

DBMS LAB 12

Name: Sachin Hansda

Roll No.: 111601019

Lab Report

Part II

6 Connect to database inventory, and collection store using PyMongo

```
> use inventory;
switched to db inventory
> db.dropDatabase()
{ "dropped" : "inventory", "ok" : 1 }
> use inventory;
switched to db inventory
> db.store.insertMany([
... { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
... { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
... { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
... { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
... { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
... ]);
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5ca3b33dd41f0b8ec0253ccb"),
    ObjectId("5ca3b33dd41f0b8ec0253ccc"),
    ObjectId("5ca3b33dd41f0b8ec0253ccd"),
    ObjectId("5ca3b33dd41f0b8ec0253cce"),
    ObjectId("5ca3b33dd41f0b8ec0253ccf")
  ]
}
```

7. Show the records using PyMongo

```
dbms@dbms-VirtualBox:~$ python
Python 2.7.15rc1 (default, Nov 12 2018, 14:31:15)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pymongo
>>> myclient = pymongo.MongoClient("mongodb://localhost:27017/")
>>> mydb = myclient["inventory"]
>>> mycol = mydb["store"]
>>> for x in mycol.find():
...     print(x)
...
{'size': {'h': 14.0, 'uom': 'cm', 'w': 21.0}, 'item': 'journal', 'id': ObjectId('5ca3b33dd41f0b8ec0253ccb'), 'status': 'A', 'qty': 25.0}
{'size': {'h': 8.5, 'uom': 'in', 'w': 11.0}, 'item': 'notebook', 'id': ObjectId('5ca3b33dd41f0b8ec0253ccc'), 'status': 'A', 'qty': 50.0}
{'size': {'h': 8.5, 'uom': 'in', 'w': 11.0}, 'item': 'paper', 'id': ObjectId('5ca3b33dd41f0b8ec0253ccd'), 'status': 'D', 'qty': 100.0}
{'size': {'h': 22.85, 'uom': 'cm', 'w': 30.0}, 'item': 'planner', 'id': ObjectId('5ca3b33dd41f0b8ec0253cce'), 'status': 'D', 'qty': 75.0}
{'size': {'h': 10.0, 'uom': 'cm', 'w': 15.25}, 'item': 'postcard', 'id': ObjectId('5ca3b33dd41f0b8ec0253ccf'), 'status': 'A', 'qty': 45.0}
>>>
```

Part III

8. Import data into test database corresponding to a `restaurants` collection using the `primer-dataset.json` file

```
> use test;
switched to db test
> db.createCollection("restaurant");
{ "ok" : 1 }
```

```
dbms@dbms-VirtualBox:~$ mongoimport --db test --collection restaurant --file primer-dataset.json;
2019-04-03T00:55:26.364+0530    connected to: localhost
2019-04-03T00:55:27.420+0530    imported 25359 documents
```

9. List all the attributes in `restaurants` collection

Python Code:

```
import pymongo
```

```
from pprint import pprint
```

```
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
```

```
mydb = myclient["test"]
```

```
cursor=mydb.restaurant.find({})
```

```
for item in cursor:
```

```
    for key in item:
```

```
        pprint(key)
```

```
    break
```

```
'_id'
'address'
'borough'
'cuisine'
'grades'
'name'
'restaurant_id'
```

10. Count total number of restaurants

Python Code:

```
import pymongo

from pprint import pprint

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["test"]

cnt = mydb.restaurant.count()

print (cnt)
```

```
50718
```

11. Count number of restaurants with American cuisine

Python Code:

```
import pymongo

from pprint import pprint

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["test"]
```

```
cnt = mydb.restaurant.find({'cuisine':'American'}).count()

print (cnt)
```

A digital display showing the number 12366 in green on a black background.

12. Show the number of restaurants corresponding to each cuisine

Python Code:

```
import pymongo

from pprint import pprint

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["test"]

data = mydb.restaurant.distinct('cuisine')

for item in data:

    cnt = mydb.restaurant.find({'cuisine':item}).count()

    print (item,cnt)
```

```
Bakery 1382
Hamburgers 866
Irish 380
American 12366
Jewish/Kosher 632
Delicatessen 642
Ice Cream, Gelato, Yogurt, Ices 696
Chinese 4836
Chicken 820
Turkish 140
Caribbean 1314
Donuts 958
Sandwiches/Salads/Mixed Buffet 510
Bagels/Pretzels 336
Continental 116
Pizza 2326
Italian 2138
Steak 172
Polish 50
Latin (Cuban, Dominican, Puerto Rican, South & Central American) 1700
German 62
French 688
Pizza/Italian 936
Mexican 1508
Spanish 1274
Café/Coffee/Tea 2428
Tex-Mex 286
Pancakes/Waffles 32
Soul Food 88
Seafood 294
```

\

13. Count number of Hawaiian cuisine restaurants with grade=A

Python Code:

```
import pymongo

from pprint import pprint

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["test"]

cnt = mydb.restaurant.find({'cuisine':'Hawaiian';'grades.grade':'A'}).count()
```

```
print (cnt)
```



14. Show the number of restaurants corresponding to each cuisine using ‘cursor’ mode (you can view it in parts and show more like setting) in sorted order so that highest number cuisines appear first.

Python Code:

```
import pymongo

from pprint import pprint

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["test"]

cursor=mydb.restaurant.aggregate(

    [

        {'$group':{'_id':'$cuisine','count':{'$sum':1}}},

        {'$sort':{'count':-1}}

    ]

)

for item in cursor:

    pprint(item)
```

```

'_id': 'American', 'count': 12366}
'_id': 'Chinese', 'count': 4836}
'_id': 'Café/Coffee/Tea', 'count': 2428}
'_id': 'Pizza', 'count': 2326}
'_id': 'Italian', 'count': 2138}
'_id': 'Other', 'count': 2022}
'_id': 'Latin (Cuban, Dominican, Puerto Rican, South & Central American)',
'count': 1700}
'_id': 'Japanese', 'count': 1520}
'_id': 'Mexican', 'count': 1508}
'_id': 'Bakery', 'count': 1382}
'_id': 'Caribbean', 'count': 1314}
'_id': 'Spanish', 'count': 1274}
'_id': 'Donuts', 'count': 958}
'_id': 'Pizza/Italian', 'count': 936}
'_id': 'Sandwiches', 'count': 918}
'_id': 'Hamburgers', 'count': 866}
'_id': 'Chicken', 'count': 820}
'_id': 'Ice Cream, Gelato, Yogurt, Ices', 'count': 696}
'_id': 'French', 'count': 688}
'_id': 'Delicatessen', 'count': 642}
'_id': 'Indian', 'count': 632}
'_id': 'Jewish/Kosher', 'count': 632}
'_id': 'Asian', 'count': 618}
'_id': 'Thai', 'count': 570}
'_id': 'Juice, Smoothies, Fruit Salads', 'count': 546}
'_id': 'Korean', 'count': 524}
'_id': 'Sandwiches/Salads/Mixed Buffet', 'count': 510}
'_id': 'Mediterranean', 'count': 438}
'_id': 'Irish', 'count': 380}
'_id': 'Bagels/Pretzels', 'count': 336}

```

15. Modify 8 such that it returns cuisines with restaurants more than 100

Python Code:

```
import pymongo
```

```
from pprint import pprint
```

```
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
```

```
mydb = myclient["test"]
```

```
cursor=mydb.restaurant.aggregate(
```



```
[
    { '$group': { '_id': '$cuisine', 'count': { '$sum': 1 } } },
    { '$match': { 'count': { '$gt': 100 } } }
]
)
```

for item in cursor:

```
    pprint(item)
```

```
{ '_id': 'Peruvian', 'count': 136 }
{ '_id': 'Vietnamese/Cambodian/Malaysia', 'count': 132 }
{ '_id': 'Juice, Smoothies, Fruit Salads', 'count': 546 }
{ '_id': 'Sandwiches', 'count': 918 }
{ '_id': 'Barbecue', 'count': 104 }
{ '_id': 'Asian', 'count': 618 }
{ '_id': 'Eastern European', 'count': 130 }
{ '_id': 'Bottled beverages, including water, sodas, juices, etc.', 'count': 144 }
{ '_id': 'Thai', 'count': 570 }
{ '_id': 'Other', 'count': 2022 }
{ '_id': 'Mediterranean', 'count': 438 }
{ '_id': 'Middle Eastern', 'count': 336 }
{ '_id': 'African', 'count': 136 }
{ '_id': 'Seafood', 'count': 294 }
{ '_id': 'Soups & Sandwiches', 'count': 102 }
{ '_id': 'Spanish', 'count': 1274 }
{ '_id': 'Mexican', 'count': 1508 }
{ '_id': 'Pizza/Italian', 'count': 936 }
{ '_id': 'Japanese', 'count': 1520 }
{ '_id': 'French', 'count': 688 }
{ '_id': 'Italian', 'count': 2138 }
{ '_id': 'Korean', 'count': 524 }
{ '_id': 'Russian', 'count': 176 }
{ '_id': 'Steak', 'count': 172 }
{ '_id': 'Indian', 'count': 632 }
{ '_id': 'Pizza', 'count': 2326 }
{ '_id': 'Donuts', 'count': 958 }
{ '_id': 'Caribbean', 'count': 1314 }
{ '_id': 'Turkish', 'count': 140 }
{ '_id': 'Chinese', 'count': 4836 }
{ '_id': 'Ice Cream, Gelato, Yogurt, Ices', 'count': 696 }
{ '_id': 'Delicatessen', 'count': 642 }
{ '_id': 'Bagels/Pretzels', 'count': 336 }
```


