

Postgresql to Pandas

[Postgresql](#), [to](#), [Pandas](#)

Step 1: Connect to the database

```
import psycopg2
import pandas as pd
# Connection parameters, yours will be different
param_dic = {
    "host"      : "localhost",
    "database"  : "globaldata",
    "user"      : "myuser",
    "password"  : "Passw0rd"
}
def connect(params_dic):
    """ Connect to the PostgreSQL database server """
    conn = None
    try:
        # connect to the PostgreSQL server
        print('Connecting to the PostgreSQL database...')
        conn = psycopg2.connect(**params_dic)
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
        sys.exit(1)
    print("Connection successful")
    return conn
```

Step 2: Get your dataframe

```
def postgresql_to_dataframe(conn, select_query, column_names):
    """
    Tranform a SELECT query into a pandas dataframe
    """
    cursor = conn.cursor()
    try:
        cursor.execute(select_query)
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error: %s" % error)
        cursor.close()
        return 1
    # Naturally we get a list of tuples
    tuples = cursor.fetchall()
    cursor.close()
    # We just need to turn it into a pandas dataframe
    df = pd.DataFrame(tuples, columns=column_names)
    return df
```

Step 3: Demo

First, this is how the MonthlyTemp table looks like

```
>>> SELECT * FROM MonthlyTemp
   id   | source   | datetime   | mean_temp
-----+-----+-----+-----
 860897 | GCAG     | 2016-12-06 |    0.7895
 860898 | GISTEMP  | 2016-12-06 |    0.81
 860899 | GCAG     | 2016-11-06 |    0.7504
 860900 | GISTEMP  | 2016-11-06 |    0.93
 860901 | GCAG     | 2016-10-06 |    0.7292
```

Demo 1: keeping the original column names

```
# Connect to the database
conn = connect(param_dic)
column_names = ["id", "source", "datetime", "mean_temp"]
# Execute the "SELECT *" query
df = postgresql_to_dataframe(conn, "select * from MonthlyTemp",
column_names)
df.head()
```

Demo 2: selecting some of the columns only, and changing their names

```
# Connect to the database
conn = connect(param_dic)
column_names = ["timestamp", "temperature"]
# Execute the "SELECT *" query
df = postgresql_to_dataframe(conn, "select datetime, mean_temp from
MonthlyTemp", column_names)
df.head()
```

close

```
# Close the connection
conn.close()
```

- <https://naysan.ca/2020/05/31/postgresql-to-pandas/>

Plugin Backlinks:

From:

<https://jace.link/> - **Various Ways**

Permanent link:

<https://jace.link/open/postgresql-to-pandas>

Last update: **2020/11/02 14:20**

