The business problem involves integrating fitness tracker data with patient medical records to enhance healthcare monitoring and patient outcomes. Key challenges include tracking health metrics, personalized recommendations, efficient data access, and dynamic updates. *Course* |. (n.d.).

A NoSQL database is ideal for this problem because it offers flexible data models, real-time processing, scalability, and data correlation. A document-based NoSQL database like MongoDB is recommended for data representation, indexing, and aggregation frameworks. MongoDB. (n.d.).

The business data will be used within the database for healthcare monitoring, alerts and notifications, healthcare insights, custom recommendations, and appointment optimization.

MongoDB supports horizontal scaling, so dynamic data growth can scale as more patients and fitness devices are added. Then, there is distributed storage, and data can automatically partition across servers, ensuring even distribution and preventing performance bottlenecks. MongoDB. (n.d.).

This document-based structure allows the database to adapt to changes in data requirements. New attributes (additional health metrics or device types) can be added to JSON documents without requiring schema migration, reducing downtime. The flexible scheme supports various data formats from fitness trackers and medical systems, enabling seamless integration. MongoDB. (n.d.).

Indexing and write optimization will handle high-volume read/write loads. Frequently accessed fields, such as patient_id, Tracker, and last_appointment_date, will be indexed for rapid querying. MongoDB supports write-intensive workloads. MongoDB. (n.d.).

Data can be partitioned by Patient ID, ensuring each patient's data is localized for efficient queries and updates. By device type, this will allow workloads to be isolated and optimized. *UpGuard*. (n.d.).

Our database design will allow replication sets to prevent data loss and ensure data durability by acknowledging writes by multiple nodes. This ensures data consistency even under heavy loads. *UpGuard*. (n.d.).

Implementing real-time data handling so MongoDB can monitor changes and trigger-time alerts, such as for irregular activity or missed patient appointments. By having an event-driven architecture, we can scale to handle real-time updates from fitness devices without affecting query performance. MongoDB. (n.d.).

Tools like MongoDB Atlas can track database performance and identify bottlenecks. We will also implement automated scaling rules based on usage patterns, ensuring the system remains responsive during peak loads. MongoDB. (n.d.).

To ensure the privacy of our database, I have proposed the following implementations and solutions: data encryption using AES-256 and Transport Layer Security for data transmitted

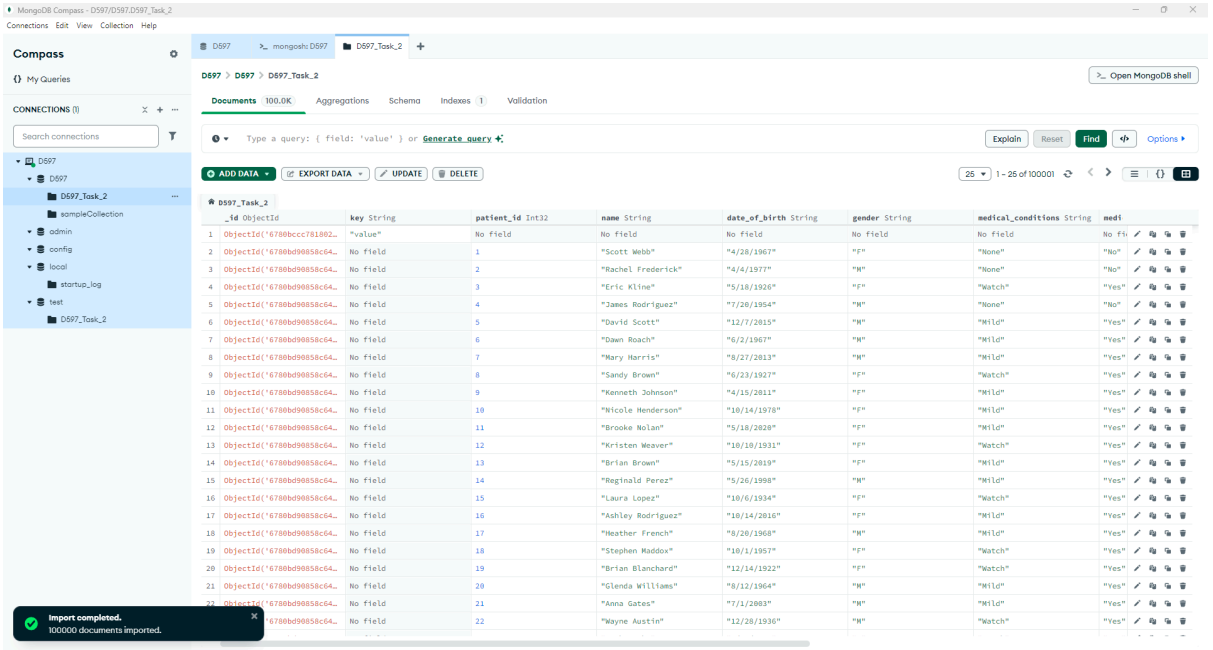
between applications, devices, and database servers. Ahmad, K. (2022, June 20). We will define roles and permissions to ensure that users and applications can only access the data necessary for their function.

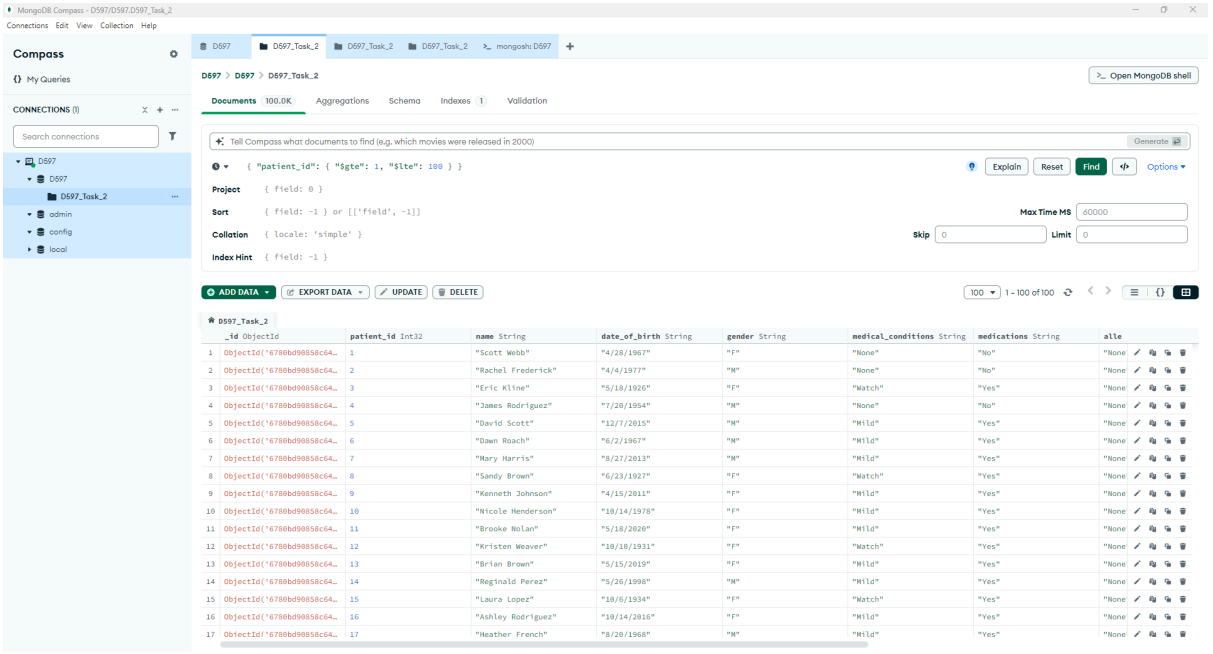
We will also implement multi-factor authentication to access the database. GeeksforGeeks, (2024, May 23). I will implement masking and anonymization to hide sensitive fields such as patient names and IDs and remove identifiable information for datasets used in research or analytics to protect patient privacy.

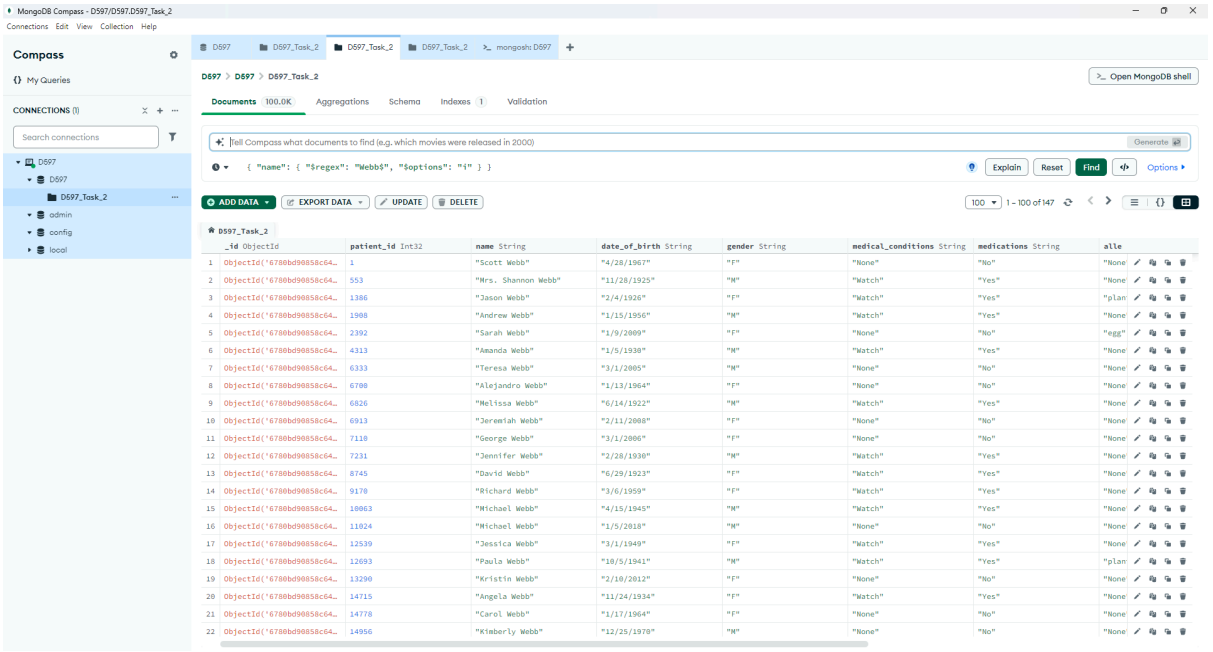
For logging and monitoring, we will have audit logs and intrusion detection systems to record all access and modification attempts and monitor database activity in real-time. *UpGuard*. (n.d.).

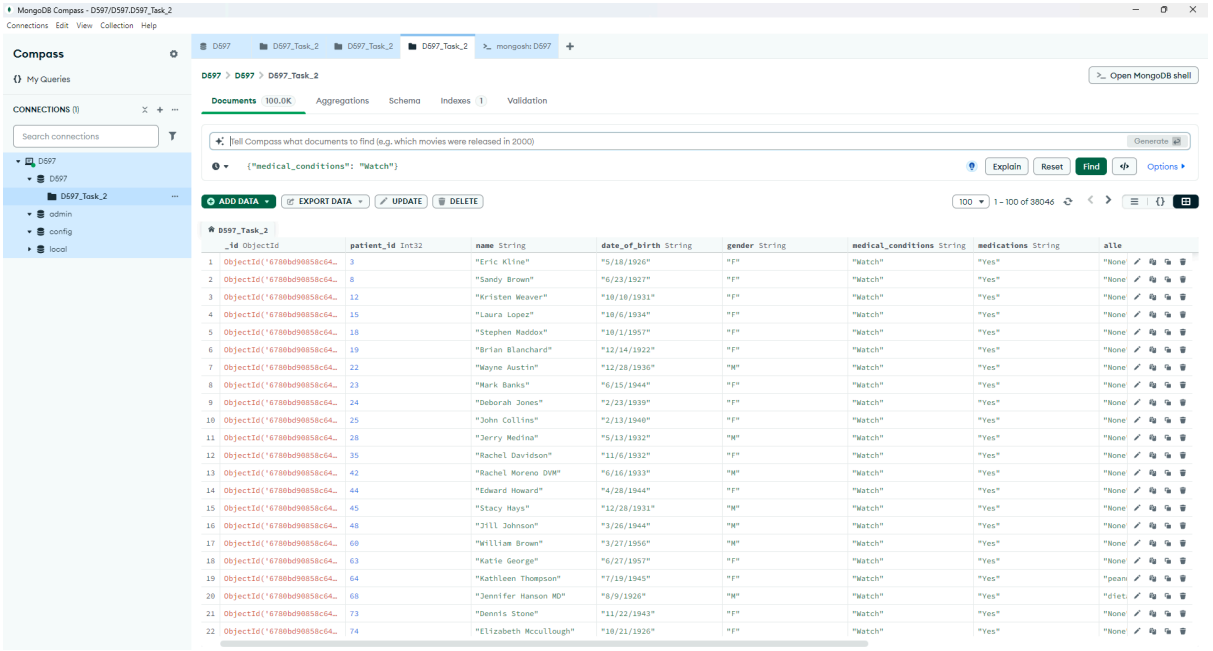
For application integration, I will be implementing API security and input validation. There will also be a need for compliance with regulations; HIPAA compliance and GDPR compliance will be essential for handling personal data, including user consent and the right to access and delete user data if requested. (*Health Insurance Portability and Accountability Act of 1996 (HIPAA)*, 2024)

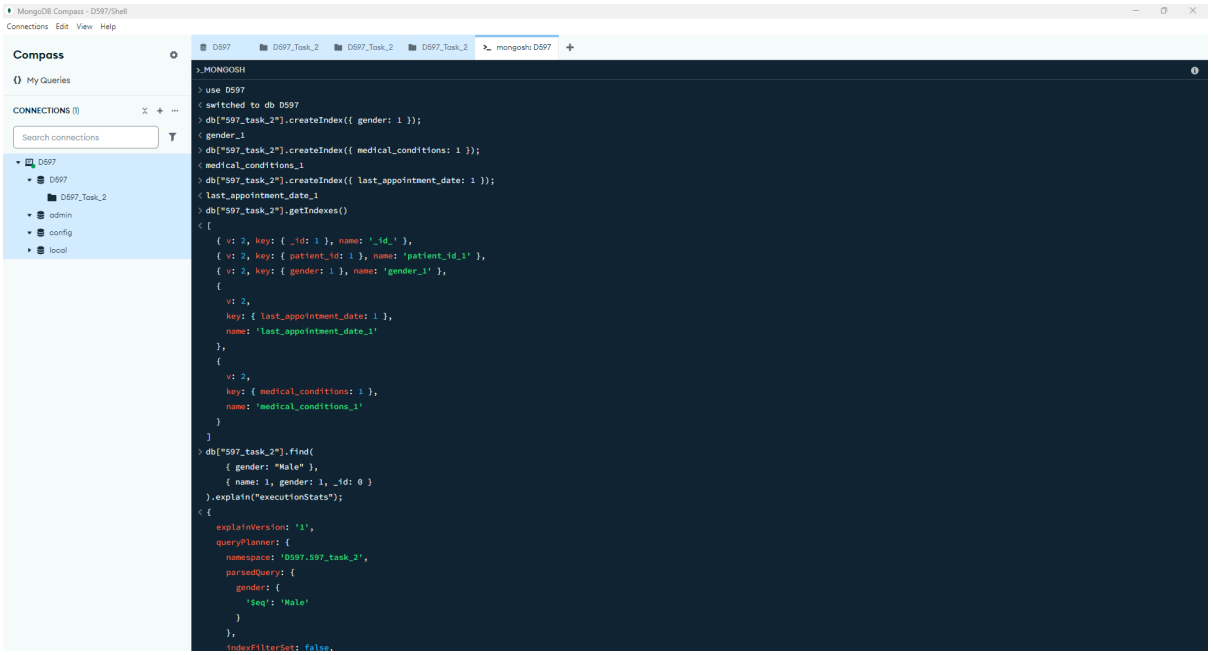
Resilience against attacks will also be achieved through rate limiting, throttling, and DDoS Protection. Data retention policies will be implemented with retention limits and secure deletion. Secure deletion methods will be used for purging old data and determining how long data should be retained. Ahmad, K. (2022, June 20).




MongoDB Compass - D597/D597.D597_Task_2
Connections Edit View Collection Help
Compass
My Queries
CONNECTIONS (1)
Search connections
D597
D597
D597_Task_2
admin
config
local
D597 D597_Task_2 D597_Task_2 D597_Task_2 mongosh: D597
D597 > D597 > D597_Task_2
Open MongoDB shell
Documents 100.0K Aggregations Schema Indexes 1 Validation
Tell Compass what documents to find (e.g. which movies were released in 2000)
Generate
{ "patient_id": { "$gte": 1, "$lte": 100 } }
Explain Reset Find Options
Project { field: 0 }
Sort { field: -1 } or [['field', -1]]
Collation { locale: 'simple' }
Index Hint { field: -1 }
Max Time MS 60000
Skip 0 Limit 0
ADD DATA EXPORT DATA UPDATE DELETE
100 1 - 100 of 100
D597_Task_2
_id ObjectId patient_id Int32 name String date_of_birth String gender String medical_conditions String medications String alle
1 ObjectId('6780bd90858c64... 1 "Scott Webb" "4/28/1967" "F" "None" "No" "None
2 ObjectId('6780bd90858c64... 2 "Rachel Frederick" "4/4/1977" "M" "None" "No" "None
3 ObjectId('6780bd90858c64... 3 "Eric Kline" "5/18/1926" "F" "Watch" "Yes" "None
4 ObjectId('6780bd90858c64... 4 "James Rodriguez" "7/20/1954" "M" "None" "No" "None
5 ObjectId('6780bd90858c64... 5 "David Scott" "12/7/2015" "M" "Mild" "Yes" "None
6 ObjectId('6780bd90858c64... 6 "Dawn Roach" "6/2/1967" "M" "Mild" "Yes" "None
7 ObjectId('6780bd90858c64... 7 "Mary Harris" "8/27/2013" "M" "Mild" "Yes" "None
8 ObjectId('6780bd90858c64... 8 "Sandy Brown" "6/23/1927" "F" "Watch" "Yes" "None
9 ObjectId('6780bd90858c64... 9 "Kenneth Johnson" "4/15/2011" "F" "Mild" "Yes" "None
10 ObjectId('6780bd90858c64... 10 "Nicole Henderson" "10/14/1978" "F" "Mild" "Yes" "None
11 ObjectId('6780bd90858c64... 11 "Brooke Nolan" "5/18/2020" "F" "Mild" "Yes" "None
12 ObjectId('6780bd90858c64... 12 "Kristen Weaver" "10/10/1931" "F" "Watch" "Yes" "None
13 ObjectId('6780bd90858c64... 13 "Brian Brown" "5/15/2019" "F" "Mild" "Yes" "None
14 ObjectId('6780bd90858c64... 14 "Reginald Perez" "5/26/1998" "M" "Mild" "Yes" "None
15 ObjectId('6780bd90858c64... 15 "Laura Lopez" "10/6/1934" "F" "Watch" "Yes" "None
16 ObjectId('6780bd90858c64... 16 "Ashley Rodriguez" "10/14/2016" "F" "Mild" "Yes" "None
17 ObjectId('6780bd90858c64... 17 "Heather French" "8/20/1968" "M" "Mild" "Yes" "None


MongoDB Compass - D597/D597.D597_Task_2
Connections Edit View Collection Help
Compass
My Queries
CONNECTIONS (1)
Search connections
D597
D597
D597_Task_2
admin
config
local
D597 D597_Task_2 D597_Task_2 D597_Task_2 mongosh: D597
D597 > D597 > D597_Task_2
Open MongoDB shell
Documents 100.0K Aggregations Schema Indexes 1 Validation
Tell Compass what documents to find (e.g. which movies were released in 2000)
Generate
{ "name": { "$regex": "Webb$", "$options": "i" } }
Explain Reset Find Options
ADD DATA EXPORT DATA UPDATE DELETE
100 1 - 100 of 147
D597_Task_2
_id ObjectId patient_id Int32 name String date_of_birth String gender String medical_conditions String medications String alle
1 ObjectId('6780bd90858c64... 1 "Scott Webb" "4/28/1967" "F" "None" "No" "None
2 ObjectId('6780bd90858c64... 553 "Mrs. Shannon Webb" "11/28/1925" "M" "Watch" "Yes" "None
3 ObjectId('6780bd90858c64... 1386 "Jason Webb" "2/4/1926" "F" "Watch" "Yes" "plan
4 ObjectId('6780bd90858c64... 1908 "Andrew Webb" "1/15/1956" "M" "Watch" "Yes" "None
5 ObjectId('6780bd90858c64... 2392 "Sarah Webb" "1/9/2009" "F" "None" "No" "egg"
6 ObjectId('6780bd90858c64... 4313 "Amanda Webb" "1/5/1930" "M" "Watch" "Yes" "None
7 ObjectId('6780bd90858c64... 6333 "Teresa Webb" "3/1/2005" "M" "None" "No" "None
8 ObjectId('6780bd90858c64... 6708 "Alejandro Webb" "1/13/1964" "F" "None" "No" "None
9 ObjectId('6780bd90858c64... 6826 "Melissa Webb" "6/14/1922" "M" "Watch" "Yes" "None
10 ObjectId('6780bd90858c64... 6913 "Jeremiah Webb" "2/11/2008" "F" "None" "No" "None
11 ObjectId('6780bd90858c64... 7110 "George Webb" "3/1/2006" "F" "None" "No" "None
12 ObjectId('6780bd90858c64... 7231 "Jennifer Webb" "2/28/1930" "M" "Watch" "Yes" "None
13 ObjectId('6780bd90858c64... 8745 "David Webb" "6/29/1923" "F" "Watch" "Yes" "None
14 ObjectId('6780bd90858c64... 9170 "Richard Webb" "3/6/1959" "F" "Watch" "Yes" "None
15 ObjectId('6780bd90858c64... 10063 "Michael Webb" "4/15/1945" "M" "Watch" "Yes" "None
16 ObjectId('6780bd90858c64... 11024 "Michael Webb" "1/5/2018" "M" "None" "No" "None
17 ObjectId('6780bd90858c64... 12539 "Jessica Webb" "3/1/1949" "F" "Watch" "Yes" "None
18 ObjectId('6780bd90858c64... 12693 "Paula Webb" "10/5/1941" "M" "Watch" "Yes" "plan
19 ObjectId('6780bd90858c64... 13290 "Kristin Webb" "2/10/2012" "F" "None" "No" "None
20 ObjectId('6780bd90858c64... 14715 "Angela Webb" "11/24/1934" "F" "Watch" "Yes" "None
21 ObjectId('6780bd90858c64... 14778 "Carol Webb" "1/17/1964" "F" "None" "No" "None
22 ObjectId('6780bd90858c64... 14956 "Kimberly Webb" "12/25/1970" "M" "None" "No" "None


MongoDB Compass - D597/D597.D597_Task_2
Connections Edit View Collection Help
Compass
My Queries
CONNECTIONS (1)
Search connections
D597
D597
D597_Task_2
admin
config
local
D597 > D597 > D597_Task_2
Open MongoDB shell
Documents 100.0K
Aggregations
Schema
Indexes 1
Validation
Tell Compass what documents to find (e.g. which movies were released in 2000)
Generate
{"medical_conditions": "Watch"}
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
100
1 - 100 of 38046
D597_Task_2
_id ObjectId
patient_id Int32
name String
date_of_birth String
gender String
medical_conditions String
medications String
alle
1 ObjectId('6780bd90858c64... 3 "Eric Kline" "5/18/1926" "F" "Watch" "Yes" "None
2 ObjectId('6780bd90858c64... 8 "Sandy Brown" "6/23/1927" "F" "Watch" "Yes" "None
3 ObjectId('6780bd90858c64... 12 "Kristen Weaver" "10/10/1931" "F" "Watch" "Yes" "None
4 ObjectId('6780bd90858c64... 15 "Laura Lopez" "10/6/1934" "F" "Watch" "Yes" "None
5 ObjectId('6780bd90858c64... 18 "Stephen Maddox" "10/1/1957" "F" "Watch" "Yes" "None
6 ObjectId('6780bd90858c64... 19 "Brian Blanchard" "12/14/1922" "F" "Watch" "Yes" "None
7 ObjectId('6780bd90858c64... 22 "Wayne Austin" "12/28/1936" "M" "Watch" "Yes" "None
8 ObjectId('6780bd90858c64... 23 "Mark Banks" "6/15/1944" "F" "Watch" "Yes" "None
9 ObjectId('6780bd90858c64... 24 "Deborah Jones" "2/23/1939" "F" "Watch" "Yes" "None
10 ObjectId('6780bd90858c64... 25 "John Collins" "2/13/1940" "F" "Watch" "Yes" "None
11 ObjectId('6780bd90858c64... 28 "Jerry Medina" "5/13/1932" "M" "Watch" "Yes" "None
12 ObjectId('6780bd90858c64... 35 "Rachel Davidson" "11/6/1932" "F" "Watch" "Yes" "None
13 ObjectId('6780bd90858c64... 42 "Rachel Moreno DVM" "6/16/1933" "M" "Watch" "Yes" "None
14 ObjectId('6780bd90858c64... 44 "Edward Howard" "4/28/1944" "F" "Watch" "Yes" "None
15 ObjectId('6780bd90858c64... 45 "Stacy Mays" "12/28/1931" "M" "Watch" "Yes" "None
16 ObjectId('6780bd90858c64... 48 "Jill Johnson" "3/26/1944" "M" "Watch" "Yes" "None
17 ObjectId('6780bd90858c64... 60 "William Brown" "3/27/1956" "M" "Watch" "Yes" "None
18 ObjectId('6780bd90858c64... 63 "Katie George" "6/27/1957" "F" "Watch" "Yes" "None
19 ObjectId('6780bd90858c64... 64 "Kathleen Thompson" "7/19/1945" "F" "Watch" "Yes" "pean
20 ObjectId('6780bd90858c64... 68 "Jennifer Hanson MD" "8/9/1926" "M" "Watch" "Yes" "diet
21 ObjectId('6780bd90858c64... 73 "Dennis Stone" "11/22/1943" "F" "Watch" "Yes" "None
22 ObjectId('6780bd90858c64... 74 "Elizabeth Mccullough" "10/21/1926" "F" "Watch" "Yes" "None


MongoDB Compass - D597/Shell
Connections Edit View Help
Compass
My Queries
CONNECTIONS (1)
Search connections
D597
D597
D597_Task_2
admin
config
local
mongosh: D597
>_MONGOSH
> use D597
< switched to db D597
> db["597_task_2"].createIndex({ gender: 1 });
< gender_1
> db["597_task_2"].createIndex({ medical_conditions: 1 });
< medical_conditions_1
> db["597_task_2"].createIndex({ last_appointment_date: 1 });
< last_appointment_date_1
> db["597_task_2"].getIndexes()
< [
{ v: 2, key: { _id: 1 }, name: '_id_' },
{ v: 2, key: { patient_id: 1 }, name: 'patient_id_1' },
{ v: 2, key: { gender: 1 }, name: 'gender_1' },
{
v: 2,
key: { last_appointment_date: 1 },
name: 'last_appointment_date_1'
},
{
v: 2,
key: { medical_conditions: 1 },
name: 'medical_conditions_1'
}
]
> db["597_task_2"].find(
{ gender: "Male" },
{ name: 1, gender: 1, _id: 0 }
).explain("executionStats");
< {
explainVersion: '1',
queryPlanner: {
namespace: 'D597.597_task_2',
parsedQuery: {
gender: {
'$eq': 'Male'
}
},
indexFilterSet: false,

Database for health records, medical data, and patient care. (n.d.). MongoDB. https://www.mongodb.com/solutions/industries/healthcare

Course |. (n.d.). https://apps.cgp-oex.wgu.edu/wgulearning/course/course-v1:WGUx+OEX0343+v01/block-v1:WGUx+OEX0343+v01+type@sequential+block@543516264ee84ad4b89a1449a5d2db90/block-v1:WGUx+OEX0343+v01+type@vertical+block@63e2765421dc476a8611a55973cafd50

MongoDB. (n.d.). *What is NoSQL? NoSQL databases explained*. https://www.mongodb.com/resources/basics/databases/nosql-explained

Ahmad, K. (2022, June 20). *What is AES-256 encryption? How does it work?* MUO. https://www.makeuseof.com/what-is-aes-256-encryption-how-does-it-work/

GeeksforGeeks. (2024, May 23). *Transport Layer Security (TLS)*. GeeksforGeeks. https://www.geeksforgeeks.org/transport-layer-security-tls/

What is Role-Based Access Control (RBAC)? Examples, Benefits, and More | UpGuard. (n.d.). https://www.upguard.com/blog/rbac

What is Role-Based Access Control (RBAC)? | Okta. (n.d.). Okta, Inc. https://www.okta.com/identity-101/what-is-role-based-access-control-rbac/

Health Insurance Portability and Accountability Act of 1996 (HIPAA). (2024, September 10). Public Health Law. https://www.cdc.gov/phlp/php/resources/health-insurance-portability-and-accountability-act-of-1996-hipaa.html