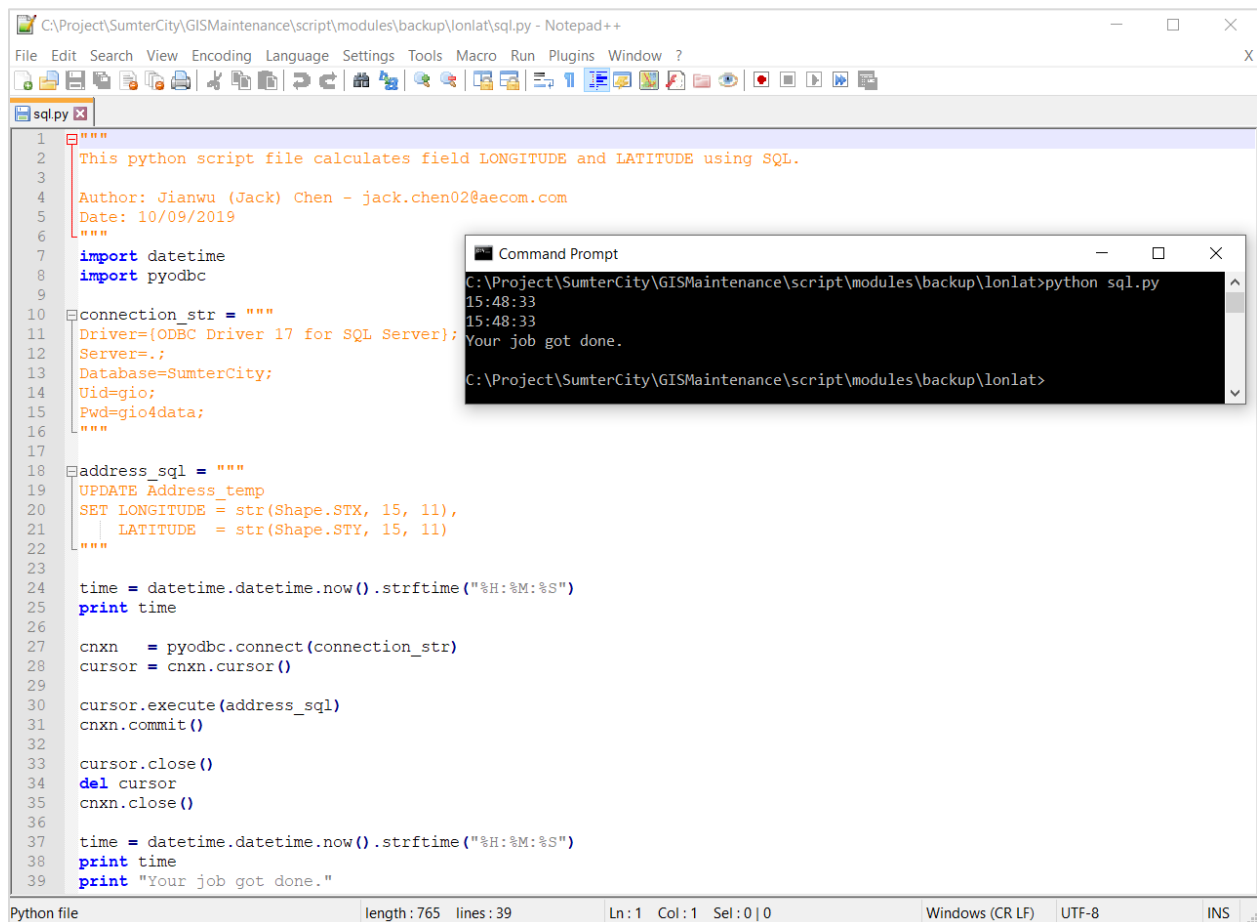


# GP tool versus Cursor versus SQL

AECOM, 11/23/19

**Summary:** field longitude and latitude in a point feature class with 58,036 points was calculated using three kinds of approaches: geoprocessing tool, cursor, and SQL. The quickest approach is SQL, 400 times faster than other two approaches.

Approach	Time
SQL	Less than 1 second
Cursor	7 minutes 56 seconds
Geoprocessing tool	8 minutes 10 seconds



```
1 """
2 This python script file calculates field LONGITUDE and LATITUDE using SQL.
3
4 Author: Jianwu (Jack) Chen - jack.chen02@aecom.com
5 Date: 10/09/2019
6 """
7 import datetime
8 import pyodbc
9
10 connection_str = """
11 Driver={ODBC Driver 17 for SQL Server};
12 Server=.;
13 Database=SumterCity;
14 Uid=gio;
15 Pwd=gio4data;
16 """
17
18 address_sql = """
19 UPDATE Address_temp
20 SET LONGITUDE = str(Shape.STX, 15, 11),
21     LATITUDE = str(Shape.STY, 15, 11)
22 """
23
24 time = datetime.datetime.now().strftime("%H:%M:%S")
25 print time
26
27 cnxn = pyodbc.connect(connection_str)
28 cursor = cnxn.cursor()
29
30 cursor.execute(address_sql)
31 cnxn.commit()
32
33 cursor.close()
34 del cursor
35 cnxn.close()
36
37 time = datetime.datetime.now().strftime("%H:%M:%S")
38 print time
39 print "Your job got done."
```

```
C:\Project\SumterCity\GISMaintenance\script\modules\backup\lonlat>python sql.py
15:48:33
15:48:33
Your job got done.
C:\Project\SumterCity\GISMaintenance\script\modules\backup\lonlat>
```

Python file length: 765 lines: 39 Ln: 1 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

```
C:\Project\SumterCity\GISMaintenance\script\modules\backup\lonlat\cursor.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
cursor.py
1 """
2 This python script file calculates field LONGITUDE and LATITUDE using cursor.
3
4 Author: Jianwu (Jack) Chen - jack.chen02@aecom.com
5 Date: 11/18/2019
6 """
7 import datetime
8 import arcpy
9
10 workspace = r"C:\Project\SumterCity\GISMaintenance\data\database_connection\gio.SumterCity.sde"
11 arcpy.env.workspace = workspace
12 arcpy.env.overwriteOutput = True
13
14 time = datetime.datetime.now().strftime("%H:%M:%S")
15 print time
16
17 address_fc = "Address_temp"
18 with arcpy.da.UpdateCursor(address_fc, ["LONGITUDE", "LATITUDE", "SHAPE@X", "SHAPE@Y"]) as cursor:
19     for row in cursor:
20         row[0] = str(row[2])
21         row[1] = str(row[3])
22         cursor.updateRow(row)
23
24 time = datetime.datetime.now().strftime("%H:%M:%S")
25 print time
26 print "Your job got done."
```

Command Prompt

```
C:\Project\SumterCity\GISMaintenance\script\modules\backup\lonlat>python cursor.py
15:35:09
15:43:05
Your job got done.
C:\Project\SumterCity\GISMaintenance\script\modules\backup\lonlat>
```

Python file length: 803 lines: 26 Ln: 2 Col: 62 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

```
C:\Project\SumterCity\GISMaintenance\script\modules\backup\lonlat\gptool.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
gptool.py
1 """
2 This python script file calculates field LONGITUDE and LATITUDE using geoprocessing tool.
3
4 Author: Jianwu (Jack) Chen - jack.chen02@aecom.com
5 Date: 11/18/2019
6 """
7 import datetime
8 import arcpy
9
10 workspace = r"C:\Project\SumterCity\GISMaintenance\data\database_connection\gio.SumterCity.sde"
11 arcpy.env.workspace = workspace
12 arcpy.env.overwriteOutput = True
13
14 time = datetime.datetime.now().strftime("%H:%M:%S")
15 print time
16
17 address_fc = "Address_temp"
18 geometry_properties = "POINT X Y Z M"
19 arcpy.AddGeometryAttributes_management(address_fc, geometry_properties)
20
21 time = datetime.datetime.now().strftime("%H:%M:%S")
22 print time
23 print "Your job got done."
```

Command Prompt

```
C:\Project\SumterCity\GISMaintenance\script\modules\backup\lonlat>python gptool.py
15:00:09
15:08:19
Your job got done.
C:\Project\SumterCity\GISMaintenance\script\modules\backup\lonlat>
```

Python file length: 725 lines: 23 Ln: 1 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS