

NoSql Tipo Documento - MongoDB

Everton André Thomas

16 de maio de 2018

1 Qual é a população de FISHERS ISLAND?

Comando:

```
1 db.zipcodes.aggregate([
2   { $match: { city: "FISHERS ISLAND" } },
3   { $group: { _id: { state: "$state", city: "$city" }, totalPop:
4     { $sum: "$pop" } } }
5 ])
```

Resultado:

```
1 { "_id" : "FISHERS ISLAND", "totalPop" : 329 }
```

2 Listar todas as cidades do estado "MA"

Comando:

```
1 db.zipcodes.distinct( "city", { state: "MA" } )
```

Resultado:

```
1 [
2   "AGAWAM",
3   "BARRE",
4   "CUSHMAN",
5   "BELCHERTOWN",
6   "BLANDFORD",
7   "CHESTER",
8   "CHICOPEE",
9   "CHESTERFIELD",
10  "BRIMFIELD",
11  "WESTOVER AFB",
12  "EAST LONGMEADOW",
13  "GILBERTVILLE",
14  "MOUNT TOM",
15  "FEEDING HILLS",
16  "GRANBY",
```

```

17     "GOSHEN",
18     "TOLLAND",
19     "CUMMINGTON",
20     "HADLEY",
21     ...
22     ...
23 ]

```

3 Listar todas as cidades com uma população entre 1 e 10 inclusive em sua população.

Comando:

```

1 db.zipcodes.aggregate([
2   { $group: { _id: { state: "$state", city: "$city"}, totalPop: {
3     $sum: "$pop" } } }, { $match: { totalPop: { $gte: 1, $lte:
4     10 }}}
5 ])

```

Resultado:

```

1 [
2   { "_id" : { "state" : "AK", "city" : "PORT ALSWORTH" }, "
3     totalPop" : 7 }
4   { "_id" : { "state" : "AK", "city" : "PLATINUM" }, "totalPop"
5     : 4 }
6   { "_id" : { "state" : "OR", "city" : "SUMMER LAKE" }, "
7     totalPop" : 1 }
8   { "_id" : { "state" : "NM", "city" : "QUAY" }, "totalPop" : 10
9     }
10  { "_id" : { "state" : "NM", "city" : "SAPELLO" }, "totalPop" :
11    2 }
12  { "_id" : { "state" : "NM", "city" : "LA JARA" }, "totalPop" :
13    5 }
14  { "_id" : { "state" : "NM", "city" : "JEMEZ SPRINGS" }, "
15    totalPop" : 1 }
16  { "_id" : { "state" : "UT", "city" : "MODENA" }, "totalPop" :
17    9 }
18  { "_id" : { "state" : "WY", "city" : "ALCOVA" }, "totalPop" :
19    10 }
20  { "_id" : { "state" : "WA", "city" : "BENGE" }, "totalPop" : 2
21    }
22  { "_id" : { "state" : "WY", "city" : "LOST SPRINGS" }, "
23    totalPop" : 6 }
24  { "_id" : { "state" : "CO", "city" : "ELK SPRINGS" }, "
25    totalPop" : 10 }
26  { "_id" : { "state" : "TX", "city" : "MULDOON" }, "totalPop" :
27    8 }
28  { "_id" : { "state" : "TX", "city" : "BEND" }, "totalPop" : 1
29    }
30  { "_id" : { "state" : "AR", "city" : "ARKANSAS CITY" }, "
31    totalPop" : 7 }

```

```

17 { "_id" : { "state" : "LA", "city" : "LOTTIE" }, "totalPop" :
18 9 }
19 { "_id" : { "state" : "NE", "city" : "LAKESIDE" }, "totalPop"
20 : 5 }
21 { "_id" : { "state" : "AK", "city" : "AMBLER" }, "totalPop" :
22 8 }
23 { "_id" : { "state" : "CA", "city" : "MAD RIVER" }, "totalPop"
24 : 6 }
25 { "_id" : { "state" : "KS", "city" : "NEW ALMELO" }, "totalPop"
    : 2 }
    ...
    ...
    ...
]

```

4 Listar todas as cidades do estado "MA" com uma população menor que 100.

Comando:

```

1 db.zipcodes.aggregate([
2   {$match: { state: "MA" }},
3   {$group: { _id: { city: "$city" }, totalPop: {$sum: "$pop"} }},
4   {$match: { totalPop: {$lt: 100} }}
5 ])

```

Resultado:

```

1 { "_id" : { "city" : "CUTTYHUNK" }, "totalPop" : 98 }
2 { "_id" : { "city" : "MONROE" }, "totalPop" : 97 }
3 { "_id" : { "city" : "BUCKLAND" }, "totalPop" : 16 }

```

5 Listar todos os estados de forma distinta, ou seja, sem repetição. (utilize distinct).

Comando:

```

1 db.zipcodes.distinct("state")

```

Resultado:

```

1 [
2   "MA",
3   "RI",
4   "NH",
5   "ME",
6   "VT",
7   "CT",
8   "NY",
9   "NJ",
10  "PA",

```

```

11     "DE",
12     "DC",
13     "MD",
14     "VA",
15     "WV",
16     "NC",
17     "SC",
18     "GA",
19     "FL",
20     "AL",
21     "TN",
22     "MS",
23     "KY",
24     "OH",
25     "IN",
26     "MI",
27     "IA",
28     "WI",
29     "MN",
30     "SD",
31     "ND",
32     "MT",
33     "IL",
34     "MO",
35     "KS",
36     "NE",
37     "LA",
38     "AR",
39     "OK",
40     "TX",
41     "CO",
42     "WY",
43     "ID",
44     "UT",
45     "AZ",
46     "NM",
47     "NV",
48     "CA",
49     "HI",
50     "OR",
51     "WA",
52     "AK"
53 ]

```

6 Listar todos os estados distintos que contém uma cidade com uma população acima de 100000.

Comando:

```

1 db.zipcodes.aggregate([
2     {$match: {pop: {$gt: 100000}}},
3     {$group: {_id: "$state"}}

```

```
4 ])
```

Resultado:

```
1 [  
2   { "_id" : "IL" }  
3   { "_id" : "NY" }  
4 ]
```

7 Qual é o tamanho da menor cidade (em população) de cada um dos estados?

Comando:

```
1 db.zipcodes.aggregate([  
2   { $group: { _id: { state: "$state", city: "$city"}, totalPop: { $sum: "$pop" } } },  
3   { $sort: { totalPop: 1 } },  
4   { $group: { _id: "$_id.state", smallPop: { $first: "$totalPop" } } } ])
```

Resultado:

```
1 [  
2   { "_id" : "DE", "smallPop" : 108 }  
3   { "_id" : "MO", "smallPop" : 44 }  
4   { "_id" : "IL", "smallPop" : 38 }  
5   { "_id" : "OH", "smallPop" : 38 }  
6   { "_id" : "NH", "smallPop" : 27 }  
7   { "_id" : "DC", "smallPop" : 21 }  
8   { "_id" : "ND", "smallPop" : 12 }  
9   { "_id" : "MD", "smallPop" : 32 }  
10  { "_id" : "MN", "smallPop" : 12 }  
11  { "_id" : "UT", "smallPop" : 9 }  
12  { "_id" : "WY", "smallPop" : 6 }  
13  { "_id" : "AZ", "smallPop" : 2 }  
14  { "_id" : "CT", "smallPop" : 25 }  
15  { "_id" : "WA", "smallPop" : 2 }  
16  { "_id" : "NV", "smallPop" : 1 }  
17  { "_id" : "NE", "smallPop" : 5 }  
18  { "_id" : "FL", "smallPop" : 0 }  
19  { "_id" : "MA", "smallPop" : 16 }  
20  { "_id" : "ME", "smallPop" : 0 }  
21  { "_id" : "NC", "smallPop" : 0 }  
22  { "_id" : "MT", "smallPop" : 7 }  
23  { "_id" : "NJ", "smallPop" : 17 }  
24  { "_id" : "WV", "smallPop" : 0 }  
25  { "_id" : "TN", "smallPop" : 2 }  
26  { "_id" : "NY", "smallPop" : 0 }  
27  { "_id" : "CO", "smallPop" : 0 }  
28  { "_id" : "AL", "smallPop" : 0 }  
29  { "_id" : "KS", "smallPop" : 0 }  
30  { "_id" : "SC", "smallPop" : 0 }
```

```

31 { "_id" : "VT", "smallPop" : 0 }
32 { "_id" : "PA", "smallPop" : 0 }
33 { "_id" : "LA", "smallPop" : 0 }
34 { "_id" : "AR", "smallPop" : 0 }
35 { "_id" : "RI", "smallPop" : 45 }
36 { "_id" : "CA", "smallPop" : 0 }
37 { "_id" : "NM", "smallPop" : 0 }
38 { "_id" : "OK", "smallPop" : 8 }
39 { "_id" : "ID", "smallPop" : 0 }
40 { "_id" : "IN", "smallPop" : 145 }
41 { "_id" : "KY", "smallPop" : 0 }
42 { "_id" : "WI", "smallPop" : 2 }
43 { "_id" : "MS", "smallPop" : 79 }
44 { "_id" : "IA", "smallPop" : 15 }
45 { "_id" : "HI", "smallPop" : 0 }
46 { "_id" : "GA", "smallPop" : 0 }
47 { "_id" : "OR", "smallPop" : 0 }
48 { "_id" : "MI", "smallPop" : 0 }
49 { "_id" : "SD", "smallPop" : 8 }
50 { "_id" : "TX", "smallPop" : 0 }
51 { "_id" : "VA", "smallPop" : 0 }
52 { "_id" : "AK", "smallPop" : 0 }
53 ]

```

8 Qual é o nome da menor cidade (em população) de cada um dos estados? Utilize a função de redução do map-reduce do comando de agrupamento/agregação para encontrar a menor população. (Desafio)

Comando:

Resultado:

9 Listar o tamanho médio da população para cada um dos estados.

Comando:

```

1 db.zipcodes.aggregate([
2   { $group: { _id: { state: "$state", city: "$city" }, totalPop:
3     { $sum: "$pop" } } },
4   { $group: { _id: "$_id.state", avgPop: { $avg: "$totalPop" }
5     } }
6 ])

```

Resultado:

```

1 [
2   { "_id" : "NH", "avgPop" : 5232.320754716981 }
3   { "_id" : "MA", "avgPop" : 14855.37037037037 }
4   { "_id" : "ME", "avgPop" : 3006.4901960784314 }
5   { "_id" : "NY", "avgPop" : 13131.680291970803 }
6   { "_id" : "VT", "avgPop" : 2315.8765432098767 }
7   { "_id" : "PA", "avgPop" : 8679.067202337472 }
8   { "_id" : "DC", "avgPop" : 303450 }
9   { "_id" : "DE", "avgPop" : 14481.91304347826 }
10  { "_id" : "VA", "avgPop" : 8526.177931034483 }
11  { "_id" : "SC", "avgPop" : 11139.626198083068 }
12  { "_id" : "FL", "avgPop" : 27400.958963282937 }
13  { "_id" : "AL", "avgPop" : 7907.2152641878665 }
14  { "_id" : "NJ", "avgPop" : 15775.89387755102 }
15  { "_id" : "WV", "avgPop" : 2771.4775888717154 }
16  { "_id" : "TN", "avgPop" : 9656.350495049504 }
17  { "_id" : "OH", "avgPop" : 12700.839578454332 }
18  { "_id" : "MD", "avgPop" : 12615.775725593667 }
19  { "_id" : "MN", "avgPop" : 5372.21375921376 }
20  { "_id" : "ND", "avgPop" : 1645.0309278350514 }
21  { "_id" : "NC", "avgPop" : 10622.815705128205 }
22 ]

```

10 Quantas cidades tem estado "WA"?

Comando:

```

1 db.zipcodes.aggregate([
2   {$match: {state: "WA"}},
3   {$group: { _id: { state: "$state", city: "$city" } }},
4   {$group: { _id: "$_id.state", cities: {$sum: 1}}}
5 ])

```

Resultado:

```

1 { "_id" : "WA", "cities" : 397 }

```

11 Listar todos os estados e o número de cidades que eles contêm. Utilize um pipeline de agregação para desenvolver esta atividade. Aqui encontra-se uma dica: não existe um operador de \$count, mas pode-se utilizar um \$sum:1 para fazer a mesma operação de contagem.

Comando:

```

1 db.zipcodes.aggregate([
2   {$group: { _id: { state: "$state", city: "$city" } }},
3   {$group: { _id: "$_id.state", cities: {$sum: 1} }}
4 ])

```

Resultado:

```
1 [
2   { "_id" : "NH", "cities" : 212 }
3   { "_id" : "MA", "cities" : 405 }
4   { "_id" : "ME", "cities" : 408 }
5   { "_id" : "NY", "cities" : 1370 }
6   { "_id" : "VT", "cities" : 243 }
7   { "_id" : "PA", "cities" : 1369 }
8   { "_id" : "DC", "cities" : 2 }
9   { "_id" : "DE", "cities" : 46 }
10  { "_id" : "VA", "cities" : 725 }
11  { "_id" : "SC", "cities" : 313 }
12  { "_id" : "FL", "cities" : 463 }
13  { "_id" : "AL", "cities" : 511 }
14  { "_id" : "NJ", "cities" : 490 }
15  { "_id" : "WV", "cities" : 647 }
16  { "_id" : "TN", "cities" : 505 }
17  { "_id" : "OH", "cities" : 854 }
18  { "_id" : "MD", "cities" : 379 }
19  { "_id" : "MN", "cities" : 814 }
20  { "_id" : "ND", "cities" : 388 }
21  { "_id" : "NC", "cities" : 624 }
22  ...
23  ...
24 ]
```

12 Listar todos os estados com menos de 100 cidades (estenda o pipeline da questão anterior).

Comando:

```
1 db.zipcodes.aggregate([
2   {$group: { _id: { state: "$state", city: "$city" } }},
3   {$group: { _id: "$_id.state", cities: {$sum: 1} }},
4   {$match: {cities: {$lt: 100}}}
5 ])
```

Resultado:

```
1 [
2   { "_id" : "DC", "cities" : 2 }
3   { "_id" : "DE", "cities" : 46 }
4   { "_id" : "RI", "cities" : 52 }
5   { "_id" : "HI", "cities" : 70 }
6   { "_id" : "NV", "cities" : 66 }
7 ]
```


13 Muitas cidades são listadas mais de uma vez, mesmo que estejam no mesmo estado. Quais são as cidades que aparecem mais de 50 vezes no mesmo estado?

Comando:

```
1 db.zipcodes.aggregate([
2   {$group: { _id: { state: "$state", city: "$city"}, cnt: {$sum
3     :1} }},
4   {$match: {cnt: {$gt: 50}}}]
```

Resultado:

```
1 [
2   { "_id" : { "state" : "CA", "city" : "LOS ANGELES" }, "cnt" :
3     56 }
4   { "_id" : { "state" : "TX", "city" : "HOUSTON" }, "cnt" : 93 }
```

14 Listar todos os nomes das cidades que aparecem em mais de vinte estados diferentes, juntamente com seus estados.

Comando:

```
1 db.zipcodes.aggregate([
2   {$group: {_id: {state: "$state", city: "$city"}}},
3   {$group: {_id: "$_id.city", cnt: {$sum: 1}, states: {$push: {
4     state: "$_id.state"}}}},
5   {$match: {cnt: {$gt: 20}}}]
```

Resultado:

```
1 [
2   {
3     "_id": "GREENVILLE",
4     "cnt": 22,
5     "states": [
6       {
7         "state": "IN"
8       },
9       {
10        "state": "OH"
11      },
12      {
13        "state": "KY"
14      },
15      {
16        "state": "MS"
17      },
18    ]
19  }
```

```

18     {
19         "state ":"TX"
20     },
21     {
22         "state ":"FL"
23     },
24     {
25         "state ":"IA"
26     },
27     {
28         "state ":"VA"
29     },
30     {
31         "state ":"NY"
32     },
33     {
34         "state ":"SC"
35     },
36     {
37         "state ":"WI"
38     },
39     {
40         "state ":"IL"
41     },
42     {
43         "state ":"WV"
44     },
45     {
46         "state ":"MI"
47     },
48     {
49         "state ":"ME"
50     },
51     {
52         "state ":"NC"
53     },
54     {
55         "state ":"UT"
56     },
57     {
58         "state ":"RI"
59     },
60     {
61         "state ":"DE"
62     },
63     {
64         "state ":"AL"
65     },
66     {
67         "state ":"MO"
68     },
69     {
70         "state ":"CA"
71     }

```

```

72     ]
73   },
74   {
75     "_id ":"ARLINGTON",
76     "cnt":22,
77     "states":[
78       {
79         "state ":"AZ"
80       },
81       {
82         "state ":"WA"
83       },
84       {
85         "state ":"CO"
86       },
87       {
88         "state ":"TX"
89       },
90       {
91         "state ":"IL"
92       },
93       {
94         "state ":"WI"
95       },
96       {
97         "state ":"IA"
98       },
99       {
100        "state ":"NE"
101      },
102      {
103        "state ":"KS"
104      },
105      {
106        "state ":"AL"
107      },
108      {
109        "state ":"KY"
110      },
111      {
112        "state ":"VA"
113      },
114      {
115        "state ":"OR"
116      },
117      {
118        "state ":"TN"
119      },
120      {
121        "state ":"NY"
122      },
123      {
124        "state ":"IN"
125      },

```

```

126         {
127             "state ":"GA"
128         },
129         {
130             "state ":"MN"
131         },
132         {
133             "state ":"OH"
134         },
135         {
136             "state ":"MA"
137         },
138         {
139             "state ":"SD"
140         },
141         {
142             "state ":"VT"
143         }
144     ],
145 },
146 {
147     "_id ":"SALEM",
148     "cnt":21,
149     "states":[
150         {
151             "state ":"NM"
152         },
153         {
154             "state ":"NE"
155         },
156         {
157             "state ":"SD"
158         },
159         {
160             "state ":"IA"
161         },
162         {
163             "state ":"IN"
164         },
165         {
166             "state ":"OH"
167         },
168         {
169             "state ":"MO"
170         },
171         {
172             "state ":"IL"
173         },
174         {
175             "state ":"AL"
176         },
177         {
178             "state ":"OR"
179         },

```

```

180         {
181             "state ":"FL"
182         },
183         {
184             "state ":"WV"
185         },
186         {
187             "state ":"KY"
188         },
189         {
190             "state ":"SC"
191         },
192         {
193             "state ":"VA"
194         },
195         {
196             "state ":"NY"
197         },
198         {
199             "state ":"NJ"
200         },
201         {
202             "state ":"WI"
203         },
204         {
205             "state ":"NH"
206         },
207         {
208             "state ":"MA"
209         },
210         {
211             "state ":"CT"
212         }
213     ]
214 },
215 {
216     "_id ":"FRANKLIN",
217     "cnt ":24,
218     "states ":[
219         {
220             "state ":"MO"
221         },
222         {
223             "state ":"MN"
224         },
225         {
226             "state ":"WI"
227         },
228         {
229             "state ":"IN"
230         },
231         {
232             "state ":"AZ"
233         },

```

```

234     {
235         "state ":"AR"
236     },
237     {
238         "state ":"TN"
239     },
240     {
241         "state ":"ID"
242     },
243     {
244         "state ":"NC"
245     },
246     {
247         "state ":"WV"
248     },
249     {
250         "state ":"LA"
251     },
252     {
253         "state ":"NY"
254     },
255     {
256         "state ":"VA"
257     },
258     {
259         "state ":"TX"
260     },
261     {
262         "state ":"AL"
263     },
264     {
265         "state ":"KY"
266     },
267     {
268         "state ":"NE"
269     },
270     {
271         "state ":"MI"
272     },
273     {
274         "state ":"NJ"
275     },
276     {
277         "state ":"VT"
278     },
279     {
280         "state ":"PA"
281     },
282     {
283         "state ":"ME"
284     },
285     {
286         "state ":"NH"
287     },

```

```

288         {
289             "state ":"MA"
290         }
291     ]
292 },
293 {
294     "_id ":"CHESTER",
295     "cnt":21,
296     "states ":[
297         {
298             "state ":"CA"
299         },
300         {
301             "state ":"OK"
302         },
303         {
304             "state ":"NE"
305         },
306         {
307             "state ":"IL"
308         },
309         {
310             "state ":"MT"
311         },
312         {
313             "state ":"TX"
314         },
315         {
316             "state ":"IA"
317         },
318         {
319             "state ":"AR"
320         },
321         {
322             "state ":"SD"
323         },
324         {
325             "state ":"SC"
326         },
327         {
328             "state ":"VA"
329         },
330         {
331             "state ":"MD"
332         },
333         {
334             "state ":"GA"
335         },
336         {
337             "state ":"PA"
338         },
339         {
340             "state ":"NY"
341         },

```

```

342     {
343         "state ":"CT"
344     },
345     {
346         "state ":"VT"
347     },
348     {
349         "state ":"NJ"
350     },
351     {
352         "state ":"WV"
353     },
354     {
355         "state ":"NH"
356     },
357     {
358         "state ":"MA"
359     }
360 ]
361 },
362 {
363     "_id ":"MADISON",
364     "cnt":23,
365     "states":[
366         {
367             "state ":"NE"
368         },
369         {
370             "state ":"MO"
371         },
372         {
373             "state ":"SD"
374         },
375         {
376             "state ":"MN"
377         },
378         {
379             "state ":"WI"
380         },
381         {
382             "state ":"IN"
383         },
384         {
385             "state ":"MS"
386         },
387         {
388             "state ":"FL"
389         },
390         {
391             "state ":"GA"
392         },
393         {
394             "state ":"WV"
395         },

```



```

396         {
397             "state ":"KS"
398         },
399         {
400             "state ":"OH"
401         },
402         {
403             "state ":"PA"
404         },
405         {
406             "state ":"TN"
407         },
408         {
409             "state ":"NY"
410         },
411         {
412             "state ":"CA"
413         },
414         {
415             "state ":"CT"
416         },
417         {
418             "state ":"ME"
419         },
420         {
421             "state ":"NJ"
422         },
423         {
424             "state ":"NH"
425         },
426         {
427             "state ":"IL"
428         },
429         {
430             "state ":"NC"
431         },
432         {
433             "state ":"SC"
434         }
435     ]
436 },
437 {
438     "_id ":"CLINTON",
439     "cnt":24,
440     "states":[
441         {
442             "state ":"WA"
443         },
444         {
445             "state ":"AR"
446         },
447         {
448             "state ":"LA"
449         },

```

```

450     {
451         "state ":" IL"
452     },
453     {
454         "state ":" WI"
455     },
456     {
457         "state ":" IN"
458     },
459     {
460         "state ":" MI"
461     },
462     {
463         "state ":" KY"
464     },
465     {
466         "state ":" MS"
467     },
468     {
469         "state ":" TN"
470     },
471     {
472         "state ":" OK"
473     },
474     {
475         "state ":" SC"
476     },
477     {
478         "state ":" MD"
479     },
480     {
481         "state ":" OH"
482     },
483     {
484         "state ":" MN"
485     },
486     {
487         "state ":" NY"
488     },
489     {
490         "state ":" NJ"
491     },
492     {
493         "state ":" NC"
494     },
495     {
496         "state ":" ME"
497     },
498     {
499         "state ":" PA"
500     },
501     {
502         "state ":" MT"
503     },

```

```

504     {
505         "state": "IA"
506     },
507     {
508         "state": "CT"
509     },
510     {
511         "state": "MA"
512     }
513 ]
514 }
515 ]

```

15 Qual é a população total de todos os códigos postais agrupados individualmente de NY?

Comando:

```

1 db.zipcodes.aggregate([
2     {$match: {state: "NY"}},
3     {$group: { _id: { zip: "$_id" }, totalPop: { $sum: "$pop" } }},
4     {$group: { _id: null, totalPopZipCode: { $sum: "$totalPop" } } }
5 ])

```

Resultado:

```

1 { "_id" : null, "totalPopZipCode" : 17990402 }

```

16 Quantos códigos postais diferentes existem no estado de NY?

Comando:

```

1 db.zipcodes.aggregate([
2     {$match: {state: "NY"}},
3     {$group: { _id: { zip: "$_id" }, cnt: { $sum: 1 } }},
4     {$group: { _id: null, cntZipCode: { $sum: "$cnt" } } }
5 ])

```

Resultado:

```

1 "_id" : null, "cntZipCode" : 1595 }

```

17 Encontre todos os estados com uma população total acima de 10 milhões (isso requer uma soma e uma seleção) - não basta pesquisar entradas (lookup) únicas com uma população de mais de 10 milhões.

Comando:

```
1 db.zipcodes.aggregate([
2   { $group: { _id: "$state", totalPop: { $sum: "$pop" } } },
3   { $match: { totalPop: { $gt: 10000000 } } }
4 ])
```

Resultado:

```
1 [
2   { "_id" : "IL", "totalPop" : 11427576 }
3   { "_id" : "OH", "totalPop" : 10846517 }
4   { "_id" : "FL", "totalPop" : 12686644 }
5   { "_id" : "NY", "totalPop" : 17990402 }
6   { "_id" : "PA", "totalPop" : 11881643 }
7   { "_id" : "TX", "totalPop" : 16984601 }
8   { "_id" : "CA", "totalPop" : 29754890 }
9 ]
```