**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.

REZOLVARE:

a)

db.proiect.aggregate([{$group: {\_id: {state\_name: "$state\_name"}, state\_population: {$sum: "$population"}}}, {$match: {state\_population: {$gt: 10000000}}}])

O imagine care conține text, Software multimedia, captură de ecran

Descriere generată automat

b)

db.proiect.aggregate([ { $group: { \_id: {state\_name: "$state\_name"}, total\_population: { $sum: "$population" }, All\_cities: { $sum: 1 } } }, { $addFields: { Average\_city\_population\_by\_state: { $divide: ["$total\_population", "$All\_cities"] } } }, { $sort: { All\_cities: -1 } }, ] )



c)

db.proiect.aggregate([ { $group: { \_id: { state\_name: "$state\_name", city: "$city" }, population: { $sum: "$population" } } }, { $sort: { population: 1 } }, { $group: { \_id: {state\_name: "$\_id.state\_name"}, cities: { $push: { city: "$\_id.city", population: "$population" } }} }, {$project: {\_id: 1, smallest\_city: { $arrayElemAt: ["$cities", 0] }, largest\_\_city: { $arrayElemAt: ["$cities", -1] } }}])

O imagine care conține text, captură de ecran, afișaj, software

Descriere generată automat

d)

db.proiect.aggregate([{ $group: { \_id: { state\_name: "$state\_name", county\_name: "$county\_name" }, population: { $sum: "$population" } } }, { $sort: { population: 1 } }, { $group: { \_id: {state\_name: "$\_id.state\_name"}, counties: { $push: { county\_name: "$\_id.county\_name", population: "$population" } }} }, {$project: {\_id: 1, smallest\_county: { $arrayElemAt: ["$counties", 0] }, largest\_\_county: { $arrayElemAt: ["$counties", -1] } }}])

O imagine care conține text, captură de ecran, Font, software

Descriere generată automat

e)

db.proiect.find({ location: { $near: { $geometry: { type: "Point", coordinates: [-87.635918, 41.878876] } } } }).limit(10)

O imagine care conține captură de ecran, text, Software multimedia, software

Descriere generată automat

f)

db.proiect.aggregate([{$geoNear: { near: { type: "Point", coordinates : [-74.044502, 40.689247] }, distanceField: "distance", minDistance: 50000, maxDistance: 200000}}, { $group: { \_id: 0, Population: {$sum: "$population"}}}])

O imagine care conține text, software, Software multimedia, captură de ecran

Descriere generată automat