# BMGT 404 Homework 5

## What to turn in

Turn in your Python codes through ELMS by **10:00AM on Wednesday, April 12th**. Please make sure your code is easy to read.

## Learning objectives:

- File operations
- Database operations

## Software requirement:

MySQL server
Python package: MySQLdb

## Task:

Please write two Python functions to build a MySQL table (doctors) in your database and insert all records from a text file: doctors.txt. Columns are separated by commas. (**Note: some specialty contains a comma, your codes should be able to handle this.**)

(10 points) def createTable(tablename):
    // your codes here

(30 points) def insertRecordsFromFile(filename, tablename):
    // your codes here

**Five sample records:**
1, Dr. David A. Warkentin, Chiropractor, 296, Mesa, AZ
2, Dr. Fawad S. Zafar, Urologist, 97, CLIVE, IA
3, Dr. Pedram A. Hendizadeh, Podiatrist, 69, FAIRFIELD, CT
4, Dr. William Parker, Gynecologist (OBGYN), 83, Los Angeles, CA
5, Dr. William H. Romero, Family Doctor / G.P., 110, Dix Hills, NY

The table should have the schema similar to the following.

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| docID      | int(11)     | NO   | PRI | NULL    |       |
| name       | varchar(50) | YES  |     | NULL    |       |
| specialty  | varchar(50) | YES  |     | NULL    |       |
| numReviews | int(11)     | YES  |     | NULL    |       |
| city       | tinytext    | YES  |     | NULL    |       |
| state      | varchar(2)  | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

**Fields:**
docID: doctor ID (primary key: must be unique)
name: doctor name
specialty: doctor specialty
numReviews: number of reviews on rateMDs
city: city where doctor lives
state: sate where doctor lives

Once the table is built and records are inserted, please write Python functions to answer the following questions.

1. Find the total number of doctors, the average number of reviews for a given state. (Use select statement with some functions like count() and avg()) (20 points)

   ```
   def findTotalAndAvgNumbers(state):
           // your codes here

           return (total, average)
   ```

2. Print out name and specialty about the top ***n*** doctors in terms of number of reviews for a given state. (Use select statement combined with order by) (20 points)

   ```
   def getTopNDoctors(state, n):
           // your codes here
   ```

3. For doctors without city or state information (denoted by a hyphen: '-' in the text file), please update their city or state to be: U. (Use update statement) (20 points)

   ```
           def updateCityState():
                   // your codes here
   ```

<u>**You can use the following code to test your functions.**</u>

```
def main():
        tab_name = raw_input("Please name a table to create: ")
        print "creating table", tab_name
        createTables(tab_name)
        filename = raw_input("Which file you want to load data? ")
        print "inserting data from", filename
        insertRecordsFromFile(filename, tab_name)

        print "getting total # of doctors and average # of reviews"
        state = raw_input("Which state?")
```

```
    print “getting numbers for state:”, state
    total, avg = findTotalAndAvgNumbers(md)
    print “total:”,total, “average:”, avg

    print “getting top doctors in terms of # of reviews”
    state = raw_input(“Which state?”)
    n = int(raw_input(“How many doctors to show? ”))
    getTopNDoctors(state, n)

    print “updating city and state”
    updateCityStae()


if __name__ == ‘__main__’:
    main()
```