

Distributed Database Systems

FALL – 2025

Assignment -1

Ashish Raj Shekhar

1223100216

1. Docker screenshot

```
ashishrajshekar@Ashishs-MacBook-Pro Assign-1 % docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS                NAMES
16a18555e97a   mongo-assign1 "docker-entrypoint.s..." 6 hours ago Up 6 hours    0.0.0.0:27017->27017/tcp   mongo-assign1
ashishrajshekar@Ashishs-MacBook-Pro Assign-1 %
```

The screenshot shows the Docker Desktop application window. The left sidebar contains navigation options: Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area displays the 'Containers' tab with a search bar and a toggle for 'Only show running containers'. Below this is a table of running containers. The table has columns for Name, Container ID, Image, Port(s), CPU (%), Last started, and Actions. Two containers are listed: 'fairtestai-db' and 'mongo-assign1'. The 'mongo-assign1' container is highlighted with a green dot, indicating it is running. Below the table, there are two walkthroughs: 'Multi-container applications' and 'Containerize your application'. The bottom status bar shows 'Engine running', RAM usage (1.91 GB / 16 GB), CPU usage (0.13%), and disk usage (19.31 GB / 58.37 GB).

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
fairtestai-db	39717c8ba5e7	postgres:15	5432:5432	0%	11 days ago	[Play] [Stop] [Delete]
mongo-assign1	16a18555e97a	mongo-assign1	27017:27017	0.95%	6 hours ago	[Play] [Stop] [Delete]

2. Mockaroo json dataset creation screenshot

The screenshot displays the Mockaroo web interface for creating a JSON dataset. The top navigation bar includes links for SCHEMAS, DATASETS, APIS, PROJECTS, SCENARIOS, FUNCTIONS, DATABASES, and DE-IDENTIFY, along with settings, help, and user options (SIGN IN, UPGRADE NOW).

The main configuration area is divided into three columns: Field Name, Type, and Options. Fields are listed in a table with their respective types and options. At the bottom, there are controls for the number of rows (100), format (JSON), and checkboxes for array and include null values. A hint suggests using "." in column names for nested objects and brackets for arrays.

Below the configuration area, there are buttons for GENERATE DATA, PREVIEW, SAVE AS..., DERIVE FROM EXAMPLE..., and MORE.

The PREVIEW section shows the generated JSON data, which is a list of 41 objects. Each object represents an order with fields like order_id, customer_id, product_id, quantity, unit_price, order_date, state, total_price, premium_customer, and city. The data is displayed in a code editor format with line numbers.

```
{
  "order_id": 1, "customer_id": 1, "product_id": 1, "quantity": 16, "unit_price": 399.4, "order_date": "10/21/2021", "state": "District of Columbia", "total_price": 6390.4
}, {
  "order_id": 2, "customer_id": 2, "product_id": 2, "quantity": 98, "unit_price": 516.2, "order_date": "10/21/2021", "state": "Alaska", "total_price": 50598.6
}, {
  "order_id": 3, "customer_id": 3, "product_id": 3, "quantity": 91, "unit_price": 43.9, "order_date": "10/21/2021", "state": "California", "total_price": 3994.9
}, {
  "order_id": 4, "customer_id": 4, "product_id": 4, "quantity": 72, "unit_price": 327.0, "order_date": "10/21/2021", "state": "New York", "total_price": 23544.0
}, {
  "order_id": 5, "customer_id": 5, "product_id": 5, "quantity": 30, "unit_price": 370.2, "order_date": "10/21/2021", "state": "Texas", "total_price": 11106.0
}, {
  "order_id": 6, "customer_id": 6, "product_id": 6, "quantity": 31, "unit_price": 874.2, "order_date": "10/21/2021", "state": "Wisconsin", "total_price": 27100.2
}, {
  "order_id": 7, "customer_id": 7, "product_id": 7, "quantity": 54, "unit_price": 791.5, "order_date": "10/21/2021", "state": "Ohio", "total_price": 42741.0
}, {
  "order_id": 8, "customer_id": 8, "product_id": 8, "quantity": 14, "unit_price": 211.8, "order_date": "10/21/2021", "state": "New York", "total_price": 2965.2
}, {
  "order_id": 9, "customer_id": 9, "product_id": 9, "quantity": 9, "unit_price": 618.8, "order_date": "10/21/2021", "state": "Texas", "total_price": 5569.2
}, {
  "order_id": 10, "customer_id": 10, "product_id": 10, "quantity": 14, "unit_price": 966.5, "order_date": "10/21/2021", "state": "Kentucky", "total_price": 13531.0
}, {
  "order_id": 11, "customer_id": 11, "product_id": 11, "quantity": 75, "unit_price": 7.3, "order_date": "10/21/2021", "state": "Georgia", "total_price": 547.5
}, {
  "order_id": 12, "customer_id": 12, "product_id": 12, "quantity": 88, "unit_price": 365.7, "order_date": "10/21/2021", "state": "Texas", "total_price": 32181.6
}, {
  "order_id": 13, "customer_id": 13, "product_id": 13, "quantity": 47, "unit_price": 855.1, "order_date": "10/21/2021", "state": "Oklahoma", "total_price": 40189.7
}, {
  "order_id": 14, "customer_id": 14, "product_id": 14, "quantity": 24, "unit_price": 439.7, "order_date": "10/21/2021", "state": "Arkansas", "total_price": 10552.8
}, {
  "order_id": 15, "customer_id": 15, "product_id": 15, "quantity": 66, "unit_price": 727.7, "order_date": "10/21/2021", "state": "District of Columbia", "total_price": 48028.2
}, {
  "order_id": 16, "customer_id": 16, "product_id": 16, "quantity": 98, "unit_price": 346.9, "order_date": "10/21/2021", "state": "California", "total_price": 33996.2
}, {
  "order_id": 17, "customer_id": 17, "product_id": 17, "quantity": 14, "unit_price": 852.8, "order_date": "10/21/2021", "state": "Maryland", "total_price": 11939.2
}, {
  "order_id": 18, "customer_id": 18, "product_id": 18, "quantity": 43, "unit_price": 114.6, "order_date": "10/21/2021", "state": "Colorado", "total_price": 4927.8
}, {
  "order_id": 19, "customer_id": 19, "product_id": 19, "quantity": 53, "unit_price": 828.6, "order_date": "10/21/2021", "state": "Minnesota", "total_price": 43915.8
}, {
  "order_id": 20, "customer_id": 20, "product_id": 20, "quantity": 56, "unit_price": 780.8, "order_date": "10/21/2021", "state": "Wisconsin", "total_price": 43724.8
}, {
  "order_id": 21, "customer_id": 21, "product_id": 21, "quantity": 76, "unit_price": 282.1, "order_date": "10/21/2021", "state": "Illinois", "total_price": 21447.6
}, {
  "order_id": 22, "customer_id": 22, "product_id": 22, "quantity": 76, "unit_price": 898.4, "order_date": "10/21/2021", "state": "Colorado", "total_price": 68278.4
}, {
  "order_id": 23, "customer_id": 23, "product_id": 23, "quantity": 30, "unit_price": 840.7, "order_date": "10/21/2021", "state": "Illinois", "total_price": 25221.0
}, {
  "order_id": 24, "customer_id": 24, "product_id": 24, "quantity": 90, "unit_price": 133.1, "order_date": "10/21/2021", "state": "Arizona", "total_price": 11979.0
}, {
  "order_id": 25, "customer_id": 25, "product_id": 25, "quantity": 2, "unit_price": 611.2, "order_date": "10/21/2021", "state": "Massachusetts", "total_price": 1222.4
}, {
  "order_id": 26, "customer_id": 26, "product_id": 26, "quantity": 3, "unit_price": 811.5, "order_date": "10/21/2021", "state": "California", "total_price": 2434.5
}, {
  "order_id": 27, "customer_id": 27, "product_id": 27, "quantity": 50, "unit_price": 14.1, "order_date": "10/21/2021", "state": "New York", "total_price": 705.0
}, {
  "order_id": 28, "customer_id": 28, "product_id": 28, "quantity": 4, "unit_price": 596.8, "order_date": "10/21/2021", "state": "Texas", "total_price": 2387.2
}, {
  "order_id": 29, "customer_id": 29, "product_id": 29, "quantity": 50, "unit_price": 755.3, "order_date": "10/21/2021", "state": "Nebraska", "total_price": 37765.0
}, {
  "order_id": 30, "customer_id": 30, "product_id": 30, "quantity": 17, "unit_price": 630.0, "order_date": "10/21/2021", "state": "North Carolina", "total_price": 10710.0
}, {
  "order_id": 31, "customer_id": 31, "product_id": 31, "quantity": 46, "unit_price": 778.1, "order_date": "10/21/2021", "state": "Texas", "total_price": 35792.6
}, {
  "order_id": 32, "customer_id": 32, "product_id": 32, "quantity": 81, "unit_price": 608.3, "order_date": "10/21/2021", "state": "Florida", "total_price": 49272.3
}, {
  "order_id": 33, "customer_id": 33, "product_id": 33, "quantity": 11, "unit_price": 630.5, "order_date": "10/21/2021", "state": "Washington", "total_price": 6935.5
}, {
  "order_id": 34, "customer_id": 34, "product_id": 34, "quantity": 69, "unit_price": 161.7, "order_date": "10/21/2021", "state": "Mississippi", "total_price": 11167.3
}, {
  "order_id": 35, "customer_id": 35, "product_id": 35, "quantity": 25, "unit_price": 216.1, "order_date": "10/21/2021", "state": "Texas", "total_price": 5402.5
}, {
  "order_id": 36, "customer_id": 36, "product_id": 36, "quantity": 57, "unit_price": 60.7, "order_date": "10/21/2021", "state": "Massachusetts", "total_price": 3459.9
}, {
  "order_id": 37, "customer_id": 37, "product_id": 37, "quantity": 50, "unit_price": 105.1, "order_date": "10/21/2021", "state": "California", "total_price": 5255.0
}, {
  "order_id": 38, "customer_id": 38, "product_id": 38, "quantity": 41, "unit_price": 360.6, "order_date": "10/21/2021", "state": "Alabama", "total_price": 14784.6
}, {
  "order_id": 39, "customer_id": 39, "product_id": 39, "quantity": 59, "unit_price": 387.0, "order_date": "10/21/2021", "state": "New York", "total_price": 22821.0
}, {
  "order_id": 40, "customer_id": 40, "product_id": 40, "quantity": 85, "unit_price": 100.1, "order_date": "10/21/2021", "state": "Massachusetts", "total_price": 8508.5
}, {
  "order_id": 41, "customer_id": 41, "product_id": 41, "quantity": 49, "unit_price": 924.0, "order_date": "10/21/2021", "state": "Michigan", "total_price": 45276.0
}
```

3. Insert the data using insert_many

```
(venv) ashishrajshekar@Ashishs-MacBook-Pro Assign-1 % python3 Assignment\ 1.py
Data inserted successfully:
[ObjectId('68de16013858a0ede4637f5e'), ObjectId('68de16013858a0ede4637f5f'), ObjectId('68de16013858a0ede4637f60'), ObjectId('68de16013858a0ede4637f61'), ObjectId('68de16013858a0ede4637f62'), ObjectId('68de16013858a0ede4637f63'), ObjectId('68de16013858a0ede4637f64'), ObjectId('68de16013858a0ede4637f65'), ObjectId('68de16013858a0ede4637f66'), ObjectId('68de16013858a0ede4637f67'), ObjectId('68de16013858a0ede4637f68'), ObjectId('68de16013858a0ede4637f69'), ObjectId('68de16013858a0ede4637f6a'), ObjectId('68de16013858a0ede4637f6b'), ObjectId('68de16013858a0ede4637f6c'), ObjectId('68de16013858a0ede4637f6d'), ObjectId('68de16013858a0ede4637f6e'), ObjectId('68de16013858a0ede4637f6f'), ObjectId('68de16013858a0ede4637f70'), ObjectId('68de16013858a0ede4637f71'), ObjectId('68de16013858a0ede4637f72'), ObjectId('68de16013858a0ede4637f73'), ObjectId('68de16013858a0ede4637f74'), ObjectId('68de16013858a0ede4637f75'), ObjectId('68de16013858a0ede4637f76'), ObjectId('68de16013858a0ede4637f77'), ObjectId('68de16013858a0ede4637f78'), ObjectId('68de16013858a0ede4637f79'), ObjectId('68de16013858a0ede4637f7a'), ObjectId('68de16013858a0ede4637f7b'), ObjectId('68de16013858a0ede4637f7c'), ObjectId('68de16013858a0ede4637f7d'), ObjectId('68de16013858a0ede4637f7e'), ObjectId('68de16013858a0ede4637f7f'), ObjectId('68de16013858a0ede4637f80'), ObjectId('68de16013858a0ede4637f81'), ObjectId('68de16013858a0ede4637f82'), ObjectId('68de16013858a0ede4637f83'), ObjectId('68de16013858a0ede4637f84'), ObjectId('68de16013858a0ede4637f85'), ObjectId('68de16013858a0ede4637f86'), ObjectId('68de16013858a0ede4637f87'), ObjectId('68de16013858a0ede4637f88'), ObjectId('68de16013858a0ede4637f89'), ObjectId('68de16013858a0ede4637f8a'), ObjectId('68de16013858a0ede4637f8b'), ObjectId('68de16013858a0ede4637f8c'), ObjectId('68de16013858a0ede4637f8d'), ObjectId('68de16013858a0ede4637f8e'), ObjectId('68de16013858a0ede4637f8f'), ObjectId('68de16013858a0ede4637f90'), ObjectId('68de16013858a0ede4637f91'), ObjectId('68de16013858a0ede4637f92'), ObjectId('68de16013858a0ede4637f93'), ObjectId('68de16013858a0ede4637f94'), ObjectId('68de16013858a0ede4637f95'), ObjectId('68de16013858a0ede4637f96'), ObjectId('68de16013858a0ede4637f97'), ObjectId('68de16013858a0ede4637f98'), ObjectId('68de16013858a0ede4637f99'), ObjectId('68de16013858a0ede4637f9a'), ObjectId('68de16013858a0ede4637f9b'), ObjectId('68de16013858a0ede4637f9c'), ObjectId('68de16013858a0ede4637f9d'), ObjectId('68de16013858a0ede4637f9e'), ObjectId('68de16013858a0ede4637f9f'), ObjectId('68de16013858a0ede4637fa0'), ObjectId('68de16013858a0ede4637fa1'), ObjectId('68de16013858a0ede4637fa2'), ObjectId('68de16013858a0ede4637fa3'), ObjectId('68de16013858a0ede4637fa4'), ObjectId('68de16013858a0ede4637fa5'), ObjectId('68de16013858a0ede4637fa6'), ObjectId('68de16013858a0ede4637fa7'), ObjectId('68de16013858a0ede4637fa8'), ObjectId('68de16013858a0ede4637fa9'), ObjectId('68de16013858a0ede4637faa'), ObjectId('68de16013858a0ede4637fab'), ObjectId('68de16013858a0ede4637fac'), ObjectId('68de16013858a0ede4637fad'), ObjectId('68de16013858a0ede4637fae'), ObjectId('68de16013858a0ede4637faf'), ObjectId('68de16013858a0ede4637fb0'), ObjectId('68de16013858a0ede4637fb1'), ObjectId('68de16013858a0ede4637fb2'), ObjectId('68de16013858a0ede4637fb3'), ObjectId('68de16013858a0ede4637fb4'), ObjectId('68de16013858a0ede4637fb5'), ObjectId('68de16013858a0ede4637fb6'), ObjectId('68de16013858a0ede4637fb7'), ObjectId('68de16013858a0ede4637fb8'), ObjectId('68de16013858a0ede4637fb9'), ObjectId('68de16013858a0ede4637fba'), ObjectId('68de16013858a0ede4637fbb'), ObjectId('68de16013858a0ede4637fbc'), ObjectId('68de16013858a0ede4637fbd'), ObjectId('68de16013858a0ede4637fbe'), ObjectId('68de16013858a0ede4637fbf'), ObjectId('68de16013858a0ede4637fc0'), ObjectId('68de16013858a0ede4637fc1')]
```

4. Total number of orders and total orders grouped by state

```
Total number of orders: 100
Number of orders per state:
State: Alaska, Count: 1
State: Arkansas, Count: 1
State: Georgia, Count: 1
State: Kentucky, Count: 1
State: Louisiana, Count: 1
State: Maryland, Count: 1
State: Minnesota, Count: 1
State: Mississippi, Count: 1
State: New Jersey, Count: 1
State: North Carolina, Count: 1
State: North Dakota, Count: 1
State: Oklahoma, Count: 1
State: Tennessee, Count: 1
State: Utah, Count: 1
State: Alabama, Count: 2
State: District of Columbia, Count: 2
State: Illinois, Count: 2
State: Indiana, Count: 2
State: Michigan, Count: 2
State: Missouri, Count: 2
State: Nevada, Count: 2
State: Arizona, Count: 3
State: Colorado, Count: 3
State: Connecticut, Count: 3
State: Nebraska, Count: 3
State: Ohio, Count: 3
State: Washington, Count: 3
State: Wisconsin, Count: 3
State: Virginia, Count: 4
State: Florida, Count: 6
State: Massachusetts, Count: 6
State: Pennsylvania, Count: 6
State: California, Count: 9
State: New York, Count: 9
State: Texas, Count: 11
```

5. Product and their frequencies

```
Product Frequencies:
Product ID: 1, Frequency: 1
Product ID: 2, Frequency: 1
Product ID: 3, Frequency: 1
Product ID: 4, Frequency: 1
Product ID: 5, Frequency: 1
Product ID: 6, Frequency: 1
Product ID: 7, Frequency: 1
Product ID: 8, Frequency: 1
Product ID: 9, Frequency: 1
Product ID: 10, Frequency: 1
Product ID: 11, Frequency: 1
Product ID: 12, Frequency: 1
Product ID: 13, Frequency: 1
Product ID: 14, Frequency: 1
Product ID: 15, Frequency: 1
Product ID: 16, Frequency: 1
Product ID: 17, Frequency: 1
Product ID: 18, Frequency: 1
Product ID: 19, Frequency: 1
Product ID: 20, Frequency: 1
Product ID: 21, Frequency: 1
Product ID: 22, Frequency: 1
Product ID: 23, Frequency: 1
Product ID: 24, Frequency: 1
Product ID: 25, Frequency: 1
Product ID: 26, Frequency: 1
Product ID: 27, Frequency: 1
Product ID: 28, Frequency: 1
Product ID: 29, Frequency: 1
Product ID: 30, Frequency: 1
Product ID: 31, Frequency: 1
Product ID: 32, Frequency: 1
Product ID: 33, Frequency: 1
Product ID: 34, Frequency: 1
Product ID: 35, Frequency: 1
Product ID: 36, Frequency: 1
Product ID: 37, Frequency: 1
Product ID: 38, Frequency: 1
Product ID: 39, Frequency: 1
Product ID: 40, Frequency: 1
Product ID: 41, Frequency: 1
Product ID: 42, Frequency: 1
Product ID: 43, Frequency: 1
Product ID: 44, Frequency: 1
Product ID: 45, Frequency: 1
Product ID: 46, Frequency: 1
Product ID: 47, Frequency: 1
Product ID: 48, Frequency: 1
Product ID: 49, Frequency: 1
```

```
Problems Output Debug Console Terminal Ports
Product ID: 48, Frequency: 1
Product ID: 49, Frequency: 1
Product ID: 50, Frequency: 1
Product ID: 51, Frequency: 1
Product ID: 52, Frequency: 1
Product ID: 53, Frequency: 1
Product ID: 54, Frequency: 1
Product ID: 55, Frequency: 1
Product ID: 56, Frequency: 1
Product ID: 57, Frequency: 1
Product ID: 58, Frequency: 1
Product ID: 59, Frequency: 1
Product ID: 60, Frequency: 1
Product ID: 61, Frequency: 1
Product ID: 62, Frequency: 1
Product ID: 63, Frequency: 1
Product ID: 64, Frequency: 1
Product ID: 65, Frequency: 1
Product ID: 66, Frequency: 1
Product ID: 67, Frequency: 1
Product ID: 68, Frequency: 1
Product ID: 69, Frequency: 1
Product ID: 70, Frequency: 1
Product ID: 71, Frequency: 1
Product ID: 72, Frequency: 1
Product ID: 73, Frequency: 1
Product ID: 74, Frequency: 1
Product ID: 75, Frequency: 1
Product ID: 76, Frequency: 1
Product ID: 77, Frequency: 1
Product ID: 78, Frequency: 1
Product ID: 79, Frequency: 1
Product ID: 80, Frequency: 1
Product ID: 81, Frequency: 1
Product ID: 82, Frequency: 1
Product ID: 83, Frequency: 1
Product ID: 84, Frequency: 1
Product ID: 85, Frequency: 1
Product ID: 86, Frequency: 1
Product ID: 87, Frequency: 1
Product ID: 88, Frequency: 1
Product ID: 89, Frequency: 1
Product ID: 90, Frequency: 1
Product ID: 91, Frequency: 1
Product ID: 92, Frequency: 1
Product ID: 93, Frequency: 1
Product ID: 94, Frequency: 1
Product ID: 95, Frequency: 1
Product ID: 96, Frequency: 1
Product ID: 97, Frequency: 1
Product ID: 98, Frequency: 1
Product ID: 99, Frequency: 1
Product ID: 100, Frequency: 1
Total high-value orders in California: 7
High-value order details:
```

6. Orders where amount is greater than 1000 and in California

```
Product ID: 100, Frequency: 1
Total high-value orders in California: 7
High-value order details:
Order ID: 3, Customer ID: 3, Product ID: 3, Quantity: 91, Unit Price: 43.9, Order Date: 10/21/2021, State: California, Total Price: 5978.49, Premium Customer: False, City: Berkeley
Order ID: 16, Customer ID: 16, Product ID: 16, Quantity: 98, Unit Price: 346.9, Order Date: 10/21/2021, State: California, Total Price: 6574.93, Premium Customer: False, City: Garden Grove
Order ID: 26, Customer ID: 26, Product ID: 26, Quantity: 3, Unit Price: 811.5, Order Date: 10/21/2021, State: California, Total Price: 8123.25, Premium Customer: True, City: Long Beach
Order ID: 37, Customer ID: 37, Product ID: 37, Quantity: 50, Unit Price: 105.1, Order Date: 10/21/2021, State: California, Total Price: 5922.43, Premium Customer: True, City: San Bernardino
Order ID: 87, Customer ID: 87, Product ID: 87, Quantity: 26, Unit Price: 687.0, Order Date: 10/21/2021, State: California, Total Price: 6943.72, Premium Customer: True, City: Glendale
Order ID: 88, Customer ID: 88, Product ID: 88, Quantity: 15, Unit Price: 339.5, Order Date: 10/21/2021, State: California, Total Price: 3173.13, Premium Customer: True, City: Santa Clara
Order ID: 100, Customer ID: 100, Product ID: 100, Quantity: 94, Unit Price: 978.9, Order Date: 10/21/2021, State: California, Total Price: 9167.86, Premium Customer: False, City: San Jose
```

7. 10 states with amount exceeding more than 500

```
Top states with high-value orders (> $500):
Rank 1: State: Texas, High-value orders: 11
Rank 2: State: New York, High-value orders: 9
Rank 3: State: California, High-value orders: 8
Rank 4: State: Florida, High-value orders: 6
Rank 5: State: Massachusetts, High-value orders: 6
Rank 6: State: Pennsylvania, High-value orders: 6
Rank 7: State: Virginia, High-value orders: 4
Rank 8: State: Arizona, High-value orders: 3
Rank 9: State: Colorado, High-value orders: 3
Rank 10: State: Connecticut, High-value orders: 3
```

8. Customers with order more than 2000

```
Total premium customers in Texas (> $2,000): 9
Premium customer order details:
Customer ID: 5, Order ID: 5, Product ID: 5, Quantity: 30, Unit Price: 370.2, Order Date: 10/21/2021, Total Price: 9077.11, Premium Customer: True, City: El Paso
Customer ID: 12, Order ID: 12, Product ID: 12, Quantity: 88, Unit Price: 365.7, Order Date: 10/21/2021, Total Price: 3334.07, Premium Customer: True, City: El Paso
Customer ID: 28, Order ID: 28, Product ID: 28, Quantity: 4, Unit Price: 596.8, Order Date: 10/21/2021, Total Price: 4895.16, Premium Customer: True, City: Houston
Customer ID: 31, Order ID: 31, Product ID: 31, Quantity: 46, Unit Price: 778.1, Order Date: 10/21/2021, Total Price: 3980.29, Premium Customer: False, City: El Paso
Customer ID: 35, Order ID: 35, Product ID: 35, Quantity: 25, Unit Price: 216.1, Order Date: 10/21/2021, Total Price: 8974.95, Premium Customer: True, City: El Paso
Customer ID: 43, Order ID: 43, Product ID: 43, Quantity: 68, Unit Price: 611.5, Order Date: 10/21/2021, Total Price: 5079.34, Premium Customer: False, City: El Paso
Customer ID: 61, Order ID: 61, Product ID: 61, Quantity: 64, Unit Price: 726.5, Order Date: 10/21/2021, Total Price: 8576.16, Premium Customer: False, City: San Antonio
Customer ID: 70, Order ID: 70, Product ID: 70, Quantity: 0, Unit Price: 384.6, Order Date: 10/21/2021, Total Price: 5428.0, Premium Customer: False, City: Midland
Customer ID: 93, Order ID: 93, Product ID: 93, Quantity: 87, Unit Price: 207.1, Order Date: 10/21/2021, Total Price: 7398.02, Premium Customer: True, City: San Antonio
```

9. New York orders on 10/21/2021

```
Customer: True, City: San Antonio
Total orders in New York City on 10/21/2021: 3
Order details:
Order ID: 39, Customer ID: 39, Product ID: 39, Quantity: 59, Unit Price: 387.0, Total Price: 3769.73, Premium Customer: True
Order ID: 84, Customer ID: 84, Product ID: 84, Quantity: 42, Unit Price: 890.0, Total Price: 2420.65, Premium Customer: False
Order ID: 90, Customer ID: 90, Product ID: 90, Quantity: 25, Unit Price: 301.9, Total Price: 6204.12, Premium Customer: False
ashishrajshekhara@Ashishs-MacBook-Pro Assign-1 %
```

10. The aggregation pipelines helps us perform operations with a sequential flow where in the first step we extract the data , then load it and then perform transformations like an etl pipeline.

- \$match : this is used to filter the documents
- \$sort: we use this to display the fetched result in the desired order(ASC/DESC)

And similarly we have others like sort, limit etc.

```
pipeline = [
  {'$match': {'state': 'Texas', 'total_price': {'$gt': 2000}}},
  {
    '$group': {
      '_id': '$customer_id',
      'orders': {'$push': '$$ROOT'},
    },
  },
  {'$sort': {'_id': 1}},
]
```

Here each state takes documents from the initial query as input , applies sorting based on greater than condition operations then provides the result back to the customer.

also find_order_totals function uses \$group to combine orders per state after counting the collection and then uses \$sort puts the states in ascending order.

```
def find_order_totals():
    """Finds the total number of orders and the number of orders per state, sorted by count in ascending order."""
    total_orders = collection.count_documents({})
    pipeline = [
        {'$group': {'_id': '$state', 'order_count': {'$sum': 1}}},
        {'$sort': {'order_count': 1, '_id': 1}},
    ]
    per_state_count = list(collection.aggregate(pipeline))

    print('Total number of orders:', total_orders)
    print('Number of orders per state:')
    for state in per_state_count:
        print(f"State: {state['_id']}, Count: {state['order_count']}")
```