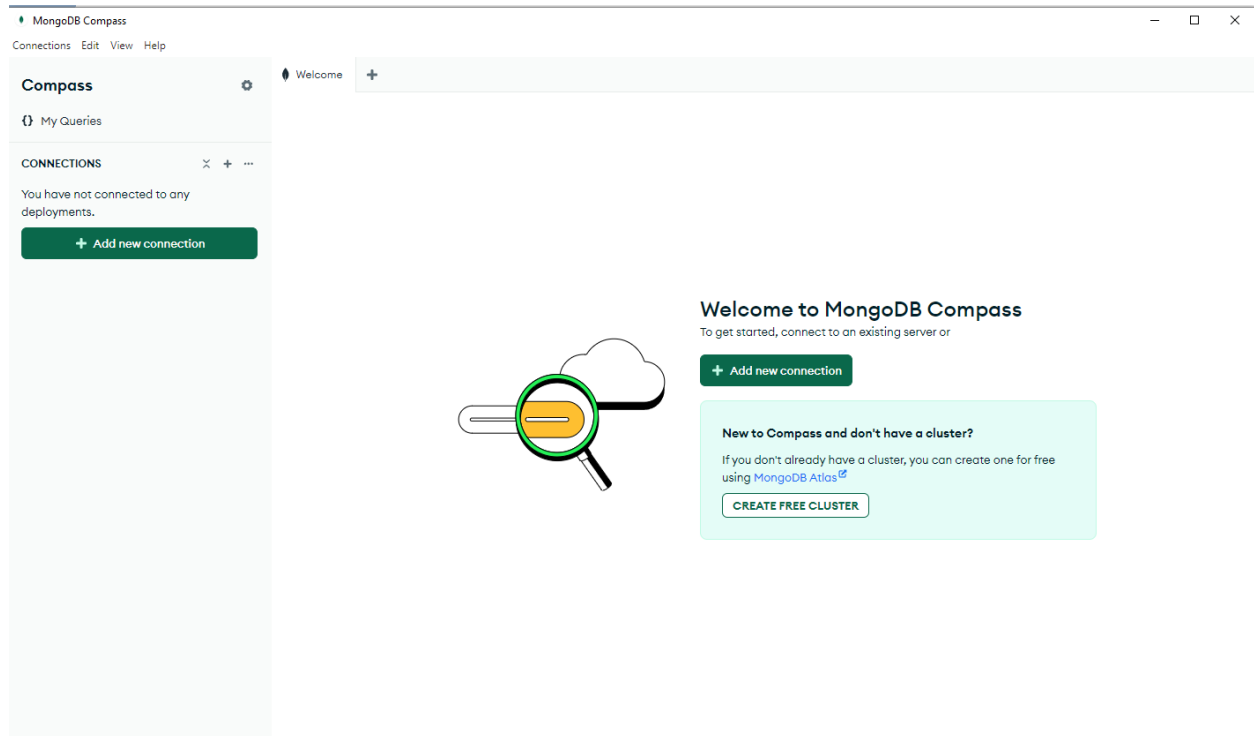
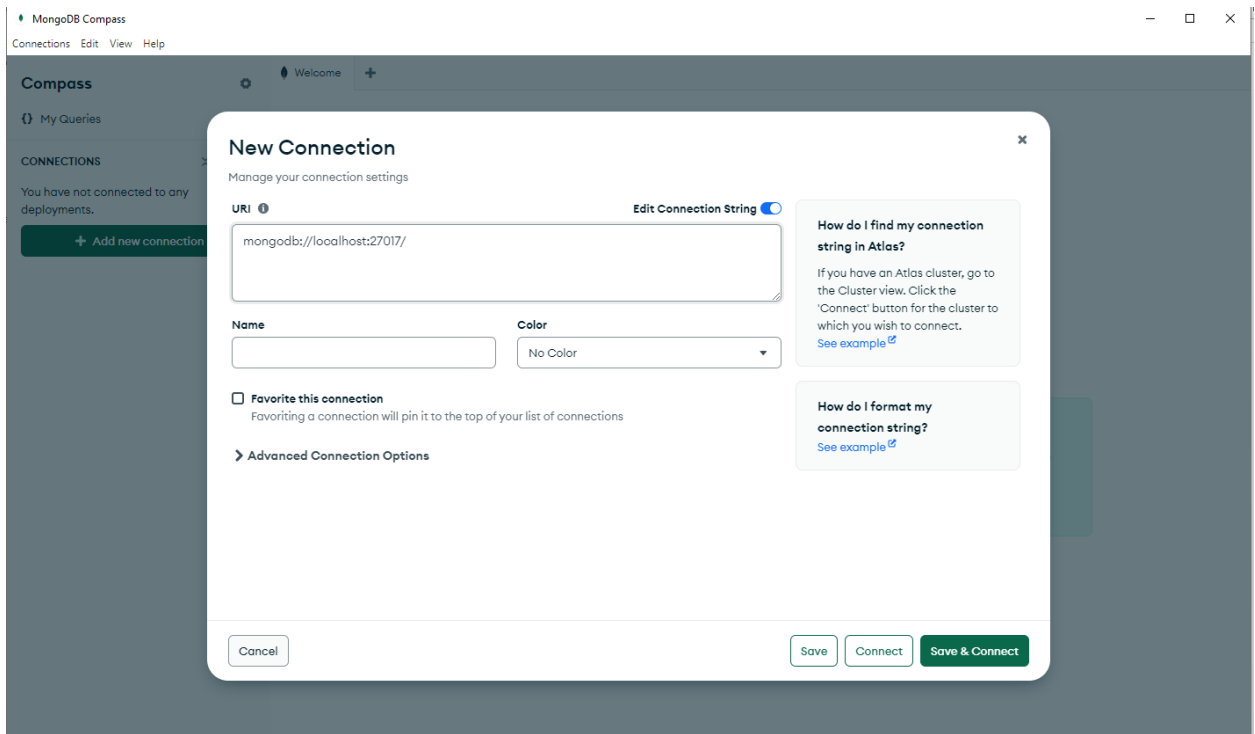


Week 2

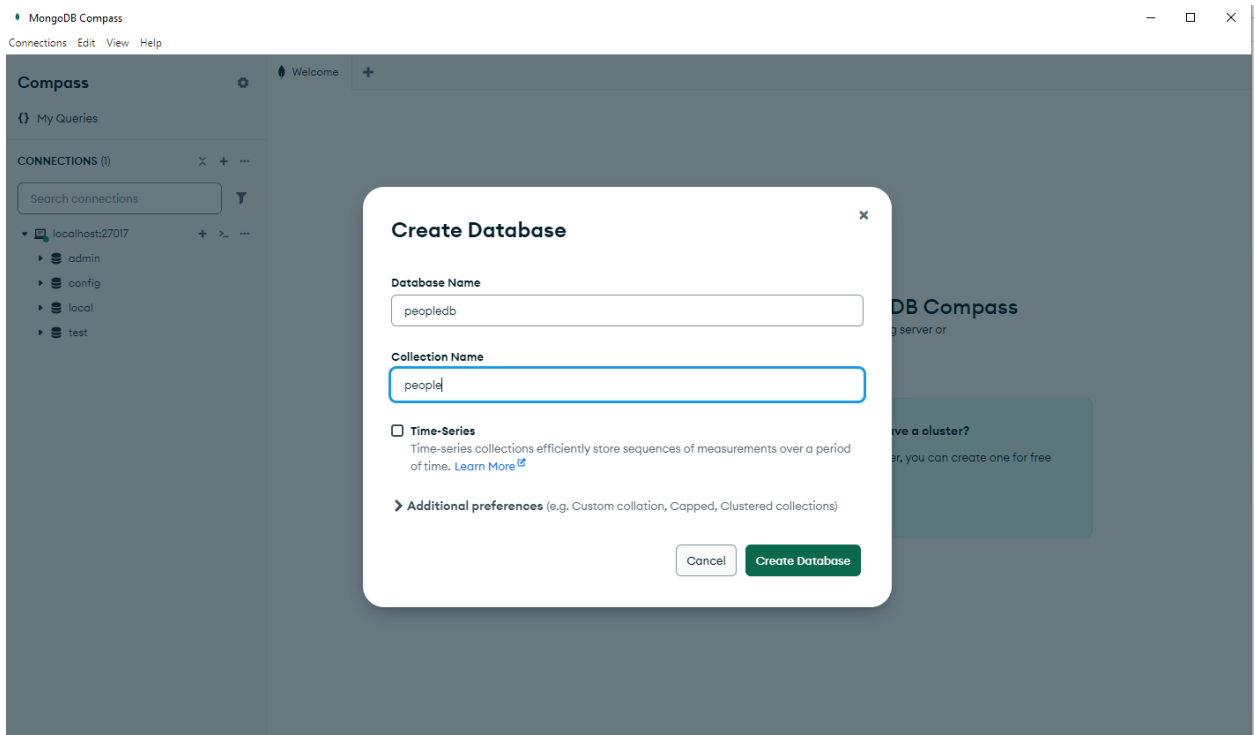
1. Downloaded MongoDB Compass and installed



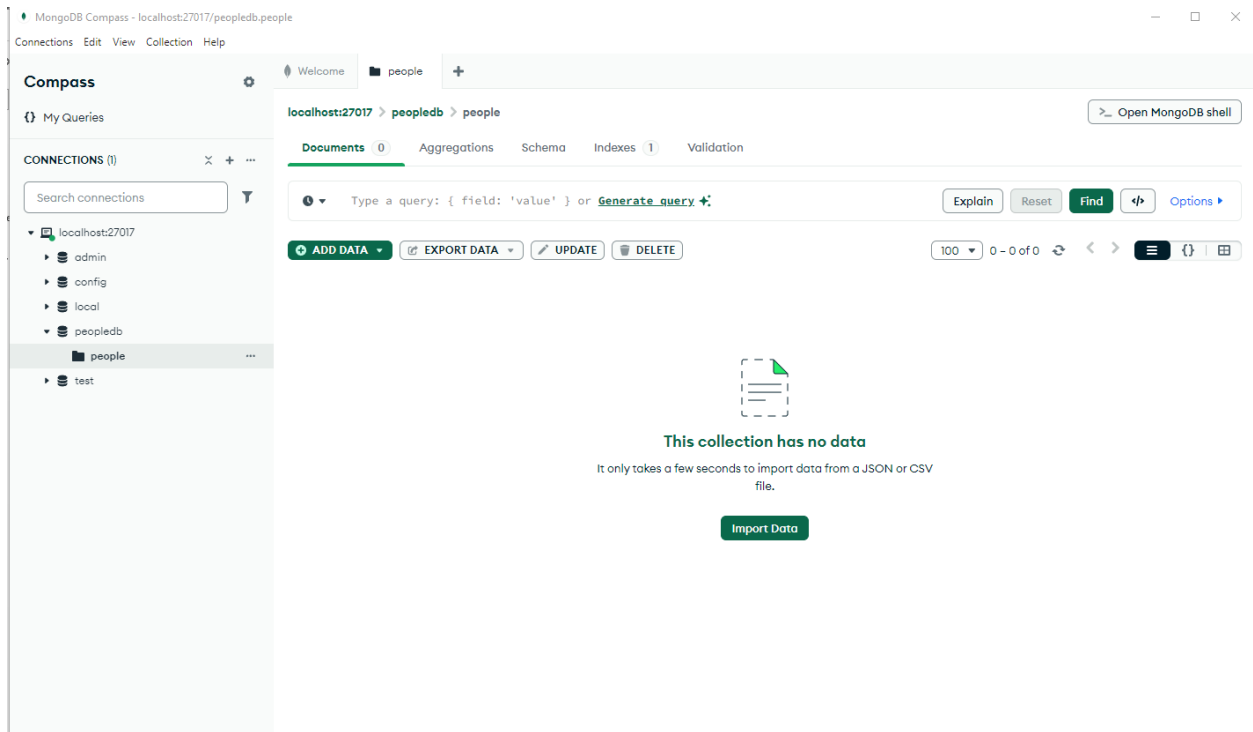
2.Created a connection with localhost



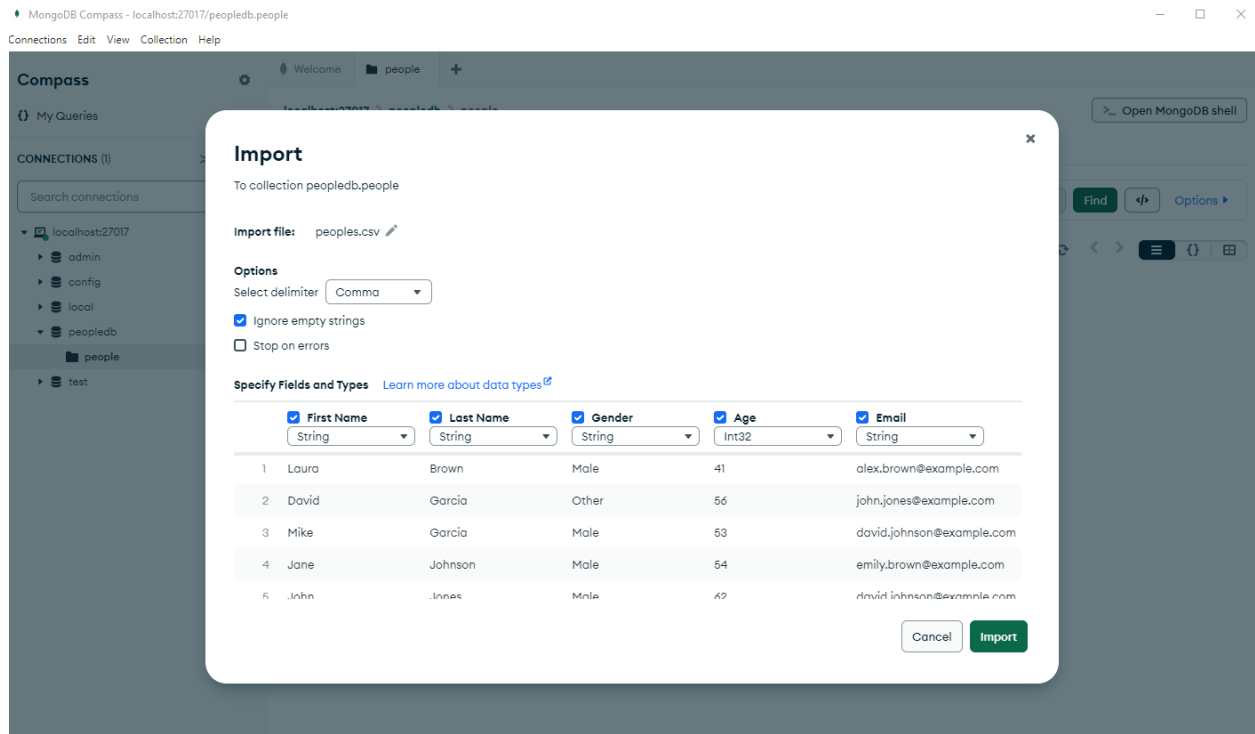
3. Created a database of **peopledb** with collection of **people**



4. Database and collection created successfully



5. Imported csv data into mongodb compass



Lab Task

1) Repeat the same process to search Education for Master and .Find the avg,min,max age and avg min max Salary of the people group by Marital status.

The screenshot shows the MongoDB Compass interface with an aggregation pipeline. Stage 1 is a \$match stage filtering for documents where education is 'Master' and age is 21. Stage 2 is a \$group stage grouping by marital status and calculating average, minimum, and maximum age and salary.

Stage 1: \$match

```
1 {
2   education: "Master",
3   age: { $gte: 21 }
4 }
```

Output after \$match stage (Sample of 3 documents)

Document 1	Document 2	Document 3
<pre>{ "_id": "ObjectId('6769a7385661ef6c679824a')", "first name": "Jane", "last name": "Smith", "gender": "Female", "age": 32, "email": "jane.smith@example.com", "education": "Master", "marital status": "Married", "salary": 67000 }</pre>	<pre>{ "_id": "ObjectId('6769a7385661ef6c679824d')", "first name": "Charlie", "last name": "Davis", "gender": "Male", "age": 36, "email": "charlie.davis@example.com", "education": "Master", "marital status": "Single", "salary": 62000 }</pre>	<pre>{ "_id": "ObjectId('6769a7385661ef6c6798251')", "first name": "Nina", "last name": "Thomas", "gender": "Female", "age": 27, "email": "nina.thomas@example.com", "education": "Master", "marital status": "Single", "salary": 57000 }</pre>

Stage 2: \$group

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$marital status",
7   AvgAge: { $avg: "$age" },
8   MinAge: { $min: "$age" },
9   MaxAge: { $max: "$age" },
10  AvgSalary: { $avg: "$salary" },
11  MinSalary: { $min: "$salary" },
12  MaxSalary: { $max: "$salary" }
13 }
```

Output after \$group stage (Sample of 2 documents)

Group 1: Married	Group 2: Single
<pre>{ "_id": "Married", "AvgAge": 32, "MinAge": 32, "MaxAge": 36, "AvgSalary": 67000, "MinSalary": 67000, "MaxSalary": 67000 }</pre>	<pre>{ "_id": "Single", "AvgAge": 31.5, "MinAge": 27, "MaxAge": 36, "AvgSalary": 59500, "MinSalary": 57000, "MaxSalary": 62000 }</pre>

2) find min,max average salary of each age group of female

The screenshot shows the MongoDB Compass interface with an aggregation pipeline. Stage 1 is a \$addFields stage adding a 'gender' field. Stage 2 is a \$group stage grouping by age and calculating average, minimum, and maximum salary for females.

Stage 1: \$addFields

```
1 { gender: "Female" }
```

Output after \$addFields stage (Sample of 10 documents)

Document 1	Document 2	Document 3
<pre>{ "_id": "ObjectId('6769a7385661ef6c679824a')", "first name": "John", "last name": "Doe", "gender": "Female", "age": 28, "email": "john.doe@example.com", "education": "Bachelor", "marital status": "Single", "salary": 55000 }</pre>	<pre>{ "_id": "ObjectId('6769a7385661ef6c679824b')", "first name": "Jane", "last name": "Smith", "gender": "Female", "age": 32, "email": "jane.smith@example.com", "education": "Master", "marital status": "Married", "salary": 67000 }</pre>	<pre>{ "_id": "ObjectId('6769a7385661ef6c679824c')", "first name": "Alice", "last name": "Johnson", "gender": "Female", "age": 24, "email": "alice.johnson@example.com", "education": "Bachelor", "marital status": "Single", "salary": 48000 }</pre>

Stage 2: \$group

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$age",
7   AvgSalary: { $avg: "$salary" },
8   MinSalary: { $min: "$salary" },
9   MaxSalary: { $max: "$salary" }
10 }
```

Output after \$group stage (Sample of 10 documents)

Age Group 1: 45	Age Group 2: 50	Age Group 3: 32
<pre>{ "_id": 45, "AvgSalary": 88000, "MinSalary": 88000, "MaxSalary": 88000 }</pre>	<pre>{ "_id": 50, "AvgSalary": 94000, "MinSalary": 94000, "MaxSalary": 94000 }</pre>	<pre>{ "_id": 32, "AvgSalary": 67000, "MinSalary": 67000, "MaxSalary": 67000 }</pre>

3) Find min, max, average salary of each age group of males

The screenshot shows the MongoDB Compass interface with an aggregation pipeline. Stage 1 is named '\$addFields' and Stage 2 is named '\$group'. The pipeline is as follows:

```
1 { gender: "Male" }
2
3
4
5
6
7
8
9
10
```

Output after \$addFields stage (Sample of 10 documents):

Document 1	Document 2	Document 3
<pre>{ "_id": ObjectId("6769a7385661f61c6798249"), "first Name": "John", "last name": "Doe", "gender": "Male", "age": 28, "email": "john.doe@example.com", "education": "Bachelor", "marital status": "Single", "salary": 55000 }</pre>	<pre>{ "_id": ObjectId("6769a7385661f61c679824a"), "first Name": "Jane", "last name": "Smith", "gender": "Male", "age": 32, "email": "jane.smith@example.com", "education": "Master", "marital status": "Married", "salary": 67000 }</pre>	<pre>{ "_id": ObjectId("6769a7385661f61c679824b"), "first Name": "Alice", "last name": "Johnson", "gender": "Male", "age": 24, "email": "alice.johnson@example.com", "education": "Bachelor", "marital status": "Single", "salary": 48000 }</pre>

Output after \$group stage (Sample of 10 documents):

Document 1	Document 2	Document 3
<pre>{ "_id": 24, "AvgSalary": 48000, "MinSalary": 48000, "MaxSalary": 48000 }</pre>	<pre>{ "_id": 36, "AvgSalary": 62000, "MinSalary": 62000, "MaxSalary": 62000 }</pre>	<pre>{ "_id": 28, "AvgSalary": 55000, "MinSalary": 55000, "MaxSalary": 55000 }</pre>

4) Count married and unmarried females and males.

The screenshot shows the MongoDB Compass interface with an aggregation pipeline. Stage 1 is named '\$group'. The pipeline is as follows:

```
1 {
2   _id: {
3     Gender: "$gender",
4     MaritalStatus: "$marital status"
5   },
6   Count: { $sum: 1 }
7 }
```

Output after \$group stage (Sample of 4 documents):

Document 1	Document 2	Document 3
<pre>{ "_id": Object, "Gender": "Female", "MaritalStatus": "Single", "Count": 3 }</pre>	<pre>{ "_id": Object, "Gender": "Female", "MaritalStatus": "Married", "Count": 2 }</pre>	<pre>{ "_id": Object, "Gender": "Male", "MaritalStatus": "Married", "Count": 3 }</pre>

Learn more about aggregation pipeline stages