

Practice 12: Update and Insert Cursors

Objectives:

- Utilize cursor to update and insert records in the vector feature class.

Datasets:

Data folder

MOcity.shp

point shapefile of Missouri cities

MOcnty.shp

polygon shapefile of Missouri counties

MOHigherEd.shp

point shapefile of Missouri College

Scripts folder

folder to store the scripts

Data Source:

Esri

Part I: Updating records with the update cursor

Create a new field and populate its value with the update cursor.

This script creates a new field for Full Address for each feature; combine the values from four different fields (Address, City, State, ZIP), and then assign the combined value into the new field as Full Address.

```
1  import arcpy
2
3  inputFC = r"c:\worker\c32641\Module4\Practice12\Data\MOHigherEd.shp"
4  fieldNew = "FullAddr"
5  fieldList = ["Address", "City", "State", "ZIP", fieldNew]
6
7  arcpy.env.workspace = arcpy.Describe(inputFC).path
8  feat_Class = arcpy.Describe(inputFC).file
9
10 fieldNew = arcpy.ValidateFieldName(fieldNew)
11
12 fields = arcpy.ListFields(feat_Class, fieldNew)
13 for fld in fields:
14     arcpy.DeleteField_management(feat_Class, fieldNew)
15
16 arcpy.AddField_management(feat_Class, fieldNew, "text", "50")
17
```

```

18     count = 1
19     with arcpy.da.UpdateCursor(feats_class, fieldList) as upCursor:
20         for row in upCursor:
21             strAddress = row[0]
22             strCity = row[1]
23             strState = row[2]
24             strZip = row[3]
25
26             strFullAdr = strAddress + ", " + strCity + ", " \
27                           + strState + strZip
28             row[4] = strFullAdr
29             upCursor.updateRow(row)
30
31             print("Update record number: {}".format(count))
32             count = count + 1
33
34     print("Update complete")
35     del upCursor

```

Code Explanations:

1. Lines 3 – 5

Assign the parameters into variables.

- The **fieldNew** variable is the name of the new field being created.
- The **fieldList** variable is the list of fields that will be accessed through the cursor.

2. Lines 7 – 8

Retrieve the file path of the feature class and assign it as the workspace

arcpy.env.workspace. Retrieve the file name and assign it to the feature class variable **feats_class**.

3. Line 10

Use the **ValidateFieldName()** function to check whether the value of the variable **fieldNew** is a valid field name in ArcGIS.

- ArcGIS does not like some special character in the field name, so it is best to check whether the assigned field name satisfied ArcGIS' requirement.

Python Syntax

```
ValidateFieldName (name, {workspace})
```

ArcGIS Pro Help: <https://pro.arcgis.com/en/pro-app/latest/arcpy/functions/validatefieldname.htm>

4. Lines 12 – 14

Check whether the field name already exists in the feature class attribute table. If yes, delete the field. If not, skip to the next line.

- There is no `exists()` function in `arcpy` module to check the existence of field name, so the script check whether there is the field name as the variable `fieldNew` with `ListFields()` function.

Python Syntax

```
arcpy.management.DeleteField(in_table, drop_field)
```

ArcGIS Pro Help: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/delete-field.htm>

5. Line 16

Add the new field into the input feature class attribute table.

Python Syntax

```
arcpy.management.AddField(in_table, field_name, field_type,  
{field_precision}, {field_scale}, {field_length}, {field_alias},  
{field_is_nullable}, {field_is_required}, {field_domain})
```

ArcGIS Pro Help: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/add-field.htm>

6. Line 18

Initial the counting variable `count` as 1.

7. Line 19

Combine `WITH` statement with cursor creation to avoid the data lock. Create an update cursor over the input feature class with only the field listed in `fieldList`.

8. Lines 20 – 32

Create a `FOR` loop to iterate the search cursor over the feature class `feat_class` and print out the value in the field. The script will exit the loop when there is no more record.

- Lines 21 – 24 ➔ Retrieve the attribute values from the input feature class (**Address, City, State, Zip**).

- Lines 26 – 27 ➔ Combine the attribute values into a string as full address. Note that I use the line connector “\” to separate the expression into two lines.
- Line 28 ➔ Assign the value of full address into the new field.
- Line 29 ➔ Use the `updateRow()` function to commit the edit
- Line 31 ➔ Print out the message for updating record
- Line 32 ➔ Increase the count by 1.

9. Line 34
Print out the message for the completion of update.

10. Line 35
Delete the update cursor.

Tasks

1. In the Python IDE, type the above codes into the window. Change the `inputFC` variable setting to correspond with the path name in your computer. Save the script as `..\Practice12\Scripts\Practice12-1.py`.
2. Run the script.
 - a) Make sure that you did not have ArcGIS open with **MOHigherEd.shp** in the Map view.
 - b) If you receive the follow error message (or similar message), it means the **MOHigherEd.shp** is locked by other applications (most likely, it is ArcGIS). Close ArcGIS, then re-run the script again.

arcgisscripting.ExecuteError: ERROR 000464: Cannot get exclusive schema lock. Either being edited or in use by another application or service.
Failed to execute (xxxxxx).

3. Fix any bug and add comments for the script.
 - a) To verify the result, start ArcGIS and open the attribute table of **MOHigherEd.shp**, there should be a new field named **FullAddr** with the full address.

MOHigherEd						
Field: Add Calculate Selection: Select By Attributes Zoom To Switch Clear Delete Copy						
City	State	ZIP	Loc_Code	Latitude	Longitude	FullAddr
St. Louis	MO	63103	GEO	38.63251	-90.225341	3026 Laclede Avenue, St. Louis, MO63103
Jefferson City	MO	65101	MAP	38.564865	-92.168809	820 Chestnut, Jefferson City, MO65101
Joplin	MO	64801-1595	MAP	37.095908	-94.461824	3950 E. Newman Road, Joplin, MO64801-1595
St. Joseph	MO	64507-2294	MAP	39.757852	-94.785452	4525 Downs Drive, St. Joseph, MO64507-2294

b) Remember to close ArcGIS before you run another script.

Part II: Use insert cursor to add a record into feature class

This script use an insert cursor to create two new points in the dataset and assign attribute value into their fields. Besides creating the insert cursor, this script also creates geometric object (the point) whose X and Y coordinates can be assigned by the users (they are currently hard-coding but you should be able to change them as parameters). Once create the geometric object, it will need to be written into the special field for geometry. This field is usually called **SHAPE** and accessed through **arcpy.da** cursor with the token **SHAPE@XY**.

```

1  import arcpy
2
3  inputFC = r"c:\worker\c32641\Module4\Practice12\Data\MOHigherEd.shp"
4  fieldList = ["TYPE", "SHAPE@XY"]
5
6  arcpy.env.workspace = arcpy.Describe(inputFC).path
7  feat_Class = arcpy.Describe(inputFC).file
8
9  # initialize the variables for x, y coordinate and one attribute
10 row_values = [('Pseudo', (400000, 4400000)),
11               ('Pseudo', (400000, 4300000))]
12
13 isCursor = arcpy.da.InsertCursor(feat_Class, fieldList)
14 for newUniversity in row_values:
15     isCursor.insertRow(newUniversity)
16     print("Add New University")
17
18 del isCursor

```

Code Explanations:

1. Lines 3 – 4

Assign the parameters into variables.

- The `fieldList` variable is the list of fields that will be accessed through the cursor.
- `fieldList[0]` is "`TYPE`", which is the attribute field needed to be updated once the new point is added into the feature class.
- `fieldList[1]` is "`SHAPE@XY`", which is XY coordinates of the new points.

2. Lines 6 – 7

Retrieve the file path of the feature class and assign it as the workspace `arcpy.env.workspace`. Retrieve the file name and assign it to the feature class variable `feat_class`.

3. Lines 10 - 11

Initialize the list with the attribute and the geometric information of the new point.

- `row_values[]` store the list of point information
 - Each point is represented by a tuple, e.g., (`Pseudo`, (`400000`, `4400000`)), as its attribute information and XY coordinates.

4. Line 13

Create an insert cursor over the input feature class over the input feature class with only the field listed in `fieldList`.

- This is an example to illustrate how to create a cursor without `WITH` statement. However, I strongly recommend that you use the `WITH` statement for the cursor creation as shown in other examples.

5. Lines 14 – 16

Create a `FOR` loop to iterate each point in the list `row_values`, and then use `insertRow()` function to add the new point into the dataset.

- `newUniversity` is each record in the list `row_values`.

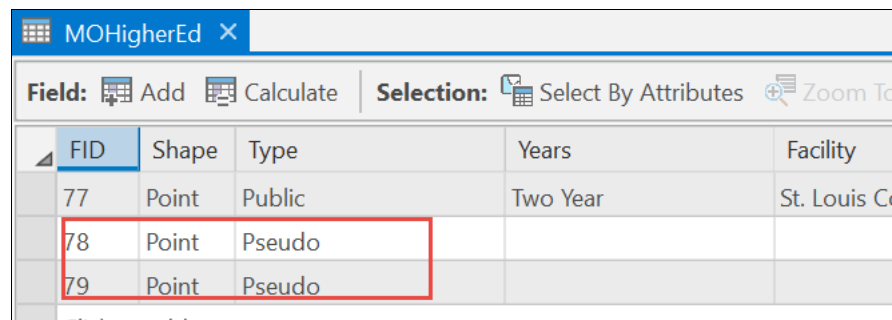
6. Line 18

Delete the cursor.

- It is very important to remember to delete the cursor when you did not combine `WITH` statement with the cursor creation.

Tasks

1. In the Python editor *Script* window, type the above codes into the window. Change the **inputFC** variable setting to correspond with the path name in your computer. Save the script as **..\Practice12\Scripts\Practice12-2.py**.
2. Run the script.
 - a) Make sure that you did not have ArcGIS open with **MOHigherEd.shp** in the Map view.
 - b) If you receive the error message, remember to check whether the data is used by other applications.
3. Fix any bug and add comments for the script.
 - a) To verify the result, start ArcGIS and open the attribute table of **MOHigherEd.shp**, there should be two new universities with Type Pseudo.



FID	Shape	Type	Years	Facility
77	Point	Public	Two Year	St. Louis C
78	Point	Pseudo		
79	Point	Pseudo		

Turn In:

- Compress the scripts as a zip file for submission.
 - There should 2 scripts files.