Exercise 8

Manipulating spatial and tabular data

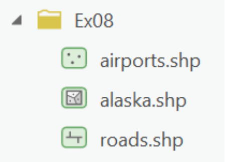
Exercise data

Exercise data for this book can be downloaded from https://go.esri.com/PythonData. This is a link to the ArcGIS® Online group called *Python Scripting for ArcGIS Pro (Esri Press)*. The data for exercise 8 is posted as a zip file called PythonScripting_Ex08_Data.zip. Download this file and extract it to a folder of your choice. The instructions use a folder called C:\PythonPro, but you can use a different folder provided you update any paths.

Work with search cursors

Cursors are used to iterate over the rows in a table. Several cursor methods exist to manipulate these rows. Search cursors are used to search records and carry out SQL expressions in Python.

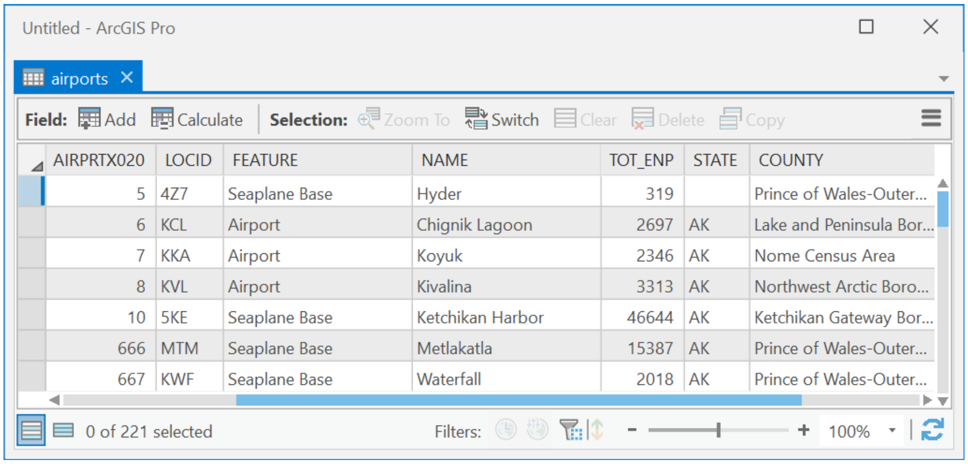
1. **Start ArcGIS® Pro with a new empty project.**
2. **Make sure the Catalog pane is visible by clicking Catalog Pane on the View tab. Dock the Catalog pane to the right side of the ArcGIS Pro interface.**
3. **Create a new folder connection to the location of the exercise data by right-clicking Folders > Add Folder Connection and navigating to the folder—e.g., \Ex08.**
4. **Examine the contents of this folder.**



Ex08
airports.shp
alaska.shp
roads.shp

5. **Right-click on the shapefile airports.shp and drag it to a new map.**

6. **Right-click on the airports feature layer in the Contents pane of the active map, and click Attribute Table. Review the fields in the attribute table.**



AIRPRTX020	LOCID	FEATURE	NAME	TOT_ENP	STATE	COUNTY
5	4Z7	Seaplane Base	Hyder	319		Prince of Wales-Outer...
6	KCL	Airport	Chignik Lagoon	2697	AK	Lake and Peninsula Bor...
7	KKA	Airport	Koyuk	2346	AK	Nome Census Area
8	KVL	Airport	Kivalina	3313	AK	Northwest Arctic Boro...
10	5KE	Seaplane Base	Ketchikan Harbor	46644	AK	Ketchikan Gateway Bor...
666	MTM	Seaplane Base	Metlakatla	15387	AK	Prince of Wales-Outer...
667	KWF	Seaplane Base	Waterfall	2018	AK	Prince of Wales-Outer...

0 of 221 selected

Notice that there are fields called FEATURE (for the type of airport), NAME (the unique name of each airport), and TOT_ENP (a measure of the total number of passengers), and that the table contains 221 records.

7. **Start IDLE.**

8. **Create a new script file, and save your script as print_values.py to the \Ex08 folder.**

9. **Enter the following lines of code:**

```
import arcpy

arcpy.env.workspace = "C:/PythonPro/Ex06"

fc = "airports.shp"

cursor = arcpy.da.SearchCursor(fc, ["NAME"])

for row in cursor:

    print("Airport name = {0}".format(row[0]))
```

The script creates a search cursor on the feature class and uses a `for` loop to iterate over all the rows of the attribute table.

10. **Save and run the script.**

The result is a list of the names of all the airports, as follows:

```
Airport name = Hyder

Airport name = Chignik Lagoon

Airport name = Koyuk

Airport name = Kivalina

Airport name = Ketchikan Harbor

...
```

Note: Be careful printing results to the interactive interpreter because a feature class or table could contain millions of records. In the preceding example, it was confirmed in advance that the number of records was small. You also can add a counter to the loop to print only a small amount of the total number of records.

Use search cursors with SQL in Python

Search cursors can carry out SQL expressions in Python.

1. **In IDLE, create a new script file, and save your script as sql.py to the \Ex08 folder.**

2. **Enter the following lines of code:**

```
import arcpy

arcpy.env.workspace = "C:/PythonPro/Ex08"

fc = "airports.shp"

sql_exp = '"TOT_ENP" > 100000'

cursor = arcpy.da.SearchCursor(fc, ["NAME"], sql_exp)

for row in cursor:

    print(row[0])
```

3. **Save and run the script.**

The result is a list of the names of the airports for which the SQL expression is true. The field TOT_ENP is a measure of the number of passengers. The result is as follows:

```
Ketchikan
Juneau International
Kenai Municipal
Fairbanks International
Bethel
Ted Stevens Anchorage International
```

Take another look at the SQL expression used: "TOT_ENP" > 100000. Field delimiters for shapefiles consist of double quotation marks—i.e., "TOT_ENP"—but there are no quotation marks around the value of 100000 because TOT_ENP is a numeric field. The entire SQL expression needs to be in quotation marks, because the WHERE clause in the syntax of the search cursor is a string. This results in the string '"TOT_ENP" > 100000'.

This syntax can create complications. For example, for text fields in SQL expressions, the values require single quotation marks—for example, "NAME" = 'Ketchikan'. The statement in the WHERE clause must be in quotation marks, but whether you use double quotation marks ("") or single quotation marks (' '), the statement will produce a syntax error. The solution is to use the escape character (\), which otherwise would cause a syntax error, in front of the quotation marks. In Python, a backslash within a string is interpreted as an escape character, which is a signal that the next character is to be given a special interpretation. So instead of '"NAME" = 'Ketchikan'', the expression becomes '"NAME" = \'Ketchikan\''.

4. **Modify the SQL expression in the script as follows:**

```
sql_exp = '"FEATURE" = \'Seaplane Base\''
```

5. **Save and run the script.**

The result is a list of the names of the airports for which the SQL expression is true, as follows:

```
Hyder

Ketchikan Harbor

Metlakatla

Waterfall

Kasaan

...
```

Other complications arise when working with SQL expressions. Specifically, the field delimiters vary with the format of the feature class. Shapefiles and file geodatabase feature classes use double quotation marks—for example, "NAME"—whereas SDE geodatabase feature classes do not use any delimiters—for example, NAME. When a tool such as Select By Attributes or other dialog-driven queries are used, this syntax is automatically applied, but in scripting, it can be handled using the `AddFieldDelimiters()` function.

6. **Modify line 4 of the script as follows:**

```
delim_field = arcpy.AddFieldDelimiters(fc, "COUNTY")

sql_exp = delim_field + " = 'Anchorage Borough'"
```

7. **Save and run the script.**

The result is a list of the names of the airports for which the SQL expression is true, as follows:

```
Girdwood
Merrill Field
Lake Hood
Elmendorf Air Force Base
Ted Stevens Anchorage International
```

SQL is also used in several geoprocessing tools, and a similar approach can be used to create valid SQL expressions.

8. **Save the existing sql.py script as select.py to the Ex08 folder.**

9. **Modify the script as follows:**

```
import arcpy
arcpy.env.workspace = "C:/PythonPro/Ex08"
infc = "airports.shp"
outfc = "airports_anchorage.shp"
```

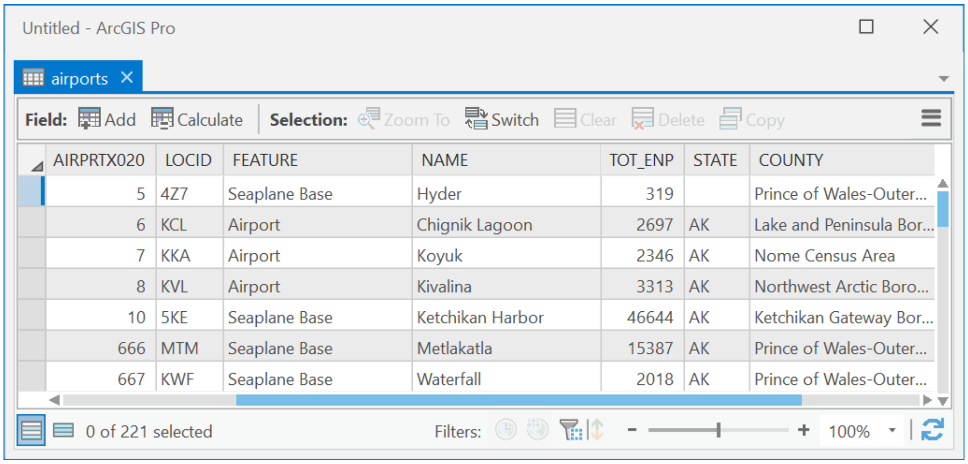
```
delim_field = arcpy.AddFieldDelimiters(infc, "COUNTY")
sql_exp =  delim_field + " = 'Anchorage Borough'"
arcpy.Select_analysis(infc, outfc, sql_exp)
```

10. **Save and run the script.**
11. **In ArcGIS Pro, confirm that the new shapefile was created with five point features.**

Work with update cursors

Two other cursor types can be used to work with row objects. Update cursors are used to make changes to existing records, and insert cursors are used to add new records. First, you will use an update cursor to update attribute values and delete records. Because this will permanently modify the data, it is a good idea to copy the data first to work on it.

1. **In the Catalog pane, navigate to the \Ex08 folder.**
2. **Right-click on the shapefile airports.shp and drag it to a new map.**
3. **Right-click on the airports feature layer in the Contents pane of the active map, and click Attribute Table. Review the fields in the attribute table.**
4. **Scroll over to the field STATE. Notice that some of the values in this field are blank.**
5. **Close the attribute table.**



AIRPRTX020	LOCID	FEATURE	NAME	TOT_ENP	STATE	COUNTY
5	4Z7	Seaplane Base	Hyder	319		Prince of Wales-Outer...
6	KCL	Airport	Chignik Lagoon	2697	AK	Lake and Peninsula Bor...
7	KKA	Airport	Koyuk	2346	AK	Nome Census Area
8	KVL	Airport	Kivalina	3313	AK	Northwest Arctic Boro...
10	5KE	Seaplane Base	Ketchikan Harbor	46644	AK	Ketchikan Gateway Bor...
666	MTM	Seaplane Base	Metlakatla	15387	AK	Prince of Wales-Outer...
667	KWF	Seaplane Base	Waterfall	2018	AK	Prince of Wales-Outer...

6. **In the Catalog pane, navigate to the \Ex08 folder and copy the airports.shp feature class so you can make edits without having to worry about keeping the original intact. Rename the copy to airports_original.shp.**

7. **Close ArcGIS Pro.**

There is no need to save your work.

Working with insert and update cursors is just like editing. When the feature classes work with in a script are open at the same time in ArcGIS Pro, it may result in errors because of you a shared lock that ArcGIS Pro places on the feature class. A typical error message looks as follows:

```
    ERROR 000464: Cannot get exclusive schema lock. Either
being edited or in use by another application or service.
```

To avoid this error, close ArcGIS Pro, run the script, and then open ArcGIS Pro again to examine the results.

8. **In IDLE, create a new script file, and save your script as update.py to the \Ex08 folder.**

9. **Enter the following lines of code:**

```
import arcpy
arcpy.env.workspace = "C:/PythonPro/Ex08"
fc = "airports.shp"
delimfield = arcpy.AddFieldDelimiters(fc, "STATE")
sql_exp = delimfield + " <> 'AK'"
with arcpy.da.UpdateCursor(fc, ["STATE"], sql_exp) as cursor:
    for row in cursor:
        row[0] = "AK"
        cursor.updateRow(row)
```

10. **Save and run the script.**

11. **Start ArcGIS Pro, and confirm that the values in the STATE field have been updated.**

12. **Close ArcGIS Pro.**

There is no need to save your work.

In addition to updating attributes using the `updateRow()` method, update cursors can also be used to delete records, which you'll do next.

13. **In IDLE, create a new script file, and save your script as delete.py to the \Ex08 folder.**

14. **Enter the following code:**

```
import arcpy

arcpy.env.workspace = "C:/PythonPro/Ex08"

fc = "airports.shp"

with arcpy.da.UpdateCursor(fc, ["TOT_ENP"]) as cursor:

    for row in cursor:

        if row[0] < 100000:

            cursor.deleteRow()
```

15. **Save and run the script.**

16. **Start ArcGIS Pro, and confirm that all the records with fewer than 100,000 passengers have been deleted.**

17. **Close ArcGIS Pro.**

There is no need to save your work.

Work with insert cursors

Insert cursors are used to create new records.

1. **In IDLE, create a new script file, and save your script as insert.py to the \Ex08 folder.**

2. **Enter the following code:**

```
import arcpy

arcpy.env.workspace = "C:/PythonPro/Ex08"

fc = "airports.shp"

with arcpy.da.InsertCursor(fc, "NAME") as cursor:

    cursor.insertRow(["New Airport"])
```

3. **Save and run the script.**

4. **Start ArcGIS Pro confirm that a new record has been added to the attribute table with the name New Airport, whereas the other fields are blank.**

5. **Close ArcGIS Pro.**

There is no need to save your work.

Note: Although a new record has been added and the attributes can be given values using the insert cursor, the new record does not have a geometry yet. This is covered in exercise 9.

Validate table and field names

ArcPy contains functions to validate the names of tables and fields. This prevents attempts to create invalid names, such as those with spaces or invalid characters.

1. **In IDLE, create a new script file, and save your script as validate.py to the \Ex08 folder.**

2. **Enter the following code:**

```
import arcpy

arcpy.env.workspace = "C:/PythonPro/Ex08"

fc = "airports.shp"

newfield = "NEW CODE"

fieldtype = "TEXT"

fieldname = arcpy.ValidateFieldName(newfield)

arcpy.AddField_management(fc, fieldname, fieldtype, "", "", 12)
```

3. **Save and run the script.**

4. **In ArcGIS Pro, confirm that a new field has been added with the name NEW_CODE.**

The space has been replaced by an underscore (_).

5. **Close ArcGIS Pro.**

   There is no need to save your work.

6. **Modify the script as follows:**

   ```
   newfield = "NEW?*&$"
   ```

   The characters ?, *, &, and $ are all invalid as field names.

7. **Save and run the script.**

8. **In ArcGIS Pro, confirm that a new field has been added with the name NEW____.**

   Each of the invalid characters has been replaced by an underscore.

Note: Validating table and field names does not determine whether the field name already exists. This requires checking the new name against the names of the existing fields.

9. **Close ArcGIS Pro.**

   There is no need to save your work.

10. **Modify the script as follows:**

    ```
    import arcpy
    arcpy.env.workspace = "C:/PythonPro/Ex08"
    fc = "airports.shp"
    newfield = "NEW CODE"
    ```

```
    fieldtype = "TEXT"

    fieldname = arcpy.ValidateFieldName(newfield)

    fieldlist = arcpy.ListFields(fc)

    fieldnames = []

    for field in fieldlist:

        fieldnames.append(field.name)

    if fieldname not in fieldnames:

        arcpy.AddField_management(fc, fieldname, fieldtype, "",
"", 12)

        print("New field has been added.")

    else:

        print("Field name already exists, no new field was
added.")
```

11. **Save and run the script.**

The preceding script creates a list of field objects using the `ListFields()` function. The names of these field objects are placed in a new empty list using the append method. The validated name of the new field is compared to this list of field names using an `if <var> not in` statement.

The `CreateUniqueName()` function also can be used to ensure a name for a new feature class or a table is unique, but it is limited to unique names in a workspace. It cannot be used to ensure a field name is unique.

12. **In IDLE, create a new script file, and save your script as unique_name.py to the \Ex08 folder.**

13. **Enter the following code:**

```
import arcpy

arcpy.env.workspace = "C:/PythonPro/Ex08"

fc = "airports.shp"

unique_name = arcpy.CreateUniqueName("buffer.shp")

arcpy.Buffer_analysis(fc, unique_name, "5000 METERS")
```

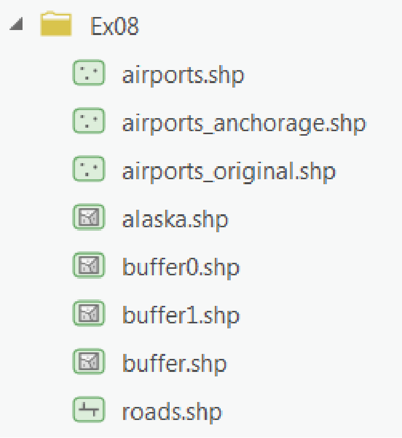
14. **Save and run the script.**

15. **In ArcGIS Pro, confirm that a new feature class named buffer.shp has been created.**

16. **Return to IDLE, and run the same script again.**

17. **In ArcGIS Pro, confirm that a new feature class named buffer0.shp has been created.**

If you don't close and open ArcGIS Pro, the new feature class is not immediately visible. Right-click on the \Ex08 folder in the Catalog pane, and click Refresh. If you keep running the script again, the next output files will be named buffer1.shp, buffer2.shp, and so on.



Ex08
airports.shp
airports_anchorage.shp
airports_original.shp
alaska.shp
buffer0.shp
buffer1.shp
buffer.shp
roads.shp

Using the `CreateUniqueName()` function can prevent accidentally overwriting files.

End of exercise 8.