**COD Quick Check Kiosk - Instructor's Guide**

(Errors and answers to questions are described in **BOLD** text - and are not included in the participant version)

**Day 2 14:00 to 15:00 Kiosk 3: COD Data Cleaning +PCodes (3 Dec)**

* Value of Pcodes (3 mins)
* Additional Resource [ITOS - quick check sheet](https://drive.google.com/open?id=0B-Api46szKw4aGN1YmFiN0RMTlE)
* Quick review of Admin Bdy CODs (exercise)

This kiosk is designed to do a quick check of the most important features of a set of COD administrative boundary data layers. It is not an exhaustive evaluation but is intended as a preliminary of quality control check, primarily to identify nesting across admin levels.

Open and follow the instructions on the [COD Quick Review Checksheet](https://drive.google.com/open?id=1_33j968sU0ndTNEMwuXAztQGYxgDq3KTetHkciNNob4) (<https://drive.google.com/open?id=1_33j968sU0ndTNEMwuXAztQGYxgDq3KTetHkciNNob4>). The checksheet has detailed instructions and snapshots of what the participants will see when they evaluate the CODs using unchecked.

Instructor can either:

1. open an ArcMap session and follow the same procedures in the application, or
2. use [this slide show](https://drive.google.com/open?id=1NLn64zNSph60jNsMP_z04pR-yQB-s3oonUB_dBNeIgo) along with this document to show the issues in the datasets: <https://drive.google.com/open?id=1NLn64zNSph60jNsMP_z04pR-yQB-s3oonUB_dBNeIgo> or
3. Use the participant version (<https://drive.google.com/open?id=1_33j968sU0ndTNEMwuXAztQGYxgDq3KTetHkciNNob4>)

***Download the zip file containing LBN data. It will be used in both COD Kiosks***.

Participants LBN Sample Data: <https://drive.google.com/open?id=0B-Api46szKw4ZkpVZHRyWGVFU1k>

Instructors LBN Sample Data: <https://drive.google.com/open?id=0B-Api46szKw4blpxcnlsZW9kN1U>

**Note**: some of the errors we are going to look at were part of the original data, but other have been added for teaching purposes. This data is derived from the original Lebanon CODs but it has been modified to create errors that didn’t exist in the original dataset. The errors we will look at are typical of the type of issues we see in many datasets.

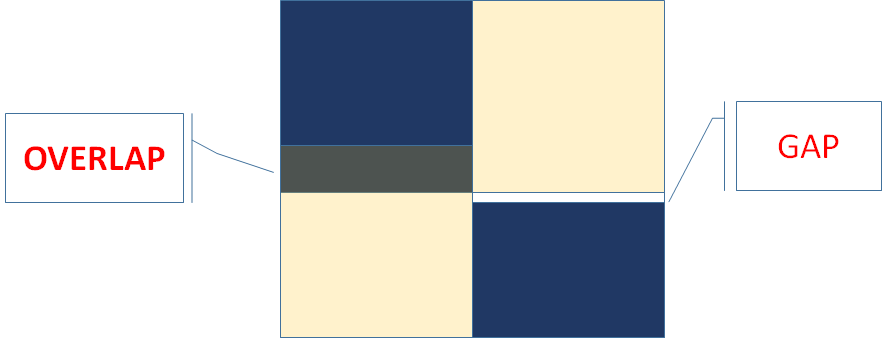
***<slide 1>***

Kiosk: Quick COD Admin Check

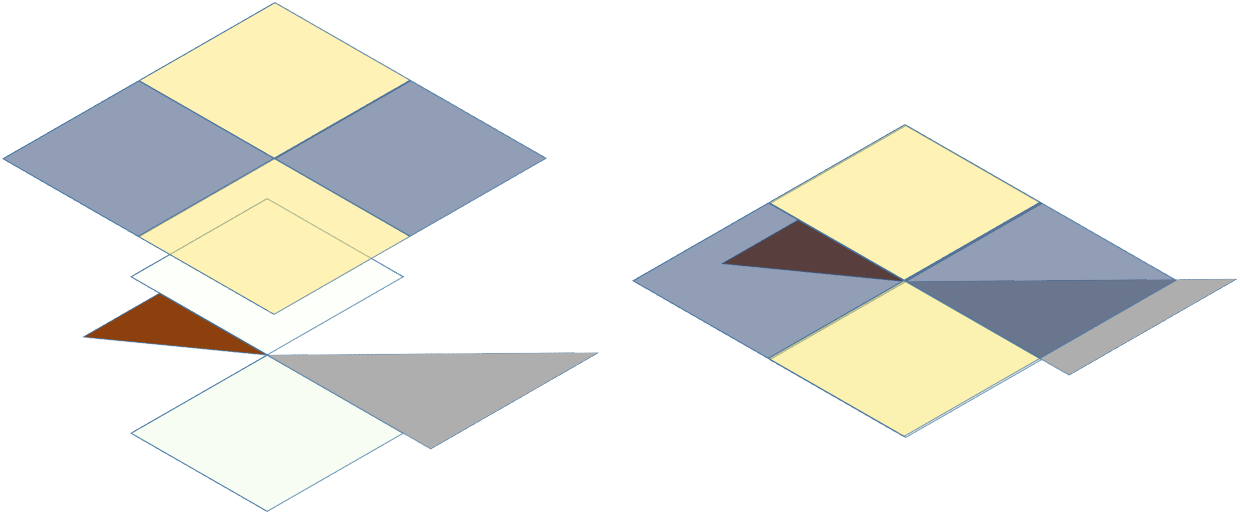
Sample data from Lebanon

A quick reminder before we get started:

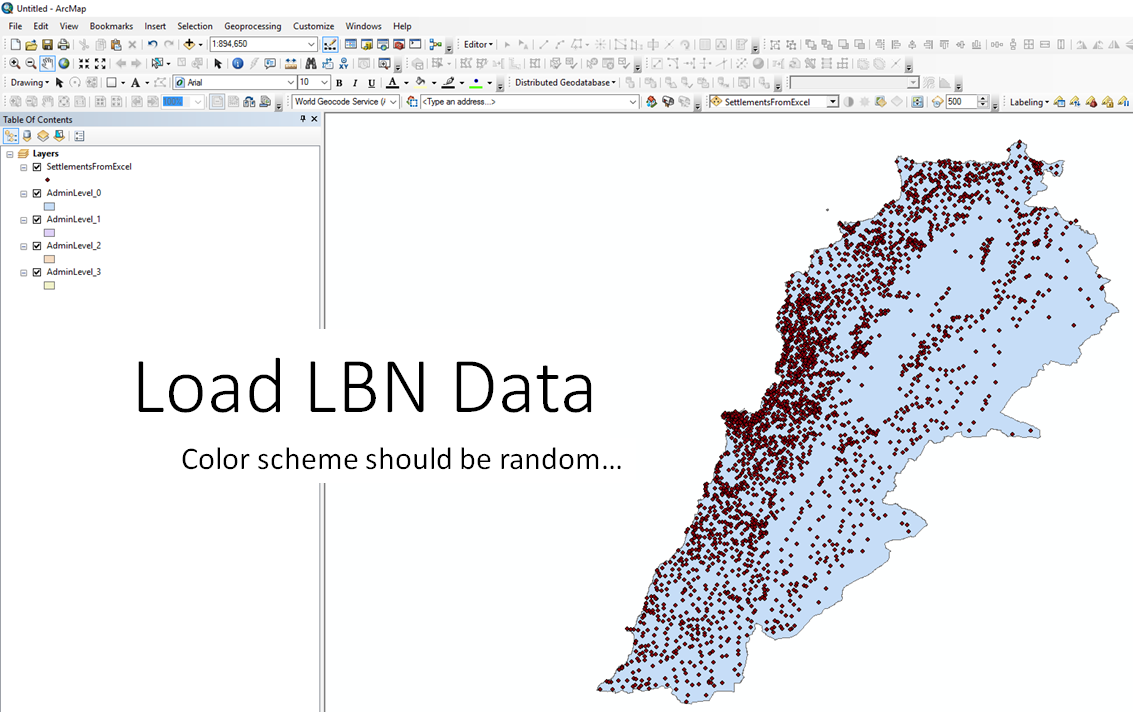
***<slide 2>*Topology** refers to errors within a single layer



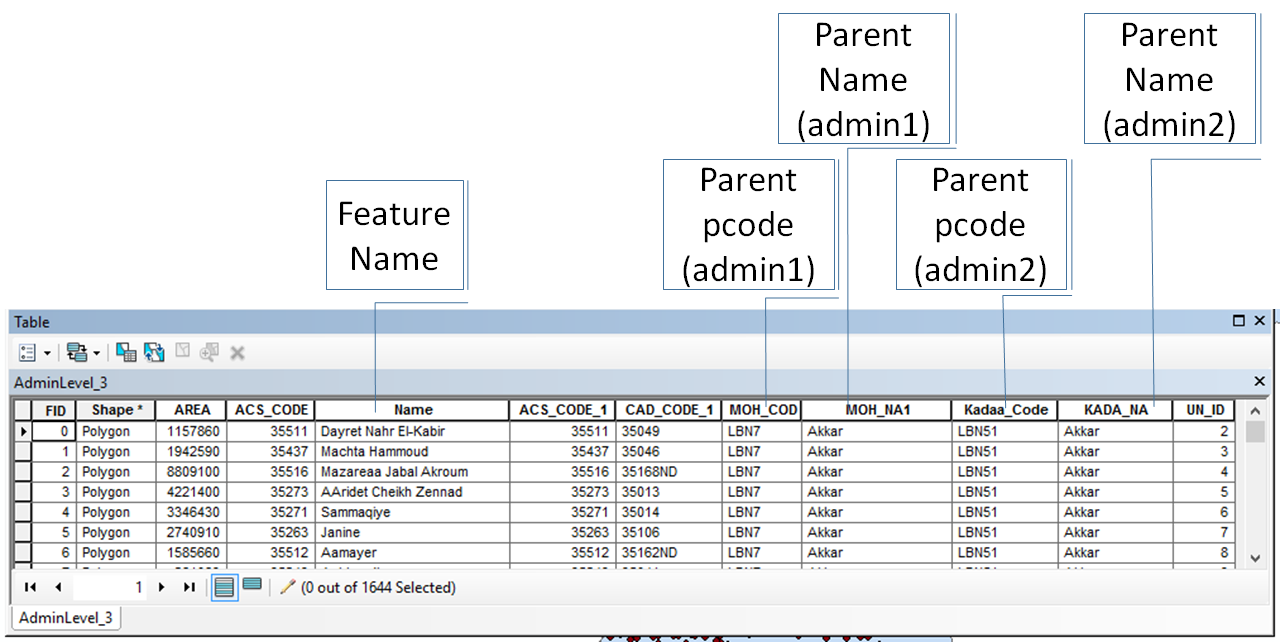
***<slide 3>*Nesting** refers to errors of spatial alignment across multiple layers



1. Download and unzip the data for Lebanon (<https://drive.google.com/open?id=0B-Api46szKw4blpxcnlsZW9kN1U>). You should see 3 admin layers and 1 settlement layer.
2. ***<slide 4>***Load all data layers into an ArcMap session. The color scheme should be random and will look something like this

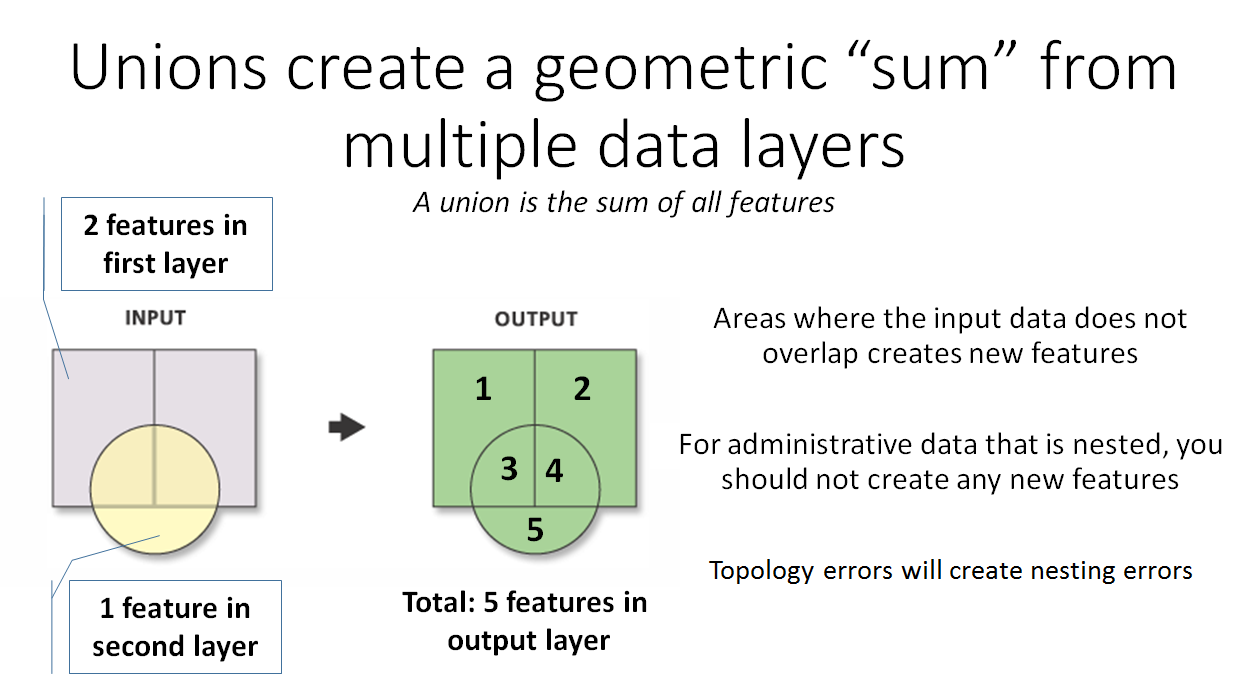


1. Visually inspect all layers’ Attribute Tables making sure all attribution that should be present, is present (single fields for feature names and pcodes, and parent feature names and pcodes. If you are working with populated places you should also have capital designations). **<slide 5>** Let’s look at admin3 first.
   1. *Are the names and pcodes for all parent layers present in the table?* ***No***. **Admin 0 name and pcode are missing**;
   2. *Is it clear which field is the pcode field for this layer?* ***No*, there are *4* possible fields for for the feature pcode: “ACS\_CODE,” “ACS\_CODE\_1,” “CAD\_CODE\_1” And “UN\_ID.”**
   3. *Repeat for all layers - divide the tasks amongst teams in the room*
      1. **Admin0: Missing feature name field**
      2. **Admin1: Missing admin 0 name and pcode fields**
      3. **Admin2: Missing admin 0 name and pcode fields**
      4. **You might also note the lack of consistency in the field names. For example,** 
         1. **in the admin1 layer the name field is “Gov\_New” and the code is “MOH\_code\_n”**
         2. **In the admin2 layer, the name field for the admin1 parent name is “Admn1\_N,” and the admin1 parent code is “Admin1\_C”**
         3. **In the admin3 layer, the name field for the admin1 parent name is “MOH\_NA1” and the admin1 parent code is “MOH\_code”**

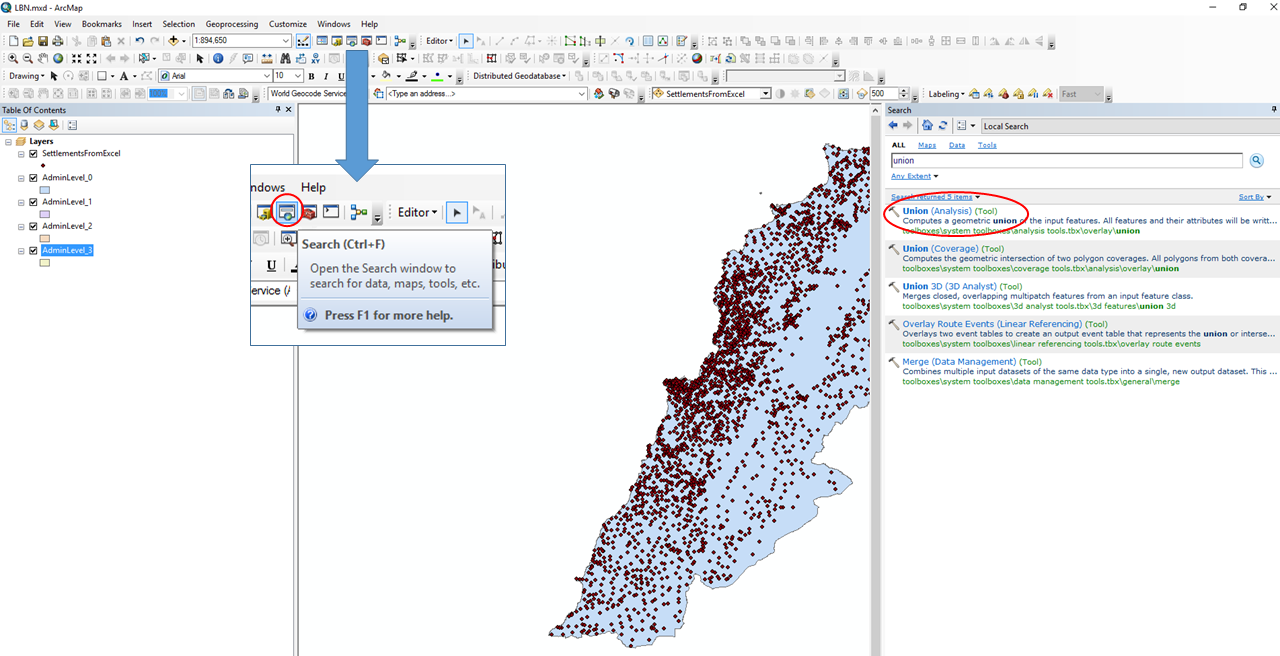


**Leave the errors for now**

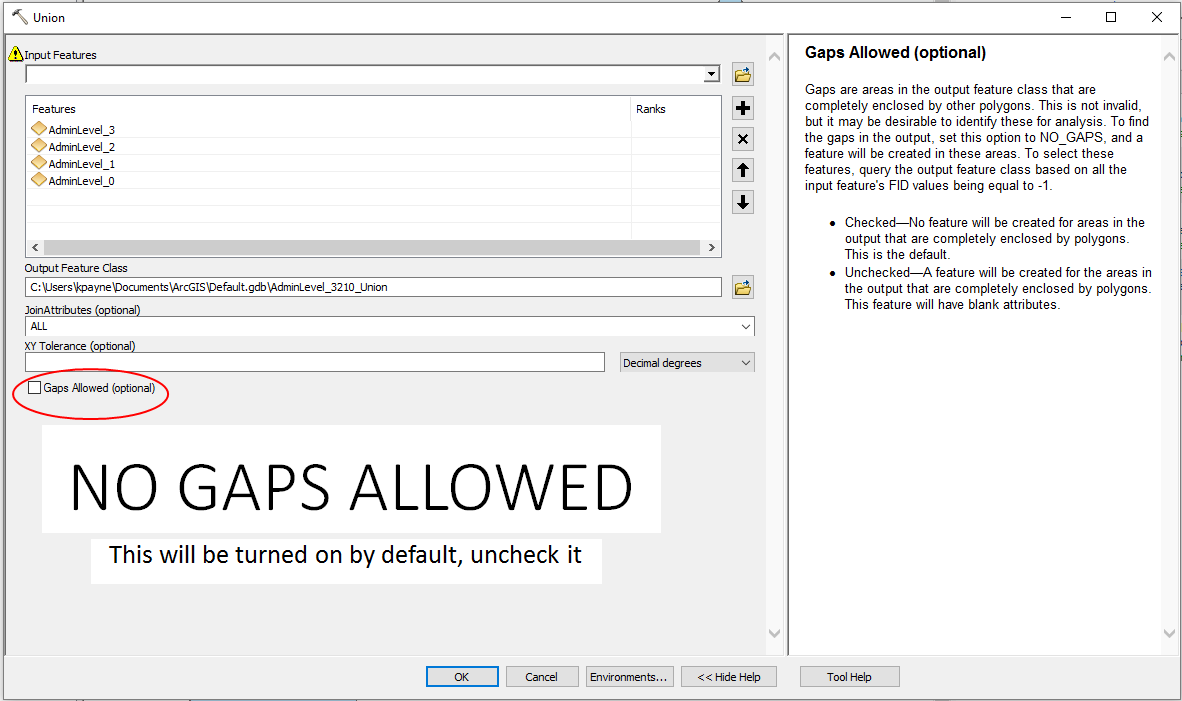
1. First, we will check to see if the data is geometrically nested. Check for nesting across the layers by creating a temporary ‘Union’ of all the Admin layers together. Reminder, a union is the sum and intersection of all input geometries or features. **<slide 6>**



* 1. **<slide 7>** Open the Union tool (I usually just search for it using the search button)

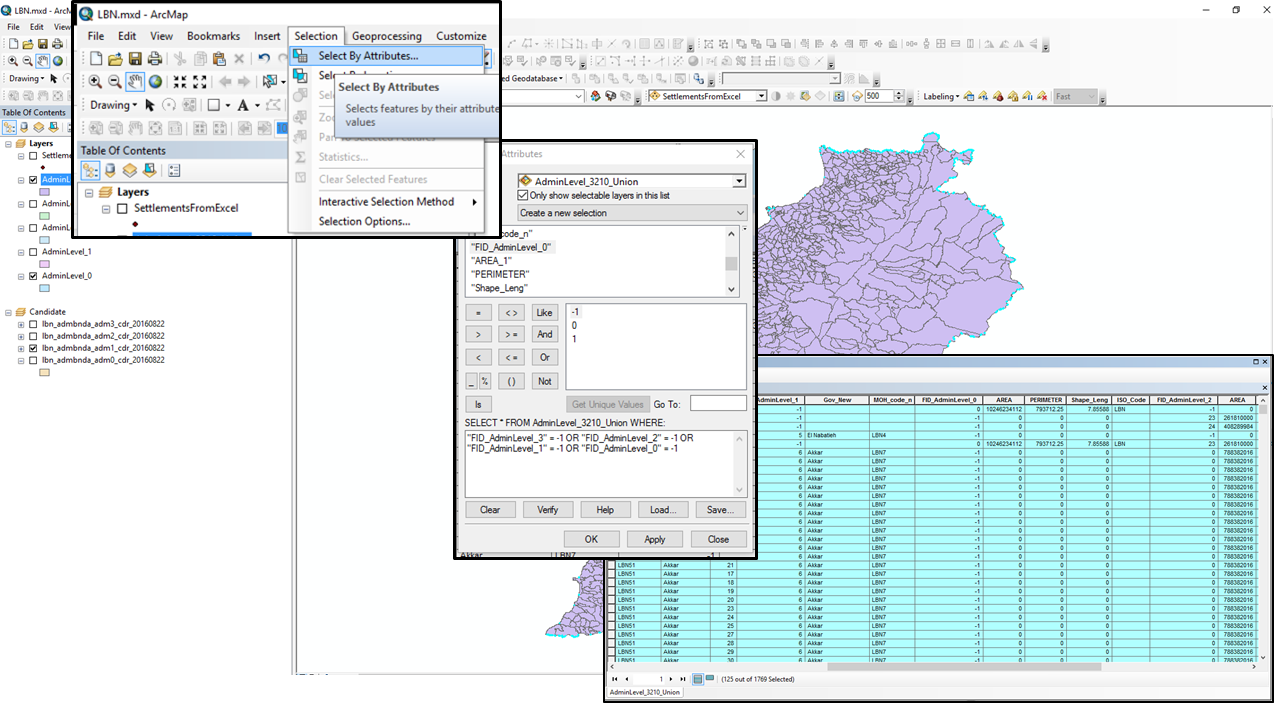


* 1. **<slide 8>** Either drag and drop or select all of the layers from the drop down menu as input features
  2. Choose an output data layer (you can accept the default or rename it - but it must be a feature class in a GDB, not a shapefile. You can write to the default GDB in the ...Users\username\Documents\ArcGIS\Default.gdb if you want)
  3. Be sure to uncheck ‘Gaps Allowed’ option. (In some rare cases gaps are allowed - if a country prefers “holes” for large water features or parks at lower admin levels for example, but generally they are incorrect and need to be checked)



1. The output dataset should be automatically added to your table of contents. If all datasets are geometrically nested, no new features will be created. Verify that the output ‘Union’ layer has the same number of polygons as the lowest level Admin layer that participated in the union. The lowest level is the highest number. For example, if you have admin layers 0, 1, 2, and 3, admin level 3 is your lowest admin layer.
   1. *How many records would you expect to see?* **Admin3 had *1,644* input features so I would expect *1,644* features in the union.**
   2. *How many features are in the output union?* ***1,772* (additional *128* features)**
2. If the unioned data does not have the same number of features (or records) as the lowest admin level, this indicates either:
   1. a geometric nesting error across layers or
   2. topology errors within one or more of the layers. *In the kiosk later this week we will focus on how to fix topology errors.*
3. *(optional exercise - we can delete for time)* If you want to see some of the common nesting errors, open the attribute table for the union.

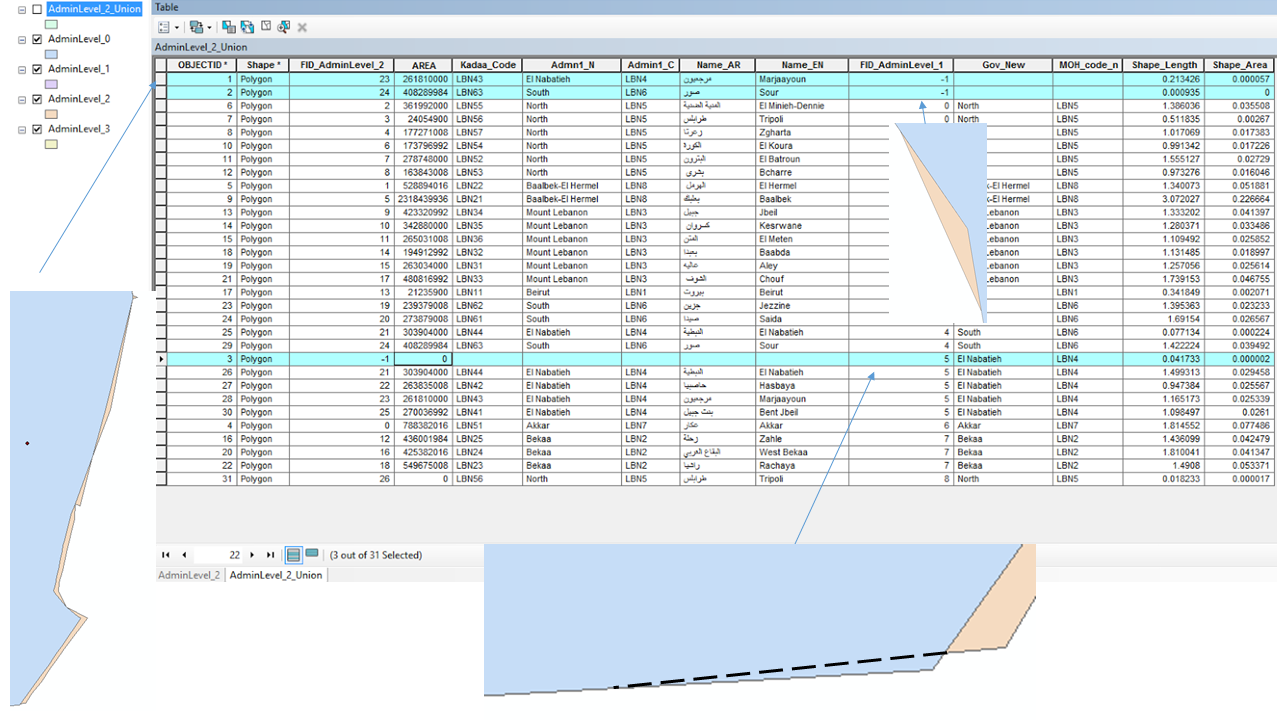
**<slide 10>**



Unioned features inherit their attribution from the parents. If a portion of the geometry does not exist in one of the input data layers, there will be no attribution. To visualize the location of ***some*** of the nesting errors, select by attributes in the unioned dataset cases where any of the input FIDs are equal to -1 using a SQL statement like: "FID\_AdminLevel\_0" = -1 OR "FID\_AdminLevel\_2" = -1 OR "FID\_AdminLevel\_3" = -1

You will select 126 features - mostly misalignment along the admin0 border.

**<slide 11>** Zoom to some of the selected features to see common alignment errors.



*(optional exercise - we can delete for time)* **<slide 11>** If you want a complete list of all errors, in the attribute table for the unioned datasets, find the attribute field for the FID for the lowest Admin level that participated in the Union. In this case, the Union was performed on layers Admin0, Admin1, Admin2, and Admin3, and the attribute field of interest in the *Admin\_Union* layer is the ‘FID\_AdminLevel\_3’ field. The records in this field which indicate nesting errors will be those that either have duplicate FID values, blank FID values, or an FID value of negative 1.

Summarize on the lowest level ‘FID\_AdminLevel\_3’ field. Include the ‘Name’ field in the output (check the ‘First’ box). Click ‘OK’ to summarize.

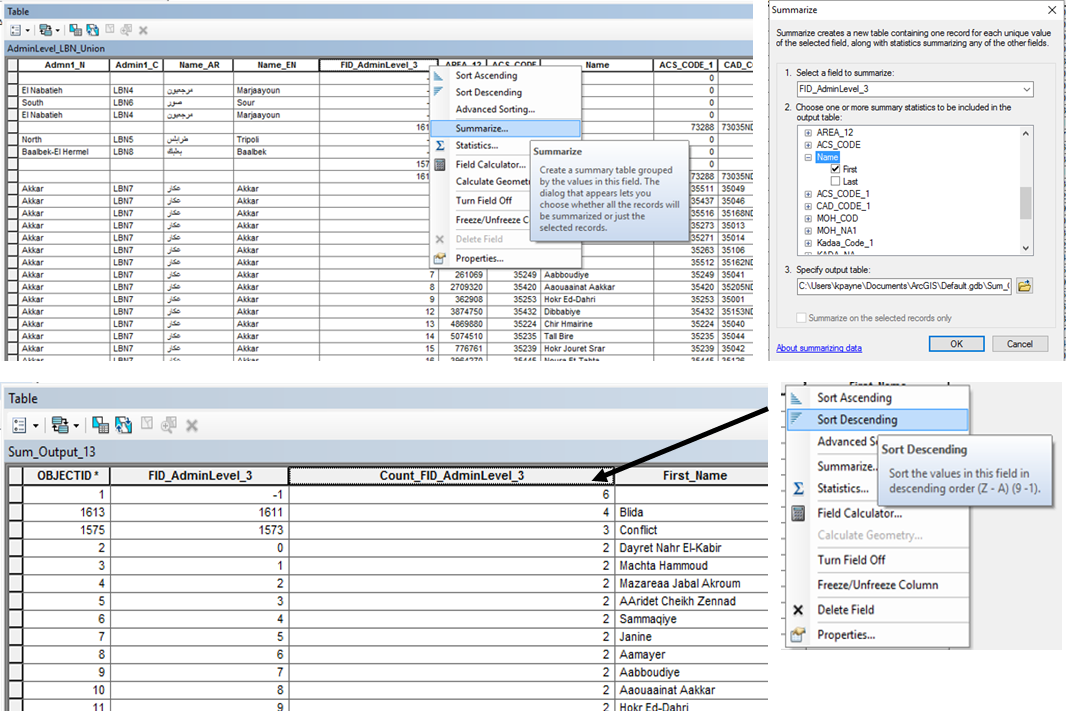
In the resulting ‘Sum\_Union’ table sort descending on the ‘Count\_FID\_AdminLevel\_3’ field and select all records with a value greater than 1.

* A ‘count’ greater than 1 indicates areas where features overlap and new features are created with all the attribute information of the input feature. The area of overlap will always generate two identical overlapping features, one for each of the features that participates in that overlap.

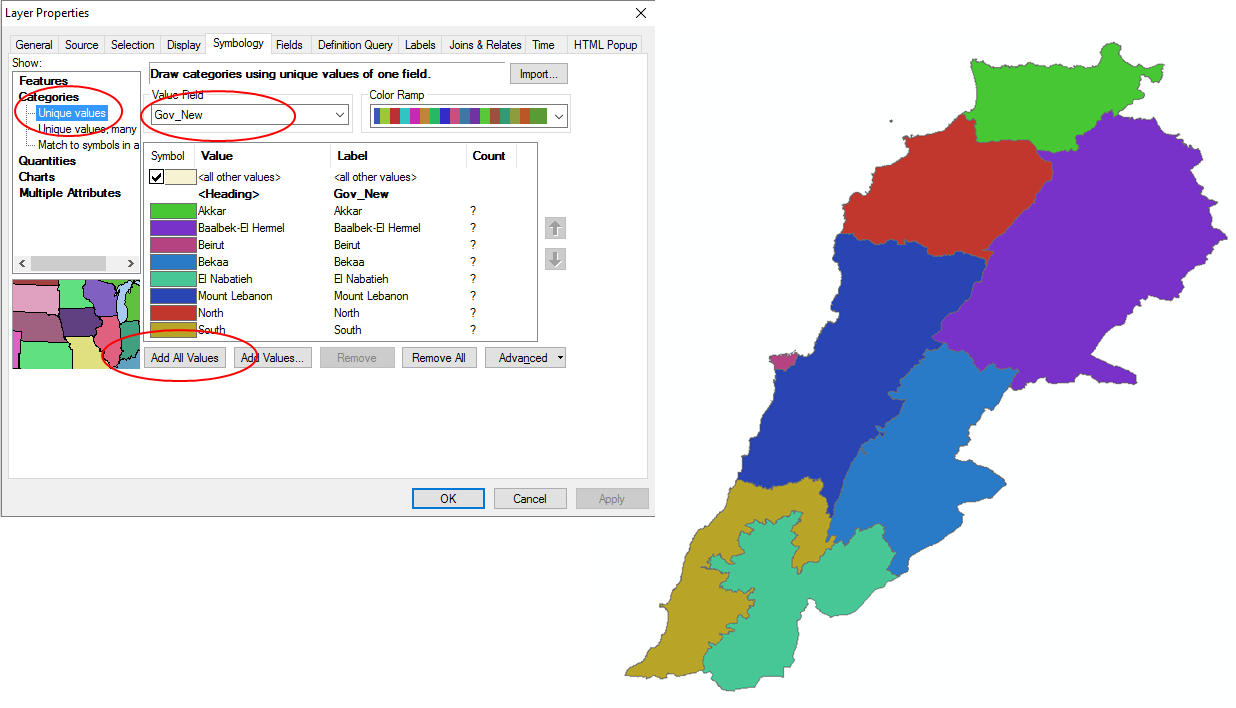
Also add to the selection (if not already selected) any records which have a blank ‘FID\_AdminLevel\_3’ field value as well as any with a negative 1 ‘FID\_AdminLevel\_3’ field value

* New features created in the gaps between polygons will have an FID = -1 and represent areas where an input feature, or any part of an input feature, does not intersect another input feature.

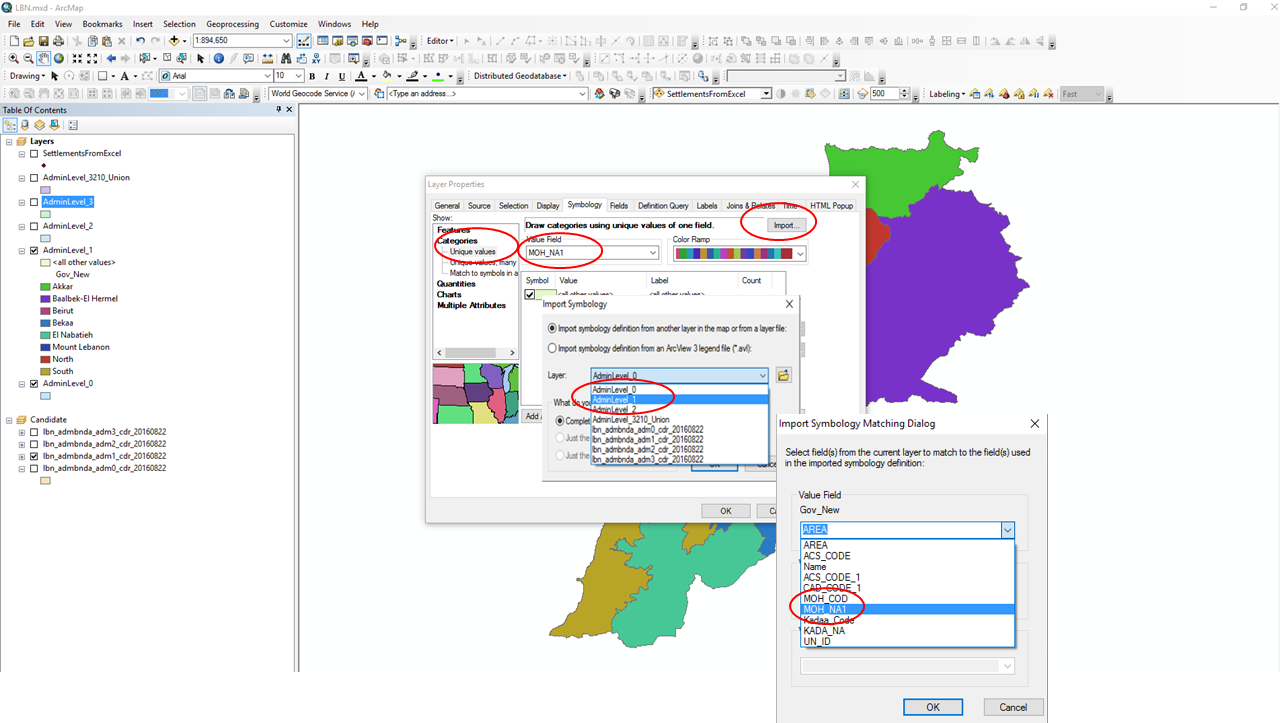
Doing this you will select 120 records in the sum table. Some of nesting errors will occur in levels 0-2, butthe ‘First\_Name\_EN’ field of these selected records will indicate the location of the lowest level Admin units where nesting errors occur.



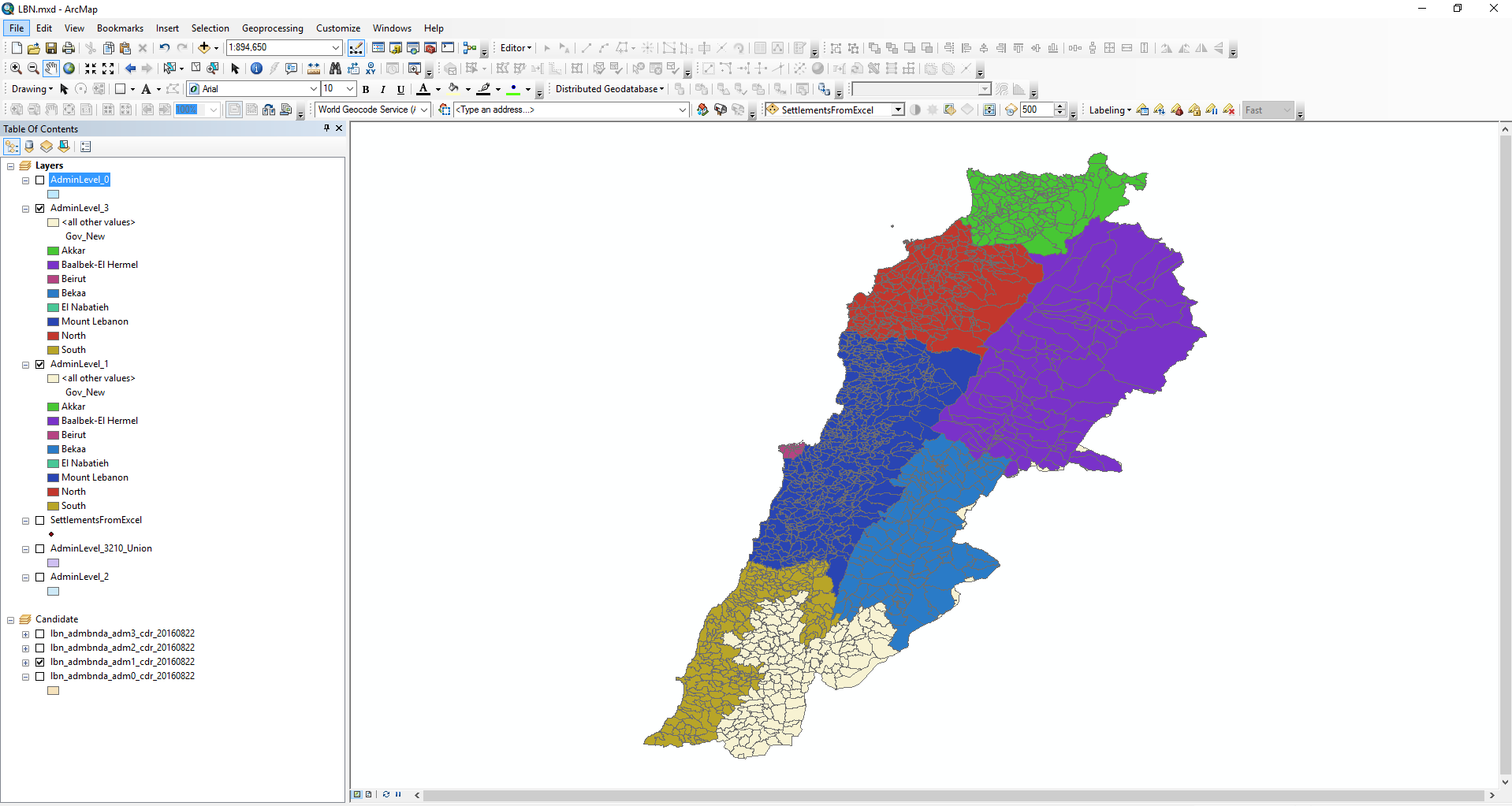
1. Next, we will do a superficial check to see if the data layers are correctly nested by attribution (temporarily ignoring any geometry nesting errors) **<slide 12>** Symbolize the Admin1 layer based on its ‘admin1Name’ so each polygon is a unique color and label the polygons with this ‘admin1Name’
   1. Double click on the admin1 layer to bring up the properties dialog box.
   2. In the symbology tab choose “Categories” then “Unique Values” and choose to symbolize your data by the Admin1 name field (‘Gov\_New’). Add all values. You can use any color ramp you like. Each unique admin1 name will now have a distinct color.



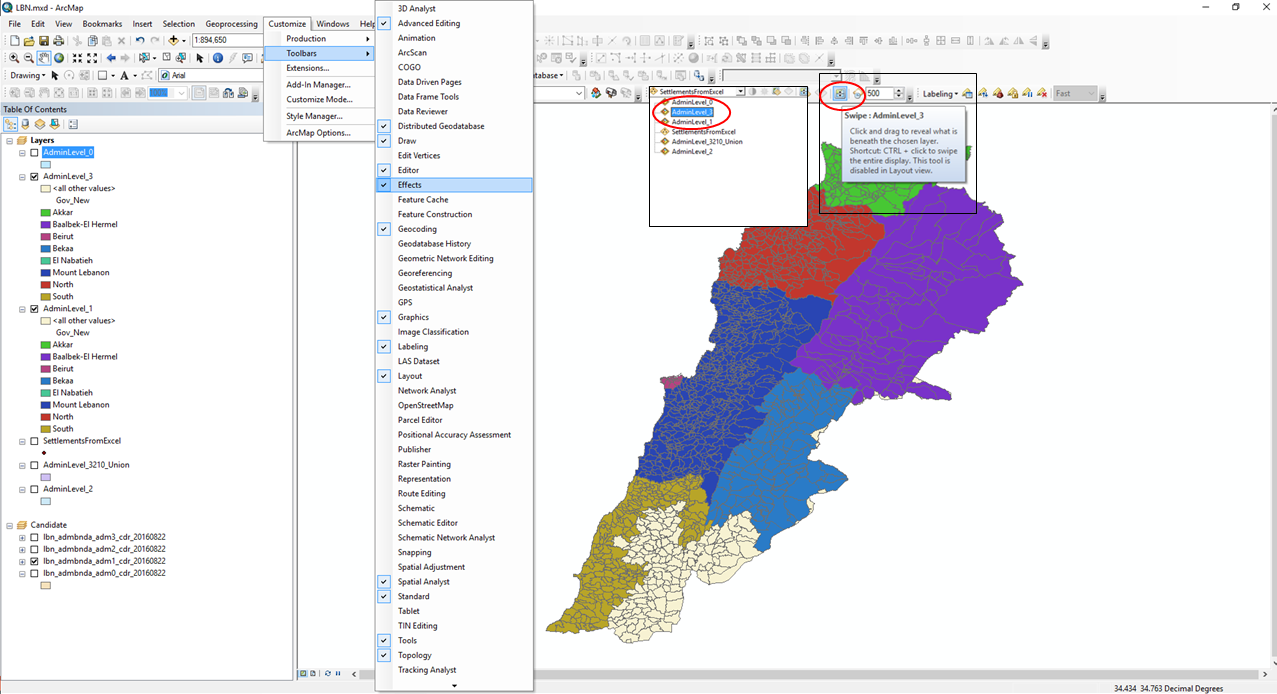
1. **<slide 13>** Now we will see if the admin3 features have the same admin1 names that are found in the admin1 layer. Symbolize the Admin3 layer using it’s ‘admin1Name’ and importing the symbology parameters from the Admin1 layer so it is using the same colors for the Admin1 units as the Admin1 layer
   1. Double click on the admin3 layer to bring up the properties dialog box.
   2. In the symbology tab choose “Categories” then “Unique Values” and choose to symbolize your data by the Admin1 name field (‘MOH\_NA1’).
   3. To make sure that the symbol set for both data layers is the same, choose ‘Import’ and select the “AdminLevel\_1” dataset to import the symbol set from
   4. In the Import Symbology Matching Dialog box, set the Value Field for ‘Gov\_New’ (which is the admin1 name field in the admin1 layer) to ‘MOH\_NA1’ (which is the admin1 name field in the admin3 layer)*. Do you see the advantage of using the same name fields for the same properties in all datasets?* This will match the names in both layers and give them the same color. Hit OK, and OK. You should now have both admin1 and admin3 layers displayed in the same color palette, based on the admin1 names in both layers. If you haven’t already, place the admin3 layer on top of the admin1 layer so that the admin3 layer draws first.



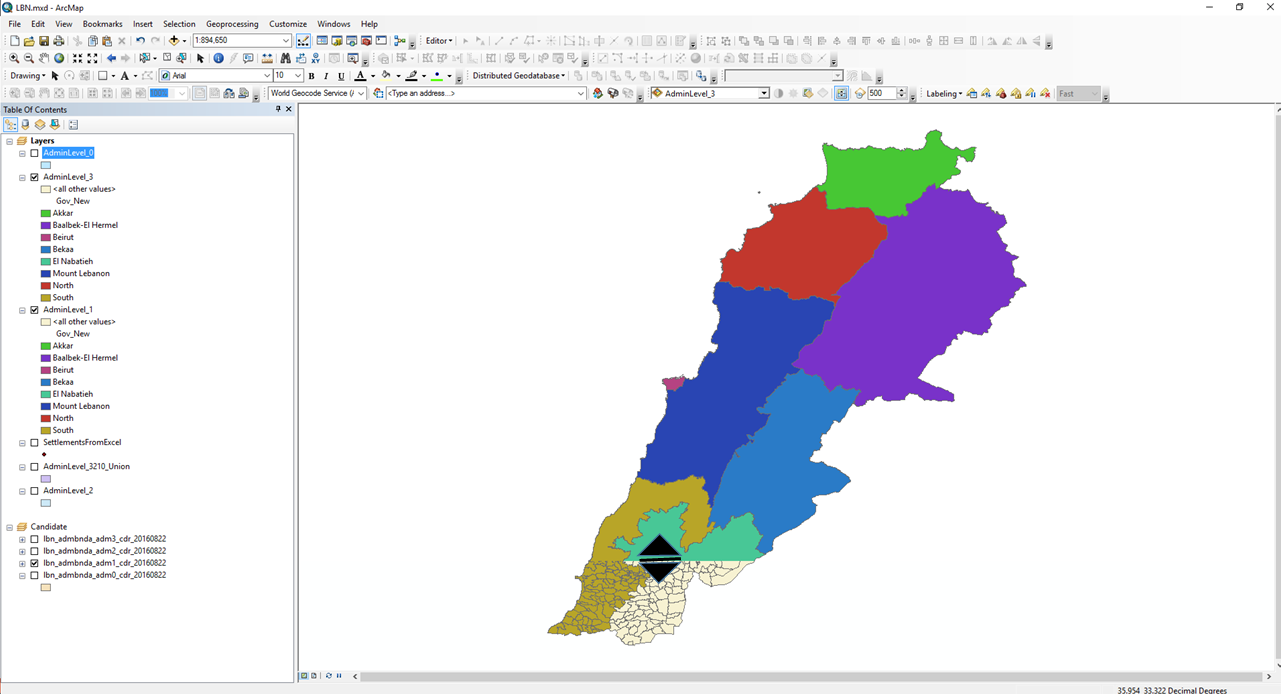
1. **<slide 14>** With the Admin3 layer drawing on top of the Admin1 layer, you can see the admin3 feature lines, but all admin3 units that belong to the same parent admin1 unit are the same color.



1. Toggle the Admin3 layer on and off in the table of contents or use the swipe tool to verify that its Admin1 attribution matches the Admin1 layer names. If the attribution values aren’t identical to that used in the Admin1 layer then when we import the symbology new symbol sets (colors) will appear, as we will see.
   1. **<slide 15>** How to use the swipe tool: If it isn’t already loaded, enable the ‘Effects’ toolbar. Choose the ‘Customize’ drop down menu, select ‘Toolbars’ then ‘Effects’ This will add a new set of tools to your ArcMap session.
   2. In the ‘Effects’ drop down menu, select the AdminLevel\_3 dataset

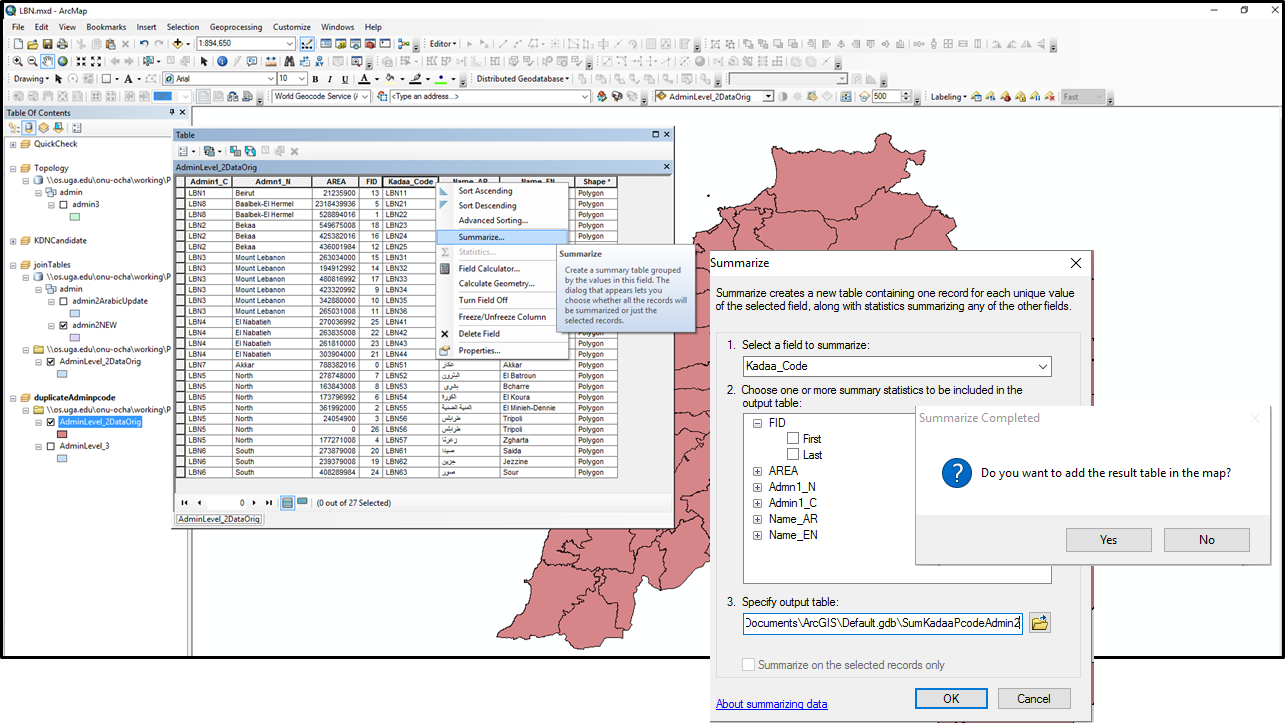


* 1. Click the ‘Swipe’ tool on the toolbar and position the pointer at the top, bottom, or side of the map window. An arrow will appear in the map display indicating the direction in which to swipe the layer. Click and hold the left mouse button to drag the arrow up and down in your viewing window. As you do, the top layer will be temporarily erased relative to the position of the pointer. The swipe layer (the ‘AdminLevel\_3’ layer) will temporarily disappear from view in the direction you drag the arrow. You will see the ‘AdminLevel\_3’ dataset disappear and reveal what is underneath (the ‘AdminLevel\_1’) dataset. You will see the AdminLevel\_3 boundary lines disappear.
  2. **<slide 16>** If the colors remain consistent as you sweep the dataset, it is an indication that the layers are largely consistent (*this won’t catch the small nesting errors we saw above*).
  3. *Do you notice any changes or discrepancies*? **The southeastern most admin1 unit changes colors, indicating a different admin1 name across the layers**. **The Admin1 layer indicates this unit is called *El Nabatieth*, while the admin3 layer refers to this area as *Nabatiye*.**

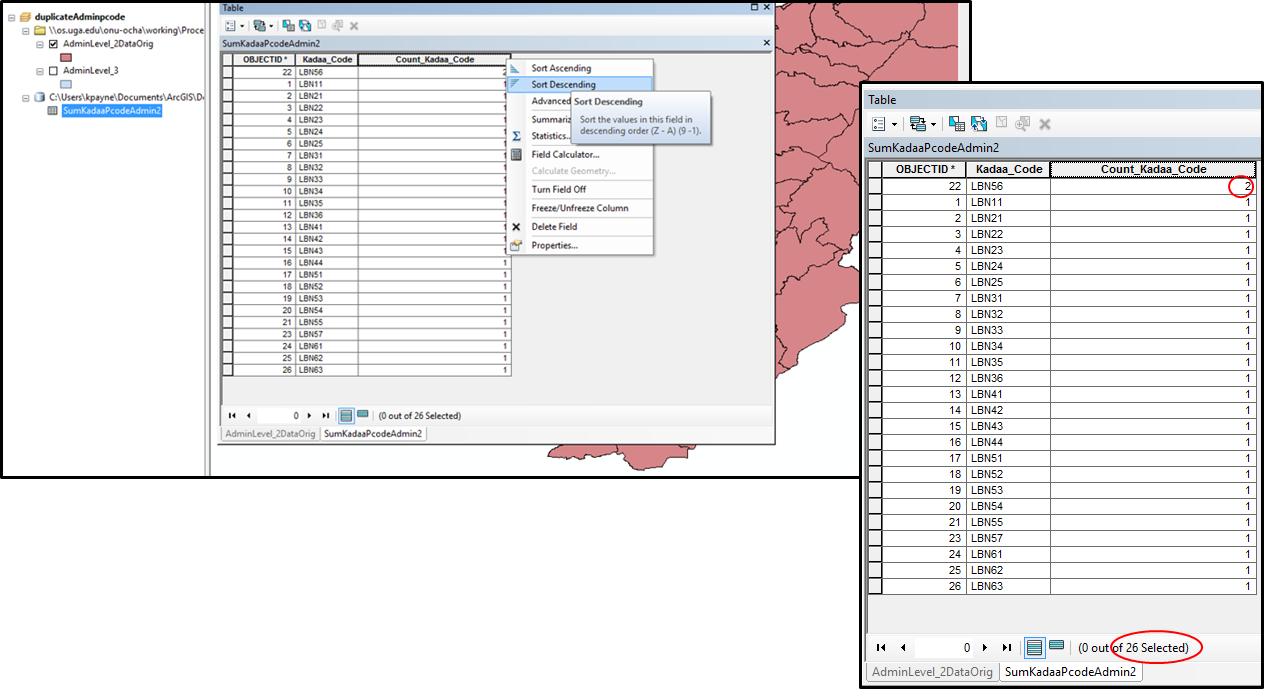


1. If you were to do a complete check of the dataset, you would:
2. Repeat this process for each additional Admin layer symbolizing by its ‘admin1Name’ attribution using the same symbol set as the Admin1 layer and then visually cross referencing it against the Admin1 layer. In this case you would symbolize the admin2 layer by the admin1 name and check it against the admin1 layer.
3. Then symbolize the Admin2 layer based on its ‘admin2Name’ and then use this symbol set to symbolize all the other layers lower than level 2 based on their ‘admin2Name’ as well (in this case the admin3 layer). Then visually cross reference the layers with each other to verify correct Admin2 attribution across all layers
4. Repeat for all layers, checking all parent admin names in all layers against all higher levels. For example, assuming the admin0 attribution was correct at all levels, if you had 3 admin levels, you would:
   1. Check the admin3 layer attribution for the parent admin1 names against the admin1 names stored in the admin1 data layer
   2. Check the admin3 layer attribution for the parent admin1 names against the admin1 names stored in the admin2 data layer
   3. Check the admin3 layer attribution for the parent admin2 names against the admin2 names stored in the admin2 data layer
   4. Check the admin2 layer attribution for the parent admin1 names against the admin1 names stored in the admin1 data layer
   5. Repeat the above cross reference symbology tests this time using ***‘Pcode’*** attribution for all layers
5. This visual swipe test is handy when there aren’t many units to check, but it should be used with caution. In areas like the Philippines where there are thousands of units it would be easy to miss color changes between units at lower admin levels. A more stringent check would be to fix all topology errors and the spatially join the layers based on attribution to thoroughly check the databases. We will cover topology in the next kiosk.
6. Visual checks are great, but it is more accurate to conduct quality checks directly with the database, especially when working with larger datasets. In this step we are going to check for duplicate pcodes.

* **<slide 17>** Open your admin2 attribute table and find the field that houses the pcode for that layer. Sort the pcodes in Ascending order and note the existence of any <Null> values. All features should have a pcode so null or blank values should be populated.
* Right click on the ‘Kadaa\_Code’ field and choose ‘Summarize’
* In the Summarize dialog box specify the output - you can send it to the default GDB or a stand alone table in your workspace.
* Choose ‘Yes’ to add the result table to the map



* **<slide 18>** Open the output summary table and sort the ‘Count’ field in descending order. Inspect the top rows of the table for any values greater than 1 in the “count” field.
* If all pcodes were unique, they would each have a count of 1, and the number of summary records would equal to the number of records in the admin layer. Any values greater than 1 (not counting those associated with <Null> values) in the sum table represent duplicate p-codes. *Do any duplicates exist?* **Yes, LBN56 is a duplicate.**



* **<slide 19>** Locate the duplicate in your input dataset using the Select by Attributes tool and zoom to the selection. *What do you see?* **One offshore island and one multipart polygon on the mainland.** COD guidance indicates that all polygons or features that belong to the same administrative area should be included in multipart polygons.
* *How would you correct this?* **In this case, because the polygons all belong to the same admin unit you would merge the two polygons, as we will see in the topology kiosk. If these two features were separate administrative units, you would edit the database and assign a new pcode to one of the features. We will look at the merge process in the next kiosk.**

