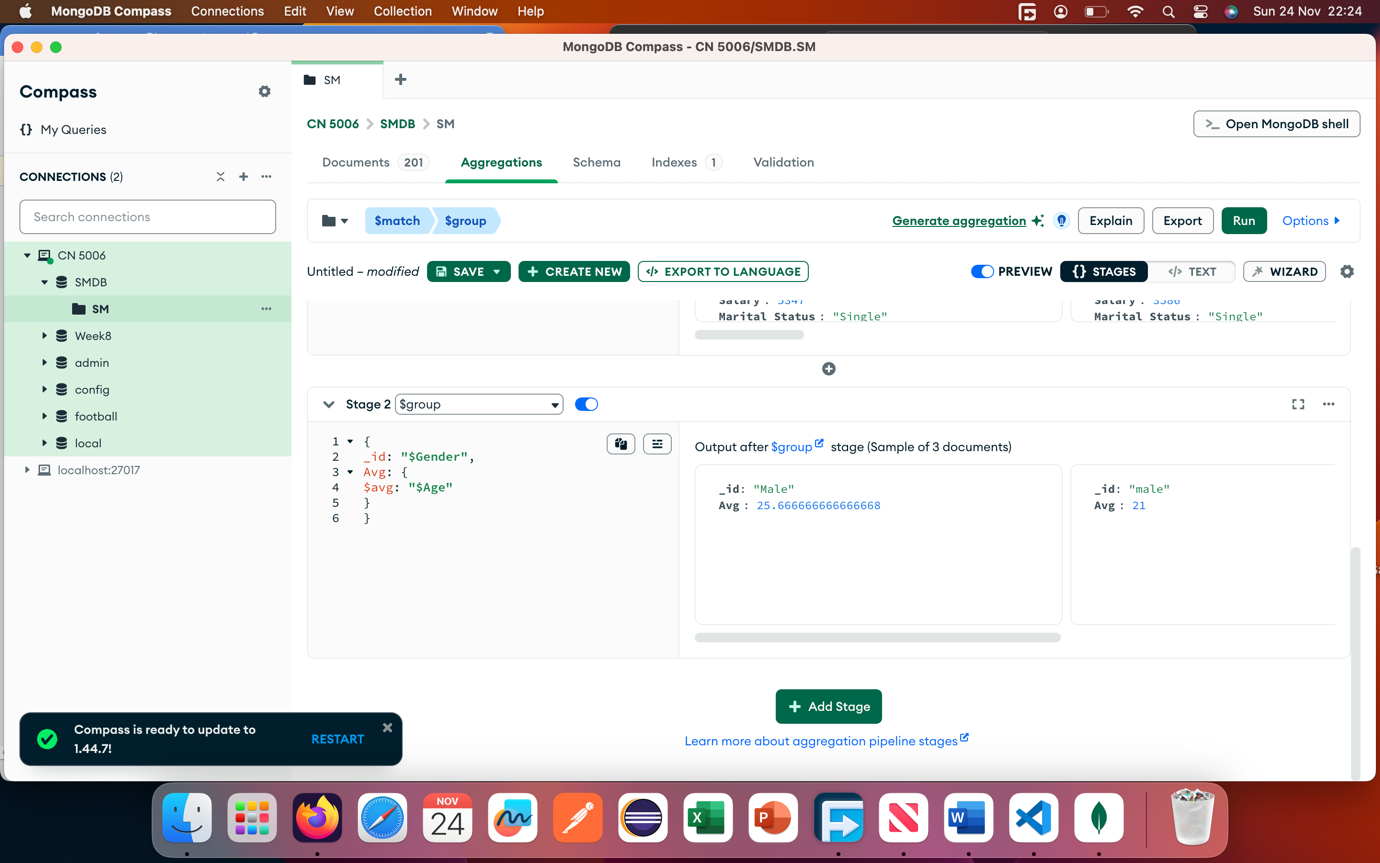
Week 2

Today's lab work focused on understanding the practical use of MongoDB by creating a database and collection, importing data, and performing CRUD (Create, Read, Update, and Delete) operations. The task was centered around managing and analyzing a dataset, people.csv, which included demographic and financial data of individuals.

**Steps and Activities**

1. **Database and Collection Creation**  
   The first step involved creating a MongoDB database and a collection to store the data. This provided a structured repository to organize and manage the dataset effectively. The database was named appropriately to reflect its purpose, and a people collection was created within it. This step laid the groundwork for subsequent data handling.
2. **Data Import and Record Addition**  
   Using MongoDB tools, the people.csv file was imported into the people collection. This process converted the CSV data into JSON documents, making them compatible with MongoDB's document-based data model. After verifying the import, additional records were manually inserted into the collection, demonstrating the flexibility of adding new data as needed.
3. **Data Updates**  
   Once the data was in place, specific records were updated to reflect changes or correct errors. For example, an individual's education level was modified from "Bachelor" to "Master's" to showcase how updates can be applied to refine the dataset. This operation highlighted MongoDB's capability to modify existing data dynamically.
4. **Record Deletion**  
   Finally, we explored how to delete records from the database. This included removing a subset of data based on specified criteria, such as deleting all records of unmarried individuals. This step demonstrated the importance of data hygiene and how MongoDB allows precise control over the data lifec
5. Exercise1

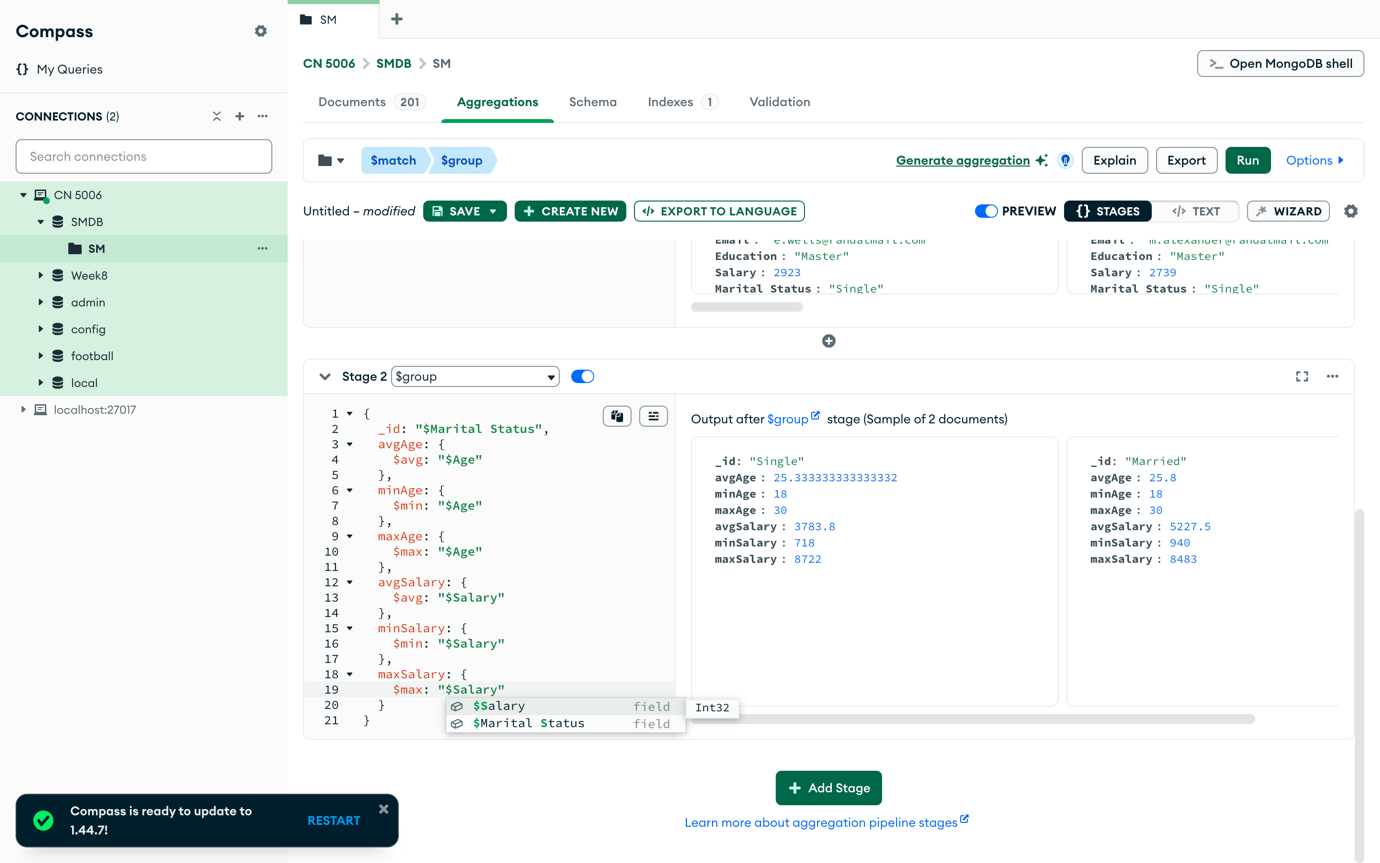


Now we will write queries for the following questions:

Question1

### 1: Master's Education Grouped by Marital Status

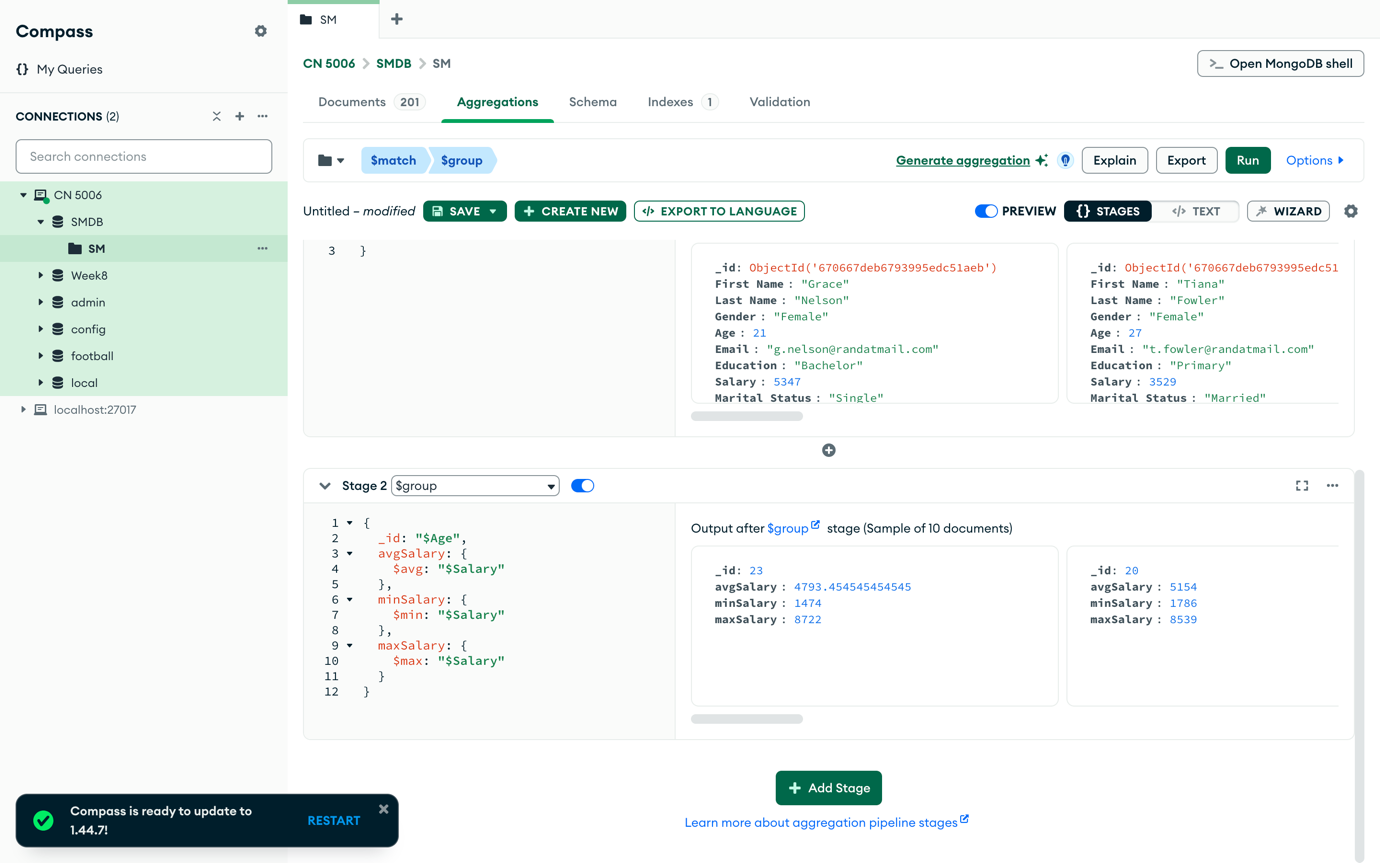
This query identifies individuals with a "Master's" level of education and groups them based on their marital status (e.g., married, single). For each marital status group, it calculates key statistics: the average, minimum, and maximum ages and salaries of the individuals. This information provides insights into how marital status correlates with age and salary trends among those with a Master's degree, enabling a demographic and financial overview of this segment.



Question 2

### Female Salary Analysis by Age Group

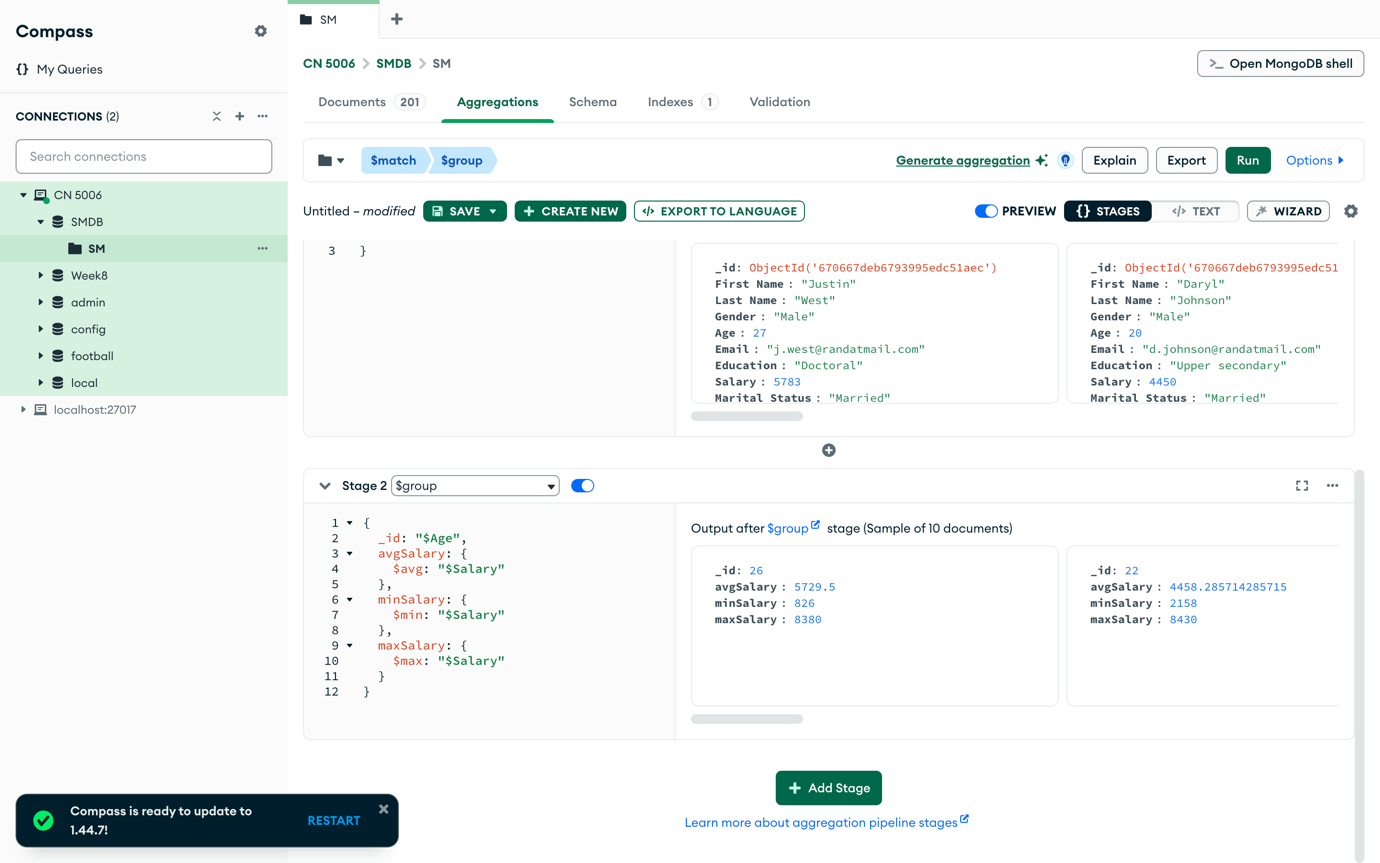
This query focuses on females, grouping them by their age to calculate the average, minimum, and maximum salaries within each age group. It helps identify how salaries vary across different age ranges for females, offering valuable insights into gender-specific earnings patterns. This analysis can be used to observe income growth or disparities over the lifespan of female professionals.



Question 3

### Male Salary Analysis by Age Group

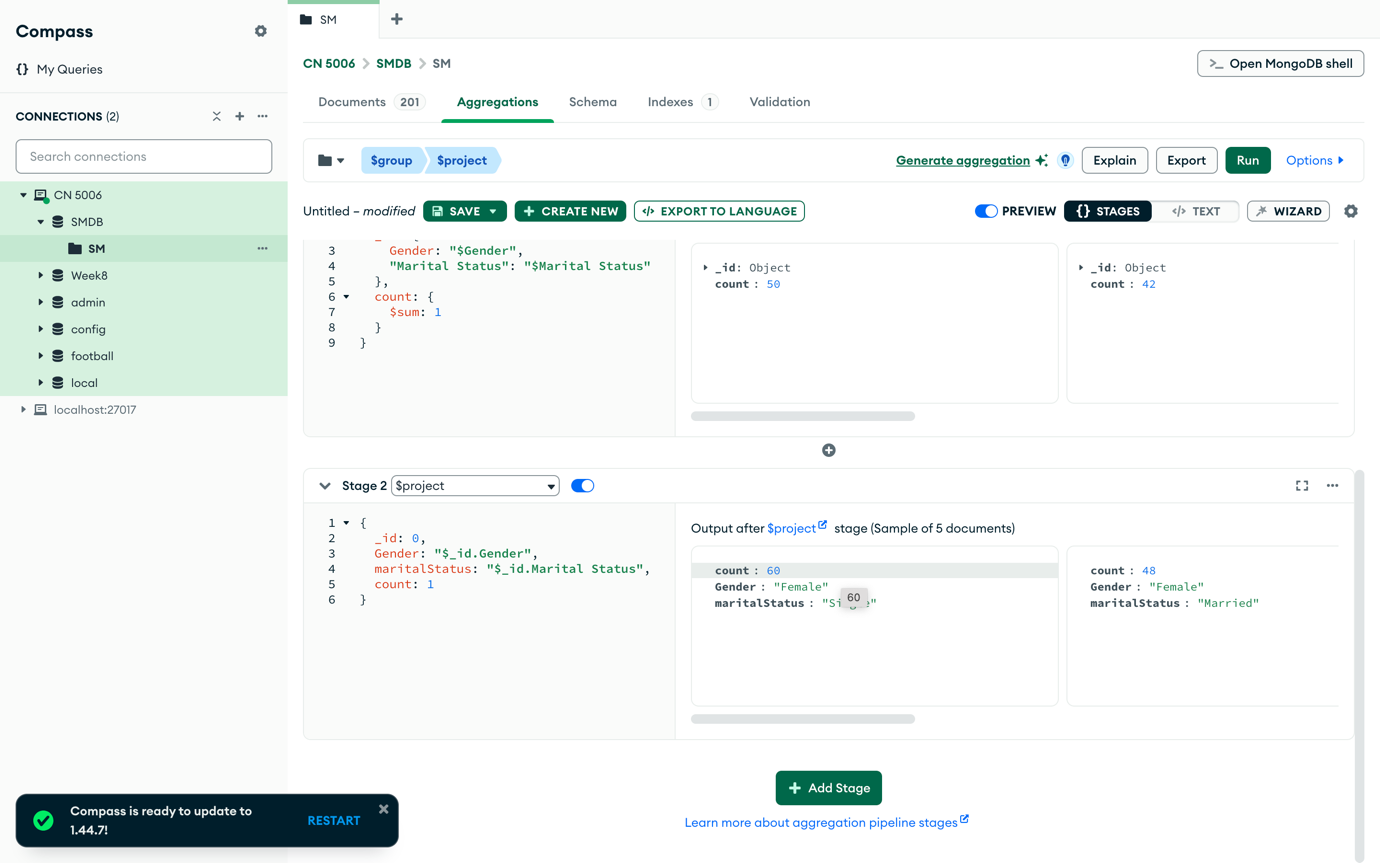
Similar to the query for females, this query evaluates salary data for males, grouping individuals by their age. For each age group, it calculates the average, minimum, and maximum salaries, highlighting earning trends for males as they age. This analysis provides comparative insights on how salary progression may differ between genders and age brackets, which could be useful for policy or organizational decision-making.



Question 4

### Count of Married and Unmarried Individuals by Gender

This query counts the number of married and unmarried individuals for both genders. By categorizing the data by gender and marital status, it provides a breakdown of how many males and females are married versus unmarried. This demographic snapshot is useful for understanding population trends and social dynamics, such as the proportion of single versus married individuals in different gender groups.



#### ****Reflection****

The lab provided hands-on experience with managing real-world datasets in MongoDB. It reinforced the importance of database design and operations for efficient data management. The exercise highlighted MongoDB’s strengths in handling diverse data types and performing complex queries with ease. Overall, the lab work deepened my understanding of how to use MongoDB effectively for both operational and analytical tasks