# Food Recipe Management System

WITH: MONGO DB

# GROUP MEMBERS


Athary Sultan Alsowat 442000884
(s442000884@st.uqu.edu.sa)

Raghad Saeed Al-Zahrani 442017353
(s442017353@st.uqu.edu.sa)

Abrar Tarik Saigh 442013240
(s442013240@st.uqu.edu.sa)

Areej Saud Alqurashl 442001962
(s442001962@st.uqu.edu.sa)

Aseel Hussien Qedear 442006014
(s442006014@st.uqu.edu.sa)

Noha Imam 441017954
(s441017954@st.uqu.edu.sa)

Sumayyah Abdoulmoen Albarakati 442012157
(s442012157@st.uqu.edu.sa)


Section: 2

Qroup: 2

Supervisor: Dr. Hanan Alhazmi

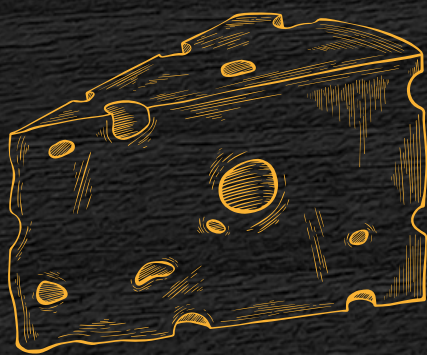
# TABLE OF CONTENTS

# Phase (1)

# Problem Description

Meals hold a significant place in our daily routines, and our collection of recipes often becomes a go-to resource. However, the abundance of culinary information from various sources, like the Internet and books, has made organizing and storing recipes a daunting task. Where recipes end up scattered across different mediums, locating a specific recipe becomes a time-consuming task involving multiple sources.

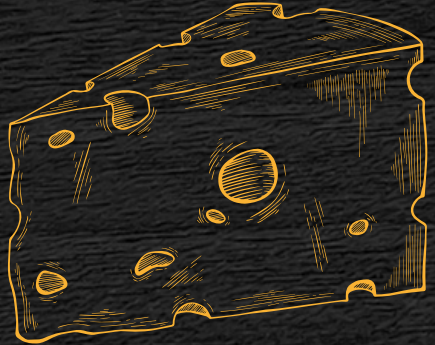
Additionally, modifying or updating recipes recorded in traditional media presents challenges, Simple adjustments to ingredients or cooking methods become impractical, and capturing personal feedback becomes cumbersome. Furthermore, keeping track of ingredient quantities poses difficulties, potentially impacting the quality of the final meal.



# Our Goal & Motivation

The overall goal of our project is to utilize MongoDB to create a database that can facilitate users in adding, modifying, and deleting recipes, as well as categorizing and organizing them based on various criteria such as type, ingredients, and more. The project also allows for the documentation of personal notes and user comments on recipes, contributing to their improvement and customization.

The database also contributes to facilitating access to information related to recipes and providing advanced search functions to facilitate browsing and categorizing recipes based on multiple criteria. The motivation behind choosing this topic is based on the increasing need to organize and manage individuals' favorite food recipes in a flexible and organized manner. Our project addresses this need by using the MongoDB database, which provides fast and efficient data retrieval and allows for flexible data storage. Additionally, MongoDB enables handling large datasets and achieving excellent performance, making it an ideal choice for storing and retrieving food recipes.

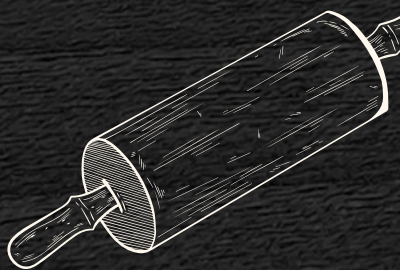


# Abstract

The Recipe Management System is a database-based solution aimed at organizing and managing recipes effectively. With the use of MongoDB, users can add, modify, and delete recipes while categorizing and organizing them based on various criteria. The system allows for the documentation of personal notes and user comments, facilitating recipe improvement and customization. It also has advanced search functions that enable easy browsing and categorization of recipes. MongoDB's performance, scalability, and data retrieval capabilities make it an ideal choice for storing and retrieving food recipes.

# Introduction

MongoDB is an open-source NoSQL database known for its performance, scalability, and flexibility. It stores data in JSON-like documents, allowing for easy storage and manipulation. MongoDB supports multiple programming languages and offers features like MongoDB Atlas that make it easy to set up and use. It's good at handling large amounts of data and can handle many tasks at once. MongoDB has been around for a while and has a lot of people who use it and help each other out. Lastly, MongoDB's advantages make it an ideal choice for our recipe management system, enabling efficient data storage, retrieval, manipulation, and supporting future growth.[1]



# HISTORICAL OVERVIEW

In 2007, Dwight Merriman and Eliot Horowitz founded MongoDB. Prior to starting MongoDB, the team had worked together at DoubleClick, an Internet advertising company. While at DoubleClick, they faced significant challenges related to scalability and agility, primarily due to the limitations of existing databases. The shortcomings of traditional databases posed a major hindrance to achieving their business goals.

Motivated by their frustrations and driven to find a better solution, the team embarked on a mission to create a database that could effectively address the challenges they had encountered at DoubleClick. This marked the birth of MongoDB, a database designed to offer improved scalability and agility.

In 2009, MongoDB was introduced to the market as an open-source database server. Its innovative approach to database management quickly gained attention and popularity among developers and businesses seeking more flexible and scalable data solutions. Recognizing the growing demand, MongoDB Inc. was established in 2013 to provide commercial support and services for MongoDB.

Since its inception, MongoDB remains one of the prominent NoSQL databases in the market, addressing the needs of modern applications that require scalability, high performance, and handling of unstructured data. [2][3]

# Usages

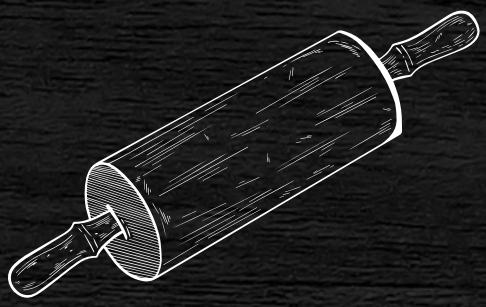
MongoDB is a powerful database that can be used in a lot of applications due to the features it has. Flexibility, scalability, document-based model, and rich feature set make it a popular choice for a wide range of applications and industries [4],[5],[6],[7]. Here are some of the commonly used applications of it:

- **Mobile Apps**: Due to MongoDB's synchronization capabilities, it enable mobile apps to work offline and sync data when a network connection is available. This is useful for apps that need to store and update data on the device even without an internet connection. Messaging apps like WhatsApp or Slack, where offline messaging and data synchronization are crucial.

- **Web Applications**: MongoDB is also commonly used in web applications because it allows developers to easily adapt the data structures as their application evolves. It can handle different types of content, such as articles, images, and user-generated content. You can add new fields or modify existing ones without altering the entire database schema. For example, E-commerce websites that handle product catalogs and customer data, allowing for easy adaptation as the business evolves.

- **Internet of Things (IoT)**: refers to physical devices, vehicles, appliances, and other objects embedded with sensors, software, and connectivity capabilities. MongoDB is a good choice for IoT applications due to its ability to handle large volumes of data. It can store data from sensors, devices, and logs efficiently, allowing real-time analysis and querying of IoT data, thus maximizing the full protentional of them. Like Smart home systems that collect sensor data from various devices and allow for real-time analysis and control.

# Usages



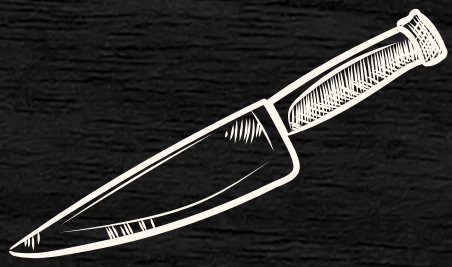
- **Content Management System**: MongoDB can store different types of content, such as text posts, images, and videos, which can benefit content management systems like news portals or blogs, where you can efficiently search and retrieve this content for displaying personalized feeds to users.

- **Real-Time Analytics**: MongoDB is well-suited for applications that require real-time analytics, like monitoring systems, gaming systems or recommendation engines that process user behavior data in real-time to provide personalized recommendations. It can process and analyze data quickly as it is generated, providing instant insights, or triggering actions based on incoming data. Its fast querying and real-time updates enable timely decision-making and personalized experiences by analyzing data in real time.

- **Social Networking**: The scalability and real-time capabilities of MongoDB make it ideal for social networking platforms. It can manage user profiles, social connections, and real-time updates, allowing users to have seamless interactions and personalized experiences. Social media platforms like Facebook or Instagram, where user profiles, connections, and real-time updates are managed to enable seamless interactions.

In conclusion, MongoDB is a powerful and versatile database. Its flexibility allows for easy adaptation of data structures, while its scalability ensures efficient handling of large volumes of data. Its document-based model simplifies data management, enabling seamless storage and retrieval of diverse content types. MongoDB provides the foundation for building robust and scalable solutions.

# Advantages



- **Flexible and Scalable:** It allows you to store and manage unstructured and semi-structured data easily, making it suitable for handling a wide variety of data types
- **High Performance:** MongoDB offers high-performance capabilities, particularly for read-intensive workloads. It supports indexing and caching mechanisms that optimize query execution and retrieval of data.
- **Automatic Sharding and Replication:** MongoDB provides built-in support for automatic sharding and replication. Sharding allows you to distribute data across multiple machines, enabling horizontal scaling and improved performance. Replication provides fault tolerance and high availability by maintaining multiple copies of data across different servers.
- **Schema-less Design:** MongoDB's schema-less design allows for flexible and dynamic data modeling. It does not enforce a rigid schema structure, which means you can easily modify the data schema
- **Rich Query Language:** MongoDB offers a powerful and expressive query language that supports a wide range of operations, including filtering, sorting, aggregations, and geospatial queries.
- **Horizontal Scalability:** MongoDB's architecture supports horizontal scalability, allowing you to scale your database by adding more servers or nodes to handle increased data storage and traffic.
- **Ease of Development:** MongoDB's flexible data model, JSON-like documents, and extensive query capabilities make it easy for developers to work with. It has a rich set of APIs and drivers available for various programming languages, simplifying the development process and reducing the learning curve.[14]

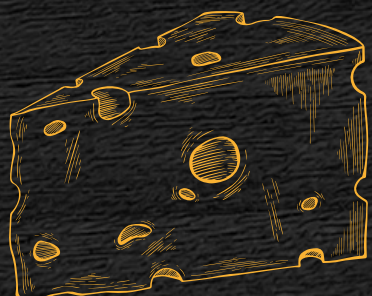
# Disadvantages

- **Memory Consumption:** MongoDB's memory usage can be higher compared to traditional relational databases, especially when dealing with large datasets.
- **Data Fragmentation:** MongoDB's dynamic schema and flexible document model can lead to data fragmentation over time. As documents are updated and modified, they can grow in size and create gaps in storage.
- **Limited Join Operations:** MongoDB's denormalized data model does not support traditional SQL-like join operations that are common in relational databases.
- **Concurrency and Locking:** MongoDB uses a document-level locking mechanism for write operations. This means that write operations on the same document can be serialized, potentially impacting concurrency.
- **Indexing:** If indexes are implemented incorrectly or contain discrepancies, MongoDB's performance can be severely impacted, resulting in slow query execution and addressing index errors can be a time-consuming process, presenting a significant limitation of MongoDB.
- **Lack of ACID Transactions:** MongoDB, by default, does not support ACID (Atomicity, Consistency, Isolation, Durability) transactions that guarantee data consistency in complex operations involving multiple documents or collections. While MongoDB does provide support for atomic operations at the document level, it may not be suitable for applications that require strict transactional integrity.[14]

# Alternatives

- **Apache Cassandra:** is a popular open-source NoSQL database known for its powerful scalability and distributed architecture. In addition to that, it's like a superhero for managing enormous amounts of data, effortlessly spreading it across multiple servers to ensure high availability and resiliency. Moreover, with automatic replication and support for multi-data center setups, it's a performance powerhouse, smoothly handling heavy workloads across multiple servers[4].

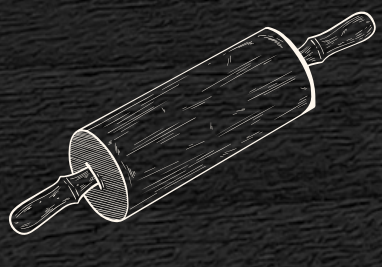

- **PostgreSQL, or Postgres:** for short, is a widely respected open-source database. In addition to that, it is favored by many corporations and developers for its outstanding performance and efficiency. Moreover, what makes PostgreSQL stand out is its extensive range of features, including support for important elements like indexing, complex queries, data consistency, transactions, and data integrity[5].

# Alternatives

- **Couchbase** is a powerful NoSQL database that excels in storing and retrieving large data sets. In addition to that, it organizes data in a flexible way using a JSON document store, allowing for easy schema design. Moreover, Couchbase is known for its impressive performance, scalability, and ability to handle low-latency applications. Also, it uses a schema-less data model, meaning you can store documents as simple key-value pairs, making development a breeze[6]

- **MariaDB**, an open-source relational database management system (RDBMS) is a robust and scalable solution for managing food recipes. It offers high performance and supports traditional relational database features, allowing you to organize recipe data efficiently. With the ability to define tables, establish relationships, and execute SQL queries, MariaDB enables easy structuring and retrieval of recipe information. Furthermore, its cost-effectiveness and strong community support make it an excellent alternative [7]



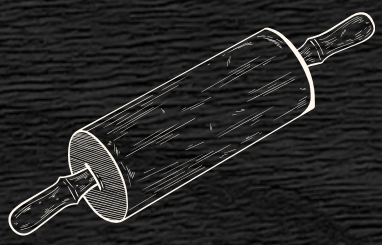
# Alternatives

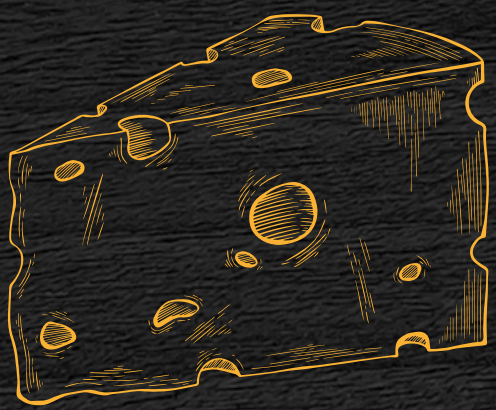
- **Firebase** is a cloud-based platform by Google that offers a range of services for web and mobile app development. In addition to that, it includes Cloud Firestore, a document-oriented NoSQL database suitable for storing food recipes. Furthermore, with Firestore, we can organize recipe data flexibly and query it easily. Moreover, it supports real-time synchronization for collaboration and offline access, ensuring uninterrupted recipe browsing[8].

**Some other efficient alternatives:**

- Redis
- SQLite
- Amazon DynamoDB

etc...

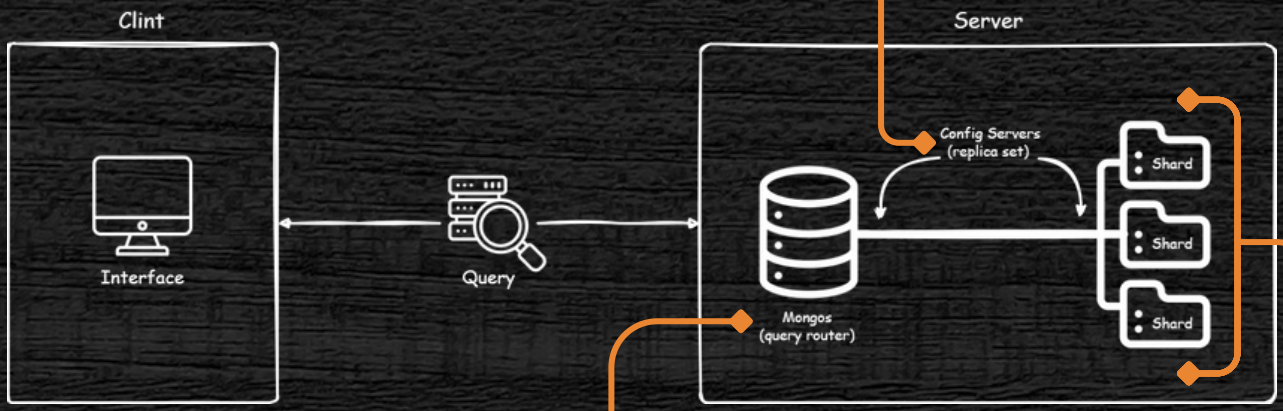




# Architecture

Sharding is a method for distributing data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

**Config Servers**: store metadata and configuration settings for the cluster.




**Mongos**: acts as a query router, providing an interface between client applications and the sharded cluster.

**Shard:** contains a subset of the collection data. Each shard can be deployed as a replica set. replica set is a feature allowing multiple database servers to share the same data, thereby ensuring redundancy and facilitating load balancing.

# Components

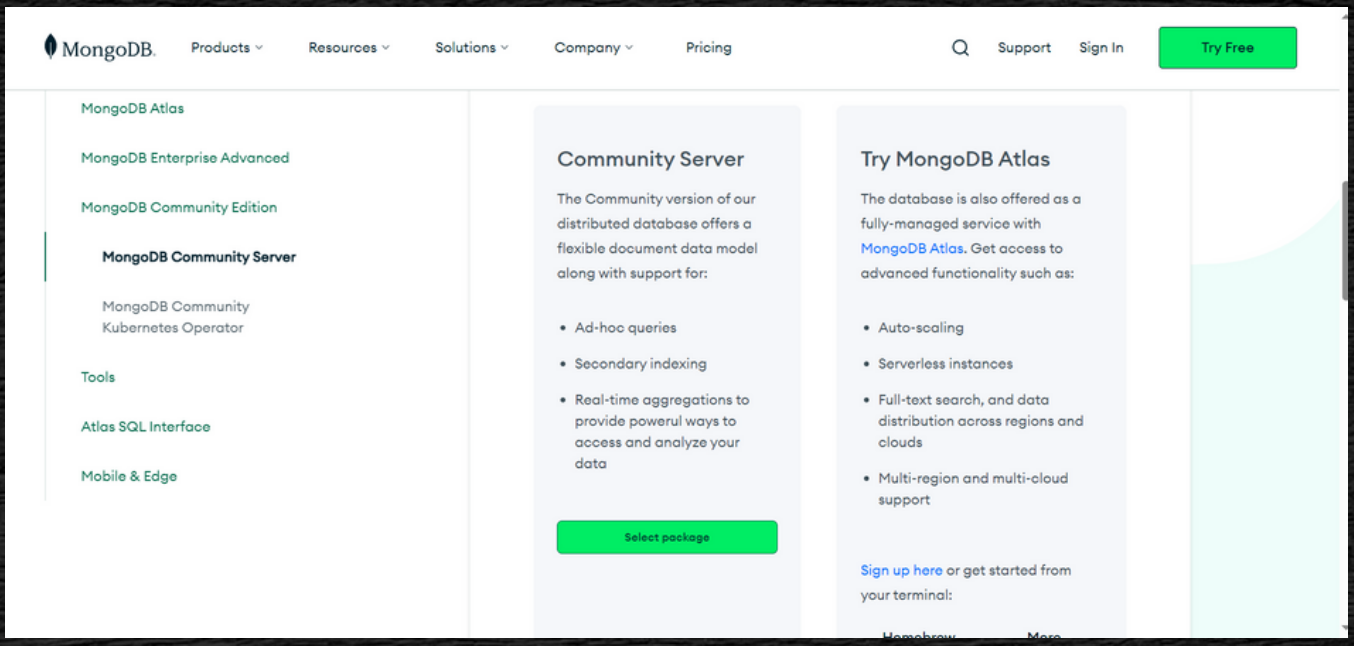
According to Banker's *et al.* "MongoDB in Action" book [13] MongoDB consists of:

- **Documents**: Fundamental unit of MongoDB, and it's in a JSON-like format that contain a set of attributes and their corresponding values each with a unique identifier, representing records of a table in relational database.
- **Collections**: group of related documents, and collections of documents are stored together in a disk, and most queries require to specify which collection to retrieve from. representing tables in relational database.
- **Databases:** group of collections stored on the same MongoDB server.
- **Server:** database software that capable of processing queries, managing data storage setting and access restrictions.
- **Clint:** application that interact with the database to get or manipulate the data, or to perform task on the dataset.

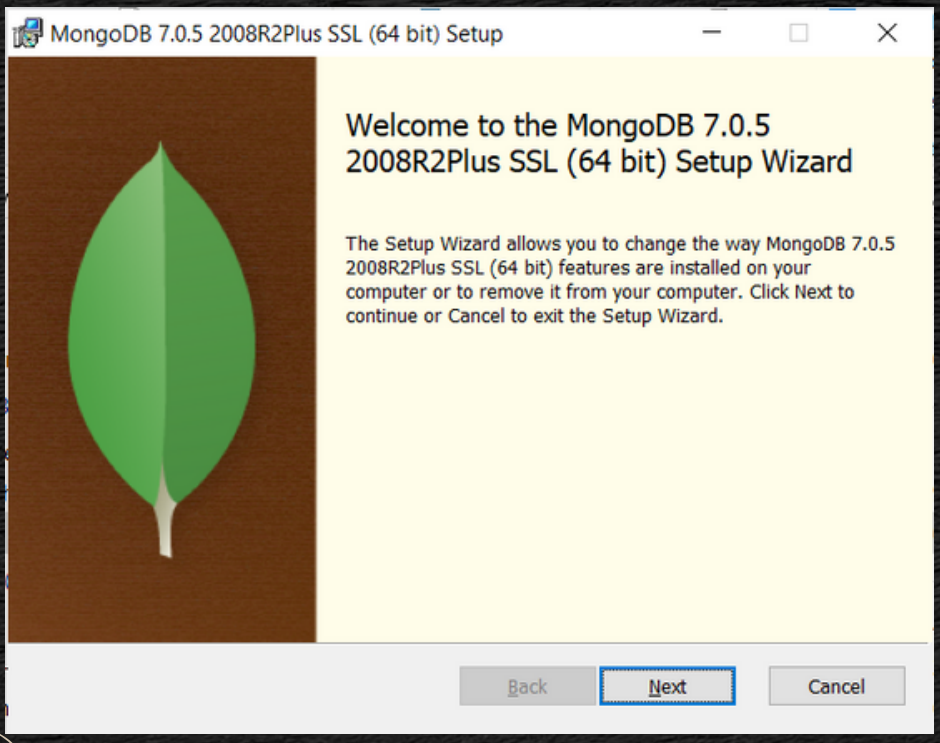
# Phase (2)

# Set up screenshots

Step 1: Downloading the MongoDB Community Server from MongoDB official website.

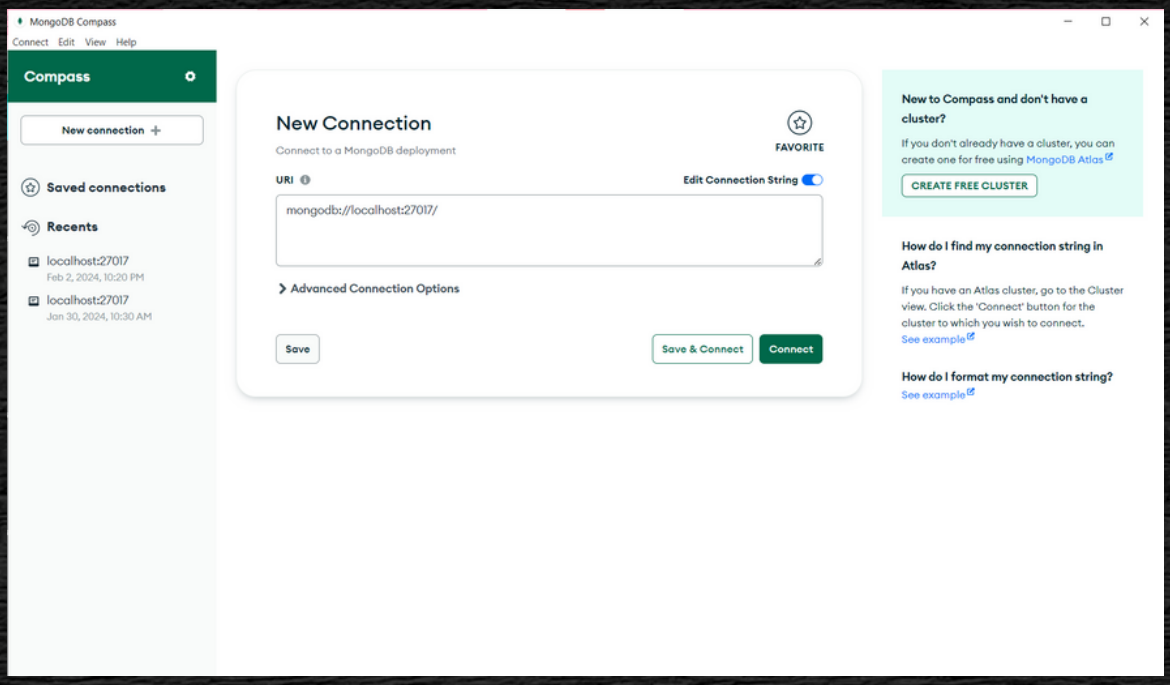



Step 2: Following the installation instructions of Mongod

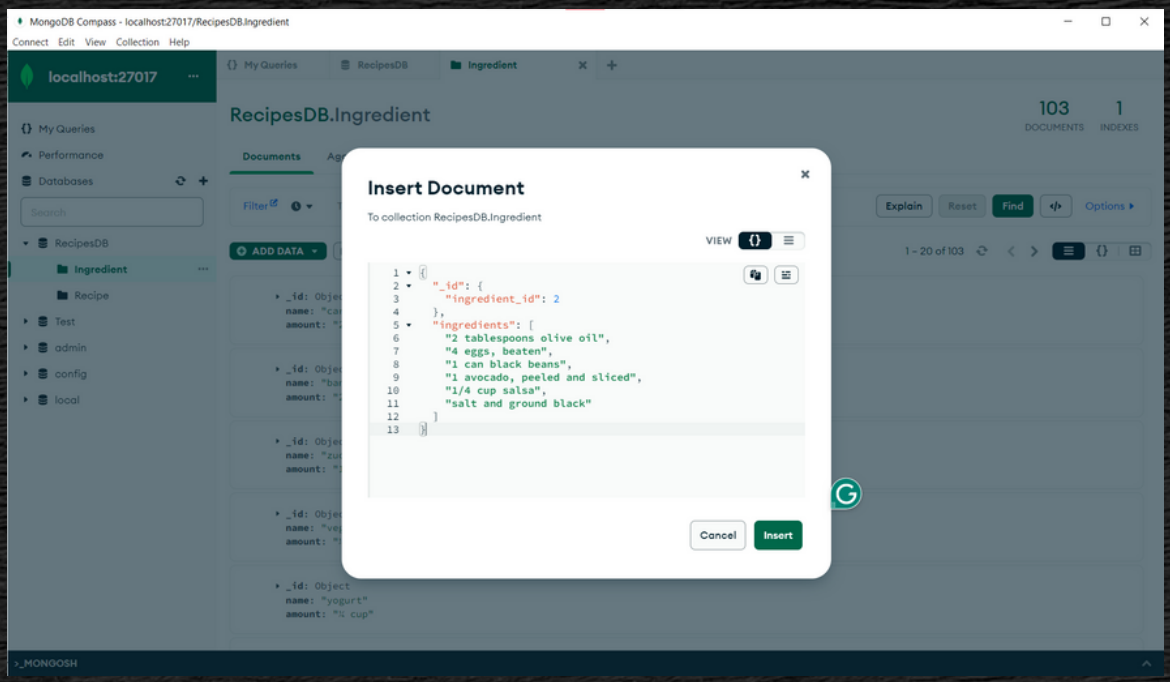


# Set up screenshots

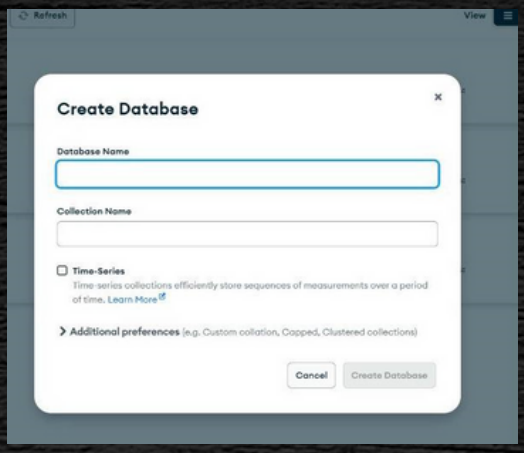
Step 3: Running MongoDB Compass and Connect to the server using a by pressing the Connect button.



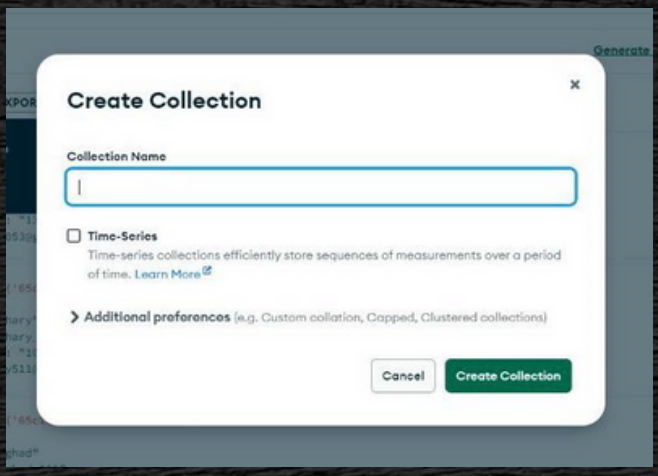

Step 4: Start coding




# Tables creation stepes




First step: create a database
called Food Recipe Management System




The second step: create a collection


The third step: create the
documents inside the collection

These are the three steps followed in
order to create all our project tables

# Recipes Documents


```
_id: ObjectId('65bcecedc9a186b789f30438')
name : "Muffins"
description : "Healthy, hearty breakfast muffins. These have a lightly sweet flavor a…"
prep_time : "20 minutes"
cook_time : "20 minutes"
servings : 12
nutrition_facts_per_serving : "227 calories, 10g fat, 32g carbs, 5g protein"
category_breakfast : ObjectId('65c252d0e49cda9d509351aa')
user_Athary : ObjectId('65c1ef7a2dcdd1e909534c3e')
user_Raghad : ObjectId('65c1efa82dcdd1e909534c40')
```

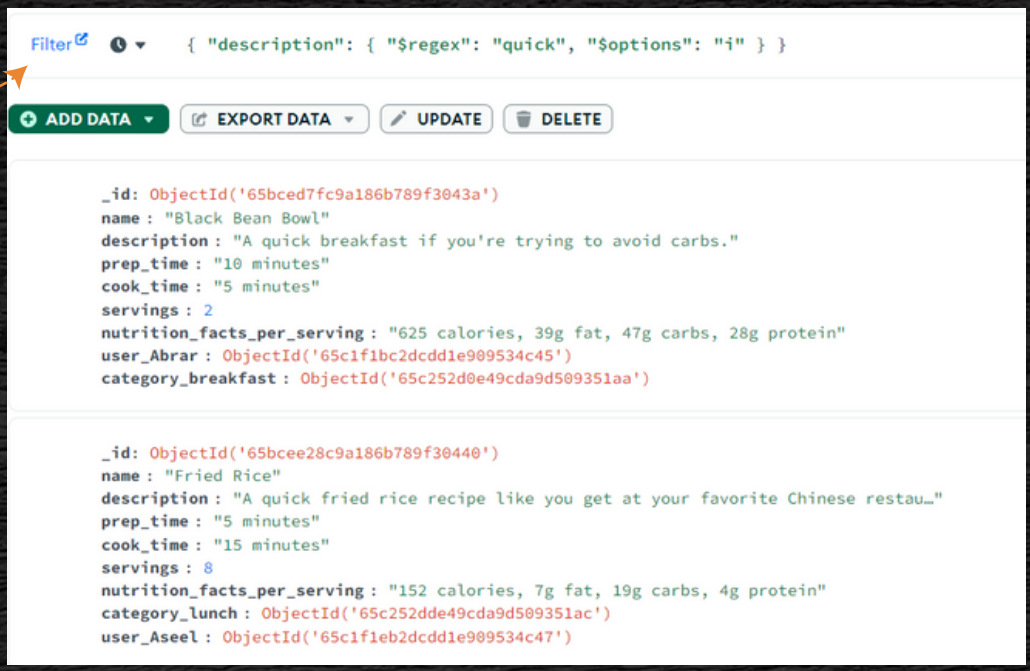
```
_id: ObjectId('65bced7fc9a186b789f3043a')
name : "Black Bean Bowl"
description : "A quick breakfast if you're trying to avoid carbs."
prep_time : "10 minutes"
cook_time : "5 minutes"
servings : 2
nutrition_facts_per_serving : "625 calories, 39g fat, 47g carbs, 28g protein"
user_Abrar : ObjectId('65c1f1bc2dcdd1e909534c45')
category_breakfast : ObjectId('65c252d0e49cda9d509351aa')
```

```
_id: ObjectId('65bcedd6c9a186b789f3043d')
name : "Pancakes"
description : "I found this pancake recipe in my Grandma's recipe book. Judging from …"
prep_time : "5 minutes"
cook_time : "15 minutes"
servings : 8
nutrition_facts_per_serving : "158 calories, 6g fat, 22g carbs, 5g protein"
user_Noha : ObjectId('65c1f24d2dcdd1e909534c4b')
category_breakfast : ObjectId('65c252d0e49cda9d509351aa')
```

```
_id: ObjectId('65bcee28c9a186b789f30440')
name : "Fried Rice"
description : "A quick fried rice recipe like you get at your favorite Chinese restau…"
prep_time : "5 minutes"
cook_time : "15 minutes"
servings : 8
nutrition_facts_per_serving : "152 calories, 7g fat, 19g carbs, 4g protein"
category_lunch : ObjectId('65c252dde49cda9d509351ac')
user_Aseel : ObjectId('65c1f1eb2dcdd1e909534c47')
```

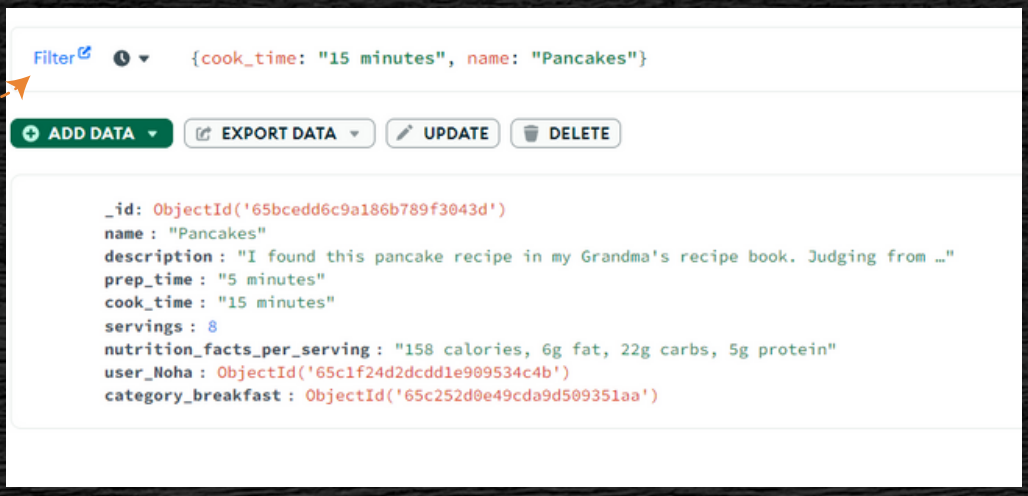

# Recipes Queries

SELECT * FROM recipes WHERE description LIKE '%quick%'



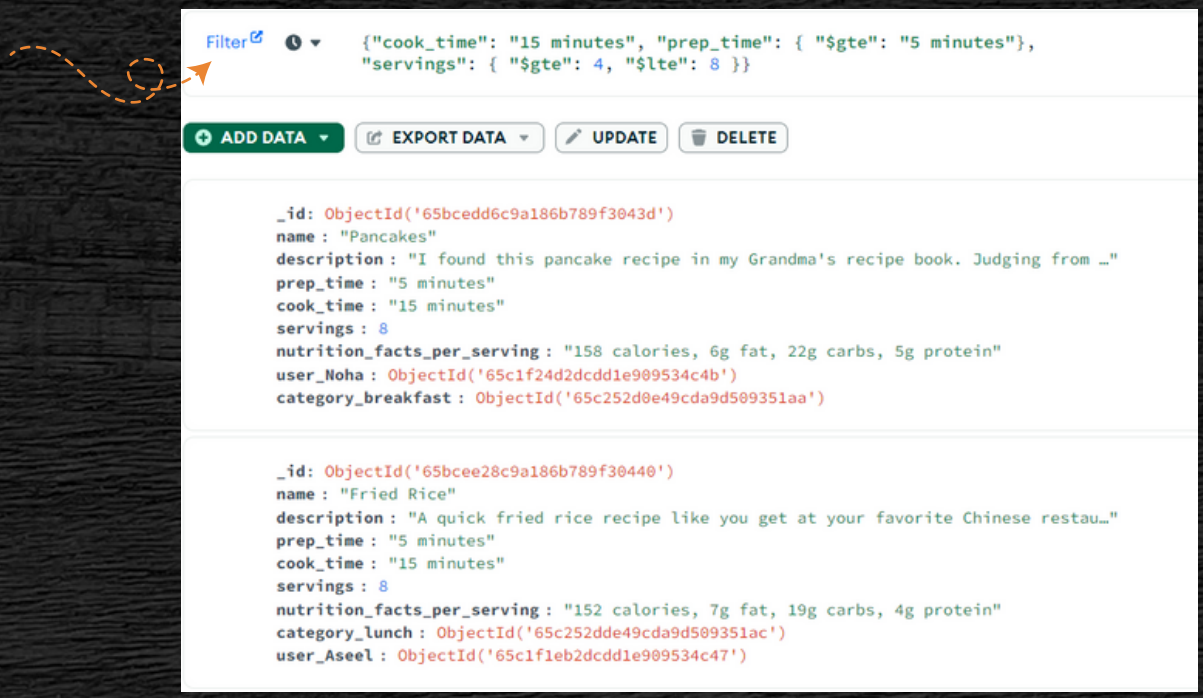

# Recipes Queries

SELECT * FROM recipes WHERE cook_time = '15 minutes' AND name = 'Pancakes'



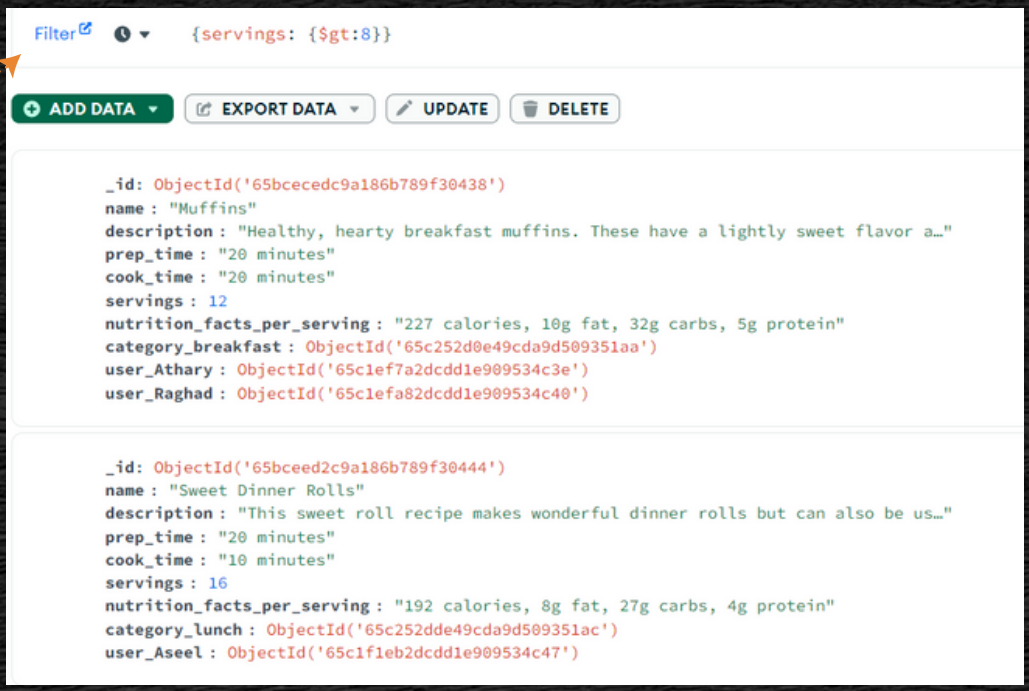

# Recipes Queries

SELECT * FROM recipes WHERE cook_time = '15 minutes'
AND prep_time >= '5 minutes'
AND servings BETWEEN 4 AND 8



# Recipes Queries

SELECT * FROM recipes WHERE servings > 8




# Review Documents


```
_id: ObjectId('65c1efe42dcdd1e909534c41')
review_id : 1
recipe_id : 1
user_id : ObjectId('65c1ef552dcdd1e909534c3c')
comment : "The combination of flavors in this dish is exquisite"
rating : 4.5
date : "2024-01-29"
```

```
_id: ObjectId('65c1f0322dcdd1e909534c42')
review_id : 2
recipe_id : 2
user_id : ObjectId('65c1ef7a2dcdd1e909534c3e')
comment : ""This meal is incredibly tasty"
rating : 4
date : "2023-05-20"
```

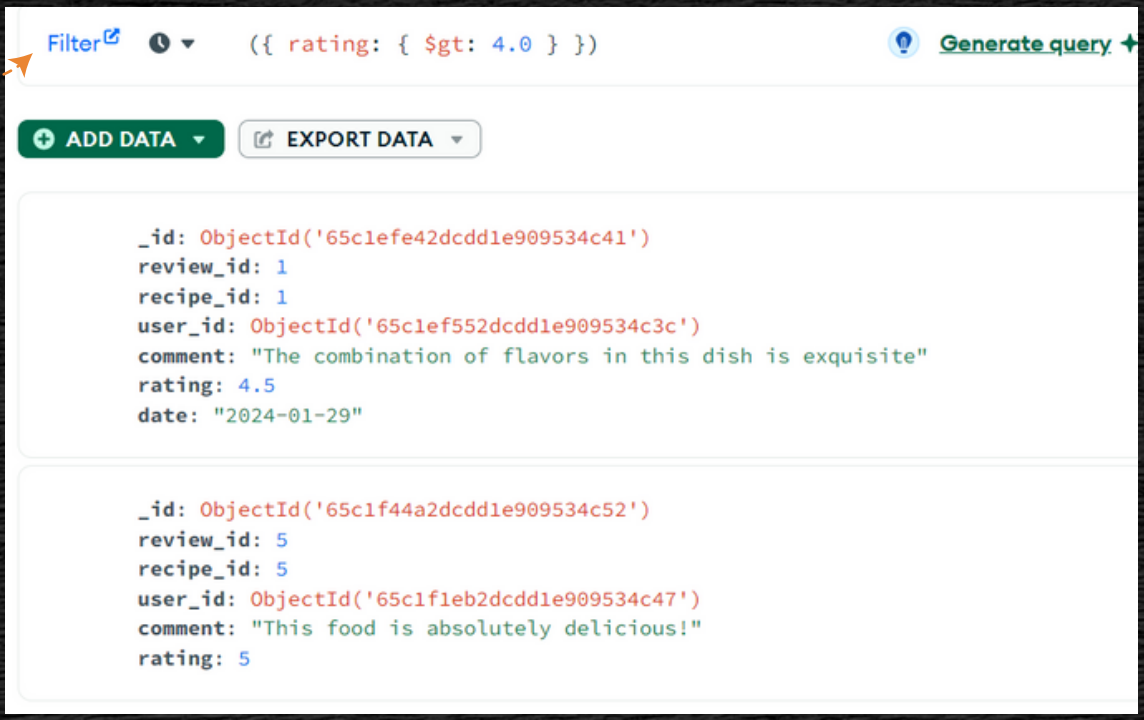
```
_id: ObjectId('65c1f04b2dcdd1e909534c43')
review_id : 3
recipe_id : 3
user_id : ObjectId('65c1efa82dcdd1e909534c40')
comment : "Unfortunately, the dish turned out poorly and wasn't enjoyable"
rating : 2
date : "2023-10-07"
```

```
_id: ObjectId('65c1f3642dcdd1e909534c4e')
review_id : 4
recipe_id : 4
user_id : ObjectId('65c1f1bc2dcdd1e909534c45')
comment : "The recipe was decent, but it didn't blow me away"
rating : 3
date : "2023-011-29"
```

```
_id: ObjectId('65c1f44a2dcdd1e909534c52')
review_id : 5
recipe_id : 5
user_id : ObjectId('65c1f1eb2dcdd1e909534c47')
comment : "This food is absolutely delicious!"
rating : 5
```


# Review Queries

SELECT * FROM review WHERE rating > 4.0




# Review, User Queries

use aggregation operations between  review and user documents and return computed results

```
_id: ObjectId('65c1efa82dcdd1e909534c40')
user_id: 3
username: "Raghad"
password: "Raghad_111"
date_of_birth: "29/4/2002"
email: "Raghad111@gmail.com"

_id: ObjectId('65c1f1bc2dcdd1e909534c45')
user_id: 4
username: "Abrar"
password: "Abrar_910"
date_of_birth: "5/5/2002"
email: "Abrar910@gmail.com"
```

example of some documents from user collection

```
_id: ObjectId('65c1f04b2dcdd1e909534c43')
review_id: 3
recipe_id: 3
user_id: ObjectId('65c1efa82dcdd1e909534c40')
comment: "Unfortunately, the dish turned out poorly and wasn't enjoyable"
rating: 2
date: "2023-10-07"

_id: ObjectId('65c1f3642dcdd1e909534c4e')
review_id: 4
recipe_id: 4
user_id: ObjectId('65c1f1bc2dcdd1e909534c45')
comment: "The recipe was decent, but it didn't blow me away"
rating: 3
date: "2023-011-29"
```

example of some documents from review collection

```
  {
    $lookup: {
      from: "user",
      localField: "user_id",
      foreignField: "_id",
      as: "userInfo",},},
  {
    $match: {
      "userInfo.user_id":1 ,},},
  {
    $project: {
      _id: 0,
      comment: 1,
      rating: 1,
      date: 1,
```

use $lookup, $match and $project operations

# Review, User Queries

```
[
  {
    $lookup: {
      from: "user",
      localField: "user_id",
      foreignField: "_id",
      as: "userInfo",},},
  {
    $match: {
      "userInfo.user_id":1 ,},},
  {
    $project: {
      _id: 0,
      comment: 1,
      rating: 1,
      date: 1,
      "userInfo.name": 1,
```

PIPELINE OUTPUT — OUTPUT OPTIONS

Sample of 1 document

```
comment: "The combination of flavors in this dish is
         exquisite"
rating: 4.5
date: "2024-01-29"
▸ userInfo: Array (1)
```

retrieves documents from the "user" collection, performs a join with the same collection when user_id =1

```
[
  {
    $lookup: {
      from: "user",
      localField: "user_id",
      foreignField: "_id",
      as: "userInfo",},},
  {
    $match: {
      "userInfo.user_id":2 ,},},
  {
    $project: {
      _id: 0,
      comment: 1,
      rating: 1,
      date: 1,
      "userInfo.name": 1,
```

PIPELINE OUTPUT — OUTPUT OPTIONS

Sample of 1 document

```
comment: ""This meal is incredibly tasty"
rating: 4
date: "2023-05-20"
▸ userInfo: Array (1)
```

retrieves documents from the "user" collection, performs a join with the same collection when user_id =2

```
[
  {
    $lookup: {
      from: "user",
      localField: "user_id",
      foreignField: "_id",
      as: "userInfo",},},
  {
    $match: {
      "userInfo.user_id":3 ,},},
  {
    $project: {
      _id: 0,
      comment: 1,
      rating: 1,
      date: 1,
      "userInfo.name": 1,
```

PIPELINE OUTPUT — OUTPUT OPTIONS

Sample of 1 document

```
comment: "Unfortunately, the dish turned out poorly and
         wasn't enjoyable"
rating: 2
date: "2023-10-07"
▸ userInfo: Array (1)
```

retrieves documents from the "user" collection, performs a join with the same collection when user_id = 3

```
[
  {
    $lookup: {
      from: "user",
      localField: "user_id",
      foreignField: "_id",
      as: "userInfo",},},
  {
    $match: {
      "userInfo.user_id":4 ,},},
  {
    $project: {
      _id: 0,
      comment: 1,
      rating: 1,
      date: 1,
      "userInfo.name": 1,
```

PIPELINE OUTPUT — OUTPUT OPTIONS

Sample of 1 document

```
comment: "The recipe was decent, but it didn't blow me
         away"
rating: 3
date: "2023-011-29"
▸ userInfo: Array (1)
```

retrieves documents from the "user" collection, performs a join with the same collection when user_id = 4

```
[
  {
    $lookup: {
      from: "user",
      localField: "user_id",
      foreignField: "_id",
      as: "userInfo",},},
  {
    $match: {
      "userInfo.user_id":5 ,},},
  {
    $project: {
      _id: 0,
      comment: 1,
      rating: 1,
      date: 1,
      "userInfo.name": 1,
```

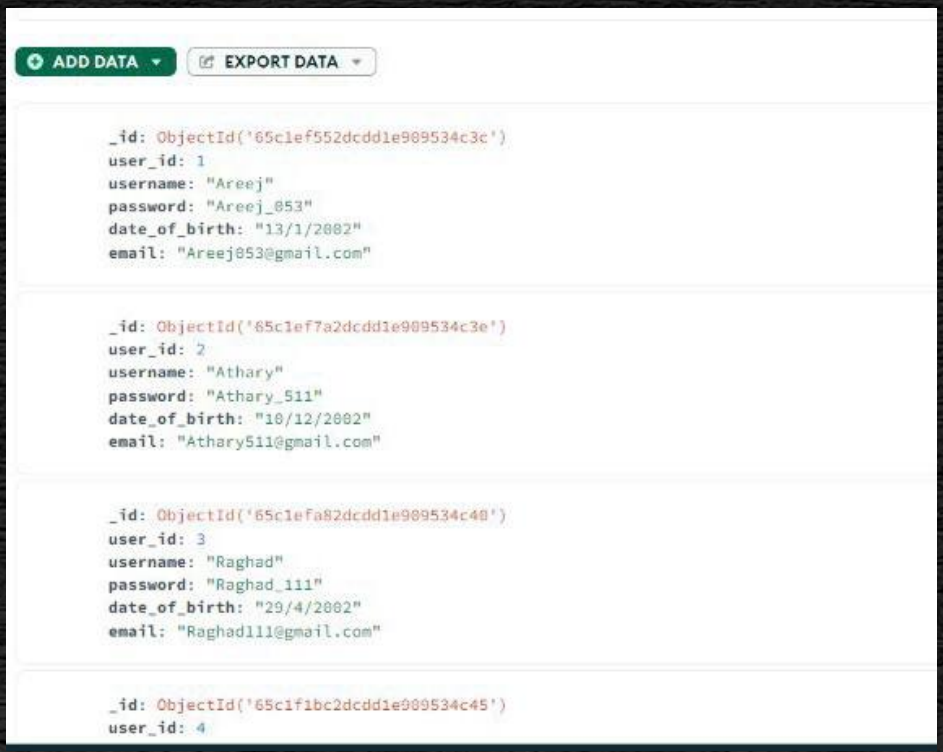
PIPELINE OUTPUT — OUTPUT OPTIONS

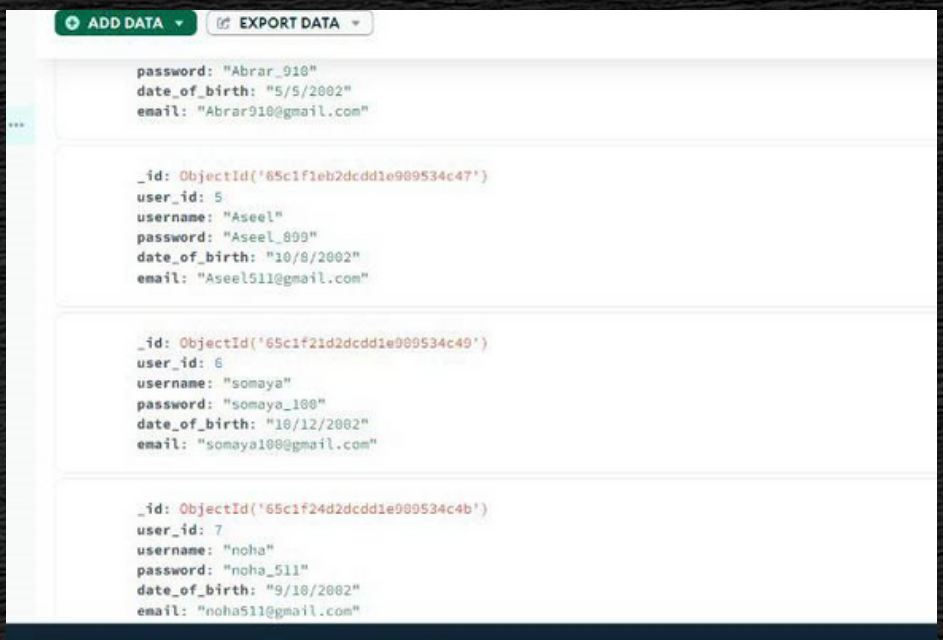
Sample of 1 document

```
comment: "This food is absolutely delicious!"
rating: 5
▸ userInfo: Array (1)
```

retrieves documents from the "user" collection, performs a join with the same collection when user_id = 5

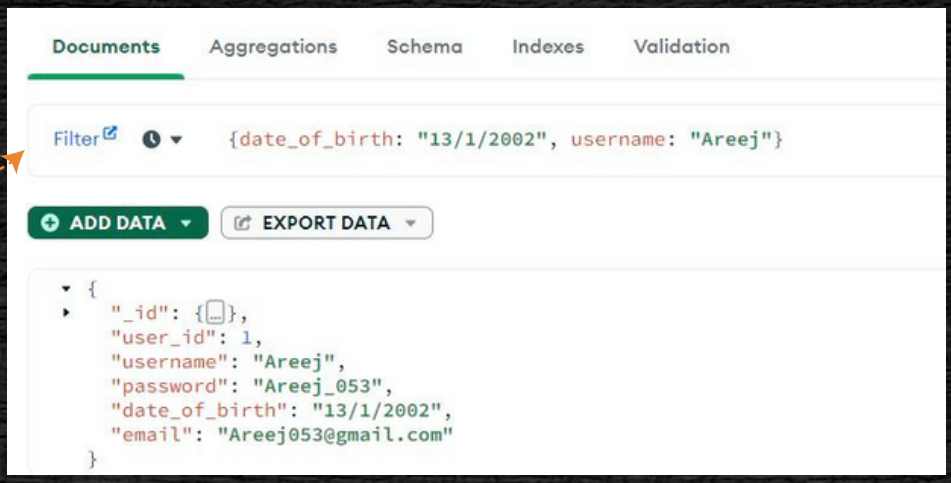
# User Documents



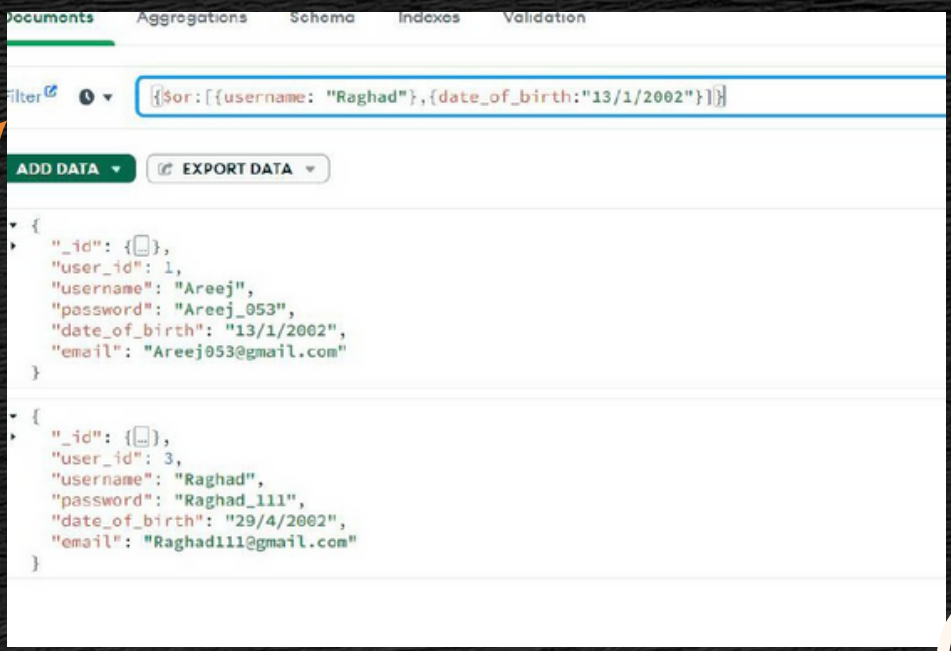




# User Queries

SELECT *FROM user
WHERE date_of_birth = '2002-01-13' AND username = 'Areej




SELECT *FROM user
WHERE username = 'Raghad'  OR date_of_birth = '2002-01-13'



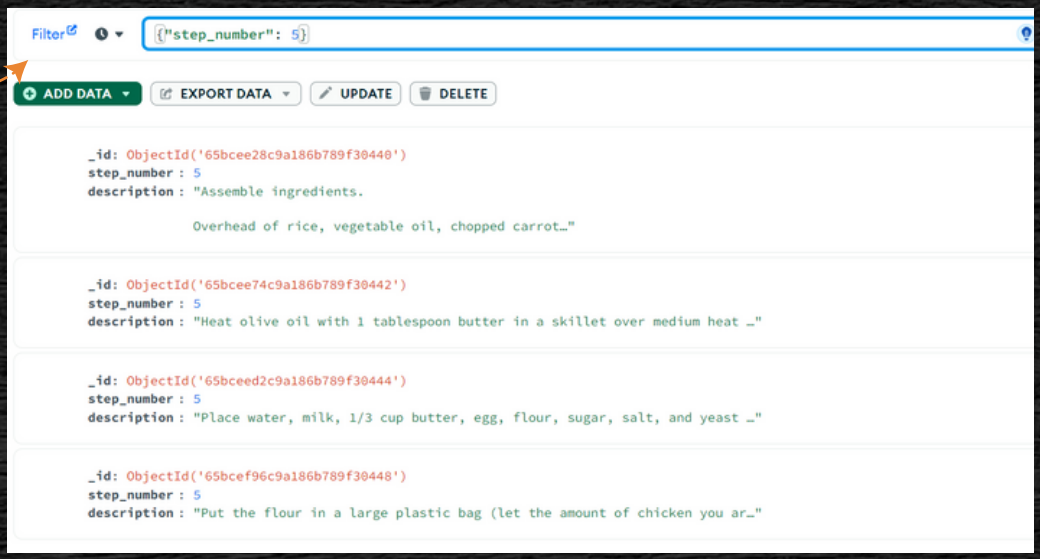
# Instructions Documents


```
▸ _id: Object
  step_number : 4
  description : "Step 1: Preheat the oven to 375 degrees F (190 degrees C). Grease 12 m…"
  recipe_Muffins : ObjectId('65bcecedc9a186b789f30438')

▸ _id: Object
  step_number : 4
  description : "Step 1: Heat olive oil in a small pan over medium heat. Cook and stir …"
  recipe_BlackBeanBowl : ObjectId('65bced7fc9a186b789f3043a')

▸ _id: Object
  step_number : 2
  description : "Step 1: Sift flour, baking powder, sugar, and salt together in a large…"
  recipe_Pancakes : ObjectId('65bcedd6c9a186b789f3043d')

▸ _id: Object
  step_number : 5
  description : "Step 1: Assemble ingredients. Step 2: Place carrots in a small saucepa…"
  recipe_FriedRice : ObjectId('65bcee28c9a186b789f30440')
```

```
▸ _id: Object
  step_number : 5
  description : "Step 1: Heat olive oil with 1 tablespoon butter in a skillet over medi…"
  recipe_Tilapia : ObjectId('65bcee74c9a186b789f30442')

▸ _id: Object
  step_number : 6
  description : "Step 1: Make spice mix: Stir together saffron, cinnamon, allspice, lim…"
  recipe_Kabsa : ObjectId('65bcec34c9a186b789f30436')

▸ _id: Object
  step_number : 5
  description : "Step 1: Place water, milk, 1/3 cup butter, egg, flour, sugar, salt, an…"
  recipe_SweetDinnerRolls : ObjectId('65bceed2c9a186b789f30444')

▸ _id: Object
  step_number : 8
  description : "Step 1: Take your cut up chicken pieces and skin them if you prefer. S…"
  recipe_CrispyFriedChicken : ObjectId('65bcef30c9a186b789f30446')
```


# Instructions Queries

SELECT * FROM Instruction WHERE step_number =5



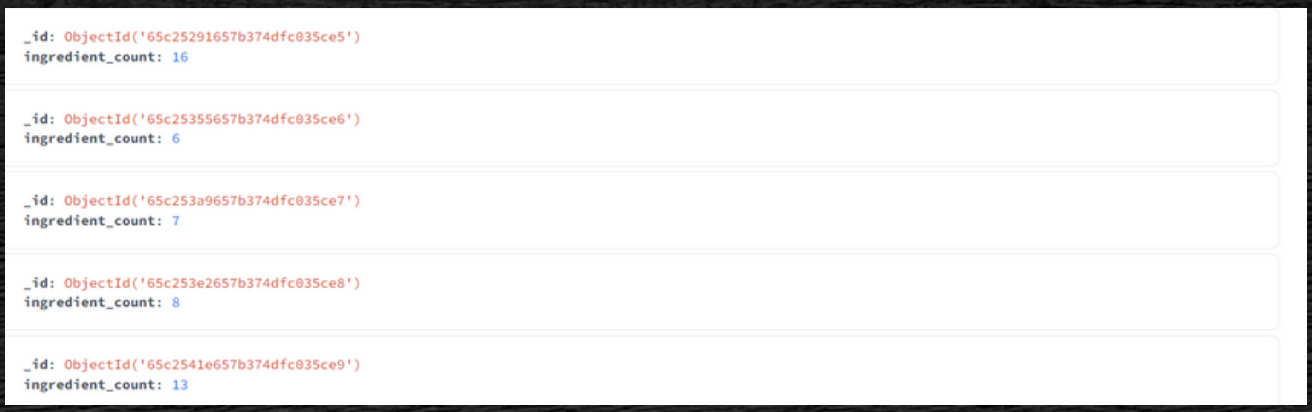

# Ingredients Documents

```
_id: ObjectId('65c25291657b374dfc035ce5')
▸ ingredients : Array (16)
recipe_id : ObjectId('65bcecedc9a186b789f30438')

_id: ObjectId('65c25355657b374dfc035ce6')
▸ ingredients : Array (6)
recipe_id : ObjectId('65bced7fc9a186b789f3043a')

_id: ObjectId('65c253a9657b374dfc035ce7')
▸ ingredients : Array (7)
recipe_id : ObjectId('65bcedd6c9a186b789f3043d')

_id: ObjectId('65c253e2657b374dfc035ce8')
▸ ingredients : Array (8)
recipe_id : ObjectId('65bcee28c9a186b789f30440')

_id: ObjectId('65c2541e657b374dfc035ce9')
▸ ingredients : Array (13)
recipe_id : ObjectId('65bcee74c9a186b789f30442')
```

```
_id: ObjectId('65c2541e657b374dfc035ce9')
▸ ingredients : Array (13)
recipe_id : ObjectId('65bcee74c9a186b789f30442')

_id: ObjectId('65c2545a657b374dfc035cea')
▸ ingredients : Array (23)
recipe_id : ObjectId('65bcec34c9a186b789f30436')

_id: ObjectId('65c25482657b374dfc035ceb')
▸ ingredients : Array (9)
recipe_id : ObjectId('65bceed2c9a186b789f30444')
```

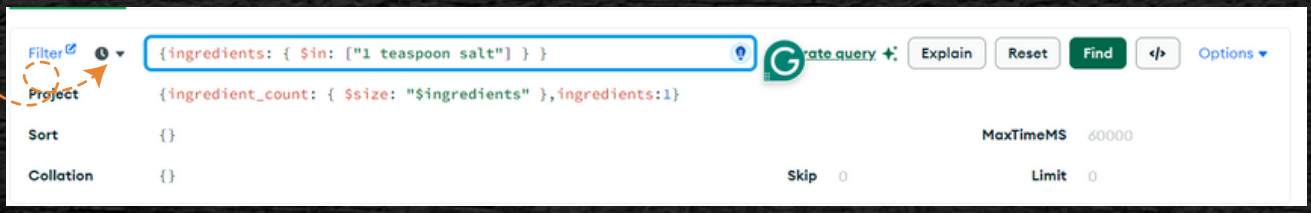
# Ingredients Queries

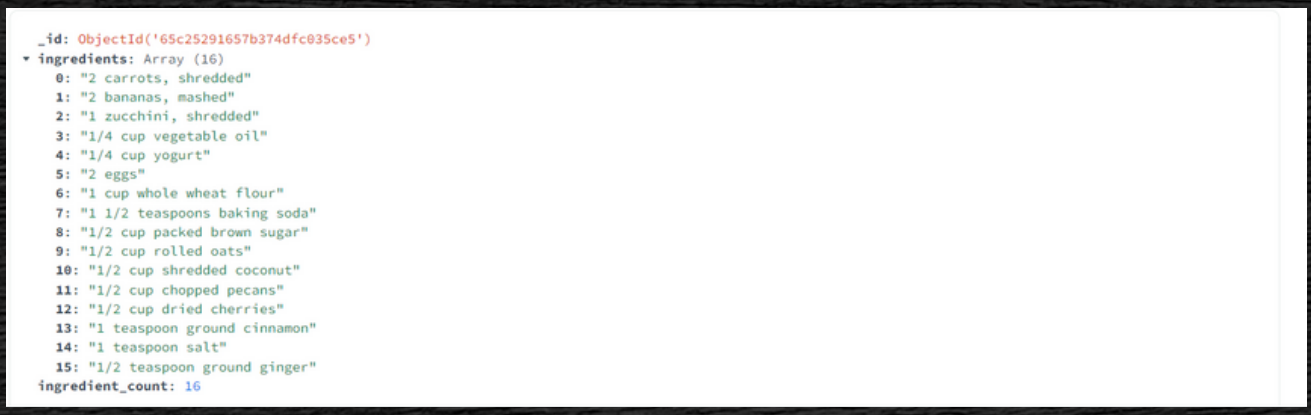
SELECT CARDINALITY(Ingredients) FROM INGREDIENT

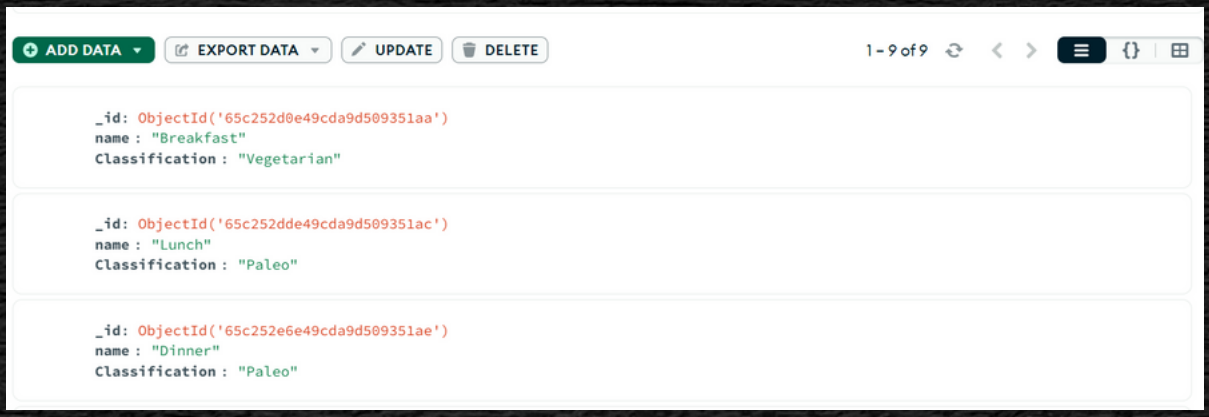
# Ingredients Queries

SELECT * FROM INGREDIENT WHERE “1 teaspoon salt” in Ingredients

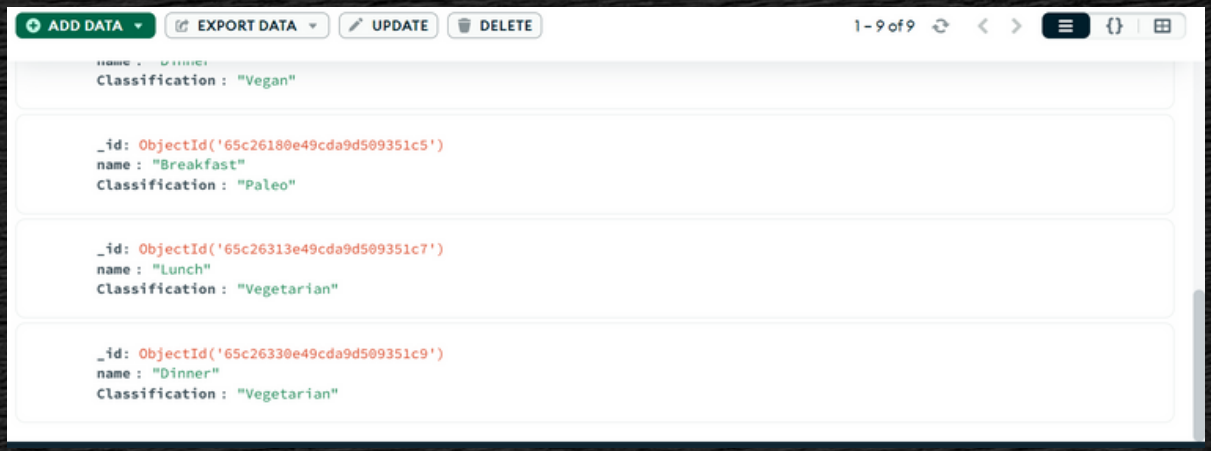


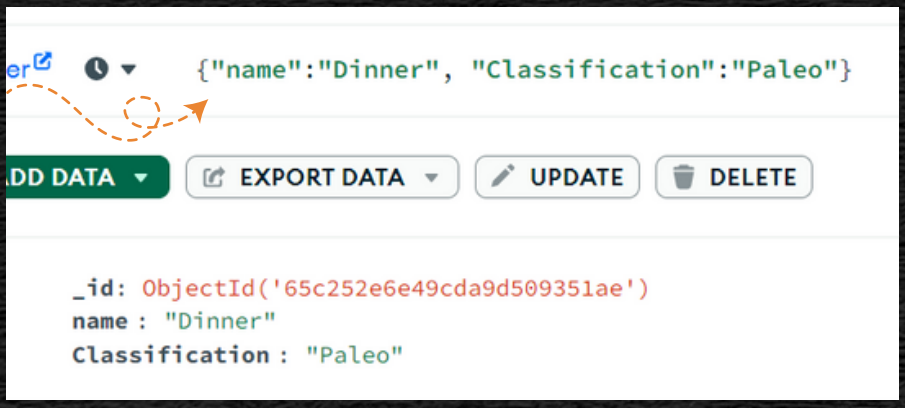
# Category Documents





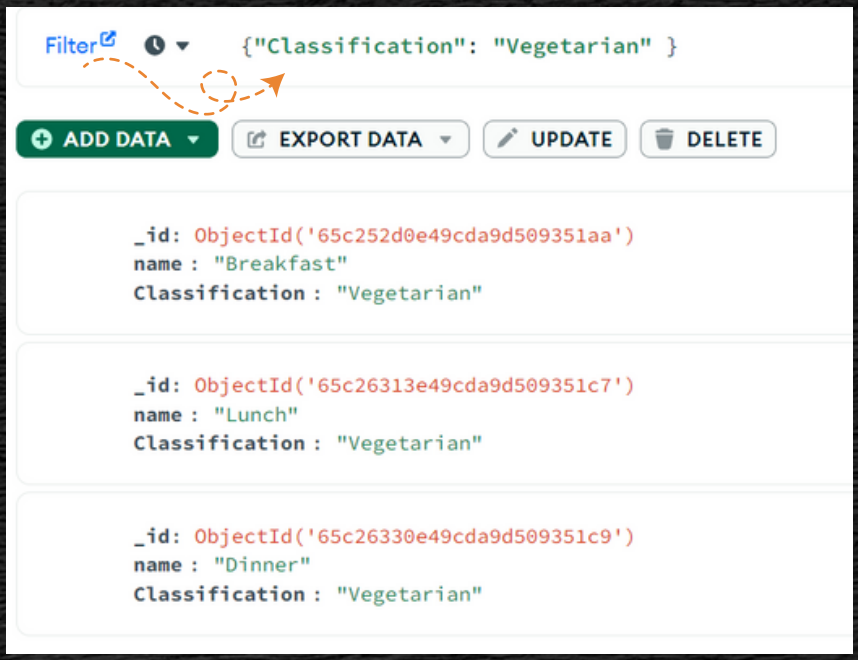


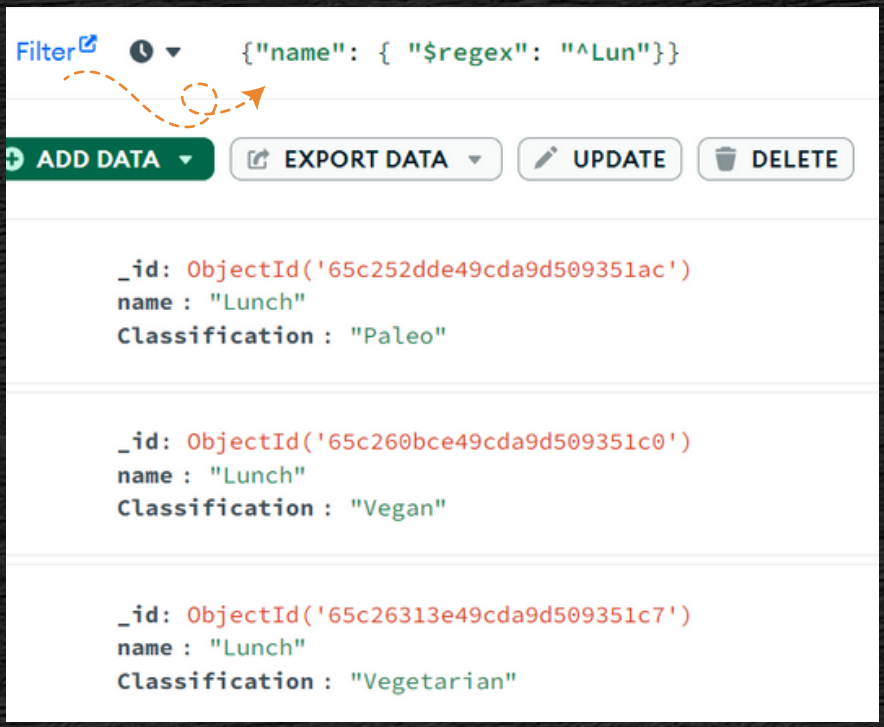
# Category Queries




SELECT * FROM Category WHERE name = 'Dinner' AND Classification = 'Paleo';

SELECT * FROM Category WHERE Classification = 'Vegetarian';




SELECT * FROM Category WHERE name LIKE 'Lun%';

# References


[1] MongoDB. Why Use MongoDB And When To Use It? MongoDB. 6 Jan 2024, [Online]. Available: https://www.mongodb.com/why-use-mongodb

[2] "MongoDB," Wikipedia, The Free Encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/MongoDB.

[3] "History of MongoDB," W3Schools Online Web Tutorials. [Online]. Available: [3] history of mongodb (w3schools.com)

[4] Hevo. (2022, February 11). Best 7 Real-World MongoDB Use Cases. Hevo. Available: https://hevodata.com/learn/mongodb-use-case/.

[5] DotNetTricks, "What is MongoDB and Why to use it?," ScholarHat, Available: https://www.scholarhat.com/tutorial/mongodb/what-is-mongodb-and-why-to-use-it.

[6] "Best 7 Real-World MongoDB Use Cases - Learn | Hevo," Hevo Data, [Online]. Available: https://hevodata.com/learn/mongodb-use-case/#What_are_the_Applications_of_MongoDB.

[7] B. K. Ragala, "Top real world use cases and applications of MongoDB," KnowledgeHut, Sep. 5, 2023. [Online]. Available: https://www.knowledgehut.com/blog/web-development/real-world-use-cases-of-mongodb


# References


[8]Ahmed Z. , "Apache Cassandra: 9 Key Benefits You Should Know," LinkedIn Pulse, Dec. 19, 2023. [Online]. Available: https://www.linkedin.com/pulse/apache-cassandra-9-key-benefits-you-should-know-zareef-ahmed

[9] Prisma, "Advantages of PostgreSQL," Prisma's Data Guide. [Online]. Available: https://www.prisma.io/dataguide/postgresql/benefits-of-postgresql

.

[10] Couchbase, "Why Couchbase," Couchbase Documentation. [Online]. Available: https://docs.couchbase.com/server/current/introduction/why-couchbase.html

[11] MariaDB, "Why MariaDB? Advantages over MySQL," MariaDB Blog. [Online]. Available: https://mariadb.com/resources/blog/why-should-you-migrate-from-mysql-to-mariadb/

[12] Tabassi, A. (2022, April 26, updated August 15, 2022). 5 Benefits of Using Google Firebase. [Online]. Available: https://infotrust.com/articles/5-benefits-of-using-google-firebase/

[13] K. Banker, P. Bakkum, S. Verch, D. Garrett, and T. Hawkins, MongoDB in Action: Covers MongoDB version 3.0. Manning Publications, 2016. Accessed: Jan. 06, 2024. [Online]. Available: https://books.google.com.sa/books/about/MongoDB_in_Action.html?id=xsvsngEACAAJ&redir_esc=y

[14] Ultimate Guide Mongodb: Definition, Advantages & Disadvantages (2023) Ultimate Guide MongoDB: Definition, Advantages & Disadvantages. Available at: https://www.knowledgenile.com/blogs/pros-and-cons-of-mongodb (Accessed: 06 January 2024).


# Task Table

PHASE 1

| | Raghad | Athary | Abrar | Areej | Aseel | Noha | Sumayyah |
|---|---|---|---|---|---|---|---|
| -problem description | | | | | |  | |
| -Goal & Motivation |  |  |  |  |  |  |  |
| Abstract |  |  |  |  |  |  |  |
| Introduction | |  | | | | | |
| Historical Overview | | | |  | | | |
| Usages | | | | |  | | |
| Advantages and Disadvantages | | |  | | | | |
| Alternatives |  | | | | | | |
| Architecture and Components | | | | | | |  |

# Task Table

PHASE 2

| | Raghad | Athary | Abrar | Areej | Aseel | Noha | Sumayyah |
|---|---|---|---|---|---|---|---|
| MongoDB Set up |  |  |  |  |  |  |  |
| Recipe table &data | | |  | |  | | |
| Review table &data |  | | | | | | |
| User table &data | | | |  | | | |
| Category table &data | | | | | |  | |
| Instruction table &data | |  | | | | | |
| RecipeIngredient table &data | | | | | | |  |
| Queries |  |  |  |  |  |  |  |

# Courses

All group members watched these lessons:

| Courses | Link | Hours |
|---|---|---|
| NoSQL VS Relational - MongoDB | https://youtu.be/OlHnhNvCWBU?si=3AT186eMi2CC4UYa | 1:03:09 |
| NOSQL types - MongoDB | https://youtu.be/WurD9y6AqaE?si=C8iy-WLM3VlqIm-- | 0:22:06 |
| Category table &data | https://youtu.be/YeNq4VnxZeg?si=8Y13sx4P64yQq4IK | 1:20:42 |
| CRUD Insert - find - MongoDB | https://youtu.be/ICR2Kmz8nik?si=bKvgMloVAhK1EGLU | 1:31:25 |
| CRUD Update - Delete - MongoDB | https://youtu.be/cMDwH3ODTP4?si=-JrOC-ifTjobn5kF | 0:10:21 |
| Relations - referencing - MongoDB | https://youtu.be/j92gNo2Y1r0?si=jdjZFZ_2T9mQh6Nb | 0:26:10 |
| Collection schema - Indexing - Backup and restore | https://youtu.be/FzRAwOo4gUU?si=53p8yT4jFZjgjmOb | 0:50:16 |
| How to Update Documents in MongoDB | https://youtu.be/p2zqQxsZsBk?si=o5FW9tZltHl6W_ho | 00:46:02 |
| Views, Capped Collections, Text Search in MongoDB | https://youtu.be/T-QU7K6vWfU?si=RKvOs8YzsUYNblgG | 0:33:30 |
| Total | | 7:03:41 |