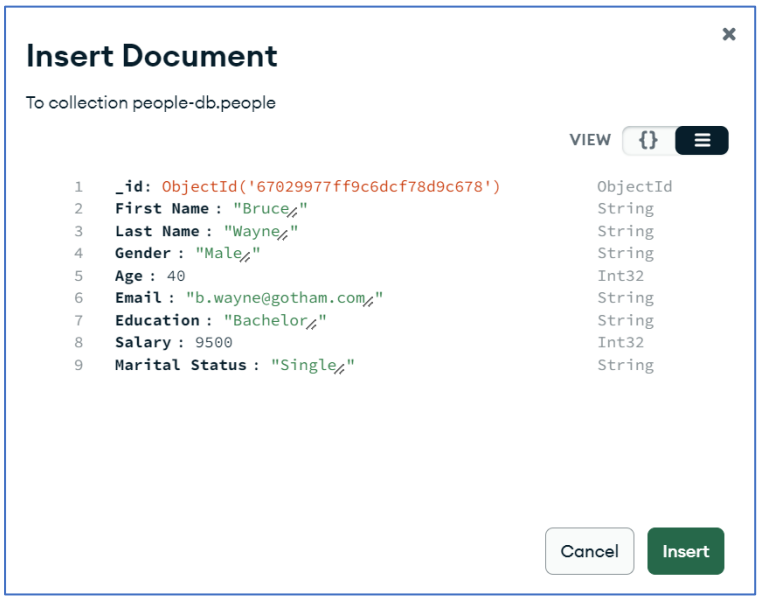
Week 2

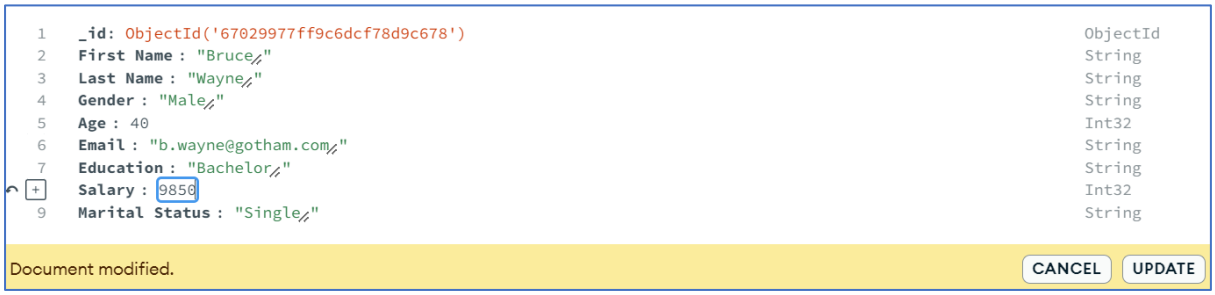
MongoDB Compass: Inserting a Document:



Explanation:

I used the MongoDB Compass GUI to insert a document. The list view also further simplified the process of adding new fields and values.

MongoDB Compass: Updating a Document:



Explanation:

I updated the salary field of one record.

MongoDB Compass: Deleting a Document:

```
_id: ObjectId('67029977ff9c6dcf78d9c678')
First Name : "Bruce"
Last Name : "Wayne"
Gender : "Male"
Age : 40
Email : "b.wayne@gotham.com"
Education : "Bachelor"
Salary : 9850
Marital Status : "Single"
```

Document flagged for deletion. CANCEL DELETE

Explanation:

I deleted one record from the database.

MongoDB Compass: Aggregate Pipe Line:

Stage 1 $match

```
/**
 * query: The query in MQL.
 */
{
  Education: "Bachelor",
  Age: {"$gte": 21}
}
```

Output after $match stage (Sample of 25 documents)

```
_id: ObjectId('67029971ff9c6dcf78d9c5b0')
First Name : "Grace"
Last Name : "Nelson"
Gender : "Female"
Age : 21
Email : "g.nelson@randatmail.com"
Education : "Bachelor"
Salary : 5347
Marital Status : "Single"
```

Stage 2 $group

```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
  _id:"$Gender",
  AvgAge: {$avg:"$Age"},
  MinAge: {$min:"$Age"},
  MaxAge: {$max:"$Age"},
  AvgSalary: {$avg:"$Salary"},
  MinSalary: {$min:"$Salary"},
  MaxSalary: {$max:"$Salary"}
}
```

Output after $group stage (Sample of 2 documents)

```
_id: "Female"
AvgAge : 25
MinAge : 21
MaxAge : 29
AvgSalary : 5020.846153846154
MinSalary : 509
MaxSalary : 8799
```

```
_id: "Male"
AvgAge : 25.6666666666666
MinAge : 22
MaxAge : 30
AvgSalary : 5252.41666666
MinSalary : 1260
MaxSalary : 9759
```

Explanation:

I performed the aggregate pipeline on the collection of records to produce the desired results. I now understand that the pipeline refers to the stages involved in producing the aggregation. The first stage required that I filter records based on bachelor education and an age of 21 or greater. The second stage required that I perform a series of functions to produce the averages, minimum values, and maximum values of age and salary within a gender grouping.

MongoDB Shell Task 1:

Output:

```
[
  {
    _id: 'Married',
    AvgAge: 25.4545454545454453,
    MinAge: 18,
    MaxAge: 30,
    AvgSalary: 4843.181818181818,
    MinSalary: 940,
    MaxSalary: 8483
  },
  {
    _id: 'Single',
    AvgAge: 25.333333333333332,
    MinAge: 18,
    MaxAge: 30,
    AvgSalary: 3783.8,
    MinSalary: 718,
    MaxSalary: 8722
  }
]
```

Explanation:

I first filtered the records based on a master education using the $match operation. Next I used $group operation to produce two documents from the values of marital status. I performed the various averages, minimum values, and maximum values functions on the values of age and salary.

MongoDB Shell Task 2:

Output:

```
[
  { _id: 18, AvgSalary: 5405.25, MinSalary: 2638, MaxSalary: 8631 },
  {
    _id: 19,
    AvgSalary: 4330.909090909091,
    MinSalary: 516,
    MaxSalary: 9846
  },
  { _id: 20, AvgSalary: 5154, MinSalary: 1786, MaxSalary: 8539 },
  {
    _id: 21,
    AvgSalary: 3576.777777777778,
    MinSalary: 901,
    MaxSalary: 5792
  },
  { _id: 22, AvgSalary: 5782, MinSalary: 3565, MaxSalary: 9925 },
  {
    _id: 23,
    AvgSalary: 4793.454545454545,
    MinSalary: 1474,
    MaxSalary: 8722
  },
  { _id: 24, AvgSalary: 4329, MinSalary: 2078, MaxSalary: 6921 },
  { _id: 25, AvgSalary: 5401, MinSalary: 707, MaxSalary: 9771 },
  { _id: 26, AvgSalary: 5319.5, MinSalary: 509, MaxSalary: 9611 },
  { _id: 27, AvgSalary: 4720, MinSalary: 1028, MaxSalary: 9319 },
  { _id: 28, AvgSalary: 5491, MinSalary: 646, MaxSalary: 9219 },
  {
    _id: 29,
    AvgSalary: 6169.222222222223,
    MinSalary: 2030,
    MaxSalary: 9913
  },
  {
    _id: 30,
    AvgSalary: 4623.363636363636,
    MinSalary: 619,
    MaxSalary: 8268
  }
]
```

Explanation:

I first filtered records by the female gender, next I grouped by age value and performed average, minimum, and maximum functions on salary, and lastly I sorted by age in ascending order.

MongoDB Shell Task 3:

Output:

```
[
  {
    _id: 18,
    AvgSalary: 4804.833333333333,
    MinSalary: 940,
    MaxSalary: 7677
  },
  { _id: 19, AvgSalary: 5469.75, MinSalary: 1221, MaxSalary: 9543 },
  {
    _id: 20,
    AvgSalary: 5309.333333333333,
    MinSalary: 1258,
    MaxSalary: 9587
  },
  { _id: 21, AvgSalary: 4426.25, MinSalary: 1810, MaxSalary: 9460 },
  { _id: 22, AvgSalary: 4026, MinSalary: 1000, MaxSalary: 8430 },
  {
    _id: 23,
    AvgSalary: 5848.166666666667,
    MinSalary: 1318,
    MaxSalary: 9854
  },
  { _id: 24, AvgSalary: 4412.4, MinSalary: 2033, MaxSalary: 8170 },
  { _id: 25, AvgSalary: 4326.5, MinSalary: 2032, MaxSalary: 7667 },
  { _id: 26, AvgSalary: 5729.5, MinSalary: 826, MaxSalary: 8380 },
  { _id: 27, AvgSalary: 5337.875, MinSalary: 1432, MaxSalary: 7548 },
  {
    _id: 28,
    AvgSalary: 5649.888888888889,
    MinSalary: 836,
    MaxSalary: 9989
  },
  { _id: 29, AvgSalary: 7562.5, MinSalary: 5226, MaxSalary: 9899 },
  {
    _id: 30,
    AvgSalary: 5363.090909090909,
    MinSalary: 1260,
    MaxSalary: 9989
  },
  { _id: 40, AvgSalary: 9999, MinSalary: 9999, MaxSalary: 9999 }
]
```

Explanation:

This task was the exact same as the last except I filtered by male instead of female.

MongoDB Shell Task 4:

Output:

```
[
  { _id: { Gender: 'Female', 'Marital Status': 'Single' }, Count: 60 },
  { _id: { Gender: 'Female', 'Marital Status': 'Married' }, Count: 48 },
  { _id: { Gender: 'Male', 'Marital Status': 'Single' }, Count: 43 },
  { _id: { Gender: 'Male', 'Marital Status': 'Married' }, Count: 51 }
]
```

Explanation:

I first grouped by two IDs of gender and marital status, followed by a count using the sum function. The result was four distinct outputs with a total.

Week 2 Reflection

This week gave me a basic understanding of non-SQL document-based databases. I gained some proficiency in MongoDB and it's respective software – MongoDB Compass and Mongo shell. I learned how data is structed within non-SQL using collections and key-value records, and how the data is in the form of JSON and is thus less constrained than relational data. I learned how to import, add, update, delete, find, and aggregate records using MongoDB Compass and shell. The shell required that I learn various commands to achieve the desired results.