

# CS185C: NoSQL Team Project

## MangoDB - Intermediate Report

### Team Members

1. Kushal Khandelwal
2. Ketki Kulkarni
3. Adil Khan

### Title of Project

Analysis of College Scorecards

### Data Wrangling

Entire dataset is spanning nearly 20 years of data and covering multiple sources including IPEDS, NSLDS, and Department of Treasury. The dataset is split across multiple CSV's ranging from 1990 to 2015. For now, we are only considering the latest data i.e. for the year 2014-15. Later however, we do plan to integrate all the segregated data into a single CSV.

Data contains around 1743 columns and covers a wide range of topics related to colleges. UNITID is primary key and represents Unit ID for institution. There's also a OPE ID for institution, in form of 6 and 8-digit. Some schools report these data at the campus level (8-digit OPE ID), which are then rolled up to the institution level (6-digit OPE ID). Other major data columns are INSTNM for Institution name, SATVR\_\_ for SAT percentile, PCIP\_\_ denoting percentage of degrees awarded in particular field of study, etc. We have based our queries on UNITID and OPEID (8-digit) and have created index over it.

### Importing Data

The data can be easily imported into MongoDB as the data is available in the CSV format.

```
mongoimport --db collegeTemp --collection collegeData --type csv --file  
MERGED2014_15 --headerline
```

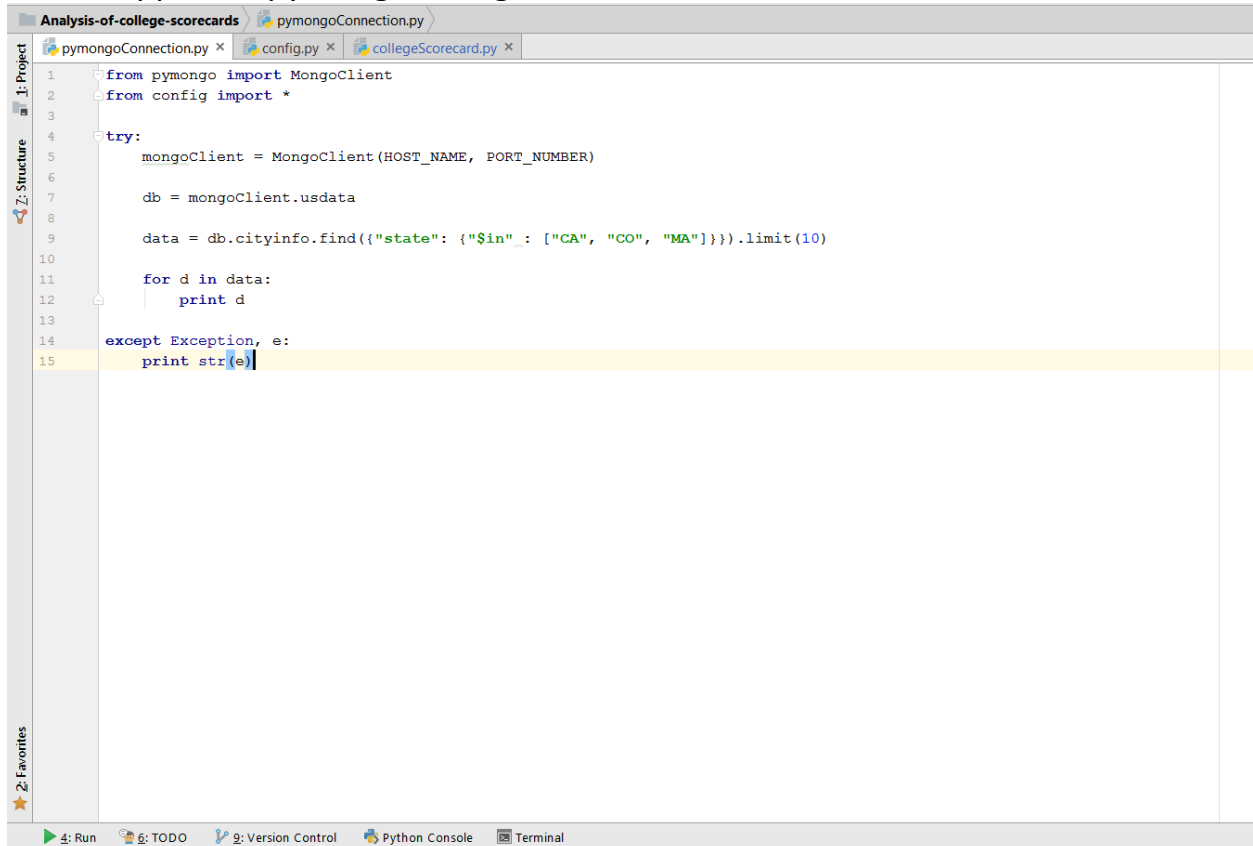
# CS185C: NoSQL Team Project

## MangoDB - Intermediate Report

### API Coding

We did the API programming in Python (3.4) and used the *PyMongo* library to connect to the MongoDB database. *PyMongo* is a Python distribution containing tools for working with MongoDB, and is the recommended way to work with MongoDB from Python.

### Code snippet for pymongo-MongoDb connection



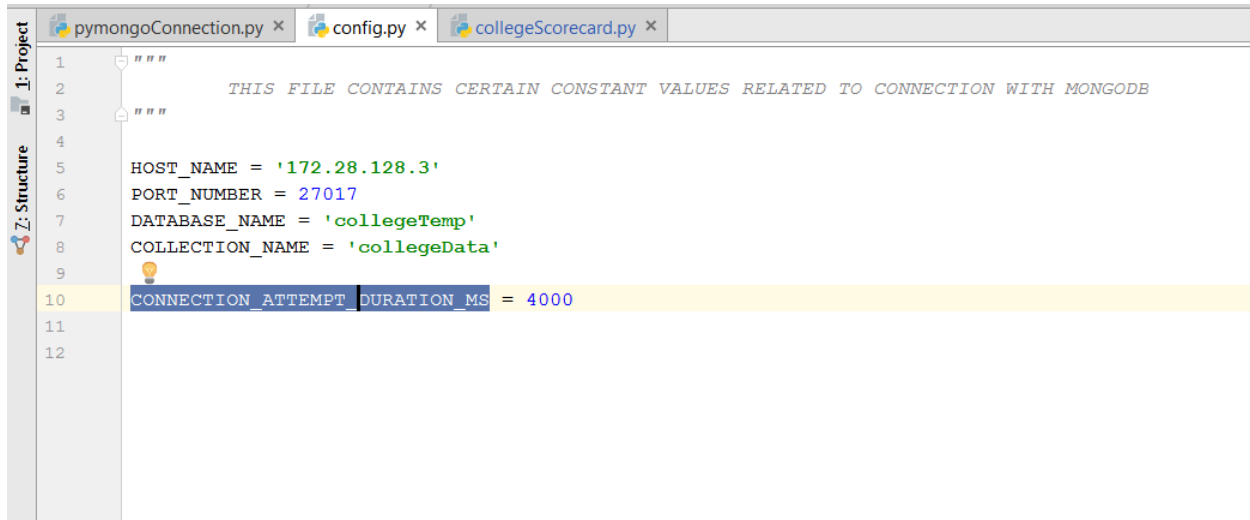
```
Analysis-of-college-scorecards > pymongoConnection.py
pymongoConnection.py x config.py x collegeScorecard.py x
1 from pymongo import MongoClient
2 from config import *
3
4 try:
5     mongoClient = MongoClient(HOST_NAME, PORT_NUMBER)
6
7     db = mongoClient.usdata
8
9     data = db.cityinfo.find({"state": {"$in": ["CA", "CO", "MA"]}}).limit(10)
10
11     for d in data:
12         print d
13
14 except Exception, e:
15     print str(e)
```

The screenshot shows an IDE window titled "Analysis-of-college-scorecards" with three tabs: "pymongoConnection.py", "config.py", and "collegeScorecard.py". The "pymongoConnection.py" tab is active, displaying a Python script. The script imports MongoClient from pymongo and imports all variables from config. It then enters a try block where it creates a MongoClient instance with HOST\_NAME and PORT\_NUMBER, connects to the 'usdata' database, and queries the 'cityinfo' collection for records where the state is in ['CA', 'CO', 'MA'], limiting the results to 10. It iterates over the results and prints each document. An except block catches any Exception and prints its string representation. The IDE interface includes a sidebar with "1: Project", "2: Structure", and "3: Favorites" views, and a bottom status bar with icons for Run, TODO, Version Control, Python Console, and Terminal.

# CS185C: NoSQL Team Project

## MangoDB - Intermediate Report

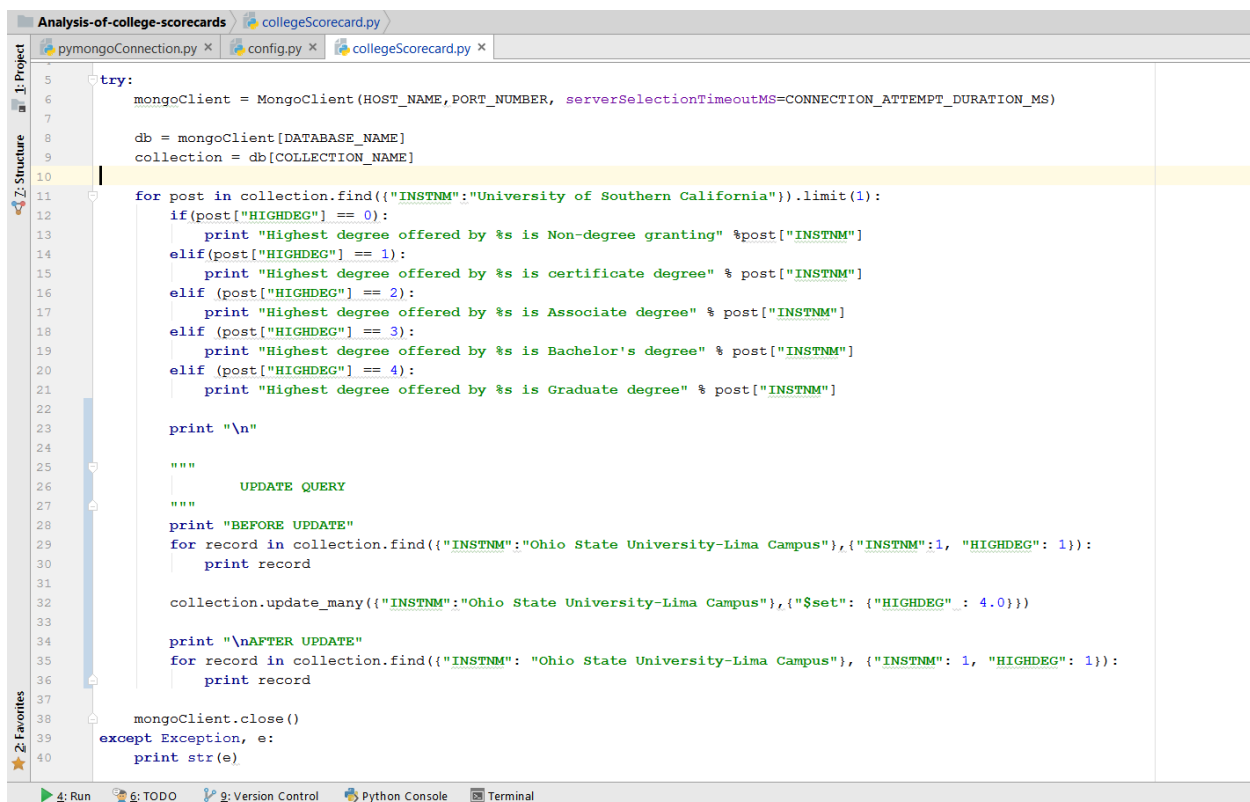
Configuration file specifying host name, port number, DB name, collection name and timeout in milliseconds



The screenshot shows a code editor with three tabs: 'pymongoConnection.py', 'config.py', and 'collegeScorecard.py'. The 'config.py' tab is active, displaying a Python file with constant values for MongoDB connection. The code is as follows:

```
1  """
2      THIS FILE CONTAINS CERTAIN CONSTANT VALUES RELATED TO CONNECTION WITH MONGODB
3  """
4
5  HOST_NAME = '172.28.128.3'
6  PORT_NUMBER = 27017
7  DATABASE_NAME = 'collegeTemp'
8  COLLECTION_NAME = 'collegeData'
9
10 CONNECTION_ATTEMPT_DURATION_MS = 4000
11
12
```

Code snippet showing queries to actual dataset



The screenshot shows a code editor with three tabs: 'pymongoConnection.py', 'config.py', and 'collegeScorecard.py'. The 'collegeScorecard.py' tab is active, displaying a Python script that connects to a MongoDB database and performs queries. The code is as follows:

```
5  try:
6      mongoClient = MongoClient(HOST_NAME, PORT_NUMBER, serverSelectionTimeoutMS=CONNECTION_ATTEMPT_DURATION_MS)
7
8      db = mongoClient[DATABASE_NAME]
9      collection = db[COLLECTION_NAME]
10
11  for post in collection.find({"INSTNM": "University of Southern California"}).limit(1):
12      if (post["HIGHDEG"] == 0):
13          print "Highest degree offered by %s is Non-degree granting" % post["INSTNM"]
14      elif (post["HIGHDEG"] == 1):
15          print "Highest degree offered by %s is certificate degree" % post["INSTNM"]
16      elif (post["HIGHDEG"] == 2):
17          print "Highest degree offered by %s is Associate degree" % post["INSTNM"]
18      elif (post["HIGHDEG"] == 3):
19          print "Highest degree offered by %s is Bachelor's degree" % post["INSTNM"]
20      elif (post["HIGHDEG"] == 4):
21          print "Highest degree offered by %s is Graduate degree" % post["INSTNM"]
22
23  print "\n"
24
25  """
26      UPDATE QUERY
27  """
28  print "BEFORE UPDATE"
29  for record in collection.find({"INSTNM": "Ohio State University-Lima Campus"}, {"INSTNM": 1, "HIGHDEG": 1}):
30      print record
31
32  collection.update_many({"INSTNM": "Ohio State University-Lima Campus"}, {"$set": {"HIGHDEG": 4.0}})
33
34  print "\nAFTER UPDATE"
35  for record in collection.find({"INSTNM": "Ohio State University-Lima Campus"}, {"INSTNM": 1, "HIGHDEG": 1}):
36      print record
37
38  mongoClient.close()
39  except Exception, e:
40      print str(e)
```

# CS185C: NoSQL Team Project

## MangoDB - Intermediate Report

### Schema Description

To improve efficiency, we added two indexes to the collections. One on the “UNITID” key and the other on the “OPEID” key as both are identification fields and are unique for every Institution.

Here is a screenshot to show that the indexes were successfully created.

```
vagrant@vagrant-ubuntu-trusty-64: ~
> db.collegeData.createIndex({"UNITID" : 1}, {name: "Unique identification number"})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.collegeData.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "collegeTemp.collegeData"
  },
  {
    "v" : 2,
    "key" : {
      "UNITID" : 1
    },
    "name" : "Unique identification number",
    "ns" : "collegeTemp.collegeData"
  }
]
```

```
vagrant@vagrant-ubuntu-trusty-64: ~
> db.collegeData.createIndex({"OPEID":1}, {"name":"8-digit OPE(Office of Postsecondary education) ID for institution"})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.collegeData.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "collegeTemp.collegeData"
  },
  {
    "v" : 2,
    "key" : {
      "OPEID" : 1
    },
    "name" : "8-digit OPE(Office of Postsecondary education) ID for institution",
    "ns" : "collegeTemp.collegeData"
  }
]
```

# CS185C: NoSQL Team Project

## MangoDB - Intermediate Report

### Queries

1. Find the number of undergraduate students enrolled at San Jose State University.

```
db.collegeData.find({INSTNM:"San Jose State University"},{UGDS:1,_id:0});
```

```
adilkhani — vagrant@vagrant-ubuntu-trusty-64: ~ — ssh -p 2222 vagrant@127...
> db.collegeData.find({INSTNM:"San Jose State University"},{UGDS:1,_id:0});
{ "UGDS" : 26528 }
>
```

2. List all colleges with a 100% graduation rate.

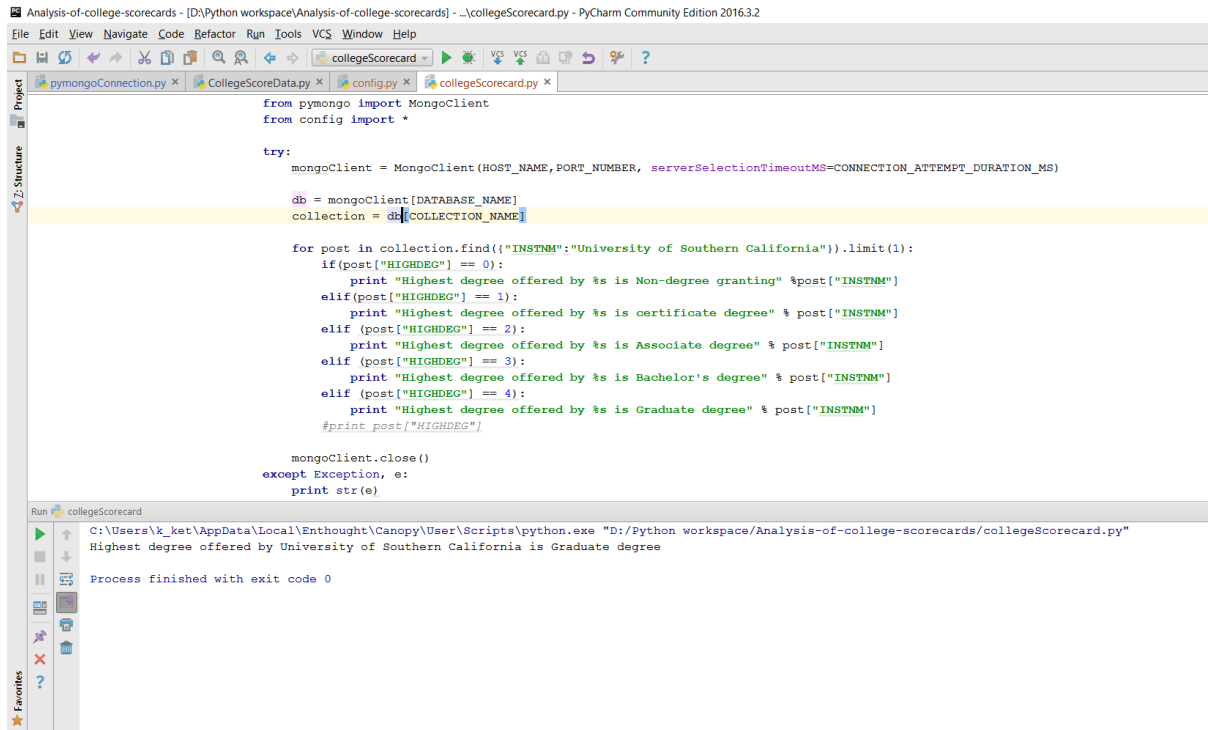
```
db.collegeData.find({"C150_4":{ $gt: 0.99 }},{"INSTNM":1,"C150_4":1})
```

```
> db.collegeData.find({"C150_4":{ $gt: 0.99 }},{"INSTNM":1,"C150_4":1})
{ "_id" : ObjectId("58cf43192114c4cea65bbdbd"), "INSTNM" : "Southern California Seminary", "C150_4" : 1 }
{ "_id" : ObjectId("58cf431a2114c4cea65bbdbd59"), "INSTNM" : "University of Southernmost Florida", "C150_4" : 1 }
{ "_id" : ObjectId("58cf431b2114c4cea65bbbf86"), "INSTNM" : "St Luke's College", "C150_4" : 1 }
{ "_id" : ObjectId("58cf431b2114c4cea65bc053"), "INSTNM" : "Saint Joseph Seminary College", "C150_4" : 1 }
{ "_id" : ObjectId("58cf431c2114c4cea65bc24e"), "INSTNM" : "Cleveland University-Kansas City", "C150_4" : 1 }
{ "_id" : ObjectId("58cf431c2114c4cea65bc26a"), "INSTNM" : "Kenrick Glennon Seminary", "C150_4" : 1 }
{ "_id" : ObjectId("58cf431d2114c4cea65bc409"), "INSTNM" : "Helene Fuld College of Nursing", "C150_4" : 1 }
{ "_id" : ObjectId("58cf431e2114c4cea65bc54f"), "INSTNM" : "Carolina Christian College", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43222114c4cea65bce12"), "INSTNM" : "Argosy University-Washington DC", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43222114c4cea65bce35"), "INSTNM" : "Argosy University-Schaumburg", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43232114c4cea65bcfe9"), "INSTNM" : "Pacific College of Oriental Medicine-Chicago", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43232114c4cea65bcff6"), "INSTNM" : "Family of Faith College", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43242114c4cea65bd12d"), "INSTNM" : "Careers Unlimited", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43242114c4cea65bd193"), "INSTNM" : "University of the West", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43242114c4cea65bd24d"), "INSTNM" : "Chamberlain College of Nursing-Illinois", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43252114c4cea65bd335"), "INSTNM" : "Pacific Rim Christian University", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43252114c4cea65bd52b"), "INSTNM" : "Academy of Couture Art", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43262114c4cea65bd5e4"), "INSTNM" : "Mid-South Christian College", "C150_4" : 1 }
{ "_id" : ObjectId("58cf43262114c4cea65bd63f"), "INSTNM" : "DeVry University-Maryland", "C150_4" : 1 }
```

# CS185C: NoSQL Team Project

## MangoDB - Intermediate Report

### 3. Find Highest degree awarded by a university. (PyMongo)



```
Analysis-of-college-scorecards - [D:\Python workspace\Analysis-of-college-scorecards] - \collegeScorecard.py - PyCharm Community Edition 2016.3.2
File Edit View Navigate Code Refactor Run Tools VCS Window Help

collegeScorecard.py x CollegeScoreData.py x config.py x collegeScorecard.py x

from pymongo import MongoClient
from config import *

try:
    mongoClient = MongoClient(HOST_NAME, PORT_NUMBER, serverSelectionTimeoutMS=CONNECTION_ATTEMPT_DURATION_MS)

    db = mongoClient[DATABASE_NAME]
    collection = db[COLLECTION_NAME]

    for post in collection.find({"INSTNM": "University of Southern California"}).limit(1):
        if (post["HIGHDEG"] == 0):
            print "Highest degree offered by %s is Non-degree granting" % post["INSTNM"]
        elif (post["HIGHDEG"] == 1):
            print "Highest degree offered by %s is certificate degree" % post["INSTNM"]
        elif (post["HIGHDEG"] == 2):
            print "Highest degree offered by %s is Associate degree" % post["INSTNM"]
        elif (post["HIGHDEG"] == 3):
            print "Highest degree offered by %s is Bachelor's degree" % post["INSTNM"]
        elif (post["HIGHDEG"] == 4):
            print "Highest degree offered by %s is Graduate degree" % post["INSTNM"]
        #print post["HIGHDEG"]

    mongoClient.close()
except Exception, e:
    print str(e)
```

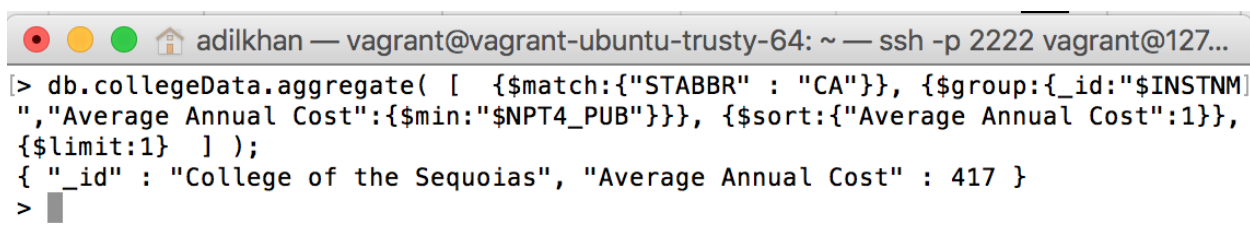
Run collegeScorecard

```
C:\Users\k_ket\AppData\Local\Enthought\Canopy\User\Scripts\python.exe "D:/Python workspace/Analysis-of-college-scorecards/collegeScorecard.py"
Highest degree offered by University of Southern California is Graduate degree

Process finished with exit code 0
```

### 4. Find the college in the state of California with the least average annual cost.

```
db.collegeData.aggregate(
[
    {$match: {"STABBR" : "CA"}},
    {$group: {_id: "$INSTNM", "Average Annual Cost": {$min: "$NPT4_PUB"}}},
    {$sort: {"Average Annual Cost": 1}},
    {$limit: 1}
]
);
```



```
adilkhan — vagrant@vagrant-ubuntu-trusty-64: ~ — ssh -p 2222 vagrant@127...

[> db.collegeData.aggregate( [ { $match: { "STABBR" : "CA" } }, { $group: { _id: "$INSTNM", "Average Annual Cost": { $min: "$NPT4_PUB" } } }, { $sort: { "Average Annual Cost": 1 } }, { $limit: 1 } ] );
{ "_id" : "College of the Sequoias", "Average Annual Cost" : 417 }
>
```

# CS185C: NoSQL Team Project

## MangoDB - Intermediate Report

- Find the Avg. SAT score for a particular college.

```
db.collegeData.find({"INSTNM":"Massachusetts Maritime Academy"},{"INSTNM":1,"SAT_AVG":1})
```

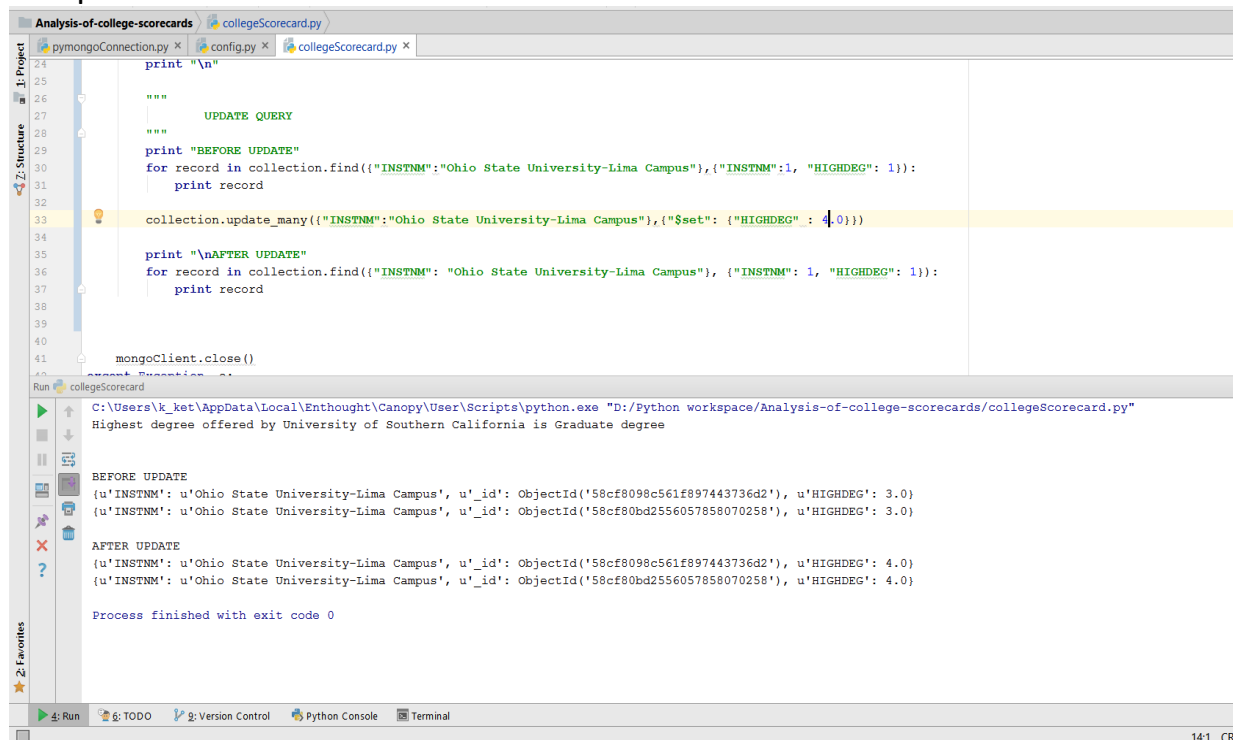
```
> db.collegeData.find({"INSTNM":"Massachusetts Maritime Academy"}, {"INSTNM":1, "SAT_AVG":1})
{ "_id" : ObjectId("58cf431c2114c4cea65bc10f"), "INSTNM" : "Massachusetts Maritime Academy", "SAT_AVG" : 1105 }
```

- Find the Admission rate for a college.

```
db.collegeData.find({"INSTNM":"Stanford University"}, {"INSTNM":1, "ADM_RATE":1})
```

```
> db.collegeData.find({"INSTNM":"Stanford University"}, {"INSTNM":1, "ADM_RATE":1})
{ "_id" : ObjectId("58cf43202114c4cea65bcb13"), "INSTNM" : "Stanford University", "ADM_RATE" : 0.0509 }
```

- Updating the highest degree offered by the 'Ohio State University-Lima Campus' from Bachelor to Graduate.



```
Analysis-of-college-scorecards collegeScorecard.py
pymongoConnection.py x config.py x collegeScorecard.py x
24 print "\n"
25
26
27     """
28     UPDATE QUERY
29     """
30     print "BEFORE UPDATE"
31     for record in collection.find({"INSTNM":"Ohio State University-Lima Campus"}, {"INSTNM":1, "HIGHDEG": 1}):
32         print record
33
34     collection.update_many({"INSTNM":"Ohio State University-Lima Campus"}, {"$set": {"HIGHDEG": 4}})
35
36     print "\nAFTER UPDATE"
37     for record in collection.find({"INSTNM": "Ohio State University-Lima Campus"}, {"INSTNM": 1, "HIGHDEG": 1}):
38         print record
39
40
41     mongoClient.close()
42
Run collegeScorecard
C:\Users\k_ket\AppData\Local\Enthought\Canopy\User\Scripts\python.exe "D:\Python workspace\Analysis-of-college-scorecards\collegeScorecard.py"
Highest degree offered by University of Southern California is Graduate degree

BEFORE UPDATE
{'INSTNM': 'Ohio State University-Lima Campus', '_id': ObjectId('58cf8098c561f897443736d2'), 'HIGHDEG': 3.0}
{'INSTNM': 'Ohio State University-Lima Campus', '_id': ObjectId('58cf80bd2556057858070258'), 'HIGHDEG': 3.0}

AFTER UPDATE
{'INSTNM': 'Ohio State University-Lima Campus', '_id': ObjectId('58cf8098c561f897443736d2'), 'HIGHDEG': 4.0}
{'INSTNM': 'Ohio State University-Lima Campus', '_id': ObjectId('58cf80bd2556057858070258'), 'HIGHDEG': 4.0}

Process finished with exit code 0
```