

Annand_Module07_lab01

February 24, 2025

0.1 Import Pymongo and Connect to Cluster

```
[1]: #import pymongo
import pymongo

#create your connection string
connect_string = "mongodb+srv://annandj:kitkatbaka@cluster0.mmpel.mongodb.net/?
↳retryWrites=true&w=majority&appName=Cluster0"
#create a connection to your Atlas cluster
client = pymongo.MongoClient(connect_string)

#connect to the sample_restaraunts database
restaurants_db = client.sample_restaurants
```

0.2 Exercise 1

Create a mongo_db connection with pymongo to your database
<https://pymongo.readthedocs.io/en/stable/examples/authentication.html>

For the homework we will be using the sample_restaurants.restaurants collection.

Using find(), write a find query to extract the Italian restaurants in Manhattan to a Python list. Use len() to count the number of restaurants located in Manhattan.

Note All MongoDB functions and fields MUST be in quotes inside of the find() method. Ex *andshouldbe*"and".

https://www.w3schools.com/python/python_mongodb_find.asp

```
[2]: ex_1_cursor = restaurants_db.restaurants.find({"$and":[{"cuisine":
↳"Italian"}, {"borough": "Manhattan"}]})
ex_1_list = list(ex_1_cursor)

print(len(ex_1_list))
```

621

0.3 Exercise 2

Using find, determine how many Japanese and Italian restaurants have an A rating in Queens.

```
[3]: ex_2_cursor = restaurants_db.restaurants.find({"$and":[{"$or":[{"cuisine":
    ↪ "Italian"}, {"cuisine": "Japanese"}]}, {"borough": "Queens"}, {"grades":
    ↪ {"$elemMatch":{"grade": "A"}}]})
ex_2_list = list(ex_2_cursor)

print(len(ex_2_list))
```

240

0.4 Exercise 3

The following MongoDB aggregation query is missing a aggregation expression that will calculate the BSON size of the documents. A list of these can be found at the end of this week's notes. Identify the missing aggregation expression. Print the 10 document ids and sizes that have the highest BSON size.

```
[4]: res = restaurants_db.restaurants.aggregate([
    { "$addFields": {
        "bsonsize": { "$bsonSize": "$$ROOT" } #use expression bsonSize
    }},
    { "$sort": { "bsonsize": -1 }}, #sort by descending BSON size
    { "$limit": 10 }, #take only the first 10 docs which should be the 10 with
    ↪ the greatest BSON size
    { "$project": {
        "_id": 1, #set to 1 to return _id
        "bsonsize": 1 #set to 1 to return bsonsize
    }}
])

ex_3_list = list(res)

for doc in ex_3_list:
    print(doc.items())
```

```
dict_items([(' _id', ObjectId('5eb3d669b31de5d588f4561a')), ('bsonsize', 694)])
dict_items([(' _id', ObjectId('5eb3d668b31de5d588f43b95')), ('bsonsize', 684)])
dict_items([(' _id', ObjectId('5eb3d669b31de5d588f45195')), ('bsonsize', 681)])
dict_items([(' _id', ObjectId('5eb3d668b31de5d588f44257')), ('bsonsize', 679)])
dict_items([(' _id', ObjectId('5eb3d668b31de5d588f42eed')), ('bsonsize', 677)])
dict_items([(' _id', ObjectId('5eb3d668b31de5d588f436af')), ('bsonsize', 677)])
dict_items([(' _id', ObjectId('5eb3d668b31de5d588f42c95')), ('bsonsize', 677)])
dict_items([(' _id', ObjectId('5eb3d668b31de5d588f43367')), ('bsonsize', 669)])
dict_items([(' _id', ObjectId('5eb3d669b31de5d588f452c7')), ('bsonsize', 662)])
dict_items([(' _id', ObjectId('5eb3d669b31de5d588f45859')), ('bsonsize', 661)])
```

0.5 Exercise 4

Find all of the restaurants that have NOT had an 'A', 'B', and 'Not Yet Graded' rating. How many restaurants is this?

```
[5]: ex_4_cursor = restaurants_db.restaurants.find({"grades.grade":{"$nin":  
    ↪ ["A","B","Not Yet Graded"]}}) #use dot notation to access embedded docs in_  
    ↪ grades  
ex_4_list = list(ex_4_cursor)  
  
print(len(ex_4_list))
```

948

```
[6]: client.close()
```