

Johnson J. Millil

December 31, 2024

Data Management

NoSQL MongoDB for Medical/Fitness Data

Part 1: Design Document

A1. Describe a Business Problem

The two datasets can work together to address a business problem:

The medical center wants to analyze patient and fitness tracker data to gain insights into device usage and its impact on health trends. This includes identifying the most used fitness trackers, analyzing patient health conditions, and linking device features to health outcomes (WGU, 2024).

A2. Justify NoSQL Database

A NoSQL database like MongoDB is suitable because it supports semi-structured data like JSON files, enabling storage and querying of both datasets. It offers:

- **Flexibility:** Handles evolving schemas in the data (e.g., adding new fitness tracker attributes) (MongoDB, n.d.).
- **Scalability:** Supports large datasets with efficient querying (MongoDB, n.d.).
- **Data Integration:** Allows linking between patient and tracker datasets via shared fields (e.g., Tracker) (WGU, 2024).

A3. Identify a NoSQL Database Type

A **Document Database** (MongoDB) is ideal for:

- Storing nested JSON data without schema constraints (MongoDB, n.d.).
- Querying across collections (e.g., finding all patients using specific trackers) (MongoDB, n.d.).

A4. Explain How Data Will Be Used

- **Medical Dataset:** Analyze patient demographics, health trends, and appointment patterns (WGU, 2024).
- **Fitness Tracker Dataset:** Identify top-performing trackers and correlate tracker features with patient health data (MongoDB, n.d.).
- **Combined Insights:**

- Determine the most-used trackers for patients with specific health conditions (WGU, 2024).
- Analyze how device features (e.g., battery life) impact usage patterns (MongoDB, n.d.).

B. Scalability Strategies

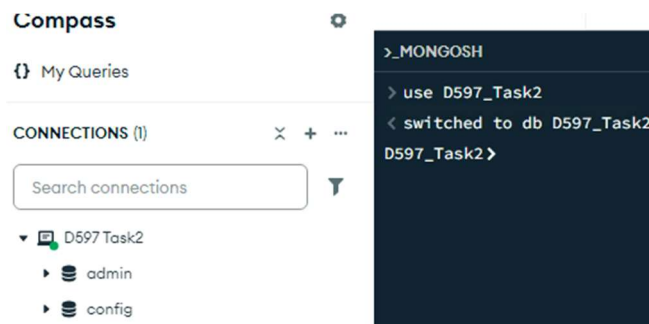
- **Sharding:** Distribute data by patient_id or Tracker for horizontal scaling (MongoDB, n.d.).
- **Indexes:** Create indexes on frequently queried fields (medical_conditions, Rating (Out of 5)) (MongoDB, n.d.).
- **Replication:** Set up replicas for fault tolerance (MongoDB, n.d.).

C. Privacy and Security Measures

- **Encryption:** Protect sensitive data at rest and in transit (WGU, 2024).
- **Access Control:** Use role-based permissions for doctors, analysts, and administrators (WGU, 2024).
- **Auditing:** Track data access and modifications (MongoDB, n.d.).

Part 2: Implementation

D1. Create Databases and Collections



D2. Import Data

Microsoft Windows [Version 10.0.20348.2402]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>cd "C:\Program Files\MongoDB\Server\7.0\bin"

C:\Program Files\MongoDB\Server\7.0\bin>dir

Volume in drive C is Windows
Volume Serial Number is E2AA-5DB5

Directory of C:\Program Files\MongoDB\Server\7.0\bin

```
04/18/2024 03:10 AM <DIR>      .
04/18/2024 03:10 AM <DIR>      ..
03/29/2024 09:37 PM          1,558 InstallCompass.ps1
04/18/2024 03:15 AM          542 mongod.cfg
03/30/2024 12:00 AM        66,338,304 mongod.exe
03/30/2024 12:00 AM    1,041,608,704 mongod.pdb
03/29/2024 11:56 PM    40,064,512 mongos.exe
03/29/2024 11:56 PM    501,851,520 mongos.pdb
               6 File(s)  1,739,865,140 bytes
               2 Dir(s)  83,714,842,624 bytes free
```

C:\Program Files\MongoDB\Server\7.0\bin>mongoimport --db medical_database --collection patients --file "C:\Users\student\Downloads\medical.json" --jsonArray

```
2024-12-27T23:00:46.909+0000 connected to: mongodb://localhost/
2024-12-27T23:00:49.909+0000 [#####] medical_database.patients 19.4MB/25.8MB (75.0%)
2024-12-27T23:00:50.840+0000 [#####] medical_database.patients 25.8MB/25.8MB (100.0%)
2024-12-27T23:00:50.840+0000 100000 document(s) imported successfully. 0 document(s) failed to import.
```

C:\Program Files\MongoDB\Server\7.0\bin>mongosh

'mongosh' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\MongoDB\Server\7.0\bin>mongoimport --db fitness_database --collection fitness_trackers --file "C:\Users\student\Downloads\Task2Scenario1 Dataset 1_fitness_trackers

```
.json" --jsonArray
2024-12-27T23:06:40.332+0000 connected to: mongodb://localhost/
2024-12-27T23:06:40.457+0000 565 document(s) imported successfully. 0 document(s) failed to import.
```

C:\Program Files\MongoDB\Server\7.0\bin>

MongoDB Compass - D597 Task2/Shell

Connections Edit View Help

Compass

{ My Queries

CONNECTIONS (1)

Search connections

▼ D597 Task2

- ▶ admin
- ▶ config
- ▼ fitness_database
 - fitness_trackers
- ▶ local
- ▼ medical_database
 - patients

> use medical_database;

< switched to db medical_database

> db.patients.count();

< DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.

< 100000

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

{ My Queries

CONNECTIONS (1)

Search connections

▼ localhost:27017

- ▶ admin
- ▶ config
- ▼ fitness_database
 - fitness_trackers
- ▶ local
- ▼ medical_database
 - patients

localhost:27017

mongosh: localhost:27017

fitness_trackers

> MONGOSH

```
}
Type "it" for more
> db.fitness_trackers.find({ "Rating (Out of 5)": { $gt: 4.5 } }).pretty();
<
> use fitness_database
< switched to db fitness_database
> db.fitness_trackers.count();
< DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
< 565
> db.fitness_trackers.find().limit(5).pretty();
< {
  _id: ObjectId('676f3300c233b1761c4730f5'),
  'Brand Name': 'Xiaomi',
  'Device Type': 'FitnessBand',
  'Model Name': 'Smart Band 5',
  Color: 'Black',
  'Selling Price': '2,469',
  'Original Price': '2,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.1,
  'Strap Material': 'Thermoplastic polyurethane',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4730f6'),
  'Brand Name': 'Xiaomi',
  'Device Type': 'Smartwatch',
  'Model Name': 'RevolveActive',
  Color: 'Black',
  'Selling Price': '12,999'
```

D3. Queries

1. Patients Using Specific Trackers:

```
> db.patients.find({ "Tracker": "Band 5" }).pretty();
< {
  _id: ObjectId('676f319f3df9dd1affad8535'),
  patient_id: 6,
  name: 'Dawn Roach',
  date_of_birth: '6/2/1967',
  gender: 'M',
  medical_conditions: 'Mild',
  medications: 'Yes',
  allergies: 'None',
  last_appointment_date: '2/2/2022',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad8537'),
  patient_id: 5,
  name: 'David Scott',
  date_of_birth: '12/7/2015',
  gender: 'M',
  medical_conditions: 'Mild',
  medications: 'Yes',
  allergies: 'None',
  last_appointment_date: '5/17/2021',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad8543'),
  patient_id: 7,
  name: 'Mary Harris',
  date_of_birth: '8/27/2013',
  gender: 'M',
  medical_conditions: 'Mild',
  medications: 'Yes',
  allergies: 'None',
  last_appointment_date: '6/2/2021',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad8572'),
  patient_id: 67,
  name: 'Chelsea Davis',
  date_of_birth: '5/25/2008',
  gender: 'F',
  medical_conditions: 'None',
  last_appointment_date: '7/17/1998',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad8574'),
  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: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad8576'),
  patient_id: 69,
  name: 'Catherine Singleton',
  date_of_birth: '7/17/1998',
  Tracker: 'Band 5'
}
{
  gender: 'M',
  medical_conditions: 'None',
  medications: 'No',
  allergies: 'pet dander',
  last_appointment_date: '12/22/2022',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad85ad'),
  patient_id: 130,
  name: 'Patricia Wright',
  date_of_birth: '9/17/1968',
  gender: 'M',
  medical_conditions: 'None',
  medications: 'No',
  allergies: 'None',
  last_appointment_date: '8/23/2022',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad85ae'),
  patient_id: 131,
  last_appointment_date: '5/8/2022',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad85ba'),
  patient_id: 129,
  name: 'Theresa Sanchez',
  date_of_birth: '2/18/2016',
  gender: 'F',
  medical_conditions: 'None',
  medications: 'No',
  allergies: 'None',
  last_appointment_date: '7/24/2022',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad85ea'),
  patient_id: 191,
  name: 'Mrs. Melissa Brown',
  date_of_birth: '11/27/1952',
  gender: 'M',
  medical_conditions: 'Watch',
  medications: 'Yes',
  allergies: 'peanut',
  last_appointment_date: '11/13/2022',
  Tracker: 'Band 5'
}
{
  _id: ObjectId('676f319f3df9dd1affad85eb'),
  patient_id: 192,
  name: 'Matthew Gaines',
  date_of_birth: '8/11/1983',
  gender: 'M',
  medical_conditions: 'None',
  medications: 'No',
  allergies: 'egg',
  last_appointment_date: '1/18/2022',
  Tracker: 'Band 5'
}
```

```

    _id: ObjectId('676f319f3df9dd1affad85ec'),
    patient_id: 193,
    name: 'Kelly Thomas',
    date_of_birth: '11/5/1929',
    gender: 'M',
    medical_conditions: 'Watch',
    medications: 'Yes',
    allergies: 'plant',
    last_appointment_date: '12/30/2022',
    Tracker: 'Band 5'
  }
]
{
  _id: ObjectId('676f319f3df9dd1affad8629'),
  patient_id: 253,
  name: 'Nicholas Mcdaniel',
  date_of_birth: '11/9/1993',
  gender: 'F',
  medical_conditions: 'None',
  medications: 'No',
  allergies: 'None',
  last_appointment_date: '4/4/2022',
  Tracker: 'Band 5'
}

    _id: ObjectId('676f319f3df9dd1affad862a'),
    patient_id: 254,
    name: 'Gregory Brooks',
    date_of_birth: '6/29/2005',
    gender: 'F',
    medical_conditions: 'None',
    medications: 'No',
    allergies: 'None',
    last_appointment_date: '1/5/2023',
    Tracker: 'Band 5'
  }
]
{
  _id: ObjectId('676f319f3df9dd1affad8643'),
  patient_id: 255,
  name: 'David Watkins',
  date_of_birth: '12/31/1990',
  gender: 'F',
  medical_conditions: 'None',
  medications: 'No',
  allergies: 'None',
  last_appointment_date: '12/25/2022',
  Tracker: 'Band 5'
}

    _id: ObjectId('676f319f3df9dd1affad8668'),
    patient_id: 316,
    name: 'David Ferguson',
    date_of_birth: '11/14/1941',
    gender: 'F',
    medical_conditions: 'Watch',
    medications: 'Yes',
    allergies: 'None',
    last_appointment_date: '9/13/2022',
    Tracker: 'Band 5'
  }
]
{
  _id: ObjectId('676f319f3df9dd1affad866a'),
  patient_id: 317,
  name: 'Alicia Turner',
  date_of_birth: '8/26/1932',
  gender: 'M',
  medical_conditions: 'Watch',
  medications: 'Yes',
  allergies: 'None',
  last_appointment_date: '5/5/2021',
  Tracker: 'Band 5'
}

    _id: ObjectId('676f319f3df9dd1affad866c'),
    patient_id: 315,
    name: 'John Howard',
    date_of_birth: '4/5/1935',
    gender: 'M',
    medical_conditions: 'Watch',
    medications: 'Yes',
    allergies: 'None',
    last_appointment_date: '12/18/2021',
    Tracker: 'Band 5'
  }
]
{
  _id: ObjectId('676f319f3df9dd1affad86a8'),
  patient_id: 377,
  name: 'Benjamin Wilson',
  date_of_birth: '2/6/2017',
  gender: 'F',
  medical_conditions: 'None',
  medications: 'No',
  allergies: 'None',
  last_appointment_date: '5/31/2021',
  Tracker: 'Band 5'
}

    date_of_birth: '2/6/2017',
    gender: 'F',
    medical_conditions: 'None',
    medications: 'No',
    allergies: 'None',
    last_appointment_date: '5/31/2021',
    Tracker: 'Band 5'
  }
]
{
  _id: ObjectId('676f319f3df9dd1affad86aa'),
  patient_id: 378,
  name: 'Kathleen Rice',
  date_of_birth: '9/20/1970',
  gender: 'M',
  medical_conditions: 'None',
  medications: 'No',
  allergies: 'None',
  last_appointment_date: '6/13/2021',
  Tracker: 'Band 5'
}
]
Type "it" for more
medical_database>

```


2. Top-Rated Fitness Trackers:

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

Search connections

- localhost:27017
 - admin
 - config
 - fitness_database
 - fitness_trackers
 - local
 - medical_database
 - patients

```
>_MONGOSH
> use fitness_database
< switched to db fitness_database
> db.fitness_trackers.find({ "Rating (Out of 5)": { $gt: 4.5 } }).pretty();
< {
  "_id": ObjectId('676f3300c233b1761c473103'),
  'Brand Name': 'FitBit',
  'Device Type': 'FitnessBand',
  'Model Name': 'Lunar',
  Color: 'White',
  'Selling Price': '10,899',
  'Original Price': '10,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.7,
  'Strap Material': 'Silicone',
  'Average Battery Life (in days)': 7,
  Reviews: ''
}
{
  "_id": ObjectId('676f3300c233b1761c4731a2'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Classic 4',
  Color: 'Black',
  'Selling Price': '32,959',
  'Original Price': '37,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
```

```
{
  "_id": ObjectId('676f3300c233b1761c4731a5'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Classic 4',
  Color: 'Silver',
  'Selling Price': '40,000',
  'Original Price': '42,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  "_id": ObjectId('676f3300c233b1761c4731a6'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Classic 4',
  Color: 'Silver',
  'Selling Price': '34,999',
  'Original Price': '37,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  "_id": ObjectId('676f3300c233b1761c4731a7'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Classic 4 LTE',
  Color: 'Black',
  'Selling Price': '28,999',
  'Original Price': '31,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 5,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  "_id": ObjectId('676f3300c233b1761c4731aa'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Watch 4 Classic LTE',
  Color: 'Black',
  'Selling Price': '40,000',
  'Original Price': '42,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  "_id": ObjectId('676f3300c233b1761c4731ac'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Watch 4 LTE',
  Color: 'Black',
  'Selling Price': '31,999',
  'Original Price': '34,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  "_id": ObjectId('676f3300c233b1761c4731ae'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Watch 4 LTE',
  Color: 'Pink',
  'Selling Price': '28,999',
  'Original Price': '31,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
```

```

{
  _id: ObjectId('676f3300c233b1761c4731b3'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Classic 4 LTE',
  Color: 'Silver, Green',
  'Selling Price': '36,999',
  'Original Price': '39,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731b6'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy Classic 4',
  Color: 'Blue',
  'Selling Price': '34,999',
  'Original Price': '34,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Silicone',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731b7'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy LTE',
  Color: 'Silver',
  'Selling Price': '31,999',
  'Original Price': '31,999',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731c5'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy 42',
  Color: 'Black',
  'Selling Price': '24,989',
  'Original Price': '25,990',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731cc'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Galaxy 42',
  Color: 'Grey',
  'Selling Price': '21,989',
  'Original Price': '25,990',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731d5'),
  'Brand Name': 'SAMSUNG',
  'Device Type': 'Smartwatch',
  'Model Name': 'Gear R730V S2',
  Color: 'Grey',
  'Selling Price': '18,532',
  'Original Price': '22,498',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Elastomer',
  'Average Battery Life (in days)': 14,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731df'),
  'Brand Name': 'FOSSIL',
  'Device Type': 'Smartwatch',
  'Model Name': 'Carlie Hybrid',
  Color: 'Gold',
  'Selling Price': '13,495',
  'Original Price': '13,495',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Stainless Steel',
  'Average Battery Life (in days)': 12,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731e1'),
  'Brand Name': 'FOSSIL',
  'Device Type': 'Smartwatch',
  'Model Name': 'Neutra Hybrid',
  Color: 'Silver',
  'Selling Price': '13,495',
  'Original Price': '13,495',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 5,
  'Strap Material': 'Stainless Steel',
  'Average Battery Life (in days)': 12,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731e4'),
  'Brand Name': 'FOSSIL',
  'Device Type': 'Smartwatch',
  'Model Name': 'Jacqueline',
  Color: 'Gold',
  'Selling Price': '13,495',
  'Original Price': '13,495',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Leather',
  'Average Battery Life (in days)': 1,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731e6'),
  'Brand Name': 'FOSSIL',
  'Device Type': 'Smartwatch',
  'Model Name': 'FTW5012 Hybrid',
  Color: 'Brown',
  'Selling Price': '9,097',
  'Original Price': '12,995',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Leather',
  'Average Battery Life (in days)': 2,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731e7'),
  'Brand Name': 'FOSSIL',
  'Device Type': 'Smartwatch',
  'Model Name': 'Q Annette',
  Color: 'Gold',
  'Selling Price': '6,995',
  'Original Price': '13,495',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Leather',
  'Average Battery Life (in days)': 1,
  Reviews: ''
}
{
  _id: ObjectId('676f3300c233b1761c4731e8'),
  'Brand Name': 'FOSSIL',
  'Device Type': 'Smartwatch',
  'Model Name': 'Q Neely',
  Color: 'Black',
  'Selling Price': '6,995',
  'Original Price': '12,995',
  Display: 'AMOLED Display',
  'Rating (Out of 5)': 4.6,
  'Strap Material': 'Leather',
  'Average Battery Life (in days)': 2,
  Reviews: ''
}

```

3. Trackers Linked to Specific Conditions:

My Queries

CONNECTIONS (1)

Search connections

- ▼ D597 Task2
 - ▶ admin
 - ▶ config
 - ▼ fitness_database
 - fitness_trackers
 - ▶ local
 - ▼ medical_database
 - patients

```
>_MONGOSH
> db.patients.aggregate([
  { $match: { "medical_conditions": "Mild" } },
  { $group: { _id: "$Tracker", count: { $sum: 1 } } }
]);
< {
  _id: 'Band Z1',
  count: 1
}
{
  _id: 'Delta',
  count: 1
}
{
  _id: '41mm',
  count: 1
}
{
  _id: 'band 4 running',
  count: 1
}
{
  _id: 'Band 5i',
  count: 1
}
{
  _id: 'S',
  count: 1
}
}
```

CONNECTIONS (1)

Search connections

- ▼ D597 Task2
 - ▶ admin
 - ▶ config
 - ▼ fitness_database
 - fitness_trackers
 - ▶ local
 - ▼ medical_database
 - patients

```
  _id: 'Band 5i',
  count: 1
}
{
  _id: 'S',
  count: 1
}
{
  _id: 'Band 5',
  count: 3
}
{
  _id: '46 mm',
  count: 1
}
{
  _id: '2S',
  count: 1
}
{
  _id: 'Watch Magic',
  count: 1
}
{
  _id: 'Xplorer',
  count: 1
}
}
medical_database>
```


D4. Optimization

1. Create Indexes:

```
> db.patients.createIndex({ "medical_conditions": 1 });
db.fitness_trackers.createIndex({ "Rating (Out of 5)": 1 });
< Rating (Out of 5)_1
medical_database>
```

2. Test Query Performance:

- ▼ D597 Task2
 - ▶ admin
 - ▶ config
 - ▼ fitness_database
 - fitness_trackers
 - ▶ local
 - ▼ medical_database
 - patients

```
< Rating (Out of 5)_1
> db.patients.find({ "medical_conditions": "Mild" }).explain("executionStats");
< {
  explainVersion: '1',
  queryPlanner: {
    namespace: 'medical_database.patients',
    indexFilterSet: false,
    parsedQuery: {
      medical_conditions: {
        '$eq': 'Mild'
      }
    },
    queryHash: '1B30A0F0',
    planCacheKey: '8E922E3A',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: {
          medical_conditions: 1
        },

```

```

>_MONGOSH
  {
    indexName: 'medical_conditions_1',
    isMultiKey: false,
    multiKeyPaths: {
      medical_conditions: []
    },
    isUnique: false,
    isSparse: false,
    isPartial: false,
    indexVersion: 2,
    direction: 'forward',
    indexBounds: {
      medical_conditions: [
        '["Wild", "Wild"]'
      ]
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 13,
    executionTimeMillis: 3,
    totalKeysExamined: 13,
    totalDocsExamined: 13,
    executionStages: {
      stage: 'FETCH',
      nReturned: 13,
      executionTimeMillisEstimate: 0,
      works: 14,
      advanced: 13,
      needTime: 0,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      keyPattern: {
        medical_conditions: 1
      },
      indexName: 'medical_conditions_1',
      isMultiKey: false,
      multiKeyPaths: {
        medical_conditions: []
      },
      isUnique: false,
      isSparse: false,
      isPartial: false,
      indexVersion: 2,
      direction: 'forward',
      indexBounds: {
        medical_conditions: [
          '["Wild", "Wild"]'
        ]
      },
      keysExamined: 13,
      seeks: 1,
      dupsTested: 0,
      dupsDropped: 0
    }
  },
  command: {
    find: 'patients',
    filter: {},
    medical_conditions: 'Wild'
  },
  '$db': 'medical_database'
},
serverInfo: {
  host: 'remote-desktop-',
  port: 27017,
  version: '7.0.8',
  gitVersion: 'c5d33e55ba38d98e2f48765ec4e55338d67a4a64'
},
gitVersion: 'c5d33e55ba38d98e2f48765ec4e55338d67a4a64'
},
serverParameters: {
  internalQueryFacetBufferSizeBytes: 104857600,
  internalQueryFacetMaxOutputDocSizeBytes: 104857600,
  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
  internalDocumentSourceGroupMaxMemoryBytes: 104857600,
  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
  internalQueryProhibitBlockingMergeOnMongoS: 0,
  internalQueryMaxAddToSetBytes: 104857600,
  internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
  internalQueryFrameworkControl: 'trySbeRestricted'
},
ok: 1
}

```

References

- MongoDB. (n.d.). *Indexing strategies*. Retrieved from <https://www.mongodb.com/docs/manual/indexes/>
- MongoDB. (n.d.). *Aggregation framework*. Retrieved from <https://www.mongodb.com/docs/manual/aggregation/>
- WGU. (2024). *Data Management D597 Course Materials*. Western Governors University.
- JSON.org. (n.d.). *Introducing JSON*. Retrieved from <https://www.json.org/>