Install MongoDB

```
# MongoDB download and installation
!wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-debian71-3.0.15.tgz  # Download MongoDB fron
!tar xfv mongodb-linux-x86_64-debian71-3.0.15.tgz     # Unpack compressed file
!rm mongodb-linux-x86_64-debian71-3.0.15.tgz          # Remove downloaded file
```

```
--2021-04-20 04:32:00--  https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-debian71-3.0.15.tgz
Resolving fastdl.mongodb.org (fastdl.mongodb.org)... 65.8.27.71, 65.8.27.36, 65.8.27.40, ...
Connecting to fastdl.mongodb.org (fastdl.mongodb.org)|65.8.27.71|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 70878938 (68M) [application/x-gzip]
Saving to: ‘mongodb-linux-x86_64-debian71-3.0.15.tgz’

mongodb-linux-x86_6 100%[===================>]  67.59M  97.9MB/s    in 0.7s

2021-04-20 04:32:00 (97.9 MB/s) - ‘mongodb-linux-x86_64-debian71-3.0.15.tgz’ saved [70878938/7087

mongodb-linux-x86_64-debian71-3.0.15/README
mongodb-linux-x86_64-debian71-3.0.15/THIRD-PARTY-NOTICES
mongodb-linux-x86_64-debian71-3.0.15/GNU-AGPL-3.0
mongodb-linux-x86_64-debian71-3.0.15/bin/mongodump
mongodb-linux-x86_64-debian71-3.0.15/bin/mongorestore
mongodb-linux-x86_64-debian71-3.0.15/bin/mongoexport
mongodb-linux-x86_64-debian71-3.0.15/bin/mongoimport
mongodb-linux-x86_64-debian71-3.0.15/bin/mongostat
mongodb-linux-x86_64-debian71-3.0.15/bin/mongotop
mongodb-linux-x86_64-debian71-3.0.15/bin/bsondump
mongodb-linux-x86_64-debian71-3.0.15/bin/mongofiles
mongodb-linux-x86_64-debian71-3.0.15/bin/mongooplog
mongodb-linux-x86_64-debian71-3.0.15/bin/mongoperf
mongodb-linux-x86_64-debian71-3.0.15/bin/mongosniff
mongodb-linux-x86_64-debian71-3.0.15/bin/mongod
mongodb-linux-x86_64-debian71-3.0.15/bin/mongos
mongodb-linux-x86_64-debian71-3.0.15/bin/mongo
```

```
# Create Default location of database which is "/data/db" folder

# Default location of database is "/data/db" folder
!mkdir /data                          # data folder creation
!mkdir /data/db                        # db folder creation inside data
```

Run MongoDB Server in Background

```
#Run mongoDB server On Colab
! nohup mongodb-linux-x86_64-debian71-3.0.15/bin/mongod --nojournal --dbpath /data/db &
```

```
nohup: appending output to 'nohup.out'
```

Install pymongo

```
#Install PyMongo
! python -m pip install pymongo==3.7.2
```

```
Collecting pymongo==3.7.2
  Downloading https://files.pythonhosted.org/packages/99/18/b50834fbfd557eaf07985c2a657b03efc691
    |████████████████████████████████| 409kB 5.6MB/s
Installing collected packages: pymongo
  Found existing installation: pymongo 3.11.3
    Uninstalling pymongo-3.11.3:
      Successfully uninstalled pymongo-3.11.3
Successfully installed pymongo-3.7.2
```

Connect Client to Server

```
#Connect Client to server
import pymongo
from pymongo import MongoClient
uri = 'localhost:27017'
client = MongoClient( uri )
```

Homework

3. Type “**show dbs**”. What is the command used for? Show your capture.

```
print (client.list_database_names())
```

```
['local']
```

เป็นคำสั่งที่ใช้ในการเรียกดูลิสต์ database

4. Type “**use labdb**”. What is the command used for? Show your capture

```
mydb = client["labdb"]
```

เป็นคำสั่งที่ใช้ในการเซ็ตให้ mydb เป็นการเรียกใช้ database 'labdb'

5. Type “**db.createCollection("labcollection")**”. Show your capture. What is the command used for?

```
mydb.create_collection("labcollection")
```

```
Collection(Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, conne
```

เป็นคำสั่งที่ใช้ในการสร้าง collection

6. Then, type “**show dbs**” again. What do you see? Show your capture. What will happen after we type the “use DB_NAME” (“use labdb”) command to select a database with no existence of that database?

```
print (client.list_database_names())
```

```
['labdb', 'local']
```

จะเห็นได้ว่ามี database 'labdb' ทั้งที่ยังไม่ได้สร้าง และถ้าเราใช้ use labdb โปรแกรมจะสร้าง database ที่ยังไม่มีให้อัตโนมัติ

7. Type “**show collections**”. What do you see? Show your capture

```
mydb.list_collection_names()
```

```
['labcollection', 'system.indexes']
```

เห็นลิสต์ collection ใน database

8. Type “**db.labcollection.insert({name:’testCreateCollection’})**” to insert data {name:’testCreateCollection’} to the “labcollection”. Show your capture.

```
mydb.labcollection.insert_one({'name': 'testCreateCollection'})
```

```
<pymongo.results.InsertOneResult at 0x7f561711a690>
```

9. Type **“db.labcollection2.insert({name : ‘testCreateCollection2’})”** .

```
mydb.labcollection2.insert_one({'name': 'testCreateCollection2'})
```

```
<pymongo.results.InsertOneResult at 0x7f561706d320>
```

9.1 Then capture your screen and type “**show colllections**”. Do you see a new collection “labcollections2” just created?

```
mydb.list_collection_names()
```

```
['labcollection', 'labcollection2', 'system.indexes']
```

10. What will happen if you use the insert command to add data to a collection which has not been created before inserting the data? Briefly explain.

ANS: จะสร้าง collection ที่ยังไม่มีอัตโนมัติ และเพิ่มข้อมูลเข้าไปใน collection ที่ถูกสร้างขึ้นใหม่

11. Type drop both the collections (labcollection and labcollection2) by using **db.COLLECTION_NAME.drop()** . Show your capture.

```
mydb.labcollection.drop()
mydb.labcollection2.drop()
```

```
mydb.list_collection_names()
```

```
['system.indexes']
```

12. Create a collection name “stock”. Show your command.

```
mydb.create_collection("stock")
```

```
Collection(Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, conne
```

13. Insert a document into the “stock” collection by using the following

```
result = mydb.stock.insert_one({\
                    'item': "tshirt",\
                    'details': {\
                          'model': "S321",\
                          'manufacturer': "KMUTT Company"
                          },\
                    'stock': [ { 'size': "S", 'qty': 20 },\
                          { 'size': "M", 'qty': 15 } ],\
                    'category': "clothing",\
                    'price':100\
                    })
```

13.1 Show your capture. What is WriteResult?

PS. WriteResult in pymongo is the object which is the return value when we call any funciton (i.e. drop, insert, update) https://api.mongodb.com/python/current/api/pymongo/results.html

```
print(result)

    <pymongo.results.InsertOneResult object at 0x7f5617075b90>
```

14. Then insert the following items into the "stock" collection by using one insert command or multiple insert commands. Use the previous document as example

Item	Details		Stock			Category	Price/unit
	Model	Manufacturer					
Jeans	K009	Levis	S-30	M-45	L-30	clothing	230
Bowl	B011	Superware	S-10	M-23	L-10	houseware	90
DVD	C122	LG	S-12	M-5	-	electronics	10
Egg	G999	CPF	S-20	-	-	food	45

```
mydb.stock.insert_many([{ 'item': "Jeans", 'details': {'model': "K009", 'manufacturer': "Levis"},\
                'stock': [ { 'size': "S", 'qty': 30 }, { 'size': "M", 'qty': 45 }, { 'size': "L", 'qty': 30}],\
                'category': "clothing", 'price':230 },\
                    {'item': "Bowl", 'details': {'model': "B011",'manufacturer': "Superware" },\
                'stock': [ { 'size': "S", 'qty': 10 },{ 'size': "M", 'qty': 23 },{ 'size': "L", 'qty': 10}],\
                'category': "houseware", 'price':90 },\
                   { 'item': "DVD", 'details': { 'model': "C122", 'manufacturer': "LG"  },\
                'stock': [ { 'size': "S", 'qty': 12 },  { 'size': "M", 'qty': 5 }],\
                'category': "electronics", 'price':10  },\
                   { 'item': "Egg", 'details': { 'model': "G999", 'manufacturer': "CPF" },\
                'stock': { 'size': "S", 'qty': 20 }, 'category': "food", 'price':45 }])

    <pymongo.results.InsertManyResult at 0x7f561707a280>
```

15. Type "**db.stock.find({})**". Show your capture. What is find() used for?

ใช้ในการ select ทั้งหมดใน collection 'stock'

```
list(mydb.stock.find({}))

    [{'_id': ObjectId('607e59987dc7b7003d707b29'),
      'category': 'clothing',
      'details': {'manufacturer': 'KMUTT Company', 'model': 'S321'},
      'item': 'tshirt',
      'price': 100,
      'stock': [{'qty': 20, 'size': 'S'}, {'qty': 15, 'size': 'M'}]},
     {'_id': ObjectId('607e59a77dc7b7003d707b2a'),
      'category': 'clothing',
      'details': {'manufacturer': 'Levis', 'model': 'K009'},
      'item': 'Jeans',
      'price': 230,
      'stock': [{'qty': 30, 'size': 'S'},
       {'qty': 45, 'size': 'M'},
       {'qty': 30, 'size': 'L'}]},
     {'_id': ObjectId('607e59a77dc7b7003d707b2b'),
      'category': 'houseware',
      'details': {'manufacturer': 'Superware', 'model': 'B011'},
      'item': 'Bowl',
      'price': 90,
      'stock': [{'qty': 10, 'size': 'S'},
       {'qty': 23, 'size': 'M'},
       {'qty': 10, 'size': 'L'}]},
     {'_id': ObjectId('607e59a77dc7b7003d707b2c'),
      'category': 'electronics',
      'details': {'manufacturer': 'LG', 'model': 'C122'},
      'item': 'DVD',
      'price': 10,
      'stock': [{'qty': 12, 'size': 'S'}, {'qty': 5, 'size': 'M'}]},
     {'_id': ObjectId('607e59a77dc7b7003d707b2d'),
      'category': 'food',
      'details': {'manufacturer': 'CPF', 'model': 'G999'},
      'item': 'Egg',
      'price': 45,
      'stock': {'qty': 20, 'size': 'S'}}]
```

16. Type “**db.stock.find({ category: ‘clothing’ })**”. Show your capture. What items do you get from executing the command?

จะได้ข้อมูลที่มี category = clothing ใน collection 'stock'

```
list(mydb.stock.find({'category':'clothing'}))
```

```
[{'_id': ObjectId('607e59987dc7b7003d707b29'),
  'category': 'clothing',
  'details': {'manufacturer': 'KMUTT Company', 'model': 'S321'},
  'item': 'tshirt',
  'price': 100,
  'stock': [{'qty': 20, 'size': 'S'}, {'qty': 15, 'size': 'M'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2a'),
  'category': 'clothing',
  'details': {'manufacturer': 'Levis', 'model': 'K009'},
  'item': 'Jeans',
  'price': 230,
  'stock': [{'qty': 30, 'size': 'S'},
   {'qty': 45, 'size': 'M'},
   {'qty': 30, 'size': 'L'}]}]
```

17. Type “**db.stock.find({ category: ‘clothing’, price: { $lt: 100 } })**”. What do you get?

```
list(mydb.stock.find({'category':"clothing", 'price': {'$lt': 100}}))
```

```
[]
```

18. Show your command if you would like to search for any item which its price is greater than 50.

```
list(mydb.stock.find({'price':{'$gt': 50}}))
```

```
[{'_id': ObjectId('607e59987dc7b7003d707b29'),
  'category': 'clothing',
  'details': {'manufacturer': 'KMUTT Company', 'model': 'S321'},
  'item': 'tshirt',
  'price': 100,
  'stock': [{'qty': 20, 'size': 'S'}, {'qty': 15, 'size': 'M'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2a'),
  'category': 'clothing',
  'details': {'manufacturer': 'Levis', 'model': 'K009'},
  'item': 'Jeans',
  'price': 230,
  'stock': [{'qty': 30, 'size': 'S'},
   {'qty': 45, 'size': 'M'},
   {'qty': 30, 'size': 'L'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2b'),
  'category': 'houseware',
  'details': {'manufacturer': 'Superware', 'model': 'B011'},
  'item': 'Bowl',
  'price': 90,
  'stock': [{'qty': 10, 'size': 'S'},
   {'qty': 23, 'size': 'M'},
   {'qty': 10, 'size': 'L'}]}]
```

19. Find all items that have size “S” by using the following command **db.stock.find({ ‘stock.size’: ‘S’ });**

```
list(mydb.stock.find({'stock.size': 'S'}))
```

```
[{'_id': ObjectId('607e59987dc7b7003d707b29'),
  'category': 'clothing',
  'details': {'manufacturer': 'KMUTT Company', 'model': 'S321'},
  'item': 'tshirt',
  'price': 100,
  'stock': [{'qty': 20, 'size': 'S'}, {'qty': 15, 'size': 'M'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2a'),
  'category': 'clothing',
  'details': {'manufacturer': 'Levis', 'model': 'K009'},
  'item': 'Jeans',
  'price': 230,
  'stock': [{'qty': 30, 'size': 'S'},
   {'qty': 45, 'size': 'M'},
   {'qty': 30, 'size': 'L'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2b'),
  'category': 'houseware',
  'details': {'manufacturer': 'Superware', 'model': 'B011'},
  'item': 'Bowl',
  'price': 90,
  'stock': [{'qty': 10, 'size': 'S'},
   {'qty': 23, 'size': 'M'},
   {'qty': 10, 'size': 'L'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2c'),
  'category': 'electronics',
  'details': {'manufacturer': 'LG', 'model': 'C122'},
  'item': 'DVD',
  'price': 10,
  'stock': [{'qty': 12, 'size': 'S'}, {'qty': 5, 'size': 'M'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2d'),
  'category': 'food',
  'details': {'manufacturer': 'CPF', 'model': 'G999'},
  'item': 'Egg',
  'price': 45,
  'stock': {'qty': 20, 'size': 'S'}}]
```

20. Show your command and result if you would like to find items which have size M and their quantity is greater than 10. Hint: Using $elemMatch for matching documents that contain an array field.

```
list(mydb.stock.find({'stock': {'$elemMatch':{'size':'M','qty':{'$gt':10}}}}))
```

```
[{'_id': ObjectId('607e59987dc7b7003d707b29'),
  'category': 'clothing',
  'details': {'manufacturer': 'KMUTT Company', 'model': 'S321'},
  'item': 'tshirt',
  'price': 100,
  'stock': [{'qty': 20, 'size': 'S'}, {'qty': 15, 'size': 'M'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2a'),
  'category': 'clothing',
  'details': {'manufacturer': 'Levis', 'model': 'K009'},
  'item': 'Jeans',
  'price': 230,
  'stock': [{'qty': 30, 'size': 'S'},
   {'qty': 45, 'size': 'M'},
   {'qty': 30, 'size': 'L'}]},
 {'_id': ObjectId('607e59a77dc7b7003d707b2b'),
  'category': 'houseware',
  'details': {'manufacturer': 'Superware', 'model': 'B011'},
  'item': 'Bowl',
  'price': 90,
  'stock': [{'qty': 10, 'size': 'S'},
   {'qty': 23, 'size': 'M'},
   {'qty': 10, 'size': 'L'}]}]
```

21. Show your command if you would like to find the same items in Step 20 but only fields “item”, “category”, and “price” sorted by “price” in ascending order are shown in the result set. Note that by default, the _id field is included in the results. To suppress the _id field from the result set, specify _id: 0 in the projection document.

```
list(mydb.stock.find({'stock': {'$elemMatch':{'size':'M','qty':{'$gt':10}}}},{'_id':0,'item':1,'category':1,\
                                                                    'price':1}).sort('price',1))
```

```
[{'category': 'houseware', 'item': 'Bowl', 'price': 90},
 {'category': 'clothing', 'item': 'tshirt', 'price': 100},
 {'category': 'clothing', 'item': 'Jeans', 'price': 230}]
```

22. For the document with item equal to "DVD", use the $set operator to update the category field and the details field to the specified values shown below. Show your capture and all the documents.

```
#{
#  'category': "computer",
#  'details': { 'model': "D222", 'manufacturer': "Samsung" }
#}
list(mydb.stock.update({'item':'DVD'},\
                {'$set':{'category': "computer",\
                        'details': { 'model': "D222", 'manufacturer': "Samsung" }}}))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: DeprecationWarning: update is deprecate
  """
['ok', 'nModified', 'n', 'updatedExisting']
```

```
list(mydb.stock.find({'item':'DVD'}))
```

```
[{'_id': ObjectId('607e59a77dc7b7003d707b2c'),
  'category': 'computer',
  'details': {'manufacturer': 'Samsung', 'model': 'D222'},
  'item': 'DVD',
  'price': 10,
  'stock': [{'qty': 12, 'size': 'S'}, {'qty': 5, 'size': 'M'}]}]
```

23. Use the following command to change the manufacture of the Egg item. Show your capture and all the documents.

```
#{ '$set': { "details.manufacturer": "Betagro" } }
list(mydb.stock.update({'item':'Egg'},{'$set': { "details.manufacturer": "Betagro" }}))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: DeprecationWarning: update is deprecate

['ok', 'nModified', 'n', 'updatedExisting']
```

```
list(mydb.stock.find({'item':'Egg'}))
```

```
[{'_id': ObjectId('607e59a77dc7b7003d707b2d'),
  'category': 'food',
  'details': {'manufacturer': 'Betagro', 'model': 'G999'},
  'item': 'Egg',
  'price': 45,
  'stock': {'qty': 20, 'size': 'S'}}]
```

24. Remove items with category “clothing” by using db.stock.remove({ category : "clothing"}). Show your result and all the documents left.

```
mydb.stock.remove({'category':'clothing'})
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: DeprecationWarning: remove is deprecat
  """Entry point for launching an IPython kernel.
{'n': 2, 'ok': 1}
```

```
list(mydb.stock.find({'category':'clothing'}))
```

```
[]
```

25. Remove all from the collection. Show the command.

```
mydb.stock.remove({})
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: DeprecationWarning: remove is deprecat
  """Entry point for launching an IPython kernel.
{'n': 3, 'ok': 1}
```

```
list(mydb.stock.find({}))
```

```
[]
```

26. Remove the collection from the database. Show the command.

```
mydb.stock.drop()
```

27. Remove the database. Show the command.

```
client.drop_database('mydb')
```

✓ 0 วินาที เสร็จสมบูรณ์เมื่อ 11:34 ● ✕