Part 1:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Part 3:

A screenshot of a computer screen

Description automatically generated

Part 4:

A screenshot of a computer

Description automatically generated

Part 5:

A screenshot of a computer

Description automatically generated

Part 6:

A screenshot of a computer

Description automatically generated

Part 7 - 9:

A screenshot of a computer

Description automatically generated

Part 10:

Aggregate operations are expressions used to generate reduced and summarized results in MongoDB. The Aggregation Framework provides a powerful set of tools to perform complex data processing and analysis operations on collections of data and allows users to create a pipeline that consists of one or more stages, each of which performs a specific operation on the data.

Data flows through a series of stages, each of which performs a specific operation on a collection of data and passes the result to the next stage. Each stage is defined by an operator (e.g., $match, $group, etc.), and different operators implement different processing logic.

Match: { $match: { key: “**condition**” } }. It filters the data that matches the conditions from a collection of data, like the “WHERE” in sql.

Group: { $group: { \_id: "$**field"**, count: { $sum: 1 } } . This operator group documents according to specified fields and performs aggregation calculations such as summing and averaging.

Sort: {$sort: { key: -1 } }. Sort the result by specified order and attributes.

Project: { $project: { <specification(s)> } } Passes along the documents with the requested fields to the next stage in the pipeline. The specified fields can be existing fields from the input documents or newly computed fields. It is like “SELECT” in sql.

In this assignment, $match is used to filter the data that matches the criteria, for example, to find order by date or find orders with total price greater than 500. And $group is used to count number of orders in the result.

Part 9:

**A computer screen with white text

Description automatically generated**

Since my mock data does not have a New York City order on 1/9/2021, I have used a Brooklyn order on 8/18 2021 here to show what happens when the query result in Part 9 has data.