**Data Aggregation Pipeline**

Justin Starr

Department of STEM

CS 340 – Client/Server Development

Professor Reed Perkins

March 3, 2024

**Data Aggregation Pipeline**

**Each page of this document contains a screenshot of each command that was executed or snippets of required code…**

1. Using the mongoimport tool, create the database “companies” by loading the documents found in the “companies.json” file into the “research” collection. Verify your load by issuing the following queries: part a – db.research.find({“name”: “AdventNet”})

A screenshot of a computer program

Description automatically generated

Part 1 – b: db.research.find({“founded\_year”: 1996},{“name”: 1}).limit(10)

A screenshot of a computer

Description automatically generated

Part 2 – a – Perform the following task using MongoDB queries: List only the first 20 names of companies founded after the year 2010, ordered alphabetically.

A screenshot of a computer screen

Description automatically generated

Part 2 – b – Perform the following task using MongoDB queries: 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 computer screen shot of a program

Description automatically generated

Part 3 – 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.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Results Continued

A screenshot of a computer

Description automatically generated

End of results

An aggregation pipeline comprises different stages and is declared in an array and sequentially executed, with the output of every stage being the input of the next stage (Gaimas, A., 2022).

We begin with the $unwind stage which transforms an array of *n* elements into *n* documents, mapping each document to one element of the array, and then passing the documents to the next stage of the pipeline (Giamas, A., 2022). The next stage is the “$match” stage where we want to look for offices with the “USA” country code. The aggregation pipeline then moves to the “$group” stage, where the results will be grouped by state code. The final stage counts the total number of offices by state code and outputs that number.

**References**

Alex Giamas. (2022). Mastering MongoDB 6.x : Expert Techniques to Run High-volume and

Fault-tolerant Database Solutions Using MongoDB 6.x: Vol. Third edition. Packt Publishing.