

Part 1:

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

E:\>docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
6582a7dd6d6f: Download complete
fdc78058942b: Download complete
ff65dddf9395b: Download complete
382d4b58543b: Download complete
b21d1c4a4df0: Download complete
cca7bf1bde66: Download complete
12e5d6c58a07: Download complete
e5f63e34d357: Download complete
Digest: sha256:9342a9279a9841fc5f8192e49dcaaf9404e6bb3a90e8cf134eb96074830dd448
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview mongo

E:\>docker run -d -p 27017:27017 --name m1 mongo
f78cf1038d945c9889c6c9a2974686144221c37eb0544fb3f554b8e9e1d66f36

E:\>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
f78cf1038d94   mongo     "docker-entrypoint.s..." 7 seconds ago  Up 6 seconds  0.0.0.0:27017->27017/tcp           m1

E:\>
```

docker desktop PERSONAL

Search for images, containers, volume... **Ctrl+K**

Containers [Give feedback](#)

Container CPU usage ⓘ
1.13% / 1600% (16 CPUs available)

Container memory usage ⓘ
107.9MB / 7.41GB

[Show charts](#)

Search

Only show running containers

<input type="checkbox"/>	Name	Image	Status	Port(s)	CPU	Actions
<input type="checkbox"/>	vigorous_solomon 7f055d5cc3ac	hello-world:latest	Exited			
<input type="checkbox"/>	gracious_bardeen 67960f8cfb0c	hello-world:<none>	Exited			
<input type="checkbox"/>	m1 f78cf1038d94	mongo:<none>	Running	27017:27017		

Showing 3 items

Walkthroughs

Multi-container applications
8 mins

Containerize your application
3 mins

Engine running

RAM 1.18 GB CPU -- % Disk -- GB avail. of -- GB

BETA Terminal ✓ v4.34.3 1

Part 3:

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

E:\workplace\CSE-512-Assignment2>conda activate first

(first) E:\workplace\CSE-512-Assignment2>python assignment2.py
Data inserted successfully
[ObjectId('671f36e0420230d46fb501e2'), ObjectId('671f36e0420230d46fb501e3'), ObjectId('671f36e0420230d46fb501e4'), Objec
tId('671f36e0420230d46fb501e5'), ObjectId('671f36e0420230d46fb501e6'), ObjectId('671f36e0420230d46fb501e7'), ObjectId('6
71f36e0420230d46fb501e8'), ObjectId('671f36e0420230d46fb501e9'), ObjectId('671f36e0420230d46fb501ea'), ObjectId('671f36e
0420230d46fb501eb'), ObjectId('671f36e0420230d46fb501ec'), ObjectId('671f36e0420230d46fb501ed'), ObjectId('671f36e042023
0d46fb501ee'), ObjectId('671f36e0420230d46fb501ef'), ObjectId('671f36e0420230d46fb501f0'), ObjectId('671f36e0420230d46fb
501f1'), ObjectId('671f36e0420230d46fb501f2'), ObjectId('671f36e0420230d46fb501f3'), ObjectId('671f36e0420230d46fb501f4'
), ObjectId('671f36e0420230d46fb501f5'), ObjectId('671f36e0420230d46fb501f6'), ObjectId('671f36e0420230d46fb501f7'), Obj
ectId('671f36e0420230d46fb501f8'), ObjectId('671f36e0420230d46fb501f9'), ObjectId('671f36e0420230d46fb501fa'), ObjectId(
'671f36e0420230d46fb501fb'), ObjectId('671f36e0420230d46fb501fc'), ObjectId('671f36e0420230d46fb501fd'), ObjectId('671f3
6e0420230d46fb501fe'), ObjectId('671f36e0420230d46fb501ff'), ObjectId('671f36e0420230d46fb50200'), ObjectId('671f36e0420
230d46fb50201'), ObjectId('671f36e0420230d46fb50202'), ObjectId('671f36e0420230d46fb50203'), ObjectId('671f36e0420230d46
fb50204'), ObjectId('671f36e0420230d46fb50205'), ObjectId('671f36e0420230d46fb50206'), ObjectId('671f36e0420230d46fb5020
7'), ObjectId('671f36e0420230d46fb50208'), ObjectId('671f36e0420230d46fb50209'), ObjectId('671f36e0420230d46fb5020a'), O
bjectId('671f36e0420230d46fb5020b'), ObjectId('671f36e0420230d46fb5020c'), ObjectId('671f36e0420230d46fb5020d'), ObjectI
d('671f36e0420230d46fb5020e'), ObjectId('671f36e0420230d46fb5020f'), ObjectId('671f36e0420230d46fb50210'), ObjectId('671
f36e0420230d46fb50211'), ObjectId('671f36e0420230d46fb50212'), ObjectId('671f36e0420230d46fb50213'), ObjectId('671f36e04
20230d46fb50214'), ObjectId('671f36e0420230d46fb50215'), ObjectId('671f36e0420230d46fb50216'), ObjectId('671f36e0420230d
46fb50217'), ObjectId('671f36e0420230d46fb50218'), ObjectId('671f36e0420230d46fb50219'), ObjectId('671f36e0420230d46fb50
21a'), ObjectId('671f36e0420230d46fb5021b'), ObjectId('671f36e0420230d46fb5021c'), ObjectId('671f36e0420230d46fb5021d'
), ObjectId('671f36e0420230d46fb5021e'), ObjectId('671f36e0420230d46fb5021f'), ObjectId('671f36e0420230d46fb50220'), Objec
tId('671f36e0420230d46fb50221'), ObjectId('671f36e0420230d46fb50222'), ObjectId('671f36e0420230d46fb50223'), ObjectId('6
71f36e0420230d46fb50224'), ObjectId('671f36e0420230d46fb50225'), ObjectId('671f36e0420230d46fb50226'), ObjectId('671f36e
0420230d46fb50227'), ObjectId('671f36e0420230d46fb50228'), ObjectId('671f36e0420230d46fb50229'), ObjectId('671f36e042023
```

Part 4:

```
C:\Windows\System32\cmd.e x + v
Total number of orders: 100
Number of orders per state:
State: Maine , Count: 1
State: Michigan , Count: 1
State: Montana , Count: 1
State: New Mexico , Count: 1
State: Missouri , Count: 1
State: Washington , Count: 1
State: District of Columbia , Count: 1
State: Maryland , Count: 1
State: Mississippi , Count: 1
State: Oklahoma , Count: 1
State: Nevada , Count: 1
State: Kansas , Count: 1
State: Utah , Count: 1
State: Idaho , Count: 2
State: Wisconsin , Count: 2
State: Louisiana , Count: 2
State: Connecticut , Count: 2
State: Indiana , Count: 2
State: Nebraska , Count: 2
State: Georgia , Count: 3
State: Illinois , Count: 3
State: Colorado , Count: 3
State: Tennessee , Count: 3
State: South Carolina , Count: 3
State: North Carolina , Count: 3
State: Arizona , Count: 3
State: Pennsylvania , Count: 4
State: New York , Count: 4
```

Part 5:

```
C:\Windows\System32\cmd.e x + v
State: California , Count: 9
=====
Product Frequencies:
Product ID: 94 , Frequency: 1
Product ID: 59 , Frequency: 1
Product ID: 45 , Frequency: 1
Product ID: 30 , Frequency: 1
Product ID: 26 , Frequency: 1
Product ID: 29 , Frequency: 1
Product ID: 66 , Frequency: 1
Product ID: 49 , Frequency: 1
Product ID: 34 , Frequency: 1
Product ID: 84 , Frequency: 1
Product ID: 98 , Frequency: 1
Product ID: 77 , Frequency: 1
Product ID: 96 , Frequency: 1
Product ID: 74 , Frequency: 1
Product ID: 51 , Frequency: 1
Product ID: 36 , Frequency: 1
Product ID: 31 , Frequency: 1
Product ID: 92 , Frequency: 1
Product ID: 91 , Frequency: 1
Product ID: 85 , Frequency: 1
Product ID: 81 , Frequency: 1
Product ID: 78 , Frequency: 1
Product ID: 61 , Frequency: 1
Product ID: 57 , Frequency: 1
Product ID: 47 , Frequency: 1
Product ID: 52 , Frequency: 1
Product ID: 50 , Frequency: 1
```

Part 6:

```
C:\Windows\System32\cmd.e x + v
Product ID: 15 , Frequency: 1
Product ID: 14 , Frequency: 1
Product ID: 58 , Frequency: 1
Product ID: 42 , Frequency: 1
=====
Total high-value orders in California: 8
High-value orders details:
Order ID: 6, Customer ID: 6, Quantity, 98, Unit price: 59.97, Order Date: 12/5/2021, State: California, Total Pirce: 640
1.71, Preminn Customer: True, City: Long Beach
Order ID: 16, Customer ID: 16, Quantity, 78, Unit price: 767.87, Order Date: 9/11/2021, State: California, Total Pirce:
7317.26, Preminn Customer: True, City: Van Nuys
Order ID: 30, Customer ID: 30, Quantity, 78, Unit price: 457.81, Order Date: 7/8/2021, State: California, Total Pirce: 6
225.36, Preminn Customer: False, City: Fresno
Order ID: 68, Customer ID: 68, Quantity, 54, Unit price: 273.28, Order Date: 12/28/2021, State: California, Total Pirce:
5747.35, Preminn Customer: True, City: San Francisco
Order ID: 69, Customer ID: 69, Quantity, 12, Unit price: 473.77, Order Date: 8/1/2021, State: California, Total Pirce: 8
623.95, Preminn Customer: True, City: Santa Clara
Order ID: 74, Customer ID: 74, Quantity, 6, Unit price: 675.02, Order Date: 10/7/2021, State: California, Total Pirce: 9
080.25, Preminn Customer: False, City: Whittier
Order ID: 91, Customer ID: 91, Quantity, 92, Unit price: 677.86, Order Date: 1/11/2021, State: California, Total Pirce:
7161.25, Preminn Customer: False, City: Sacramento
Order ID: 99, Customer ID: 99, Quantity, 97, Unit price: 931.39, Order Date: 10/10/2021, State: California, Total Pirce:
3395.12, Preminn Customer: False, City: San Francisco
=====
Top 10 States with High-Value Orders (>$500):
Rank 1: State: California, Order Count: 9
Rank 2: State: Florida, Order Count: 8
Rank 3: State: Texas, Order Count: 7
Rank 4: State: Virginia, Order Count: 6
Rank 5: State: Ohio, Order Count: 5
```

Part 7 - 9:

```
C:\Windows\System32\cmd.e x + v
=====
Top 10 States with High-Value Orders (>$500):
Rank 1: State: California, Order Count: 9
Rank 2: State: Florida, Order Count: 8
Rank 3: State: Texas, Order Count: 7
Rank 4: State: Virginia, Order Count: 6
Rank 5: State: Ohio, Order Count: 5
Rank 6: State: Alabama, Order Count: 5
Rank 7: State: New York, Order Count: 4
Rank 8: State: Pennsylvania, Order Count: 4
Rank 9: State: Minnesota, Order Count: 4
Rank 10: State: South Carolina, Order Count: 3
=====
Total premium customer in Texas: 5
Premium customer details:
Order ID: 1, Customer ID: 1, Quantity, 27, Unit price: 386.3, Order Date: 11/29/2021, State: Texas, Total Pirce: 2080.1,
Preminn Customer: False, City: Houston
Order ID: 18, Customer ID: 18, Quantity, 22, Unit price: 439.59, Order Date: 12/28/2021, State: Texas, Total Pirce: 4782
.59, Preminn Customer: False, City: Dallas
Order ID: 55, Customer ID: 55, Quantity, 94, Unit price: 793.6, Order Date: 8/28/2021, State: Texas, Total Pirce: 3603.7
9, Preminn Customer: False, City: El Paso
Order ID: 59, Customer ID: 59, Quantity, 99, Unit price: 704.61, Order Date: 5/2/2021, State: Texas, Total Pirce: 6370.8
8, Preminn Customer: True, City: Beaumont
Order ID: 75, Customer ID: 75, Quantity, 94, Unit price: 589.66, Order Date: 6/18/2021, State: Texas, Total Pirce: 2970.
24, Preminn Customer: True, City: Fort Worth
=====
Total orders placed in New York City on 1/9/2021: 0
Order Detail: None

(first) E:\workplace\CSE-512-Assignment2>
```

Part 10:

Aggregate operations are expressions used to generate reduced and summarized results in MongoDB. The Aggregation Framework provides a powerful set of tools to perform complex data processing and analysis operations on collections of data and allows users to create a pipeline that consists of one or more stages, each of which performs a specific operation on the data.

Data flows through a series of stages, each of which performs a specific operation on a collection of data and passes the result to the next stage. Each stage is defined by an operator (e.g., `$match`, `$group`, etc.), and different operators implement different processing logic.

Match: `{ $match: { key: "condition" } }`. It filters the data that matches the conditions from a collection of data, like the "WHERE" in sql.

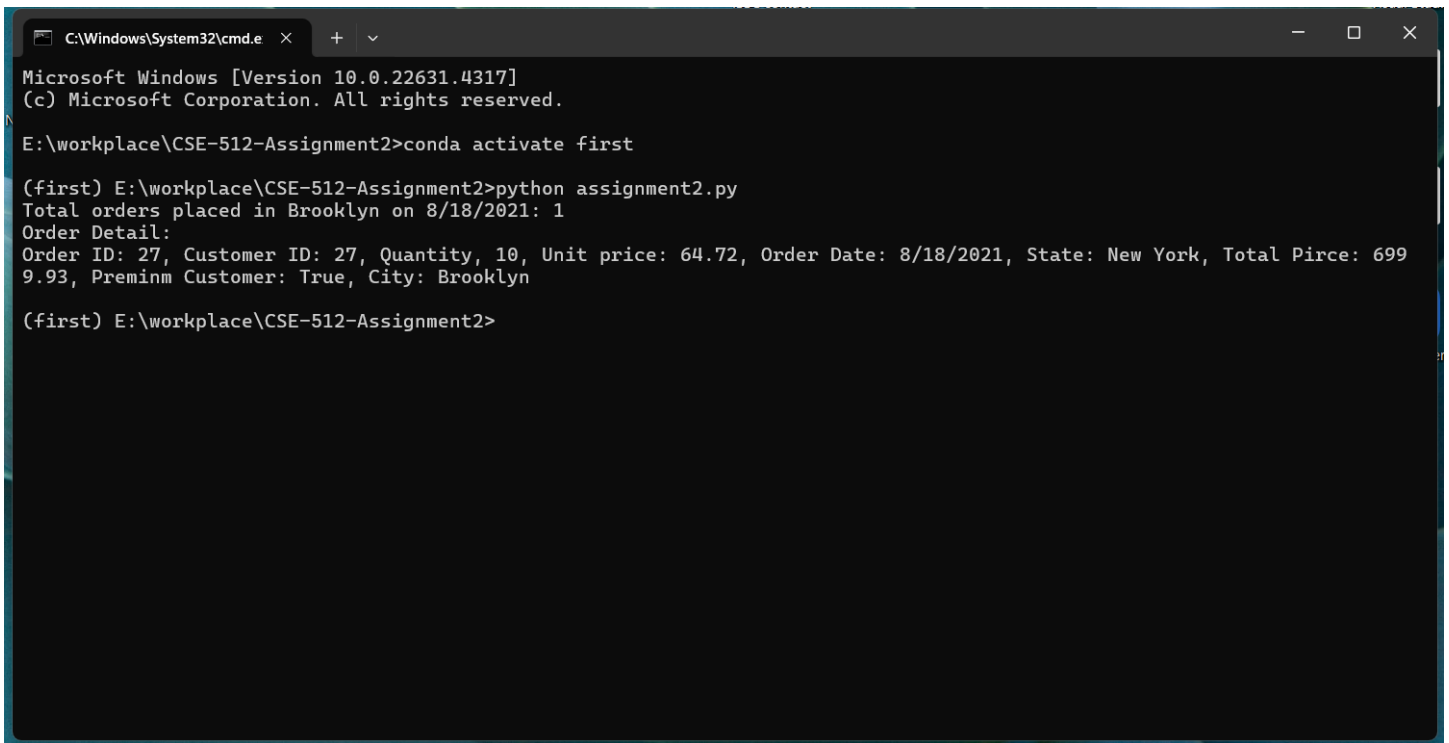
Group: `{ $group: { _id: "$field", count: { $sum: 1 } } }`. This operator group documents according to specified fields and performs aggregation calculations such as summing and averaging.

Sort: `{ $sort: { key: -1 } }`. Sort the result by specified order and attributes.

Project: `{ $project: { <specification(s)> } }` Passes along the documents with the requested fields to the next stage in the pipeline. The specified fields can be existing fields from the input documents or newly computed fields. It is like "SELECT" in sql.

In this assignment, `$match` is used to filter the data that matches the criteria, for example, to find order by date or find orders with total price greater than 500. And `$group` is used to count number of orders in the result.

Part 9:



```
C:\Windows\System32\cmd.e  ×  +  -  ×  
Microsoft Windows [Version 10.0.22631.4317]  
(c) Microsoft Corporation. All rights reserved.  
  
E:\workplace\CSE-512-Assignment2>conda activate first  
  
(first) E:\workplace\CSE-512-Assignment2>python assignment2.py  
Total orders placed in Brooklyn on 8/18/2021: 1  
Order Detail:  
Order ID: 27, Customer ID: 27, Quantity, 10, Unit price: 64.72, Order Date: 8/18/2021, State: New York, Total Pirce: 699  
9.93, Preminn Customer: True, City: Brooklyn  
  
(first) E:\workplace\CSE-512-Assignment2>
```

Since my mock data does not have a New York City order on 1/9/2021, I have used a Brooklyn order on 8/18 2021 here to show what happens when the query result in Part 9 has data.