



CN5005



MODULE: WEB AND MOBILE APPLICATION DEVELOPMENT (CN5006)

INSTRUCTOR: DR. NADEEM QAZI

NAME: SELVAKUMARAN THAMILNILA

STUDENT ID: U2835683

DATE: 6 NOVEMBER 2025

Week 2 Reflection:

In this lab, I learned how to manage and manipulate data using MongoDB Compass. Firstly, I created a database named PeopleDB and a collection called people, then imported the provided Peoples.csv file. This introduced me to MongoDB's document-oriented structure, where collections take the place of tables and documents are individual records. I practiced CRUD operations: inserting, updating, and deleting documents, which reinforced my understanding of document fields, types, and the system-generated _id field.

A significant portion of the lab was on aggregation pipelines. I had used the \$match stage to filter documents, such as selecting people with a Bachelor's degree over 21 years old. Then, I wanted to apply the \$group stage to calculate aggregated results - Average, minimum, and maximum of age and salary, grouped by gender or marital status. These exercises allowed me to understand how MongoDB can process and summarize large datasets efficiently, similar to SQL aggregate functions but with more flexibility for varied data structures.

This lab strengthened my practical skills in database management, query formulation, and aggregation operations. Now, I can really appreciate the flexibility of MongoDB while handling dynamic and complex data sets, which is really demanded in modern applications for both web and mobile development.

The screenshot displays the MongoDB Compass interface with two separate connections.

Local MongoDB Connection:

- Connections:** Local MongoDB, admin, config, local, startup_log.
- Startup Log Collection:** CAPPED, Storage size: 24.58 kB, Documents: 10, Avg. document size: 3.71 kB, Indexes: 1, Total index size: 36.86 kB.

Week2labConnection Connection:

- Connections:** Local MongoDB, Week2labConnection, PeopleDB, People.
- People Collection:** Documents (202), Aggregations, Schema, Indexes (1), Validation.
- Query Bar:** Type a query: { field: 'value' } Explain, Reset, Find, Options.
- Result Preview:** Two documents shown.

Document 1: _id: ObjectId('6913ad3ed474025e519fd9d1'), First Name: ""

Document 2: _id: ObjectId('6913adb4d474025e519fd9d2'), First Name: "Nadeem", age: "25", gender: "Male", education: "Bachelor", salary: "40000", martialstatus: "Single"

```

1  _id: ObjectId('6913adb4d474025e519fd9d2')
2  First Name : "Nadeem"
3  age : "25"
4  gender : "Male"
5  education : "Bachelor"
6  salary : "[446765]"
7  maritalstatus : "Single"

```

Document modified.

CANCEL UPDATE

```

_id: ObjectId('6913ad21d474025e519fd986')
firstName : "Ned"
lastName : "Ellis"
gender : "Male"
age : 20
email : "n.ellis@randatmail.com"
education : "Upper secondary"
salary : 1258
maritalstatus : "Single"

```

Document flagged for deletion.

CANCEL DELETE

Stage 1 \$match

```

1 /**
2  * query: The query in MQL.
3  */
4 {
5   education: "Bachelor",
6   age: { "$gte": 21 }
7 }

```

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

<code>_id: ObjectId('6913ad21d474025e519fd909')</code>	<code>_id: ObjectId('6913ad21d474025e519fd90f')</code>
<code>firstName: "Grace"</code>	<code>firstName: "Charlie"</code>
<code>lastName: "Nelson"</code>	<code>lastName: "Perkins"</code>
<code>gender: "Female"</code>	<code>gender: "Male"</code>
<code>age: 21</code>	<code>age: 28</code>
<code>email: "g.nelson@randatmail.com"</code>	<code>email: "c.perkins@randatmail.com"</code>
<code>education: "Bachelor"</code>	<code>education: "Bachelor"</code>
<code>salary: 5347</code>	<code>salary: 3586</code>
<code>maritalstatus: "Single"</code>	<code>maritalstatus: "Single"</code>

Stage 2 \$group

```

1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$gender",
7   AvgAge: [
8     $avg: "$age"
9   ]
10 }
11

```

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

<code>_id: "Female"</code>	<code>_id: "Male"</code>
<code>AvgAge: 25</code>	<code>AvgAge: 25.666666666666666</code>

[+ Add stage](#)

Stage 2 \$group

```

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}
11
12

```

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

<code>_id: "Female"</code> <code>Avg : 25</code> <code>MinAge : 21</code> <code>MaxAge : 29</code>	<code>_id: "Male"</code> <code>Avg : 25.666666666666668</code> <code>MinAge : 22</code> <code>MaxAge : 30</code>
---	---

Stage 2 \$group

```

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  MaxSalary: { $max: "$salary" },
11  MinSalary: { $min: "$salary" },
12  AvgSalary: { $avg: "$salary" }
13}
14
15

```

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

<code>_id: "Female"</code> <code>Avg : 25</code> <code>MinAge : 21</code> <code>MaxAge : 29</code> <code>MaxSalary : 8799</code> <code>MinSalary : 509</code> <code>AvgSalary : 5020.846153846154</code>	<code>_id: "Male"</code> <code>Avg : 25.666666666666668</code> <code>MinAge : 22</code> <code>MaxAge : 30</code> <code>MaxSalary : 9759</code> <code>MinSalary : 1260</code> <code>AvgSalary : 5252.416666666667</code>
--	---