psycopg2

```
%pyspark                                                              FINISHED
# Make sure psycopg2 is installed:
import psycopg2
import json
from psycopg2 import IntegrityError, InternalError
import datetime
```
Took 0 sec. Last updated by anonymous at May 08 2017, 4:59:57 PM.

```
%pyspark                                                              FINISHED
# Load data from S3
raw_quakes = sc.wholeTextFiles("s3a://bucketshakesforyou/*")
```
Took 0 sec. Last updated by anonymous at May 08 2017, 5:26:32 PM.

```
%pyspark                                                              FINISHED
test = raw_quakes.collect()
```
Took 7 sec. Last updated by anonymous at May 08 2017, 5:19:23 PM.

```
%pyspark                                                              FINISHED
json.loads(test[0][1]).keys()
```
[u'type', u'features', u'bbox', u'metadata']

Took 0 sec. Last updated by anonymous at May 08 2017, 5:20:35 PM.

```
%pyspark                                                              FINISHED
raw_quakes.take(1)
```

[(u's3a://bucketshakesforyou/2017-05-05', u'{"type":"FeatureCollection","metadata":{"generated":14940288
62000,"url":"https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2017-05-05&endtim
e=2017-05-06","title":"USGS Earthquakes","status":200,"api":"1.5.7","count":274},"features":[{"type":"Fe
ature","properties":{"mag":2,"place":"83km WNW of Skagway, Alaska","time":1494028123519,"updated":149402
8703147,"tz":-480,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/ak15930044","detail":"http
s://earthquake.usgs.gov/fdsnws/event/1/query?eventid=ak15930044&format=geojson","felt":null,"cdi":nul
l,"mmi":null,"alert":null,"status":"automatic","tsunami":0,"sig":62,"net":"ak","code":"15930044","id
s":",ak15930044,","sources":",ak,","types":",geoserve,origin,","nst":null,"dmin":null,"rms":0.78,"gap":n
ull,"magType":"ml","type":"earthquake","title":"M 2.0 - 83km WNW of Skagway, Alaska"},"geometry":{"typ
e":"Point","coordinates":[-136.6783,59.7576,0]},"id":"ak15930044"},\n{"type":"Feature","properties":{"ma
g":1.29,"place":"4km NNE of Kernville, California","time":1494027851380,"updated":1494028059359,"tz":-48
0,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/ci37640999","detail":"https://earthquake.usg
s.gov/fdsnws/event/1/query?eventid=ci37640999&format=geojson","felt":null,"cdi":null,"mmi":null,"alert":
null,"status":"automatic","tsunami":0,"sig":26,"net":"ci","code":"37640999","ids":",ci37640999,","source
s":",ci,","types":",geoserve,nearby-cities,origin,phase-data,scitech-link,","nst":16,"dmin":0.111,"rms":
0.13,"gap":89,"magType":"ml","type":"earthquake","title":"M 1.3 - 4km NNE of Kernville, California"},"ge
ometry":{"type":"Point","coordinates":[-118.4115,35.7951667,8.69]},"id":"ci37640999"},\n{"type":"Featur
e","properties":{"mag":2.36,"place":"9km NW of Cobb, California","time":1494027039890,"updated":14940273
```

Took 1 sec. Last updated by anonymous at May 08 2017, 5:00:37 PM.

```
%pyspark                                                              FINISHED
```

```
def get_quakes(quake):
    '''

    INPUT: dict of quakes
    OUTPUT: list of length 6 w/ the following values:
            quake_id
            isotime
            magnitude
            longitude
            latitude
            altitude
    '''

    import datetime
    try:
        quake_id = quake["id"]
        # isotime = quake.get("properties").get("time")
        properties = quake.get("properties")
        if properties:
            epochtime = properties.get("time")  # machine readable timestamp
            isotime = datetime.datetime.utcfromtimestamp(epochtime/1000).isoformat()  # human readable t
            magnitude = properties.get("mag")

        geometry = quake.get("geometry")
        if geometry:
            longitude = geometry.get("coordinates")[0]
            latitude = geometry.get("coordinates")[1]
            altitude = geometry.get("coordinates")[2]

        return [quake_id, isotime, magnitude, longitude, latitude, altitude]

    except ValueError:
        pass
```

Took 0 sec. Last updated by anonymous at May 08 2017, 5:06:50 PM.

```
def insert_rows(text_file):
    '''

    INPUT: None
    OUTPUT: None
        Inserts all tweets into postgres using `get_quakes`
    '''

    import psycopg2
    from psycopg2 import IntegrityError, InternalError
    conn = psycopg2.connect(**{'dbname':          ,
                               'host':                                    ,
                               'password':          ,
                               'user':          })
    cur = conn.cursor()

    # line_string = ''
    # for line in text_file[1:]:
    #     line_string += line

    for file in text_file:
        dictionary = json.loads(file[1])

        for quake in dictionary['features']:
            q = get_quakes(quake)

            try:
                cur.execute("INSERT INTO quakes VALUES (%s,%s,%s,%s,%s,%s)", q)
                conn.commit()

            except (IntegrityError, InternalError) as e:
```

```
        conn.commit()
        conn.close()
```

Took 0 sec. Last updated by anonymous at May 08 2017, 5:26:02 PM.

```
%pyspark
raw_quakes.foreachPartition(insert_rows)
```

FINISHED

Took 17 min 29 sec. Last updated by anonymous at May 08 2017, 5:44:31 PM.

```
%pyspark
```

READY