Tutorial 7
Python Driver for MongoDB

The main objective of this tutorial is to understand how MongoDB Driver works, particularly by using the Python Driver (pymongo).

MongoDB Driver

1. Install pymongo driver in the command prompt/terminal
   If python is installed then the following can be used:
   ```
   python -m pip install pymongo
   ```

   [Note: If there is an error saying : no module named pip then it could be that you are using an anaconda python installation and therefore the following can be used:
   ```
   conda install -c anaconda pymongo
   ```

2. Then run python in the terminal to start the python shell
   ```
   python
   ```

3. The pymongo driver can now be imported using
   ```
   import pymongo
   ```

4. Now we can connect to the localhost where MongoDB server is running using
   ```
   client = pymongo.MongoClient('localhost', 27017)
   ```
   or
   ```
   client = pymongo.MongoClient("mongodb://localhost:27017/")
   ```

5. Next we can create a database called 'tutorial5Db' if it doesn't already exist
   ```
   db = client['tutorial7Db']
   ```

   If the database is created properly then the following command should run without errors:
   ```
   db
   ```

6. Now we can create our first collection called 'tutorial5Collection'
   ```
   collection = db['tutorial7Collection']
   ```

   And count the number of documents in the newly created collection using:
   ```
   collection.count()
   ```

7. Let's try adding some data. For this we can first create a python dictionary called ‘student1_data’, ‘student2_data’, ‘student3_data’ store the fields name and age:

   ```
   student1_data = {
    'name':'student1',
    'age':25
   }

   student2_data = {
   ```

```
 'name':'student2',
 'age':20
}

student3_data = {
 'name':'student3',
 'age':23
}
```

We can see the contents by just calling data and see if the fields were correctly created.

```
student1_data
student2_data
student3_data
```

8. Next we can insert the 'data' to the collection as a document:

   ```
   insert_result = collection.insert_one(student1_data)
   ```

   [Notice how the syntax is slightly different from the MongoShell syntax since we are using python commands]

   We can check if the insert happened correctly by using:

   ```
   insert_result.acknowledged
   ```

   To check the id of the inserted document, we can use:

   ```
   insert_result.inserted_id
   ```

   For student2

   ```
   insert_result = collection.insert_one(student2_data)
   insert_result.acknowledged
   insert_result.inserted_id
   ```

   For student3

   ```
   insert_result = collection.insert_one(student3_data)
   insert_result.acknowledged
   insert_result.inserted_id
   ```

9. Now let's try counting again (if the previous count returned zero, then this could should return 3.)

   ```
   collection.count()
   ```

10. To find and display one document:

    ```
     x = collection.find_one()
    print(x)
    ```

    To find more than one document

    ```
    for x in collection.find():
      print(x)
    ```

    (need to click enter twice)

11. To query the collection:

```
query = { "name": "student1" }

result = collection.find(query)

for x in result:
  print(x)
```

12. Using aggreate pipeline:

```
pipeline_query = [
    {"$match": { "age": { "$gte": 20}}},
    {"$sort": { "age": pymongo.ASCENDING }}
]

results = collection.aggregate(pipeline_query)

for x in results:
  print(x)
```

13. Updating a document e.g. student2's age to 27:

```
condition_query = { "name": "student2" }
set_values = { "$set": { "age": "27" } }

collection.update_one(condition_query, set_values)
```

Checking for the change

```
query = { "name": "student2" }

result = collection.find(query)

for x in result:
  print(x)
```

14. Now let's get rid of our test document by deleting the document

```
delete_result = collection.delete_one({'age':25})

collection.count()
```

15. Drop the collection:

```
collection.drop()
```

16. You save your python code in a runnable python script using the following at the start of the script:

```
!/usr/bin/env python27
Followed by your python code and saved in a .py file
```

References:
https://www.w3schools.com/python/python_mysql_where
https://pymongo.readthedocs.io/en/stable/genindex.html

The End