PROJECT 2: MONGODB

Made by: Marina Gómez Rey and

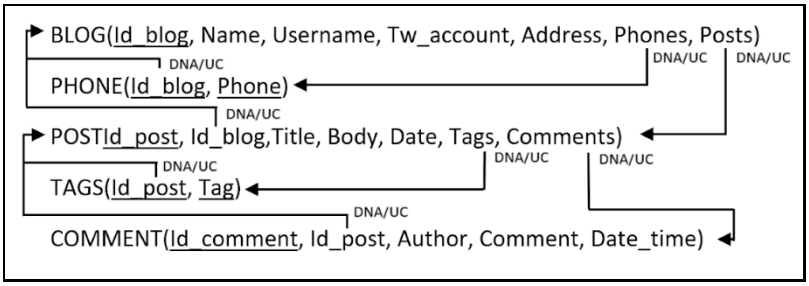
María Ángeles Magro Garrote (group 4)

INDEX

EXERCISE 1

1. Schema

Relational diagram



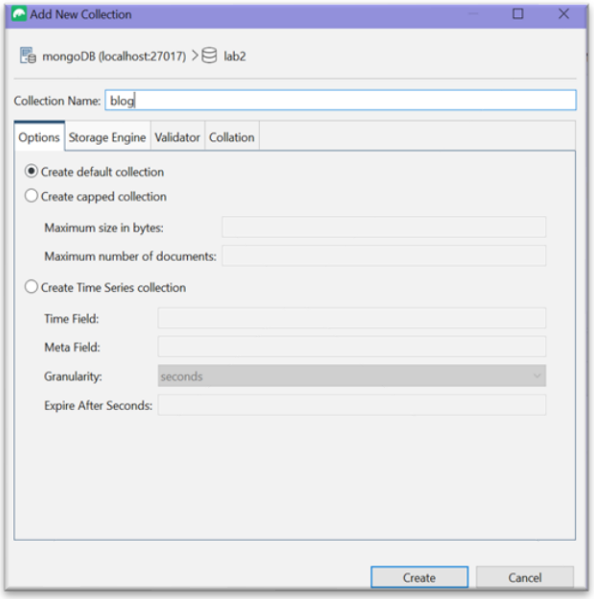
Semantic assumptions to the statement

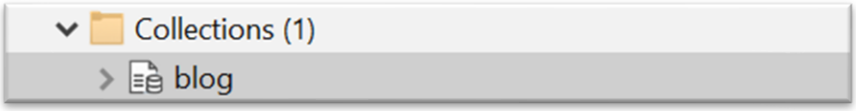
- Each blog, post and comment is identified by its own ID.

2. Inserts

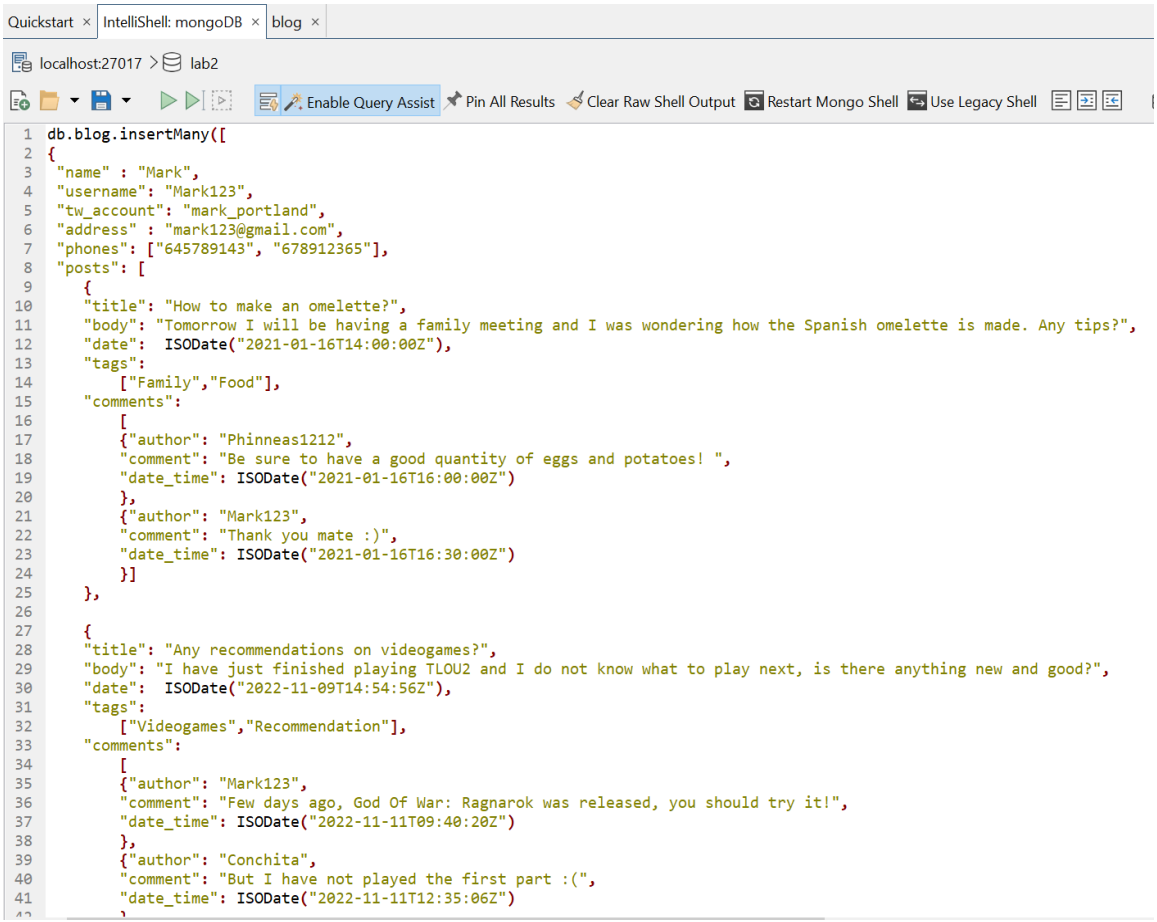
NOTE: All the codes are included in the last part of the document in case they need to be used.

As this is a non-relational model, everything needs to be collected in one collection because other ways would not let us see all the relations between the variables.





The data insertion was done with the **insertMany** function. Everything was programmed at the mongoShell.



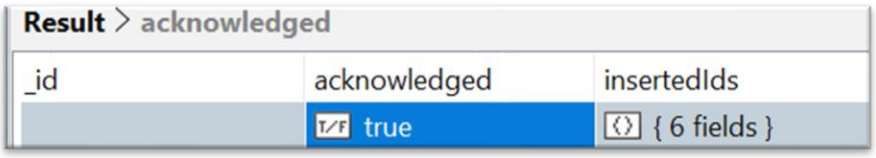

```
db.blog.insertMany([
{
 "name" : "Mark",
 "username": "Mark123",
 "tw_account": "mark_portland",
 "address" : "mark123@gmail.com",
 "phones": ["645789143", "678912365"],
 "posts": [
    {
    "title": "How to make an omelette?",
    "body": "Tomorrow I will be having a family meeting and I was wondering how the Spanish omelette is made. Any tips?",
    "date":  ISODate("2021-01-16T14:00:00Z"),
    "tags":
        ["Family","Food"],
    "comments":
        [
        {"author": "Phinneas1212",
        "comment": "Be sure to have a good quantity of eggs and potatoes! ",
        "date_time": ISODate("2021-01-16T16:00:00Z")
        },
        {"author": "Mark123",
        "comment": "Thank you mate :)",
        "date_time": ISODate("2021-01-16T16:30:00Z")
        }]
    },

    {
    "title": "Any recommendations on videogames?",
    "body": "I have just finished playing TLOU2 and I do not know what to play next, is there anything new and good?",
    "date":  ISODate("2022-11-09T14:54:56Z"),
    "tags":
        ["Videogames","Recommendation"],
    "comments":
        [
        {"author": "Mark123",
        "comment": "Few days ago, God Of War: Ragnarok was released, you should try it!",
        "date_time": ISODate("2022-11-11T09:40:20Z")
        },
        {"author": "Conchita",
        "comment": "But I have not played the first part :(",
        "date_time": ISODate("2022-11-11T12:35:06Z")
        },
        {"author": "Edward777",
        "comment": "Do not worry, the game includes a summary of the first part!",
        "date_time": ISODate("2022-11-11T16:19:03Z")
        }]}]
}
,
{
 "name" : "Phinneas",
 "username": "Phinneas1212",
 "tw_account": "phinneas_loves_ny",
 "address" : "phinneas1212@gmail.com",
 "phones": ["645789143"],
 "posts":
    [
    {
        "title": "How can I start a business online?",
        "body": "I am looking forward to selling my drawings online in order to save some money. Despite this, I have no clue where to start.
        "date":  ISODate("2019-05-20T06:00:00Z"),
        "tags":
            ["Branding","Online marketing"],
        "comments":
            [
            {"author": "Mark123",
            "comment": "In the past, I used Patreon although I do not know if it continues to be available.",
            "date_time": ISODate("2019-05-20T10:00:00Z")
            },
            {"author": "Edward777",
            "comment": "Create commissions and put them on your socials! It would amaze you how extraordinary they are.",
            "date_time": ISODate("2019-05-20T10:00:00Z")
            },
            {"author": "Juan3",
            "comment": "I have the same problem! It would be amazing if we could share some ideas.",
            "date_time": ISODate("2019-05-20T15:00:00Z")
            }]}]
},

{
 "name" : "Mery",
 "username": "Mery123",
 "tw_account": "mery_gg",
 "address" : "mery123gg@gmail.com",
```

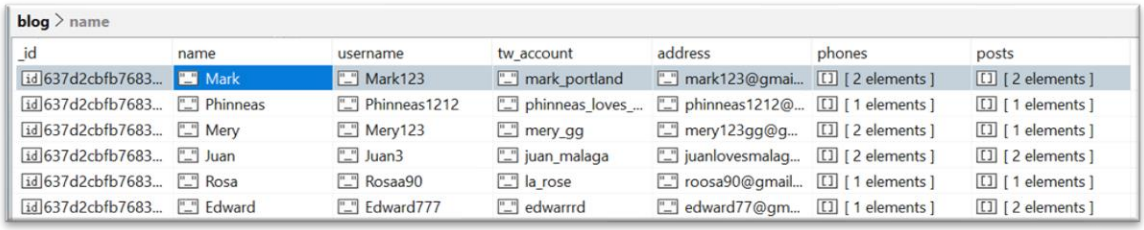
```
"phones":
    ["657576889", "678910965"],
"posts":

   [
   {
   "title": "Any help with mongoDB?",
   "body": "I am trying to install mongoDB but it does not work, does anyone have an idea?",
   "date":  ISODate("2019-06-10T18:03:02Z"),
   "tags":
       ["mongoDB","Data Science","database"],
   "comments":
       [
       {"author": "Juan3",
       "comment": "Take care of what folder you use to install it.",
       "date_time": ISODate("2019-08-16T20:10:50Z")
       },
       {"author": "Phinneas1212",
       "comment": "Make sure to follow this amazing steps:          https://medium.com/@LondonAppBrewery/how-to-download-install-mongodb-on-w
       "date_time": ISODate("2019-06-11T20:30:13Z")
       }]}]
},
{
"name" : "Juan",
"username": "Juan3",
"tw_account": "juan_malaga",
"address" : "juanlovesmalaga@gmail.com",
"phones":
    ["687527849","687573943"],
"posts": [
   {
   "title": "Planning to make a meeting with you!",
   "body": "This blog has been open for a couple of years now and I have met wonderful people. I was thinking of making a real-life plan. Mayl
   "date":  ISODate("2022-07-10T17:30:56Z"),
   "tags":
       ["Real life","Meeting"],
   "comments":
       [
       {
       "author": "Phinneas1212",
       "comment": "Amazing and extraordinary idea!!!",
```

```
       "author": "Phinneas1212",
       "comment": "Amazing and extraordinary idea!!!",
       "date_time": ISODate("2022-08-21T15:56:00Z")
       }]
},

   {
   "title": "Any plans for Christmas?",
   "body": "I am spending this Christmas on Madrid and I would like to know any interesting plan.",
   "date":  ISODate("2022-11-15T23:04:26Z"),
   "tags":
       ["meeting","Christmas","Holidays"],
   "comments":
       [
       {"author": "Mery123",
       "comment": "There is an amazing light show at the city center that will be extraordinary",
       "date_time": ISODate("2022-11-16T09:40:20Z")
       }]}]
},

{
"name" : "Rosa",
"username": "Rosaa90",
"tw_account": "la_rose",
"address" : "roosa90@gmail.com",
"phones":
    ["628364834"],
"posts":
   [{
   "title": "Do you prefer SQL or NoSQL",
   "body": "I am starting a project and I do not know how to choose among both!",
   "date":  ISODate("2019-05-22T14:00:56Z"),
   "tags":
       ["nosql","database","sql"],
   "comments":
       [
       {"author": "Edward777",
       "comment": "It will depend on what you are searching for. Make sure to know the differences among both (relations, robustness…)",
       "date_time": ISODate("2022-11-11T09:40:20Z")
       }]}]
},
```

```
        "date_time": ISODate("2022-11-11T09:40:20Z")
        }]}]
},

{
 "name" : "Edward",
 "username": "Edward777",
 "tw_account": "edwarrrd",
 "address" : "edward77@gmail.com",
 "phones":
    ["668397927"],

"posts":
   [{
   "title": "I wanna try data science",
   "body": "I love programming and I have heard about data science. Can anyone give some tips on how to start?",
   "date":  ISODate("2019-05-22T23:23:23Z"),
   "tags":
       ["Data Science"],
   "comments":
       [
       {
       "author": "Mark123",
       "comment": "Data science is amazing and allows you to do lots of things. You should try using R to begin, it has many useful commands
       "date_time": ISODate("2019-05-26T08:21:04Z")
       }]},
   {
   "title": "I want to promote my product on Tik Tok. Any useful tips?",
   "body": "Gm! I have decided to start an online business promoting my sweaters. I have noticed that all successful businesses have a popula
   "date":  ISODate("2021-08-14T15:44:02Z"),
   "tags":
       ["Video Marketing","Online Marketing","Social Media","Promotions"],
   "comments":
       [
       {"author": "Mark123",
       "comment": "Your initiative is extraordinary! Keep doing what defines your product, that would work. Amazing work.",
       "date_time": ISODate("2021-08-14T16:55:00Z")
       }]}]
}
])
```

The obtained result was:





Let's take the first insertion and analyze it:

```
{
 "name" : "Mark",
 "username": "Mark123",
 "tw_account": "mark_portland",
 "address" : "mark123@gmail.com",
 "phones": ["645789143", "678912365"],
 "posts": [
       {
    "title": "How to make an omelette?",
    "body": "Tomorrow I will be having a family meeting and I was wondering
how the Spanish omelette is made. Any tips?",
    "date":  ISODate("2021-01-16T14:00:00Z"),
```

```
"tags":
    ["Family","Food"],
"comments":
    [
    {"author": "Phinneas1212",
    "comment": "Be sure to have a good quantity of eggs and potatoes! ",
    "date_time": ISODate("2021-01-16T16:00:00Z")
    },
    {"author": "Mark123",
    "comment": "Thank you mate :)",
    "date_time": ISODate("2021-01-16T16:30:00Z")
    }]
  }
```

The organization of the documents is the following:

Firstly, the author is included with all its attributes, the name (which is the first name of the person), username (which id the username of the person in the blog), Twitter account (which is the Twitter account of the user), address (which is the email address) and phones (that can be one or more). Also, all the posts (publications of the user at the post) that person has created in the blog. This is done this way because other way would imply having to repeat the values of the user's data repeatedly, which is not optimum. The posts also have several attributes including its title (a phrase that represents what the post is about), body (the whole text of the post), date (the moment, with day, month, year and time in which the post was published), tags (that represent what the post is about with few words) and the comments (answers to the post other users have made). As well as the posts, the comments also include the author (username of the creator of the comments), the comment (text of the comment) and the date (same format as the date of the post).

As previously stated, we think this is the best way to organize everything in such that a manner that same data does not need to be repeated and that relations between classes are clearly seen.

3. Queries

3.1

Retrieve all blog posts that contain in the 'comment' the values 'amazing' and 'extraordinary' greater than equal to the publication date, the query result should return the fields from author (name, Twitter account), from the blog (title, publication date) and the result sort it ascending.

Query:

To find the comments that have both 'amazing' and 'extraordinary', the $all function is included, another option could have been the $and. The words are between /.* .*/I, this is important because this makes possible to search for these words without taking into consideration if they are in capital letters or not. Also, as the statement asks for 'greater than equal' at the date, we add $gte with the date of the post and the date of the comments.

On top of that, it asks to retrieve the data from the author and the title and date from the post, so we add this fields after the search conditions with a 1 to get all the values. To access values

that are inside the post collection, we add posts. and the requested attribute. Finally, to sort it, the sort function with the dates is applied.

```
db.blog2.find({'posts.comments.comment':
    {$all: [/.*amazing.*/i, /.*extraordinary.*/i]},
    $expr:{$gte:['$date', '$comments.date']}},
    {"name": 1, "tw_account": 1, "posts.title": 1,
        "posts.date": 1}).sort({"posts.date": 1});
```

Results:


```
{
    "_id" : ObjectId("637d017ad483f8211afea455"),
    "name" : "Phinneas",
    "tw_account" : "phinneas_loves_ny",
    "posts" : [
        {
            "title" : "How can I start a business online?",
            "date" : ISODate("2019-05-20T06:00:00.000+0000")
        }
    ]
}
{
    "_id" : ObjectId("637d017ad483f8211afea459"),
    "name" : "Edward",
    "tw_account" : "edwarrrd",
    "posts" : [
        {
            "title" : "I wanna try data science",
            "date" : ISODate("2019-05-22T23:23:23.000+0000")
        },
        {
            "title" : "I want to promote my product on Tik Tok. Any useful tips?",
            "date" : ISODate("2021-08-14T15:44:02.000+0000")
        }
    ]
}
{
    "_id" : ObjectId("637d017ad483f8211afea457"),
    "name" : "Juan",
    "tw_account" : "juan_malaga",
    "posts" : [
        {
            "title" : "Planning to make a meeting with you!",
            "date" : ISODate("2022-07-10T17:30:56.000+0000")
        },
        {
            "title" : "Any plans for Christmas?",
            "date" : ISODate("2022-11-15T23:04:26.000+0000")
        }
    ]
}
```


It can be seen that the results include the requested fields correctly.

3.2

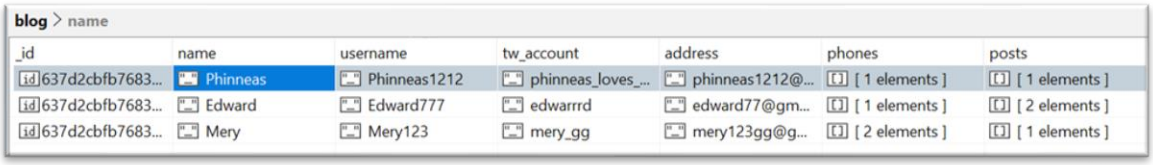
Retrieve at least all blog posts that contain in the values tags such 'Video Marketing', 'Online Marketing', 'Data Science', 'Branding' and the result sort it ascending.

Query:

To do this query, the $or function is included with the requested tags, that are as before created to not have problems with capital letters. Also, they are sorted again with the sort function.

```
db.blog.find(
{$or:[
{"posts.tags": /.*video marketing.*/i},
{"posts.tags": /.*online marketing.*/i},
{"posts.tags": /.*data science.*/i},
{"posts.tags": /.*branding.*/i}
]}).sort({"posts.date":1}).pretty()
```

Results:



blog > name

_id	name	username	tw_account	address	phones	posts
637d2cbfb7683...	Phinneas	Phinneas1212	phinneas_loves_...	phinneas1212@...	[1 elements]	[1 elements]
637d2cbfb7683...	Edward	Edward777	edwarrrd	edward77@gm...	[1 elements]	[2 elements]
637d2cbfb7683...	Mery	Mery123	mery_gg	mery123gg@g...	[2 elements]	[1 elements]

The tags from the results are:

"tags" : ["Branding", "Online marketing"],	"tags" : ["Data Science"],
"tags" : ["mongoDB", "Data Science", "database"],	"tags" : ["Video Marketing", "Online Marketing", "Social Media", "Promotions"],

Which are the requested tags.

3.3

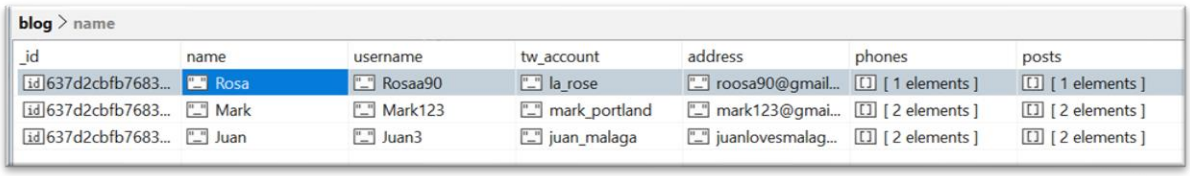
Retrieve all blog posts that not contain in the values tags such 'Video Marketing', 'Online Marketing', 'Data Science', 'Branding' and the result sort it ascending.

Query:

This query is exactly the same as the previous one but using instead of $or, $nor, which takes the values that do not satisfy what you have written.

```
db.blog.find(
{$nor:[
{"posts.tags": /.*video marketing.*/i},
{"posts.tags": /.*online marketing.*/i},
{"posts.tags": /.*data science.*/i},
{"posts.tags": /.*branding.*/i}
]}).sort({"posts.date":1}).pretty()
```

Results:



blog > name

_id	name	username	tw_account	address	phones	posts
637d2cbfb7683...	Rosa	Rosaa90	la_rose	roosa90@gmail...	[1 elements]	[1 elements]
637d2cbfb7683...	Mark	Mark123	mark_portland	mark123@gmai...	[2 elements]	[2 elements]
637d2cbfb7683...	Juan	Juan3	juan_malaga	juanlovesmalag...	[2 elements]	[2 elements]

Tags of the results:

"tags" : ["Family", "Food"	"tags" : ["Videogames", "Recommendation"	"tags" : ["Films", "Movies", "Recommendations"
"tags" : ["Real life", "Meeting"	"tags" : ["nosql", "database", "sql"	"tags" : ["meeting", "Christmas", "Holidays"

It can be seen that they do not include the ones we stated in the query.

3.4

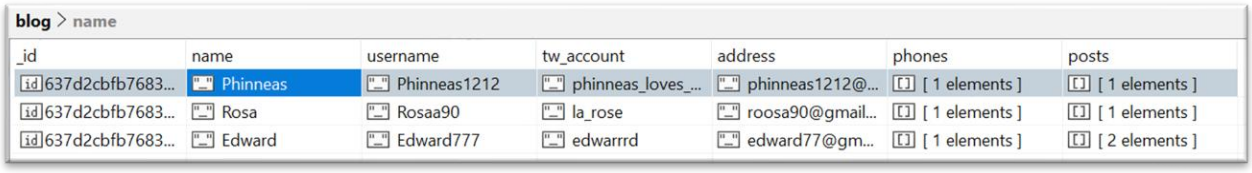
Retrieve title of the blog posts that have been written in May 2019.

Query:

To find the values of the posts that were written in 2019, we search for the values of the date that are between the 1st of January and the 31st of December. That is done with the grater than ($gt) and the less than ($lt) functions.

```
db.blog.find({
    "posts.date": {
        $gt: ISODate("2019-05-01"),
        $lt: ISODate("2019-05-31")
    }
}).pretty()
```

Results:



blog > name

_id	name	username	tw_account	address	phones	posts
637d2cbfb7683...	Phinneas	Phinneas1212	phinneas_loves_...	phinneas1212@...	[1 elements]	[1 elements]
637d2cbfb7683...	Rosa	Rosaa90	la_rose	roosa90@gmail...	[1 elements]	[1 elements]
637d2cbfb7683...	Edward	Edward777	edwarrrd	edward77@gm...	[1 elements]	[2 elements]

Dates of the results:

```
"date" : ISODate("2019-05-20T06:00:00.000+0000"),
"date" : ISODate("2019-05-22T23:23:23.000+0000"),
"date" : ISODate("2019-05-22T14:00:56.000+0000"),
```

3.5

Retrieve title of the blog posts that have been written by the user with username "Mery123" with any of the tags: "nosql", "database" or "mongoDB", and published in 2019.

Query:

The query first uses $and to find at the same time the author the tags and the corresponding date. Inside the tags, the $or must be included because it can be anyone of the asked ones. Inside the date, we must include the greater than and less than we added in the previous query.

```
db.post.find({
    $and: [{"author": "Mery123"},
    {$or: [{"tags":"nosql"}, {"tags":"database"}, {"tags":"mongoDB"}]},
    {"date":   {$gt: ISODate("2019-01-01"),
        $lt: ISODate("2019-12-31")
    }}]
})
```

Results:

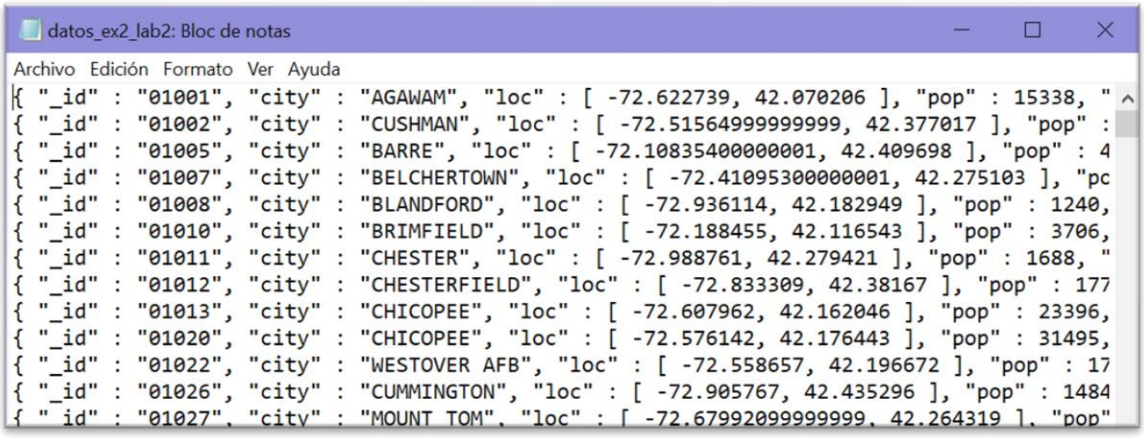
blog > name

_id	name	username	tw_account	address	phones	posts
637d2cbfb7683...	Mery	Mery123	mery_gg	mery123gg@g...	[2 elements]	[1 elements]

EXERCISE 2

Import data

To add the data to the collection (a new collection called city was created before), first the data was accessed from the link and then it was saved in a text document. That document was saved in a folder called data in our computer.



After that, we access our computer's shell and we enter the folder where our mongodb application is located at.

```
C:\>cd Program Files

C:\Program Files>cd MongoDB

C:\Program Files\MongoDB>cd Server

C:\Program Files\MongoDB\Server>cd 6.0

C:\Program Files\MongoDB\Server\6.0>cd bin
```

Once this was done, the data is transferred using the mongo port and the location of the document at our computer.

```
C:\Program Files\MongoDB\Server\6.0\bin>Mongoimport --host localhost --port 27017 -d lab2 -c city C:\data\db\datos_ex2_lab2.json
2022-11-16T20:22:30.429+0100    connected to: mongodb://localhost:27017/
2022-11-16T20:22:31.067+0100    29353 document(s) imported successfully. 0 document(s) failed to import.
```

Queries

1.1

Recover the minimum value of the field 'pop' for all cities.

Query:

To make this query we need the $min function aggregate has, as it gives as a result the minimum value of the specified attribute. Also, as it is for each city, we need to add $group and group it by city.

```
db.city.aggregate([{
    $group:
        {
          _id:"$city",
          minQuantity: { $min: "$pop" }
        }
 }])
```

Result:

```
{
    "_id" : "SPRINGVILLE",
    "minQuantity" : NumberInt(1391)
}
{
    "_id" : "ORIENTAL",
    "minQuantity" : NumberInt(1985)
}
{
    "_id" : "FLOODWOOD",
    "minQuantity" : NumberInt(1475)
}
{
    "_id" : "MATAGORDA",
    "minQuantity" : NumberInt(1455)
}
```

This is an extract from the results, we can see that it includes the name of the city and the least number of inhabitants it has.

1.2

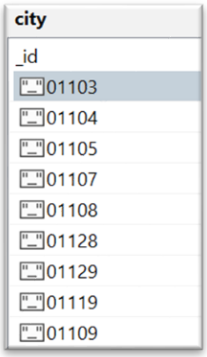
Retrieve all the zip codes in the city "Springfield". How many?

Query:

We search for the values of Springfield (in capital letters as everything in this document is done that way) and we add the _id to retrieve only the id of the cities.

```
db.city.find({"city": "SPRINGFIELD"}, {_id:1})
```

Results:



city
_id
01103
01104
01105
01107
01108
01128
01129
01119
01109

Then, in order to retrieve the specific number we add the count function.

Query:

```
db.city.find({"city": "SPRINGFIELD"}, {_id:1}).count()
```

Result:

41

1.3

Retrieve the number of cities in state "GA" with less than equal 1500 population.

Query:

To do this query, we search for the values of population less than ($lt function) in the GA state, and we add the count function again.

```
db.city.find({$and:[
    {"state":"GA"},
    {"pop":{$lt:1500}}]}).count()
```

Result:

123

1.4

Update all cities from Massachusetts (state: ”MA”), a new field { area : “27 340 km”}.

Query:

Now, instead of using find, we need to use update. We establish what we want to update, which in this case is the state MA and with the $set function the area is changed. Also, multi is true as there is more than one value to update.

```
db.city.update(
 { state : "MA" },
 { $set: { "area": "27340 km" }},
 {multi: true}
)
```

Result:

Result > acknowledged

_id	acknowledged	insertedId	matchedCount	modifiedCount	upsertedCount
	true	null	474.0	474.0	0.0

1.5

Recover all cities that are located between -81.000001 and 32.915592.

Query:

As the ones where we searched for the dates in exercise 1, here we use the greater than ($gt) and less than ($lt) to get the cities located between those values.

```
db.city.find({$and:[
    {"loc.0":{$gt:-81.000001}},
    {"loc.1":{$lt:32.915592}}]
    })
```

Results:

_id	city	loc	pop	state
29414	CHARLESTON	[2 elements]	19404	SC
29418	CHARLESTON	[2 elements]	39784	SC
29426	JERICHO	[2 elements]	1723	SC
29403	CHARLESTON	[2 elements]	24620	SC
29406	NORTH CHARLE...	[2 elements]	27726	SC
29404	CHARLESTON	[2 elements]	5420	SC
29405	CHARLESTON	[2 elements]	30621	SC
29401	CHARLESTON	[2 elements]	12475	SC
29407	CHARLESTON	[2 elements]	38597	SC
29412	CHARLESTON	[2 elements]	29969	SC

ç

1.6

Update the populations to 50000 for those cities that have a population between 45000 and 50000.

Query:

The same concept applied before has to be applied here but with the update function. Then we use $set to change the population to 75000.

```
db.city.updateMany(
    {$and:[{"pop":{$gt:45000}},
           {"pop":{$lt:50000}}
    ]
    },
    {$set:{"pop":75000}
    })
```

Result:

Result > acknowledged

_id	acknowledged	insertedId	matchedCount	modifiedCount	upsertedCount
	true	null	227.0	227.0	0.0

CODES

EXERCISE 1

INSERTS

```
db.blog.insertMany([
{
 "name" : "Mark",
 "username": "Mark123",
 "tw_account": "mark_portland",
 "address" : "mark123@gmail.com",
 "phones": ["645789143", "678912365"],
 "posts": [
      {
    "title": "How to make an omelette?",
    "body": "Tomorrow I will be having a family meeting and I was wondering
how the Spanish omelette is made. Any tips?",
    "date":  ISODate("2021-01-16T14:00:00Z"),
    "tags":
        ["Family","Food"],
    "comments":
        [
        {"author": "Phinneas1212",
        "comment": "Be sure to have a good quantity of eggs and potatoes! ",
        "date_time": ISODate("2021-01-16T16:00:00Z")
        },
        {"author": "Mark123",
        "comment": "Thank you mate :)",
        "date_time": ISODate("2021-01-16T16:30:00Z")
        }]
       },

    {
    "title": "Any recommendations on videogames?",
    "body": "I have just finished playing TLOU2 and I do not know what to
play next, is there anything new and good?",
    "date":  ISODate("2022-11-09T14:54:56Z"),
    "tags":
        ["Videogames","Recommendation"],
    "comments":
        [
        {"author": "Mark123",
        "comment": "Few days ago, God Of War: Ragnarok was released, you
should try it!",
        "date_time": ISODate("2022-11-11T09:40:20Z")
        },
        {"author": "Conchita",
        "comment": "But I have not played the first part :(",
        "date_time": ISODate("2022-11-11T12:35:06Z")
        },
        {"author": "Edward777",
        "comment": "Do not worry, the game includes a summary of the first
part!",
        "date_time": ISODate("2022-11-11T16:19:03Z")
        }]}]
}
,
{
```

```
 "name" : "Phinneas",
 "username": "Phinneas1212",
 "tw_account": "phinneas_loves_ny",
 "address" : "phinneas1212@gmail.com",
 "phones": ["645789143"],
 "posts":
    [
    {
        "title": "How can I start a business online?",
        "body": "I am looking forward to selling my drawings online in order
to save some money. Despite this, I have no clue where to start. ",
        "date":  ISODate("2019-05-20T06:00:00Z"),
        "tags":
            ["Branding","Online marketing"],
        "comments":
            [
            {"author": "Mark123",
            "comment": "In the past, I used Patreon although I do not know if
it continues to be available.",
            "date_time": ISODate("2019-05-20T10:00:00Z")
            },
            {"author": "Edward777",
            "comment": "Create commissions and put them on your socials! It
would amaze you how extraordinary they are.",
            "date_time": ISODate("2019-05-20T10:00:00Z")
            },
            {"author": "Juan3",
            "comment": "I have the same problem! It would be amazing if we
could share some ideas.",
            "date_time": ISODate("2019-05-20T15:00:00Z")
            }]}]
},

{
 "name" : "Mery",
 "username": "Mery123",
 "tw_account": "mery_gg",
 "address" : "mery123gg@gmail.com",
 "phones":
     ["657576889", "678910965"],
 "posts":

    [
    {
    "title": "Any help with mongoDB?",
    "body": "I am trying to install mongoDB but it does not work, does anyone
have an idea?",
    "date":  ISODate("2019-06-10T18:03:02Z"),
    "tags":
        ["mongoDB","Data Science","database"],
    "comments":
        [
        {"author": "Juan3",
        "comment": "Take care of what folder you use to install it.",
        "date_time": ISODate("2019-08-16T20:10:50Z")
        },
        {"author": "Phinneas1212",
```

```
        "comment": "Make sure to follow this amazing steps:
https://medium.com/@LondonAppBrewery/how-to-download-install-mongodb-on-
windows-4ee4b3493514",
        "date_time": ISODate("2019-06-11T20:30:13Z")
        }]}]
},
{
 "name" : "Juan",
 "username": "Juan3",
 "tw_account": "juan_malaga",
 "address" : "juanlovesmalaga@gmail.com",
 "phones":
     ["687527849","687573943"],
 "posts": [
    {
    "title": "Planning to make a meeting with you!",
    "body": "This blog has been open for a couple of years now and I have met
wonderful people. I was thinking of making a real-life plan. Maybe we can
discuss the place (nearest to everyone) and a date. Hoping to hear from
you!",
    "date":  ISODate("2022-07-10T17:30:56Z"),
    "tags":
        ["Real life","Meeting"],
    "comments":
        [
        {
        "author": "Phinneas1212",
        "comment": "Amazing and extraordinary idea!!!",
        "date_time": ISODate("2022-08-21T15:56:00Z")
        }]
},

    {
    "title": "Any plans for Christmas?",
    "body": "I am spending this Christmas on Madrid and I would like to know
any interesting plan.",
    "date":  ISODate("2022-11-15T23:04:26Z"),
    "tags":
        ["meeting","Christmas","Holidays"],
    "comments":
        [
        {"author": "Mery123",
        "comment": "There is an amazing light show at the city center that
will be extraordinary",
        "date_time": ISODate("2022-11-16T09:40:20Z")
        }]}]
},

{
 "name" : "Rosa",
 "username": "Rosaa90",
 "tw_account": "la_rose",
 "address" : "roosa90@gmail.com",
 "phones":
     ["628364834"],
"posts":
    [{
    "title": "Do you prefer SQL or NoSQL",
```

```
    "body": "I am starting a project and I do not know how to choose among
both!",
    "date":  ISODate("2019-05-22T14:00:56Z"),
    "tags":
        ["nosql","database","sql"],
    "comments":
        [
        {"author": "Edward777",
        "comment": "It will depend on what you are searching for. Make sure
to know the differences among both (relations, robustness…)",
        "date_time": ISODate("2022-11-11T09:40:20Z")
        }]}]
},

{
 "name" : "Edward",
 "username": "Edward777",
 "tw_account": "edwarrrd",
 "address" : "edward77@gmail.com",
 "phones":
     ["668397927"],

"posts":
    [{
    "title": "I wanna try data science",
    "body": "I love programming and I have heard about data science. Can
anyone give some tips on how to start?",
    "date":  ISODate("2019-05-22T23:23:23Z"),
    "tags":
        ["Data Science"],
    "comments":
        [
        {
        "author": "Mark123",
        "comment": "Data science is amazing and allows you to do lots of
things. You should try using R to begin, it has many useful commands for this
field.",
        "date_time": ISODate("2019-05-26T08:21:04Z")
        }]},
    {
    "title": "I want to promote my product on Tik Tok. Any useful tips?",
    "body": "Gm! I have decided to start an online business promoting my
sweaters. I have noticed that all successful businesses have a popular Tik
Tok account. Any advice on how to build a big account???",
    "date":  ISODate("2021-08-14T15:44:02Z"),
    "tags":
        ["Video Marketing","Online Marketing","Social Media","Promotions"],
    "comments":
        [
        {"author": "Mark123",
        "comment": "Your initiative is extraordinary! Keep doing what defines
your product, that would work. Amazing work.",
        "date_time": ISODate("2021-08-14T16:55:00Z")
        }]}]
}
])
```

QUERIES

1.

```
db.blog2.find({'posts.comments.comment':
    {$all: [/.*amazing.*/i, /.*extraordinary.*/i]},
    $expr:{$gte:['$date', '$comments.date']}},
    {"name": 1, "tw_account": 1, "posts.title": 1,
        "posts.date": 1}).sort({"posts.date": 1});
```

2.

```
db.blog.find( {$or: [ {"posts.tags": /.*video marketing.*/i},
              {"posts.tags": /.*online marketing.*/i},
              {"posts.tags": /.*data science.*/i},
{"posts.tags":/.*branding.*/i}]}).sort({"posts.date":1}).pretty()
```

3.

```
db.blog.find(
{$nor:[
{"posts.tags": /.*video marketing.*/i},
{"posts.tags": /.*online marketing.*/i},
{"posts.tags": /.*data science.*/i},
{"posts.tags": /.*branding.*/i}
]}).sort({"posts.date":1}).pretty()
```

4.

```
db.blog.find({
    "posts.date": {
        $gt: ISODate("2019-05-01"),
        $lt: ISODate("2019-05-31")
    }
}).pretty()
```

5.

```
db.blog.find({
    $and: [{"username": "Mery123"},
    {$or: [{"posts.tags":"nosql"}, {"posts.tags":"database"},
{"posts.tags":"mongoDB"}]},
    {"posts.date":   {$gt: ISODate("2019-01-01"),
        $lt: ISODate("2019-12-31")
    }}]
})
```

EXERCISE 2

QUERIES

1.

```
db.city.aggregate([{
    $group:
         {
           _id:"$city",
           minQuantity: { $min: "$pop" }
         }
 }])
```

2.

```
db.city.find({"city": "SPRINGFIELD"}, {_id:1})
db.city.find({"city": "SPRINGFIELD"}, {_id:1}).count()
```

3.

```
db.city.find({$and:[
    {"state":"GA"},
    {"pop":{$lt:1500}}]}).count()
```

4.

```
db.city.update(
 { state : "MA" },
 { $set: { "area": "27340 km" }},
 {multi: true}
)
```

5.

```
db.city.find({$and:[
    {"loc.0":{$gt:-81.000001}},
    {"loc.1":{$lt:32.915592}}]
    })
```

6.

```
db.city.updateMany(
    {$and:[{"pop":{$gt:45000}},
           {"pop":{$lt:50000}}
    ]
    },
    {$set:{"pop":75000}
    })
```