

PROIECT ABD

Proiect "Administrarea bazelor de date"

1. Overview

The goal of the project is to assess the MongoDB database skills of the student. The project has a medium difficulty level and is relevant to industry employers of today.

2. Project description

You are required to make several statistical computations on some US Zips dataset using MongoDB as the database platform.

Prerequisites:

- Download the latest US Zips dataset from <https://simplemaps.com/data/us-zips> (choose the free tier). The dataset has approximately 33k entries.
- Create a MongoDB instance. You may use your own MongoDB Atlas instance in cloud or use a local instance. For local instances Docker is preferred, but you may also choose to install MongoDB as a standalone server on your OS.
- Import the dataset into the MongoDB instance.

Requirements:

- a) Get the states with a total population of over 10 million.
- b) Get the average city population by state.
- c) Get the largest and the smallest city in each state.
- d) Get the largest and the smallest counties in each state.
- e) Get the nearest 10 zips from one of Chicago's landmarks, the Willis Tower situated at coordinates 41.878876, -87.635918.
- f) Get the total population situated between 50 and 200 kms around New York's landmark, the Statue of Liberty at coordinates 40.689247, -74.044502.

Notes:

- Create the indexes you deem relevant for your collection. You will be asked on the performance of your indexes so be prepared to defend your choice, preferably by analyzing the execution statistics.
- For requirements e) and f), you may add a geo field to your collection in order to leverage geospatial query operators.
- Your solution must be original so please don't rely on cheating.

3. Scoring:

- Requirement a) - 1 point
- Requirement b) - 1.5 points
- Requirement c) - 1.5 points
- Requirement d) - 1.5 points
- Requirement e) - 2 points
- Requirement f) - 2.5 points

In order to pass, you must earn 5 points or more.

4. Solution Delivery

Upload your final project solution to the designated area in the virtual campus.

PROIECT ABD

1. Open PowerShell

Change the directory to where the project is.

Ex: `cd E:\Master\ABD`

2. Start a local MongoDB instance running inside a Docker container

`docker run -d --name mongo-project-abd -p 27017:27017 -e MONGO_INITDB_ROOT_USERNAME=madaUser -e MONGO_INITDB_ROOT_PASSWORD=madaUser mongo`

3. Use mongoimport tool to import the sample dataset

`mongoimport --db=abd_project --collection=states --file=simplemaps/uszips.csv --type=csv --headerline mongod://madaUser:madaUser@localhost:27017/?authSource=admin`

4. Connect to the local MongoDB instance using Mongo Shell

`mongosh mongod://madaUser:madaUser@localhost:27017/?authSource=admin`

5. Change the database to the one from your project

Ex: `use abd_project`

6. Create index for state_name and population:

`db.states.createIndex({state_name : 1})`

`db.states.createIndex({population : 1})`

7. Get the states with a total population of over 10 million.

`db.states.aggregate([
{`

```
$group: {
  _id: "$state_name",
  total_population_over_10million: { $sum: "$population" }
},
{
  $match: {
    total_population_over_10million: { $gt: 10000000 }
  }
}
])
```

8. Get the average city population by state.

```
db.states.aggregate([
{
  $group: {
    _id: "$state_name",
    average_city_population: { $avg: "$population" }
  }
}
])
```

9. Get the largest and the smallest city in each state.

```
db.states.aggregate([
{
  $sort: {
    state_name: 1,
    population: -1
  }
},
{
  $group: {
    _id: "$state_name",
    LARGEST_CITY: { $first: { city: "$city", population: "$population" } },
    SMALLEST_CITY: { $last: { city: "$city", population: "$population" } }
  }
},
{
  $project: {
    state_name: "$_id",
    LARGEST_CITY: "$LARGEST_CITY.city",
    LARGEST_CITY_POPULATION: "$LARGEST_CITY.population",
    SMALLEST_CITY: "$SMALLEST_CITY.city",
    SMALLEST_CITY_POPULATION: "$SMALLEST_CITY.population"
  }
}
])
```

10. Get the largest and the smallest counties in each state.

```
db.states.aggregate([
{
  $sort: {
    state_name: 1,
    population: -1
  }
},
{
  $group: {
    _id: "$state_name",
    LARGEST_COUNTY: { $first: { county_name: "$county_name", population: "$population" } },
    SMALLEST_COUNTY: { $last: { county_name: "$county_name", population: "$population" } }
  }
}
])
```

```
}  
},  
{  
$project: {  
  state_name: "$_id",  
  LARGEST_COUNTY: "$LARGEST_COUNTY.county_name",  
  LARGEST_COUNTY_POPULATION: "$LARGEST_COUNTY.population",  
  SMALLEST_COUNTY: "$SMALLEST_COUNTY.county_name",  
  SMALLEST_COUNTY_POPULATION: "$SMALLEST_COUNTY.population"  
}  
}  
}  
})
```

Get the states with a total population of over 10 million.

```
mongosh mongodb://<credentials>@localhost:27017/?authSource=admin&directConnection=true&serverSelectionTimeoutMS=2000  
PS C:\Users\turlea> cd E:\Master\ABD  
PS E:\Master\ABD> docker run -d --name mongo-project-abd -p 27017:27017 -e MONGO_INITDB_ROOT_USERNAME=madaUser -e MONGO_INITDB_ROOT_PASSWORD=madaUser mongo  
951c3f4863826c2d43fb0ce971602252d2bc2c533095d226a7087b2b856f623  
PS E:\Master\ABD> mongoimport --db=abd project --collection=states --file=simplemaps/us/zips.csv --type=csv --headerline mongod://madaUser:madaUser@localhost:27017/?authSource=admin  
2024-05-21T18:44:12.783+0300   connected to: mongod://[**REDACTED**]@localhost:27017/?authSource=admin  
2024-05-21T18:44:14.025+0300   33788 document(s) imported successfully. 0 document(s) failed to import.  
PS E:\Master\ABD> mongosh mongodb://madaUser:madaUser@localhost:27017/?authSource=admin  
Current Mongosh Log ID: 664cc1aad51e791b803e86bf  
Connecting to:      mongodb://<credentials>@localhost:27017/?authSource=admin&directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.0  
Using Mongosh:      2.0.0  
Mongosh 2.2.6 is available for download: https://www.mongodb.com/try/download/shell  
For mongosh info see: https://docs.mongodb.com/mongosh-shell/  
  
-----  
The server generated these startup warnings when booting  
2024-05-21T15:43:15.924+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem  
2024-05-21T15:43:16.702+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never' in this binary version  
2024-05-21T15:43:16.702+00:00: vm.max_map_count is too low  
-----  
test> use abd_project  
switched to db abd_project  
abd_project> db.states.createIndex({state_name : 1})  
state_name_1  
abd_project> db.states.createIndex({population : 1})  
population_1  
abd_project> db.states.aggregate([  
... {  
...   $group: {  
...     _id: "$state_name",  
...     total_population_over_10million: { $sum: "$population" }  
...   },  
...   $match: {  
...     total_population_over_10million: { $gt: 10000000 }  
...   }  
... })  
[  
  { _id: "Michigan", total_population_over_10million: 10057902 },  
  { _id: "California", total_population_over_10million: 39354820 },  
  { _id: "Georgia", total_population_over_10million: 10722352 },  
  { _id: "New York", total_population_over_10million: 19994379 },  
  { _id: "Florida", total_population_over_10million: 21632200 },  
  { _id: "Pennsylvania", total_population_over_10million: 12989208 },  
  { _id: "Illinois", total_population_over_10million: 12757583 },  
  { _id: "North Carolina", total_population_over_10million: 10470214 },  
  { _id: "Ohio", total_population_over_10million: 11774683 },  
  { _id: "Texas", total_population_over_10million: 29242696 }  
]  
abd_project>
```

Get the average city population by state.

```
abd_project> db.states.aggregate([  
... {  
...   $group: {  
...     _id: "$state_name",  
...     average_city_population: { $avg: "$population" }  
...   }  
... })  
[  
  { _id: "New York", average_city_population: 10949.824205914567 },  
  { _id: "Arkansas", average_city_population: 4908.718699186992 },  
  { _id: "Georgia", average_city_population: 14277.4327561249 },  
  { _id: "New Mexico", average_city_population: 5690.2318052992 },  
  { _id: "Mississippi", average_city_population: 6929.10772837237 },  
  { _id: "Idaho", average_city_population: 6644.727598566308 },  
  { _id: "Nevada", average_city_population: 17154.093922651933 },  
  { _id: "Wyoming", average_city_population: 3228.921787709497 },  
  { _id: "Guam", average_city_population: null },  
  { _id: "Northern Mariana Islands", average_city_population: null },  
  { _id: "Tennessee", average_city_population: 10874.68710691824 },  
  { _id: "South", average_city_population: 14954.525773195875 },  
  { _id: "Missouri", average_city_population: 7512.318007662835 },  
  { _id: "California", average_city_population: 21839.522752497225 },  
  { _id: "Iowa", average_city_population: 3287.955670103093 },  
  { _id: "West Virginia", average_city_population: 2429.009485094851 },  
  { _id: "New Hampshire", average_city_population: 5585.748890688259 },  
  { _id: "New Jersey", average_city_population: 15466.660535117056 },  
  { _id: "Michigan", average_city_population: 10139.014112903225 },  
  { _id: "Nada Island", average_city_population: 13909.604930271605 }  
]  
Type "it" for more  
abd_project>
```

Get the largest and the smallest city in each state.

```
abd_project> db.states.aggregate([
...   {
...     $sort: {
...       state_name: 1,
...       population: -1
...     },
...   },
...   {
...     $group: {
...       _id: "$state_name",
...       LARGEST_CITY: { $first: { city: "$city", population: "$population" } },
...       SMALLEST_CITY: { $last: { city: "$city", population: "$population" } }
...     },
...   },
...   {
...     $project: {
...       state_name: "$_id",
...       LARGEST_CITY: "$LARGEST_CITY.city",
...       LARGEST_CITY_POPULATION: "$LARGEST_CITY.population",
...       SMALLEST_CITY: "$SMALLEST_CITY.city",
...       SMALLEST_CITY_POPULATION: "$SMALLEST_CITY.population"
...     }
...   }
... ])
[
  {
    _id: 'American Samoa',
    state_name: 'American Samoa',
    LARGEST_CITY: ' ',
    LARGEST_CITY_POPULATION: ' ',
    SMALLEST_CITY: ' ',
    SMALLEST_CITY_POPULATION: ' ',
  },
  {
    _id: 'New Hampshire',
    state_name: 'New Hampshire',
    LARGEST_CITY: 'Manchester',
    LARGEST_CITY_POPULATION: 37893,
    SMALLEST_CITY: ' ',
    SMALLEST_CITY_POPULATION: 0
  },
  {
    _id: 'Iowa',
    state_name: 'Iowa',
    LARGEST_CITY: 'Des Moines',
    LARGEST_CITY_POPULATION: 45195,
    SMALLEST_CITY: ' ',
    SMALLEST_CITY_POPULATION: 0
  },
  {
    _id: 'New Jersey',
    state_name: 'New Jersey',
    LARGEST_CITY: 'Lakewood',
  },
]
```

Get the largest and the smallest counties in each state.

```
abd_project> db.states.aggregate([
...   {
...     $sort: {
...       state_name: 1,
...       population: -1
...     },
...   },
...   {
...     $group: {
...       _id: "$state_name",
...       LARGEST_COUNTY: { $first: { county_name: "$county_name", population: "$population" } },
...       SMALLEST_COUNTY: { $last: { county_name: "$county_name", population: "$population" } }
...     },
...   },
...   {
...     $project: {
...       state_name: "$_id",
...       LARGEST_COUNTY: "$LARGEST_COUNTY.county_name",
...       LARGEST_COUNTY_POPULATION: "$LARGEST_COUNTY.population",
...       SMALLEST_COUNTY: "$SMALLEST_COUNTY.county_name",
...       SMALLEST_COUNTY_POPULATION: "$SMALLEST_COUNTY.population"
...     }
...   }
... ])
[
  {
    _id: 'Rhode Island',
    state_name: 'Rhode Island',
    LARGEST_COUNTY: 'Providence',
    LARGEST_COUNTY_POPULATION: 83451,
    SMALLEST_COUNTY: 'Bristol',
    SMALLEST_COUNTY_POPULATION: 0
  },
  {
    _id: 'New Hampshire',
    state_name: 'New Hampshire',
    LARGEST_COUNTY: 'Hillsborough',
    LARGEST_COUNTY_POPULATION: 37893,
    SMALLEST_COUNTY: 'Cheshire',
    SMALLEST_COUNTY_POPULATION: 0
  },
  {
    _id: 'New Jersey',
    state_name: 'New Jersey',
    LARGEST_COUNTY: 'Ocean',
    LARGEST_COUNTY_POPULATION: 134808,
    SMALLEST_COUNTY: 'Hudson',
  },
]
```