In [12]:

```python
#Overall: new database connection information
#Section1:
# Define the spatial reference (WGS 84) for the geometry
import arcpy
spatial_reference = arcpy.SpatialReference(4326)
# Define three points to create a polygon: creates a polygon using arcpy geometry prim
point1 = arcpy.Point(-93.2650, 44.9778)
point2 = arcpy.Point(-93.1780, 45.0569)
point3 = arcpy.Point(-93.0900, 44.9537)
# Create an array of points
array = arcpy.Array([point1, point2, point3])
# Create a polygon using the array of points
polygon = arcpy.Polygon(array)
#section 2
#Convert to WKT (allows geometry to viewed as text and copied up to our database)
polygon_wkt = polygon.WKT

#Section3
#Import psycopg2 module for interacting with PostgreSQL database
import psycopg2
from psycopg2 import sql
#Define database connection parameters
database_key = 'loganandgreg'
conn = psycopg2.connect(
    dbname="gis5572",
    user="postgres",
    password=database_key,
    host="35.232.21.197",
    port="5432"
)
#Create a cursor object to execute SQL commands
cur = conn.cursor()

#Creates table in SQL-- to talk to database
#Define your SQL query to insert the polygon
insert_query = sql.SQL("""
    CREATE TABLE lab1 (
    id SERIAL PRIMARY KEY,
    shape GEOMETRY(MultiPolygon, 4326)
    );
    """)

#Execute the SQL query to create the table
cur.execute(insert_query)
#adds shape into database and polygon into the table
#%s is pasting polygon text WKT
insert_query = sql.SQL("""
    INSERT INTO lab1(shape)
    VALUES (ST_GeomFromText(%s, 4326));
    """)
##Execute the SQL query to insert the polygon, providing the WKT as a parameter
cur.execute(insert_query, [polygon_wkt])
#Commit the transaction to save changes to the database
conn.commit()
#Close the cursor and database connection
```

```
cur.close()
conn.close()
```

In [ ]: