Title: Deli-Meds

Milestone: NoSQL Implementation

Yaswanth Reddy Nalamalapu
Venkata Mani Sivasai Shanmukha Goparaju

617-777-5405 (Tel of Student 1)
857-397-5588 (Tel of Student 2)

nalamalapu.y@northeastern.edu
goparaju.v@northeastern.edu

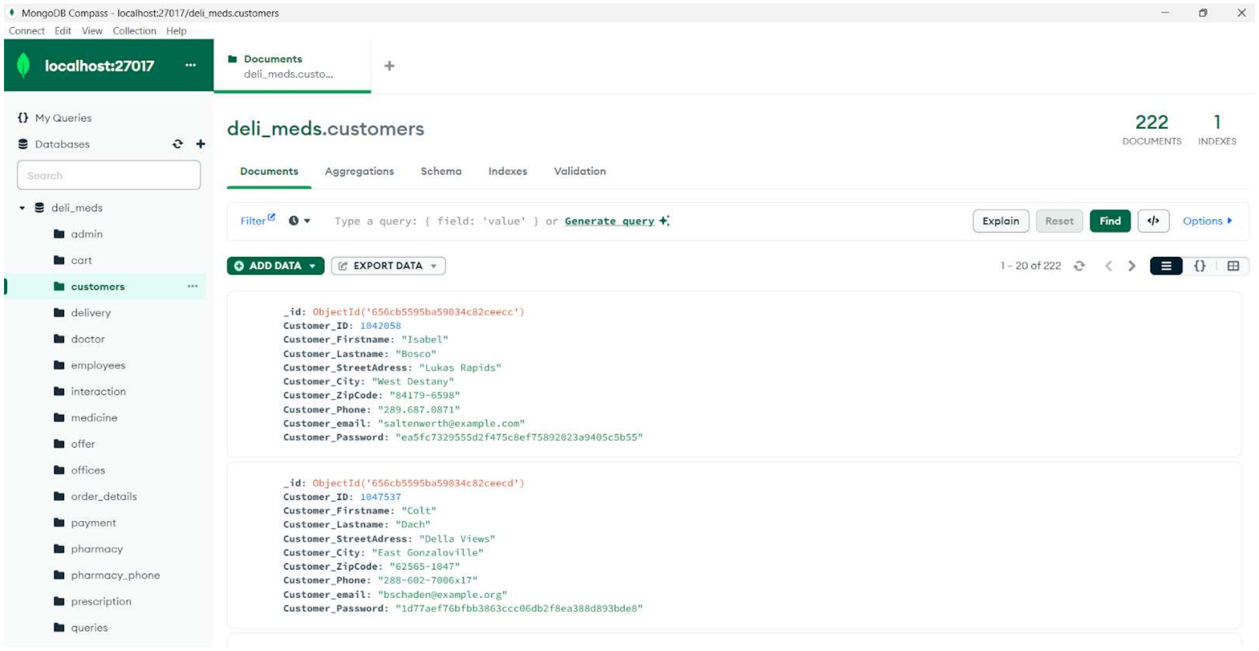
Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Yaswanth Reddy Nalamalapu

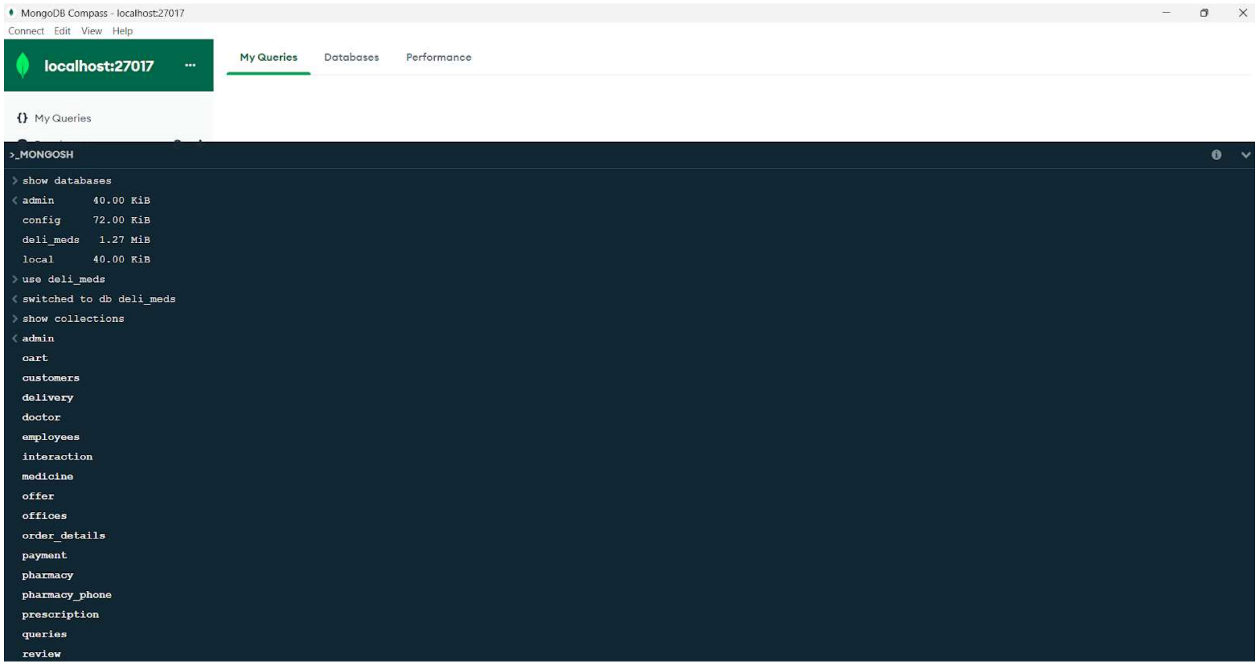
Signature of Student 2: Venkata Mani Sivasai Shanmukha Goparaju

Submission Date: 12/03/2023

- 

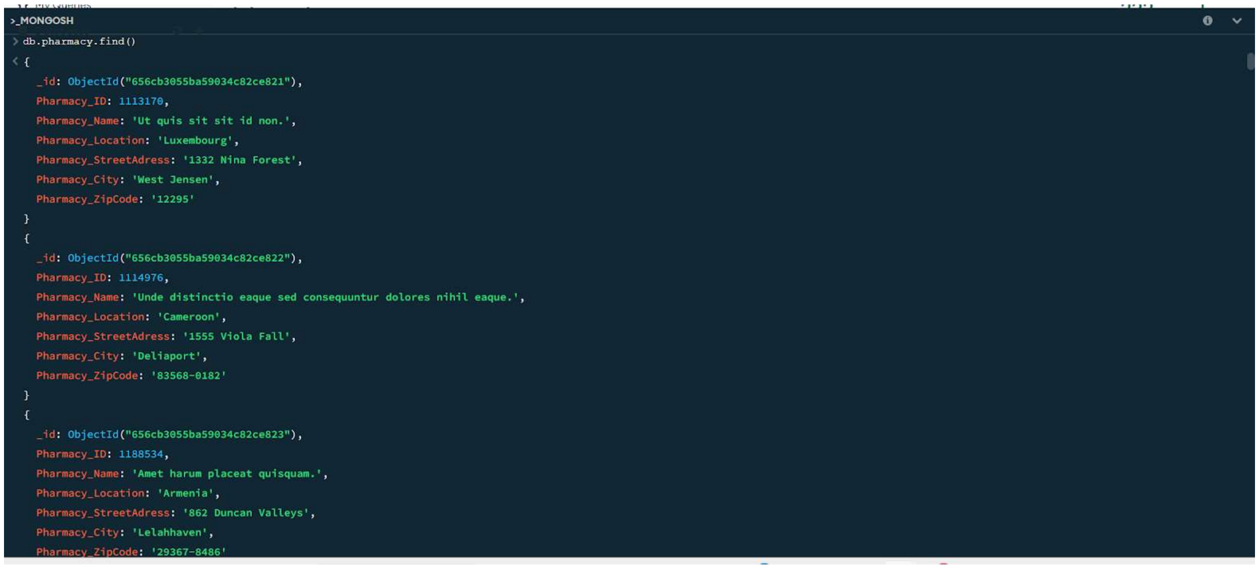
```
_id: ObjectId('656cb5595ba59034c82ceecc')
Customer_ID: 1042058
Customer_Firstname: "Isabel"
Customer_Lastname: "Bosco"
Customer_StreetAdress: "Lukas Rapids"
Customer_City: "West Destany"
Customer_ZipCode: "84179-6598"
Customer_Phone: "289.687.0871"
Customer_email: "saltenwerth@example.com"
Customer_Password: "ea5fc7329555d2f475c8ef75892023a9405c5b55"

_id: ObjectId('656cb5595ba59034c82ceecd')
Customer_ID: 1047537
Customer_Firstname: "Colt"
Customer_Lastname: "Dach"
Customer_StreetAdress: "Della Views"
Customer_City: "East Gonzaloville"
Customer_ZipCode: "62565-1047"
Customer_Phone: "288-602-7006x17"
Customer_email: "bschaden@example.org"
Customer_Password: "1d77aef76bfbb3863ccc06db2f8ea388d893bde8"
```

Created a database named **deli_meds** in MongoDB through adding most of the tables as collections by importing the data in **JSON** format from **MySQL.**

- 

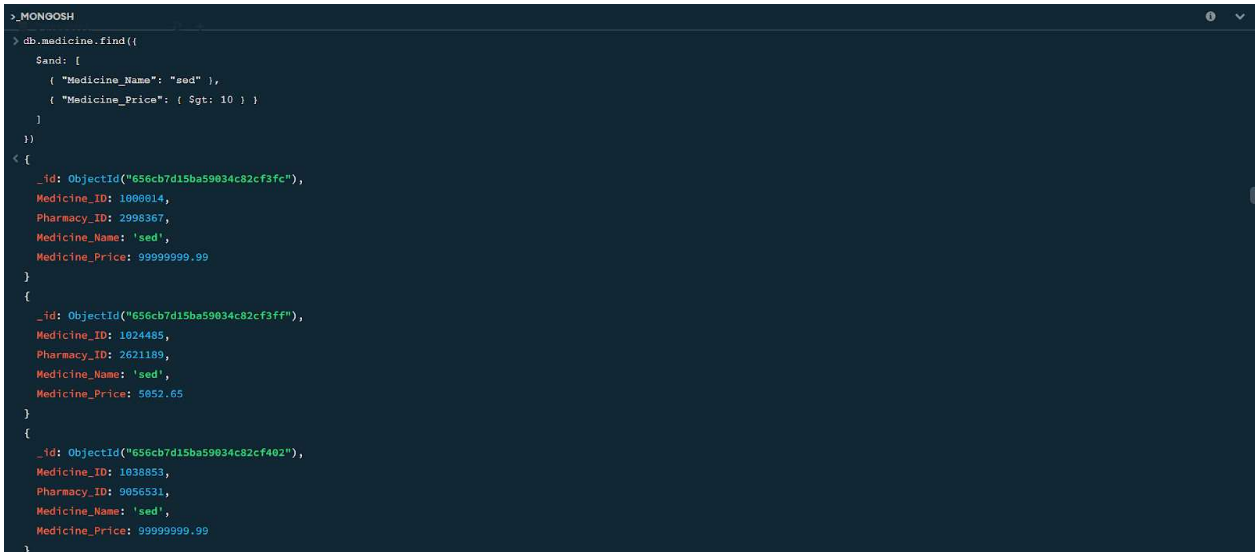
```
> show databases
< admin       40.00 KiB
  config      72.00 KiB
  deli_meds    1.27 MiB
  local       40.00 KiB
> use deli_meds
< switched to db deli_meds
> show collections
< admin
  cart
  customers
  delivery
  doctor
  employees
  interaction
  medicine
  offer
  offices
  order_details
  payment
  pharmacy
  pharmacy_phone
  prescription
  queries
  review
```

This can be seen in the **MongoDB shell** that combines **MongoDB** query language with a modern command-line experience(**CLI**).

- 

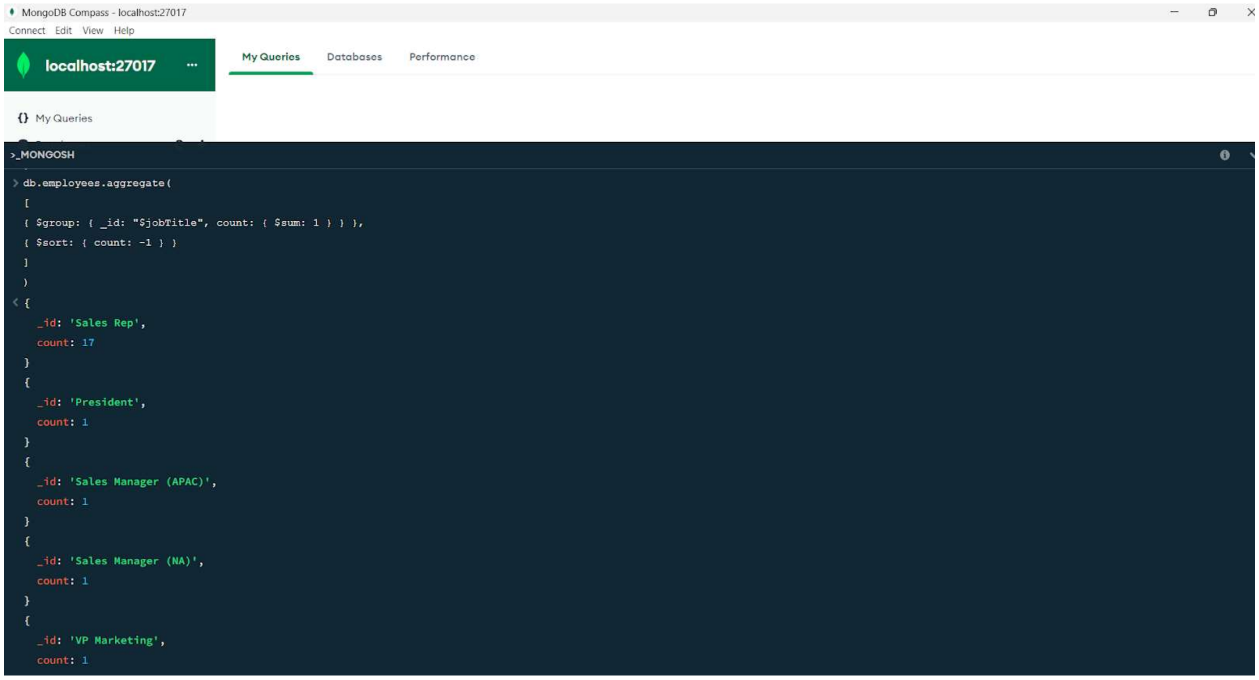
```
>_MONGOSH
> db.pharmacy.find()
< {
    _id: ObjectId("656cb3055ba59034c82ce821"),
    Pharmacy_ID: 1113170,
    Pharmacy_Name: 'Ut quis sit sit id non.',
    Pharmacy_Location: 'Luxembourg',
    Pharmacy_StreetAdress: '1332 Nina Forest',
    Pharmacy_City: 'West Jensen',
    Pharmacy_ZipCode: '12295'
  }
  {
    _id: ObjectId("656cb3055ba59034c82ce822"),
    Pharmacy_ID: 1114976,
    Pharmacy_Name: 'Unde distinctio eaque sed consequuntur dolores nihil eaque.',
    Pharmacy_Location: 'Cameroon',
    Pharmacy_StreetAdress: '1555 Viola Fall',
    Pharmacy_City: 'Deliaport',
    Pharmacy_ZipCode: '83568-0182'
  }
  {
    _id: ObjectId("656cb3055ba59034c82ce823"),
    Pharmacy_ID: 1188534,
    Pharmacy_Name: 'Amet harum placeat quisquam.',
    Pharmacy_Location: 'Armenia',
    Pharmacy_StreetAdress: '862 Duncan Valleys',
    Pharmacy_City: 'Lelahhaven',
    Pharmacy_ZipCode: '29367-8486'
```

A simple query can be executed to fetch the data from a collection named **pharmacy** present in the **deli_meds** database.

- 

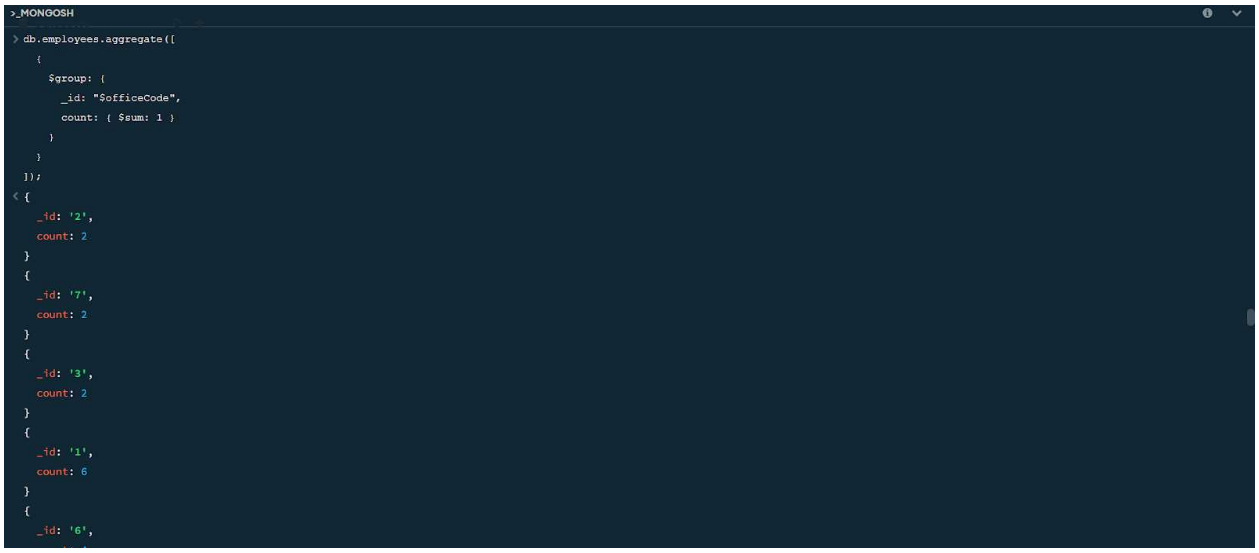
```
>_MONGOSH
> db.medicine.find({
    $and: [
      { "Medicine_Name": "sed" },
      { "Medicine_Price": { $gt: 10 } }
    ]
  })
< {
    _id: ObjectId("656cb7d15ba59034c82cf3fc"),
    Medicine_ID: 1000014,
    Pharmacy_ID: 2998367,
    Medicine_Name: 'sed',
    Medicine_Price: 99999999.99
  }
  {
    _id: ObjectId("656cb7d15ba59034c82cf3ff"),
    Medicine_ID: 1024485,
    Pharmacy_ID: 2621189,
    Medicine_Name: 'sed',
    Medicine_Price: 5052.65
  }
  {
    _id: ObjectId("656cb7d15ba59034c82cf402"),
    Medicine_ID: 1038853,
    Pharmacy_ID: 9056531,
    Medicine_Name: 'sed',
    Medicine_Price: 99999999.99
```

The query shown in the above picture would be fetching the medicine_name with “**sed**” and with a medicine_price greater than **$10.**

- 

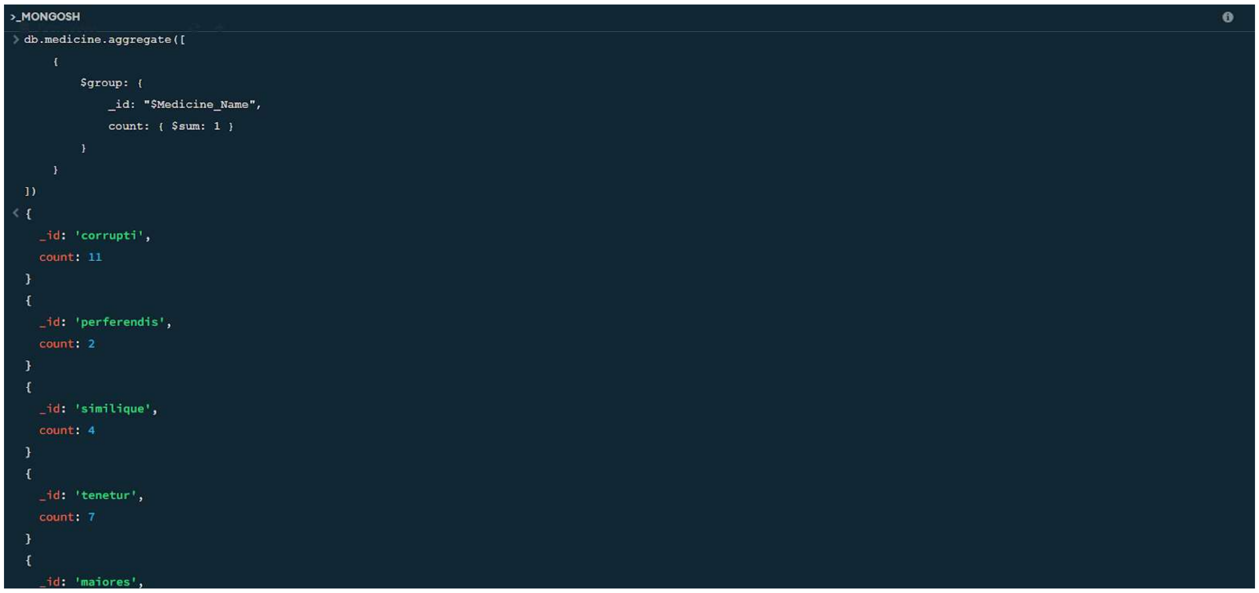
```
> db.employees.aggregate(
  [
  { $group: { _id: "$jobTitle", count: { $sum: 1 } } },
  { $sort: { count: -1 } }
  ]
  )
< {
    _id: 'Sales Rep',
    count: 17
  }
  {
    _id: 'President',
    count: 1
  }
  {
    _id: 'Sales Manager (APAC)',
    count: 1
  }
  {
    _id: 'Sales Manager (NA)',
    count: 1
  }
  {
    _id: 'VP Marketing',
    count: 1
```

The aggregated query shown in the above picture would be fetching the **number of employees** present with each individual **jobTitle** from the **employees** collection in the descending order of the number of employees per each jobTitle

- 

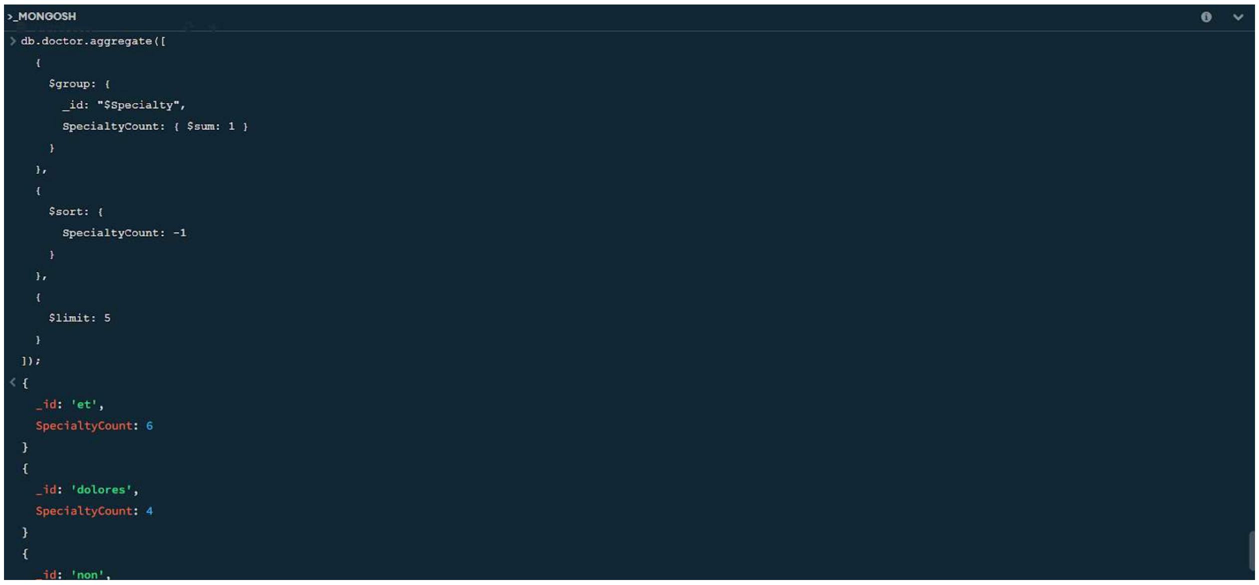
```
> db.employees.aggregate([
    {
      $group: {
        _id: "$officeCode",
        count: { $sum: 1 }
      }
    }
  ]);
< {
    _id: '2',
    count: 2
  }
  {
    _id: '7',
    count: 2
  }
  {
    _id: '3',
    count: 2
  }
  {
    _id: '1',
    count: 6
  }
  {
    _id: '6',
```

As described above, the same goes with the number of office codes present in the employees collection and the respective count.

- 

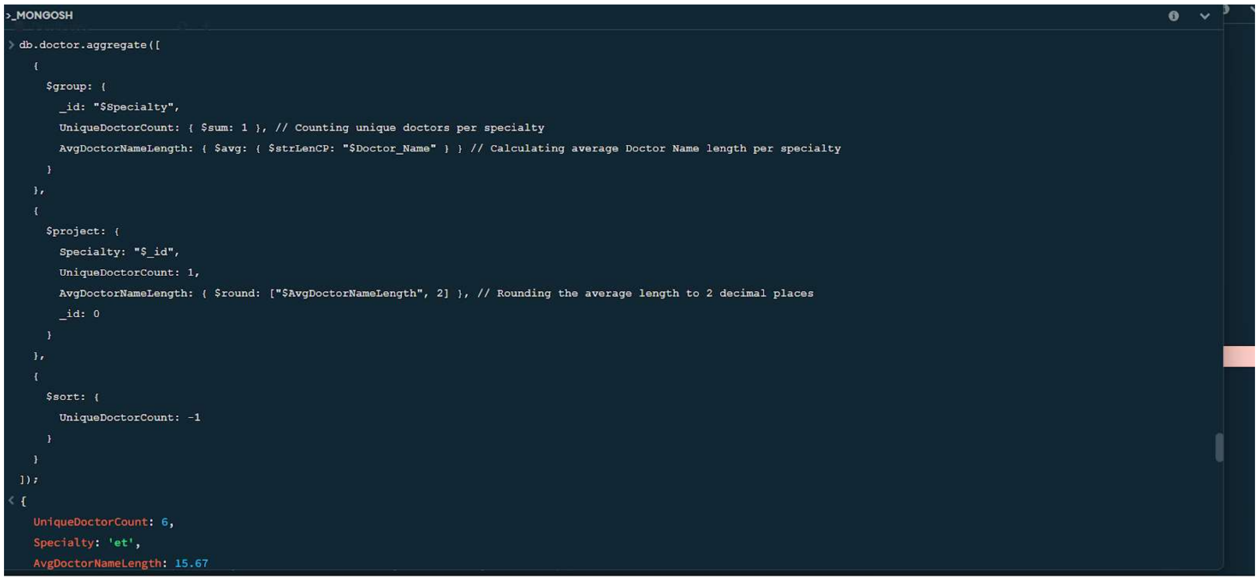
```
>_MONGOSH
> db.medicine.aggregate([
      {
          $group: {
              _id: "$Medicine_Name",
              count: { $sum: 1 }
          }
      }
  ])
< {
    _id: 'corrupti',
    count: 11
  }
  {
    _id: 'perferendis',
    count: 2
  }
  {
    _id: 'similique',
    count: 4
  }
  {
    _id: 'tenetur',
    count: 7
  }
  {
    _id: 'maiores',
```

Getting the individual medicine count from medicine collection

- 

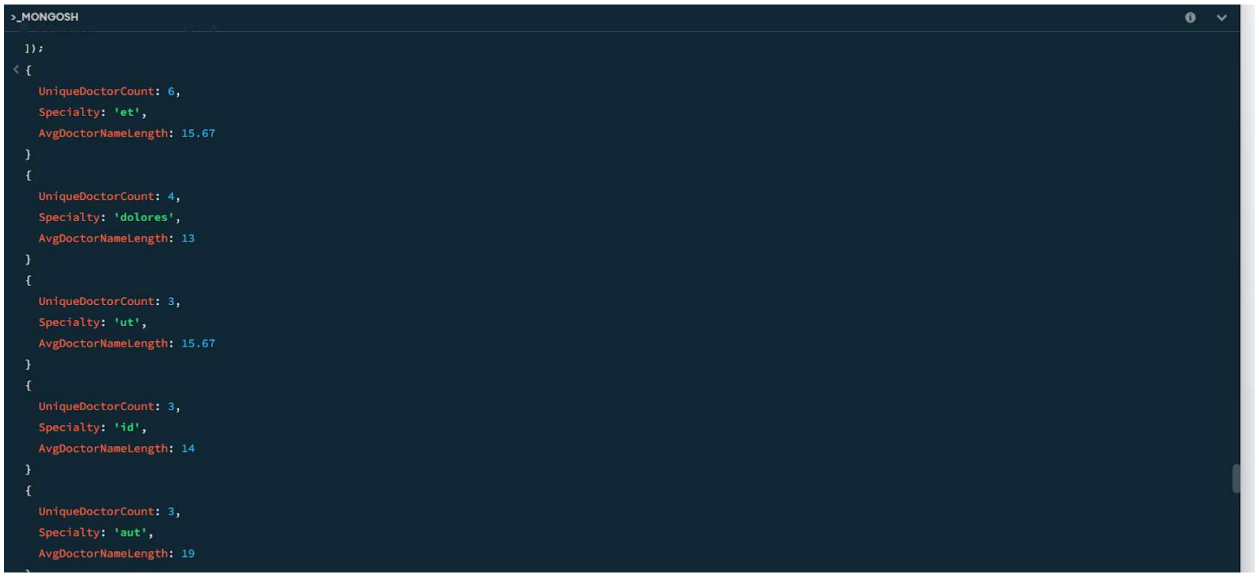
```
>_MONGOSH
> db.doctor.aggregate([
    {
      $group: {
        _id: "$Specialty",
        SpecialtyCount: { $sum: 1 }
      }
    },
    {
      $sort: {
        SpecialtyCount: -1
      }
    },
    {
      $limit: 5
    }
  ]);
< {
    _id: 'et',
    SpecialtyCount: 6
  }
  {
    _id: 'dolores',
    SpecialtyCount: 4
  }
  {
    _id: 'non',
```

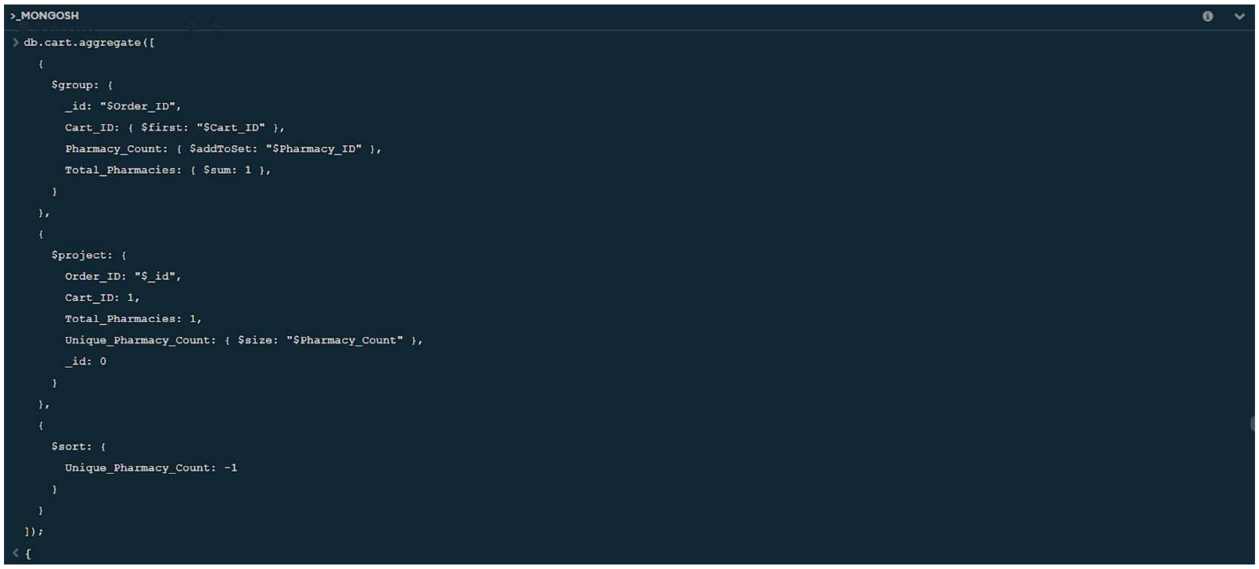
Getting the individual speciality count from the doctor collection

- 
```
>_MONGOSH
> db.doctor.aggregate([
    {
      $group: {
        _id: "$Specialty",
        UniqueDoctorCount: { $sum: 1 }, // Counting unique doctors per specialty
        AvgDoctorNameLength: { $avg: { $strLenCP: "$Doctor_Name" } } // Calculating average Doctor Name length per specialty
      }
    },
    {
      $project: {
        Specialty: "$_id",
        UniqueDoctorCount: 1,
        AvgDoctorNameLength: { $round: ["$AvgDoctorNameLength", 2] }, // Rounding the average length to 2 decimal places
        _id: 0
      }
    },
    {
      $sort: {
        UniqueDoctorCount: -1
      }
    }
  ]);
< {
    UniqueDoctorCount: 6,
    Specialty: 'et',
    AvgDoctorNameLength: 15.67
```

Finding the unique specialties among the doctors and also calculating statistics such as the average length of Doctor Names for each specialty.

Below is the output.

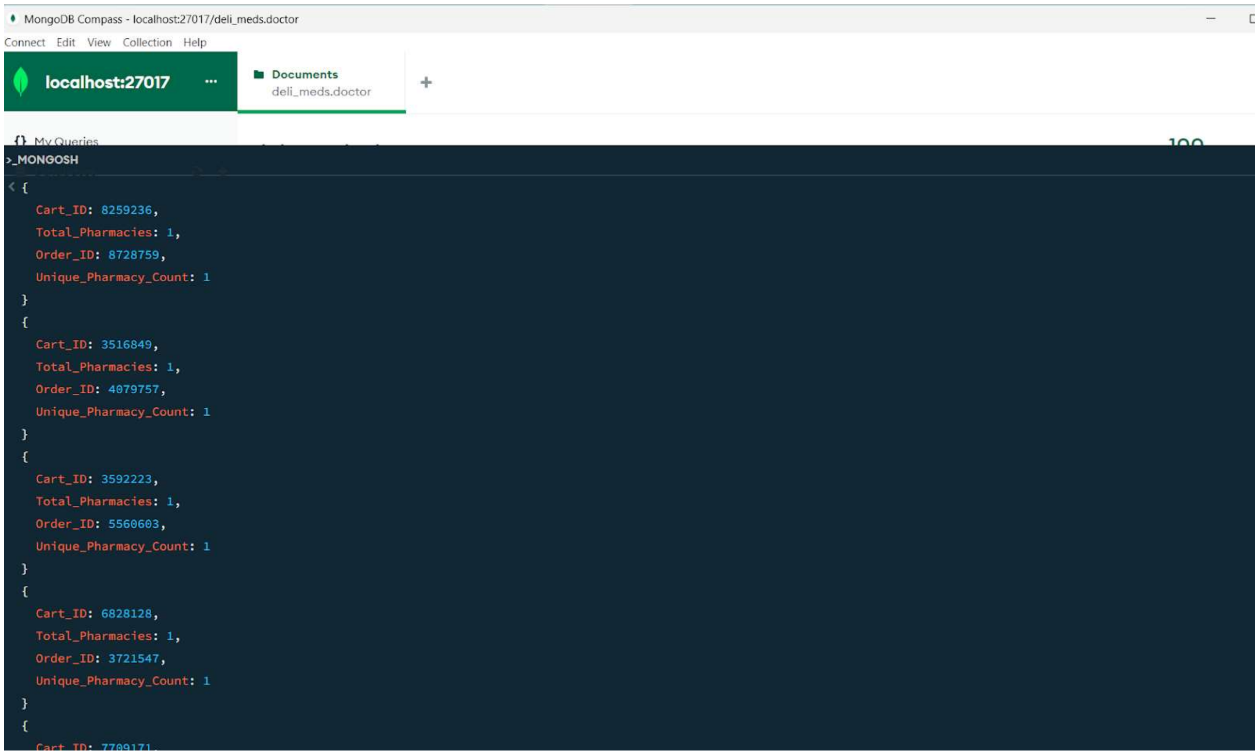
- 
```
>_MONGOSH
  ]);
< {
    UniqueDoctorCount: 6,
    Specialty: 'et',
    AvgDoctorNameLength: 15.67
  }
  {
    UniqueDoctorCount: 4,
    Specialty: 'dolores',
    AvgDoctorNameLength: 13
  }
  {
    UniqueDoctorCount: 3,
    Specialty: 'ut',
    AvgDoctorNameLength: 15.67
  }
  {
    UniqueDoctorCount: 3,
    Specialty: 'id',
    AvgDoctorNameLength: 14
  }
  {
    UniqueDoctorCount: 3,
    Specialty: 'aut',
    AvgDoctorNameLength: 19
```

- 

```
>_MONGOSH
> db.cart.aggregate([
    {
      $group: {
        _id: "$Order_ID",
        Cart_ID: { $first: "$Cart_ID" },
        Pharmacy_Count: { $addToSet: "$Pharmacy_ID" },
        Total_Pharmacies: { $sum: 1 },
      }
    },
    {
      $project: {
        Order_ID: "$_id",
        Cart_ID: 1,
        Total_Pharmacies: 1,
        Unique_Pharmacy_Count: { $size: "$Pharmacy_Count" },
        _id: 0
      }
    },
    {
      $sort: {
        Unique_Pharmacy_Count: -1
      }
    }
  ]);
< {
```

Query to find/collect the unique Pharmacy IDs per Order and count the number of unique pharmacies per order from the **cart** collection.

Below is the output.

- 

```
MongoDB Compass - localhost:27017/deli_meds.doctor
Connect  Edit  View  Collection  Help
localhost:27017   Documents deli_meds.doctor
My Queries
>_MONGOSH
< {
    Cart_ID: 8259236,
    Total_Pharmacies: 1,
    Order_ID: 8728759,
    Unique_Pharmacy_Count: 1
  }
  {
    Cart_ID: 3516849,
    Total_Pharmacies: 1,
    Order_ID: 4079757,
    Unique_Pharmacy_Count: 1
  }
  {
    Cart_ID: 3592223,
    Total_Pharmacies: 1,
    Order_ID: 5560603,
    Unique_Pharmacy_Count: 1
  }
  {
    Cart_ID: 6828128,
    Total_Pharmacies: 1,
    Order_ID: 3721547,
    Unique_Pharmacy_Count: 1
  }
  {
    Cart_ID: 7709171,
```