**Nicole Gallo**

**D597 – Database Management**

**June 10, 2025**

**MKN1 Task 2: Non-Relational Database Design and Implementation**

Table of Contents

[Part 1: Design Document 2](#_Toc200652907)

[A1: Business Problem 2](#_Toc200652908)

[A2: Proposal for a NoSQL Database 2](#_Toc200652909)

[A3: Justification for NoSQL Database 2](#_Toc200652910)

[A4: Business Data Usage 2](#_Toc200652911)

[B: Logical Data Model for Storage 3](#_Toc200652912)

[C: Database Collections and File Attributes 4](#_Toc200652913)

[D: Scalability 4](#_Toc200652914)

[E: Privacy and Security Measures 4](#_Toc200652915)

[Part 2: Implementation 5](#_Toc200652916)

[F. Implement the proposed database design in MongoDB 5](#_Toc200652917)

[F1 - Write script to create a database named “D597 Task 2” and its respective collections (patients and fitness trackers) 5](#_Toc200652918)

[F2 - Write script to insert the data records from JSON file into the database (fitness\_trackers.json and medical.json) 5](#_Toc200652919)

[F3 - Write script for **three** queries to solve the identified business problem 9](#_Toc200652920)

[F4 – Optimization Techniques to improve runtime with Indexes 13](#_Toc200652921)

[Part 3: Presentation 27](#_Toc200652922)

[H: Citations 27](#_Toc200652923)

# **Part 1: Design Document**

## **A1: Business Problem**

A healthcare clinic provides wearable fitness trackers to patients for remote monitoring of their activity levels and general well-being. The clinic has acquired multiple fitness tracker brands and models but lacks a centralized system to evaluate their effectiveness.

Clinicians want to assess device performance based on patient profiles (e.g., age, condition, medication use) and make informed decisions about which tracker to recommend. They need to analyze battery life, rating, and display types of the trackers used by different patient groups to optimize healthcare outcomes and resource allocation.

## **A2: Proposal for a NoSQL Database**

A non-relational JSON-based structure is ideal for this problem due to the flexible and nested nature of fitness tracker specifications and diverse patient metadata. The data structure will include two main collections:

1. fit\_trackers: Each document contains detailed specifications of wearable devices, such as brand, display type, rating, and battery life.
2. patients: Each document stores patient data including demographics, medical background, and the tracker they use.

These collections can be joined via the **\_id** attribute (and possibly incorporating the **tracker name**), allowing the separate datasets to reference one another for further analysis.

## **A3: Justification for NoSQL Database**

A database solution enables clinicians to:

* Quickly retrieve tracker performance stats by patient condition or age group
* Analyze device usage trends over time
* Automate recommendations using query logic

A NoSQL document store is especially justified here due to its support for semi-structured data, high scalability, and ease of importing JSON files directly.

## **A4: Business Data Usage**

The fitness tracker data will be used to extract insights about each device's cost, battery life, and rating. The patient dataset will provide demographic and medical information, allowing for:

* Filtering trackers by user satisfaction (rating)
* Grouping devices by effectiveness in different medical contexts
* Identifying cost-effective devices for bulk purchasing

## **B: Logical Data Model for Storage**

**Collection 1: fit\_trackers**

{

"\_id": {

"$oid": "68486cbdf4687d58595c5cdd"

},

"Brand\_Name": "Xiaomi",

"Device\_Type": "FitnessBand",

"Model\_Name": "Smart Band 4",

"Color": "Black",

"Selling\_Price": "2,099",

"Original\_Price": "2,499",

"Display": "AMOLED Display",

"Rating": 4.2,

"Strap\_Material": "Thermoplastic polyurethane",

" Battery\_Life\_Days": 14,

"Reviews": ""

}

**Collection 2: patients**

{

"\_id": {

"$oid": "68486cdaab2670c2b818af7f"

},

"patient\_id": 9,

"name": "Kenneth Johnson",

"date\_of\_birth": "4/15/2011",

"gender": "F",

"medical\_conditions": "Mild",

"medications": "Yes",

"allergies": "None",

"last\_appointment\_date": "3/12/2023",

"Tracker\_Model": "Band 5i"

}

## **C: Database Collections and File Attributes**

**Collections:**

* fit\_trackers
* patients

**Key Attributes:**

* Fit Tracker attributes: model\_name, brand\_name, device\_type, display, battery\_life\_days, rating, selling\_price, color, original\_price, strap\_material, reviews
* Patient attributes: patient\_id, name, date\_of\_birth, gender, medical\_conditions, tracker\_model, medications, allergies, last\_appointment\_date

## **D: Scalability**

**The use of MongoDB ensures horizontal scalability, high write throughput, and flexible schema evolution as new devices, more attributes, and future patients are added. Indexing on tracker\_model and patient\_id supports efficient queries.**

**Strategies for scalability are as follows:**

* **Sharding by tracker\_model to distribute workload**
* **Caching frequent queries such as finding the avg battery life per medical condition**
* **Aggregation pipelines for real-time analytics**

## **E: Privacy and Security Measures**

To protect patient and device data:

* **Encryption at rest** and **in transit**
* Role-based access control (RBAC): clinicians can query data; admins can update
* Audit logs for all data access
* Tokenization or anonymization of personally identifiable information (PII)
* Compliance with HIPAA standards

# **Part 2: Implementation**

## **F. Implement the proposed database design in MongoDB**

### F1 - Write script to create a database named “D597 Task 2” and its respective collections (patients and fitness trackers)

A screen shot of a computer code

AI-generated content may be incorrect.

### F2 - Write script to insert the data records from JSON file into the database (fitness\_trackers.json and medical.json)

**Fitness Trackers**

***Using the terminal & bash$ to import datasets***

mongoimport \

--jsonArray \

--db D597\_Task2 \

--collection fit\_trackers \

--file /Users/nicolegallo/Desktop/WGU/D597/Task\_2\_Scenario1/Task2Scenario1\_Dataset1\_fitness\_trackers.json

A computer code with black text

AI-generated content may be incorrect.

**Patients**

mongoimport \

--jsonArray \

--db D597\_Task2 \

--collection patients \

--file /Users/nicolegallo/Desktop/WGU/D597/Task\_2\_Scenario1/medical.json

A close-up of a computer code

AI-generated content may be incorrect.

**MongoDB Compass UI Screenshots of Imported Data**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

### F3 - Write script for **three** queries to solve the identified business problem

Query #1 - Average Rating by Tracker Model for Medicated Patients

db['patients'].aggregate([

{ $match: { medications: "Yes" } },

{ $lookup: {

from: "fit\_trackers",

localField: "tracker\_model",

foreignField: "model\_name",

as: "tracker"

}},

{ $unwind: "$tracker" },

{ $group: {

\_id: "$tracker.model\_name",

avg\_rating: { $avg: "$tracker.rating" }

}}

])

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect. A screen shot of a computer program

AI-generated content may be incorrect.

Query #2 - Most Common Tracker Among Patients with MILD Conditions

db['patients'].aggregate([

{ $match: { medical\_conditions: "Mild" } },

{ $group: {

\_id: "$tracker\_model",

count: { $sum: 1 }

}},

{ $sort: { count: -1 } },

{ $limit: 1 }

])



Query #3 - Total Cost of Devices Assigned to Each Gender

db['patients'].aggregate([

{ $lookup: {

from: "fit\_trackers",

localField: "tracker\_model",

foreignField: "model\_name",

as: "device"

}},

{ $unwind: "$device" },

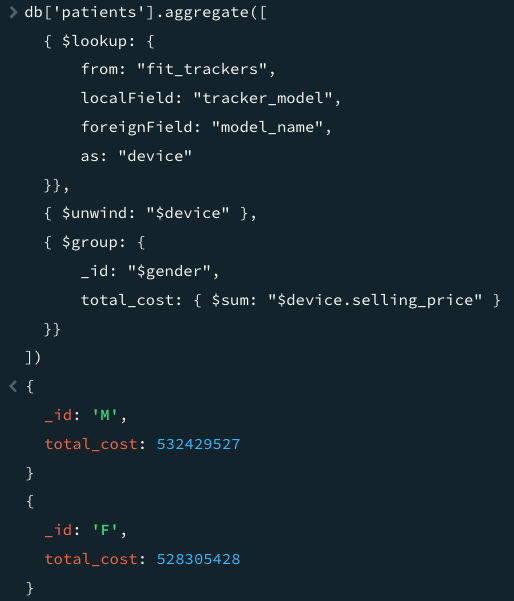
{ $group: {

\_id: "$gender",

total\_cost: { $sum: "$device.selling\_price" }

}}

])



### F4 – Optimization Techniques to improve runtime with Indexes

Creating indexes on model\_name, tracker\_model, medical\_conditions, medications, and gender.

db.fit\_trackers.createIndex({ model\_name: 1 })

db.patients.createIndex({ tracker\_model: 1 })

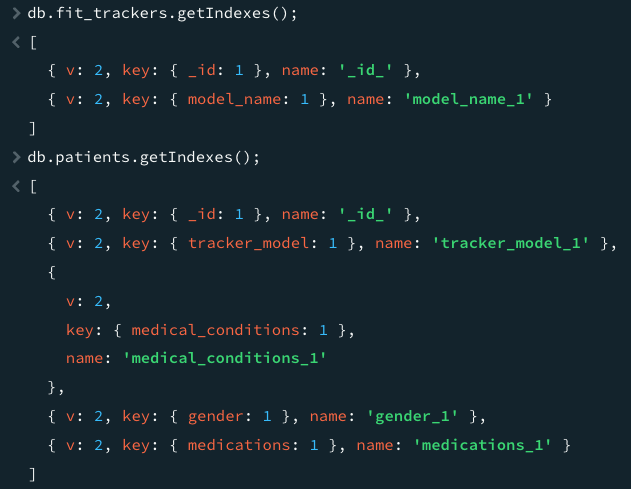
db.patients.createIndex({ medical\_conditions: 1 })

db.patients.createIndex({ gender: 1 })

db.patients.createIndex({ medications: 1 })

A screen shot of a computer code

AI-generated content may be incorrect.



To test before and after execution runtime metrics, we can use .explain("executionStats").

#### Query #1 - Average Rating by Tracker Model for Medicated Patients

Before index – executionTimeMillis: 6234

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

After index – executionTimeMillis: 908

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

#### Query #2 - Most Common Tracker Among Patients with MILD Conditions

Before index – executionTimeMillis: 62A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

After index – executionTimeMillis: 0

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

#### Query #3 - Total Cost of Devices Assigned to Each Gender

Before index – executionTimeMillis: 16204

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

After index – executionTimeMillis: 2417

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

# **Part 3: Presentation**

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6c59dfcd-23ed-442c-8b53-b2f9011b9b1e>

## **H: Citations**

* MongoDB documentation: <https://www.mongodb.com/docs/manual/>
* HIPAA privacy guidance: <https://www.hhs.gov/hipaa>