

This is a practice project for anyone who wants to practice using MongoDB. The dataset used in this project is a json file containing information about different countries. The dataset is in correct format, however, all the data itself is not correct or updated. It does provide a really good sample dataset to practice using MongoDB and thus, one can ignore the false facts. What is here meant by some false information can be demonstrated with these results:

```
[ db.countryInfo.distinct("name", {"$and" : [{"region": "Europe" }, {"currency": "EUR"}]}).sort()
  "Aland Islands",
  "Andorra",
  "Austria",
  "Belgium",
  "Cyprus",
  "Estonia",
  "Finland",
  "France",
  "Germany",
  "Greece",
  "Ireland",
  "Italy",
  "Kosovo",
  "Latvia",
  "Lithuania",
  "Luxembourg",
  "Malta",
  "Monaco",
  "Montenegro",
  "Netherlands",
  "Portugal",
  "San Marino",
  "Slovakia",
  "Slovenia",
  "Spain",
  "Vatican City State (Holy See)"
]
```

Here we can see that Aland Islands are listed as an individual country even though it is part of Finland.

These small faults can be ignored, and the dataset used for practice purposes.

1. First, we need to import the json file to MongoDB. This can be done on the following way:

```
C:\Program Files\MongoDB\Server\5.0\bin>mongoimport --db=WBProjects --collection=countryInfo --jsonArray --file "C:\Program Files\MongoDB\countries.json"
2021-11-09T08:54:53.477+0100   connected to: mongodb://localhost/
2021-11-09T08:54:53.535+0100   250 document(s) imported successfully. 0 document(s) failed to import.
C:\Program Files\MongoDB\Server\5.0\bin>
```

2. Start the MongoDB shell to connect to the MongoDB server:

```
C:\Users\sebastiano>cd "C:\Program Files\MongoDB\Server\5.0\bin"
C:\Program Files\MongoDB\Server\5.0\bin>mongo
```

3. Now we can start issuing commands to the server and switch to a specific database. In the following screenshot we switch to WBProject Database. Show collections in it and show all the information inside of countryInfo (where the practice file is uploaded).

```
> use WBProjects
switched to db WBProjects
> show collections
airbnb
countryInfo
world_bank
> db.countryInfo.find()
{ "_id" : ObjectId("618a294d73166f12b5705a63"), "id" : 2, "name" : "Aland Islands", "iso3" : "ALA", "iso2" : "AX", "numeric_code" : "248", "phone_code" : "+358-18", "capital" : "Mariehamn", "currency" : "EUR", "currency_symbol" : "€", "tld" : ".ax", "native" : "Åland", "region" : "Europe", "subregion" : "Northern Europe", "timezones" : [ { "zoneName" : "Europe/Mariehamn", "gmtOffset" : 7200, "gmtOffsetName" : "UTC+02:00", "abbreviation" : "EET", "tzName" : "Eastern European Time" } ], "translations" : { "kr" : "Åland", "bn" : "Îlhas de Aland", "pt" : "Ilhas de Aland", "nl" : "Åländska öarna", "hr" : "Ålandski otoci", "fa" : "جزایر آلند", "de" : "Åland", "es" : "Alandia", "fr" : "Åland", "ja" : "Åland", "it" : "Isola Åland", "cn" : "Åland", "latitude" : "60.11666700", "longitude" : "19.90000000", "emoji" : "🇦🇱", "emojiU" : "U+1F1E6 U+1F1F6" } },
{ "_id" : ObjectId("618a294d73166f12b5705a64"), "id" : 6, "name" : "Andorra", "iso3" : "AND", "iso2" : "AD", "numeric_code" : "020", "phone_code" : "376", "capital" : "Andorra la Vella", "currency" : "EUR", "currency_symbol" : "€", "tld" : ".ad", "native" : "Andorra", "region" : "Europe", "subregion" : "Southern Europe", "timezones" : [ { "zoneName" : "Europe/Andorra", "gmtOffset" : 3600, "gmtOffsetName" : "UTC+01:00", "abbreviation" : "CET", "tzName" : "Central European Time" } ], "translations" : { "kr" : "안도라", "bn" : "আন্দোরা", "pt" : "Andorra", "nl" : "Andorra", "hr" : "Andora", "fa" : "آندورا", "de" : "Andorra", "es" : "Andorra", "fr" : "Andorre", "ja" : "アンドール", "it" : "Andorra", "cn" : "安多拉", "latitude" : "42.50000000", "longitude" : "1.50000000", "emoji" : "🇦🇩", "emojiU" : "U+1F1E6 U+1F1E9" } },
{ "_id" : ObjectId("618a294d73166f12b5705a65"), "id" : 10, "name" : "Antigua And Barbuda", "iso3" : "ATG", "iso2" : "AG", "numeric_code" : "028", "phone_code" : "+1-268", "capital" : "St. John's", "currency" : "XCD", "currency_symbol" : "$", "tld" : ".ag", "native" : "Antigua and Barbuda", "region" : "Americas", "subregion" : "Caribbean", "timezones" : [ { "zoneName" : "America/Antigua", "gmtOffset" : -14400, "gmtOffsetName" : "UTC-04:00", "abbreviation" : "AST", "tzName" : "Atlantic Standard Time" } ], "translations" : { "kr" : "앤티가 바부다", "bn" : "আন্তিগা ও বার্বুদা", "pt" : "Antigua e Barbuda", "nl" : "Antigua en Barbuda", "hr" : "Antigva i Barbuda", "fa" : "آنتیگوا و باربودا", "de" : "Antigua und Barbuda", "es" : "Antigua y Barbuda", "fr" : "Antigua-et-Barbuda", "ja" : "アンティグア・バルブダ", "it" : "Antigua e Barbuda", "cn" : "安提瓜和巴布达", "latitude" : "17.05000000", "longitude" : "-61.80000000", "emoji" : "🇦🇩", "emojiU" : "U+1F1E6 U+1F1EC" } },
{ "_id" : ObjectId("618a294d73166f12b5705a66"), "id" : 9, "name" : "Antarctica", "iso3" : "ATA", "iso2" : "AQ", "numeric_code" : "010", "phone_code" : "672", "capital" : "", "currency" : "AAD", "currency_symbol" : "$", "tld" : ".aq", "native" : "Antarctica", "region" : "Polar", "subregion" : "", "timezones" : [ { "zoneName" : "Antarctica/Casey", "gmtOffset" : 39600, "gmtOffsetName" : "UTC+11:00", "abbreviation" : "AWST", "tzName" : "Australian Western Standard Time" }, { "zoneName" : "Antarctica/Davis", "gmtOffset" : 25200, "gmtOffsetName" : "UTC+07:00", "abbreviation" : "DAVT", "tzName" : "Davis Time" }, { "zoneName" : "Antarctica/DumontD'Urville", "gmtOffset" : 36000, "gmtOffsetName" : "UTC+10:00", "abbreviation" : "DDUT", "tzName" : "Dumont d'Urville Time" }, { "zoneName" : "Antarctica/Mawson", "gmtOffset" : 18000, "gmtOffsetName" : "UTC+05:00", "abbreviation" : "MAWT", "tzName" : "Mawson Station Time" }, { "zoneName" : "Antarctica/McMurdo", "gmtOffset" : 46800, "gmtOffsetName" : "UTC+13:00", "abbreviation" : "NZDT", "tzName" : "New Zealand Daylight Time" }, { "zoneName" : "Antarctica/Palmer", "gmtOffset" : -18000, "gmtOffsetName" : "UTC-03:00", "abbreviation" : "CLST", "tzName" : "Chile Summer Time" }, { "zoneName" : "Antarctica/Rothera", "gmtOffset" : -18000, "gmtOffsetName" : "UTC-03:00", "abbreviation" : "ROTT", "tzName" : "Rothera Research Station Time" }, { "zoneName" : "Antarctica/Syowa", "gmtOffset" : 10800, "gmtOffsetName" : "UTC+03:00", "abbreviation" : "SYOT", "tzName" : "Showa Station Time" }, { "zoneName" : "Antarctica/Troll", "gmtOffset" : 0, "gmtOffsetName" : "UTC+00", "abbreviation" : "GMT", "tzName" : "Greenwich Mean Time" }, { "zoneName" : "Antarctica/Vostok", "gmtOffset" : 21600, "gmtOffsetName" : "UTC+06:00", "abbreviation" : "VOST", "tzName" : "Vostok Station Time" } ], "translations" : { "kr" : "안타르티카", "bn" : "আন্তার্কটিকা", "pt" : "Antártida", "nl" : "Antarktika", "hr" : "Antarktika", "fa" : "آنتارکیتیکا", "de" : "Antarktika", "es" : "Antártica", "fr" : "Antarctique", "ja" : "アンタールティカ", "it" : "Antartide", "cn" : "南极洲", "latitude" : "-74.65000000", "longitude" : "4.48000000", "emoji" : "🇦🇩", "emojiU" : "U+1F1E6 U+1F1F6" } },
{ "_id" : ObjectId("618a294d73166f12b5705a67"), "id" : 7, "name" : "Angola", "iso3" : "AGO", "iso2" : "AO", "numeric_code" : "024", "phone_code" : "244", "capital" : "Luanda", "currency" : "AOA", "currency_symbol" : "Kz", "tld" : ".ao", "native" : "Angola", "region" : "Africa", "subregion" : "Middle Africa", "timezones" : [ { "zoneName" : "Africa/Luanda", "gmtOffset" : 3600, "gmtOffsetName" : "UTC+01:00", "abbreviation" : "WAT", "tzName" : "West Africa Time" } ], "translations" : { "kr" : "앙골라", "bn" : "আঙ্গোলা", "pt" : "Angola", "nl" : "Angola", "hr" : "Angola", "fa" : "آنگولا", "de" : "Angola", "es" : "Angola", "fr" : "Angola", "ja" : "アンゴラ", "it" : "Angola", "cn" : "安哥拉", "latitude" : "60.11666700", "longitude" : "19.90000000", "emoji" : "🇦🇩", "emojiU" : "U+1F1E6 U+1F1F6" } }
```

4. To make more sense of data we can use pretty() method to change the output to easier to read format.

```
type "it" for more
> db.countryInfo.find().pretty()
{
  "_id" : ObjectId("618a294d73166f12b5705a63"),
  "id" : 2,
  "name" : "Aland Islands",
  "iso3" : "ALA",
  "iso2" : "AX",
  "numeric_code" : "248",
  "phone_code" : "+358-18",
  "capital" : "Mariehamn",
  "currency" : "EUR",
  "currency_symbol" : "€",
  "tld" : ".ax",
  "native" : "Åland",
  "region" : "Europe",
  "subregion" : "Northern Europe",
  "timezones" : [
    {
      "zoneName" : "Europe/Mariehamn",
      "gmtOffset" : 7200,
      "gmtOffsetName" : "UTC+02:00",
      "abbreviation" : "EET",
      "tzName" : "Eastern European Time"
    }
  ],
  "translations" : {
    "kr" : "Åland",
    "bn" : "Îlhas de Aland",
    "pt" : "Ilhas de Aland",
    "nl" : "Åländska öarna",
    "hr" : "Ålandski otoci",
    "fa" : "جزایر آلند",
    "de" : "Åland",
    "es" : "Alandia",
    "fr" : "Åland",
    "ja" : "Åland",
    "it" : "Isola Åland",
    "cn" : "Åland",
    "latitude" : "60.11666700",
    "longitude" : "19.90000000",
    "emoji" : "🇦🇱",
    "emojiU" : "U+1F1E6 U+1F1F6"
  }
}
```

5. db.stats() method is used to return a document that reports on the state of the current database.

```

> db.stats()
{
  "db" : "WBProjects",
  "collections" : 3,
  "views" : 0,
  "objects" : 251,
  "avgObjSize" : 945.0438247011953,
  "dataSize" : 237206,
  "storageSize" : 135168,
  "freeStorageSize" : 0,
  "indexes" : 3,
  "indexSize" : 45056,
  "indexFreeStorageSize" : 0,
  "totalSize" : 180224,
  "totalFreeStorageSize" : 0,
  "scaleFactor" : 1,
  "fsUsedSize" : 80928571392,
  "fsTotalSize" : 510770802688,
  "ok" : 1
}
>

```

6. Here we want to count all the records in countryInfo collection.

```

> db.countryInfo.find().count()
250
>

```

7. With SQL one can retrieve a single record from a database with: `SELECT* FROM countryInfo LIMIT 1;` - With MongoDB we can write the same query by typing:

```

> db.countryInfo.findOne()
{
  "_id" : ObjectId("618a294d73166f12b5705a63"),
  "id" : 2,
  "name" : "Åland Islands",
  "iso3" : "ALA",
  "iso2" : "AX",
  "numeric_code" : "248",
  "phone_code" : "+358-18",
  "capital" : "Mariehamn",
  "currency" : "EUR",
  "currency_symbol" : "€",
  "tld" : ".ax",
  "native" : "Åland",
  "region" : "Europe",
  "subregion" : "Northern Europe",
  "timezones" : [
    {
      "zoneName" : "Europe/Mariehamn",
      "gmtOffset" : 7200,
      "gmtOffsetName" : "UTC+02:00",
      "abbreviation" : "EET",
      "tzName" : "Eastern European Time"
    }
  ],
  "translations" : {
    "kr" : "Ålândia",
    "br" : "Ilhas de Aland",
    "pt" : "Ilhas de Aland",
    "nl" : "Åländeilanden",
    "hr" : "Ålandski otoci",
    "fa" : "جزایر آلاند",
    "de" : "Åland",
    "es" : "Alandia",
    "fr" : "Åland",
    "ja" : "Åland",

```

8. We can add an aggregate method count() in order to count the number of record that match a query. Here we want to know how many countries are listed in the region of Europe.

```
> db.countryInfo.find( {"region" : "Europe" } ).count()
53
>
```

9. MongoDB provides equivalents to the WHERE clause in SQL in the form of Query and Projection operators. Here we specify that a value of a certain key must be one of those in each list. Here we want to know how many countries there are in Europe and Africa regions together. Countries are also shown with pretty()

```
0
> db.countryInfo.find( {"region" : {"$in" : ["Europe", "Africa"]} }).count()
113
> db.countryInfo.find({"region" : {"$in" : ["Europe", "Africa"]} }).pretty()
{
  "_id" : ObjectId("618a294d73166f12b5705a63"),
  "id" : 2,
  "name" : "Aland Islands",
  "iso3" : "ALA",
  "iso2" : "AX",
  "numeric_code" : "248",
  "phone_code" : "+358-18",
  "capital" : "Mariehamn",
  "currency" : "EUR",
  "currency_symbol" : "€",
  "tld" : ".ax",
  "native" : "Åland",
  "region" : "Europe",
  "subregion" : "Northern Europe",
  "timezones" : [
    {
      "zoneName" : "Europe/Mariehamn",
      "gmtOffset" : 7200,
      "gmtOffsetName" : "UTC+02:00",
      "abbreviation" : "EET",
      "tzName" : "Eastern European Time"
    }
  ],
  "translations" : {
    "kr" : "Ålândia",
    "br" : "Ilhas de Aland",
    "pt" : "Ilhas de Aland",
    "nl" : "Åländeilanden",
    "hr" : "Ålandski otoci",
    "fa" : "جزایر آلاند",
    "de" : "Åland",
    "es" : "Alandia",
    "fr" : "Åland",
    "ja" : "Åland諸島",
    "it" : "Isole Aland",
    "cn" : "Åland群岛"
  }
}
```

10. We can combine two selection criteria with the operator \$and:

```
> db.countryInfo.find({"$and" : [{"region" : "Europe"}, {"currency" : "EUR"}]}).count()
26
>
```

11. There are also methods for aggregation in MongoDB. Here `distinct()` is used to get all distinct countries in Europe with EUR currency:

```
[ db.countryInfo.distinct("name", {"$and" : [{"region": "Europe" }, {"currency": "EUR"}]}).sort()
  "Aland Islands",
  "Andorra",
  "Austria",
  "Belgium",
  "Cyprus",
  "Estonia",
  "Finland",
  "France",
  "Germany",
  "Greece",
  "Ireland",
  "Italy",
  "Kosovo",
  "Latvia",
  "Lithuania",
  "Luxembourg",
  "Malta",
  "Monaco",
  "Montenegro",
  "Netherlands",
  "Portugal",
  "San Marino",
  "Slovakia",
  "Slovenia",
  "Spain",
  "Vatican City State (Holy See)"
]
```

12. To show how many countries have latitude greater than 4.00000000, we can do following:

```
> db.countryInfo.find({"latitude" : {"$gt" : 4.00000000} }).count()
0
> db.countryInfo.find({"latitude" : {"$gt" : "4.00000000"}}).count()
78
>
```

Note the difference here when not using "" for the 4.00000000 – this is due to the data formatting on the source file.

13. Here we show the latitude in descending order and limit the results to one:

```
> db.countryInfo.find().sort({"latitude" : -1}).limit(1)
{ "_id" : ObjectId("618a294d73166f12b5705a76"), "id" : 24, "name" : "Benin", "iso3" : "BEN", "iso2" : "BJ", "numeric_code" : "204", "phone_code" : "229", "capital" : "Porto
Novo", "currency" : "XOF", "currency_symbol" : "CFA", "tld" : ".bj", "native" : "B nin", "region" : "Africa", "subregion" : "Western Africa", "timezones" : [ { "zoneName"
: "Africa/Porto-Novo", "gmtOffset" : 3600, "gmtOffsetName" : "UTC+01:00", "abbreviation" : "WAT", "tzName" : "West Africa Time" } ], "translations" : { "kr" : "    ", "br"
: "Benin", "pt" : "Benin", "nl" : "Benin", "hr" : "Benin", "fa" : "      ", "de" : "Benin", "es" : "Benin", "fr" : "B nin", "ja" : "      ", "it" : "Benin", "cn" : "    " },
"latitude" : "9.50000000", "longitude" : "2.25000000", "emoji" : "   ", "emojiU" : "U+1F1E7 U+1F1EF"
}
> db.countryInfo.find().sort({"latitude" : -1}).limit(1).pretty()
{
  "_id" : ObjectId("618a294d73166f12b5705a76"),
  "id" : 24,
  "name" : "Benin",
  "iso3" : "BEN",
  "iso2" : "BJ",
  "numeric_code" : "204",
  "phone_code" : "229",
  "capital" : "Porto-Novo",
  "currency" : "XOF",
  "currency_symbol" : "CFA",
  "tld" : ".bj",
  "native" : "B nin",
  "region" : "Africa",
  "subregion" : "Western Africa",
  "timezones" : [
    {
      "zoneName" : "Africa/Porto-Novo",
      "gmtOffset" : 3600,
      "gmtOffsetName" : "UTC+01:00",
      "abbreviation" : "WAT",
      "tzName" : "West Africa Time"
    }
  ],
  "translations" : {
    "kr" : "    ",
    "br" : "Benin",
    "pt" : "Benin",
  }
}
```