

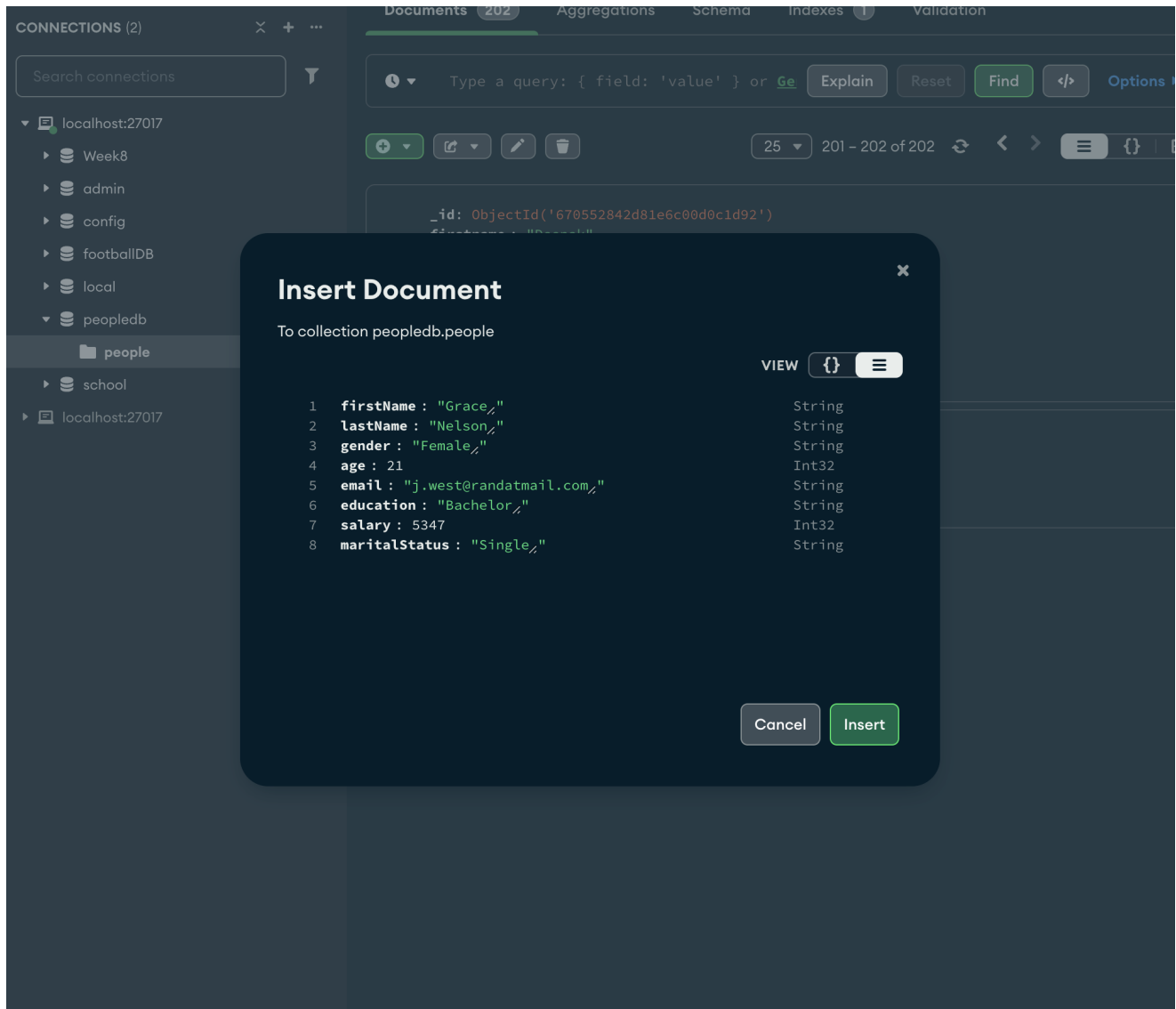
# Week 2 Portfolio

Name: Deepak Kumar

Student ID: 2284279

## 1. Screenshots

### a. Inserting a document in the collection:



### b. Updating the document:

```
_id: ObjectId('670552842d81e6c00d0c1d92')
firstname: "Deepak"
lastname: "Kumar"
gender: "male"
age: 21
email: "deepak21cb@gmail.com"
education: "Bacehlor"
salary: 388383
maritalstatus: "single"
```

```
_id: ObjectId('6743c211403674223d00beaf')
firstName: "Grace"
lastName: "Nelson"
gender: ""
```

1	_id: ObjectId('6743c3fc74bc20ded5bdf281')	ObjectId
2	firstName: "Deepak"	String
3	lastName: "Kumar"	String
4	gender: "Female"	String
5	age: 21	Int32
6	email: "j.west@randatmail.com"	String
7	education: "Bachelor"	String
8	salary: 5347	Int32
9	maritalStatus: "Single"	String

Document modified.

CANCEL

UPDATE

## c. Deleting the documents:

```
_id: ObjectId('670552842d81e6c00d0c1d92')
firstname: "Deepak"
lastname: "Kumar"
gender: "male"
age: 21
email: "deepak21cb@gmail.com"
education: "Bacehlor"
salary: 388383
maritalstatus: "single"
```

```
_id: ObjectId('6743c211403674223d00beaf')
firstName: "Grace"
lastName: "Nelson"
gender: ""
```

Remove document



```
_id: ObjectId('6743c3fc74bc20ded5bdf281')
firstName: "Deepak"
lastName: "Kumar"
gender: "Female"
age: 21
email: "j.west@randatmail.com"
education: "Bachelor"
salary: 5347
maritalStatus: "Single"
```



## d. Aggregation:

Aggregation to filter the data using \$match and \$group key stages.

\$match: It filters documents to pass only those that match specified criteria.

\$group: It groups documents by a specified field (or fields).

The screenshot shows the MongoDB Compass interface for a database named 'people' in a 'peopledb' collection. The 'Aggregations' tab is active, showing a pipeline with two stages: '\$match' and '\$group'.

**Stage 1: \$match**

- Query: 

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   Education: "Bachelor",
6   Age: {$gte: 21}
7 }
```
- Output: A sample of 10 documents is shown, including details like \_id, First Name, Last Name, Gender, Age, Email, Education, Salary, and Marital Status.

**Stage 2: \$group**

- Query: 

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$Gender",
7   Avg: {$avg: "$Age"},
8   MinAge: {$min: "$Age"},
9   MaxAge: {$max: "$Age"}
10 }
```
- Output: A sample of 2 documents is shown, grouped by Gender. Each document shows the average, minimum, and maximum age for that gender.

Buttons for 'Generate aggregation', 'Explain', 'Export', 'Run', and 'Options' are visible at the top. A 'Preview' button and 'STAGES' tab are also present. A '+ Add Stage' button is at the bottom.

MongoDB queries for the following using either command shell:

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.

Tell Compass what aggregation to build (e.g. how many movies were made each year) Generate

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

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

**Stage 1** **\$match** Output after \$match stage (Sample of 10 documents)

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

```
{
  "_id": ObjectId("6705513e2d81e6c00d0c1cd8"),
  "First Name": "Evelyn",
  "Last Name": "Wells",
  "Gender": "Female",
  "Age": 24,
  "Email": "e.wells@randatmail.com",
  "Education": "Master",
  "Salary": 2923,
  "Marital Status": "Single"
}
```

```
{
  "_id": ObjectId("6705513e2d81e6c00d0c1cd9"),
  "First Name": "Martin",
  "Last Name": "Alexander",
  "Gender": "Male",
  "Age": 26,
  "Email": "m.alexander@randatmail.com",
  "Education": "Master",
  "Salary": 2739,
  "Marital Status": "Single"
}
```

```
{
  "_id": ObjectId("6705513e2d81e6c00d0c1cd9"),
  "First Name": "P.",
  "Last Name": "Jol",
  "Gender": "Male",
  "Age": 25,
  "Email": "p.john",
  "Education": "Ma",
  "Salary": 2693,
  "Marital Status": "Single"
}
```

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

```
1 {
2   _id: "$Marital Status",
3   avgAge: {
4     $avg: "$Age"
5   },
6   minAge: {
7     $min: "$Age"
8   },
9   maxAge: {
10    $max: "$Age"
11  },
12  avgSalary: {
13    $avg: "$Salary"
14  },
15  minSalary: {
16    $min: "$Salary"
17  },
18  maxSalary: {
19    $max: "$Salary"
20  }
21 }
```

```
{
  "_id": "Single",
  "avgAge": 25.333333333333332,
  "minAge": 18,
  "maxAge": 30,
  "avgSalary": 3783.8,
  "minSalary": 718,
  "maxSalary": 8722
}
```

```
{
  "_id": "Married",
  "avgAge": 25.8,
  "minAge": 18,
  "maxAge": 30,
  "avgSalary": 5227.5,
  "minSalary": 940,
  "maxSalary": 8483
}
```

2. Find min,max average salary of each age group of female.

Tell Compass what aggregation to build (e.g. how many movies were made each year) Generate

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

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

```
Last Name: "Nelson"
Gender: "Female"
Age: 21
Email: "g.nelson@randatmail.com"
Education: "Bachelor"
Salary: 5347
```

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

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

```
Last Name: "Fow"
Gender: "Male"
Age: 27
Email: "t.fowle"
Education: "Pri"
Salary: 3529
```

**Stage 1** **\$match** Output after \$match stage (Sample of 10 documents)

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

```
{
  "_id": ObjectId("6705513e2d81e6c00d0c1cc9"),
  "First Name": "Grace",
  "Last Name": "Nelson",
  "Gender": "Female",
  "Age": 21,
  "Email": "g.nelson@randatmail.com",
  "Education": "Bachelor",
  "Salary": 5347,
  "Marital Status": "Single"
}
```

```
{
  "_id": ObjectId("6705513e2d81e6c00d0c1ccc"),
  "First Name": "Tiana",
  "Last Name": "Fowler",
  "Gender": "Female",
  "Age": 27,
  "Email": "t.fowler@randatmail.com",
  "Education": "Primary",
  "Salary": 3529,
  "Marital Status": "Married"
}
```

```
{
  "_id": ObjectId("6705513e2d81e6c00d0c1ccc"),
  "First Name": "K",
  "Last Name": "Al",
  "Gender": "Female",
  "Age": 21,
  "Email": "k.allen",
  "Education": "Low",
  "Salary": 5792,
  "Marital Status": "Single"
}
```

**Stage 2** **\$group** Output after \$group stage (Sample of 10 documents)

```
1 {
2   _id: "$Age",
3   minSalary: {
4     $min: "$Salary"
5   },
6   maxSalary: {
7     $max: "$Salary"
8   },
9   avgSalary: {
10    $avg: "$Salary"
11  }
12 }
```

```
{
  "_id": 24,
  "minSalary": 2078,
  "maxSalary": 6921,
  "avgSalary": 4329
}
```

```
{
  "_id": 21,
  "minSalary": 901,
  "maxSalary": 5792,
  "avgSalary": 3576.7777777777778
}
```

```
{
  "_id": 25,
  "minSalary": 707,
  "maxSalary": 977,
  "avgSalary": 540
}
```

3. Find min,max average salary of each age group of male.

Tell Compass what aggregation to build (e.g. how many movies were made each year) Generate

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

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

**Stage 1** **\$match** Output after \$match stage (Sample of 10 documents)

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

```
_id: ObjectId('6705513e2d81e6c00d0c1cca')
First Name: "Justin"
Last Name: "West"
Gender: "Male"
Age: 27
Email: "j.west@randatmail.com"
Education: "Doctoral"
Salary: 5783
Marital Status: "Married"
```

```
_id: ObjectId('6705513e2d81e6c00d0c1ccb')
First Name: "Daryl"
Last Name: "Johnson"
Gender: "Male"
Age: 20
Email: "d.johnson@randatmail.com"
Education: "Upper secondary"
Salary: 4450
Marital Status: "Married"
```

```
_id: ObjectId('6705513e2d81e6c00d0c1ccb')
First Name: "A"
Last Name: "Bai"
Gender: "Male"
Age: 26
Email: "a.bai@randatmail.com"
Education: "Upper secondary"
Salary: 6332
Marital Status: "Married"
```

**Stage 2** **\$group** Output after \$group stage (Sample of 10 documents)

```
1 {
2   _id: "$Age",
3   minSalary: {
4     $min: "$Salary"
5   },
6   maxSalary: {
7     $max: "$Salary"
8   },
9   avgSalary: {
10    $avg: "$Salary"
11  }
12 }
```

```
_id: 24
minSalary: 2033
maxSalary: 8170
avgSalary: 4412.4
```

```
_id: 21
minSalary: 1810
maxSalary: 9460
avgSalary: 4426.25
```

```
_id: 25
minSalary: 2033
maxSalary: 7660
avgSalary: 4320
```

+ Add Stage [Learn more about aggregation pipeline stages](#)

#### 4. Count married and unmarried females and males.

Tell Compass what aggregation to build (e.g. how many movies were made each year) Generate

**\$group** Explain Export Run Options

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

**203 Documents in the collection**

Preview of documents

```
_id: ObjectId('6705513e2d81e6c00d0c1cc9')
First Name: "Grace"
Last Name: "Nelson"
Gender: "Female"
Age: 21
Email: "g.nelson@randatmail.com"
Education: "Bachelor"
Salary: 5347
```

```
_id: ObjectId('6705513e2d81e6c00d0c1cca')
First Name: "Justin"
Last Name: "West"
Gender: "Male"
Age: 27
Email: "j.west@randatmail.com"
Education: "Doctoral"
Salary: 5783
```

```
_id: ObjectId('6705513e2d81e6c00d0c1ccb')
First Name: "Daryl"
Last Name: "Johnson"
Gender: "Male"
Age: 20
Email: "d.johnson@randatmail.com"
Education: "Upper secondary"
Salary: 4450
```

```
_id: ObjectId('6705513e2d81e6c00d0c1ccb')
First Name: "Tia"
Last Name: "Fowler"
Gender: "Female"
Age: 27
Email: "t.fowler@randatmail.com"
Education: "Primary"
Salary: 3529
```

**Stage 1** **\$group** Output after \$group stage (Sample of 5 documents)

```
1 {
2   _id: {
3     Gender: "$Gender",
4     "Marital Status": "$Marital Status"
5   },
6   Count: {
7     $sum: 1
8   }
9 }
```

```
_id: Object
  Gender: "Female"
  Marital Status: "Single"
  Count: 60
```

```
_id: Object
  Count: 3
```

```
_id: Object
  Gender: "Male"
  Marital Status: "Married"
  Count: 50
```

+ Add Stage [Learn more about aggregation pipeline stages](#)

## 2. Reflection

In the week 2 lab session, I learned how to install and work with MongoDB Compass. The first step was to create a database in there and upload a CSV file into a collection. This taught me how MongoDB handles data differently from relational databases such as SQL (which i am familiar with from my other module).

I learned inserting, updating, and deleting documents— which improved my understanding of CRUD operations. In aggregation pipeline, which I learned how to filter and group data to calculate average, minimum, and maximum values based on various criteria. Overall, this lab introduced me to how MongoDB offers dynamic handling of unstructured data.