```
Query 1
          '$match': {
            'Education': 'Master'
       }, {
          '$group': {
            ' id': '$Marital Status',
            'AvgAge': {
               '$avg': '$Age'
            },
             'MinAge': {
               '$min': '$Age'
             'MaxAge': {
               '$max': '$Age'
             'MinSalary': {
               '$min': '$Salary'
             'MaxSalary': {
               '$max': '$Salary'
         } }]
 Documents 200
                       Aggregations Schema Indexes 1
 ■ ▼ $match $group
                                                                                                     Generate aggregation → • • • Explain Export Run Options ▶
PREVIEW {} STAGES </P>

✓ Stage 1 ($match)

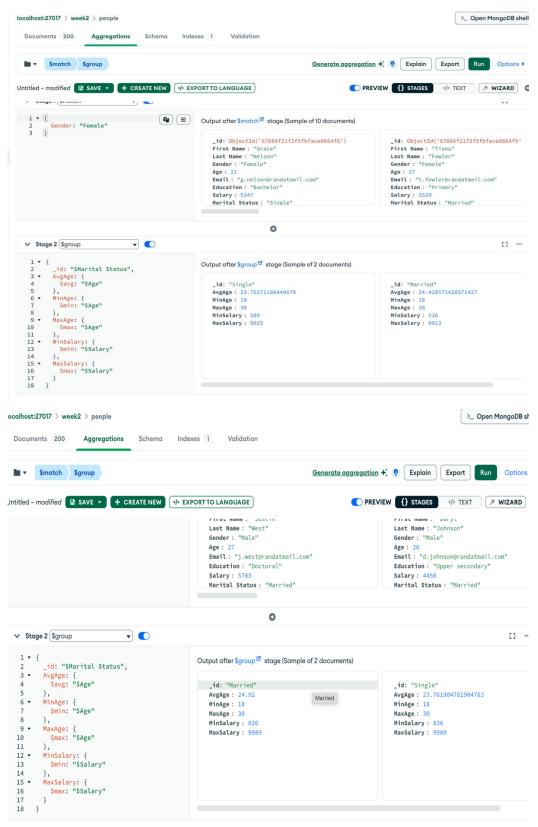
                                                Output after $match stage (Sample of 10 documents)
1 • {
2 Education: "Master|"
                                                                  _id: ObjectId('67066f21f2f5fbface066505')
First Name: "Evelyn"
Last Name: "Wells"
Gender: "Female"
Age: 24
Email: "e.wells@randatmail.com"
Education: "Master"
Salary: 2923
Marital Status: "Married"
                                                                                                                                _id: ObjectId('67066f21f2f5fbface066506'
                                                                                                                               _id: ObjectId('67066f21f2f5fbface060
First Name: "Martin"
Last Name: "Alexander"
Gender: "Male"
Age: 26
Email: "m.alexander@randatmail.com"
Education: "Master"
                                                                                                                                Salary: 2739
Marital Status: "Single"

✓ Stage 2 $\frac{1}{2}$ sgroup

                                                                                                                                                                      E3 ···
                                                               Output after $group stage (Sample of 2 documents)
          _id: "Single"
AvgAge: 25.428571428571427
MinAge: 18
MaxAge: 30
MinSalary: 718
MaxSalary: 8722
                                                                                                                                 id: "Married"
          },
MinAge: {
   $min: "$Age"
                                                                                                                                AvgAge: 25.636363636363637
MinAge: 18
MaxAge: 30
         },
MaxAge: {
   $max: "$Age"
                                                                                                                               MinSalary: 940
MaxSalary: 8483
          },
MinSalary: {
   $min: "$Salary"
       },
MaxSalary: {
   $max: "$Salary"
}
```

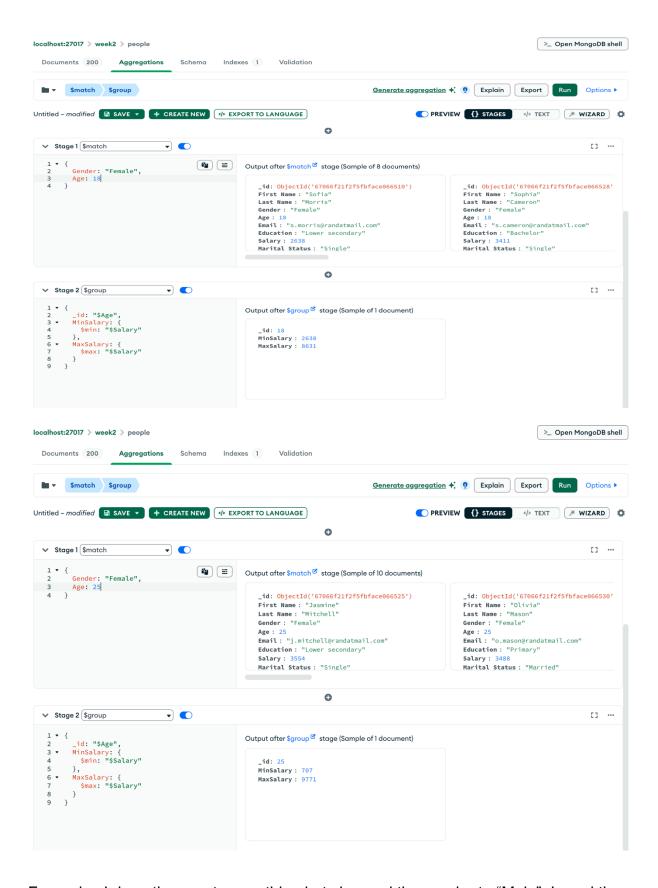
Query 2

I used the practise pipeline to gain the age range for the documents sample which was between 18-30 for females and males as you can see in the screenshots below.

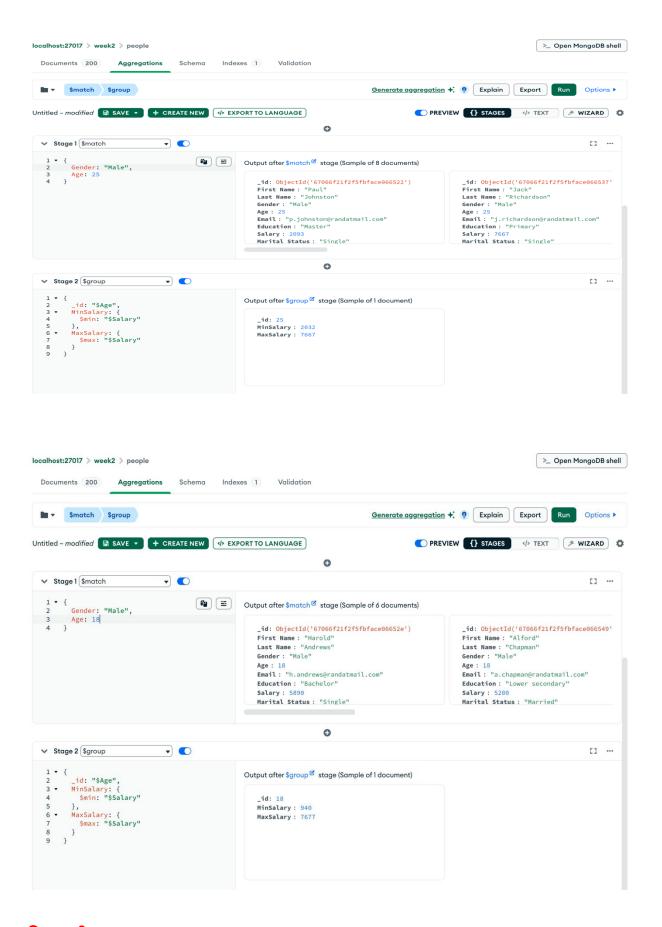


Next,

to find the minimum and maximum salary for each age group for females I simply wrote the code as shown below, editing the "Age" in the stage 1 "\$match". This allows me to toggle between all age groups. I have shown age groups 18 and 25 for examples.



For males I done the exact same thing but changed the gender to "Male". I used the same age ranges of 18 and 25 for my examples:



```
{
    '$match': {
        'Marital Status': 'Married'
    }
}, {
    '$group': {
        '_id': '$Gender',
        'fieldN': {
            '$sum': 1
        }
    }
}
```

As shown in the screenshots below, you can see that we use the \$match stage to find only the "Married" or the "Single" people. We then use the \$group stage in order to find the number of people of each gender who are married and unmarried

