day17_17_feb_nosql_mongodb_database

1 mongoDB connection

- in case of mongodb it stores data in format of documents/JSON file (key:value) pair
- not like SQL, it stores data in a form of table (row/column) >* inside database store table and inside table row and column >* database -> table -> row and column

1.0.1 in mongoDB

- cluster is system with memory and ram, inside cluster will create Database and inside Database is collection, inside collection is JSON file/documents in KEY:VALUE pair >* Database -> collection -> documents/JSON (key:value pair) file

[ ]:

```

```

[3]:

```python
import pymongo
```

1.1 install pymongo with pip(package manager)

- if we create some library we have to host with pip because it let download and install all libraries to user

[2]:

```
pip install pymongo
```

```
Collecting pymongo
  Downloading
pymongo-4.3.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (492
kB)
                          492.9/492.9 kB
17.1 MB/s eta 0:00:00
Collecting dnspython<3.0.0,>=1.16.0
  Downloading dnspython-2.3.0-py3-none-any.whl (283 kB)
                          283.7/283.7 kB
38.4 MB/s eta 0:00:00
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.3.0 pymongo-4.3.3
Note: you may need to restart the kernel to use updated packages.
```

[8]:

```
import pymongo

from pymongo.mongo_client import MongoClient # without importing MongoClient it␣
 ↪won't work

client = MongoClient("mongodb+srv://pwskills:pwskills1@cluster0.xe9xplu.mongodb.
 ↪net/?retryWrites=true&w=majority")
db = client.test
```

[9]:

```
client
```

[9]:

```
MongoClient(host=['ac-5s8jdzq-shard-00-00.xe9xplu.mongodb.net:27017',
'ac-5s8jdzq-shard-00-01.xe9xplu.mongodb.net:27017', 'ac-5s8jdzq-
shard-00-02.xe9xplu.mongodb.net:27017'], document_class=dict, tz_aware=False,
connect=True, retrywrites=True, w='majority', authsource='admin',
replicaset='atlas-jho4jc-shard-0', tls=True)
```

[ ]:

```

```

Need to create ID on MongoDB website and

- on Atlas tab create a Database Deployment
- then click on connect and get connected server link

[ ]:

```
# copied server link from mongoDb site

mongodb+srv://pwskills:<password>@cluster0.xe9xplu.mongodb.net/
 ↪?retryWrites=true&w=majority
# replace password with entered password at Database creating time
```

1.2 mongodb connection

- copied from website

[11]:

```
#copied DB connection from MongoDB hosting website

from pymongo.mongo_client import MongoClient

uri = "mongodb+srv://pwills:pwlls1@cluster0.xelu.mongodb.net/?
 ↪retryWrites=true&w=majority"

# Create a new client and connect to the server
client = MongoClient(uri)

# Send a ping to confirm a successful connection
try:
    client.admin.command('ping')
    print("Pinged your deployment. You successfully connected to MongoDB!")
except Exception as e:
    print(e)
```

```
Pinged your deployment. You successfully connected to MongoDB!
```

[12]:
```
client #authsource = admin, so we have admin control on database
```

[12]:
```
MongoClient(host=['ac-5s8jdzq-shard-00-00.xe9xplu.mongodb.net:27017',
'ac-5s8jdzq-shard-00-01.xe9xplu.mongodb.net:27017', 'ac-5s8jdzq-
shard-00-02.xe9xplu.mongodb.net:27017'], document_class=dict, tz_aware=False,
connect=True, retrywrites=True, w='majority', authsource='admin',
replicaset='atlas-jho4jc-shard-0', tls=True)
```

1.3 create database

- named pwskills_db

[13]:
```
db = client['pwskills_db']
```

1.4 create collection

- named 'record_collection'

[14]:
```
collection_create = db["record_collection"]
```

[16]:
```
data = {'name' : "reshep",
        "class" : "data science masters",
        "timing" : "flexible"
}
```

1.5 insert JSON data

[17]:
```
collection_create.insert_one(data)
```

[17]:
```
<pymongo.results.InsertOneResult at 0x7fe0a4086c20>
```

[ ]:
```
data1 = {"mail_id" : "res@gmail.com".
         "phone" : 4345345
        }
```

[ ]:
```
collection_create.insert(data1)
```

[21]:
```
data2 = {'list_course' : ['data science masters ', 'web dev', "java with DSA"],
         "mentor" : ["steve jobs", "bill gates", 'albert einstein']
        }
```

[22]:
```
collection_create.insert_one(data2)
```

[22]:
```
<pymongo.results.InsertOneResult at 0x7fe08f912170>
```

[23]:
```
 data3 = [
  { "name": "Amy", "address": "Apple st 652" },
  { "name": "Hannah", "address": "Mountain 21" },
  { "name": "Michael", "address": "Valley 345" },
  { "name": "Sandy", "address": "Ocean blvd 2" },
  { "name": "Betty", "address": "Green Grass 1" },
  { "name": "Richard", "address": "Sky st 331" },
  { "name": "Susan", "address": "One way 98" },
  { "name": "Vicky", "address": "Yellow Garden 2" },
  { "name": "Ben", "address": "Park Lane 38" },
  { "name": "William", "address": "Central st 954" },
  { "name": "Chuck", "address": "Main Road 989" },
  { "name": "Viola", "address": "Sideway 1633" }
]
```

insert_one for single dictionary data

- insert_many is for many dictionary data

[24]:
```
collection_create.insert_many(data3)
```

[24]:
```
<pymongo.results.InsertManyResult at 0x7fe08f912350>
```

[25]:
```
data4 = {
 "name": "notebook",
 "qty": 50,
 "rating": [ { "score": 8 }, { "score": 9 } ],
 "size": { "height": 11, "width": 8.5, "unit": "in" },
 "status": "A",
 "tags": [ "college-ruled", "perforated"]
}
```

[26]:
```
collection_create.insert_one(data4)
```

[26]:
```
<pymongo.results.InsertOneResult at 0x7fe0a409eb90>
```

[27]:
```
list_of_records = [
    {'companyName': 'iNeuron',
     'product': 'Affordable AI',
     'courseOffered': 'Machine Learning with Deployment'},

    {'companyName': 'iNeuron',
     'product': 'Affordable AI',
     'courseOffered': 'Deep Learning for NLP and Computer vision'},

    {'companyName': 'iNeuron',
     'product': 'Master Program',
     'courseOffered': 'Data Science Masters Program'}
]
```

[28]:
```
collection_create.insert_many(list_of_records)
```

[28]:
```
<pymongo.results.InsertManyResult at 0x7fe08c1a1180>
```

1.6 find all data in database

[29]:
```
for i in collection_create.find():
    print(i)
```

```
{'_id': ObjectId('6455f9b0577bc8b3ad835e76'), 'name': 'reshep', 'class': 'data
science masters', 'timing': 'flexible'}
{'_id': ObjectId('6455fb36577bc8b3ad835e77'), 'mail_id': 'res@gmail.com',
'phone': 4345345}
{'_id': ObjectId('6455fc52577bc8b3ad835e78'), 'list_course': ['data science
masters ', 'web dev', 'java with DSA'], 'mentor': ['steve jobs', 'bill gates',
'albert einstein']}
{'_id': ObjectId('6455fcdc577bc8b3ad835e79'), 'name': 'Amy', 'address': 'Apple
st 652'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7a'), 'name': 'Hannah', 'address':
'Mountain 21'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7b'), 'name': 'Michael', 'address':
'Valley 345'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7c'), 'name': 'Sandy', 'address': 'Ocean
blvd 2'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7d'), 'name': 'Betty', 'address': 'Green
Grass 1'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7e'), 'name': 'Richard', 'address': 'Sky
st 331'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7f'), 'name': 'Susan', 'address': 'One
way 98'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e80'), 'name': 'Vicky', 'address':
'Yellow Garden 2'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e81'), 'name': 'Ben', 'address': 'Park
Lane 38'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e82'), 'name': 'William', 'address':
'Central st 954'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e83'), 'name': 'Chuck', 'address': 'Main
Road 989'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e84'), 'name': 'Viola', 'address':
'Sideway 1633'}
```

```
{'_id': ObjectId('6455fd2f577bc8b3ad835e85'), 'name': 'notebook', 'qty': 50,
'rating': [{'score': 8}, {'score': 9}], 'size': {'height': 11, 'width': 8.5,
'unit': 'in'}, 'status': 'A', 'tags': ['college-ruled', 'perforated']}
{'_id': ObjectId('6455fdc7577bc8b3ad835e86'), 'companyName': 'iNeuron',
'product': 'Affordable AI', 'courseOffered': 'Machine Learning with Deployment'}
{'_id': ObjectId('6455fdc7577bc8b3ad835e87'), 'companyName': 'iNeuron',
'product': 'Affordable AI', 'courseOffered': 'Deep Learning for NLP and Computer
vision'}
{'_id': ObjectId('6455fdc7577bc8b3ad835e88'), 'companyName': 'iNeuron',
'product': 'Master Program', 'courseOffered': 'Data Science Masters Program'}
```

[30]:

```
random_data = [
    {'_id': '3', 'companyName': 'iNeuron', 'Faculty': 'XYZ'},
    {'_id': '4', 'companyName': 'iNeuron', 'Faculty': 'ABC'},
    {'_id': '5', 'companyName': 'iNeuron', 'Faculty': 'PQR'},
]
```

[31]:

```
collection_create.insert_many(random_data)
```

[31]:

```
<pymongo.results.InsertManyResult at 0x7fe08c1a3e50>
```

[32]:

```
for i in collection_create.find():
    print(i)
```

```
{'_id': ObjectId('6455f9b0577bc8b3ad835e76'), 'name': 'reshep', 'class': 'data
science masters', 'timing': 'flexible'}
{'_id': ObjectId('6455fb36577bc8b3ad835e77'), 'mail_id': 'res@gmail.com',
'phone': 4345345}
{'_id': ObjectId('6455fc52577bc8b3ad835e78'), 'list_course': ['data science
masters ', 'web dev', 'java with DSA'], 'mentor': ['steve jobs', 'bill gates',
'albert einstein']}
{'_id': ObjectId('6455fcdc577bc8b3ad835e79'), 'name': 'Amy', 'address': 'Apple
st 652'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7a'), 'name': 'Hannah', 'address':
'Mountain 21'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7b'), 'name': 'Michael', 'address':
'Valley 345'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7c'), 'name': 'Sandy', 'address': 'Ocean
blvd 2'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7d'), 'name': 'Betty', 'address': 'Green
Grass 1'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7e'), 'name': 'Richard', 'address': 'Sky
st 331'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7f'), 'name': 'Susan', 'address': 'One
way 98'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e80'), 'name': 'Vicky', 'address':
'Yellow Garden 2'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e81'), 'name': 'Ben', 'address': 'Park
Lane 38'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e82'), 'name': 'William', 'address':
'Central st 954'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e83'), 'name': 'Chuck', 'address': 'Main
Road 989'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e84'), 'name': 'Viola', 'address':
'Sideway 1633'}
{'_id': ObjectId('6455fd2f577bc8b3ad835e85'), 'name': 'notebook', 'qty': 50,
'rating': [{'score': 8}, {'score': 9}], 'size': {'height': 11, 'width': 8.5,
'unit': 'in'}, 'status': 'A', 'tags': ['college-ruled', 'perforated']}
{'_id': ObjectId('6455fdc7577bc8b3ad835e86'), 'companyName': 'iNeuron',
'product': 'Affordable AI', 'courseOffered': 'Machine Learning with Deployment'}
{'_id': ObjectId('6455fdc7577bc8b3ad835e87'), 'companyName': 'iNeuron',
'product': 'Affordable AI', 'courseOffered': 'Deep Learning for NLP and Computer
vision'}
{'_id': ObjectId('6455fdc7577bc8b3ad835e88'), 'companyName': 'iNeuron',
'product': 'Master Program', 'courseOffered': 'Data Science Masters Program'}
{'_id': '3', 'companyName': 'iNeuron', 'Faculty': 'XYZ'}
{'_id': '4', 'companyName': 'iNeuron', 'Faculty': 'ABC'}
{'_id': '5', 'companyName': 'iNeuron', 'Faculty': 'PQR'}
```

1.7 find_one() is for finding single dictionary data

- it'll print first found data

```
[34]: collection_create.find_one() # it'll give one record, the first one it'll find␣
       ↪it'll print
```

```
[34]: {'_id': ObjectId('6455f9b0577bc8b3ad835e76'),
        'name': 'reshep',
        'class': 'data science masters',
        'timing': 'flexible'}
```

```
[38]: collection_create.find({'companyName':'iNeuron'}) # condition where company␣
       ↪name is ineuron it'll print
```

```
[38]: <pymongo.cursor.Cursor at 0x7fe08c2b65f0>
```

```
[39]: for i in collection_create.find({'companyName':'iNeuron'}):
          print(i)
```

```
{'_id': ObjectId('6455fdc7577bc8b3ad835e86'), 'companyName': 'iNeuron',
'product': 'Affordable AI', 'courseOffered': 'Machine Learning with Deployment'}
{'_id': ObjectId('6455fdc7577bc8b3ad835e87'), 'companyName': 'iNeuron',
'product': 'Affordable AI', 'courseOffered': 'Deep Learning for NLP and Computer
vision'}
{'_id': ObjectId('6455fdc7577bc8b3ad835e88'), 'companyName': 'iNeuron',
'product': 'Master Program', 'courseOffered': 'Data Science Masters Program'}
```

```
{'_id': '3', 'companyName': 'iNeuron', 'Faculty': 'XYZ'}
{'_id': '4', 'companyName': 'iNeuron', 'Faculty': 'ABC'}
{'_id': '5', 'companyName': 'iNeuron', 'Faculty': 'PQR'}
```

```
[40]: collection_create.find({"_id":{"$gte":'4'}})
```

```
[40]: <pymongo.cursor.Cursor at 0x7fe08c2a0160>
```

```
[41]: for i in collection_create.find({"_id":{"$gte":'4'}}):
          print(i)
```

```
{'_id': '4', 'companyName': 'iNeuron', 'Faculty': 'ABC'}
{'_id': '5', 'companyName': 'iNeuron', 'Faculty': 'PQR'}
```

2 update

- update_many() for all related condition will get updated
- updata_ont() is for first found, condition True data (i think)

```
[42]: collection_create.update_many({'companyName':"iNeuron"},{'$set':{"companyName":
        ↪'pwskill'}})
```

```
[42]: <pymongo.results.UpdateResult at 0x7fe08c23c430>
```

```
[43]: collection_create.find()
```

```
[43]: <pymongo.cursor.Cursor at 0x7fe08c22b700>
```

```
[44]: for i in collection_create.find():
          print(i)
```

```
{'_id': ObjectId('6455f9b0577bc8b3ad835e76'), 'name': 'reshep', 'class': 'data
science masters', 'timing': 'flexible'}
{'_id': ObjectId('6455fb36577bc8b3ad835e77'), 'mail_id': 'res@gmail.com',
'phone': 4345345}
{'_id': ObjectId('6455fc52577bc8b3ad835e78'), 'list_course': ['data science
masters ', 'web dev', 'java with DSA'], 'mentor': ['steve jobs', 'bill gates',
'albert einstein']}
{'_id': ObjectId('6455fcdc577bc8b3ad835e79'), 'name': 'Amy', 'address': 'Apple
st 652'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7a'), 'name': 'Hannah', 'address':
'Mountain 21'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7b'), 'name': 'Michael', 'address':
'Valley 345'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7c'), 'name': 'Sandy', 'address': 'Ocean
blvd 2'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7d'), 'name': 'Betty', 'address': 'Green
Grass 1'}
```

```
{'_id': ObjectId('6455fcdc577bc8b3ad835e7e'), 'name': 'Richard', 'address': 'Sky
st 331'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e7f'), 'name': 'Susan', 'address': 'One
way 98'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e80'), 'name': 'Vicky', 'address':
'Yellow Garden 2'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e81'), 'name': 'Ben', 'address': 'Park
Lane 38'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e82'), 'name': 'William', 'address':
'Central st 954'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e83'), 'name': 'Chuck', 'address': 'Main
Road 989'}
{'_id': ObjectId('6455fcdc577bc8b3ad835e84'), 'name': 'Viola', 'address':
'Sideway 1633'}
{'_id': ObjectId('6455fd2f577bc8b3ad835e85'), 'name': 'notebook', 'qty': 50,
'rating': [{'score': 8}, {'score': 9}], 'size': {'height': 11, 'width': 8.5,
'unit': 'in'}, 'status': 'A', 'tags': ['college-ruled', 'perforated']}
{'_id': ObjectId('6455fdc7577bc8b3ad835e86'), 'companyName': 'pwskill',
'product': 'Affordable AI', 'courseOffered': 'Machine Learning with Deployment'}
{'_id': ObjectId('6455fdc7577bc8b3ad835e87'), 'companyName': 'pwskill',
'product': 'Affordable AI', 'courseOffered': 'Deep Learning for NLP and Computer
vision'}
{'_id': ObjectId('6455fdc7577bc8b3ad835e88'), 'companyName': 'pwskill',
'product': 'Master Program', 'courseOffered': 'Data Science Masters Program'}
{'_id': '3', 'companyName': 'pwskill', 'Faculty': 'XYZ'}
{'_id': '4', 'companyName': 'pwskill', 'Faculty': 'ABC'}
{'_id': '5', 'companyName': 'pwskill', 'Faculty': 'PQR'}
```

3 drop()

- delete or drop complete collection

[45]:

```
collection_create.drop()
```

[46]:

```
for i in collection_create.find():
    print(i)
```

3.0.1 entire collection is dropped so find()-its unable to print it

[ ]:

```

```