8-1 Assignment: Data Aggregation Pipeline

Using the mongoimport tool, **create the database** “companies” by loading the documents found in the “companies.json” file into the “research” collection. This file is located in the “/usr/local/datasets/” directory in Apporto. Verify your load by issuing the following queries:

1. db.research.find({"name" : "AdventNet"})
2. db.research.find({"founded\_year" : 1996},{"name" : 1}).limit(10)

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

Perform the following tasks **using MongoDB queries**:

1. List only the first 20 names of companies founded after the year 2010, ordered alphabetically.
2. List only the first 20 names of companies with offices in either California or Texas, ordered by the number of employees and sorted largest to smallest.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

**Design and implement a MongoDB aggregation pipeline** to show the total number of offices by state for all companies that have offices in the United States. Be sure that you account for the fact that some companies have offices in several states. Explain your aggregation pipeline.

A computer screen shot of a program code

AI-generated content may be incorrect.

db.research.aggregate([])

initiates the aggregation pipeline on the research collection of the companies database. It will process documents through a series of stages defined inside the square brackets.

{$match: {"offices.country\_code": "USA"}},

filters the documents

It only passes documents where the country\_code field inside the offices array equals "USA"

{$unwind: "$offices"},

deconstructs the offices array, creating a separate document for each office.

{$group: {\_id: "$offices.state\_code", totalOffices: {$sum: 1}}},

groups the documents by the state\_code field. It counts how many offices are in each state using $sum: 1

{$sort: {\_id: 1}},

sorts the results alphabetically by the \_id field

{$project: {state\_code: "$\_id", totalOffices: 1, \_id: 0}}

formats the final output. It renames the \_id field to state\_code for clarity. It includes the totalOffices field, which shows the count of offices for each state. It removes the default \_id field from the output by setting it to 0.