D597 Task 2 Scenario 1
Aaron Pacheco

Section A

1. A business problem that can be solved with a database solution is developing a way for the company to identify patients who are more at risk and ensure they are using devices best suited for their needs.

2. A NoSQL solution is ideal for this data, first because of the sheer amount of data we are handling. With data on 100000 patients it is essential that we are using a database that can easily navigate that amount of data. Secondly, health data can require certain flexibility that NoSQL can provide, which SQL cannot. In the future, if Healthtrack wants to make their data more specific, i.e. labeling specific medical conditions, rather than simply “none”, “mild”, or “watch”, the ability to input additional data about conditions would be much easier in a more flexible database.

3. A Document store NoSQL database type allows us to keep all of a patient’s data located in one easy-to-access document that will allow us to quickly assess the patient’s needs and determine if their device is best suited to them.

4. Within the database solution, the business data will first be combined by embedding each patient’s tracker data into the patient’s document, for ease of access. Then, we will assess medical conditions to identify patients that may be at an above average risk for medical issues. Then we will look at the patient’s tracker and ensure they are using a device which best suits them, meaning that it has a long battery life or that it is highly rated. We can also identify patients whose trackers do not match the trackers in our database. This will allow us to identify patients who may be in need, so we can reach out and ensure they are doing well and suggest that we can improve their tracker.

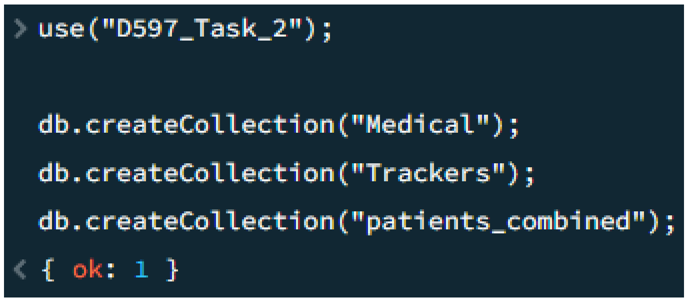
Section B

This database can be easily scaled when the need would arise. Rather than attempting to vertically scale with a more and more expensive machine, we could easily scale this data horizontally across multiple machines using sharding. If Healthtrack decides they need to improve the database as the amount of data increases, we could shard the data across multiple machines using shard keys like “Brand Name” for the Trackers collection and a hashed “patient_id” for the Medical collection. These keys have high cardinality and would likely greatly increase performance if sharded across multiple machines.

Section C

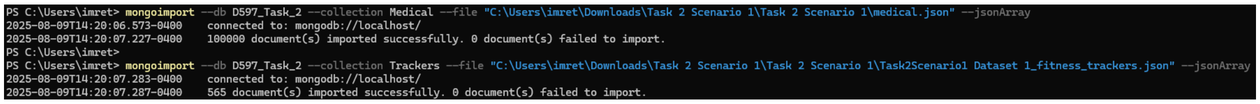
In a database of this nature, where medical data is being stored, it would be very important to store information in a secure manner. Client-side Field Level Encryption would be the ideal way to ensure that this data is encrypted, not only when it is at rest in the database, but also when it is in transit and in use within the database, ensuring that any sensitive data would be as secure as possible. In this data, I would highly recommend field level encryption be applied to the “medical_conditions”, “medications”, and “allergies” fields to ensure that all health-related data is properly secured. On top of that, it would be worth discussing with Healthtrack if other fields, such as “date_of_birth” should also be encrypted to protect clients from social engineering attacks should a data breach occur.

Section D

1. 

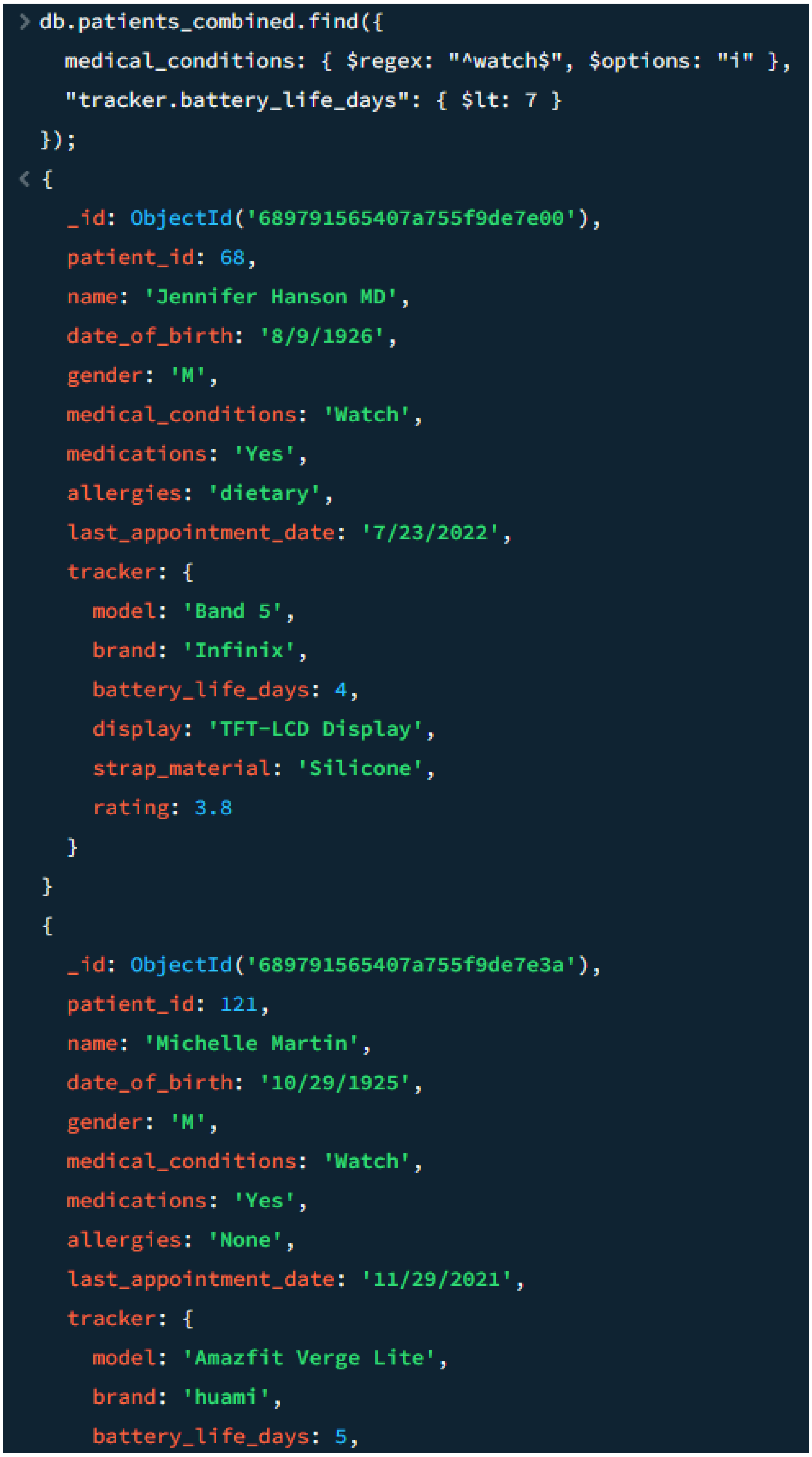
```
> use("D597_Task_2");

  db.createCollection("Medical");
  db.createCollection("Trackers");
  db.createCollection("patients_combined");
< { ok: 1 }
```

2. 

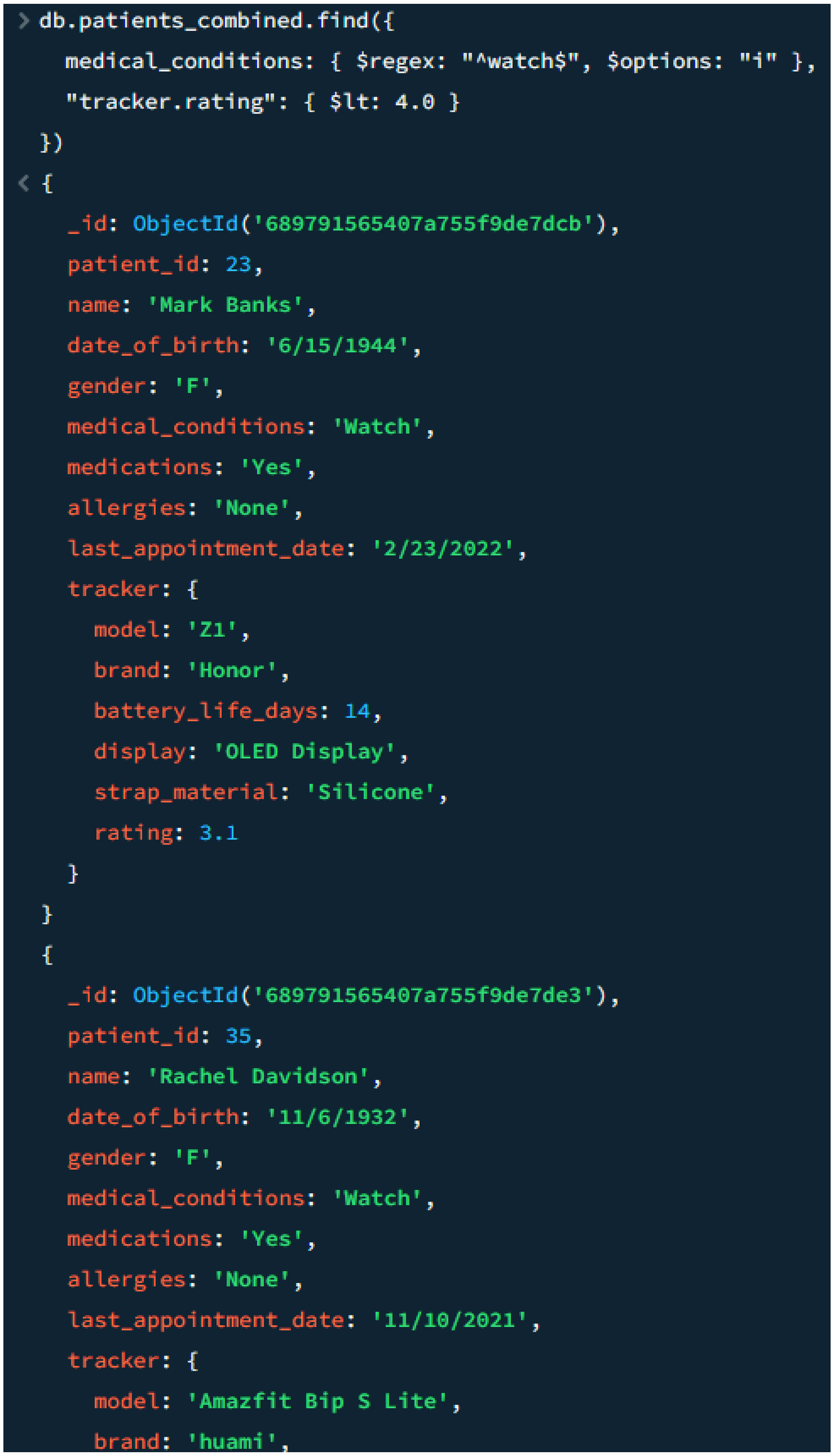
```
PS C:\Users\imret> mongoimport --db D597_Task_2 --collection Medical --file "C:\Users\imret\Downloads\Task 2 Scenario 1\Task 2 Scenario 1\medical.json" --jsonArray
2025-08-09T14:20:06.573-0400    connected to: mongodb://localhost/
2025-08-09T14:20:07.227-0400    100000 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\imret>
PS C:\Users\imret> mongoimport --db D597_Task_2 --collection Trackers --file "C:\Users\imret\Downloads\Task 2 Scenario 1\Task 2 Scenario 1\Task2Scenario1 Dataset 1_fitness_trackers.json" --jsonArray
2025-08-09T14:20:07.283-0400    connected to: mongodb://localhost/
2025-08-09T14:20:07.287-0400    565 document(s) imported successfully. 0 document(s) failed to import.
```

3. This query finds all patients with a medical condition of “watch” that have trackers with battery lives shorter than 7 days, leaving them additionally at risk.



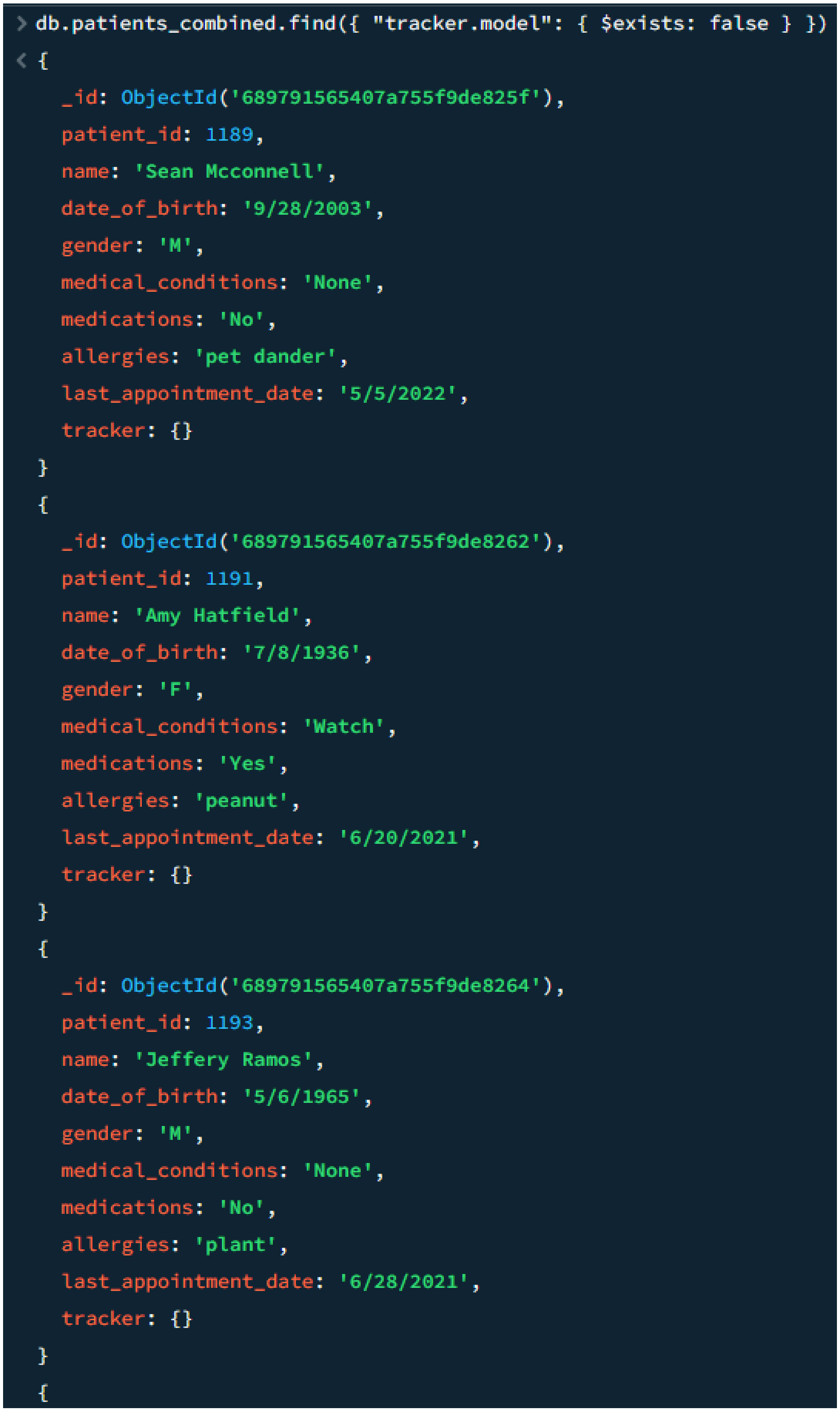
```
> db.patients_combined.find({
    medical_conditions: { $regex: "^watch$", $options: "i" },
    "tracker.battery_life_days": { $lt: 7 }
  });
< {
    _id: ObjectId('689791565407a755f9de7e00'),
    patient_id: 68,
    name: 'Jennifer Hanson MD',
    date_of_birth: '8/9/1926',
    gender: 'M',
    medical_conditions: 'Watch',
    medications: 'Yes',
    allergies: 'dietary',
    last_appointment_date: '7/23/2022',
    tracker: {
      model: 'Band 5',
      brand: 'Infinix',
      battery_life_days: 4,
      display: 'TFT-LCD Display',
      strap_material: 'Silicone',
      rating: 3.8
    }
  }
  {
    _id: ObjectId('689791565407a755f9de7e3a'),
    patient_id: 121,
    name: 'Michelle Martin',
    date_of_birth: '10/29/1925',
    gender: 'M',
    medical_conditions: 'Watch',
    medications: 'Yes',
    allergies: 'None',
    last_appointment_date: '11/29/2021',
    tracker: {
      model: 'Amazfit Verge Lite',
      brand: 'huami',
      battery_life_days: 5,
```

This query identifies patients with a medical condition of “watch” whose trackers are rated lower than a 4.0



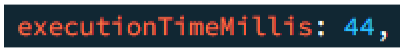
```
> db.patients_combined.find({
    medical_conditions: { $regex: "^watch$", $options: "i" },
    "tracker.rating": { $lt: 4.0 }
  })
< {
    _id: ObjectId('689791565407a755f9de7dcb'),
    patient_id: 23,
    name: 'Mark Banks',
    date_of_birth: '6/15/1944',
    gender: 'F',
    medical_conditions: 'Watch',
    medications: 'Yes',
    allergies: 'None',
    last_appointment_date: '2/23/2022',
    tracker: {
      model: 'Z1',
      brand: 'Honor',
      battery_life_days: 14,
      display: 'OLED Display',
      strap_material: 'Silicone',
      rating: 3.1
    }
  }
  {
    _id: ObjectId('689791565407a755f9de7de3'),
    patient_id: 35,
    name: 'Rachel Davidson',
    date_of_birth: '11/6/1932',
    gender: 'F',
    medical_conditions: 'Watch',
    medications: 'Yes',
    allergies: 'None',
    last_appointment_date: '11/10/2021',
    tracker: {
      model: 'Amazfit Bip S Lite',
      brand: 'huami',
```

This query identifies patients who have trackers not in our current Tracker collection

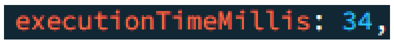


```
> db.patients_combined.find({ "tracker.model": { $exists: false } })
< {
    _id: ObjectId('689791565407a755f9de825f'),
    patient_id: 1189,
    name: 'Sean Mcconnell',
    date_of_birth: '9/28/2003',
    gender: 'M',
    medical_conditions: 'None',
    medications: 'No',
    allergies: 'pet dander',
    last_appointment_date: '5/5/2022',
    tracker: {}
  }
  {
    _id: ObjectId('689791565407a755f9de8262'),
    patient_id: 1191,
    name: 'Amy Hatfield',
    date_of_birth: '7/8/1936',
    gender: 'F',
    medical_conditions: 'Watch',
    medications: 'Yes',
    allergies: 'peanut',
    last_appointment_date: '6/20/2021',
    tracker: {}
  }
  {
    _id: ObjectId('689791565407a755f9de8264'),
    patient_id: 1193,
    name: 'Jeffery Ramos',
    date_of_birth: '5/6/1965',
    gender: 'M',
    medical_conditions: 'None',
    medications: 'No',
    allergies: 'plant',
    last_appointment_date: '6/28/2021',
    tracker: {}
  }
  {
```

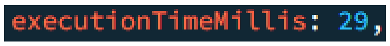
4. Before indexing any fields, below are the execution times of the three queries from D3 in order.



```
executionTimeMillis: 44,
```

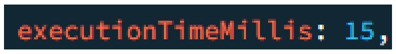


```
executionTimeMillis: 34,
```

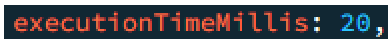


```
executionTimeMillis: 29,
```

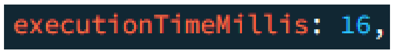
After indexing the “medical_conditions”, “battery_life_days”, “rating”, and “model” fields, the execution times of the three queries in order are below.



```
executionTimeMillis: 15,
```



```
executionTimeMillis: 20,
```



```
executionTimeMillis: 16,
```