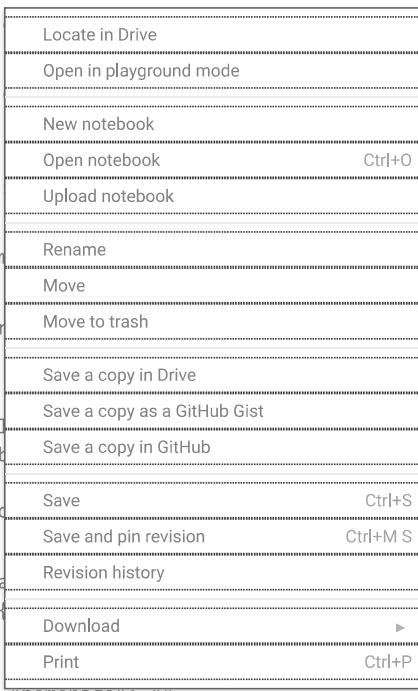


1 pip install pymongo



```
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (677 kB)
  └── 677.1/677.1 kB 4.6 MB/s eta 0:00:00
    ├── pymongo
    ├── y.whl (300 kB)
    └── 300.4/300.4 kB 9.2 MB/s eta 0:00:00
      ├── pymongo
      └── mongo-4.6.1
```

```
1 in Rename
2 Move
3 Move to trash
4 Save a copy in Drive
5 Save a copy as a GitHub Gist
6 Save a copy in GitHub
7 Save
8 Save and pin revision
9 Revision history
10 Download
11 Print
12 Ctrl+S
13 Ctrl+M S
14
15 fiq19hk:H0134440h.@cluster0.zga9xvn.mongodb.net/?retryWrites=true&w=majority")
16
17
18 homepage : "",
19 "_event_time": "2021-04-01T18:34:08.119000Z",
20 "overview": "In this fast-paced, noirish road movie, a computer expert embezzles half a million dollars and races off
21 "poster_path": "/g2o1J0ulttuwovqlc0ho910MR95.jpg",
22 "status": "Released",
23 "vote_average": 4.8,
24 "_id": "5fd124fd5b85548af6a3938a",
25 "budget": 1000000,
26 "video": "False",
27 "imdb_id": "",
28 "genres": "[{"id": 80, "name": "Crime"}]",
29 "tagline": "She only wanted love. But money's better than nothing.",
30 "runtime": 100,
31 "original_language": "en",
32 "_meta": { "mongodb": { "database": "netflix", "collection": "movies" } },
33 "production_countries": "[{"iso_3166_1": "US", "name": "United States of America"}]",
34 "adult": "False",
35 "spoken_languages": "[{"iso_639_1": "en", "name": "English"}]",
36 "vote_count": 3,
37 "production_companies": "[{"name": "Cineville", "id": 2832}]",
38 "revenue": 0,
39 "popularity": 0.156722,
40 "release_date": "1991-06-07",
41 "id": 36337,
42 "title": "Delusion"
43 }
44 ]
```

```
1 # insert many documents in collection  
2 collection.insert_many(data)
```

```
InsertManyResult(['5fd124fd5b85548af6a3938a'], acknowledged=True)
```

```
1 # {
2 #     status: 'A',
3 #     $or: [
4 #         { qty: { $lt: 30 } }, { item: { $regex: '^p' } }
5 #     ]
6 # }
```

```
1 raw=collection.find({})
2 for data in raw:
3     print(data)

1 data=collection.find({}, {"database": "netflix"})
2 for mydata in data:
3     print(mydata)

1
```

✓ Mongodb Sample Data

```

1 collection.insert_many(
2   [
3     {
4       "item": "canvas",
5       "qty": 100,
6       "size": {"h": 28, "w": 35.5, "uom": "cm"},
7       "status": "A",
8     },
9     {
10       "item": "journal",
11       "qty": 25,
12       "size": {"h": 14, "w": 21, "uom": "cm"},
13       "status": "A",
14     },
15     {
16       "item": "mat",
17       "qty": 85,
18       "size": {"h": 27.9, "w": 35.5, "uom": "cm"},
19       "status": "A",
20     },
21     {
22       "item": "mousepad",
23       "qty": 25,
24       "size": {"h": 19, "w": 22.85, "uom": "cm"},
25       "status": "P",
26     },
27     {
28       "item": "notebook",
29       "qty": 50,
30       "size": {"h": 8.5, "w": 11, "uom": "in"},
31       "status": "P",
32     },
33     {
34       "item": "paper",
35       "qty": 100,
36       "size": {"h": 8.5, "w": 11, "uom": "in"},
37       "status": "D",
38     },
39     {
40       "item": "planner",
41       "qty": 75,
42       "size": {"h": 22.85, "w": 30, "uom": "cm"},
43       "status": "D",
44     },
45     {
46       "item": "postcard",
47       "qty": 45,
48       "size": {"h": 10, "w": 15.25, "uom": "cm"},
49       "status": "A",
50     },
51     {
52       "item": "sketchbook",
53       "qty": 80,
54       "size": {"h": 14, "w": 21, "uom": "cm"},
55       "status": "A",
56     },
57     {
58       "item": "sketch pad",
59       "qty": 95,
60       "size": {"h": 22.85, "w": 30.5, "uom": "cm"},
61       "status": "A",
62     },
63   ],
64 )

```

```

InsertManyResult([ObjectId('658f2bbacad3daf94f42a821'), ObjectId('658f2bbacad3daf94f42a822'), ObjectId('658f2bbacad3daf94f42a823'),
ObjectId('658f2bbacad3daf94f42a824'), ObjectId('658f2bbacad3daf94f42a825'), ObjectId('658f2bbacad3daf94f42a826'),
ObjectId('658f2bbacad3daf94f42a827'), ObjectId('658f2bbacad3daf94f42a828'), ObjectId('658f2bbacad3daf94f42a829'),
ObjectId('658f2bbacad3daf94f42a82a')], acknowledged=True)

```

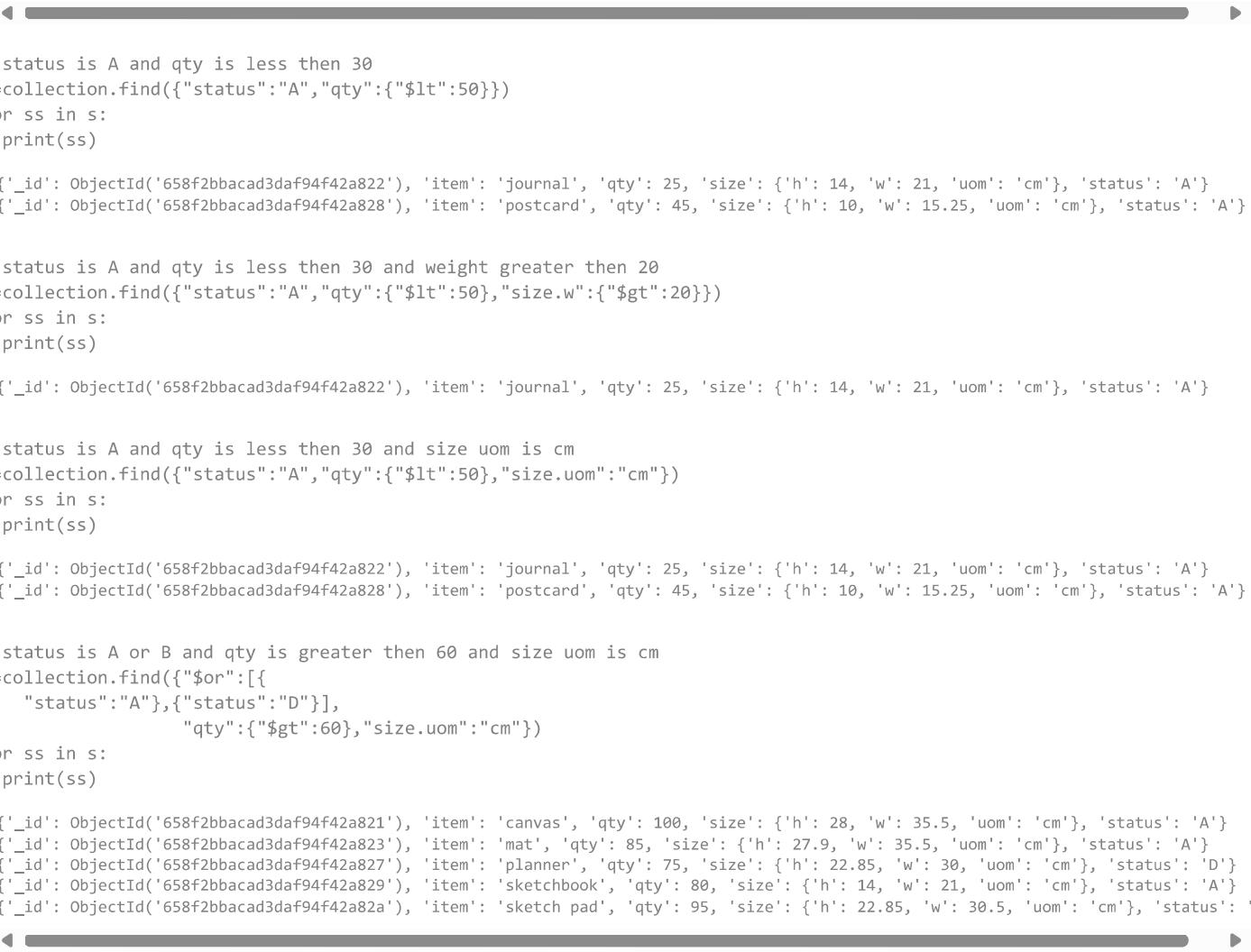
```

1 s=collection.find({"item":"sketch pad"})

1 # print record where size.w is 11
2

1 for ss in s:
2   print(ss)

{'_id': ObjectId('658f2bbacad3daf94f42a82a'), 'item': 'sketch pad', 'qty': 95, 'size': {'h': 22.85, 'w': 30.5, 'uom': 'cm'}, 'status': 'A'}
```



```

1 # status is A and qty is less then 30
2 s=collection.find({"status":"A","qty":{$lt:50}})
3 for ss in s:
4   print(ss)

{'_id': ObjectId('658f2bbacad3daf94f42a822'), 'item': 'journal', 'qty': 25, 'size': {'h': 14, 'w': 21, 'uom': 'cm'}, 'status': 'A'}
{'_id': ObjectId('658f2bbacad3daf94f42a828'), 'item': 'postcard', 'qty': 45, 'size': {'h': 10, 'w': 15.25, 'uom': 'cm'}, 'status': 'A'}
```

```

1 # status is A and qty is less then 30 and weight greater then 20
2 s=collection.find({"status":"A","qty":{$lt:50}, "size.w":{$gt:20}})
3 for ss in s:
4   print(ss)

{'_id': ObjectId('658f2bbacad3daf94f42a822'), 'item': 'journal', 'qty': 25, 'size': {'h': 14, 'w': 21, 'uom': 'cm'}, 'status': 'A'}
```

```

1 # status is A and qty is less then 30 and size uom is cm
2 s=collection.find({"status":"A","qty":{$lt:50}, "size.uom":"cm"})
3 for ss in s:
4   print(ss)

{'_id': ObjectId('658f2bbacad3daf94f42a822'), 'item': 'journal', 'qty': 25, 'size': {'h': 14, 'w': 21, 'uom': 'cm'}, 'status': 'A'}
{'_id': ObjectId('658f2bbacad3daf94f42a828'), 'item': 'postcard', 'qty': 45, 'size': {'h': 10, 'w': 15.25, 'uom': 'cm'}, 'status': 'A'}
```

```

1 # status is A or B and qty is greater then 60 and size uom is cm
2 s=collection.find({"$or": [
3   {"status":"A"}, {"status":"D"}],
4   {"qty":{$gt:60}, "size.uom":"cm"})
5 for ss in s:
6   print(ss)

{'_id': ObjectId('658f2bbacad3daf94f42a821'), 'item': 'canvas', 'qty': 100, 'size': {'h': 28, 'w': 35.5, 'uom': 'cm'}, 'status': 'A'}
{'_id': ObjectId('658f2bbacad3daf94f42a823'), 'item': 'mat', 'qty': 85, 'size': {'h': 27.9, 'w': 35.5, 'uom': 'cm'}, 'status': 'A'}
{'_id': ObjectId('658f2bbacad3daf94f42a827'), 'item': 'planner', 'qty': 75, 'size': {'h': 22.85, 'w': 30, 'uom': 'cm'}, 'status': 'D'}
{'_id': ObjectId('658f2bbacad3daf94f42a829'), 'item': 'sketchbook', 'qty': 80, 'size': {'h': 14, 'w': 21, 'uom': 'cm'}, 'status': 'A'}
{'_id': ObjectId('658f2bbacad3daf94f42a82a'), 'item': 'sketch pad', 'qty': 95, 'size': {'h': 22.85, 'w': 30.5, 'uom': 'cm'}, 'status': 'A'}
```

```

1 # change the status D to A
2 collection.update_one({"item":"paper"},
3   {"$set":{"status":"A"})

UpdateResult({'n': 1, 'electionId': ObjectId('7ffffffff000000000000c6'), 'opTime': {'ts': Timestamp(1703883186, 28), 't': 198},
  'nModified': 1, 'ok': 1.0, '$clusterTime': {'clusterTime': Timestamp(1703883186, 28), 'signature': {'hash': 'b``\xa5>\xab\xb0\xfd\x11\x87\x13\xb3\x18\xd2\x1d\xcc\xd3\x07V\xc2\xf5\x1b', 'keyId': 720441575035895810}}, 'operationTime':
  Timestamp(1703883186, 28), 'updatedExisting': True}, acknowledged=True)

1 collection.find_one({"item":"paper"})

1 # northwind data

1 import pprint
2 import json
3 from bson.json_util import dumps
4 from bson import json_util
5 # import
6 data=collection.find({"employees.country":"USA"},{'_id':0,"employees.first_name":1,"employees.last_name":1,"employees.c
7   for data in data:
```

```

7   for datas in data:
8     # print(dict(datas))
9     # print(json.dumps(datas, indent=4, default=json_util.default)))
10    pprint.pprint(datas)
11    # json.dumps(datas,default=json_util.default)
12

13  {'employees': [{'city': 'Seattle',
14    'country': 'USA',
15    'first_name': 'Nancy',
16    'last_name': 'Davolio'},
17   {'city': 'Tacoma',
18    'country': 'USA',
19    'first_name': 'Andrew',
20    'last_name': 'Fuller'},
21   {'city': 'Kirkland',
22    'country': 'USA',
23    'first_name': 'Janet',
24    'last_name': 'Leverling'},
25   {'city': 'Redmond',
26    'country': 'USA',
27    'first_name': 'Margaret',
28    'last_name': 'Peacock'},
29   {'city': 'London',
30    'country': 'UK',
31    'first_name': 'Steven',
32    'last_name': 'Buchanan'},
33   {'city': 'London',
34    'country': 'UK',
35    'first_name': 'Michael',
36    'last_name': 'Suyama'},
37   {'city': 'London',
38    'country': 'UK',
39    'first_name': 'Robert',
40    'last_name': 'King'},
41   {'city': 'Seattle',
42    'country': 'USA',
43    'first_name': 'Laura',
44    'last_name': 'Callahan'},
45   {'city': 'London',
46    'country': 'UK',
47    'first_name': 'Anne',
48    'last_name': 'Dodsworth'}]}
49

50 # create index
51 collection.create_index({"employees.name":1})
52 collection.drop_index("employees.name_1")
53 collection.create_index({"employees.first_name":1})
54
55 "employees.first_name_1"

56
57 collection=db["northwind"]

58
59 1 data=collection.find({"employees.country":"USA"}, {"employees.first_name", "employees.country"})
60 2 for datas in data:
61 3   # print(dict(datas))
62 4   # print(json.dumps(datas, indent=4, default=json_util.default))  pprint.pprint(datas)
63 5   pprint.pprint(datas)
64 6   # print(datas)

65
66 1 import pprint
67 2 # data=collection.find({filter},{select})
68 3 data=collection.find({'employees.country':{'$eq':'USA'}},{'employees.first_name':1, 'employees.last_name':1, '_id':0})
69 4 for datas in data:
70 5   # print(dict(datas))
71 6   # print(json.dumps(datas, indent=4, default=json_util.default))  pprint.pprint(datas)
72 7   pprint.pprint(datas)

73 None

74
75 collection=db["northwind"]

76
77 1 collection.count_documents({})

```

```
1
```

```
1 # collection.create_index([('customer.city',1)])
2 dt=collection.find({'customers.city':'Berlin'})
3 for data in dt:
4     mydata=data.get("cdata")
5     # print(dict(datas))
6     # print(json.dumps(datas, indent=4, default=json_util.default)))
7     pprint.pprint(mydata)
```

```
None
```

```
1
```

## ▼ Player Data

```
1 ##players data
2 collection=db["player"]

1 collection.count_documents({})
```

```
1034
```

```
1 # team count
2 team=collection.distinct('Team')
3 sum=[]
4 for teams in team:
5     sum.append(teams)
6     print(teams)
7 print("Number of Teams",len(sum))
```

```
ANA
ARZ
ATL
BAL
BOS
CHC
CLE
COL
CWS
Cortina Innovation Network
DET
FLA
HOU
KC
LA
MIN
MLW
NYM
NYY
OAK
PHI
PIT
SD
SEA
SF
STL
TB
TEX
TOR
WAS
Number of Teams 30
```

```
1
```

## ▼ Aggregation

```
db.collection.aggregate(pipeline,option) match / --sort/- avg / --max/min/- sum / --group/- gt / --lt/--
```

```

1 # team player name with group by team
2 import pprint
3 names=collection.aggregate([{'$group':{'_id':'$Team','names_player':{'$push':'$Name'}}}])
4 for name in names:
5     pprint.pprint(name)

    'Reed Johnson',
    'Alex Rios',
    'Vernon Wells',
    'Frank Thomas',
    'Adam Lind',
    'Shaun Marcum',
    'Casey Janssen',
    'Gustavo Chacin',
    'A.J. Burnett',
    'Roy Halladay',
    'John Thomson',
    'Tomo Ohka',
    'B.J. Ryan',
    'Scott Downs',
    'Brian Tallet',
    'Matt Roney',
    'Tracy Thorpe',
    'Jean Machi',
    'Brandon League',
    'Dustin McGowan',
    'Jason Frasor',
    'Francisco Rosario',
    'Davis Romero',
    'Jeremy Accardo']}

{'_id': 'LA',
 'names_player': ['Mike Lieberthal',
                  'Russell Martin',
                  'Olmedo Saenz',
                  'James Loney',
                  'Nomar Garciaparra',
                  'Jeff Kent',
                  'Ramon Martinez',
                  'Marlon Anderson',
                  'Rafael Furcal',
                  'Wilson Betemit',
                  'Andy LaRoche',
                  'Matt Kemp',
                  'Andre Ethier',
                  'Delwyn Young',
                  'Jason Repko',
                  'Juan Pierre',
                  'Luis Gonzalez',
                  'Jason Schmidt',
                  'Randy Wolf',
                  'Brad Penny',
                  'Derek Lowe',
                  'Mark Hendrickson',
                  'Chad Billingsley',
                  'Takashi Saito',
                  'Jonathan Broxton',
                  'Hong-Chih Kuo',
                  'Eric Stults',
                  'Chin-Hui Tsao',
                  'Tim Hamulack',
                  'Yhency Brazoban',
                  'Brett Tomko',
                  'Joe Beimel',
                  'Elmer Dessens']]}

1 # all record using Root
2 names=collection.aggregate([{'$group':{'_id':'$Team','allrecord':{'$push':'$$ROOT'}}}])
3 for name in names:
4     pprint.pprint(name)

```

```

        'Name': 'Brandon Meadres',
        'Position': 'Relief Pitcher',
        'Team': 'ARZ',
        '_id': ObjectId('658e933dc391a7ebaf9d3b59')},
        {'Age': 25.14,
         'Body - Height(inches)': 74,
         'Body - Weight(lbs)': 190,
         'Name': 'Tony Pe?a',
         'Position': 'Relief Pitcher',
         'Team': 'ARZ',
         '_id': ObjectId('658e933dc391a7ebaf9d3b5a')},
        {'Age': 27.07,
         'Body - Height(inches)': 77,
         'Body - Weight(lbs)': 200,
         'Name': 'Doug Slaten',
         'Position': 'Relief Pitcher',
         'Team': 'ARZ',
         '_id': ObjectId('658e933dc391a7ebaf9d3b5b')},
        {'Age': 24.02,
         'Body - Height(inches)': 72,
         'Body - Weight(lbs)': 215,
         'Name': 'Edgar Gonzalez',
         'Position': 'Relief Pitcher',
         'Team': 'ARZ',
         '_id': ObjectId('658e933dc391a7ebaf9d3b5c')},
        {'Age': 27.6,
         'Body - Height(inches)': 76,
         'Body - Weight(lbs)': 254,
         'Name': 'Jose Valverde',
         'Position': 'Relief Pitcher',
         'Team': 'ARZ',
         '_id': ObjectId('658e933dc391a7ebaf9d3b5d')},
        {'Age': 27.99,
         'Body - Height(inches)': 73,
         'Body - Weight(lbs)': 232,
         'Name': 'Jorge Julio',
         'Position': 'Relief Pitcher',
         'Team': 'ARZ',
         '_id': ObjectId('658e933dc391a7ebaf9d3b5e')},
        {'Age': 27.56,
         'Body - Height(inches)': 73,
         'Body - Weight(lbs)': 180,
         'Name': 'Brandon Lyon',
         'Position': 'Relief Pitcher',
         'Team': 'ARZ',
         '_id': ObjectId('658e933dc391a7ebaf9d3b5f')}]}
    }

1 # count name of player who are in COL team
2 names=collection.find({'Team':'COL'},{'Name':1,'_id':0})
3 for name in names:
4     pprint.pprint(name)

{'Name': 'Yorvit Torrealba'}
{'Name': 'Chris Iannetta'}
{'Name': 'Alvin Colina'}
{'Name': 'Todd Helton'}
{'Name': 'Jamey Carroll'}
{'Name': 'Kaz Matsui'}
{'Name': 'Troy Tulowitzki'}
{'Name': 'Clint Barnes'}
{'Name': 'Garrett Atkins'}
{'Name': 'Ryan Spilborghs'}
{'Name': 'Cory Sullivan'}
{'Name': 'Jeff Salazar'}
{'Name': 'Willy Taveras'}
{'Name': 'Matt Holliday'}
{'Name': 'Brad Hawpe'}
{'Name': 'Jeff Baker'}
{'Name': 'Javy Lopez'}
{'Name': 'Byung-Hyun Kim'}
{'Name': 'Rodrigo Lopez'}
{'Name': 'Brian Lawrence'}
{'Name': 'Josh Fogg'}
{'Name': 'Aaron Cook'}
{'Name': 'Denny Bautista'}
{'Name': 'Ubaldo Jimenez'}
{'Name': 'Jason Hirsh'}
{'Name': 'Jeff Francis'}
{'Name': 'Taylor Buchholz'}
{'Name': 'Ryan Speier'}
{'Name': 'Ramon Ramirez'}
{'Name': 'Manny Corpas'}
{'Name': 'Juan Morillo'}

```

```

{'Name': 'Brian Fuentes'}
{'Name': 'LaTroy Hawkins'}
{'Name': 'Tom Martin'}
{'Name': 'Jeremy Affeldt'}

1 # count of player in COL team
2 names=collection.count_documents({'Team':'COL'})
3 print(names)

35

1 # change the name of Cortina Innovation Network to CIN
2 collection.update_many({'Team':'Cortina Innovation Network'},{'$set':{'Team':'CIN'}})
3

1 # total player in Teams
2 names=collection.aggregate([{'$group':{'_id':'$Team','Total':{'$sum':1}})])
3 for name in names:
4     pprint.pprint(name)

{'Total': 33, '_id': 'CWS'}
{'Total': 28, '_id': 'ARZ'}
{'Total': 37, '_id': 'DET'}
{'Total': 33, '_id': 'MIN'}
{'Total': 36, '_id': 'CIN'}
{'Total': 32, '_id': 'STL'}
{'Total': 38, '_id': 'NYM'}
{'Total': 37, '_id': 'OAK'}
{'Total': 33, '_id': 'TB'}
{'Total': 36, '_id': 'PHI'}
{'Total': 34, '_id': 'TOR'}
{'Total': 35, '_id': 'KC'}
{'Total': 35, '_id': 'COL'}
{'Total': 33, '_id': 'LA'}
{'Total': 34, '_id': 'SEA'}
{'Total': 35, '_id': 'MLW'}
{'Total': 33, '_id': 'SD'}
{'Total': 34, '_id': 'SF'}
{'Total': 35, '_id': 'CLE'}
{'Total': 36, '_id': 'BOS'}
{'Total': 35, '_id': 'BAL'}
{'Total': 35, '_id': 'ANA'}
{'Total': 32, '_id': 'FLA'}
{'Total': 37, '_id': 'ATL'}
{'Total': 35, '_id': 'TEX'}
{'Total': 36, '_id': 'WAS'}
{'Total': 36, '_id': 'CHC'}
{'Total': 34, '_id': 'HOU'}
{'Total': 35, '_id': 'PIT'}
{'Total': 32, '_id': 'NYY'}

1 # Max number of player team
2 names=collection.aggregate([{'$group':{'_id':'$Team','Total':{'$sum':1}}},{'$sort':{'Total':-1}},{'$limit':1}])
3 for name in names:
4     pprint.pprint(name)

{'Total': 38, '_id': 'NYM'}

1 # Minimum number of player team
2 names=collection.aggregate([{'$group':{'_id':'$Team','Total':{'$sum':1}}},{'$sort':{'Total':1}},{'$limit':1}])
3 for name in names:
4     pprint.pprint(name)

{'Total': 28, '_id': 'ARZ'}

1 names=collection.find({'Name':'Paul Bako'})
2 for name in names:
3     pprint.pprint(name)

{'Age': 34.69,
 'Body - Height(inches)': 74,
 'Body - Weight(lbs)': 215,
 'Name': 'Paul Bako',
 'Position': 'Catcher',
```

```

'Team': 'BAL',
'_id': ObjectId('658e933dc391a7ebaf9d390c')}

1 # avg_weight by each team
2 names=collection.aggregate([{"$group": {"_id": "$Team", "avgweight": {"$avg": "$Body - Weight(lbs)"}}, {"$sort": {"avgweight": 1}}])
3 for name in names:
4     pprint.pprint(name)

{'_id': 'ANA', 'avgweight': 201.0857142857143}
{'_id': 'BAL', 'avgweight': 196.34285714285716}
{'_id': 'CLE', 'avgweight': 201.65714285714284}
{'_id': 'BOS', 'avgweight': 204.83333333333334}
{'_id': 'SF', 'avgweight': 202.7941176470588}
{'_id': 'SD', 'avgweight': 203.75757575757575}
{'_id': 'MLW', 'avgweight': 205.28571428571428}
{'_id': 'CWS', 'avgweight': 210.5151515151515}
{'_id': 'ARZ', 'avgweight': 208.07142857142858}
{'_id': 'NYY', 'avgweight': 208.3125}
{'_id': 'CHC', 'avgweight': 204.13888888888889}
{'_id': 'HOU', 'avgweight': 198.05882352941177}
{'_id': 'TEX', 'avgweight': 202.6}
{'_id': 'WAS', 'avgweight': 199.75}
{'_id': 'ATL', 'avgweight': 199.51351351351352}
{'_id': 'PIT', 'avgweight': 204.37142857142857}
{'_id': 'FLA', 'avgweight': 202.375}
{'_id': 'TB', 'avgweight': 197.45454545454547}
{'_id': 'OAK', 'avgweight': 198.97297297297297}
{'_id': 'NYM', 'avgweight': 197.21052631578948}
{'_id': 'STL', 'avgweight': 201.625}
{'_id': 'DET', 'avgweight': 203.83783783783784}
{'_id': 'MIN', 'avgweight': 201.1818181818182}
{'_id': 'CIN', 'avgweight': 203.94285714285715}
{'_id': 'SEA', 'avgweight': 198.6764705882353}
{'_id': 'LA', 'avgweight': 203.87878787878788}
{'_id': 'COL', 'avgweight': 198.37142857142857}
{'_id': 'KC', 'avgweight': 195.9142857142857}
{'_id': 'TOR', 'avgweight': 203.85294117647058}
{'_id': 'PHI', 'avgweight': 195.19444444444446}

1 # Team who have Heaviest Player
2 names=collection.aggregate([{"$group": {"_id": "$Team", "max_avgweight": {"$avg": "$Body - Weight(lbs)"}}}, {"$sort": {"avgwei..."}])
3 for name in names:
4     pprint.pprint(name)

{'_id': 'CLE', 'max_avgweight': 201.65714285714284}

1 # all record of Maximum weight Player
2 names=collection.aggregate([{"$sort": {"Body - Weight(lbs)": -1}}, {"$limit": 1}])
3 for name in names:
4     pprint.pprint(name)

{'Age': 26.61,
 'Body - Height(inches)': 79,
 'Body - Weight(lbs)': 290,
 'Name': 'C.C. Sabathia',
 'Position': 'Starting Pitcher',
 'Team': 'CLE',
 '_id': ObjectId('658e933dc391a7ebaf9d39aa')}

1 # Team who have Heaviest Player
2 names=collection.aggregate([{"$group": {"_id": "$Team", "avgweight": {"$max": "$Body - Weight(lbs)"}}, {"$sort": {"avgweight": 1}}])
3 for name in names:
4     pprint.pprint(name)

{'_id': 'CLE', 'avgweight': 290}

```

## ▼ Indexing

```

1 collection.create_index({'Name': 1})
2
3     "Name_1"

```

```

1 collection.index_information()

{'_id_': {'v': 2, 'key': [('_id', 1)]},
 'Name_1': {'v': 2, 'key': [('Name', 1)]}}


1 from sys import executable
2 query={'Name':'Chris Gomez'}
3 names=collection.find(query).explain()
4 names
5
6 # results:
7 # 'inputStage': {'stage': 'IXSCAN',
8 #     'nReturned': 1,
9 #     'totalDocsExamined': 1,
10 #     'totalDocsExamined': 1,
11 #     'executionStages': {'stage': 'FETCH',
12 #         'nReturned': 1,
13 #         'executionTimeMillisEstimate': 0,
14 #         'works': 2,
15 #         'advanced': 1,
16 #         'needTime': 0,
17 #         'needYield': 0,
18 #         'saveState': 0,
19 #         'restoreState': 0,
20 #         'isEOF': 1,
21 #         'docsExamined': 1,
22 #         'alreadyHasObj': 0,
23 #         'inputStage': {'stage': 'IXSCAN',
24 #             'nReturned': 1,
25 #             'executionTimeMillisEstimate': 0,
26 #             'works': 2,
27 #             'advanced': 1,
28 #             'needTime': 0,
29 #             'needYield': 0,
30 #             'saveState': 0,
31 #             'restoreState': 0,
32 #             'isEOF': 1,
33 #             'keyPattern': {'Name': 1},
34 #             'indexName': 'Name_1',
35 #             'isMultiKey': False,
36 #             'multiKeyPaths': {'Name': []},
37 #             'isUnique': False,
38 #             'isSparse': False,
39 #             'isPartial': False,
40 #             'indexVersion': 2,
41 #             'direction': 'forward',
42 #             'indexBounds': {'Name': [[["Chris Gomez", "Chris Gomez"]]]},
43 #             'keysExamined': 1,
44 #             'seeks': 1,
45 #             'dupsTested': 0,
46 #             'dupsDropped': 0}},
47 #     'allPlansExecution': [],
48 #     'command': {'find': 'player',
49 #     'filter': {'Name': 'Chris Gomez'},
50 #     '$db': 'test_data'},
51 #     'serverInfo': {'host': 'ac-imuvfw9-shard-00-02.zga9xvn.mongodb.net',
52 #     'port': 27017,
53 #     'version': '6.0.12',
54 #     'gitVersion': '21e6e8e11a45dfbdb7ca6cf95fa8c5f859e2b118'},
55 #     'serverParameters': {'internalQueryFacetBufferSizeBytes': 104857600,
56 #     'internalQueryFacetMaxOutputDocSizeBytes': 104857600,
57 #     'internalLookupStageIntermediateDocumentMaxSizeBytes': 16793600,
58 #     'internalDocumentSourceGroupMaxMemoryBytes': 104857600,
59 #     'internalQueryMaxBlockingSortMemoryUsageBytes': 33554432,
60 #     'internalQueryProhibitBlockingMergeOnMongoS': 0,
61 #     'internalQueryMaxAddToSetBytes': 104857600,
62 #     'internalDocumentSourceSetWindowFieldsMaxMemoryBytes': 104857600},
63 #     'ok': 1.0,
64 #     '$clusterTime': {'clusterTime': Timestamp(1703973943, 9),
65 #     'signature': {'hash': b'\xa6_\~\x84\xf3&\xd\xae\x0c\xe1.\xb\x95^f\x9b\xab(,',
66 #     'keyId': 7280441575035895810}},
67 #     'operationTime': Timestamp(1703973943, 9)}

```

1

## NorthWind Data

```

1 collection=db['northwind']

1 results=collection.find({},{'products.product_name'})
2 for result in results:
3     for product in result.get('products',[]):
4         print(product)
5 #     pprint.pprint(name)
6 # names

{'product_name': 'Realtime Chocolate Discards'},
{'product_name': "Sir Rodney's Marmalade"},
{'product_name': "Sir Rodney's Scones"},
{'product_name': "Gustaf's Knäckebröd"},
{'product_name': 'Tunnbröd'},
{'product_name': 'Guaraná Fantástica'},
{'product_name': "NuNuCa Nuß-Nougat-Creme"},
{'product_name': "Gumbär Gummibärchen"},
{'product_name': "Schoggi Schokolade"},
{'product_name': "Rössle Sauerkraut"},
{'product_name': "Thüringer Rostbratwurst"},
{'product_name': "Nord-Ost Matjeshering"},
{'product_name': "Gorgonzola Telino"},
{'product_name': "Mascarpone Fabioli"},
{'product_name': "Geitost"},
{'product_name': "Sasquatch Ale"},
{'product_name': "Steeleye Stout"},
{'product_name': "Inlagd Sill"},
{'product_name': "Gravad lax"},
{'product_name': "Côte de Blaye"},
{'product_name': "Chartreuse verte"},
{'product_name': "Boston Crab Meat"},
{'product_name': "Jack's New England Clam Chowder"},
{'product_name': "Singaporean Hokkien Fried Mee"},
{'product_name': "Ipoh Coffee"},
{'product_name': "Gula Malacca"},
{'product_name': "Rogede sild"},
{'product_name': "Spegesild"},
{'product_name': "Zaanse koeken"},
{'product_name': "Chocolade"},
{'product_name': "Maxilaku"},
{'product_name': "Valkoinen suklaa"},
{'product_name': "Manjimup Dried Apples"},
{'product_name': "Filo Mix"},
{'product_name': "Perth Pasties"},
{'product_name': "Tourtière"},
{'product_name': "Pâté chinois"},
{'product_name': "Gnocchi di nonna Alice"},
{'product_name': "Ravioli Angelo"},
{'product_name': "Escargots de Bourgogne"},
{'product_name': "Raclette Courdavault"},
{'product_name': "Camembert Pierrot"},
{'product_name': "Sirop d'érable"},
{'product_name': "Tarte au sucre"},
{'product_name': "Vegie-spread"},
{'product_name': "Wimmers gute Semmelknödel"},
{'product_name': "Louisiana Fiery Hot Pepper Sauce"},
{'product_name': "Louisiana Hot Spiced Okra"},
{'product_name': "Laughing Lumberjack Lager"},
{'product_name': "Scottish Longbreads"},
{'product_name': "Gudbrandsdalsost"},
{'product_name': "Outback Lager"},
{'product_name': "Flotemysost"},
{'product_name': "Mozzarella di Giovanni"},
{'product_name': "Röd Kaviar"},
{'product_name': "Longlife Tofu"},
{'product_name': "Rhönbräu Klosterbier"},
{'product_name': "Lakkaliköori"},
{'product_name': "Original Frankfurter grüne Soße"}


1 import pprint
2 results=collection.find({},{'products.product_id','products.product_name'})
3 for result in results:
4     pprint.pprint(result)

{'_id': ObjectId('65902e2ac391a7ebaf31606f'),
 'products': [{'product_id': 1, 'product_name': 'Chai'},
              {'product_id': 2, 'product_name': 'Chang'},
              {'product_id': 3, 'product_name': 'Aniseed Syrup'},
              {'product_id': 4, 'product_name': "Chef Anton's Cajun Seasoning"},
              {'product_id': 5, 'product_name': "Chef Anton's Gumbo Mix"},
              {'product_id': 6, 'product_name': "Grandma's Boysenberry Spread"}],
}

```

```

{'product_id': 7,
 'product_name': "Uncle Bob's Organic Dried Pears"},

{'product_id': 8, 'product_name': 'Northwoods Cranberry Sauce'},
{'product_id': 9, 'product_name': 'Mishi Kobe Niku'},
{'product_id': 10, 'product_name': 'Ikura'},
{'product_id': 11, 'product_name': 'Queso Cabrales'},
{'product_id': 12, 'product_name': 'Queso Manchego La Pastora'},
{'product_id': 13, 'product_name': 'Konbu'},
{'product_id': 14, 'product_name': 'Tofu'},
{'product_id': 15, 'product_name': 'Genen Shouyu'},
{'product_id': 16, 'product_name': 'Pavlova'},
{'product_id': 17, 'product_name': 'Alice Mutton'},
{'product_id': 18, 'product_name': 'Carnarvon Tigers'},
{'product_id': 19, 'product_name': 'Teatime Chocolate Biscuits'},
{'product_id': 20, 'product_name': "Sir Rodney's Marmalade"},
{'product_id': 21, 'product_name': "Sir Rodney's Scones"},
{'product_id': 22, 'product_name': "Gustaf's Knäckebröd"},
{'product_id': 23, 'product_name': 'Tunnbröd'},
{'product_id': 24, 'product_name': 'Guaraná Fantástica'},
{'product_id': 25, 'product_name': 'NuNuCa Nuß-Nougat-Creme'},
{'product_id': 26, 'product_name': 'Gumbär Gummibärchen'},
{'product_id': 27, 'product_name': 'Schoggi Schokolade'},
{'product_id': 28, 'product_name': 'Rössle Sauerkraut'},
{'product_id': 29, 'product_name': 'Thüringer Rostbratwurst'},
{'product_id': 30, 'product_name': 'Nord-Ost Matjeshering'},
{'product_id': 31, 'product_name': 'Gorgonzola Telino'},
{'product_id': 32, 'product_name': 'Mascarpone Fabioli'},
{'product_id': 33, 'product_name': 'Geitost'},
{'product_id': 34, 'product_name': 'Sasquatch Ale'},
{'product_id': 35, 'product_name': 'Steeleye Stout'},
{'product_id': 36, 'product_name': 'Inlagd Sill'},
{'product_id': 37, 'product_name': 'Gravad lax'},
{'product_id': 38, 'product_name': 'Côte de Blaye'},
{'product_id': 39, 'product_name': 'Chartreuse verte'},
{'product_id': 40, 'product_name': 'Boston Crab Meat'},
{'product_id': 41,
 'product_name': "Jack's New England Clam Chowder"},

{'product_id': 42,
 'product_name': 'Singaporean Hokkien Fried Mee'},
{'product_id': 43, 'product_name': 'Ipoh Coffee'},
{'product_id': 44, 'product_name': 'Gula Malacca'},
{'product_id': 45, 'product_name': 'Rogede sild'},
{'product_id': 46, 'product_name': 'Spegesild'},
{'product_id': 47, 'product_name': 'Zaanse koeken'},
{'product_id': 48, 'product_name': 'Chocolade'},
{'product_id': 49, 'product_name': 'Maxilaku'},
{'product_id': 50, 'product_name': 'Valkoinen suklaa'},
{'product_id': 51, 'product_name': 'Manjimup Dried Apples'},
{'product_id': 52, 'product_name': 'Filo Mix'},
{'product_id': 53, 'product_name': 'Perth Pasties'},

... ...

```

```

1 import pprint
2 results=collection.find_one({'products.product_id':14},{'_id':0,"products.product_name":1})
3 for result in results:
4     print(result)

    products

1 # how many tofu sale: product_id is 14
2 #order_details
3 # sum of quantity where .order_details.product.id=14
4 results=collection.aggregate([{"$match":{"order_details.product_id":11}},
5                             {"$group":{"_id":None,'t_sale':{'$sum':'$order_details.quantity'}}}]
6                             )
7
8 for result in results:
9     pprint.pprint(result)

{'_id': None, 't_sale': 0}

1 # test question
2 # name of product with their categories
3 results=collection.find({},{'products.product_name','categories.category_name'})

```

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.