

DM_W4_13032024

Phạm Ngọc Hải, Lê Thị Minh Anh, Cao Diệu Ly

March 13, 2024

1 Nội dung thực hành

1. Cài MongoDB
2. Dùng Python làm việc với MongoDB (tạo dữ liệu, truy vấn)
3. Thực hành 1 và 2 cho dự án cuối môn

2 Cài MongoDB

Dưới đây là hướng dẫn cài đặt trên ArchLinux

Yêu cầu: Cài sẵn package manager như yay, ...

```
[ ]: # !yay -S install mongodb-bin  
  
# check mongodb version  
!mongod --version
```

db version v7.0.6

Build Info: {

```
  "version": "7.0.6",  
  "gitVersion": "66cdc1f28172cb33ff68263050d73d4ade73b9a4",  
  "opensslVersion": "OpenSSL 1.1.1w 11 Sep 2023",  
  "modules": [],  
  "allocator": "tcmalloc",  
  "environment": {  
    "distmod": "ubuntu2004",  
    "distarch": "x86_64",  
    "target_arch": "x86_64"  
  }  
}
```

}

Dưới đây là tùy chọn khởi động mongod

```
[ ]: # !sudo systemctl start mongod  
# !sudo systemctl status mongod  
# !systemctl enable mongod
```

3 Dùng Python làm việc với MongoDB

3.1 Cài đặt gói pymongo - API để code Python hoạt động với MongoDB

```
[ ]: !pip install pymongo
```

```
Requirement already satisfied: pymongo in  
/home/harito/venv/py/lib/python3.11/site-packages (4.6.1)  
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in  
/home/harito/venv/py/lib/python3.11/site-packages (from pymongo) (2.4.2)
```

3.2 Chương trình Python kết nối với cơ sở dữ liệu MongoDB

```
[ ]: import pymongo  
  
client = pymongo.MongoClient("mongodb://localhost:27017/")  
  
db = client["image_db"]
```

```
[ ]: print(client.list_database_names())
```

```
['admin', 'config', 'homework_lap9', 'local', 'test']
```

Không thấy image_db vì Database chỉ được khởi tạo khi nó không rỗng

3.3 Khởi tạo collection ‘images’

```
[ ]: col = db["images"]
```

```
[ ]: print(db.list_collection_names())
```

```
[]
```

Tương tự không thấy images vì 1 collection chỉ được khởi tạo khi có content

3.4 Thêm Document vào Collection: giống việc thêm 1 Record vào 1 Table

Có thể dùng insert_one() hoặc insert_many() tùy nhu cầu

```
[ ]: import cv2  
import gridfs # use to store image  
  
data_path = "/mnt/DataK/Project/ClassifyThyroidCancer/data/B6/0.jpg"  
image1 = cv2.imread(data_path)  
image1 = cv2.cvtColor(image1, cv2.COLOR_BGR2RGB)  
  
# convert ndarray to string  
imageString = image1.tostring()  
  
fs = gridfs.GridFS(db)
```

```

# store the image
image1_ID = fs.put(imageString, encoding="utf-8")

# create our image meta data
meta = {
    "label": "B6",
    "image": {"imageID": image1_ID, "shape": image1.shape, "dtype": str(image1.
↳dtype)},
}

# insert the meta data
x = col.insert_one(meta)    # also can use col.insert_many()

```

```

/tmp/ipykernel_12419/3385559093.py:9: DeprecationWarning: tostring() is
deprecated. Use tobytes() instead.
    imageString = image1.tostring()

```

3.5 Truy vấn Document (Find Query) và hiện ra ảnh kết quả lưu trữ

Có thể dùng `find()` hoặc `find_one()` tùy nhu cầu

```

[ ]: doc = col.find({'label': 'B6'})
print(doc[0])
print(doc[0].get('image')[0].get('imageID'))

```

```

{'_id': ObjectId('65f1164576ec7822a99d1255'), 'label': 'B6', 'image':
[{'imageID': ObjectId('65f1164576ec7822a99d124a'), 'shape': [768, 1024, 3],
'dtype': 'uint8'}]}
65f1164576ec7822a99d124a

```

```

[ ]: import numpy as np

# get the image meta data
query = {'label': 'B6'}
doc = col.find(query)

# get the image from gridfs
gOut = fs.get(doc[0].get('image')[0].get('imageID'))

# convert bytes to ndarray
img = np.frombuffer(gOut.read(), dtype=np.uint8)

# reshape to match the image size
img = np.reshape(img, doc[0].get('image')[0].get('shape'))

```

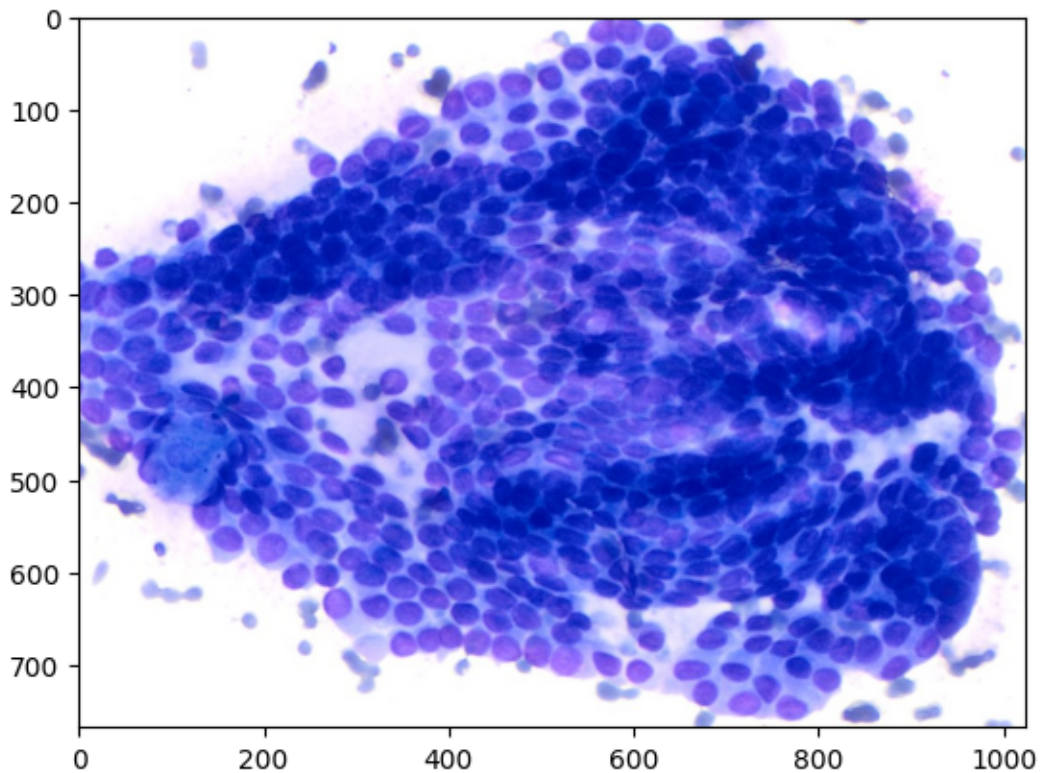
```

[ ]: import matplotlib.pyplot as plt

plt.imshow(img)

```

```
plt.show()
```



Phía trên là ví dụ truy vấn đơn giản. Ngoài ra có thể cải thiện biểu thức truy vấn query bằng:

1. Advanced Query:
E.g. to find the documents where the “address” field starts with the letter “S” or higher (alphabetically), use the greater than modifier: `query = {"$gt": "S"}`
2. Filter With Regular Expressions:
To find only the documents where the “address” field starts with the letter “S”, use the regular expression: `query = {"$regex": "^S"}`

3.6 Sắp xếp kết quả truy vấn theo thứ tự mong muốn

```
[ ]: doc = col.find().sort("_id")
# Tùy chọn:
# sort("_id", 1) #ascending
# sort("_id", -1) #descending

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

```
{'_id': ObjectId('65f1164576ec7822a99d1255'), 'label': 'B6', 'image':
```

```
[{'imageID': ObjectId('65f1164576ec7822a99d124a'), 'shape': [768, 1024, 3],
'dtype': 'uint8'}}]
{'_id': ObjectId('65f1167976ec7822a99d1261'), 'label': 'B6', 'image':
{'imageID': ObjectId('65f1167976ec7822a99d1256'), 'shape': [768, 1024, 3],
'dtype': 'uint8'}}
{'_id': ObjectId('65f116fc76ec7822a99d126d'), 'id': '0', 'label': 'B6', 'image':
{'imageID': ObjectId('65f116fc76ec7822a99d1262'), 'shape': [768, 1024, 3],
'dtype': 'uint8'}}
{'_id': ObjectId('65f1178b76ec7822a99d1279'), 'id': '0', 'label': 'B6', 'image':
{'imageID': ObjectId('65f1178b76ec7822a99d126e'), 'shape': [768, 1024, 3],
'dtype': 'uint8'}}
```

3.7 Xóa Document trong 1 Collection: Giống việc xóa 1 Record trong 1 Table

Sử dụng delete_one(query) hoặc delete_many(query) tùy theo nhu cầu

```
[ ]: query = {'label': 'B6'}

x = col.delete_many(query)

# hoặc dùng lệnh sau để xóa mọi document trong collection
# x = col.delete_many({})

print(x.deleted_count, " documents deleted.")
```

4 documents deleted.

3.8 Xóa Collection trong 1 DataBase: Giống việc xóa 1 Table trong 1 DataBase

Sử dụng db.collection.drop()

```
[ ]: # Liệt kê các collection có
print(db.list_collection_names())

['fs.chunks', 'images', 'fs.files']
```

```
[ ]: # col = db["images"]
# col.drop()
```

Sau khi chạy 2 lệnh trên thì Collection ‘images’ sẽ bị xóa

3.9 Sửa đổi Document trong 1 Collection: Giống việc thay đổi 1 Record trong 1 Table

Sử dụng phương thức update_one() hoặc update_many().

```
'''{python}
myquery = { "address": "Valley 345" } - đây là para1
newvalues = { "$set": { "address": "Canyon 123" } } - đây là para2
col.update_one(myquery, newvalues)
```

```
col.update_many(myquery, newvalues)
”
```

3.10 Xóa Document trong 1 Collection: Giống việc xóa 1 Record trong 1 Table

Sử dụng phương thức `delete_one(query)` hoặc `delete_many(query)`.

3.11 Limit Result Return: Giới hạn số lượng Document truy vấn về

Sử dụng phương thức `limit(number)`

Eg: Giới hạn lại chỉ ở 5 kết quả

```
myresult = mycol.find().limit(5)
```

3.12 Xóa 1 Database

Sử dụng `db.collection.drop()`

```
[ ]: print(client.list_database_names())
['admin', 'config', 'homework_lap9', 'image_db', 'local', 'test']

[ ]: client.drop_database('homework_lap9')

[ ]: print(client.list_database_names())
['admin', 'config', 'image_db', 'local', 'test']

[ ]: client.drop_database('image_db')
print(client.list_database_names())
['admin', 'config', 'local', 'test']
```

4 Thực hành cho dự án cuối môn

Ví dụ thực hành ở trên là phục vụ cho lưu trữ dữ liệu ảnh + nhãn phân loại của ảnh đó.

Qua đó giúp lưu trữ & truy vấn data cho dự án cuối kỳ của nhóm em: Phân loại ung thư dạ dày.