Week 2 Lab Summary: MongoDB CRUD and Aggregation Tasks

**1. Setting Up the Database and Importing Data**

* **Action**: Created a database named peopledb and a collection named people. Imported the people.csv file into the people collection using MongoDB Compass.
* **Reflection**: This step helped me understand how to use MongoDB Compass for setting up databases and collections and importing data directly from CSV files. I ensured fields like Age and Salary were imported as numeric types for accurate data processing.

**Performing CRUD Operations**

* **Insert a Document**:
  + Used the insertOne command to manually add a new document to the people collection.

A computer screen shot of a computer program

Description automatically generated

**Update a Document**:

* Located a document by a specific field (like First Name and Last Name) and updated fields using updateOne.

A computer screen shot of a program

Description automatically generated

**Delete a Document**:

* Found and deleted a document by a unique identifier.

**Reflection**: These CRUD operations reinforced the basics of manipulating data in MongoDB. It was particularly useful to see how updates and deletions are handled in a document-oriented database.

**Aggregation Pipeline Queries**

* **Query 1**: Average, Min, and Max Age and Salary for People with Master’s Degree, Grouped by Marital Status.
* **Command**: Used $match to filter by Education: Master and $group to aggregate AvgAge, MinAge, MaxAge, AvgSalary, MinSalary, and MaxSalary by marital status.

A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Query 2**: Min, Max, and Average Salary for Each Age Group of Females.

* **Command**: Used $match to filter for Gender: Female and $group to aggregate salaries for each age group.

A screen shot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

**Query 3**: Min, Max, and Average Salary for Each Age Group of Males.

* **Command**: Used $match to filter for Gender: Male and $group to aggregate salaries by age.

A screen shot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

**Query 4**: Count of Married and Unmarried Males and Females.

* **Command**: Used $group to count documents based on both Gender and Marital Status.

A screen shot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

**Reflection**: Working with the aggregation pipeline was a great way to learn how MongoDB handles complex data analysis. The $match and $group stages made it clear how to filter and summarize data, and I could see the efficiency of MongoDB’s approach to real-time analytics.

**Reflection Report**

* I documented each query and took screenshots of the results in MongoDB Compass to capture the output. I also summarized my observations, including the ease of performing data analysis with MongoDB’s aggregation framework.
* **Key Takeaways**: This lab solidified my understanding of MongoDB’s CRUD operations and aggregation capabilities. The hands-on experience with MongoDB Compass provided practical insight into managing and analyzing data, especially with the pipeline method for aggregations.