

## 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 Validation

Match Group Generate aggregation Explain Export Run Options

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 1 \$match

```
1 {
2   Education: "Master"
3 }
```

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

<pre>{   "_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" }</pre>	<pre>{   "_id": ObjectId('67066f21f2f5fbface066506'),   "First Name": "Martin",   "Last Name": "Alexander",   "Gender": "Male",   "Age": 26,   "Email": "m.alexander@randatmail.com",   "Education": "Master",   "Salary": 2739,   "Marital Status": "Single" }</pre>
--	---

Stage 2 \$group

```
1 {
2   _id: "$Marital Status",
3   AvgAge: {
4     $avg: "$Age"
5   },
6   MinAge: {
7     $min: "$Age"
8   },
9   MaxAge: {
10    $max: "$Age"
11  },
12  MinSalary: {
13    $min: "$Salary"
14  },
15  MaxSalary: {
16    $max: "$Salary"
17 }
```

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

<pre>{   "_id": "Single",   "AvgAge": 25.428571428571427,   "MinAge": 18,   "MaxAge": 30,   "MinSalary": 718,   "MaxSalary": 8722 }</pre>	<pre>{   "_id": "Married",   "AvgAge": 25.636363636363637,   "MinAge": 18,   "MaxAge": 30,   "MinSalary": 940,   "MaxSalary": 8483 }</pre>
---	--

## Query 2

```
[
  {
    '$match': {
      'Gender': 'Female',
      'Age': 18
    }
  }, {
    '$group': {
      '_id': '$Age',
      'MinSalary': {
        '$min': '$Salary'
      },
      'MaxSalary': {
        '$max': '$Salary'
      }
    }
  }
]
```

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.

localhost:27017 > week2 > people Open MongoDB shell

Documents 200 Aggregations Schema Indexes 1 Validation

**\$match** **\$group** Generate aggregation Explain Export Run Options

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 1 **\$match**

```
1 {
2   Gender: "Female"
3 }
```

Output after **\$match** stage (Sample of 10 documents)

```
_id: ObjectId('67066f21f2f5fbface0664f6')
First Name: "Grace"
Last Name: "Nelson"
Gender: "Female"
Age: 21
Email: "g.nelson@randatmail.com"
Education: "Bachelor"
Salary: 5347
Marital Status: "Single"

_id: ObjectId('67066f21f2f5fbface0664f9')
First Name: "Tiana"
Last Name: "Fowler"
Gender: "Female"
Age: 27
Email: "t.fowler@randatmail.com"
Education: "Primary"
Salary: 3529
Marital Status: "Married"
```

Stage 2 **\$group**

```
1 {
2   _id: "$Marital Status",
3   AvgAge: {
4     $avg: "$Age"
5   },
6   MinAge: {
7     $min: "$Age"
8   },
9   MaxAge: {
10    $max: "$Age"
11  },
12  MinSalary: {
13    $min: "$Salary"
14  },
15  MaxSalary: {
16    $max: "$Salary"
17  }
18 }
```

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

```
_id: "Single"
AvgAge: 23.76271186440678
MinAge: 18
MaxAge: 30
MinSalary: 509
MaxSalary: 9925

_id: "Married"
AvgAge: 24.428571428571427
MinAge: 18
MaxAge: 30
MinSalary: 516
MaxSalary: 9913
```

localhost:27017 > week2 > people Open MongoDB shell

Documents 200 Aggregations Schema Indexes 1 Validation

**\$match** **\$group** Generate aggregation Explain Export Run Options

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 1 **\$match**

```
1 {
2   Gender: "Female"
3 }
```

Output after **\$match** stage (Sample of 10 documents)

```
First Name: "Justin"
Last Name: "West"
Gender: "Male"
Age: 27
Email: "j.west@randatmail.com"
Education: "Doctoral"
Salary: 5783
Marital Status: "Married"

First Name: "Daisy"
Last Name: "Johnson"
Gender: "Male"
Age: 20
Email: "d.johnson@randatmail.com"
Education: "Upper secondary"
Salary: 4450
Marital Status: "Married"
```

Stage 2 **\$group**

```
1 {
2   _id: "$Marital Status",
3   AvgAge: {
4     $avg: "$Age"
5   },
6   MinAge: {
7     $min: "$Age"
8   },
9   MaxAge: {
10    $max: "$Age"
11  },
12  MinSalary: {
13    $min: "$Salary"
14  },
15  MaxSalary: {
16    $max: "$Salary"
17  }
18 }
```

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

```
_id: "Married"
AvgAge: 24.92
MinAge: 18
MaxAge: 30
MinSalary: 826
MaxSalary: 9989

_id: "Single"
AvgAge: 23.761904761904763
MinAge: 18
MaxAge: 30
MinSalary: 836
MaxSalary: 9989
```

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.

localhost:27017 > week2 > people Open MongoDB shell

Documents 200 **Aggregations** Schema Indexes 1 Validation

**\$match** **\$group** Generate aggregation Explain Export Run Options

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 1 **\$match**

```
1 {
2   Gender: "Female",
3   Age: 18
4 }
```

Output after **\$match** stage (Sample of 8 documents)

```
_id: ObjectId('67066f21f2f5fbface066510')
First Name: "Sofia"
Last Name: "Morris"
Gender: "Female"
Age: 18
Email: "s.morris@randatmail.com"
Education: "Lower secondary"
Salary: 2638
Marital Status: "Single"
```

```
_id: ObjectId('67066f21f2f5fbface066528')
First Name: "Sophia"
Last Name: "Cameron"
Gender: "Female"
Age: 18
Email: "s.cameron@randatmail.com"
Education: "Bachelor"
Salary: 3411
Marital Status: "Single"
```

Stage 2 **\$group**

```
1 {
2   _id: "$Age",
3   MinSalary: {
4     $min: "$Salary"
5   },
6   MaxSalary: {
7     $max: "$Salary"
8   }
9 }
```

Output after **\$group** stage (Sample of 1 document)

```
_id: 18
MinSalary: 2638
MaxSalary: 8631
```

localhost:27017 > week2 > people Open MongoDB shell

Documents 200 **Aggregations** Schema Indexes 1 Validation

**\$match** **\$group** Generate aggregation Explain Export Run Options

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 1 **\$match**

```
1 {
2   Gender: "Female",
3   Age: 25
4 }
```

Output after **\$match** stage (Sample of 10 documents)

```
_id: ObjectId('67066f21f2f5fbface066525')
First Name: "Jasmine"
Last Name: "Mitchell"
Gender: "Female"
Age: 25
Email: "j.mitchell@randatmail.com"
Education: "Lower secondary"
Salary: 3554
Marital Status: "Single"
```

```
_id: ObjectId('67066f21f2f5fbface066530')
First Name: "Olivia"
Last Name: "Mason"
Gender: "Female"
Age: 25
Email: "o.mason@randatmail.com"
Education: "Primary"
Salary: 3488
Marital Status: "Married"
```

Stage 2 **\$group**

```
1 {
2   _id: "$Age",
3   MinSalary: {
4     $min: "$Salary"
5   },
6   MaxSalary: {
7     $max: "$Salary"
8   }
9 }
```

Output after **\$group** stage (Sample of 1 document)

```
_id: 25
MinSalary: 707
MaxSalary: 9771
```

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:

localhost:27017 > week2 > people Open MongoDB shell

Documents 200 Aggregations Schema Indexes 1 Validation

\$match \$group Generate aggregation Explain Export Run Options

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 1 \$match

```
1 {
2   Gender: "Male",
3   Age: 25
4 }
```

Output after \$match stage (Sample of 8 documents)

```
{
  "_id": ObjectId('67066f21f2f5fbface066522'),
  "First Name": "Paul",
  "Last Name": "Johnston",
  "Gender": "Male",
  "Age": 25,
  "Email": "p.johnston@randatmail.com",
  "Education": "Master",
  "Salary": 2093,
  "Marital Status": "Single"
}
```

```
{
  "_id": ObjectId('67066f21f2f5fbface066537'),
  "First Name": "Jack",
  "Last Name": "Richardson",
  "Gender": "Male",
  "Age": 25,
  "Email": "j.richardson@randatmail.com",
  "Education": "Primary",
  "Salary": 7667,
  "Marital Status": "Single"
}
```

Stage 2 \$group

```
1 {
2   _id: "$Age",
3   MinSalary: {
4     $min: "$Salary"
5   },
6   MaxSalary: {
7     $max: "$Salary"
8   }
9 }
```

Output after \$group stage (Sample of 1 document)

```
{
  "_id": 25,
  "MinSalary": 2032,
  "MaxSalary": 7667
}
```

localhost:27017 > week2 > people Open MongoDB shell

Documents 200 Aggregations Schema Indexes 1 Validation

\$match \$group Generate aggregation Explain Export Run Options

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 1 \$match

```
1 {
2   Gender: "Male",
3   Age: 18
4 }
```

Output after \$match stage (Sample of 6 documents)

```
{
  "_id": ObjectId('67066f21f2f5fbface06652e'),
  "First Name": "Harold",
  "Last Name": "Andrews",
  "Gender": "Male",
  "Age": 18,
  "Email": "h.andrews@randatmail.com",
  "Education": "Bachelor",
  "Salary": 5890,
  "Marital Status": "Single"
}
```

```
{
  "_id": ObjectId('67066f21f2f5fbface066549'),
  "First Name": "Alford",
  "Last Name": "Chapman",
  "Gender": "Male",
  "Age": 18,
  "Email": "a.chapman@randatmail.com",
  "Education": "Lower secondary",
  "Salary": 5200,
  "Marital Status": "Married"
}
```

Stage 2 \$group

```
1 {
2   _id: "$Age",
3   MinSalary: {
4     $min: "$Salary"
5   },
6   MaxSalary: {
7     $max: "$Salary"
8   }
9 }
```

Output after \$group stage (Sample of 1 document)

```
{
  "_id": 18,
  "MinSalary": 940,
  "MaxSalary": 7677
}
```

## Query 3

```
[
  {
    '$match': {
      'Gender': 'Male',
      'Age': 18
    }
  }, {
    '$group': {
      '_id': '$Age',
      'MinSalary': {
        '$min': '$Salary'
      },
      'MaxSalary': {
        '$max': '$Salary'
      }
    }
  }
]
```

#### Query 4

```
[
```

```
{
  '$match': {
    'Marital Status': 'Married'
  }
}, {
  '$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

My Queries

people

+

localhost:27017 > week2 > people

Documents 200

Aggregations

Schema

Indexes 1

Validation

\$match

\$group

\$group

Generate aggregation

Explain

Export

Run

Options

Counts of mar... - modified

SAVE

CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

WIZARD

Stage 1 \$match

```

1 {
2   "Marital Status": "Married",
3 }

```

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

```

_id: ObjectId('67066f21f2f5fbface0664f7')
First Name: "Justin"
Last Name: "West"
Gender: "Male"
Age: 27
Email: "j.west@randatmail.com"
Education: "Doctoral"
Salary: 5783
Marital Status: "Married"

```

```

_id: ObjectId('67066f21f2f5fbface0664f8')
First Name: "Daryl"
Last Name: "Johnson"
Gender: "Male"
Age: 20
Email: "d.johnson@randatmail.com"
Education: "Upper secondary"
Salary: 4450
Marital Status: "Married"

```

Stage 2 \$group

```

1 {
2   _id: "$Gender",
3   fieldN: {
4     $sum: 1
5   }
6 }
7
8 }

```

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

```

_id: "Male"
fieldN: 50

```

```

_id: "Female"
fieldN: 49

```

My Queries

people

+

localhost:27017 > week2 > people

Documents 200

Aggregations

Schema

Indexes 1

Validation

\$match

\$group

\$group

Generate aggregation

Explain

Export

Run

Options

Counts of mar... - modified

SAVE

CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

WIZARD

Stage 1 \$match

```

1 {
2   "Marital Status": "Single",
3 }

```

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

```

_id: ObjectId('67066f21f2f5fbface0664f6')
First Name: "Grace"
Last Name: "Nelson"
Gender: "Female"
Age: 21
Email: "g.nelson@randatmail.com"
Education: "Bachelor"
Salary: 5347
Marital Status: "Single"

```

```

_id: ObjectId('67066f21f2f5fbface0664fc')
First Name: "Charlie"
Last Name: "Perkins"
Gender: "Male"
Age: 28
Email: "c.perkins@randatmail.com"
Education: "Bachelor"
Salary: 3586
Marital Status: "Single"

```

Stage 2 \$group

```

1 {
2   _id: "$Gender",
3   fieldN: {
4     $sum: 1
5   }
6 }
7
8 }

```

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

```

_id: "Female"
fieldN: 59

```

```

_id: "Male"
fieldN: 42

```