Connecting to your own RDB on AWS!

Now that you've followed the lesson on setting up a Postgres database on AWS it's time to connect to it!

Below we're just going to load in our library. I already have our trusty `get_conn_cur()` function. You just need to modify to connect you your database. **To do this just enter in your host, database name, username, and password.**

After that you can run the function and store your connection and cursor. Check them out after to make sure you're connected!

After completing this notebook, run all cells **(except the drop table cell)**, and create pdf, and upload it as the answer to Review Quiz 5.

```
import psycopg2
import pandas as pd


def get_conn_cur(): # define function name and arguments (there aren't any)
  # Make a connection
  conn = psycopg2.connect(
    host="test-hw-db.chb3guhvlmeq.us-west-2.rds.amazonaws.com",
    database="hw3_db",
    user="postgres",
    password="uW5uK$4xcBNvKL",
    port='5432')

  cur = conn.cursor()   # Make a cursor after

  return(conn, cur)   # Return both the connection and the cursor


# Get conn and cursor object
conn, cur = get_conn_cur()


# Check it!
conn
```

```
<connection object at 0x7f5bded7d6e0; dsn: 'user=postgres password=xxx
dbname=hw3_db host=test-hw-db.chb3guhvlmeq.us-west-2.rds.amazonaws.com
port=5432', closed: 0>
```

Go and get your SQL functions

✓ 0s completed at 10:06 AM

data were inserted.

```
# put functions here and run
# Same run_query function
def run_query(query_string):

  conn, cur = get_conn_cur() # get connection and cursor

  cur.execute(query_string) # executing string as before

  my_data = cur.fetchall() # fetch query data as before

  # here we're extracting the 0th element for each item in cur.description
  colnames = [desc[0] for desc in cur.description]

  cur.close() # close
  conn.close() # close

  return(colnames, my_data) # return column names AND data

# Column name function for checking out what's in a table
def get_column_names(table_name): # arguement of table_name
  conn, cur = get_conn_cur() # get connection and cursor

  # Now select column names while inserting the table name into the WERE
  column_name_query =  """SELECT column_name FROM information_schema.columns
       WHERE table_name = '%s' """ %table_name

  cur.execute(column_name_query) # exectue
  my_data = cur.fetchall() # store

  cur.close() # close
  conn.close() # close

  return(my_data) # return

# Check table_names
def get_table_names():
  conn, cur = get_conn_cur() # get connection and cursor

  # query to get table names
  table_name_query = """SELECT table_name FROM information_schema.tables
       WHERE table_schema = 'public' """

  cur.execute(table_name_query) # execute
  my_data = cur.fetchall() # fetch results

  cur.close() #close cursor
```

```
  conn.close() # close connection

  return(my_data) # return your fetched results
```

Now make a table and push some data over - 10 points

We're going to learn how to do this next week, but for now just run these cells to make a table in your database and then push a row of data over.

This practice is all or nothing as it needs to work for credit.

```
# Drop table
# *** Only run this code if you've already created your table and want to delete it
#
conn, cur = get_conn_cur()
table_name = 'profs' # what table to drop
drop_table_statement = "DROP TABLE %s;"%table_name # make your statement
cur.execute(drop_table_statement) # execute it
conn.commit()
cur.close() # close your cursor
```

```
# make table
conn, cur = get_conn_cur()
tq = """CREATE TABLE profs (
          prof_id INTEGER PRIMARY KEY,
          prof_age INTEGER NOT NULL ,
          prof_last_name VARCHAR(255) NOT NULL
          );"""
cur.execute(tq)
conn.commit()
```

```
# push a row
iq = """INSERT INTO profs (prof_id, prof_age, prof_last_name) VALUES(23097100, 37,
conn, cur = get_conn_cur()
cur.execute(iq)
conn.commit()
```

```
# Now select the rows in the profs table. You should see one row with my info
# After running all cells upload the pdf of your notebook for Review Quiz 5
sql = "select * from profs"
run_query(sql)
```

```
    (['prof_id', 'prof_age', 'prof_last_name'], [(23097100, 37, 'riazi')])
```

Colab paid products - Cancel contracts here