MongoDB Report

Junhui LI, Ge QIU, Zhiheng WU, Yue JI
A4 IBO4

Dataset:
Stock exchange

1. Introduction

MongoDB is a database based on distributed file storage. It is a product between relational database and non-relational database, which has the most abundant functions and is the most similar to relational database. It supports a very loose data structure, a json-like bson format, so you can store more complex data types. The biggest feature of Mongo is that it supports a very powerful query language. Its syntax is somewhat similar to the object-oriented query language, and it can almost achieve most of the functions similar to the single table query of relational database, and it also supports the index of data.

2. Queries

Question 1.

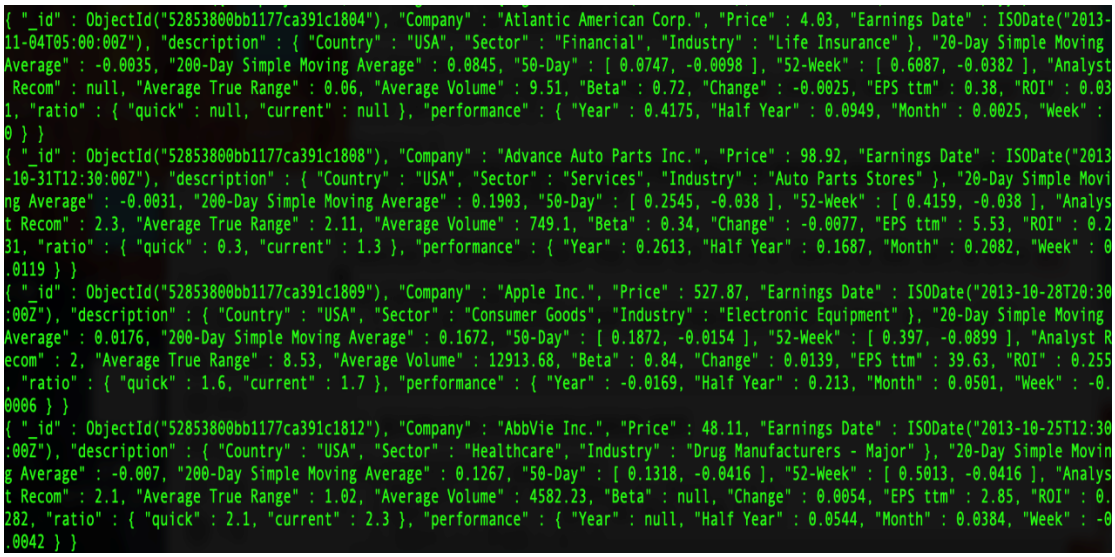
Find stocks which price are between 1000 and 1500.

db.stockCollection.find({Price : {$gt : 1000,$lt: 1500}})

Question 2.

Find stocks which are from '2013-10-25' to '2013-11-07',and started with 'A'.

db.stockCollection.find({"Company":/^A/, "Earnings Date": { $gte: new Date('2013-10-25'), $lte: new Date('2013-11-07') }})

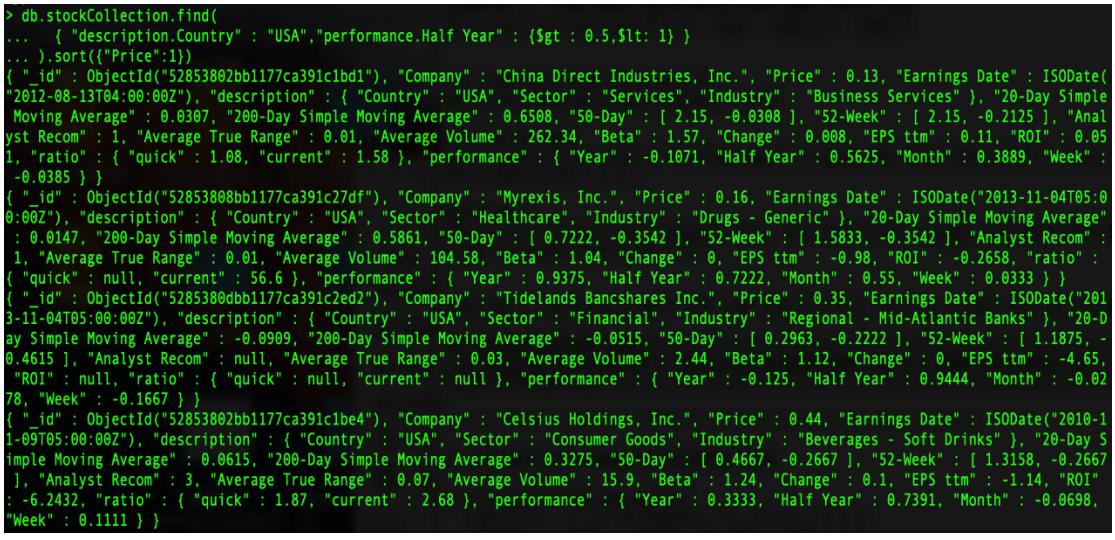


```
{ "_id" : ObjectId("52853800bb1177ca391c1804"), "Company" : "Atlantic American Corp.", "Price" : 4.03, "Earnings Date" : ISODate("2013-
11-04T05:00:00Z"), "description" : { "Country" : "USA", "Sector" : "Financial", "Industry" : "Life Insurance" }, "20-Day Simple Moving
Average" : -0.0035, "200-Day Simple Moving Average" : 0.0845, "50-Day" : [ 0.0747, -0.0098 ], "52-Week" : [ 0.6087, -0.0382 ], "Analyst
 Recom" : null, "Average True Range" : 0.06, "Average Volume" : 9.51, "Beta" : 0.72, "Change" : -0.0025, "EPS ttm" : 0.38, "ROI" : 0.03
1, "ratio" : { "quick" : null, "current" : null }, "performance" : { "Year" : 0.4175, "Half Year" : 0.0949, "Month" : 0.0025, "Week" :
0 } }
{ "_id" : ObjectId("52853800bb1177ca391c1808"), "Company" : "Advance Auto Parts Inc.", "Price" : 98.92, "Earnings Date" : ISODate("2013
-10-31T12:30:00Z"), "description" : { "Country" : "USA", "Sector" : "Services", "Industry" : "Auto Parts Stores" }, "20-Day Simple Movi
ng Average" : -0.0031, "200-Day Simple Moving Average" : 0.1903, "50-Day" : [ 0.2545, -0.038 ], "52-Week" : [ 0.4159, -0.038 ], "Analys
t Recom" : 2.3, "Average True Range" : 2.11, "Average Volume" : 749.1, "Beta" : 0.34, "Change" : -0.0077, "EPS ttm" : 5.53, "ROI" : 0.2
31, "ratio" : { "quick" : 0.3, "current" : 1.3 }, "performance" : { "Year" : 0.2613, "Half Year" : 0.1687, "Month" : 0.2082, "Week" : 0
.0119 } }
{ "_id" : ObjectId("52853800bb1177ca391c1809"), "Company" : "Apple Inc.", "Price" : 527.87, "Earnings Date" : ISODate("2013-10-28T20:30
:00Z"), "description" : { "Country" : "USA", "Sector" : "Consumer Goods", "Industry" : "Electronic Equipment" }, "20-Day Simple Moving
Average" : 0.0176, "200-Day Simple Moving Average" : 0.1672, "50-Day" : [ 0.1872, -0.0154 ], "52-Week" : [ 0.397, -0.0899 ], "Analyst R
ecom" : 2, "Average True Range" : 8.53, "Average Volume" : 12913.68, "Beta" : 0.84, "Change" : 0.0139, "EPS ttm" : 39.63, "ROI" : 0.255
, "ratio" : { "quick" : 1.6, "current" : 1.7 }, "performance" : { "Year" : -0.0169, "Half Year" : 0.213, "Month" : 0.0501, "Week" : -0.
0006 } }
{ "_id" : ObjectId("52853800bb1177ca391c1812"), "Company" : "AbbVie Inc.", "Price" : 48.11, "Earnings Date" : ISODate("2013-10-25T12:30
:00Z"), "description" : { "Country" : "USA", "Sector" : "Healthcare", "Industry" : "Drug Manufacturers - Major" }, "20-Day Simple Movin
g Average" : -0.007, "200-Day Simple Moving Average" : 0.1267, "50-Day" : [ 0.1318, -0.0416 ], "52-Week" : [ 0.5013, -0.0416 ], "Analys
t Recom" : 2.1, "Average True Range" : 1.02, "Average Volume" : 4582.23, "Beta" : null, "Change" : 0.0054, "EPS ttm" : 2.85, "ROI" : 0.
282, "ratio" : { "quick" : 2.1, "current" : 2.3 }, "performance" : { "Year" : null, "Half Year" : 0.0544, "Month" : 0.0384, "Week" : -0
.0042 } }
```


Question 3.

Find U.S. stocks of which half Year are between 0.5 and 1, and sort in ascending order by price.

db.stockCollection.find(
  { "description.Country" : "USA","performance.Half Year" : {$gt : 0.5,$lt: 1} }
).sort({"Price":1})



```
> db.stockCollection.find(
...   { "description.Country" : "USA","performance.Half Year" : {$gt : 0.5,$lt: 1} }
... ).sort({"Price":1})
{ "_id" : ObjectId("52853802bb1177ca391c1bd1"), "Company" : "China Direct Industries, Inc.", "Price" : 0.13, "Earnings Date" : ISODate(
"2012-08-13T04:00:00Z"), "description" : { "Country" : "USA", "Sector" : "Services", "Industry" : "Business Services" }, "20-Day Simple
 Moving Average" : 0.0307, "200-Day Simple Moving Average" : 0.6508, "50-Day" : [ 2.15, -0.0308 ], "52-Week" : [ 2.15, -0.2125 ], "Anal
yst Recom" : 1, "Average True Range" : 0.01, "Average Volume" : 262.34, "Beta" : 1.57, "Change" : 0.008, "EPS ttm" : 0.11, "ROI" : 0.05
1, "ratio" : { "quick" : 1.08, "current" : 1.58 }, "performance" : { "Year" : -0.1071, "Half Year" : 0.5625, "Month" : 0.3889, "Week" :
 -0.0385 } }
{ "_id" : ObjectId("52853808bb1177ca391c27df"), "Company" : "Myrexis, Inc.", "Price" : 0.16, "Earnings Date" : ISODate("2013-11-04T05:0
0:00Z"), "description" : { "Country" : "USA", "Sector" : "Healthcare", "Industry" : "Drugs - Generic" }, "20-Day Simple Moving Average"
 : 0.0147, "200-Day Simple Moving Average" : 0.5861, "50-Day" : [ 0.7222, -0.3542 ], "52-Week" : [ 1.5833, -0.3542 ], "Analyst Recom" :
 1, "Average True Range" : 0.01, "Average Volume" : 104.58, "Beta" : 1.04, "Change" : 0, "EPS ttm" : -0.98, "ROI" : -0.2658, "ratio" :
{ "quick" : null, "current" : 56.6 }, "performance" : { "Year" : 0.9375, "Half Year" : 0.7222, "Month" : 0.55, "Week" : 0.0333 } }
{ "_id" : ObjectId("5285380dbb1177ca391c2ed2"), "Company" : "Tidelands Bancshares Inc.", "Price" : 0.35, "Earnings Date" : ISODate("201
3-11-04T05:00:00Z"), "description" : { "Country" : "USA", "Sector" : "Financial", "Industry" : "Regional - Mid-Atlantic Banks" }, "20-D
ay Simple Moving Average" : -0.0909, "200-Day Simple Moving Average" : -0.0515, "50-Day" : [ 0.2963, -0.2222 ], "52-Week" : [ 1.1875, -
0.4615 ], "Analyst Recom" : null, "Average True Range" : 0.03, "Average Volume" : 2.44, "Beta" : 1.12, "Change" : 0, "EPS ttm" : -4.65,
 "ROI" : null, "ratio" : { "quick" : null, "current" : null }, "performance" : { "Year" : -0.125, "Half Year" : 0.9444, "Month" : -0.02
78, "Week" : -0.1667 } }
{ "_id" : ObjectId("52853802bb1177ca391c1be4"), "Company" : "Celsius Holdings, Inc.", "Price" : 0.44, "Earnings Date" : ISODate("2010-1
1-09T05:00:00Z"), "description" : { "Country" : "USA", "Sector" : "Consumer Goods", "Industry" : "Beverages - Soft Drinks" }, "20-Day S
imple Moving Average" : 0.0615, "200-Day Simple Moving Average" : 0.3275, "50-Day" : [ 0.4667, -0.2667 ], "52-Week" : [ 1.3158, -0.2667
 ], "Analyst Recom" : 3, "Average True Range" : 0.07, "Average Volume" : 15.9, "Beta" : 1.24, "Change" : 0.1, "EPS ttm" : -1.14, "ROI"
: -6.2432, "ratio" : { "quick" : 1.87, "current" : 2.68 }, "performance" : { "Year" : 0.3333, "Half Year" : 0.7391, "Month" : -0.0698,
"Week" : 0.1111 } }
```

Question 4.

Find companies and prices of stocks with prices between 50 and 100, and sort in descending order by price.

```
db.stockCollection.aggregate([
   { $project : { _id :0,Company : 1 , Price : 1 } },
   { $match : { Price : { $gt : 50, $lte : 100 } } },
   { "$sort" : { Price : -1 } }
])
```

```
> db.stockCollection.aggregate([
...    { $project : { _id :0,Company : 1 , Price : 1 } },
...    { $match : { Price : { $gt : 50, $lte : 100 } } },
...    { "$sort" : { Price : -1 } }
... ])
{ "Company" : "iSharesBond 2016 Corp ex-Fincls Term", "Price" : 99.96 }
{ "Company" : "Thermo Fisher Scientific, Inc.", "Price" : 99.84 }
{ "Company" : "iShares Dow Jones US Aerospace & Defense", "Price" : 99.5 }
{ "Company" : "iSharesBond 2023 Corporate Term", "Price" : 99.4 }
{ "Company" : "ProShares UltraPro Dow30", "Price" : 99.01 }
{ "Company" : "Advance Auto Parts Inc.", "Price" : 98.92 }
{ "Company" : "SVB Financial Group", "Price" : 98.89 }
{ "Company" : "Vanguard Health Care ETF", "Price" : 98.82 }
{ "Company" : "Winland Electronics Inc.", "Price" : 98.78 }
{ "Company" : "ProShares Ultra Technology", "Price" : 98.72 }
{ "Company" : "iSharesBond 2018 Corp ex-Fincls Term", "Price" : 98.71 }
{ "Company" : "Hershey Co.", "Price" : 98.4 }
{ "Company" : "ACE Limited", "Price" : 98.31 }
{ "Company" : "Vanguard S&P Small-Cap 600 Gr Idx ETF", "Price" : 98.3 }
{ "Company" : "iShares Dow Jones US Utilities", "Price" : 98.05 }
{ "Company" : "Tyler Technologies, Inc.", "Price" : 97.9 }
{ "Company" : "NetSuite Inc.", "Price" : 97.81 }
{ "Company" : "CurrencyShares Japanese Yen Trust", "Price" : 97.74 }
{ "Company" : "ProShares Ultra SmallCap600", "Price" : 97.57 }
{ "Company" : "McDonald's Corp.", "Price" : 97.48 }
Type "it" for more
```

Question 5.

Calculate the number of stocks of each country, and sort in descending order by total.

```
db.stockCollection.aggregate([
{"$group":{"_id":"$description.Country", "total" : { "$sum" : 1}}},
{ "$sort" : { total : -1 } }]);
```

```
> db.stockCollection.aggregate([
... {"$group":{"_id":"$description.Country", "total" : { "$sum" : 1}}},
... { "$sort" : { total : -1 } }]);
{ "_id" : "USA", "total" : 5914 }
{ "_id" : "Canada", "total" : 176 }
{ "_id" : "China", "total" : 175 }
{ "_id" : "Israel", "total" : 65 }
{ "_id" : "United Kingdom", "total" : 42 }
{ "_id" : "Bermuda", "total" : 42 }
{ "_id" : "Brazil", "total" : 31 }
{ "_id" : "Netherlands", "total" : 25 }
{ "_id" : "Greece", "total" : 24 }
{ "_id" : "Japan", "total" : 24 }
{ "_id" : "Ireland", "total" : 21 }
{ "_id" : "Hong Kong", "total" : 20 }
{ "_id" : "Mexico", "total" : 17 }
{ "_id" : "Switzerland", "total" : 16 }
{ "_id" : "Argentina", "total" : 15 }
{ "_id" : "Chile", "total" : 14 }
{ "_id" : "Taiwan", "total" : 12 }
{ "_id" : "India", "total" : 12 }
{ "_id" : "South Korea", "total" : 10 }
{ "_id" : "Australia", "total" : 10 }
Type "it" for more
```

Question 6.

Show the companies and prices of stocks of the USA, and sort in descending order by price.

```
db.stockCollection.aggregate(
    {"$unwind":"$description"},
    {"$match":{"description.Country":"USA"}},
    {$project: {_id: 0,
                Company: "$Company",
                Price: "$Price" }
     },
{ "$sort" : { Price : -1 } })
```

```
> db.stockCollection.aggregate(
...     {"$unwind":"$description"},
...     {"$match":{"description.Country":"USA"}},
...     {$project: {
...                 _id: 0,
...                 Company: "$Company",
...                 Price: "$Price" }
...      },
... { "$sort" : { Price : -1 } }
... )
{ "Company" : "Berkshire Hathaway Inc.", "Price" : 173099.5 }
{ "Company" : "Seaboard Corp.", "Price" : 2750 }
{ "Company" : "priceline.com Incorporated", "Price" : 1137.47 }
{ "Company" : "Google Inc.", "Price" : 1037.5 }
{ "Company" : "NVR Inc.", "Price" : 930.51 }
{ "Company" : "Mastercard Incorporated", "Price" : 742.38 }
{ "Company" : "The Washington Post Company", "Price" : 651.8 }
{ "Company" : "Altisource Asset Management Corporation", "Price" : 597.98 }
{ "Company" : "Chipotle Mexican Grill, Inc.", "Price" : 543.75 }
{ "Company" : "Markel Corp.", "Price" : 540.38 }
{ "Company" : "Apple Inc.", "Price" : 527.87 }
{ "Company" : "AutoZone Inc.", "Price" : 458.44 }
{ "Company" : "Biglari Holdings Inc.", "Price" : 457.2 }
{ "Company" : "Intuitive Surgical, Inc.", "Price" : 397.53 }
{ "Company" : "Alleghany Corp.", "Price" : 395.65 }
{ "Company" : "Amazon.com Inc.", "Price" : 367.23 }
{ "Company" : "Netflix, Inc.", "Price" : 343.7 }
{ "Company" : "NewMarket Corp.", "Price" : 320.14 }
{ "Company" : "Alexander's Inc.", "Price" : 303.7 }
{ "Company" : "BlackRock, Inc.", "Price" : 302.64 }
Type "it" for more
```

Question 7.

Show the average price of all stocks of each country, and sort in descending order by avgprice.

```
db.stockCollection.aggregate([
{"$group":{"_id":"$description.Country", "avgPrice" : { "$avg" : "$Price"}}},
{ "$sort" : { avgPrice : -1 } }]);
```

```
> db.stockCollection.aggregate([
...  {"$group":{"_id":"$description.Country", "avgPrice" : { "$avg" : "$Price"}}},
... { "$sort" : { avgPrice : -1 } }]);
{ "_id" : "Denmark", "avgPrice" : 173.56 }
{ "_id" : "Panama", "avgPrice" : 90.33 }
{ "_id" : "Belgium", "avgPrice" : 81.075 }
{ "_id" : "Philippines", "avgPrice" : 64.18 }
{ "_id" : "USA", "avgPrice" : 63.47855472721111 }
{ "_id" : "Switzerland", "avgPrice" : 58.331875000000004 }
{ "_id" : "Germany", "avgPrice" : 53.857142857142854 }
{ "_id" : "Sweden", "avgPrice" : 51.339999999999996 }
{ "_id" : "Bermuda", "avgPrice" : 50.80880952380953 }
{ "_id" : "Peru", "avgPrice" : 44.3425 }
{ "_id" : "Netherlands", "avgPrice" : 40.464 }
{ "_id" : "Channel Islands", "avgPrice" : 39.95 }
{ "_id" : "United Kingdom", "avgPrice" : 39.3619512195122 }
{ "_id" : "Luxembourg", "avgPrice" : 39.12285714285714 }
{ "_id" : "Japan", "avgPrice" : 39.08583333333333 }
{ "_id" : "Indonesia", "avgPrice" : 38.59 }
{ "_id" : "Mexico", "avgPrice" : 35.476470588235294 }
{ "_id" : "Colombia", "avgPrice" : 35.45333333333333 }
{ "_id" : "Ireland", "avgPrice" : 33.055238095238096 }
{ "_id" : "Chile", "avgPrice" : 32.104285714285716 }
Type "it" for more
```

Question 8.

Show details of the most expensive stocks in each sector,and sort in descending order by max.

```
db.stockCollection.aggregate([
{"$group":{"_id":"$description.Sector", "max" : { "$max" : "$Price"}}},
{ "$sort" : { max : -1 } }]);
```

```
> db.stockCollection.aggregate([
... {"$group":{"_id":"$description.Sector", "max" : { "$max" : "$Price"}}},
... { "$sort" : { max : -1 } }]);
{ "_id" : "Financial", "max" : 173099.5 }
{ "_id" : "Consumer Goods", "max" : 2750 }
{ "_id" : "Services", "max" : 1137.47 }
{ "_id" : "Technology", "max" : 1037.5 }
{ "_id" : "Industrial Goods", "max" : 930.51 }
{ "_id" : "Healthcare", "max" : 397.53 }
{ "_id" : "Basic Materials", "max" : 320.14 }
{ "_id" : "Conglomerates", "max" : 281.15 }
{ "_id" : "Utilities", "max" : 93.85 }
```

Question 9.

Show how many stocks each country has in each sector, and sort in ascending order by id.coutry,descending order by count.

```
db.stockCollection.aggregate([{
"$group":{
      "_id":{"coutry":"$description.Country","sector":"$description.Sector"},
      count:{$sum:1}
      }
},
{ "$sort" : { "_id.coutry": 1,count :-1 } }]);
```

```
> db.stockCollection.aggregate([{
... "$group":{
... "_id":{"coutry":"$description.Country","sector":"$description.Sector"},
... count:{$sum:1}
... }
... },
[... { "$sort" : { "_id.coutry": 1,count :-1 } }]);
{ "_id" : { "coutry" : "Argentina", "sector" : "Financial" }, "count" : 5 }
{ "_id" : { "coutry" : "Argentina", "sector" : "Utilities" }, "count" : 3 }
{ "_id" : { "coutry" : "Argentina", "sector" : "Basic Materials" }, "count" : 2 }
{ "_id" : { "coutry" : "Argentina", "sector" : "Technology" }, "count" : 2 }
{ "_id" : { "coutry" : "Argentina", "sector" : "Services" }, "count" : 2 }
{ "_id" : { "coutry" : "Argentina", "sector" : "Consumer Goods" }, "count" : 1 }
{ "_id" : { "coutry" : "Australia", "sector" : "Healthcare" }, "count" : 4 }
{ "_id" : { "coutry" : "Australia", "sector" : "Basic Materials" }, "count" : 4 }
{ "_id" : { "coutry" : "Australia", "sector" : "Consumer Goods" }, "count" : 1 }
{ "_id" : { "coutry" : "Australia", "sector" : "Financial" }, "count" : 1 }
{ "_id" : { "coutry" : "Bahamas", "sector" : "Services" }, "count" : 2 }
{ "_id" : { "coutry" : "Belgium", "sector" : "Services" }, "count" : 1 }
{ "_id" : { "coutry" : "Belgium", "sector" : "Consumer Goods" }, "count" : 1 }
{ "_id" : { "coutry" : "Bermuda", "sector" : "Financial" }, "count" : 18 }
{ "_id" : { "coutry" : "Bermuda", "sector" : "Services" }, "count" : 15 }
{ "_id" : { "coutry" : "Bermuda", "sector" : "Basic Materials" }, "count" : 4 }
{ "_id" : { "coutry" : "Bermuda", "sector" : "Utilities" }, "count" : 2 }
{ "_id" : { "coutry" : "Bermuda", "sector" : "Technology" }, "count" : 2 }
{ "_id" : { "coutry" : "Bermuda", "sector" : "Consumer Goods" }, "count" : 1 }
{ "_id" : { "coutry" : "Brazil", "sector" : "Basic Materials" }, "count" : 8 }
Type "it" for more
```

Question 10.

Find how many German stocks are priced between 30 and 100 in each sector,and calculate the average price, sort in descending order by count.

```
db.stockCollection.aggregate(
    {"$unwind":"$description"},
    {"$match":{"description.Country":"Germany",Price:{$gte :30 , $lte:100 }}},
    {"$group":{"_id":"$description.Sector","count" : { "$sum" : 1},"avgPrice" : { "$avg" : "$Price"}}},
    {"$sort" : { count : -1 } }
)
```

```
> db.stockCollection.aggregate(
...     {"$unwind":"$description"},
...     {"$match":{"description.Country":"Germany",Price:{$gte :30 , $lte:100 }}},
...     {"$group":{"_id":"$description.Sector","count" : { "$sum" : 1},"avgPrice" : { "$avg" : "$Price"}}},
...     {"$sort" : { count : -1 } }
... )
{ "_id" : "Technology", "count" : 2, "avgPrice" : 66.51 }
{ "_id" : "Healthcare", "count" : 1, "avgPrice" : 32.17 }
{ "_id" : "Financial", "count" : 1, "avgPrice" : 45.86 }
```

3. Conclusion:

This project really allowed us to understand the basics of MongoDB. We realized that an organization, clarity and good structure in the MongoDB is paramount so as not to fall into the mistakes and pitfalls of such language.

Finally, this project allowed us to concretely practice the MongoDB and the data structure, which may later in our journey help us better understand the logic of NoSQL.