

Question 1)

الف) ابتدا پایگاه داده [MongoDB](#) را دانلود کرده، آن را نصب و اجرا کنید. در ادامه موارد خواسته شده را انجام دهید.
۱- تفاوت پایگاه داده‌ی رابطه‌ای و پایگاه داده‌ی NoSQL در چیست؟ پایگاه داده‌ی MongoDB چه نوع NoSQL است؟

Firstly, Relational databases are table-based. NoSQL databases can be document based, graph databases, key-value pairs, or wide-column stores. Relational databases were built during a time that data was mostly structured and clearly defined by their relationships. Today, we know that data today is much more complex. NoSQL databases are designed to handle the more complex, unstructured data, (such as texts, social media posts, photos, videos, email) which increasingly make up much of the data that exists today. Secondly, Relational databases are vertically scalable but typically expensive. Since they require a single server to host the entire database, in order to scale, you need to buy a bigger, more expensive server. Scaling a NoSQL database is much cheaper, compared to a relational database, because you can add capacity by scaling horizontally over cheap, commodity servers. Thirdly, NoSQL databases tend to be more a part of the open-source community. Relational databases are typically closed source with licensing fees baked into the use of their software. Finally, NoSQL databases feature dynamic schema, and allow you to use what's known as "unstructured data." This means you can build your application without having to first define the schema. In a relational database, you are required to define your schema before adding data to the database. Not needing a predefined schema makes NoSQL databases much easier to update as data and requirements change. Changing the schema structure in a relational database can be extremely expensive, time-consuming, and often involve downtime or service interruptions.

MongoDB is a document database which pairs each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

Question 2)

۲- با استفاده از **Mongo Shell** یک دیتابیس جدید با نام **School** بسازید. لیست دیتابیس‌های موجود در سیستم را نمایش دهید. سپس دو کالکشن با نام‌های **Student** و **Courses** ساخته و لیست کالکشن‌ها را نیز چاپ نمایید. تغییرات فوق را در **MongoDBCompass** نیز بررسی کنید.

Creating database “School”:

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> use School
switched to db School
```

The newly created database (School) is not present in the list. To display database, we need to insert at least one document into it.

Showing a list of the system databases:

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> db
School
```

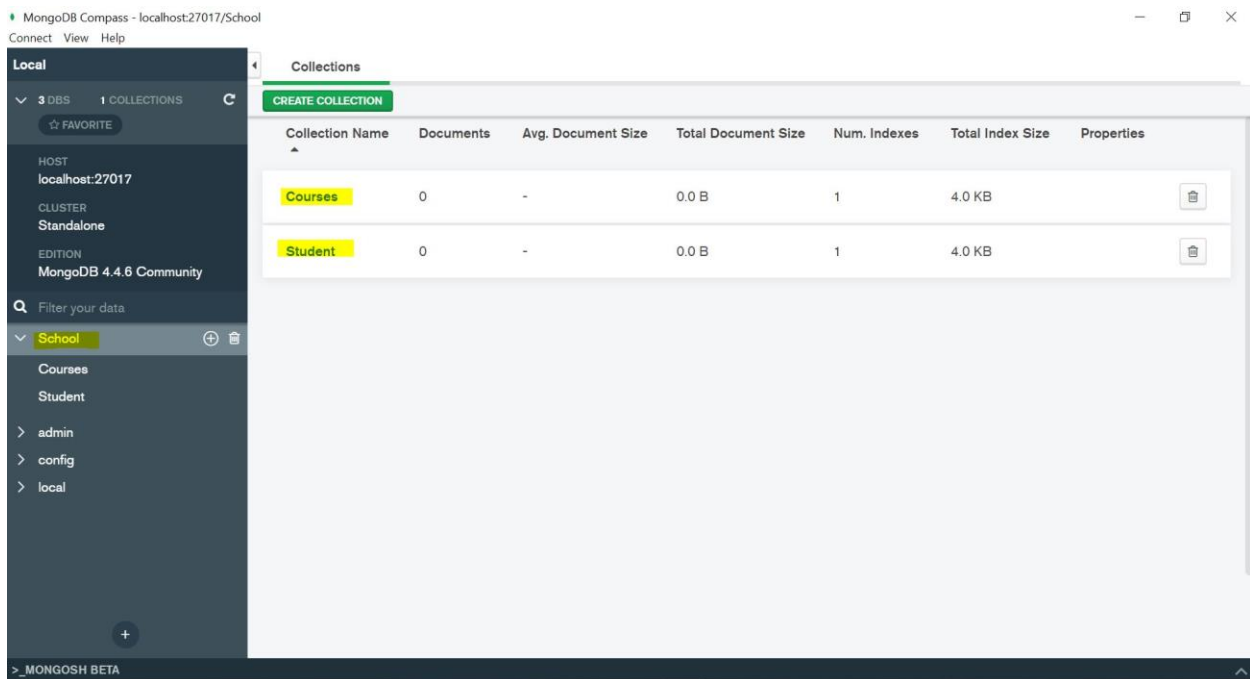
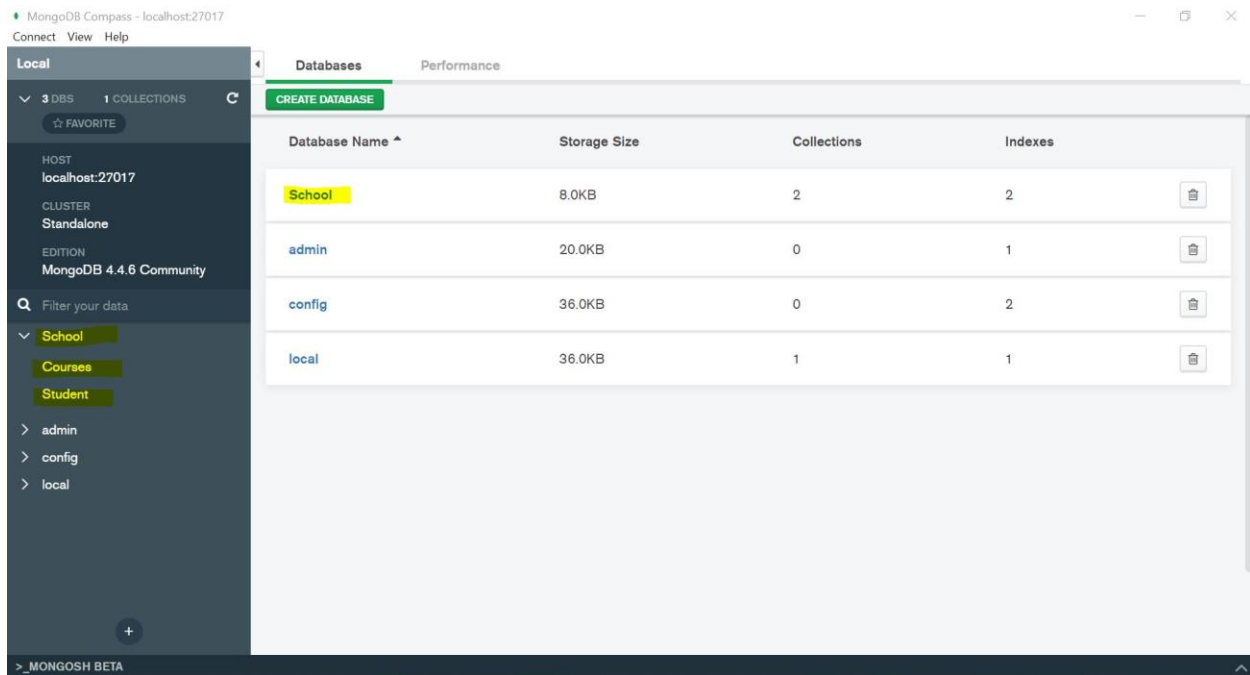
Creating two collections “Student” and “Courses” in the “School” database:

```
> db.createCollection('Student')
{ "ok" : 1 }
> db.createCollection('Courses')
{ "ok" : 1 }
```

Showing the collections:

```
> show dbs
School  0.000GB
admin    0.000GB
config  0.000GB
local    0.000GB
> show collections
Courses
Student
```

Showing how they look like in MongoDB Compass:



Question 3)

۳- با استفاده از MongoShell دو دانشجو با مشخصات دلخواه برای فیلدهای name, last_name, age, degree و نیز address وارد کنید. فیلد degree خود آرایه‌ای باشد که هر المان از آن دارای فیلدهای major, minor (اختیاری) و gpa باشد. فیلد address نیز دارای فیلدهای city, country, street است.

Inserting data into collection "Student":

```
> db.Student.insert({
... "name": "Sherlock",
... "last_name": "Holmes",
... "age": 25,
... "degree": {
... "university": "Oxford",
... "major": "Criminology",
... "minor": "Psychology",
... "gpa": 4
... },
... "address": {
... "country": "England",
... "city": "London",
... "street": "Baker Street"
... }})
WriteResult({ "nInserted" : 1 })
> db.Student.insert({
... "name": "Tina",
... "last_name": "Gholami",
... "age": 23,
... "degree": {
... "university": "AUT",
... "major": "Electrical Engineering",
... "minor": "Computer Engineering",
... "gpa": 3.8
... },
... "address": {
... "country": "Iran",
... "city": "Tehran",
... "street": "College Avenue"
... }})
WriteResult({ "nInserted" : 1 })
>
```

Showing the data:

```
> db.Student.find().pretty()
{
  "_id" : ObjectId("60d596a9264ea4a875fcd149"),
  "name" : "Sherlock",
  "last_name" : "Holmes",
  "age" : 25,
  "degree" : {
    "university" : "Oxford",
    "major" : "Criminology",
    "minor" : "Psychology",
    "gpa" : 4
  },
  "address" : {
    "country" : "England",
    "city" : "London",
    "street" : "Baker Street"
  }
}
{
  "_id" : ObjectId("60d5971d264ea4a875fcd14a"),
  "name" : "Tina",
  "last_name" : "Gholami",
  "age" : 23,
  "degree" : {
    "university" : "AUT",
    "major" : "Electrical Engineering",
    "minor" : "Computer Engineering",
    "gpa" : 3.8
  },
  "address" : {
    "country" : "Iran",
    "city" : "Tehran",
    "street" : "College Avenue"
  }
}
>
```

Question 4)

۴- کالکشن Courses را حذف کنید.

If we wish to only remove the data (documents) in the collection “Course”, we can use:

```
> db.Courses.remove({})
WriteResult({ "nRemoved" : 0 })
> show collections
Courses
Student
```

But if we wish to drop the collection “Courses” entirely, we can use:

```
> db.Courses.drop()
true
> show collections
Student
>
```

Question 5)

۵- داکيومنت‌های موجود در کالکشن Students را به صورت pretty نمایش دهید. با استفاده از دستور update سن یکی از دانش‌آموزان را ۳ سال افزایش دهید. سپس فیلد سن را از دانش‌آموز دیگر حذف کنید.

Showing the documents in the collection “Students”:

```

> db.Student.find().pretty()
{
  "_id" : ObjectId("60d596a9264ea4a875fcd149"),
  "name" : "Sherlock",
  "last_name" : "Holmes",
  "age" : 25,
  "degree" : {
    "university" : "Oxford",
    "major" : "Criminology",
    "minor" : "Psychology",
    "gpa" : 4
  },
  "address" : {
    "country" : "England",
    "city" : "London",
    "street" : "Baker Street"
  }
}
{
  "_id" : ObjectId("60d5971d264ea4a875fcd14a"),
  "name" : "Tina",
  "last_name" : "Gholami",
  "age" : 23,
  "degree" : {
    "university" : "AUT",
    "major" : "Electrical Engineering",
    "minor" : "Computer Engineering",
    "gpa" : 3.8
  },
  "address" : {
    "country" : "Iran",
    "city" : "Tehran",
    "street" : "College Avenue"
  }
}
>

```

Increasing one of the students' age (student "Sherlock") by 3 years:

```

> db.Student.update({"name":"Sherlock"}, {$inc:{"age":3}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

As can be seen bellow, the age of student "Sherlock" increased from 25 to 28:


```
> db.Student.find().pretty()
{
  "_id" : ObjectId("60d596a9264ea4a875fcd149"),
  "name" : "Sherlock",
  "last_name" : "Holmes",
  "age" : 28,
  "degree" : {
    "university" : "Oxford",
    "major" : "Criminology",
    "minor" : "Psychology",
    "gpa" : 4
  },
  "address" : {
    "country" : "England",
    "city" : "London",
    "street" : "Baker Street"
  }
}
```

Deleting the age of the other student (student “Tina”):

```
> db.Student.update({"name":"Tina"}, {$unset:{"age":1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

As can be seen bellow, the field “age” is deleted for student “Tina”:

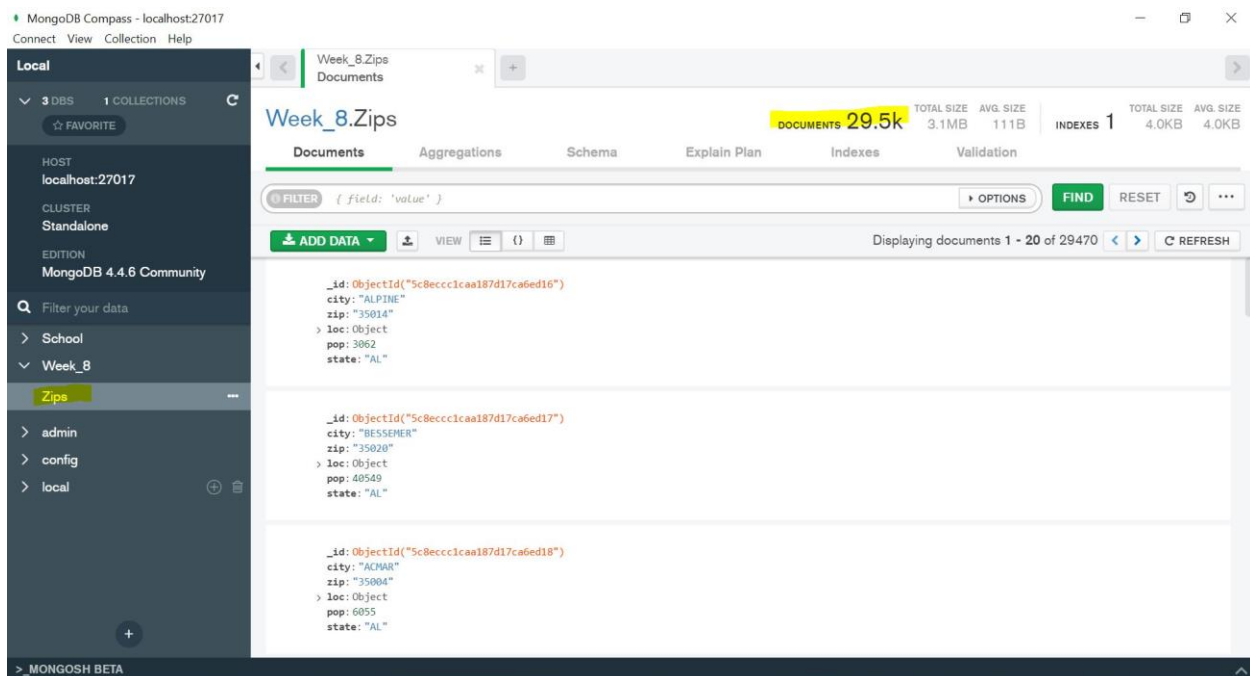
```
{
  "_id" : ObjectId("60d5971d264ea4a875fcd14a"),
  "name" : "Tina",
  "last_name" : "Gholami",
  "degree" : {
    "university" : "AUT",
    "major" : "Electrical Engineering",
    "minor" : "Computer Engineering",
    "gpa" : 3.8
  },
  "address" : {
    "country" : "Iran",
    "city" : "Tehran",
    "street" : "College Avenue"
  }
}
```


Question 6)

ب) فایل zip.json که در اختیار شما قرار داده شده است را دانلود کنید. این فایل حاوی اطلاعات تعدادی شهر در ایالات مختلف است. می‌خواهیم داده‌های این فایل را Import کنیم. برای این کار در MongoDB Compass یک کالکشن با نام zips بسازید. سپس از منوی Collection گزینه‌ی Import Data را انتخاب کرده و مراحل را برای import ادامه دهید. حال به سوالات زیر در محیط MongoShell پاسخ دهید.

۶- با استفاده از find داکيومنت‌هایی را بیابید که مقدار city آن‌ها برابر SHELTON باشد.

First, the collection “Zips” is created and the json file is uploaded:



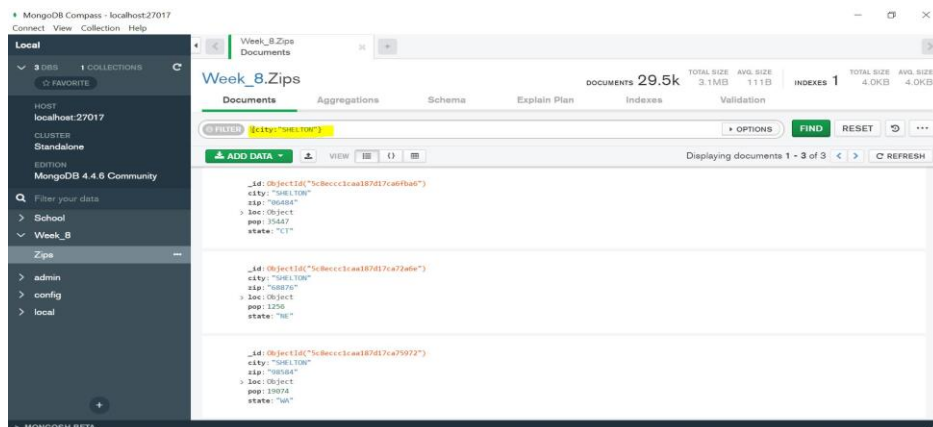
Next, cities of “SHELTON” are found in MongoDB Shell:

```

> db.Zips.find({"city":"SHELTON"}, {}).pretty()
{
  "_id" : ObjectId("5c8eccc1caa187d17ca6fba6"),
  "city" : "SHELTON",
  "zip" : "06484",
  "loc" : {
    "y" : 41.304689,
    "x" : 73.129439
  },
  "pop" : 35447,
  "state" : "CT"
},
{
  "_id" : ObjectId("5c8eccc1caa187d17ca72a6e"),
  "city" : "SHELTON",
  "zip" : "68876",
  "loc" : {
    "y" : 40.771703,
    "x" : 98.743453
  },
  "pop" : 1256,
  "state" : "NE"
},
{
  "_id" : ObjectId("5c8eccc1caa187d17ca75972"),
  "city" : "SHELTON",
  "zip" : "98584",
  "loc" : {
    "y" : 47.20863,
    "x" : 123.072862
  },
  "pop" : 19074,
  "state" : "WA"
}
}
>

```

And also in MongoDB Compass:



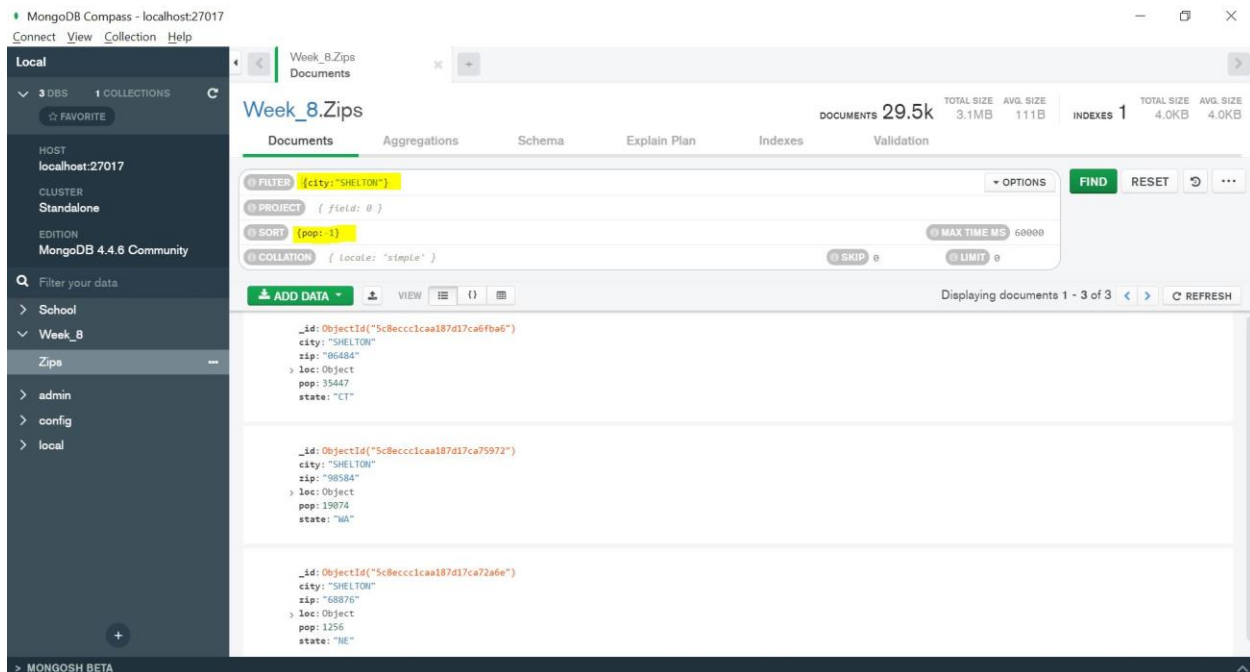
Question 7)

۷- با استفاده از دستور sort خروجی سوال ۶ را بر اساس جمعیت آن شهر به صورت نزولی مرتب کنید.

Sorting the cities by their poulation descendingly in MongoDB Shell:

```
> db.Zips.find({"city":"SHELTON"}, {}).sort({"pop":-1}).pretty()
{
  "_id" : ObjectId("5c8eccc1caa187d17ca6fba6"),
  "city" : "SHELTON",
  "zip" : "06484",
  "loc" : {
    "y" : 41.304689,
    "x" : 73.129439
  },
  "pop" : 35447,
  "state" : "CT"
}
{
  "_id" : ObjectId("5c8eccc1caa187d17ca75972"),
  "city" : "SHELTON",
  "zip" : "98584",
  "loc" : {
    "y" : 47.20863,
    "x" : 123.072862
  },
  "pop" : 19074,
  "state" : "WA"
}
{
  "_id" : ObjectId("5c8eccc1caa187d17ca72a6e"),
  "city" : "SHELTON",
  "zip" : "68876",
  "loc" : {
    "y" : 40.771703,
    "x" : 98.743453
  },
  "pop" : 1256,
  "state" : "NE"
}
>
```

And in MongoDB Shell:



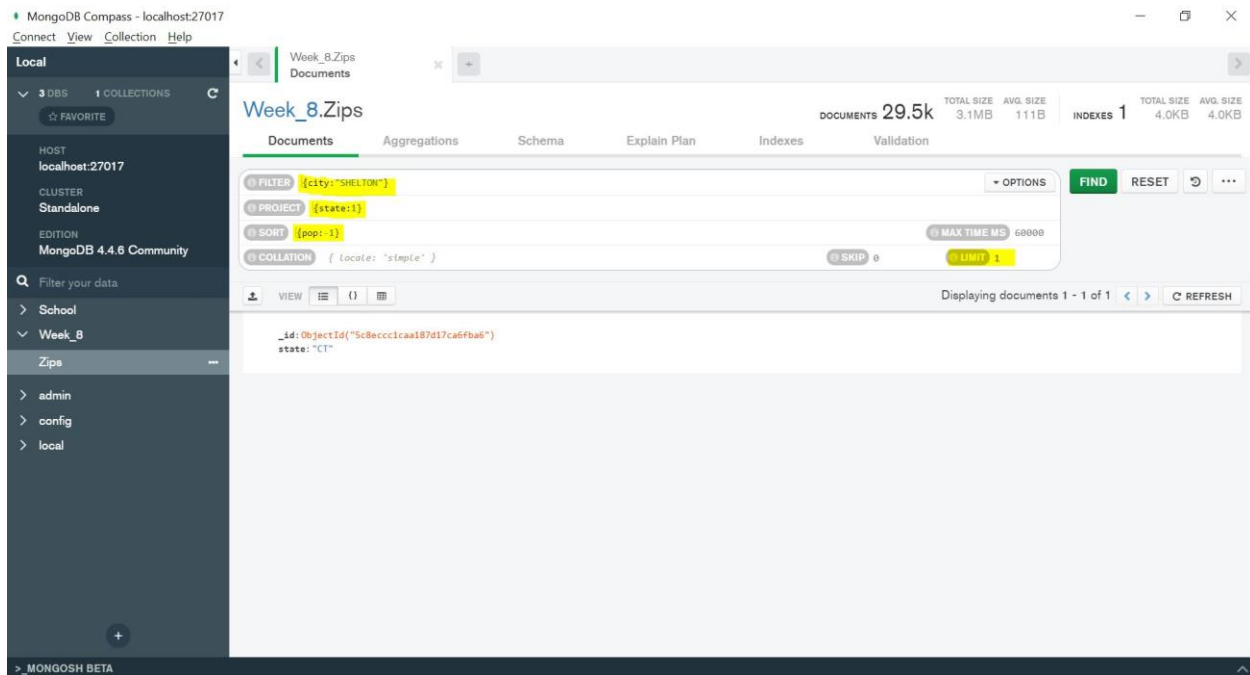
Question 8)

۸- با استفاده از دستور `top` و عملیات `projection` بر روی خروجی سوال ۷ نام ایالتی که پرجمعیت‌ترین شهر با نام Shelton در آن واقع است را بیابید. این خروجی را در MongoDBCompass نیز بگیرید.

We have to choose the first result appearing in question 7 as the most populated city using the “limit” method:

```
> db.Zips.find({"city": "SHELTON"}, {"state": 1}).sort({"pop": -1}).limit(1).pretty()
{ "_id" : ObjectId("5c8eccc1caa187d17ca6fba6"), "state" : "CT" }
>
```

Here is the result in MongoDB Compass:



Question 9)

۹- نام ۱۰ شهر پرجمعیت موجود در این کالکشن را بیابید.

The top 10 populated cities in the document are:

```
> db.Zips.find({}, {"city":1}).sort({"pop":-1}).limit(10).pretty()
{ "_id" : ObjectId("5c8eccc1caa187d17ca7044d"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca7307f"), "city" : "BROOKLYN" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca72fa0"), "city" : "NEW YORK" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca72fa5"), "city" : "NEW YORK" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca6f39d"), "city" : "BELL GARDENS" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca70447"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca6f366"), "city" : "LOS ANGELES" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca70466"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca70455"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca6f3db"), "city" : "NORWALK" }
```


But since there are only 7 distinct cities, I decided to limit the cities to 20 to be able to find the top distinct ones:

1. Chicago
2. Brooklyn
3. New York
4. Bell Gardens
5. Los Angeles
6. Norwalk
7. Jackson Heights
8. Arleta
9. South Gate
10. RidgeWood

```
> db.Zips.find({}, {"city":1}).sort({"pop":-1}).limit(20).pretty()
{ "_id" : ObjectId("5c8eccc1caa187d17ca7044d"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca7307f"), "city" : "BROOKLYN" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca72fa0"), "city" : "NEW YORK" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca72fa5"), "city" : "NEW YORK" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca6f39d"), "city" : "BELL GARDENS" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca70447"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca6f366"), "city" : "LOS ANGELES" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca70466"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca70455"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca6f3db"), "city" : "NORWALK" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca7044a"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca70454"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca7043e"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca70448"), "city" : "CHICAGO" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca730a1"), "city" : "JACKSON HEIGHTS" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca6f423"), "city" : "ARLETA" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca73072"), "city" : "BROOKLYN" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca6f3b7"), "city" : "SOUTH GATE" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca730a4"), "city" : "RIDGEWOOD" }
{ "_id" : ObjectId("5c8eccc1caa187d17ca72fd3"), "city" : "BRONX" }
>
```

Question 10)

۱۰- داکيومنت‌هایی را بیابید که جمعیت آن‌ها یا کمتر از ۱۰۰ نفر است یا بیش از ۱۰۰۰۰۰ نفر.

Finding the documents having a population of less than 100 or great

```
> db.Zips.find({$or:[{"pop": {$lt: 100}}, {"pop": {$gt: 100000}}]}, {}).pretty()
{
  "_id" : ObjectId("5c8eccc1caa187d17ca6eea3"),
  "city" : "ALLEN",
  "zip" : "36419",
  "loc" : {
    "y" : 31.624266,
    "x" : 87.66746
  },
  "pop" : 0,
  "state" : "AL"
}
{
  "_id" : ObjectId("5c8eccc1caa187d17ca6eebf"),
  "city" : "CARLTON",
  "zip" : "36515",
  "loc" : {
    "y" : 31.322449,
    "x" : 87.837793
  },
  "pop" : 30,
  "state" : "AL"
}
{
  "_id" : ObjectId("5c8eccc1caa187d17ca6ef1c"),
  "city" : "MORVIN",
  "zip" : "36762",
  "loc" : {
    "y" : 31.967305,
    "x" : 87.972897
  },
  "pop" : 24,
  "state" : "AL"
}
{
  "_id" : ObjectId("5c8eccc1caa187d17ca6ef65"),
  "city" : "CHEVAK",
```

Since there are 720 documents with city population less than 100 or greater than 100000, only a fraction of it is show above.

```
> db.Zips.find({$or:[{"pop": {$lt: 100}}, {"pop": {$gt: 100000}}]}, {}).count()
720
>
```


Question 11)

۱۱- تعداد شهرهایی را بیابید که در ایالت نیورک (NY) واقع شده‌اند.

There are 1596 cities in the state “NY”:

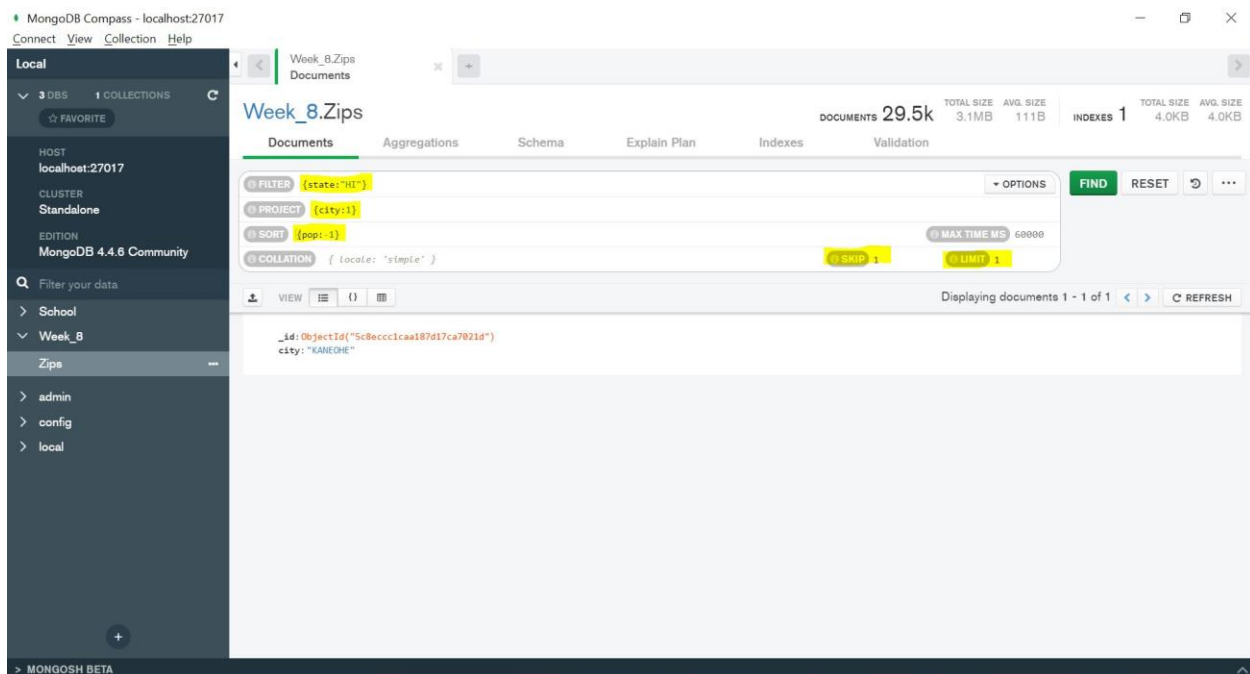
```
> db.Zips.find({"state":"NY"}, {"city":1}).count()  
1596
```

Question 12)

پ) با استفاده از نرم‌افزار MongoDB Compass به سوال زیر پاسخ گوید.

۱۲- نام دومین شهر پرجمعیت ایالت "HI" را بیابید.

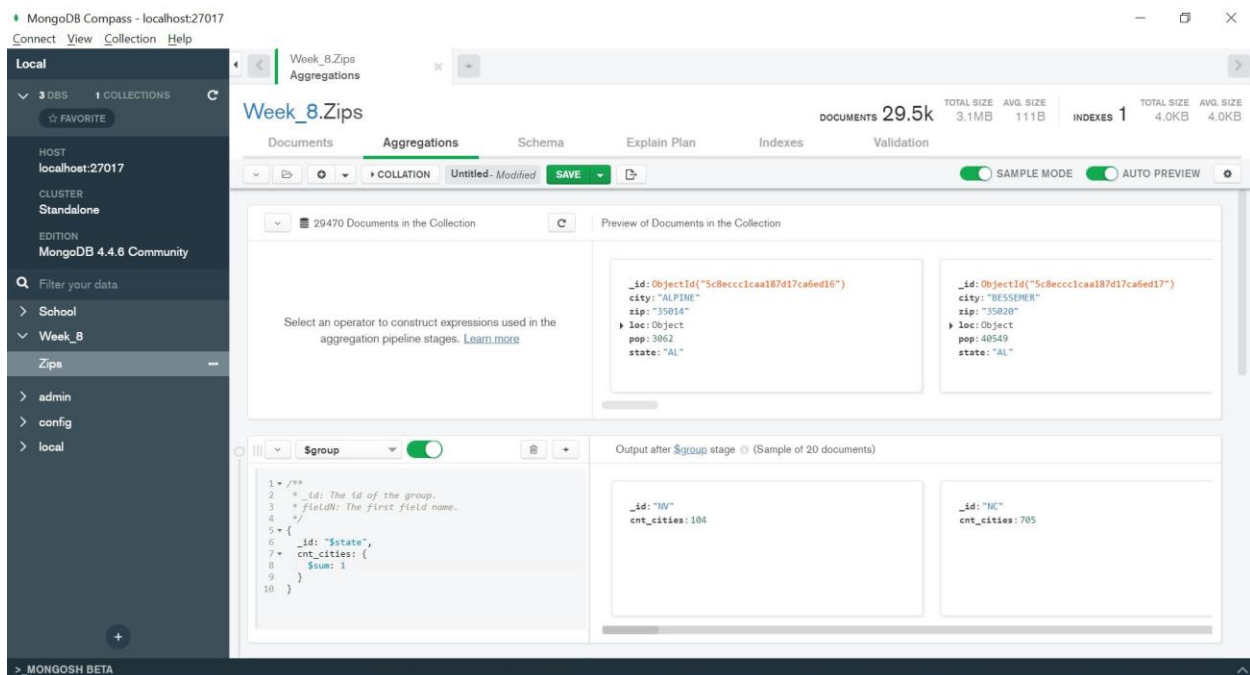
We can use the “skip : 1” feature in MongoDB Compass:



Question 13)

ت) در تب aggregation پایپلاین‌هایی ایجاد کنید تا به سوالات زیر پاسخ دهند.
۱۳- نام تمام ایالت‌ها را به همراه تعداد شهرهایشان مشخص کنید. سپس آن‌ها را به صورت نزولی مرتب کرده و نام ۳ ایالتی که بیشترین تعداد شهر دارند را مشخص کنید.

Grouping by the “state”s and calculating each of their number of cities:



Sorting by the number of cities in each of the states:



Projecting only the state and number of cities, and also limiting them to the top 3 populated states:

The screenshot shows the MongoDB Compass interface for the **\$project** stage. The configuration pane on the left contains the following JSON:

```
1 // **
2 * specifications: The fields to
3 * include or exclude.
4 */
5 {
6   "state":1,
7   "cnt_cities":1
8 }
```

The output pane on the right, titled "Output after \$project stage (Sample of 20 documents)", displays two sample documents:

```
{ "_id": "TX", "cnt_cities": 1676 }
{ "_id": "NY", "cnt_cities": 1596 }
```

And:

The screenshot shows the MongoDB Compass interface for the **\$limit** stage. The configuration pane on the left contains the following JSON:

```
1 // **
2 * Provide the number of documents to limit.
3 */
4 3
```

The output pane on the right, titled "Output after \$limit stage (Sample of 3 documents)", displays two sample documents:

```
{ "d": "NY", "t_cities": 1596 }
{ "_id": "CA", "cnt_cities": 1523 }
```

Question 14)

۱۴- پایپلانی را طراحی کنید که از بین ۱۰ ایالت پر جمعیت موجود در کالکشن، ۳ تا را به صورت راندوم انتخاب کند.

We should continue what was done for question 13 but limit the documents to 10 in the final stage:

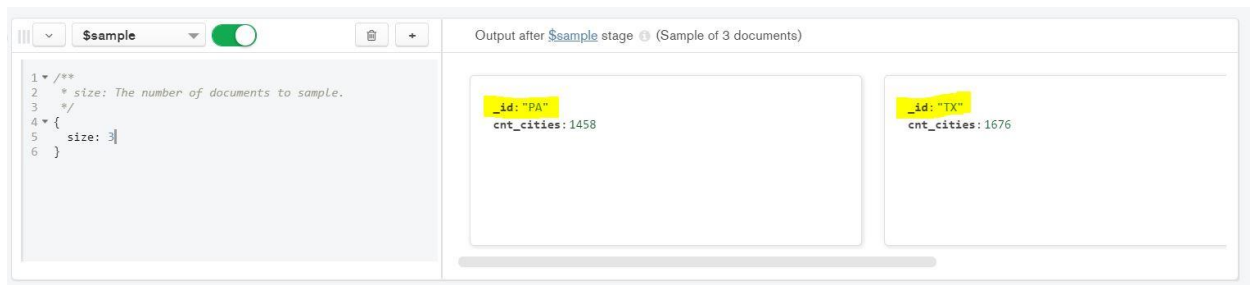
The screenshot shows the MongoDB Compass interface for the **\$limit** stage. The configuration pane on the left contains the following JSON:

```
1 // **
2 * Provide the number of documents to limit.
3 */
4 10
```

The output pane on the right, titled "Output after \$limit stage (Sample of 10 documents)", displays two sample documents:

```
{ "_id": "TX", "cnt_cities": 1676 }
{ "_id": "NY", "cnt_cities": 1596 }
```

Then, we should use “sample” to randomly chose 3 of the top 10 populated states:



And:

