layer.

3. Points created

Woohoo! Now we get a points layer of all of the stations. Take a moment to look at the attribute table, and you'll see that it's the same as the data table we loaded in. You can also try to make a selection - you won't be able to though, because the data don't have primary keys! Let's fix that now. We'll export the data layer into a geodatabase, but first let's create that geodatabase.

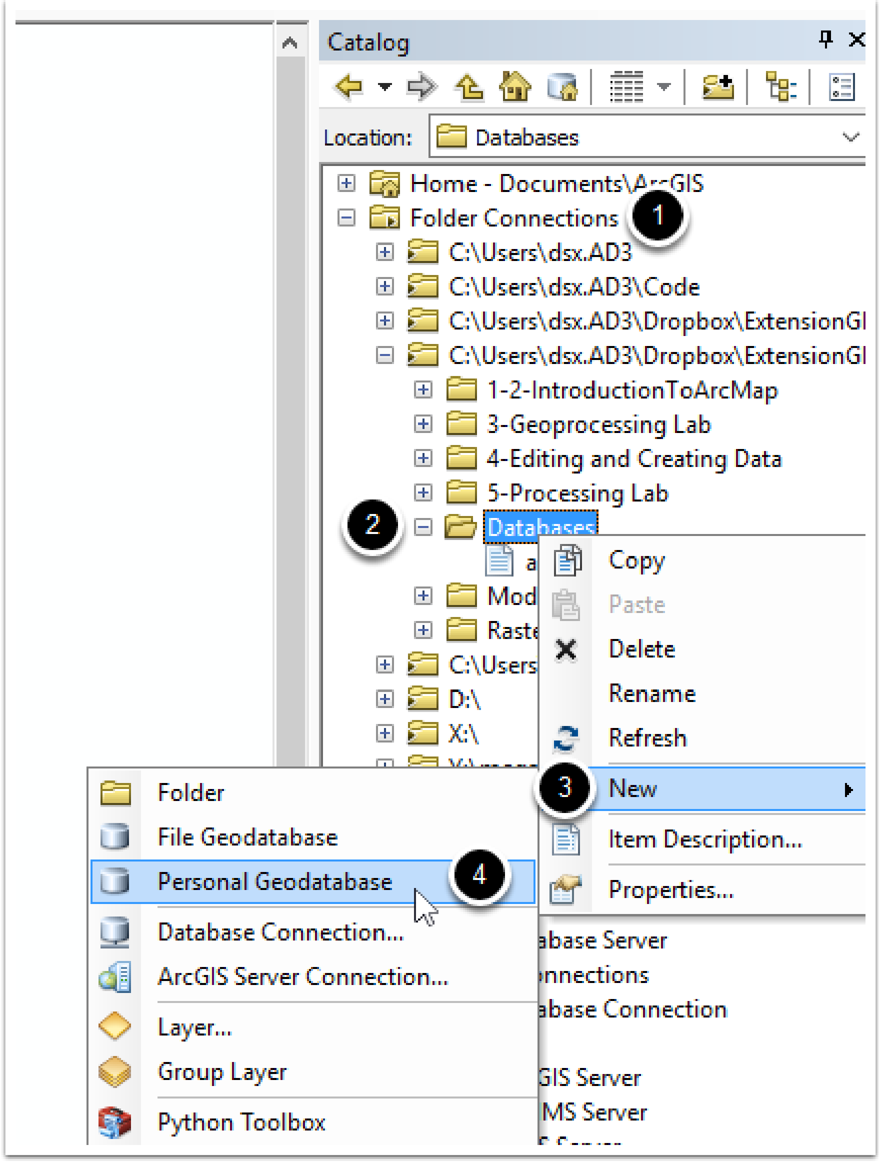
Extra: If you'd like to have more context for the data you're working with, feel free to add a basemap and explore before proceeding. The screenshots won't have it, but we won't be working with the data frame much in this tutorial so it shouldn't be a problem for following along.


Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:7,508,803
Table Of Contents
Layers
C:\Users\dsx.AD3\Dropbox\ExtensionGIS\Labs\Databases
air_quality_stations.csv Events
air_quality_stations.csv
Snapping Environment
Attributes
Create Features
Results
ArcToolbox
Catalog
-122.805 37.823 Decimal Degrees

3.1 Create a new personal geodatabase

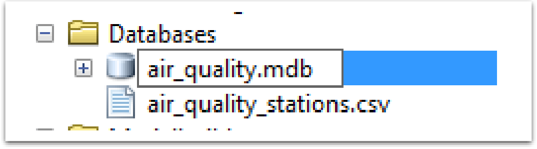
Since we're going to use these air quality stations to associate plenty of other relational data, let's create a *personal geodatabase* for this data.

1. In the Catalog window, find the folder that holds the data table we just used.
2. Right click on that folder
3. Hover your mouse over the *New* option so the menu flies out
4. Click *Personal Geodatabase* to create a new personal geodatabase in that folder.



3.2 Rename the geodatabase

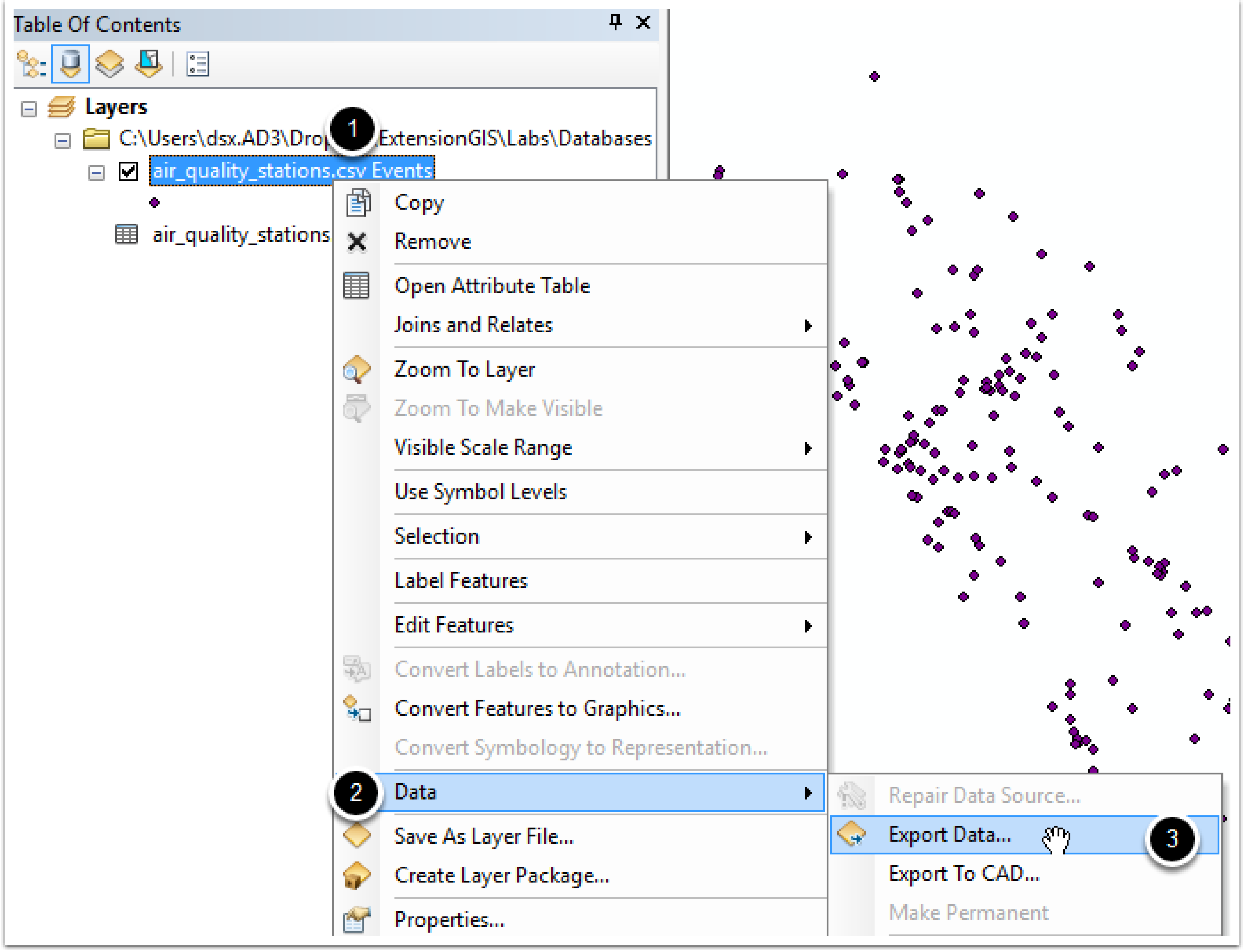
It will prompt you for a name (defaulting to *New Personal Geodatabase.mdb*). Name the database *air_quality.mdb*.


Databases
air_quality.mdb
air_quality_stations.csv

3.3 Export the data

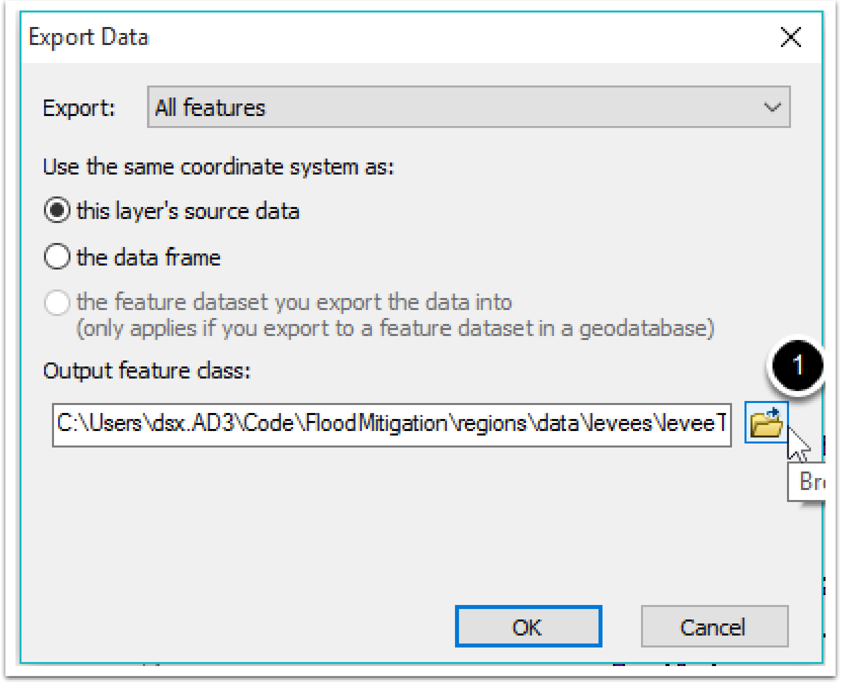
Now that we have our geodatabase created, let's export out data into it. The layer is currently temporary and doesn't have an OBJECTID field. To rectify both of those issues, let's export the layer to a new dataset.

1. Right click on the layer
2. Go to the *Data* option and hover your mouse
3. Click on *Export Data*



3.4 Check the options and browse

Check the options to make sure they all look ok, then click the Browse button (folder icon)



Export Data

Export: All features

Use the same coordinate system as:

● this layer's source data

○ the data frame

○ the feature dataset you export the data into
(only applies if you export to a feature dataset in a geodatabase)

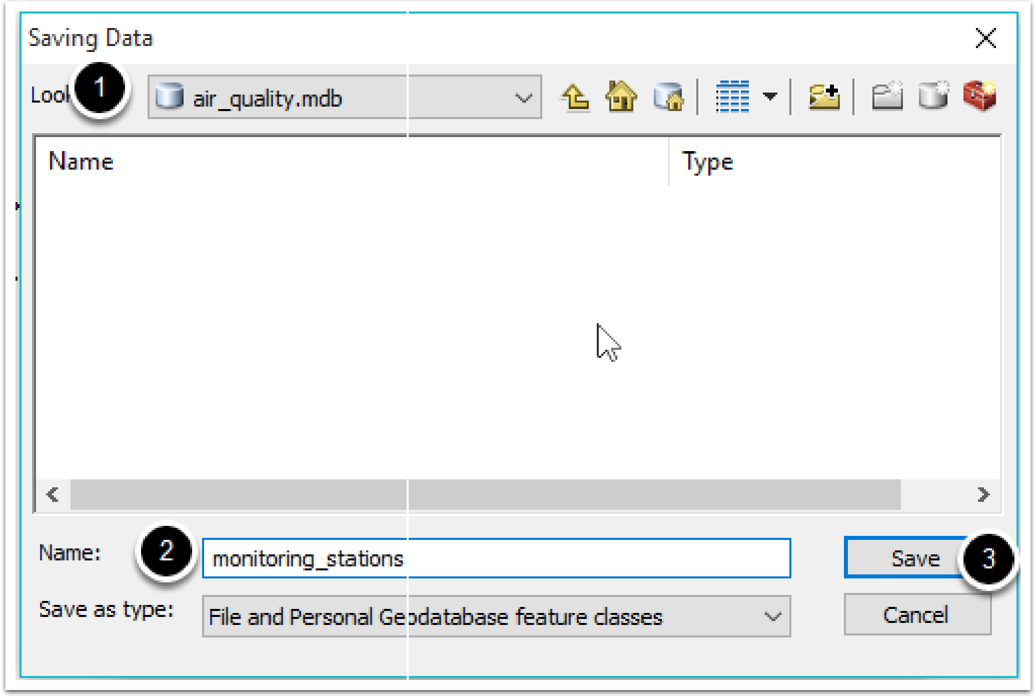
Output feature class:

C:\Users\dsx.AD3\Code\FloodMitigation\regions\data\levees\leveeT

3.5 Save the layer

1. Navigate to your newly created *air_quality.mdb* and open the database up
2. Type the name *monitoring_stations* in the bottom box to give the layer that name.
3. Click Save to name the layer in this location

Then click OK on the parent dialog box to finalize the export



Saving Data

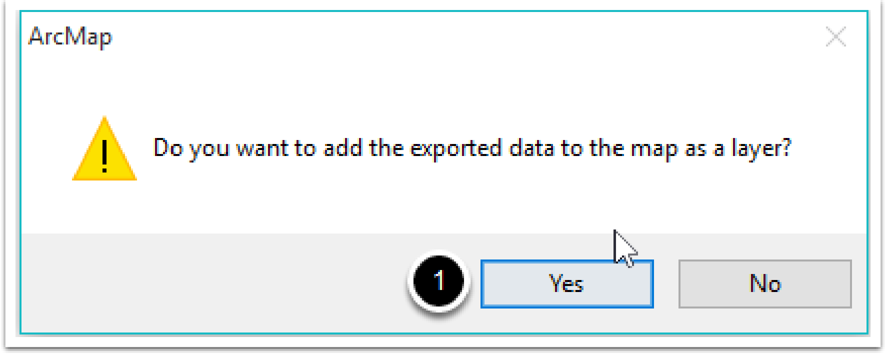
Look in: air_quality.mdb

Name: monitoring_stations

Save as type: File and Personal Geodatabase feature classes

3.6 Add to map

When it finishes exporting the data, it will ask if you want to add it to the map. Click *Yes* (1) to do that.



ArcMap

Do you want to add the exported data to the map as a layer?

3.7 Remove the old data

Remove the original events layer (not our exported copy), as well as the csv data table from your map document by right clicking on the layer (1), then clicking *Remove* (2).

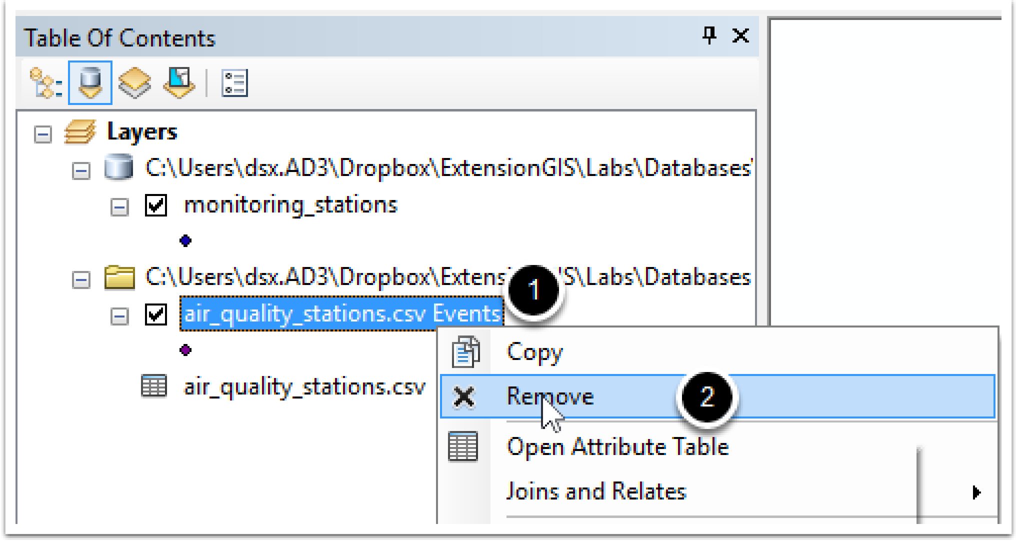
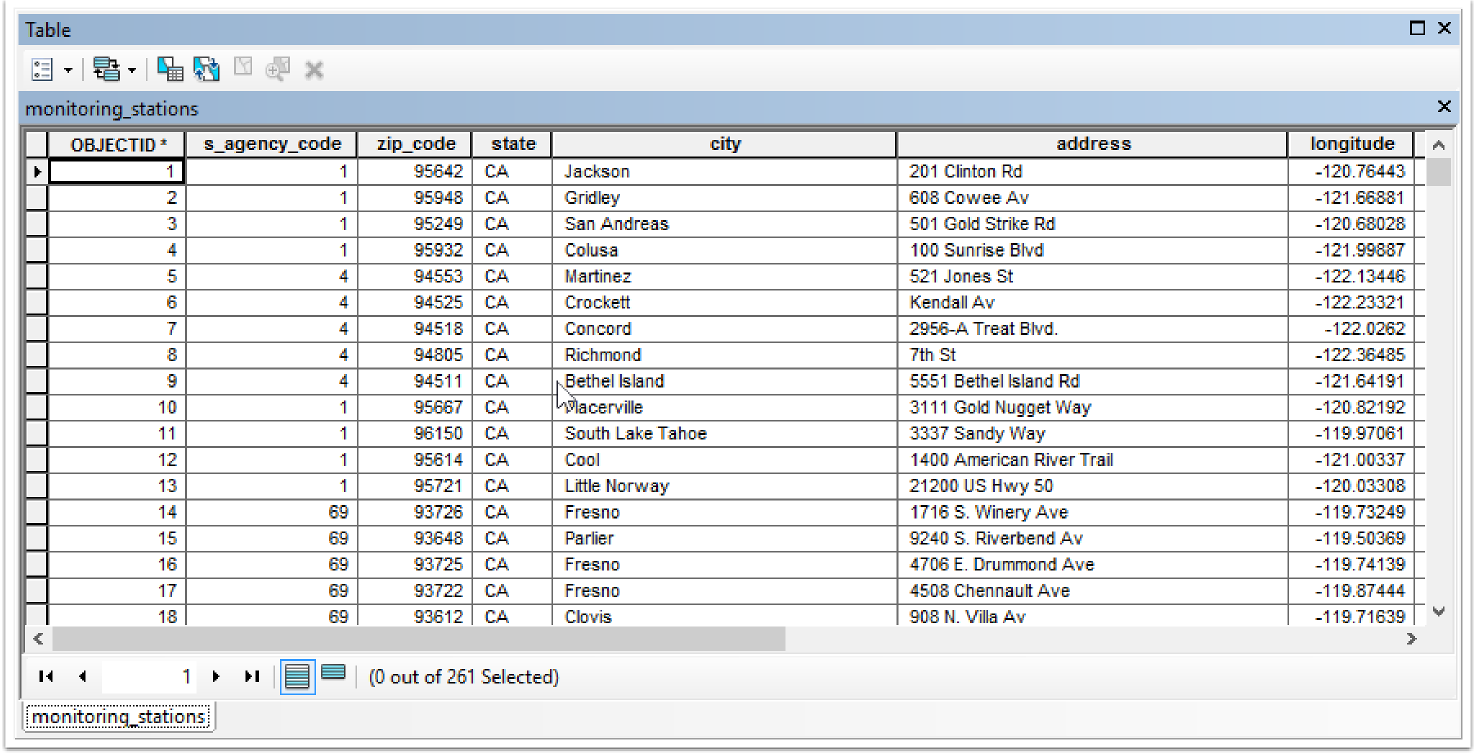


Table Of Contents
Layers
C:\Users\dsx.AD3\Dropbox\ExtensionGIS\Labs\Databases
☑ monitoring_stations
C:\Users\dsx.AD3\Dropbox\Extens S\Labs\Databases
☑ air_quality_stations.csv Events
air_quality_stations.csv
1
Copy
Remove
2
Open Attribute Table
Joins and Relates

3.8 Look, a primary key!

Open the attribute table for the *monitoring_stations* layer, and look again at the first field. When we exported the events layer as a true feature class, it added a primary key for us, named *OBJECTID.* If you'd like, you can try making a selection again, and notice that the layer is fully functional.

Bonus: Think about why a layer might have reduced functionality without an OBJECTID and discuss it in the course discussion forum.



OBJECTID *	s_agency_code	zip_code	state	city	address	longitude
1	1	95642	CA	Jackson	201 Clinton Rd	-120.76443
2	1	95948	CA	Gridley	608 Cowee Av	-121.66881
3	1	95249	CA	San Andreas	501 Gold Strike Rd	-120.68028
4	1	95932	CA	Colusa	100 Sunrise Blvd	-121.99887
5	4	94553	CA	Martinez	521 Jones St	-122.13446
6	4	94525	CA	Crockett	Kendall Av	-122.23321
7	4	94518	CA	Concord	2956-A Treat Blvd.	-122.0262
8	4	94805	CA	Richmond	7th St	-122.36485
9	4	94511	CA	Bethel Island	5551 Bethel Island Rd	-121.64191
10	1	95667	CA	Placerville	3111 Gold Nugget Way	-120.82192
11	1	96150	CA	South Lake Tahoe	3337 Sandy Way	-119.97061
12	1	95614	CA	Cool	1400 American River Trail	-121.00337
13	1	95721	CA	Little Norway	21200 US Hwy 50	-120.03308
14	69	93726	CA	Fresno	1716 S. Winery Ave	-119.73249
15	69	93648	CA	Parlier	9240 S. Riverbend Av	-119.50369
16	69	93725	CA	Fresno	4706 E. Drummond Ave	-119.74139
17	69	93722	CA	Fresno	4508 Chennault Ave	-119.87444
18	69	93612	CA	Clovis	908 N. Villa Av	-119.71639

1 (0 out of 261 Selected)

monitoring_stations

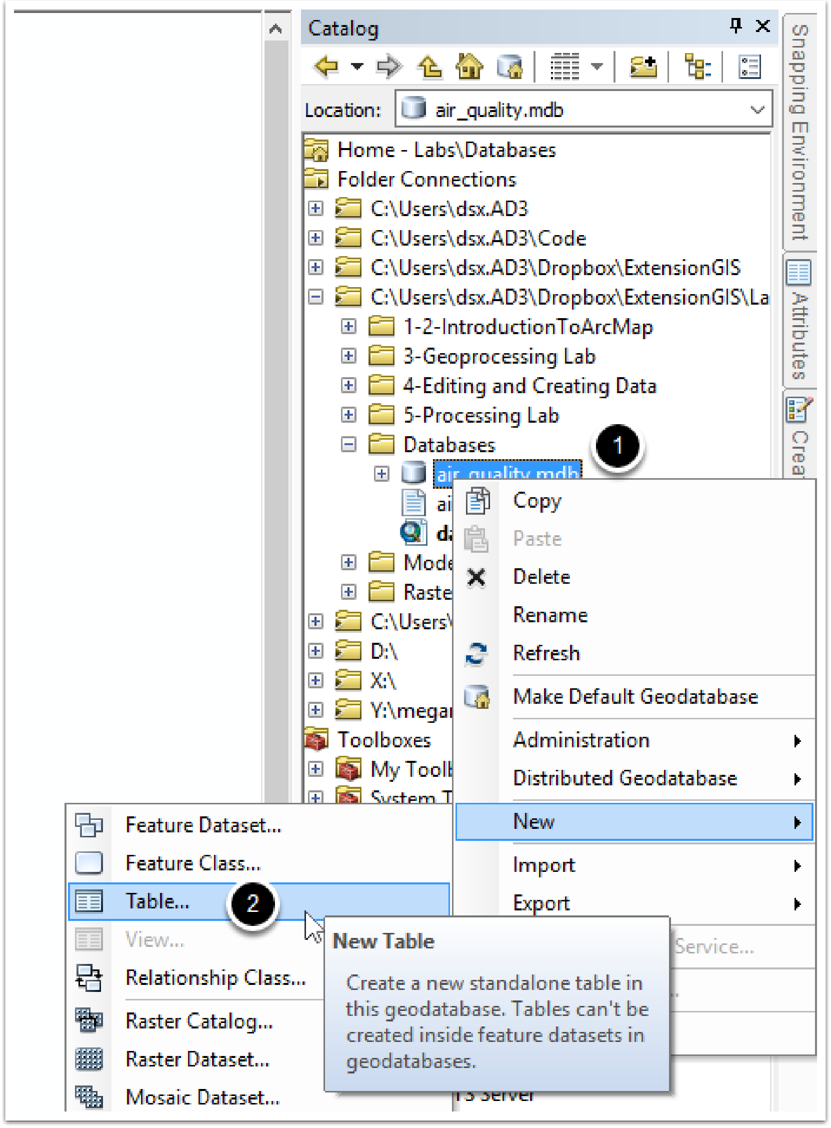
4. Create the monitoring data table

OK, so now we have a database that has our monitoring locations in it, but we don't have any data that has been collected at those locations!

Let's think about this for a moment. We could just add a field to our feature class indicating the last value for an air pollutant at that station, but what happens if we want to keep track of values over time? Or what if we want to track multiple pollutants? This can quickly get complicated depending on our design goals, but it's likely best if we create a separate table for our data for now

1. In the Catalog window, right click on *air_quality.mdb*

2. Go to *New* and click on *Table* to create a new table



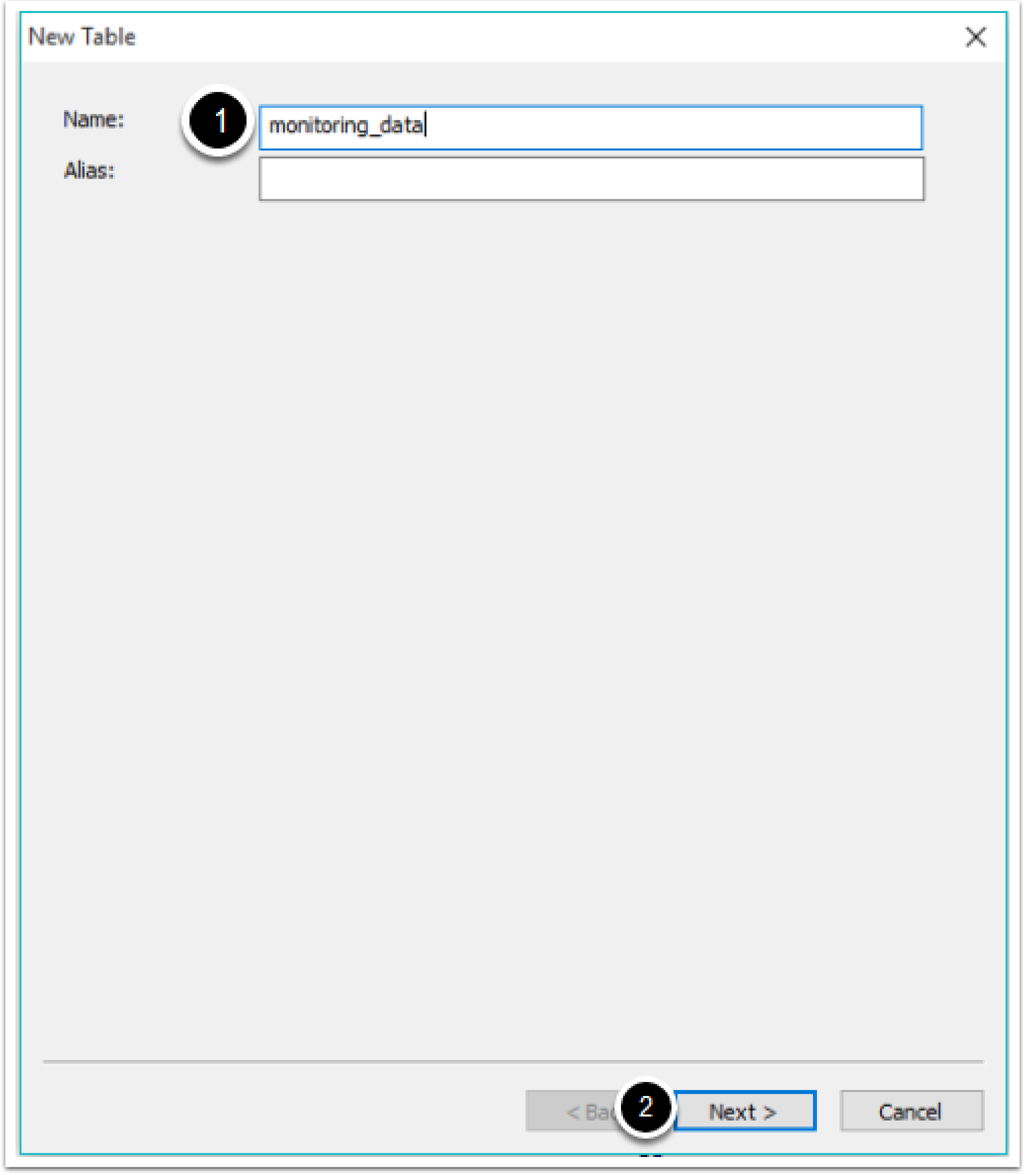
New Table

Create a new standalone table in this geodatabase. Tables can't be created inside feature datasets in geodatabases.

4.1 The new table dialog

The *New Table* dialog will come up to guide you through creating the table. It doesn't have a lot on the first page, but we do provide a name here.

1. Give it the name *monitoring_data*
2. Click *Next* to proceed



New Table

Name: monitoring_data

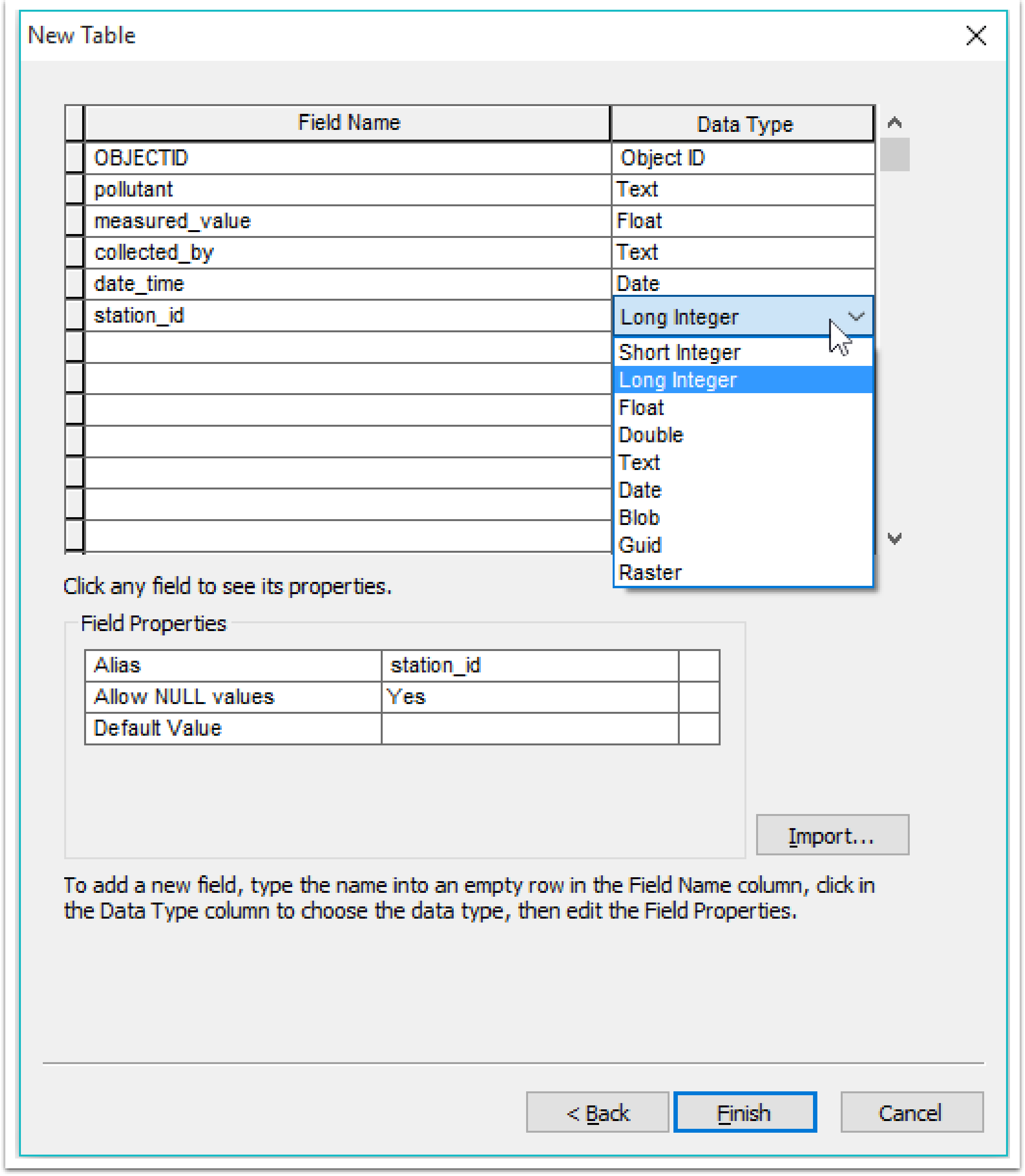
Alias:

4.2 Building out the table

On the next panel, we need to fill in the names and data types of all of the fields in the data table. Each record will need to track the measured value of a specific pollutant at a measuring station. It will also include the data and time of the measurement, and the name of the person who measured it. Leave the OBJECTID field as it is, and add the following fields:

1. *pollutant:* type *Text*. This is the field where we'll put in the name of the pollutant (for example, NO2, or O3)
2. *measured_value:* type *Float*. This is the actual concentration of the pollutant in the atmosphere at this location. We'll use the Float data type because even though many pollutants are expressed in whole Parts Per Million (PPM), it's possible that a new pollutant could be measured in a fractional manner.
3. *collected_by*: type *Text.* This field will hold the name of the person who collected the sample from the measurement device
4. *date_time:* type *Date.* This field will hold the date and time of the measurement in the record
5. *station_id*: type *Long Integer.* This field holds the foreign key that specifies which record in the *monitoring_stations* feature class this record applies to.

Once you have all of those entered, click Finish to continue



New Table

Field Name	Data Type
OBJECTID	Object ID
pollutant	Text
measured_value	Float
collected_by	Text
date_time	Date
station_id	Long Integer

Click any field to see its properties.

Field Properties

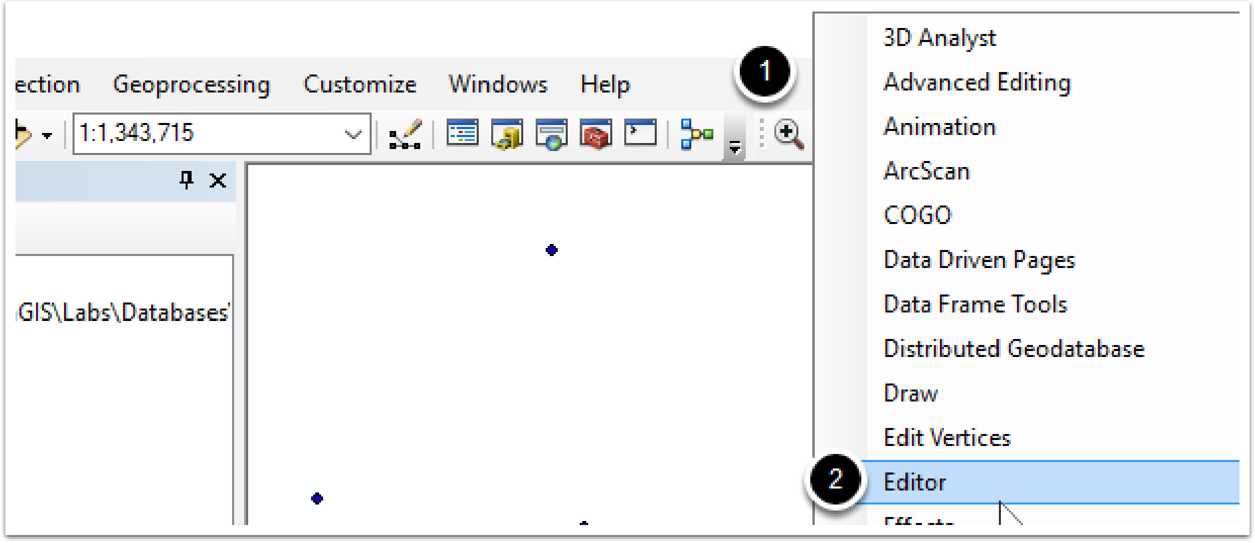
Alias	station_id
Allow NULL values	Yes
Default Value	

To add a new field, type the name into an empty row in the Field Name column, click in the Data Type column to choose the data type, then edit the Field Properties.

4.3 Adding records

Now, let's add records to our new table. To do so, we'll bring up the Editor tools.

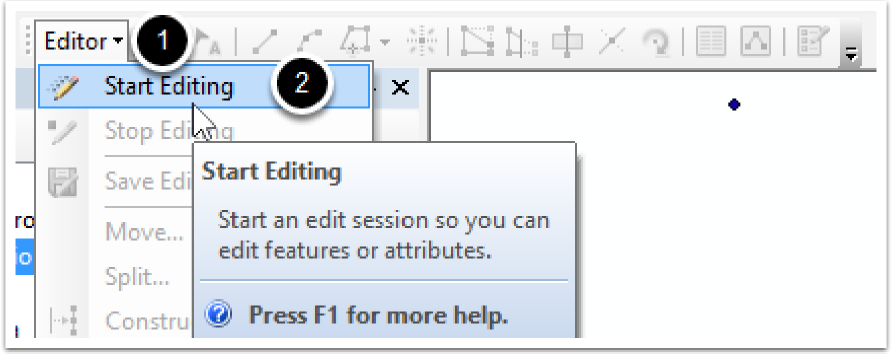
1. Right click in a blank spot on the toolbar somewhere, and
2. Click *Editor* in the menu that pops up to add the Editor toolbar



4.4 Start Editing

To being editing data, we need to start an *Edit Session,* which lets us roll back changes and decide when to save them as a group (so we can make all changes or no changes if we'd like, helping us ensure data integrity).

1. On the Editor toolbar, click *Editor*
2. Then click *Start Editing*



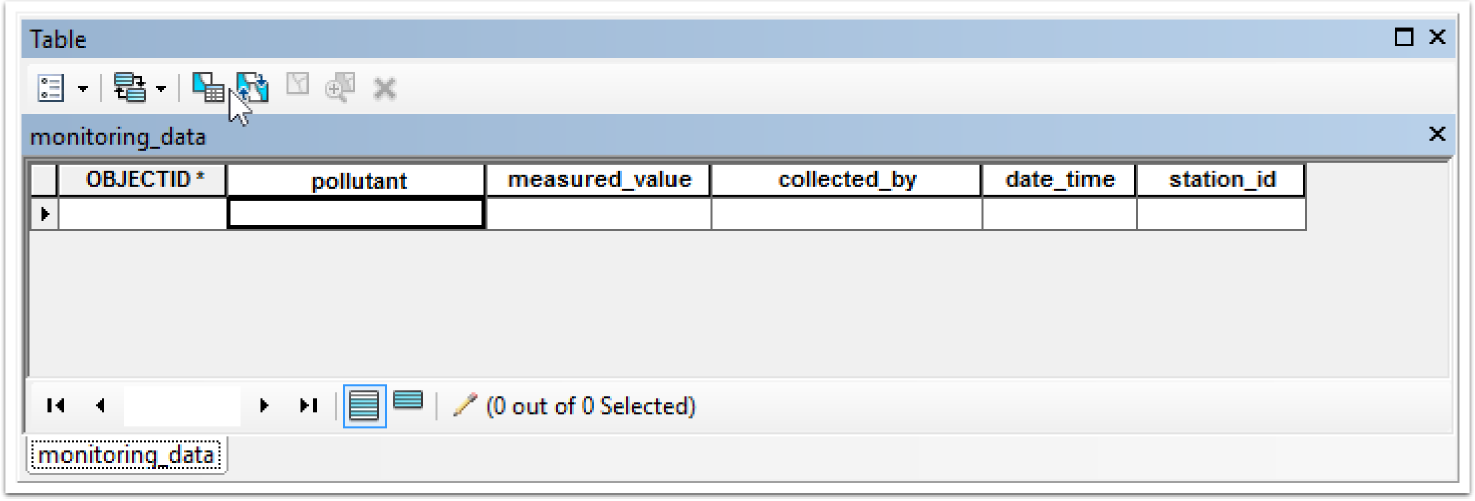
Start Editing

Start an edit session so you can edit features or attributes.

Press F1 for more help.

4.5 A blank record

Open the attribute table for *monitoring_data.* Our data table doesn't have any records in it yet, so let's change that now.



Table

monitoring_data

OBJECTID *	pollutant	measured_value	collected_by	date_time	station_id

(0 out of 0 Selected)

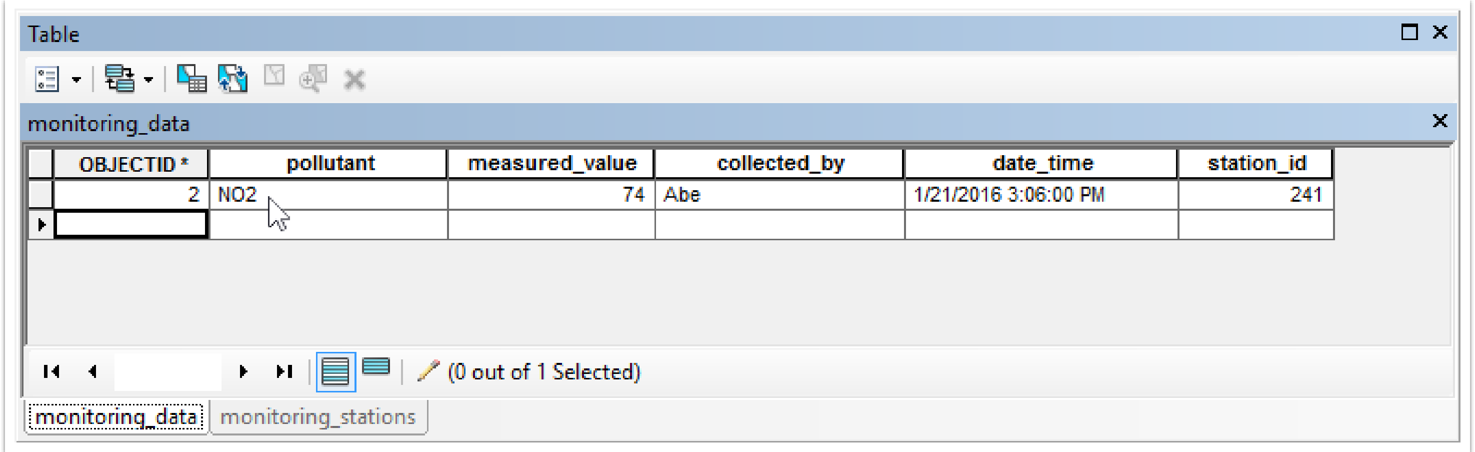
monitoring_data

4.6 Adding records

Skip the OBJECTID field. Since it's a primary key, let the database add it for you. As a result, your primary key (OBJECTID) values may differ from the screenshots. **This is OK.** Don't let it worry you. Enter the following values into the data table by double clicking into each box.

- For *pollutant,* enter *NO2*
- For *measured_value,* enter *74* (measured in parts per billion - where might you put units like that?)
- For *collected_by,* enter *Abe*
- For *date_time,* enter *1/21/2016 3:06:00 PM*
- For *station_id*, enter *241.*

Station 241 is a monitoring station in Long Beach, CA that this record applies to. As a bonus, can you find (via attribute table or select by attributes) monitoring station 241, if we know that *station_id* is a foreign key to *OBJECTID* in the *monitoring_stations* table?



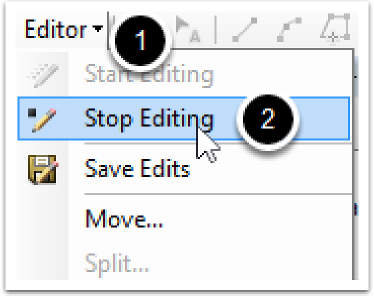
OBJECTID *	pollutant	measured_value	collected_by	date_time	station_id
2	NO2	74	Abe	1/21/2016 3:06:00 PM	241

(0 out of 1 Selected)

4.7 Stop Editing

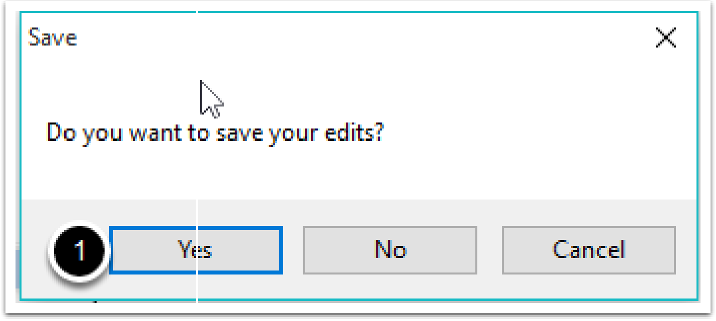
Once you've entered the information, we'll stop editing and save our edits.

1. On the Editor toolbar, click *Editor*
2. Then click *Stop Editing*



4.8 Save your edits

It will ask if you want to save your edits. Click *Yes* to save them.


Save

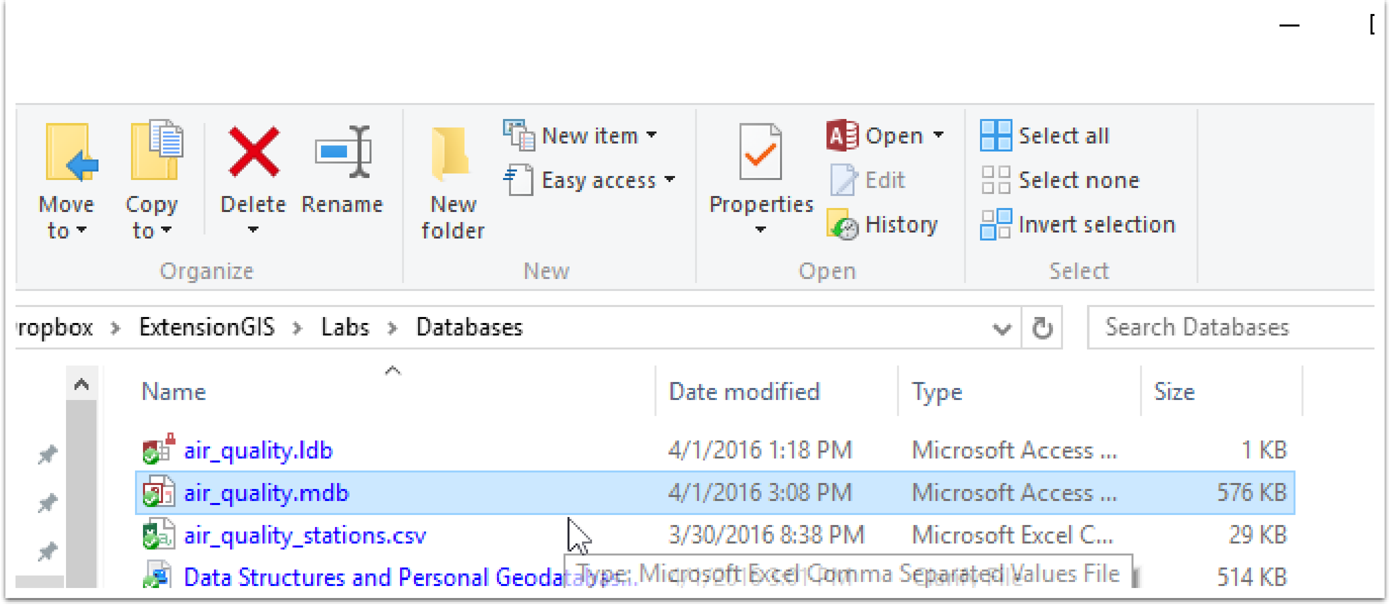
Do you want to save your edits?

5. Open it up in Access

Branching moment!

The next phase of this tutorial includes instructions that only work if you have Microsoft Access installed on your computer (this tutorial was made with the 2013 version, but similar steps work from Access 2007 on up, and personal geodatabases can be opened in Access 2003 and above). **If you don't have Microsoft Access installed, that's OK.** Instead, skip to step 5.11 and enter the records in the image there using the method you just learned (Start Editing and enter the records in the attribute table). If you want to try it in a separate database software package, you can try using a free software package like LibreOffice or OpenOffice - similar steps should exist in those packages. But don't worry about it if you don't want to. The important part is that you enter the data one way or another (again, skip to 5.11 if you don't have Microsoft Access), and this next phase is just to show you an alternative method.

If you do have Microsoft Access, then save your map document somewhere you can find it and close it out for now (so we won't experience file locking issues on the database). In Windows, find your *air_quality.mdb* and open it in Microsoft Access by double clicking on it.

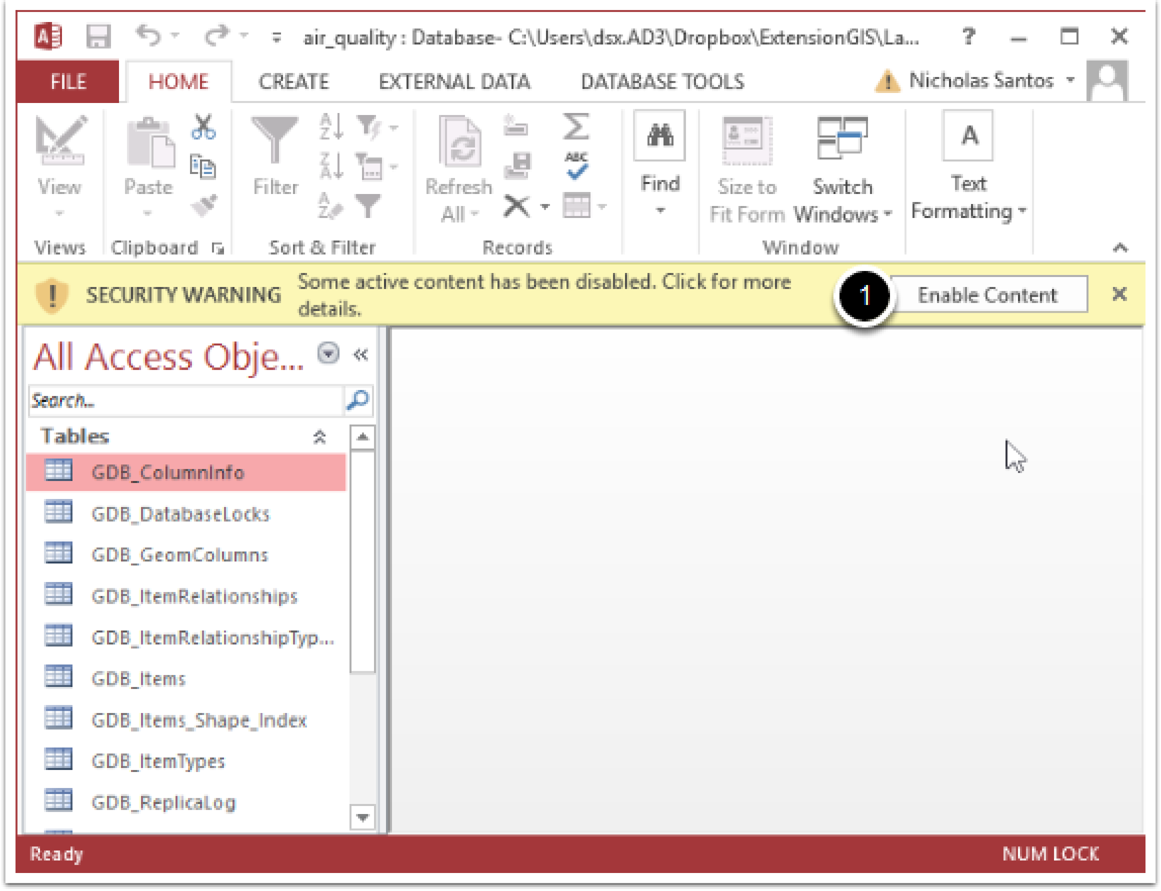


Name	Date modified	Type	Size
air_quality.ldb	4/1/2016 1:18 PM	Microsoft Access ...	1 KB
air_quality.mdb	4/1/2016 3:08 PM	Microsoft Access ...	576 KB
air_quality_stations.csv	3/30/2016 8:38 PM	Microsoft Excel C...	29 KB
Data Structures and Personal Geoda			514 KB

Type: Microsoft Excel Comma Separated Values File

5.1 Enable Content

Initially, for security reasons, Microsoft Access disables active content in your database - if you'd gotten from another source instead of making it yourself, it could include malicious code. In this case, it's fine, so click the *Enable Content* button (1).

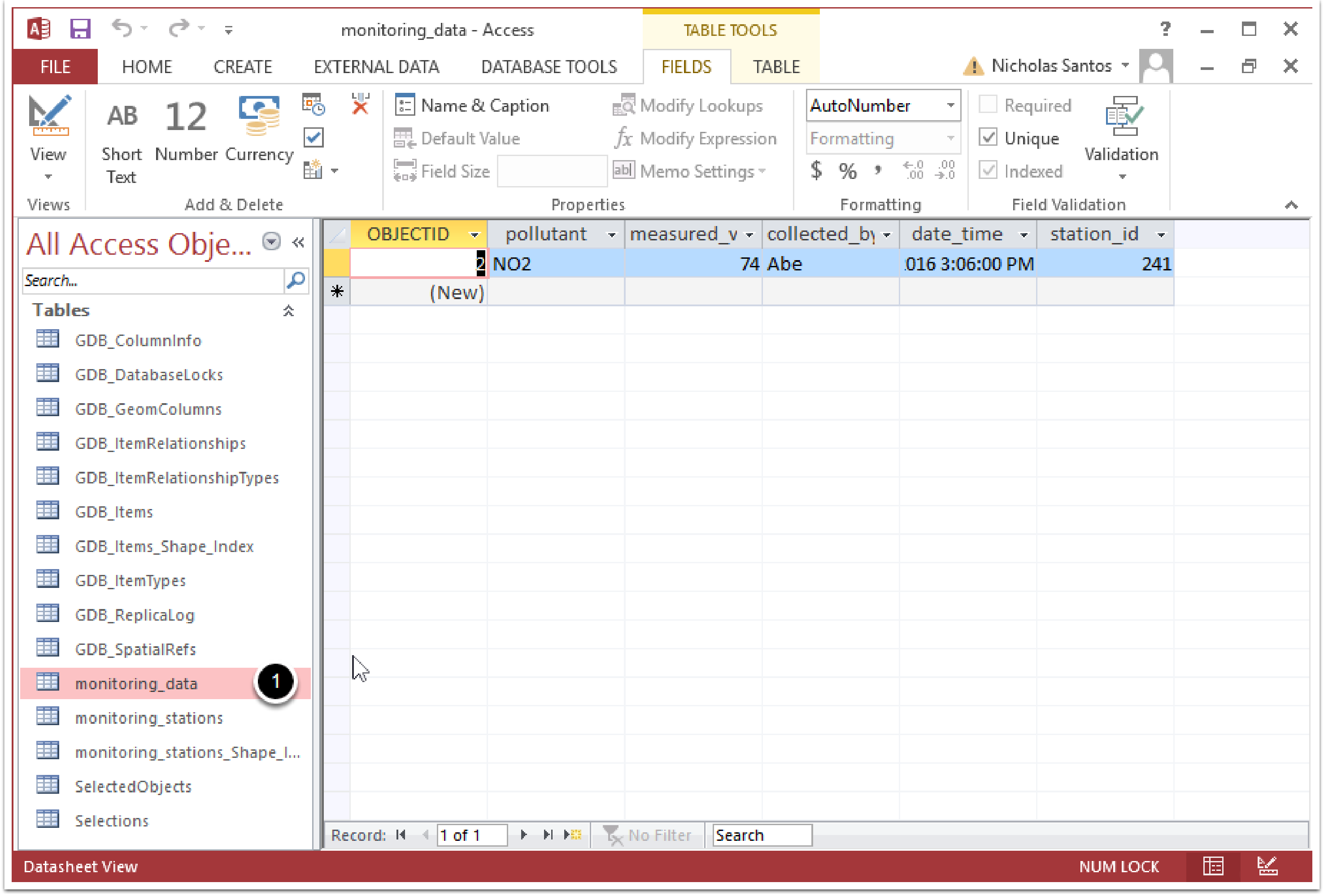

air_quality : Database- C:\Users\dsx.AD3\Dropbox\ExtensionGIS\La...

SECURITY WARNING Some active content has been disabled. Click for more details.

5.2 Open the table

Now, Microsoft Access is a database viewing and editing system, for self-contained databases. And the Personal Geodatabase uses the Microsoft Access format as its underlying storage system, so the software is well-suited to work with the data. For data tables we create, we can work with them inside of Microsoft Access and see the changes reflected in ArcGIS. Microsoft Access lets us run queries against the data, but also create forms and reports that can come in handy for larger projects.

For now, double click the *monitoring_data* item in the lefthand bar to open the data table up. You should see the record we created in ArcGIS show up in a data table view.

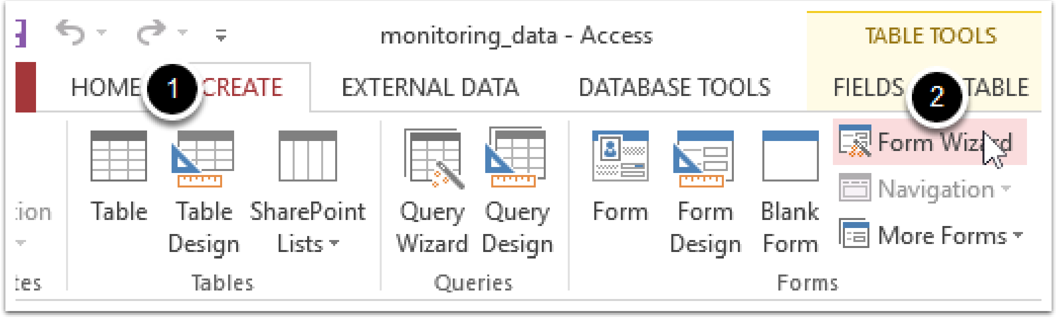


OBJECTID	pollutant	measured_v	collected_by	date_time	station_id
2	NO2	74	Abe	016 3:06:00 PM	241
(New)					

5.3 Building a form

Now, we could enter data directly on the form view we see now, but let's create a form to make it even nicer to enter data.

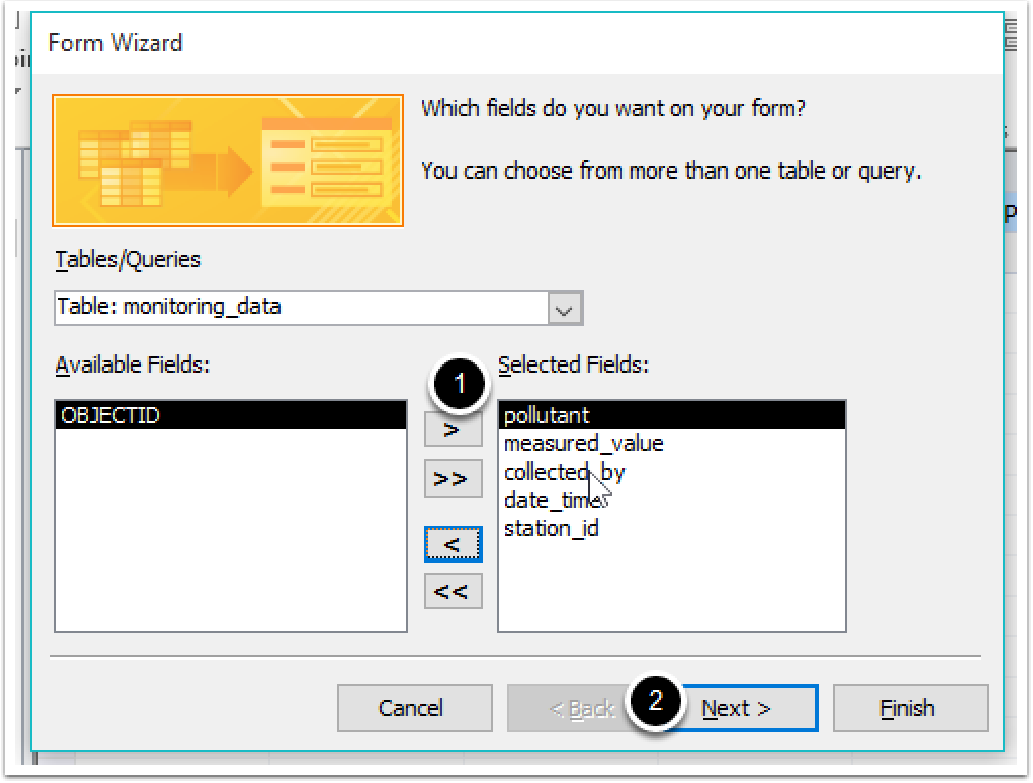
1. In the top toolbars, switch to the *Create* tab
2. Then in the *Forms* section, click *Form Wizard* to bring up the form creation tool.



5.4 Select the fields for the form

A dialog box will pop up, helping us create the form. To start out, we need to tell Access what fields to put in the form. In this case, we want to include all of the fields we created because we want someone to be able to enter those fields into the database, but we don't need the OBJECTID field since it will be automatically created.

1. Move all of the fields except OBJECTID to the right by selecting them and clicking the right arrow at the center to move them.



Form Wizard

Which fields do you want on your form?

You can choose from more than one table or query.

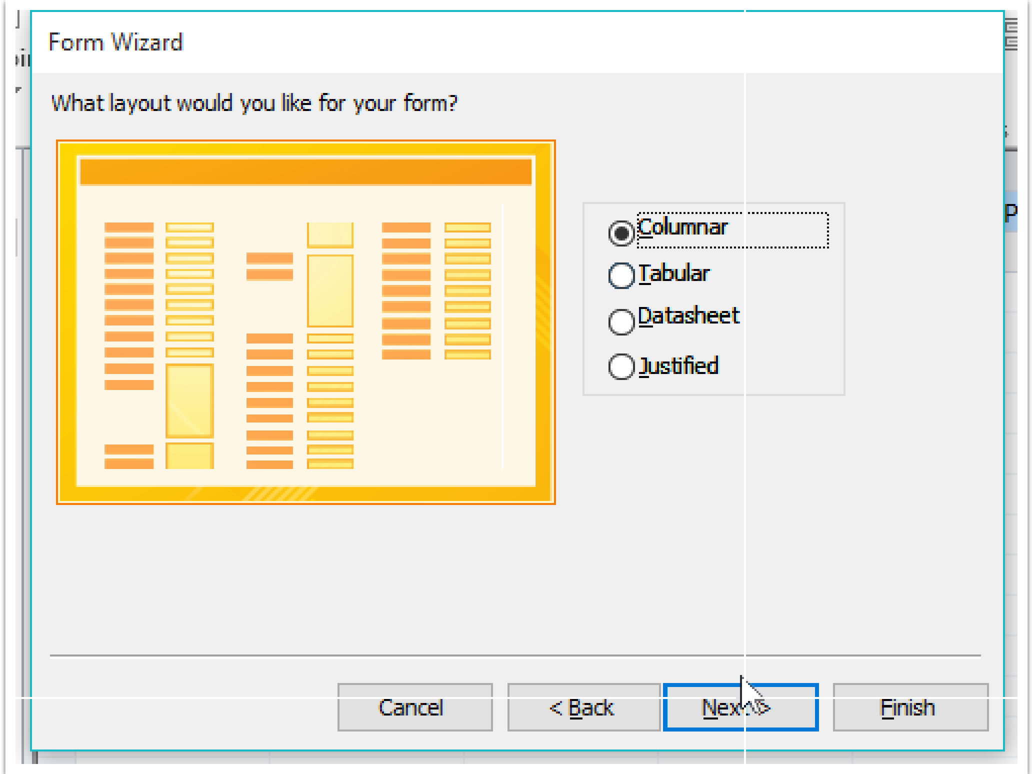
Tables/Queries: Table: monitoring_data

Available Fields: OBJECTID

Selected Fields: pollutant, measured_value, collected_by, date_time, station_id

5.5 Form Layout

Now, we can select the way that our form will look. We'll leave it as the default of "Columnar" - you can experiment with the others, but then your screen won't match the screenshots here. Click *Next* when ready.



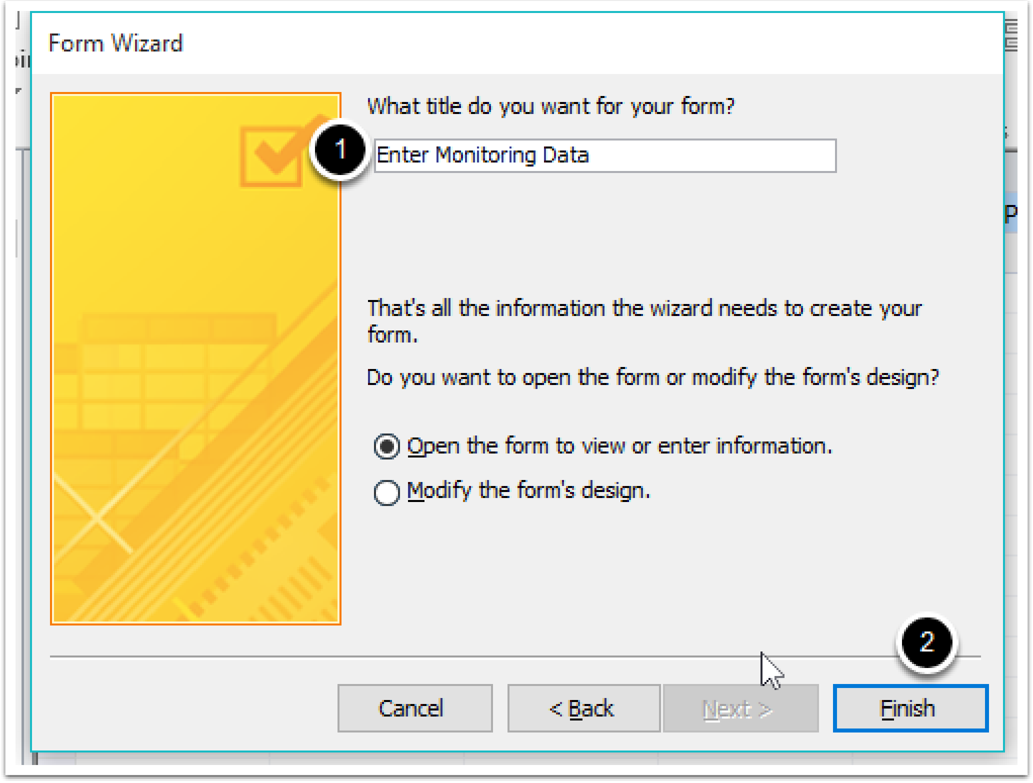
Form Wizard

What layout would you like for your form?

- ● Columnar
- ○ Tabular
- ○ Datasheet
- ○ Justified

5.6 Finalizing the Form

Now, we just need to finalize the form by giving it a name. Give it the name *Enter Monitoring Data* (1), then click *Finish*.



Form Wizard

What title do you want for your form?

1 Enter Monitoring Data

That's all the information the wizard needs to create your form.

Do you want to open the form or modify the form's design?

● Open the form to view or enter information.

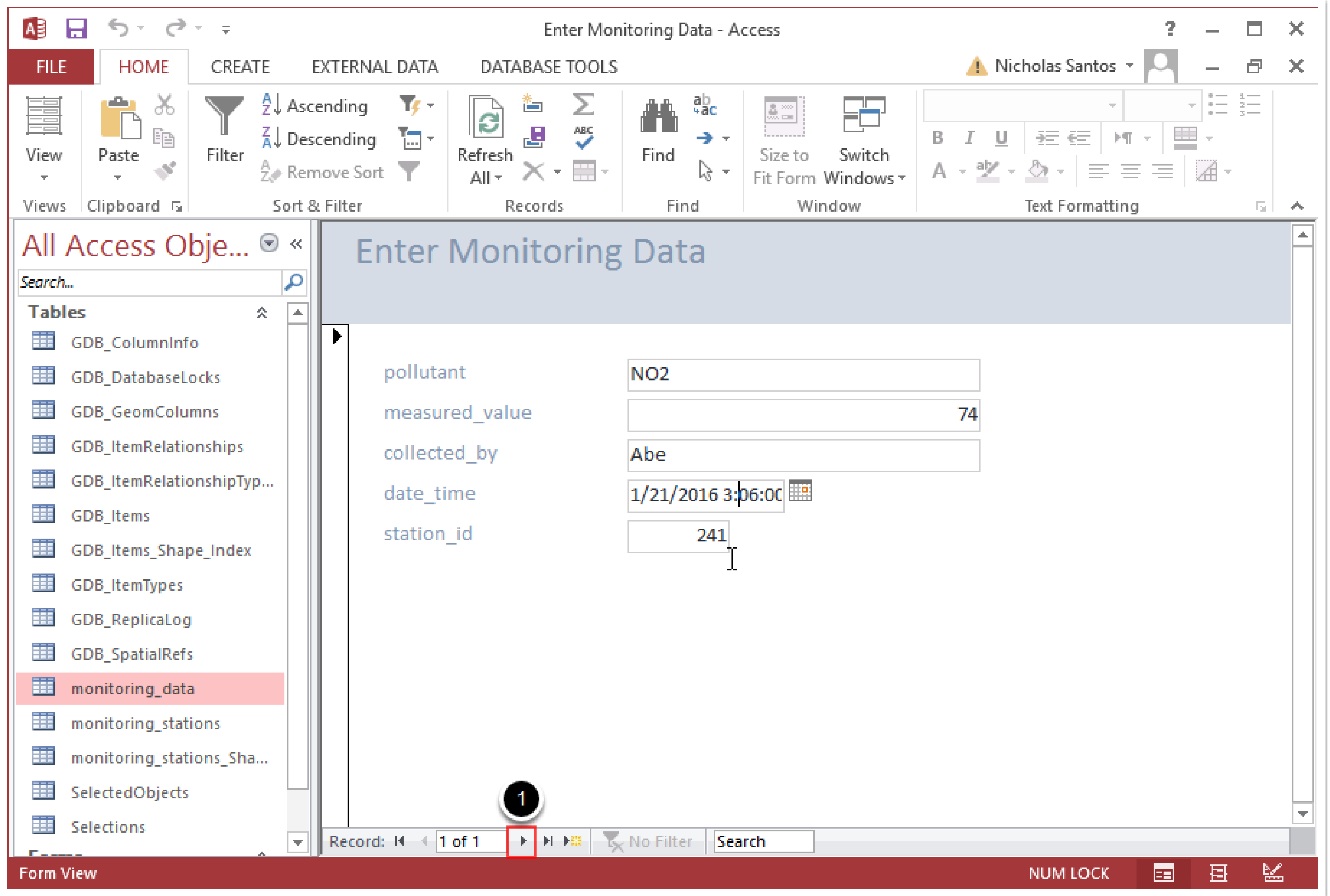
○ Modify the form's design.

5.7 Using the form

The form should open automatically. If it doesn't, then you can double click its name in the lefthand bar in the *Forms* section to open it up (and that's where you can access it in the future).

By default, the form will display the first record, that we already entered. If we wanted to, we could modify the data here, but we won't.

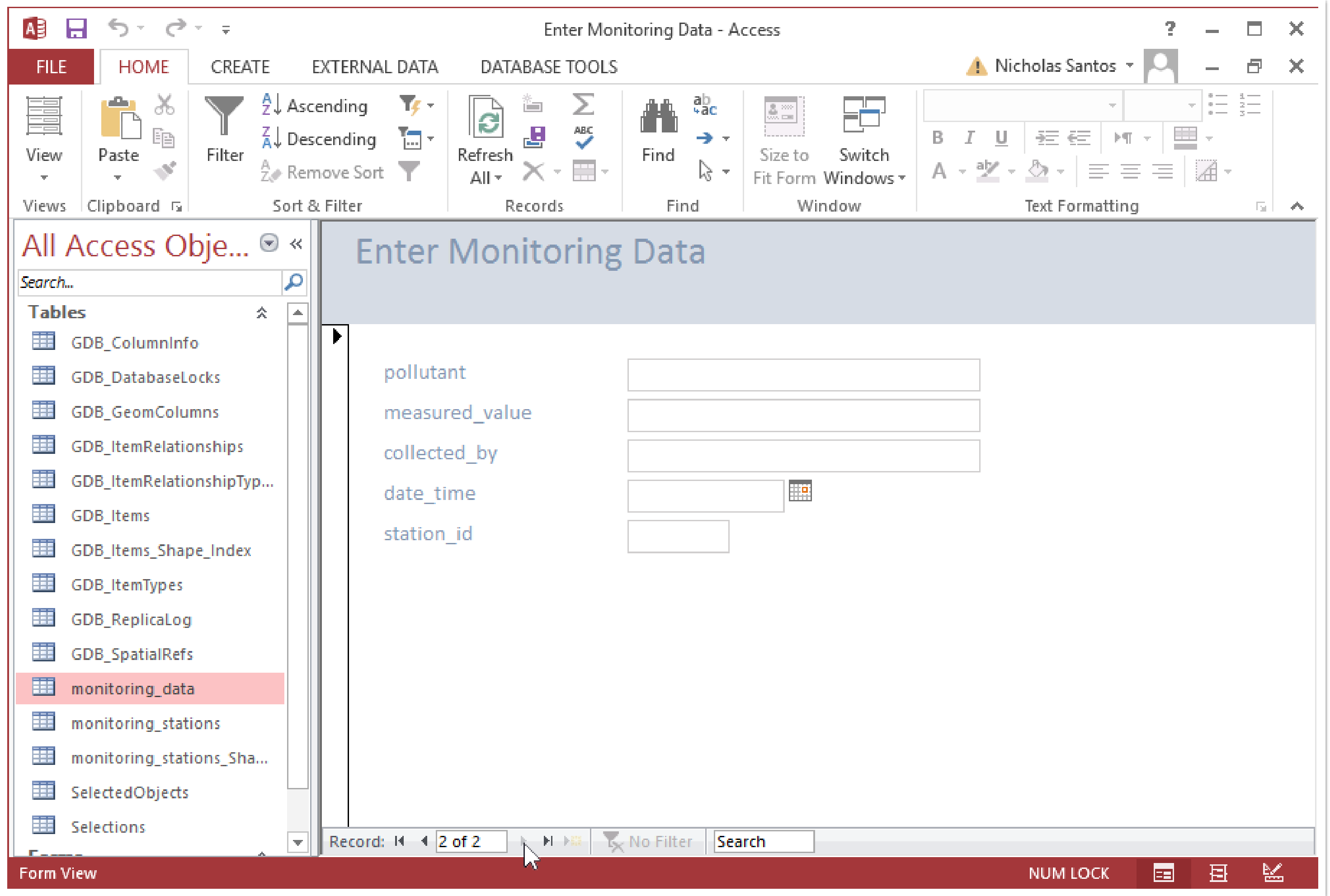
1. Let's enter a new record. At the bottom of the window, click the right arrow next to *1 of 1* to move to a new record



pollutant	NO2
measured_value	74
collected_by	Abe
date_time	1/21/2016 3:06:0C
station_id	241

5.8 The new form

By default, access just gives you a single new record, but note that we can't just create another new record without filling in this one first - we can't click the right arrow in this form yet.



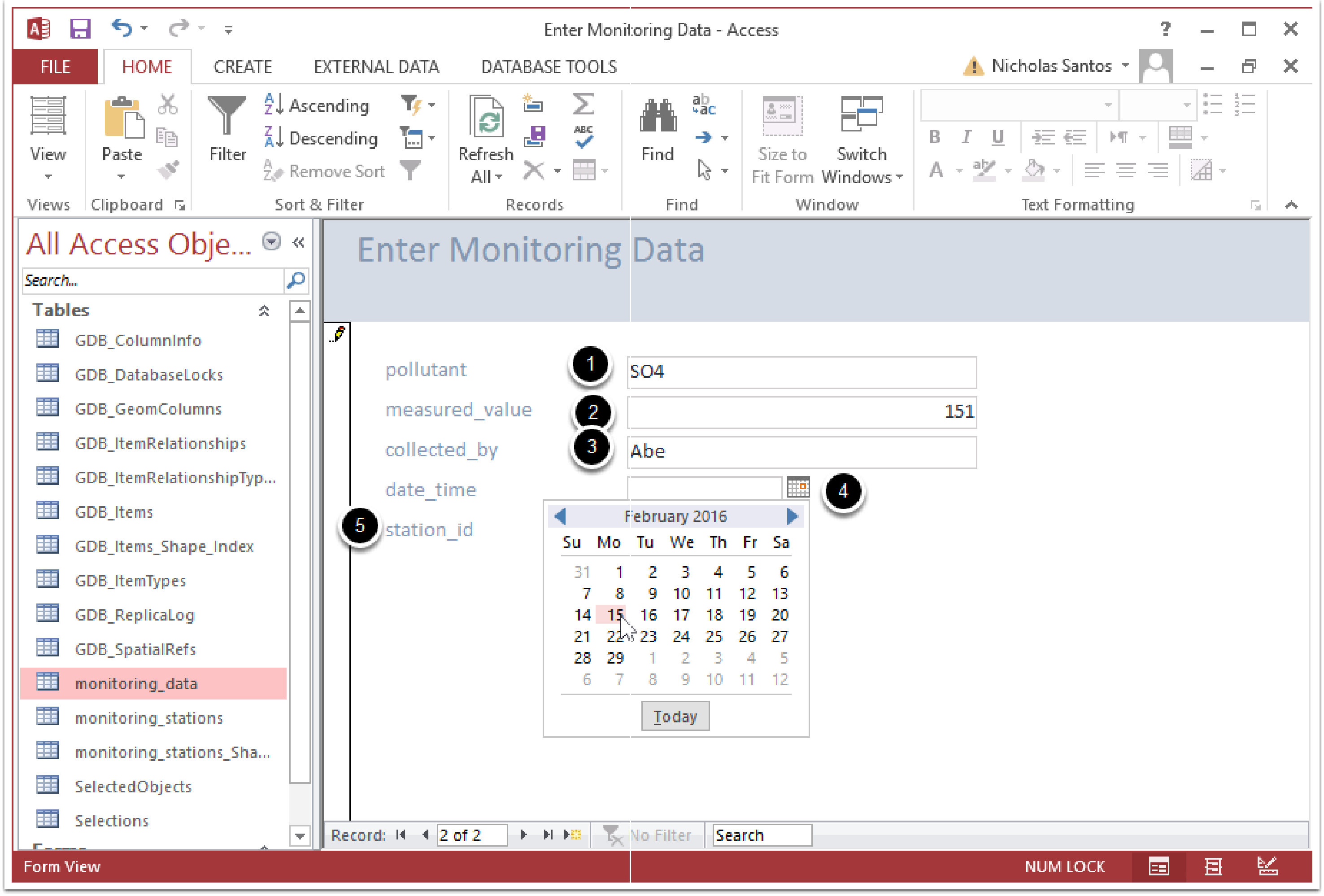
5.9 Entering a new record

Now, let's fill in the form. The data we're using is made just for this exercise, but in reality, this data would come from field measurements that someone had collected, and we'd use this form to enter and store all of our data into a single database for analysis.

1. For *pollutant,* enter SO4 - this will stand in as an abbreviation for sulfate
2. The measured value was 151 parts per billion, so enter that in the *measured_value* field

3. This measurement was also collected by a man named *Abe,* so enter that in the *collected_by* field
4. This value was collected on February 15th, 2016 - use the calendar picker to select that date fopr the *date_time* field.
5. The station is also station 241, so enter that for the *station_id* foreign key (not pictured fully).

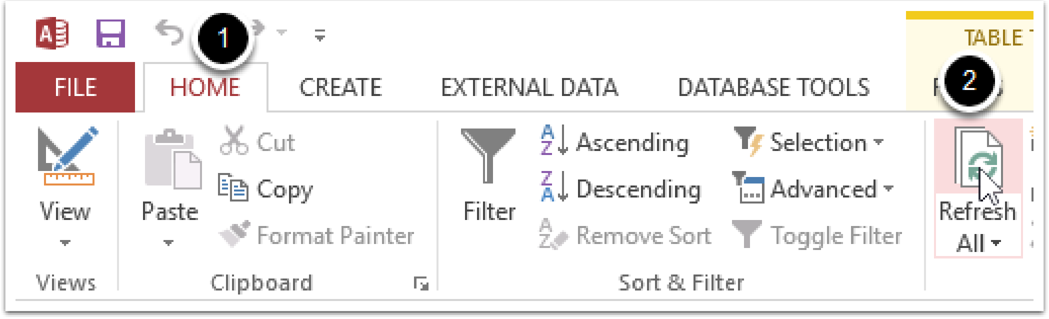
As a note, Access allows us to set up table lookups so that we don't have to enter numbers for our foreign keys. Instead, we can set up our forms, etc to perform lookups against another table, so that we could reference the station in the foreign key by name instead of by ID. This is much easier for most people - we won't cover how to do it in this tutorial, but know that it exists as a potential route for future learning!



pollutant: SO4
measured_value: 151
collected_by: Abe
date_time:
station_id:

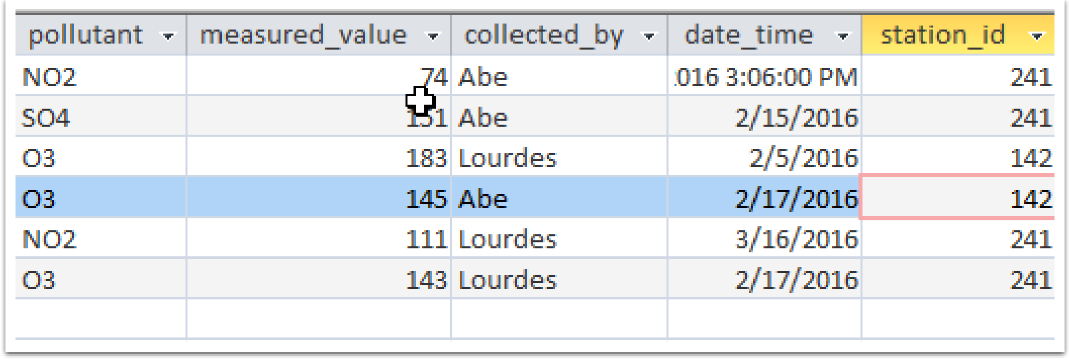
5.10 Switch back to the monitoring_data

Now, switch back to the *monitoring_data* table view (you can double click it again in the table of contents if you can't find it). You may not see your new record yet, but to get it to show up, click the *Refresh All (2)* button on the *Home tab* (1). Your new record should show up in the data table.



5.11 Enter the remaining rows

Now, using any data entry method that suits you - the form in Access, the datasheet in Access, or the Editor tools in ArcMap, enter the remaining records pictured below into your data table. You should have already entered the first two, but enter the remaining four!



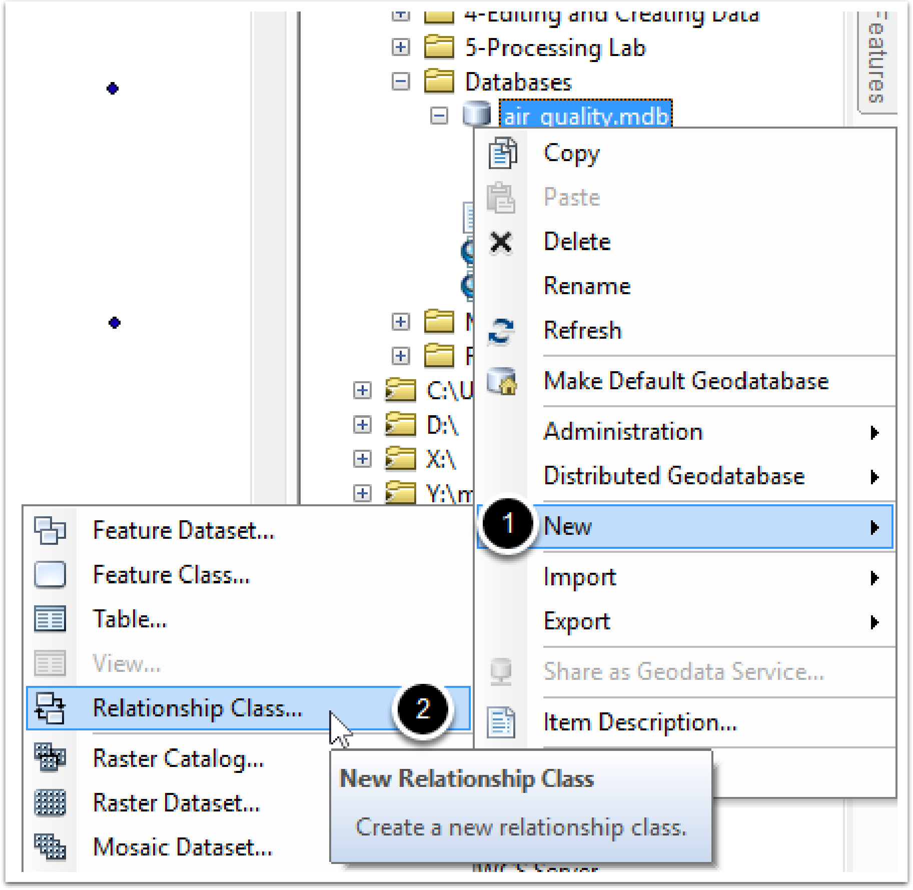
pollutant	measured_value	collected_by	date_time	station_id
NO2	74	Abe	016 3:06:00 PM	241
SO4	151	Abe	2/15/2016	241
O3	183	Lourdes	2/5/2016	142
O3	145	Abe	2/17/2016	142
NO2	111	Lourdes	3/16/2016	241
O3	143	Lourdes	2/17/2016	241

6. Relationships!

OK, now we have all of our data entered for now - at least until people collect more that you'd want to enter. Now, let's set up the relationships between our data that ArcGIS understands so that we can query and select our data. Switch back to your map in ArcMap if you aren't there already.

To do so, we'll use a *Relationship Class*, which formally defines the relationship between the feature class and the data table. This is similar to a table join that we're used to, but we can use it in cases like this where we have many records in our monitoring_data table that correspond to each feature in our monitoring_stations feature class.

1. In the Catalog window, right click on *air_quality.mdb*
2. Go down to *New* and click *Relationship Class*

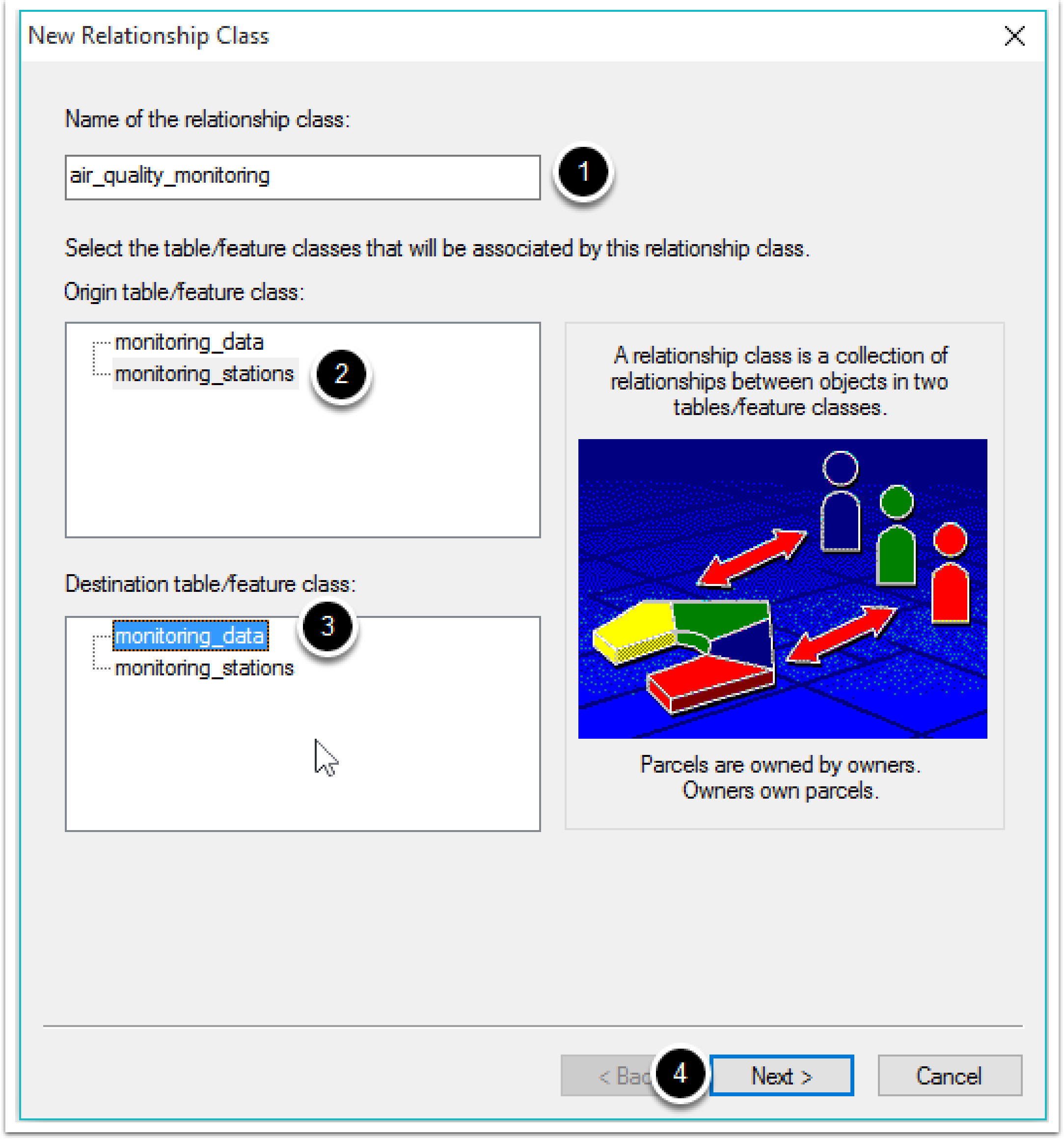


6.1 A confusing dialog!

OK, so now we start a multipage dialog to set up our data relationship.

1. First, let's give it a name - give it something descriptive that tells you at a glance what two tables/features it relates together. In this case I used *air_quality_monitoring* but you can choose something else if you want to.
2. For the *origin table/feature class*, choose the *monitoring_stations* feature class. This is slightly backward from how we viewed spatial joins, where the origin is the source of the records we want to append to the other table. If we'd gotten it wrong, we would have had trouble on a later page of the dialog, so we wouldn't have been able to proceed.

3. For the *Destination table/feature class,* choose the *monitoring_data* table. This is the source of the records we want to attach to the previous table.
4. Click *Next* to continue.



New Relationship Class

Name of the relationship class:

air_quality_monitoring

Select the table/feature classes that will be associated by this relationship class.

Origin table/feature class:

- monitoring_data
- monitoring_stations

Destination table/feature class:

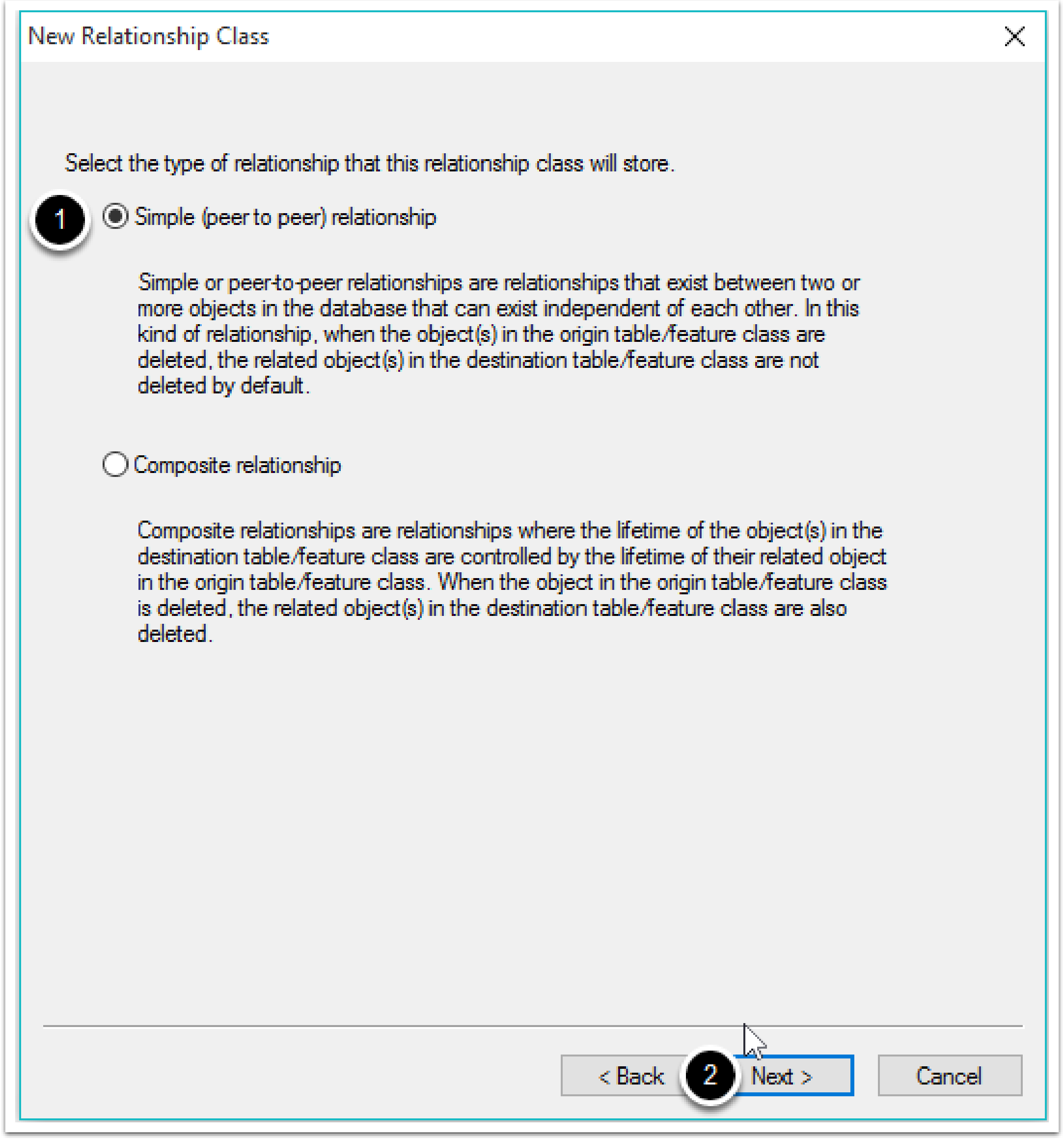
- monitoring_data
- monitoring_stations

A relationship class is a collection of relationships between objects in two tables/feature classes.

Parcels are owned by owners.
Owners own parcels.

6.2 Relationship type

This next page allows us to make a deep link between our two tables. Simple relationships mean that we have the relationship, but that changes (such as deletions) to records in one table don't affect the related records in the other table. In contrast, if we used the *Composite* relationship and deleted a monitoring station, any records in the *monitoring_data* table linked to that station (via the foreign key field) would also be deleted. How this behaves is up to you. In this case, choose the *Simple* relationship (1), then click *Next* (2).



New Relationship Class

Select the type of relationship that this relationship class will store.

1 ● Simple (peer to peer) relationship

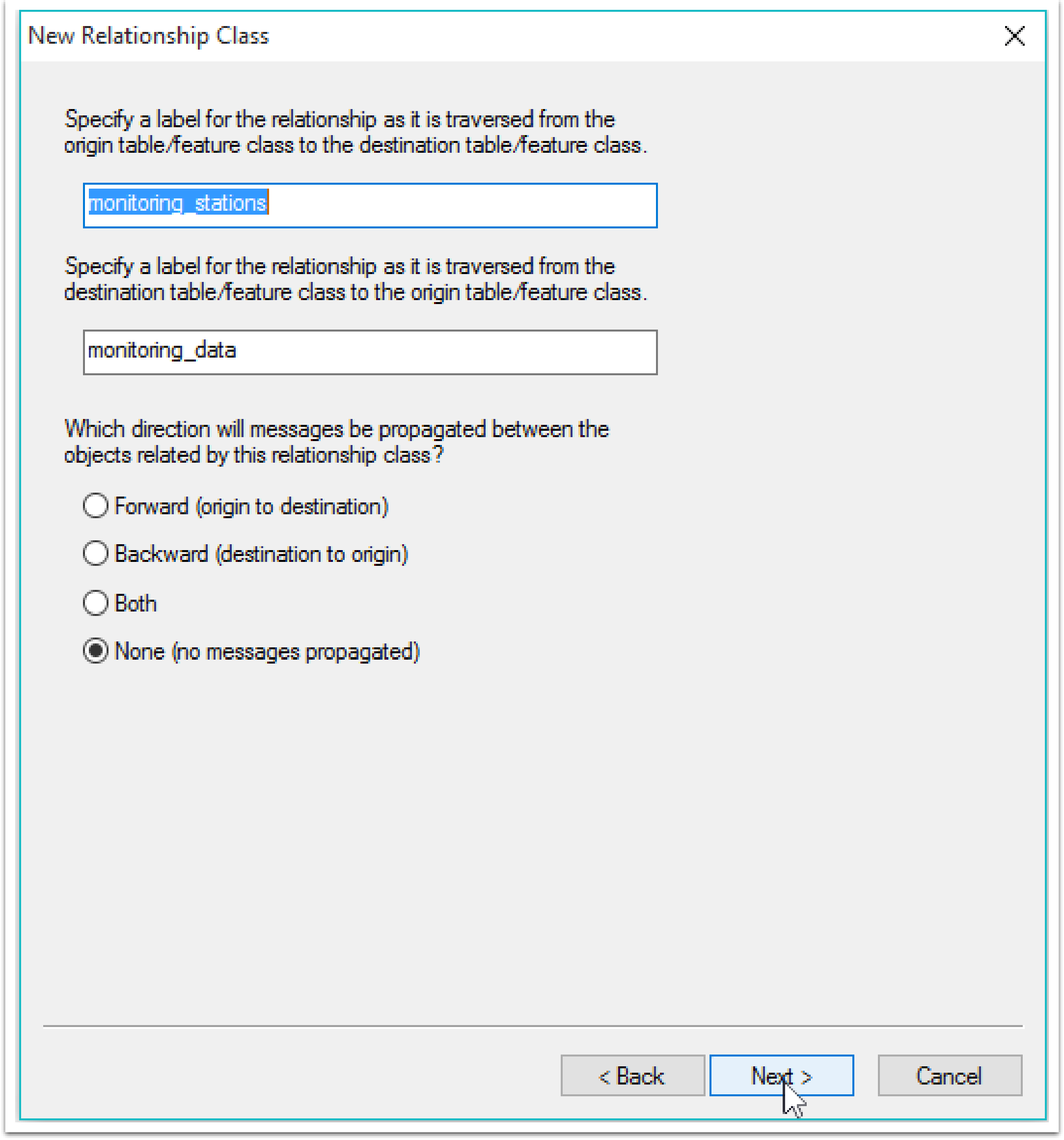
Simple or peer-to-peer relationships are relationships that exist between two or more objects in the database that can exist independent of each other. In this kind of relationship, when the object(s) in the origin table/feature class are deleted, the related object(s) in the destination table/feature class are not deleted by default.

○ Composite relationship

Composite relationships are relationships where the lifetime of the object(s) in the destination table/feature class are controlled by the lifetime of their related object in the origin table/feature class. When the object in the origin table/feature class is deleted, the related object(s) in the destination table/feature class are also deleted.

6.3 Labels

On this panel, we'll keep the defaults. It asks for names to reference each table by when viewing the other attribute table in ArcGIS (we'll use these in a moment). I left them as defaults, which are the names of the table - you can do what you like. Click *Next* to proceed.



New Relationship Class

Specify a label for the relationship as it is traversed from the origin table/feature class to the destination table/feature class.

monitoring_stations

Specify a label for the relationship as it is traversed from the destination table/feature class to the origin table/feature class.

monitoring_data

Which direction will messages be propagated between the objects related by this relationship class?

○ Forward (origin to destination)

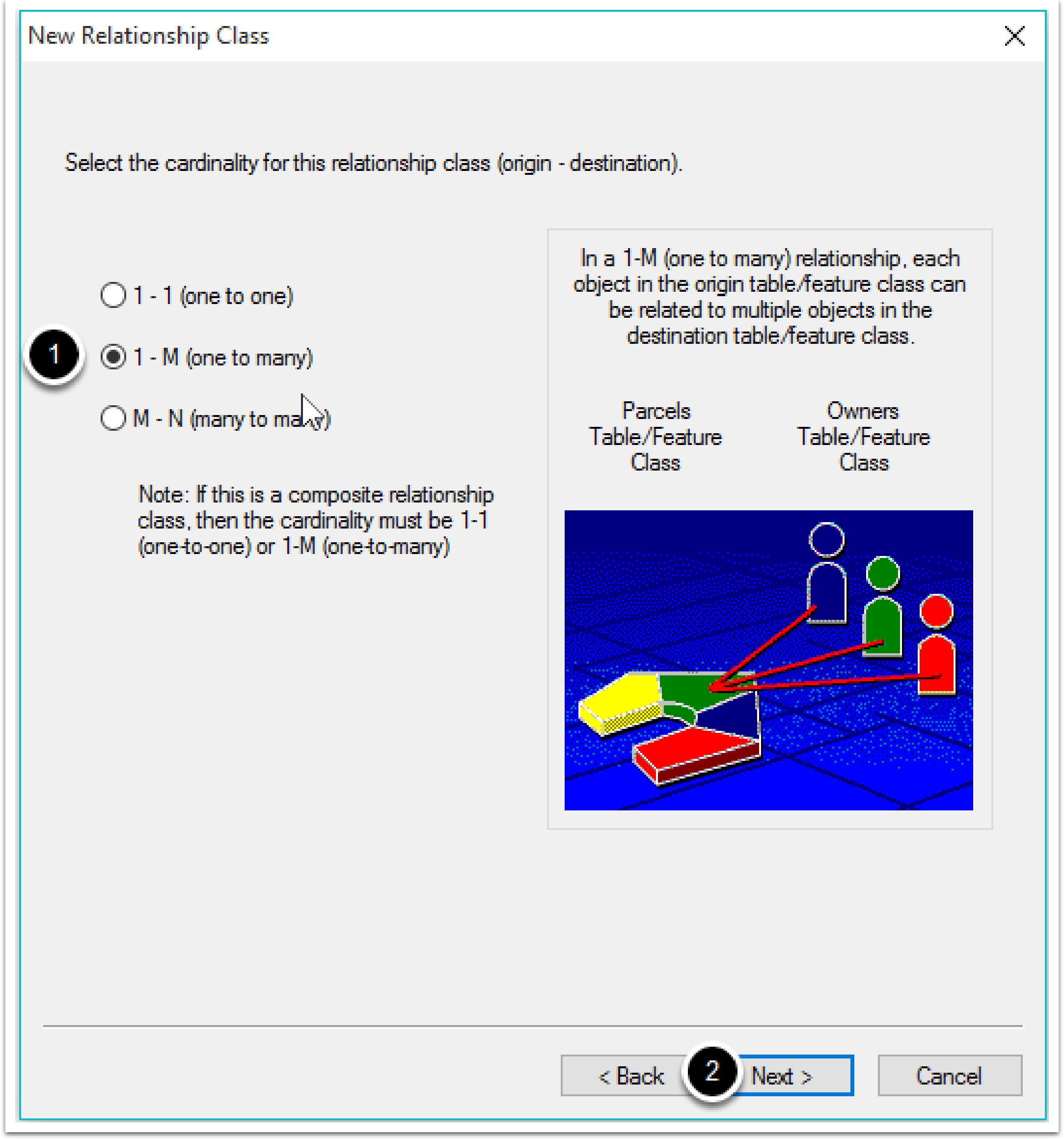
○ Backward (destination to origin)

○ Both

● None (no messages propagated)

6.4 Relationship Type

Now we need to select the type of relationship. If we had data that we could do a standard join with, it'd be a one to one relationship, but it's not. We have many records in our destination table (monitoring_data) to attach to a single record in our origin table (monitoring_stations). So, we have a one to many relationship where many records are attached to a single record!. Select *1-M (one to many)* (1) then click *Next* (2) to proceed


New Relationship Class

Select the cardinality for this relationship class (origin - destination).

○ 1 - 1 (one to one)

1 ● 1 - M (one to many)

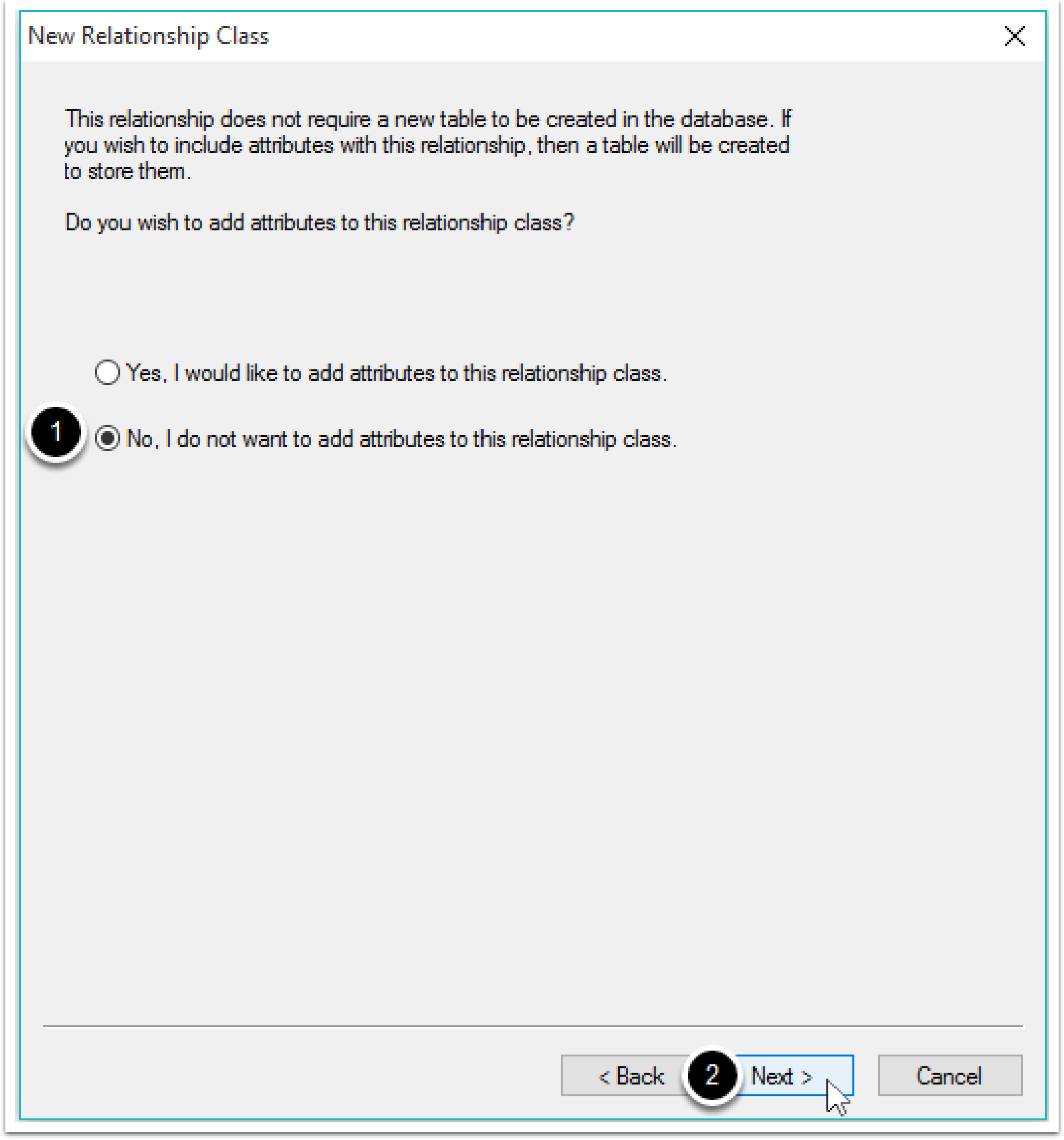
○ M - N (many to many)

Note: If this is a composite relationship class, then the cardinality must be 1-1 (one-to-one) or 1-M (one-to-many)

In a 1-M (one to many) relationship, each object in the origin table/feature class can be related to multiple objects in the destination table/feature class.

6.5 Adding Attributes

Our relationship classes can have its own attributes. If we were doing a many to many relationship that requires an additional table, we naturally might add additional attributes to that table. In this case, we can choose to create a new table to store attributes about the relationship, but we won't. Select *No* (1), then click *Next* (2) to continue.



New Relationship Class

This relationship does not require a new table to be created in the database. If you wish to include attributes with this relationship, then a table will be created to store them.

Do you wish to add attributes to this relationship class?

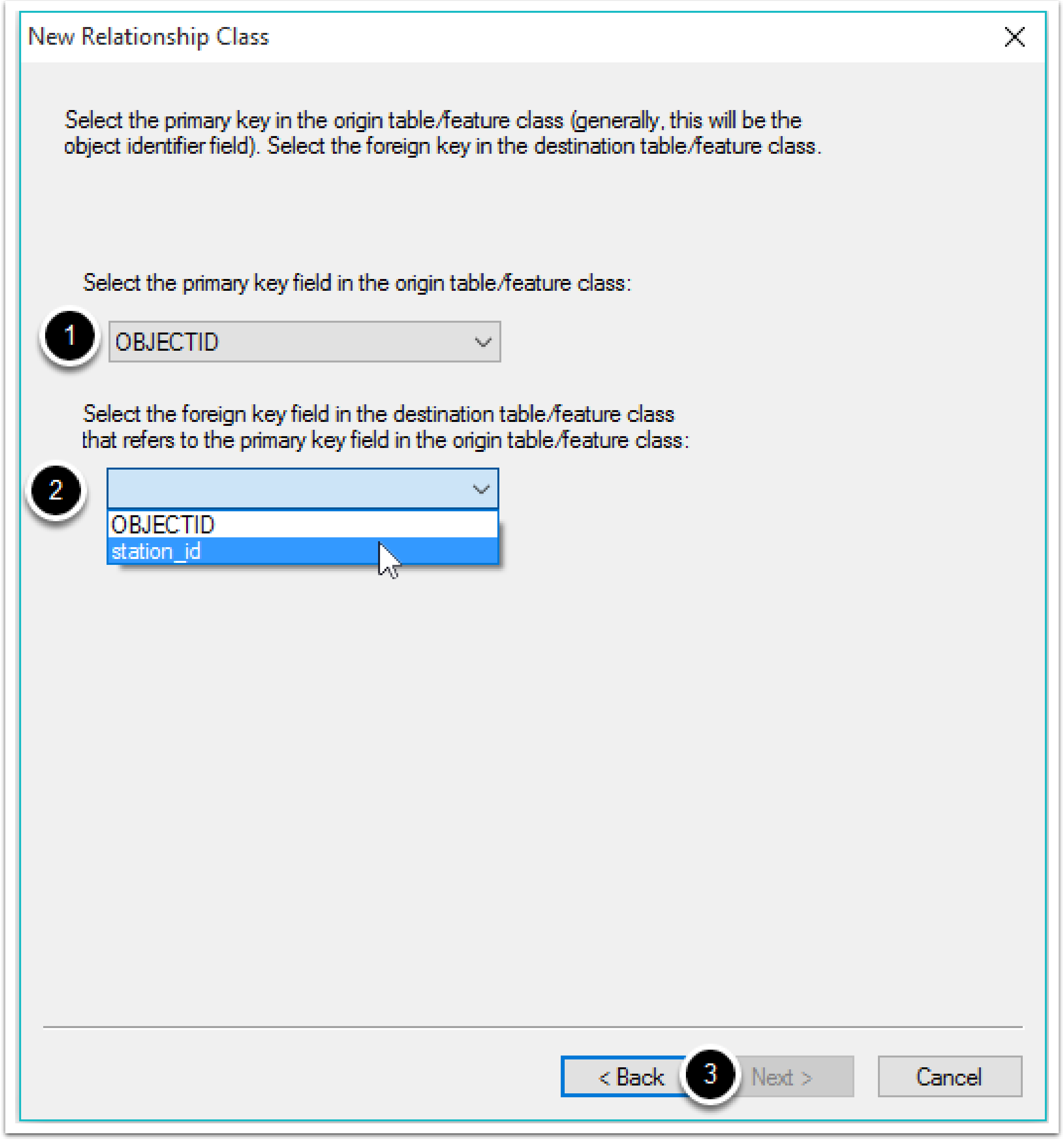
○ Yes, I would like to add attributes to this relationship class.

1 ● No, I do not want to add attributes to this relationship class.

6.6 Telling it how the tables relate

Now, the moment many of you have been waiting for. We need to tell it what the primary and foreign keys are so it can properly relate records in the tables.

1. Remember that the origin table is the *monitoring_stations* feature class, so our primary key is OBJECTID. Select that now
2. And for the foreign key field, we'll select the field that references the OBJECTEDID in *monitoring_stations* from the *monitoring_data* table. That field is the *station_id* field. Select that from the dropdown.
3. Click *Next* to proceed.



New Relationship Class

Select the primary key in the origin table/feature class (generally, this will be the object identifier field). Select the foreign key in the destination table/feature class.

Select the primary key field in the origin table/feature class:

1 OBJECTID

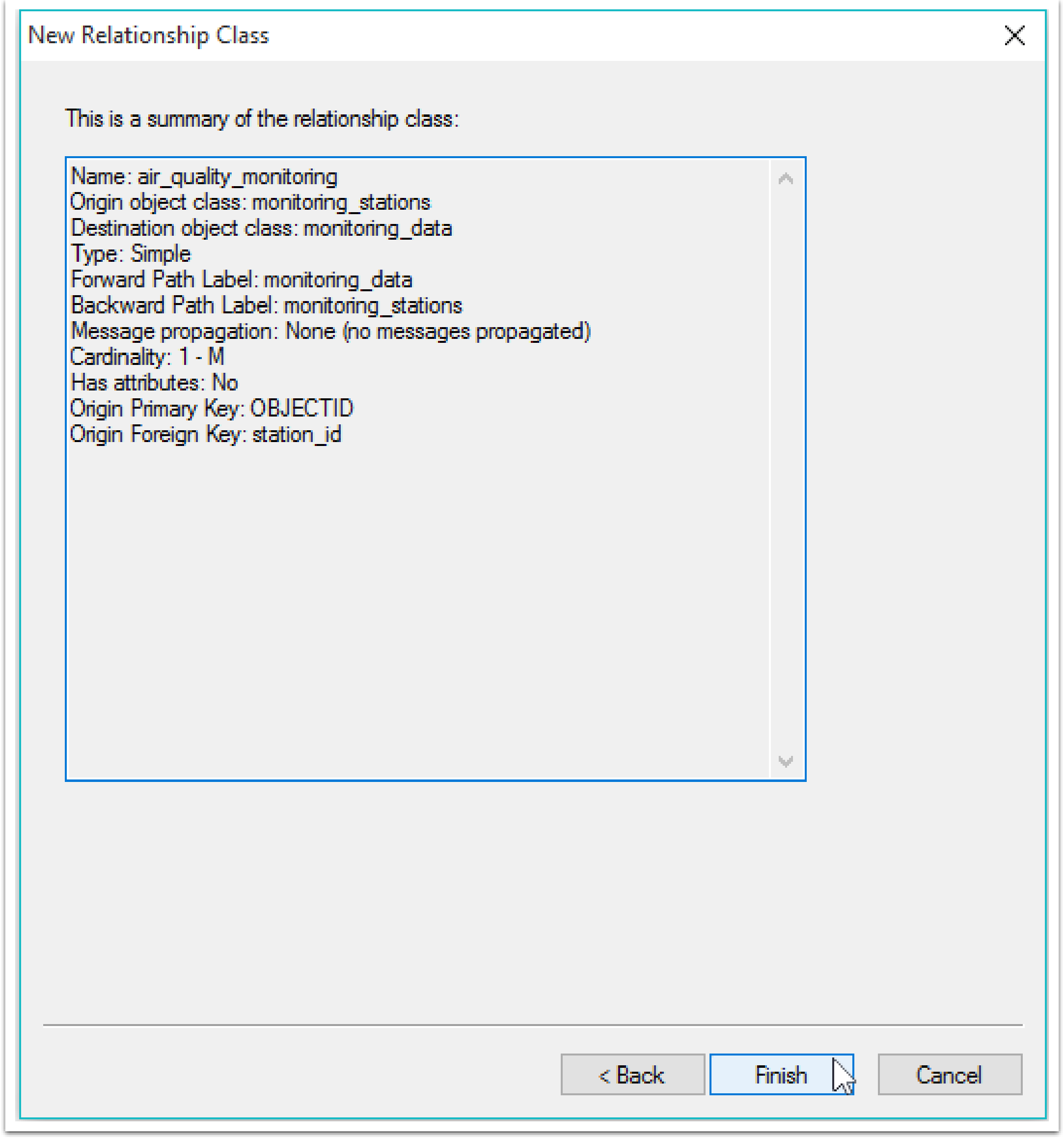
Select the foreign key field in the destination table/feature class that refers to the primary key field in the origin table/feature class:

2

OBJECTID
station_id

6.7 Summary

Look over the summary that comes up to make sure everything looks good, then click *Finish* to create the relationship class.

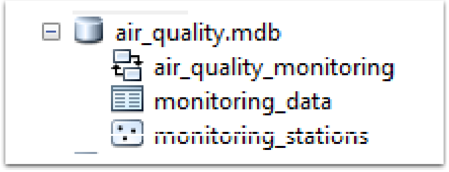


This is a summary of the relationship class:

Name: air_quality_monitoring
Origin object class: monitoring_stations
Destination object class: monitoring_data
Type: Simple
Forward Path Label: monitoring_data
Backward Path Label: monitoring_stations
Message propagation: None (no messages propagated)
Cardinality: 1 - M
Has attributes: No
Origin Primary Key: OBJECTID
Origin Foreign Key: station_id

6.8 A relationship class is created!

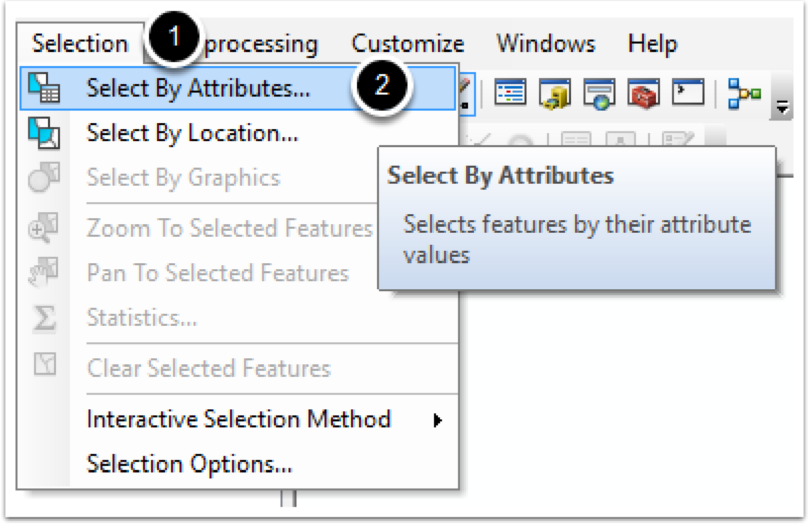
Now, if we look in our geodatabase, we'll see the relationship class exists as an object there too.



air_quality.mdb
air_quality_monitoring
monitoring_data
monitoring_stations

7. Looking up our data

The relationship class allows us to view the related records automatically when viewing the features in our attribute table, but to make this easier, let's use select by attributes to find the two stations we entered data for. If you need a refresher, *Select By Attributes* (2) can be found under the *Selection* menu in ArcMap (1).



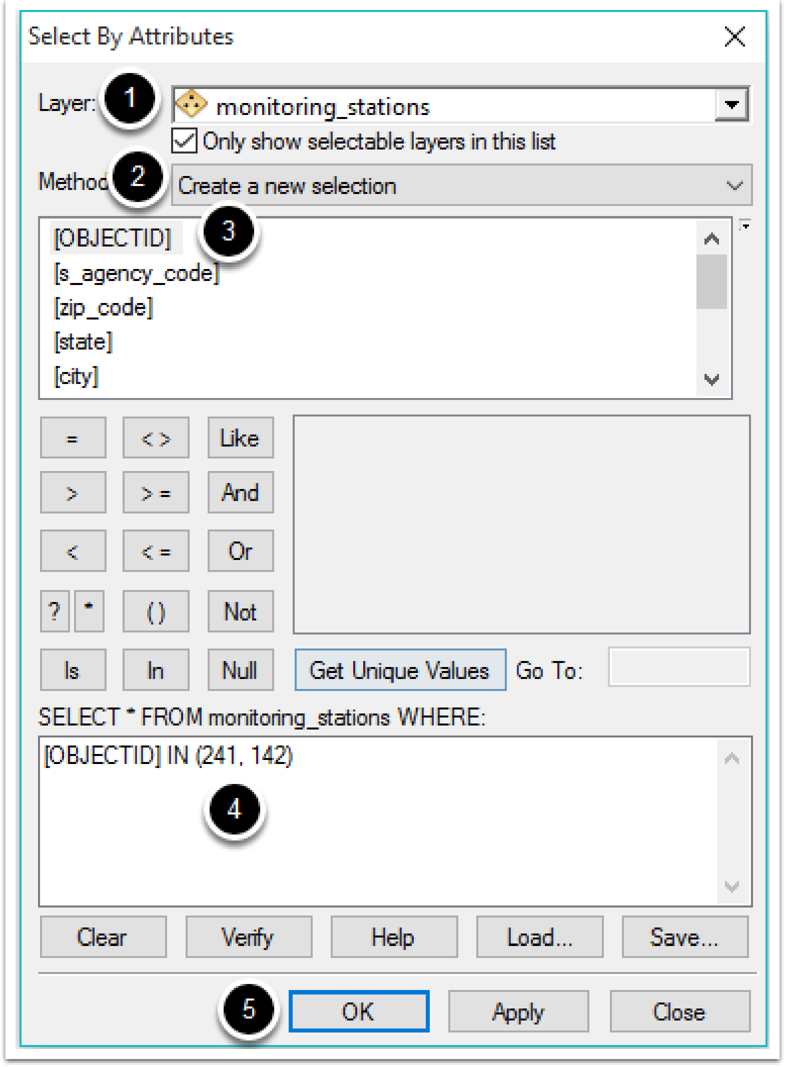
7.1 Writing the query

Let's find our two stations by ID. We are looking for stations with the OBJECTID of 241 and 142. We can write this query a few ways, and we'll cover both here.

First, the logic required to select multiple stations by OBJECTID is the *OR* logic - we want stations with the ID 241 OR 142. So, that query would be something like *[OBJECTID] = 241 OR [OBJECTID] = 142.* We use the square brackets because this is a personal geodatabase, but if we forgot that, we could see them used around fields in the double click editor for select by attributes.

Let's write this query with an alternative operator though. We can condense this query a bit - it might not matter much when we have only two stations to select, but if we had many more, like dozens, writing a long query with many *OR* clauses would be cumbersome. So we can collapse them using the *IN* operator.

1. First, make sure that the *monitoring_stations* layer is the layer we're making the selection on.
2. Confirm also that you are creating a new selection
3. Double click *[OBJECTID]* to add it to the query string
4. Then, in the bottom box, after *[OBJECTID],* type *IN (241, 142)* - see graphic below for the full string to type. The *IN* operator lets us provide a set of values to match against, similar to the OR operator, but in a smaller space. Interpreting this query, it matches any record where the value for OBJECTID is a number in the set of numbers we provide in parentheses. So, if OBJECTID is 241 or 142, this query will match the record.
5. Click OK to run the query



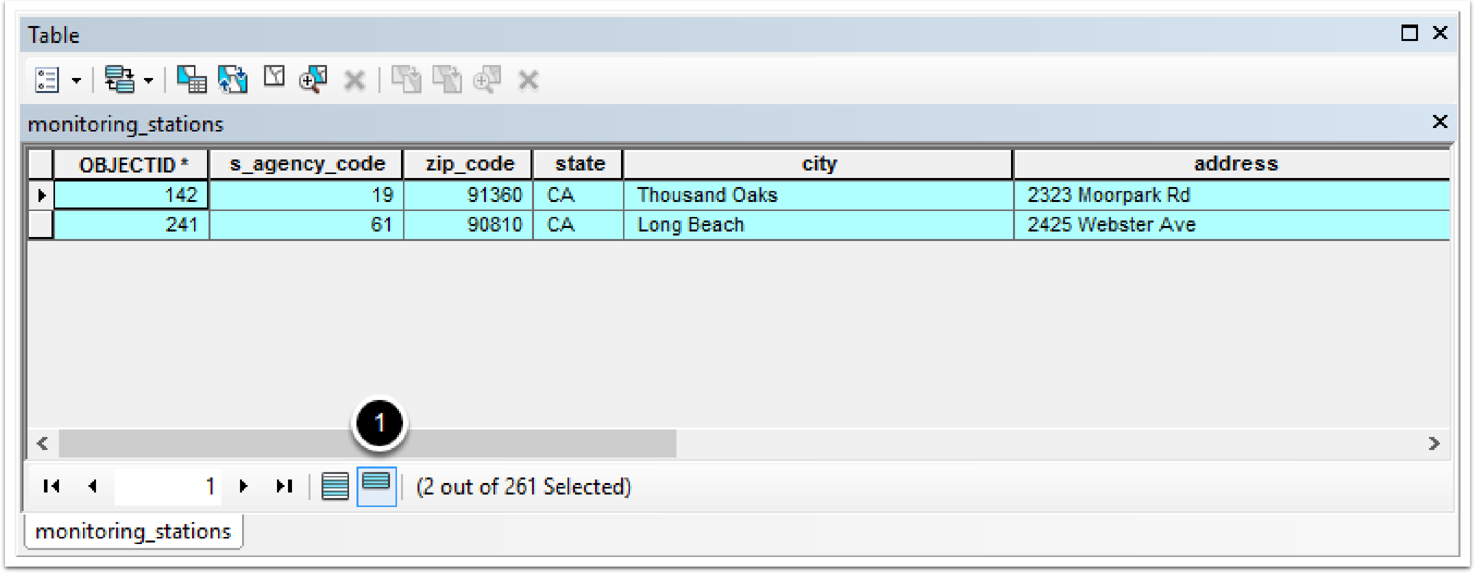
SELECT * FROM monitoring_stations WHERE:

[OBJECTID] IN (241, 142)

7.2 Viewing selected rows

Now, let's view the selected rows in the attribute table

1. Open the attribute table, and near the bottom click the button to toggle to the mode that shows only the selected rows.



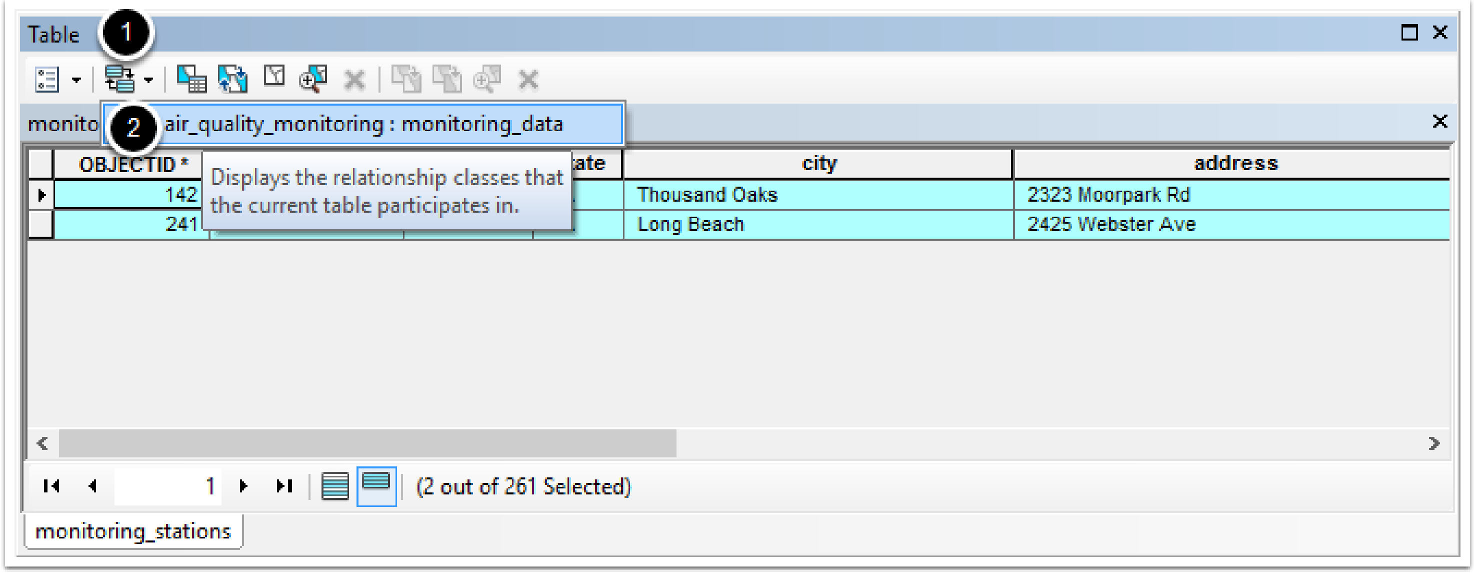
OBJECTID *	s_agency_code	zip_code	state	city	address
142	19	91360	CA	Thousand Oaks	2323 Moorpark Rd
241	61	90810	CA	Long Beach	2425 Webster Ave

(2 out of 261 Selected)

7.3 Viewing related records

Now, this shows us the records we selected, but what about the data we entered that is related to these records? Well, we can access that by traversing the relationship class. The second icon from the left (the *Related Tables* icon) on the top bar of the *Table* window lets us do that.

1. Click the *Related Tables* icon
2. In the options that come up, click the relationship *air_quality_monitoring : monitoring_data,* which shows the name of our relationship class, then the name of the table as we defined it in the relationship class.



Displays the relationship classes that the current table participates in.

OBJECTID *	ate	city	address
142		Thousand Oaks	2323 Moorpark Rd
241		Long Beach	2425 Webster Ave

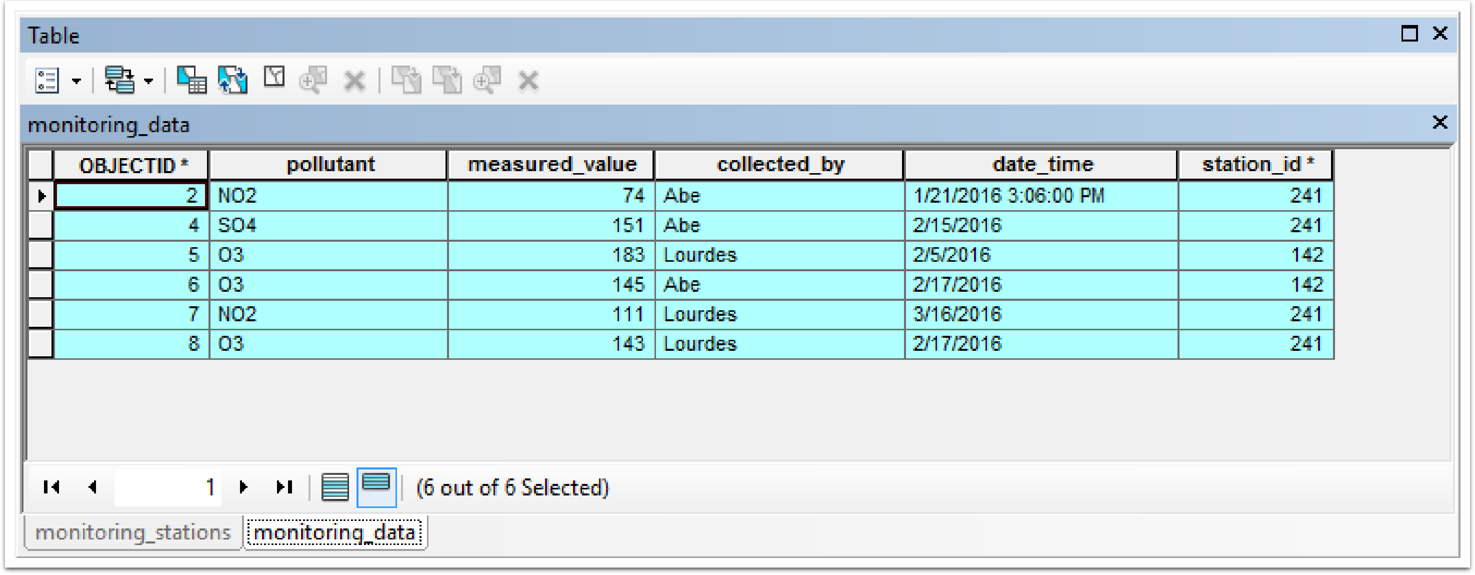
(2 out of 261 Selected)

7.4 Related records

Now, we see the records related to our selected air quality stations. In this case, it happens to be all of the records we entered, but if we had only one of the stations selected, it would only show those records here! In this way, we can view the values for pollution measurements at these stations.

If you want to explore further, try selecting just a single one of the monitoring stations using select by attributes and look at just the records for that station. You can also try using the graphing tools in the *Table* menu to explore the values. There are many possible directions to go from here, but for now, this tutorial is complete.

You now know how to create a spatial layer from a data table with coordinates, how to create new data tables, how to enter data into new tables, and how to relate tables together into more complex data structures and databases. Nice work!



OBJECTID *	pollutant	measured_value	collected_by	date_time	station_id *
2	NO2	74	Abe	1/21/2016 3:06:00 PM	241
4	SO4	151	Abe	2/15/2016	241
5	O3	183	Lourdes	2/5/2016	142
6	O3	145	Abe	2/17/2016	142
7	NO2	111	Lourdes	3/16/2016	241
8	O3	143	Lourdes	2/17/2016	241