Annand_DSE6210_lab6

February 22, 2025

0.1 Joseph Annand DSE 6210 Lab 6

0.2 Import pymongo

[1]:

```
import pymongo
from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi
```

0.3 Exercise 1

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

[2]:

```
#https://pymongo.readthedocs.io/en/stable/examples/authentication.html
#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
try:
    client = MongoClient(connect_string, server_api=ServerApi('1'))
except:
    print("Connection failed :/")
```

0.4 Exercise 2

Using your client created from exercise 1, connect to a new database, homework6.
Once you have connected to the database set your collection to a new collection, students.
https://pymongo.readthedocs.io/en/stable/tutorial.html –> getting database and getting collection

[3]:

```
db = client.homework6
collection = db.students
```

0.5 Exercise 3

I have created a list of student data containing documents that you will need to insert into MongoDB.
Using insert_many, insert the list of documents into the students collection.

[4]:

```
student_data = [

    {"instructor":"Martin",
     "class":"Chemistry",
     "max_students":25,
     "term":"SP2",
     "students":["Bob Mackey","George Straight","Bill Cowher","Stanley␣
↪Kubrick",'Martin Sheen',"Charlize Theron"]},
    {"instructor":"Lowhorn",
     "class":"Big Data",
     "max_students":10,
     "term":"SU1",
     "students":["Charles Barkely","Charlie Sheen","Tina Turner","Paul␣
↪Walker",'Dwayne Johnson',"Courtney Cox", "Margot Robbie"]},
    {"instructor":"Carlin",
     "class":"Discrete Math",
     "max_students":25,
     "term":"SP2",
     "students":["Tim Couch","George Straight","Michael Douglas","Peyton␣
↪Manning",'Wade Boggs',"Doc Rivers","Drew Bledsoe","Ray Bourque"]},
    {"instructor":"Lowhorn",
     "class":"Programming for DS",
     "max_students":25,
     "term":"SP2",
     "students":["Roger Clemens","Ray Allen","Marcus Smart","Kevin Garnett",'Mo␣
↪Vaughn',"Uma Thurman","Conan O'Brien","Mark Wahlberg"]},
    ]

collection.insert_many(student_data)
```

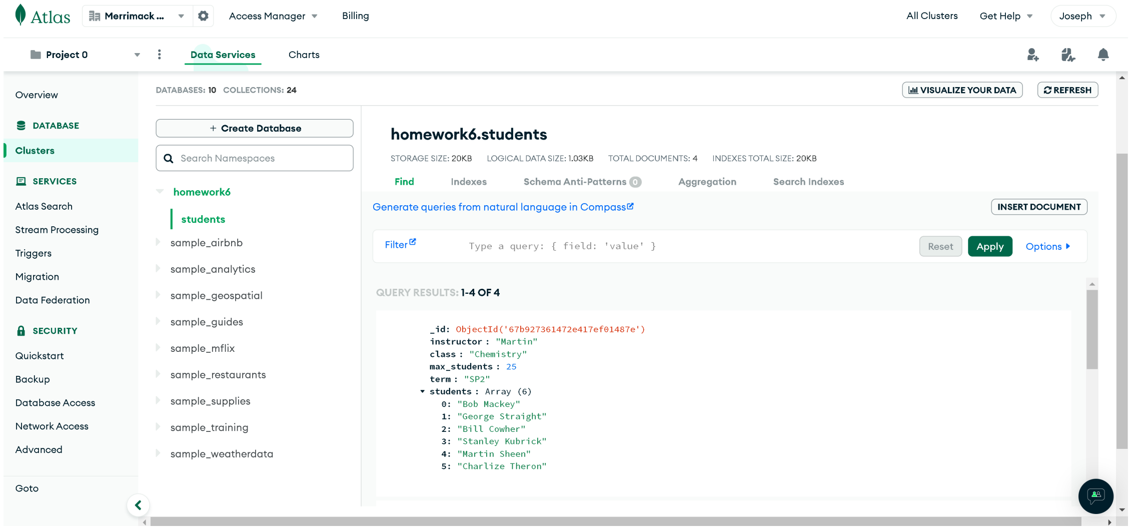
[4]:

```
InsertManyResult([ObjectId('67b9ecc0bb0218368e2ddee8'),
ObjectId('67b9ecc0bb0218368e2ddee9'), ObjectId('67b9ecc0bb0218368e2ddeea'),
ObjectId('67b9ecc0bb0218368e2ddeeb')], acknowledged=True)
```

0.6 Exercise 4

What MongoDB type do Python lists get converted to?
Submit a screen shot of your collection in MongoDB with this python file.

Python lists are converted to Arrays.



```
_id: ObjectId('67b927361472e417ef01487e')
instructor : "Martin"
class : "Chemistry"
max_students : 25
term : "SP2"
▾ students : Array (6)
    0: "Bob Mackey"
    1: "George Straight"
    2: "Bill Cowher"
    3: "Stanley Kubrick"
    4: "Martin Sheen"
    5: "Charlize Theron"
```

0.7 Exercise 5

George Straight accidentally registered for two courses in the SP2 Session.
Using a pymongo.update(), remove him from Carlin's class
Note: Your key is instructor.
Use the $pull method to extract the element from the array.
https://www.geeksforgeeks.org/python-mongodb-update_one/
https://www.mongodb.com/docs/manual/reference/operator/update/pull/

[5]:
```
filter = {'instructor': 'Carlin'}
collection.update_one(filter, { "$pull": { "students" : "George Straight"  } })
```

[5]:
```
UpdateResult({'n': 1, 'electionId': ObjectId('7fffffff0000000000000177'),
'opTime': {'ts': Timestamp(1740238012, 26), 't': 375}, 'nModified': 1, 'ok':
1.0, '$clusterTime': {'clusterTime': Timestamp(1740238012, 26), 'signature':
{'hash': b'T&nz4\x19i\xd0\xf1S7HX\xf8#\x8f\xacU\x88\x0c', 'keyId':
7427919520641056769}}, 'operationTime': Timestamp(1740238012, 26),
'updatedExisting': True}, acknowledged=True)
```

0.8 Exercise 6

A new student has signed up for all three SP2 sessions, his name is Tom Brady.
Update the SP2 classes by inserting the student Tom Brady into the students object.
Note: Many not one. Push not pull.

[6]:
```
filter = { "term": "SP2" }
collection.update_many(filter, { "$push": { "students": "Tom Brady"  } })
```

[6]:
```
UpdateResult({'n': 3, 'electionId': ObjectId('7fffffff0000000000000177'),
'opTime': {'ts': Timestamp(1740238012, 35), 't': 375}, 'nModified': 3, 'ok':
1.0, '$clusterTime': {'clusterTime': Timestamp(1740238012, 35), 'signature':
{'hash': b'T&nz4\x19i\xd0\xf1S7HX\xf8#\x8f\xacU\x88\x0c', 'keyId':
```

```
7427919520641056769}}, 'operationTime': Timestamp(1740238012, 35),
'updatedExisting': True}, acknowledged=True)
```

0.9 Exercise 7

The college has decided that Chemistry was not a good fit for the data science program. Delete it from the collection.
https://www.geeksforgeeks.org/python-mongodb-delete_one/

[7]:
```
filter = { "class": "Chemistry" }
collection.delete_one(filter)
```

[7]:
```
DeleteResult({'n': 1, 'electionId': ObjectId('7fffffff0000000000000177'),
'opTime': {'ts': Timestamp(1740238012, 37), 't': 375}, 'ok': 1.0,
'$clusterTime': {'clusterTime': Timestamp(1740238012, 37), 'signature': {'hash':
b'T&nz4\x19i\xd0\xf1S7HX\xf8#\x8f\xacU\x88\x0c', 'keyId': 7427919520641056769}},
'operationTime': Timestamp(1740238012, 37)}, acknowledged=True)
```

0.10 Exercise 8

Using find, print all of the documents to the console. This should be a query against the MongoDB database.

[8]:
```
for x in collection.find():
    print(x)
```

```
{'_id': ObjectId('67b9ecc0bb0218368e2ddee9'), 'instructor': 'Lowhorn', 'class':
'Big Data', 'max_students': 10, 'term': 'SU1', 'students': ['Charles Barkely',
'Charlie Sheen', 'Tina Turner', 'Paul Walker', 'Dwayne Johnson', 'Courtney Cox',
'Margot Robbie']}
{'_id': ObjectId('67b9ecc0bb0218368e2ddeea'), 'instructor': 'Carlin', 'class':
'Discrete Math', 'max_students': 25, 'term': 'SP2', 'students': ['Tim Couch',
'Michael Douglas', 'Peyton Manning', 'Wade Boggs', 'Doc Rivers', 'Drew Bledsoe',
'Ray Bourque', 'Tom Brady']}
{'_id': ObjectId('67b9ecc0bb0218368e2ddeeb'), 'instructor': 'Lowhorn', 'class':
'Programming for DS', 'max_students': 25, 'term': 'SP2', 'students': ['Roger
Clemens', 'Ray Allen', 'Marcus Smart', 'Kevin Garnett', 'Mo Vaughn', 'Uma
Thurman', "Conan O'Brien", 'Mark Wahlberg', 'Tom Brady']}
```

0.11 Exercise 9

Instead of using the default hash _id, what would you recommend as a unique ID for each document?

I'd recommend using the instructor, class, and term fields together to create a new unique ID for each document. Assuming that a single instructor does not teach two sections of the same course ina given term, each instructor-class-term combination may act as a unique identifier for each document. A compound unique index based on these three fields could be created.

[9]:
```
collection.create_index([("instructor", pymongo.ASCENDING), ("class", pymongo.
 ↪ASCENDING), ("term", pymongo.ASCENDING)], unique=True)
```

[9]:
```
'instructor_1_class_1_term_1'
```

0.12 Exercise 10

Drop the students collection from the database AND close your client.

[10]:
```
collection.drop()
client.close()
```