MongoDB Installation and Features for Mac and Windows

Kyle-Jacob La'akea Gamiao[1]
Chaminade University[1], kyle-jacob.gamiao@student.chaminade.edu

Keywords: Final Project, Tutorial, Non-relational Database, MongoDB, Installation, Features

Table of Contents

1 Description

The following pages will describe for both Mac and PC users, how to install and start using *MongoDB*. Before installing, configuring, and using *MongoDB* it is important to at least understand what *MongoDB* is. *MongoDB* is, "an open source NoSQL[1] database management program," and can be used to, "manage document-oriented information, store, or retrieve information," (Gillis & Botelho, 2023). This makes it exceedingly useful when you are working with sets of data that are large and need to be distributed (like to people within an organization).

2 Key Features (Compared to *MySQL*)

While both *MySQL* and *MongoDB* are free-to-use and open-source software, there are some stark differences. While this list of differences is not comprehensive, it still covers some of the major key points between the two programming languages/styles.

Section 2.1 Data Storage

The most glaringly obvious difference is that *MySQL* is a structured query language. The data is stored in rows and columns which make up tables. The user must then, "construct an SQL query that joins multiple tables together to create the view on the data they require. Database schemas and data models need to be defined ahead of time, and data must match this schema to be stored in the database," (*Comparing the Differences - MongoDB vs MySQL*, n.d.). These also follow the same rigid outline.

Storing data this way is relatively safer but loses flexibility. Especially if/when the user has to update information, a longer process must occur. Depending on how big the database has grown or currently is, as well as the information being added, it has the potential to be both difficult and expensive. This isn't to say that a rigid procedure to database creation is bad, it just doesn't fit the purpose that *MongoDB* does.

In *MongoDB* information is stored in, "JSON-like documents (actually stored as binary JSON, or BSON)[2], as opposed to the table and row format of relational systems," (*Comparing the Differences - MongoDB vs MySQL*, n.d.). This allows for a more flexible approach to coding the data.

Section 2.2 Scalability

Scalability for *MongoDB* is greater due to a feature called a, "sharded cluster," which allows, "a portion of the database to be configured as a replica set[3]," (*Comparing the Differences - MongoDB vs MySQL*, n.d.). Thus allowing you to scale up or scale down your database(s). This horizontal scaling is something that *MongoDB* does better than *MySQL*.

[1] Stands for not only SQL or non-relational and are, "non-tabular databases and store data differently than relational tables. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads," (*What is NoSQL?, 2023).*

[2] "JSON and BSON are close cousins, as their nearly identical names imply, but you wouldn't know it by looking at them side by side. JSON, or JavaScript Object Notation, is the wildly popular standard for data interchange on the web, on which BSON (Binary JSON) is based," (*JSON and BSON*, n.d.).

[3] "A replica set is the replication of a group of MongoDB servers that hold the same data, ensuring high availability and disaster recovery," (*Comparing the Differences - MongoDB vs MySQL*, n.d.).

In *MySQL* scalability tends to be more limited, due to the fact that you can either choose vertical stability or adding read replicas (both of which push the systems to their limits).

Section 2.3 Flexibility

MongoDB lacks the use of a schema[4] which contributes to how easy it is to improve, build on, and enhance applications overtime. *MySQL* in comparison requires longer and more complex processes to do similar actions. The specifics are more complex than this, but the general idea is still as previously mentioned. Relational databases tend to require more expensive and intricate processes in order to build upon and/or enhance a pre existing database.

3 Download, Installation, and Setup

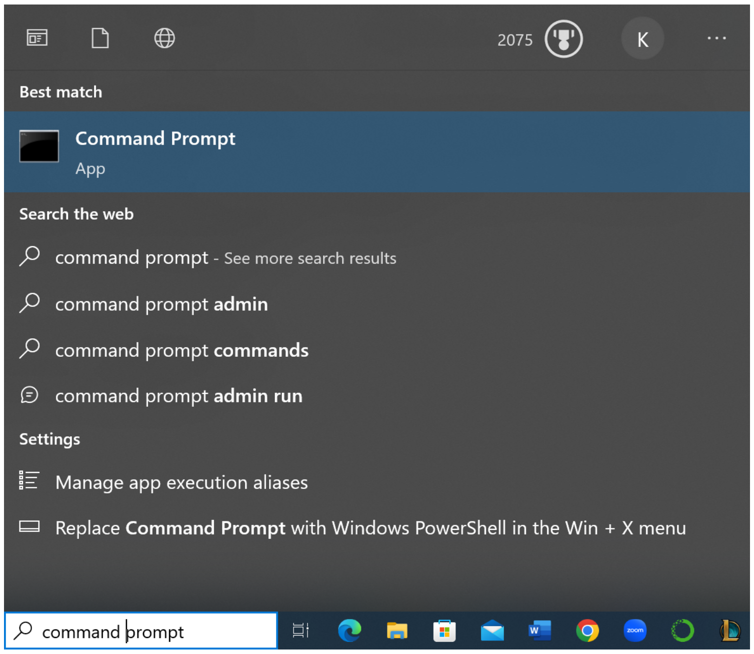
Section 3.1 Windows Procedures

Below are the step-by-step instructions for downloading, installing and setting up *MongoDB* on Windows.

Section 3.1.1 Downloading for Windows

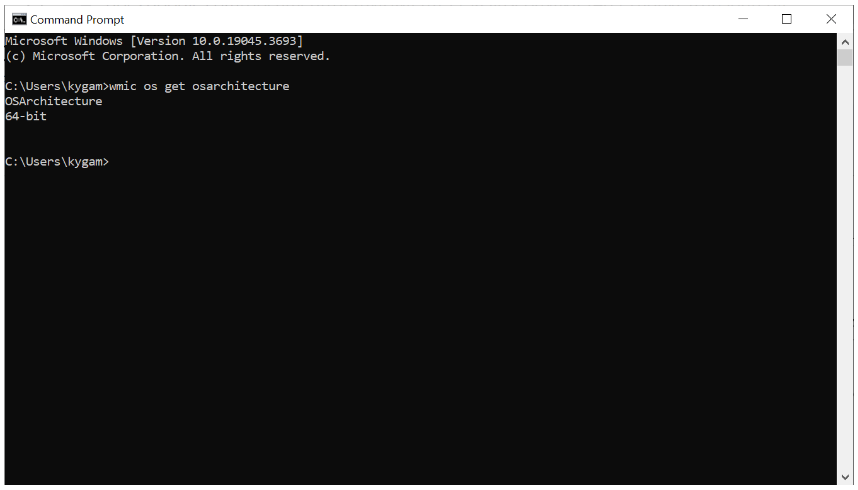
Determine if you are able to use *MongoDB* on your current computer.

Step 1: In order to see which version you're currently running, go to your "Command Prompt" application that comes preinstalled on most PC computers.



Step 2: Input the following command prompt: **wmic os get osarchitecture** and hit enter.

[4] "A database schema refers to the logical and visual configuration of the entire relational database. The database objects are often grouped and displayed as tables, functions, and relations," (*What is a Database Schema?,* n.d.)

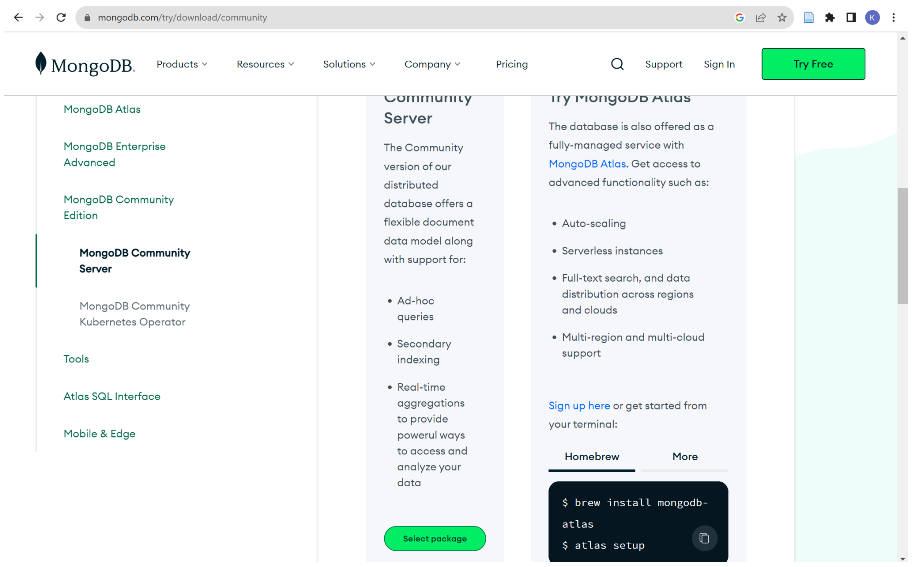

```
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kygam>wmic os get osarchitecture
OSArchitecture
64-bit


C:\Users\kygam>
```

As you can see, below the code, the “64-bit” information is displayed. So you can use the most recent version of *MongoDB*.

Step 3: Type https://www.mongodb.com/try/download/community into your preferred internet browser. There is a section for “Community Server”, below it should be a prompt to “Select package” click that.



Community Server

The Community version of our distributed database offers a flexible document data model along with support for:

- Ad-hoc queries
- Secondary indexing
- Real-time aggregations to provide powerul ways to access and analyze your data

Try MongoDB Atlas

The database is also offered as a fully-managed service with MongoDB Atlas. Get access to advanced functionality such as:

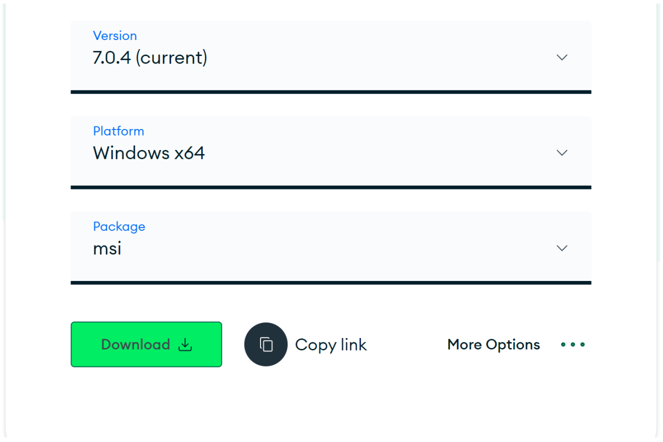
- Auto-scaling
- Serverless instances
- Full-text search, and data distribution across regions and clouds
- Multi-region and multi-cloud support

Sign up here or get started from your terminal:

```
$ brew install mongodb-atlas
$ atlas setup
```

This will bring up a new option where you can select the version, platform, and package that you would like to install.

Step 4: For “Version” select the most current version. For the “platform”, select the one that corresponds to your computer, and for the “package” select msi. Then click “Download”



Version
7.0.4 (current)

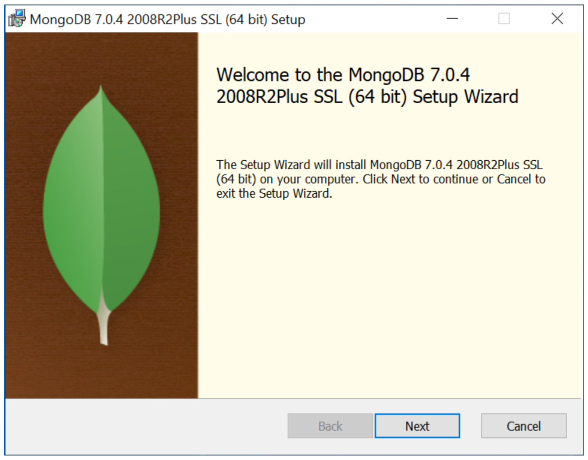
Platform
Windows x64

Package
msi

Section 3.1.2 Installation and Setup for Windows

Locate your download file from the end of *Section 3.1.1*.

Step 1: Open your file and the following screen should appear. Hit "next".

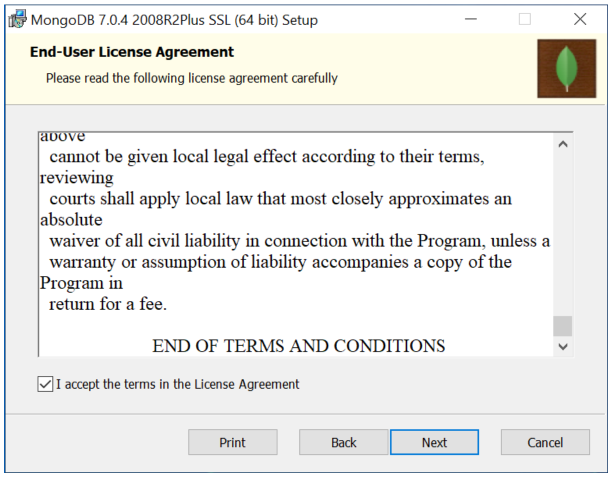


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup

Welcome to the MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard will install MongoDB 7.0.4 2008R2Plus SSL (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Step 2: Read through the End-User License Agreement, click the checkbox if you agree to it, and hit "next" again.



MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup

End-User License Agreement

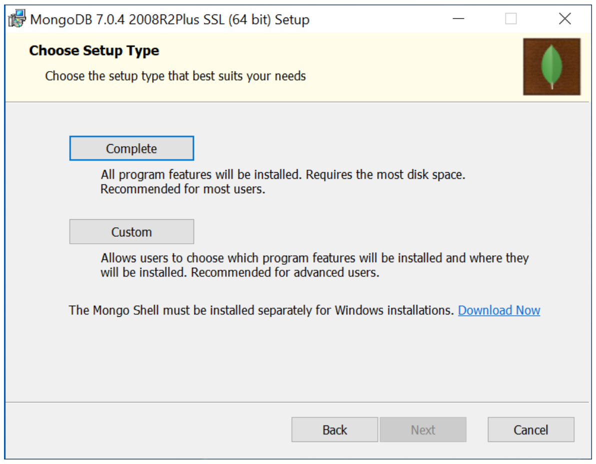
Please read the following license agreement carefully

```
above
 cannot be given local legal effect according to their terms,
reviewing
 courts shall apply local law that most closely approximates an
absolute
 waiver of all civil liability in connection with the Program, unless a
 warranty or assumption of liability accompanies a copy of the
Program in
 return for a fee.

            END OF TERMS AND CONDITIONS
```

☑ I accept the terms in the License Agreement

Step 3: For your setup type, opt for the complete setup (unless you consider yourself an advanced user). Click "Complete".


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup

Choose Setup Type

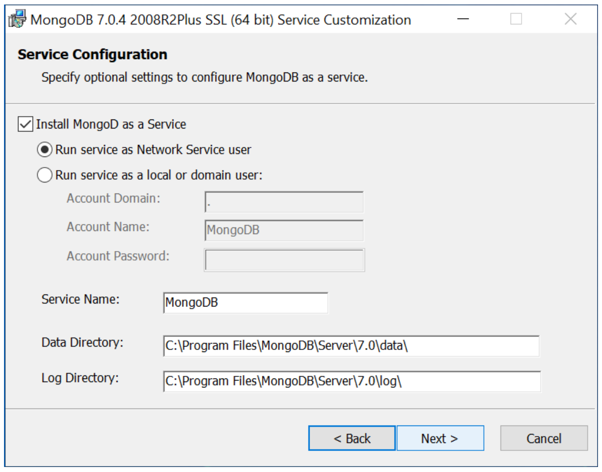
Choose the setup type that best suits your needs

All program features will be installed. Requires the most disk space. Recommended for most users.

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

The Mongo Shell must be installed separately for Windows installations. Download Now

Step 4: Check "Install MongoD as a Service" and then click "Next".


MongoDB 7.0.4 2008R2Plus SSL (64 bit) Service Customization

Service Configuration

Specify optional settings to configure MongoDB as a service.

[✓] Install MongoD as a Service

● Run service as Network Service user

○ Run service as a local or domain user:

Account Domain: .

Account Name: MongoDB

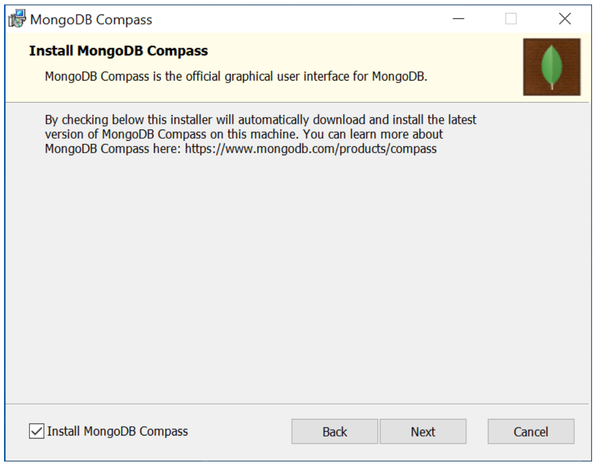
Account Password:

Service Name: MongoDB

Data Directory: C:\Program Files\MongoDB\Server\7.0\data\

Log Directory: C:\Program Files\MongoDB\Server\7.0\log\

Step 5: Make sure the "Install MongoDB Compass" option is checked and then hit next.


MongoDB Compass

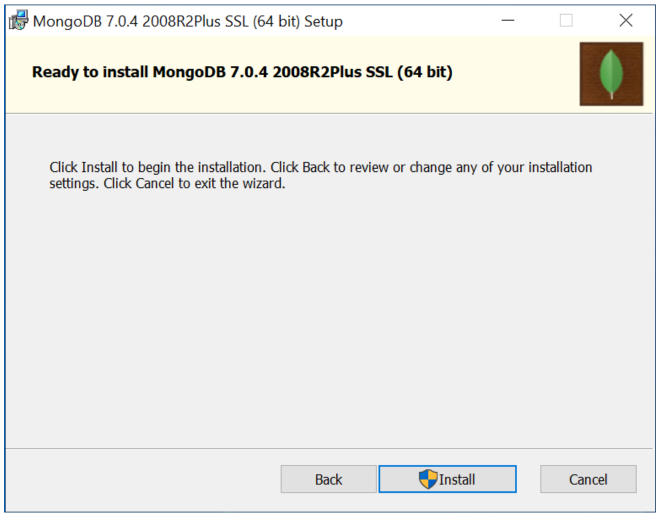
Install MongoDB Compass

MongoDB Compass is the official graphical user interface for MongoDB.

By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: https://www.mongodb.com/products/compass

[✓] Install MongoDB Compass

Step 6: On this final screen, click "Install".



MongoDB 7.0.4 2008R2Plus SSL (64 bit) Setup

Ready to install MongoDB 7.0.4 2008R2Plus SSL (64 bit)

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

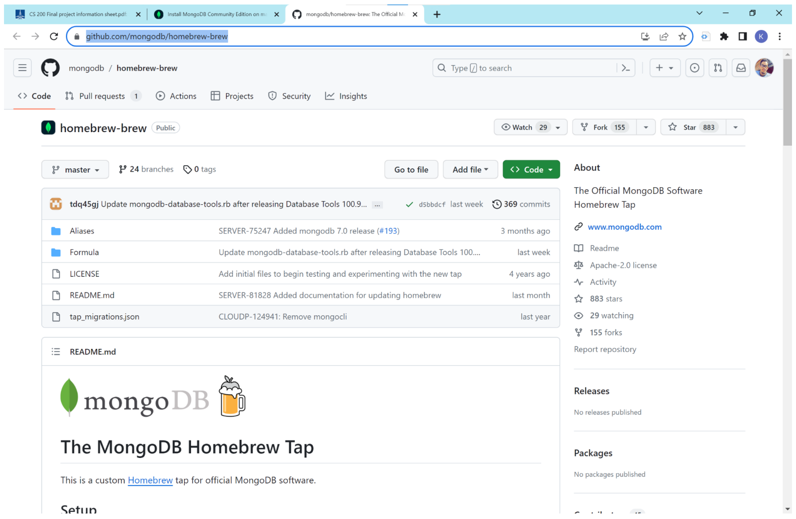
Depending on how well your computer runs, it may take a long time for all of *MongoDB* to be installed. Once installed, simply hit "Finish" and you should be able to use *MongoDB*.

Section 3.2.1 Downloading, installing, and setting up for Mac

Step 1: Install the" Xcode command-line tools" by searching in "Finder" for the "Terminal" Application. Then type in the following command prompt: **xcode-select --install** and hit enter. Leave the "Terminal" open.

Step 2: Install "Homebrew" by typing the following command prompt into "Terminal": **/bin/bash -c "$(curl -fsSLhttps://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"** and hit enter.

Step 3: Go to https://github.com/mongodb/homebrew-brew to download "MongoDB Homebrew Tap"



github.com/mongodb/homebrew-brew

mongodb / homebrew-brew

homebrew-brew Public

master · 24 branches · 0 tags

tdq45gj Update mongodb-database-tools.rb after releasing Database Tools 100.9... d5bbdcf last week · 369 commits

Aliases	SERVER-75247 Added mongodb 7.0 release (#193)	3 months ago
Formula	Update mongodb-database-tools.rb after releasing Database Tools 100....	last week
LICENSE	Add initial files to begin testing and experimenting with the new tap	4 years ago
README.md	SERVER-81828 Added documentation for updating homebrew	last month
tap_migrations.json	CLOUDP-124941: Remove mongocli	last year

About

The Official MongoDB Software Homebrew Tap

www.mongodb.com

Readme · Apache-2.0 license · Activity · 883 stars · 29 watching · 155 forks

Releases: No releases published

Packages: No packages published

README.md

The MongoDB Homebrew Tap

This is a custom Homebrew tap for official MongoDB software.

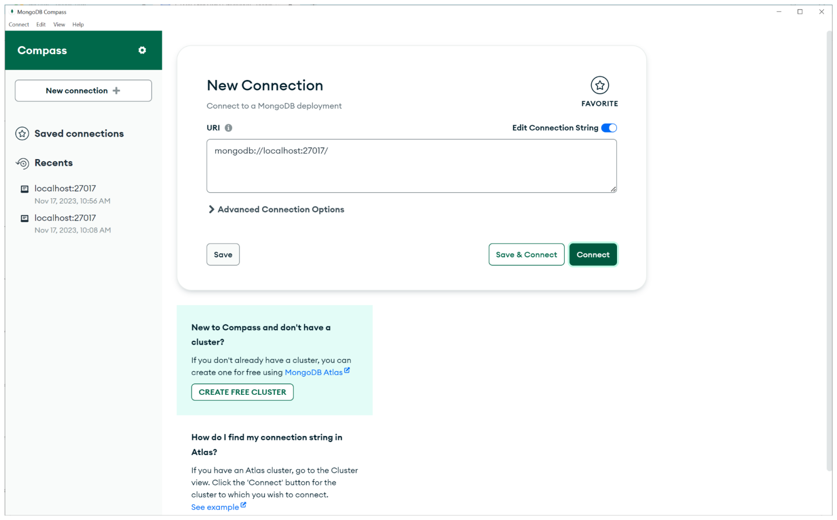
Setup

Step 4: Go back to your terminal and type the following command prompt: **brew tap mongodb/brew** and hit enter. This will install the latest version of *MongoDB* Community Server.

4 Basic Concepts

Before getting into code cases, it is important to learn the fundamental/basic concepts necessary to actually code. The following subsections will detail textually and graphically how to use some of these basic codes, before it is actually utilized in a code case. Most of these concepts are meant to be learned and practiced sequentially.

Step 1: Open up *MongoDB* and hit "Connect".



Connect to a MongoDB deployment

```
mongodb://localhost:27017/
```

localhost:27017
Nov 17, 2023, 10:56 AM

localhost:27017
Nov 17, 2023, 10:08 AM

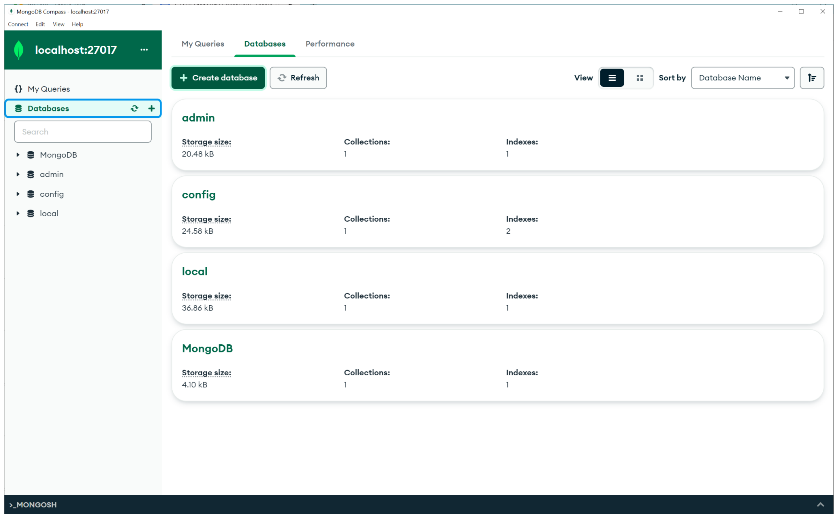
New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using MongoDB Atlas

How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

Step 2: Click "Mongosh" on the bottom of the screen.

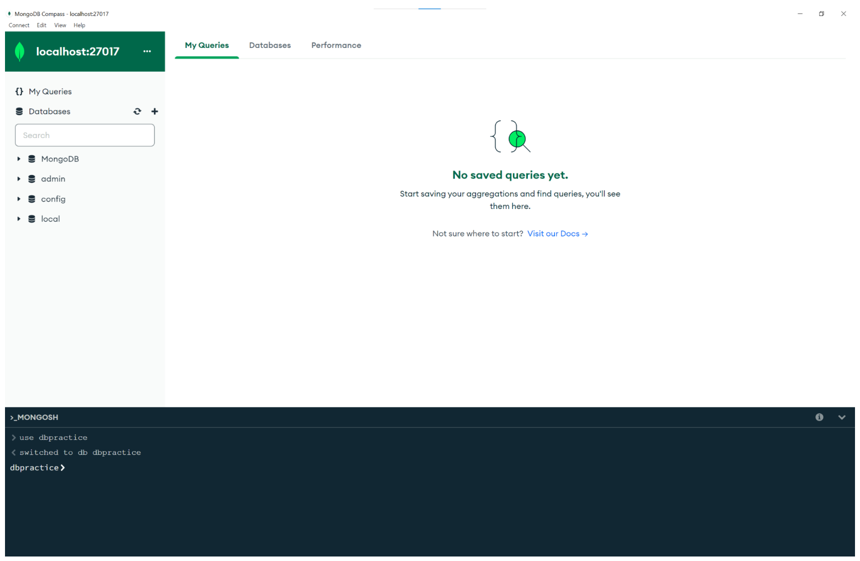


	Storage size:	Collections:	Indexes:
admin	20.48 kB	1	1
config	24.58 kB	1	2
local	36.86 kB	1	1
MongoDB	4.10 kB	1	1

You now have access to the coding aspect of *MongoDB*. It may be slower than using the interface, but this guide will detail how to code for everything in the event you don't have access to features like the graphical user interface.

Section 4.1 Creating a Database

In mongosh, type the command: **use ______** the "______" will be substituted with whatever you want to title your database. In this instance, **use dbpractice**, is the command being used. Then hit enter.



```
> use dbpractice
< switched to db dbpractice
dbpractice>
```

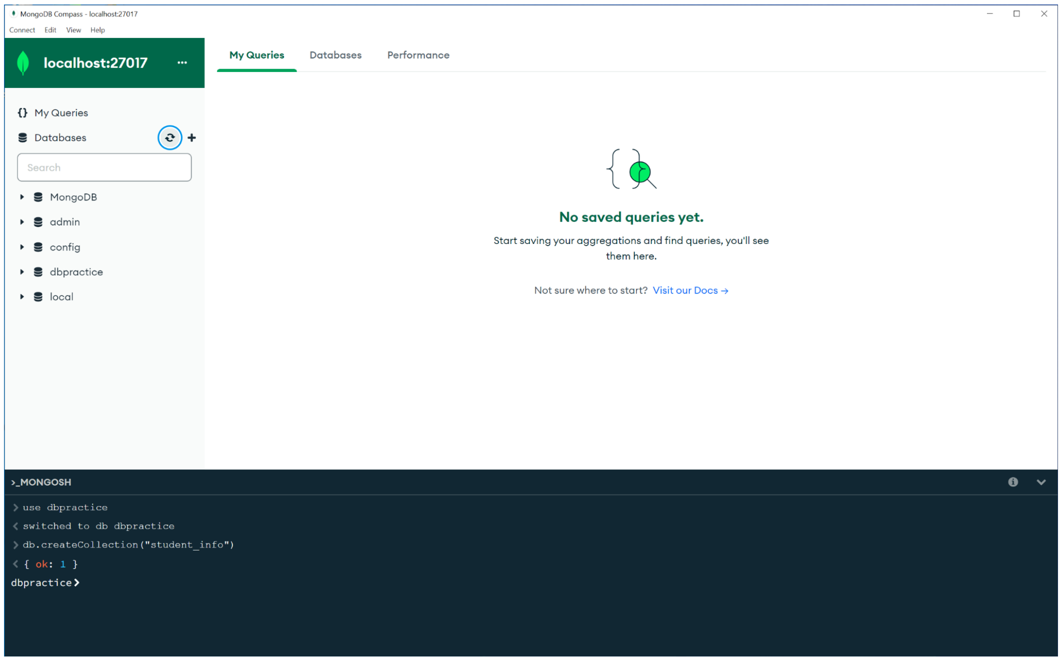
The left side of mongosh should now have **"the name of your database">**. Ordinarily if you refresh your server the database will pop up, but since there is no data inserted into it, you cannot see the table yet.

Section 4.2 Creating Collections

Creating collections in NoSQL databases is akin to creating tables in SQL databases. They are labeled differently to help distinguish the two, and because they do not necessarily function the same way.

Code: db.createCollection("student_info")

Then click enter. Now that you've added some form of information to the database, you may refresh the server at the top left and the new database you created should appear, along with the collection. **Important note: do not use spaces or special characters when naming your collection**.



```
> use dbpractice
< switched to db dbpractice
> db.createCollection("student_info")
< { ok: 1 }
dbpractice>
```

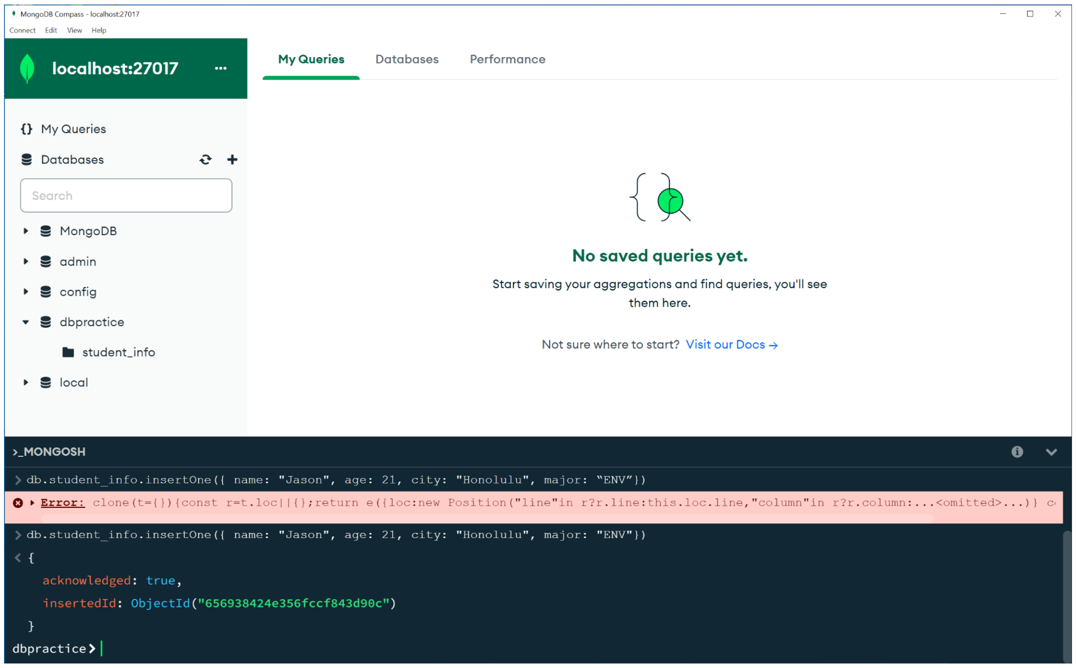
Section 4.3 Inserting Information Into Collections

This section will be broken up into two subsections. The first will be simply inserting one line of information. The second subsection will contain information on how to insert more than one line of information to a collection at a time.

Section 4.3.1 Inserting One Line of Information

Code: db.thenameofyourcollection.insertOne({ name: "Jason", age: 21, city: "Honolulu", major: “ENV”})

You basically use the first part of the code above, followed by the information you want to insert in the parentheses. The first word basically designates a category, followed by a colon. Then the information is input for that category. The first category is “name”. The following data to be input is “Jason”. “Jason” is a name, which is a word. Words are typically characterized as “characters” or “strings”. So they belong inside quotation marks. If there is more information in this line you follow it up with a comma and then the next bit of information. The next category is age. Categories do not need to be put in quotation marks. This is followed by a colon, and then the data. The data in this case is 21. A number, or in coding language, any whole number is typically seen as an integer. Integers do not need quotation marks, unless you want them to be read as a string/character. The process is repeated until all the data for this line is inserted and the last section of data ends with the parentheses instead of a comma.



No saved queries yet.

Start saving your aggregations and find queries, you'll see them here.

Not sure where to start?

```
> db.student_info.insertOne({ name: "Jason", age: 21, city: "Honolulu", major: “ENV”})
Error: clone(t={}){const r=t.loc||{};return e({loc:new Position("line"in r?r.line:this.loc.line,"column"in r?r.column:...<omitted>...)} c
> db.student_info.insertOne({ name: "Jason", age: 21, city: "Honolulu", major: "ENV"})
< {
    acknowledged: true,
    insertedId: ObjectId("656938424e356fccf843d90c")
  }
dbpractice>
```

As you can see, originally the code threw an error, because the quotation marks around “ENV” were incorrect. Small details like this are important to double-check because they can lead to errors and issues/bugs in the code.

Section 4.3.2 Inserting Multiple Lines of Information

This is similar to the previous code. The biggest difference is that you separate each line from each other within the curly brackets “{}” and commas. You also use regular brackets “[]” after the first and last parentheses. The benefit is that you are able to streamline writing multiple lines of information for your database, without having to add each line individually. Hit CTRL+Enter to go to the next line, manually fix the indentation as you see fit.

Code:

```
db.student_info.insertMany([
        {name: "Jeff", age: 22, city: "Aiea", major: "SO"},
        {name: "Andrew", age: 21, city: "Honolulu", major: "ENV"},
        {name: "Zack", age: 24, city: "Kaneohe", major: "PSY"}
```

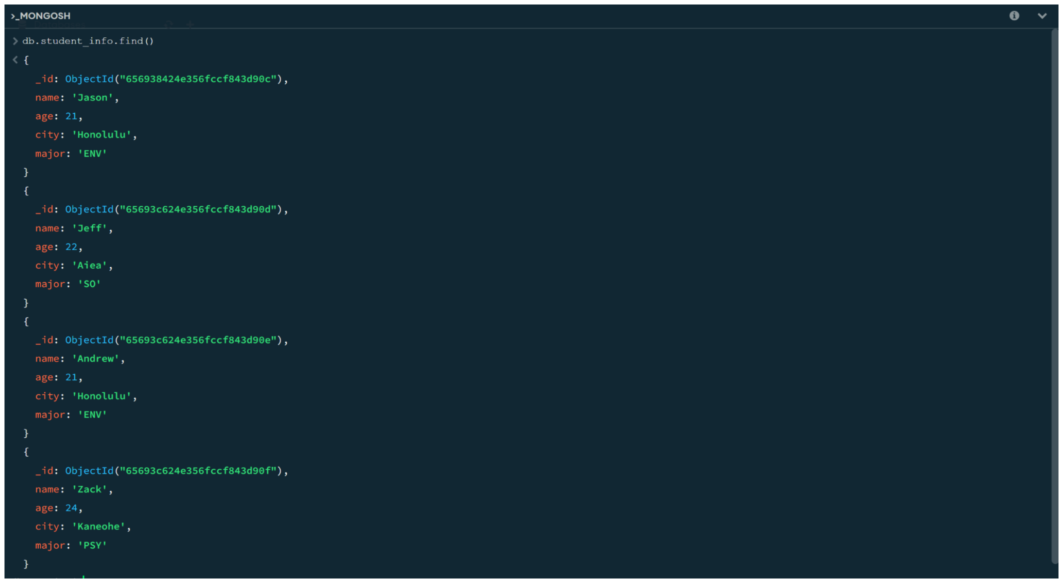
Section 4.4 Querying

There are multiple ways to query information. This guide will not cover all of them, but some of the more frequently used ones.

Section 4.4.1 Querying All Data in a Collection

Code: db.thenameofyourcollection.find()

Example: db.student_info.find()



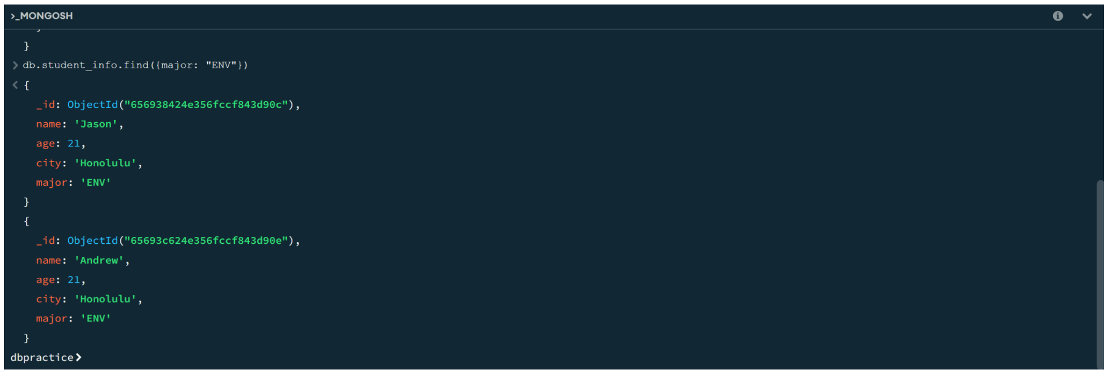
```
>_MONGOSH
> db.student_info.find()
< {
    _id: ObjectId("656938424e356fccf843d90c"),
    name: 'Jason',
    age: 21,
    city: 'Honolulu',
    major: 'ENV'
  }
  {
    _id: ObjectId("65693c624e356fccf843d90d"),
    name: 'Jeff',
    age: 22,
    city: 'Aiea',
    major: 'SO'
  }
  {
    _id: ObjectId("65693c624e356fccf843d90e"),
    name: 'Andrew',
    age: 21,
    city: 'Honolulu',
    major: 'ENV'
  }
  {
    _id: ObjectId("65693c624e356fccf843d90f"),
    name: 'Zack',
    age: 24,
    city: 'Kaneohe',
    major: 'PSY'
  }
```

Section 4.4.2 Querying Using Find

Code: db.thenameofyourcollection.find(category you want: data you want)

Example: db.student_info.find({major: "ENV"})

This should pull up two student’s information, Jason who we originally inputted in the single line insert and Andrew from the multiple line insert.



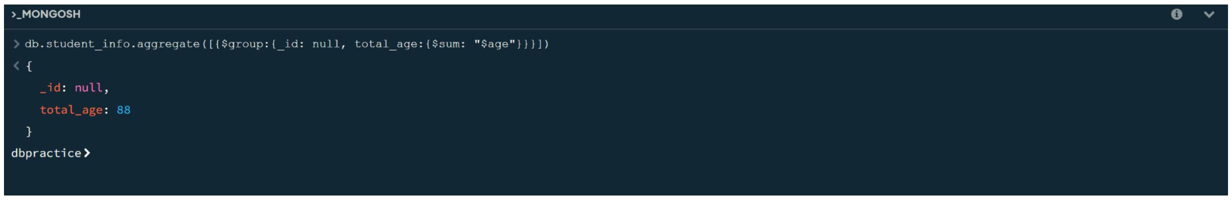
```
>_MONGOSH
  }
> db.student_info.find({major: "ENV"})
< {
    _id: ObjectId("656938424e356fccf843d90c"),
    name: 'Jason',
    age: 21,
    city: 'Honolulu',
    major: 'ENV'
  }
  {
    _id: ObjectId("65693c624e356fccf843d90e"),
    name: 'Andrew',
    age: 21,
    city: 'Honolulu',
    major: 'ENV'
  }
dbpractice>
```

Section 4.4.3 Querying the Total of a Category

This only works for categories where the data is entirely made up of integers.

Code: db.thenameofyourcollection.aggregate([{ $group: { _id: null, total_nameofcategory: { $sum: "$nameofcategory" } } }])

Example: db.student_info.aggregate([{$group:{_id: null, total_age:{$sum: "$age"}}}])

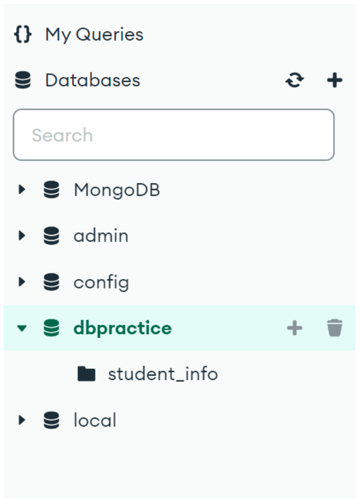


```
>_MONGOSH
> db.student_info.aggregate([{$group:{_id: null, total_age:{$sum: "$age"}}}])
< {
    _id: null,
    total_age: 88
  }
dbpractice>
```

Section 4.5 Saving Code and Transferring to a Shareable Format

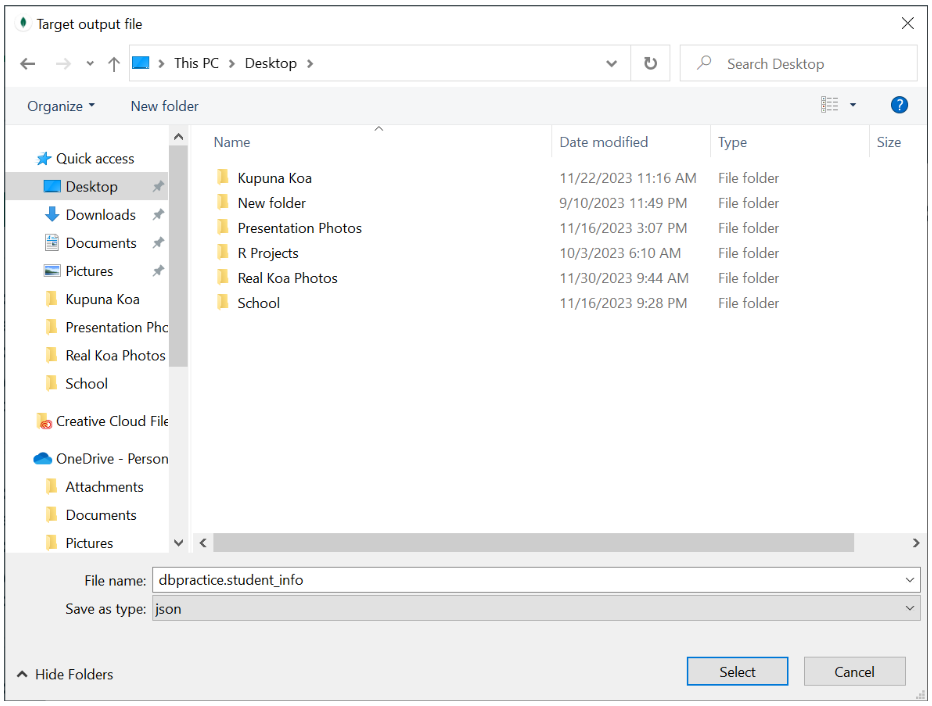
The information for the database you created and the individual documents on them are already saved to *MongoDB*. In order to share them you must export them into a script, typically in the form of a JSON document.

Step 1: Click on the document you want to save externally and/or share



Step 2: Click "Export Data". You will be prompted on what you would like to export and in what form you would like to export it in. For now we chose to export everything, and in the form of JSON.

Step 3: Name your document and choose where you would like it to be saved to.



Name	Date modified	Type	Size
Kupuna Koa	11/22/2023 11:16 AM	File folder	
New folder	9/10/2023 11:49 PM	File folder	
Presentation Photos	11/16/2023 3:07 PM	File folder	
R Projects	10/3/2023 6:10 AM	File folder	
Real Koa Photos	11/30/2023 9:44 AM	File folder	
School	11/16/2023 9:28 PM	File folder	

File name: dbpractice.student_info

Save as type: json

5 Code Cases

Disclaimer: The following code cases are made up of imaginative figures. The data and names are entirely fictitious and in no way correspond or correlate to real people and their information. These cases are designed to familiarize users with MongoDB and most of the basic functions one can do within it.

Section 5.1 Code Case 1

Using the following data tables:

1. Create a database with three collections
2. Five categories and corresponding data inputs for each collection
3. Query all data
4. Query using "find" or "aggregate"
5. Save and export the code case

Table 1. Private Information

Name	Age	Gender	Marital Status	Income
Kyle	23	M	Married	$82,000
Alex	32	M	Single	$47,000
Jack	27	M	Divorced	$67,000

Table 2. Public Information

Name	City	State	Home Phone Number	University
Kyle	Honolulu	HI	(808) 555 - 7323	Chaminade University of Honolulu
Alex	Seattle	WA	(206) 555 - 4276	University of Hawaii at Manoa
Jack	Hilo	HI	(808) 555 - 7449	Unversity of Hawaii at Hilo

Table 3. Random Information

Name	Favorite Food	Favorite Drink	Favorite Number	Favorite Animal
Kyle	Rice	Sprite	7	Lion
Alex	Steak	Coke	39	Wolf
Jack	Salad	Pepsi	42	Walrus

Once you complete the exercise, go to the answer key section of this document to double-check your work. Abbreviations may vary, as well as titles, but results should be generally the same.

Section 5.2 Code Case 2

Don't worry about italicizing the scientific names. Font changes are not necessary in this case.

Using the following data tables:

1. Create a database with three collections
2. Five categories and corresponding data inputs for each collection
3. Query all data
4. Save and export the code case

Table 1. Endangered Mammals

Animal	Scientific Name	Primary Location	Amount in the Wild	Amount in Captivity
Gorilla	*Gorilla beringei*	Eastern Africa	570	100
Orangutans	*Pongo pygmaeus*	Central Africa	350	15
Killer Whale	*Orcinus orca*	Atlantic Ocean	300	20

Table 2. Endangered Plants

Plant	Scientific Name	Primary Location	Amount in the Wild	Amount in Nurseries
Ohi'a Lehua	*Metrosideros polymorpha*	Hawai'i	312	200
Koa	*Acacia Koa*	Hawai'i	421	100
Limu Kohu	*Asparagopsis taxiformis*	Hawai'i	444	123

Table 3. Endangered Sea Creatures

Animal	Scientific Name	Primary Location	Amount in the Wild	Amount in Captivity
Red Abalone	*Haliotis rufescens*	Pacific Ocean	420	200
Sea Otters	*Enhydra lutris*	Atlantic Ocean	120	123
Leatherback Sea Turtles	*Dermochelys coriacea*	Atlantic Ocean	23	10

Once you complete the exercise, go to the answer key section of this document to double-check your work. Abbreviations may vary, as well as titles, but results should be generally the same.

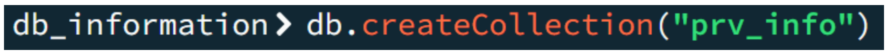
6 Answer Key

DO NOT LOOK BELOW AT THE ANSWERS UNTIL YOU HAVE COMPLETED THE CODE CASE SECTION OF THIS GUIDE!!!

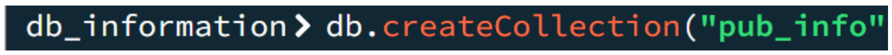
Section 6.1 Answers to Code Case 1

Create a database of your own choosing. The naming scheme doesn't matter for these code cases, but should be something you are comfortable with doing repetitively.

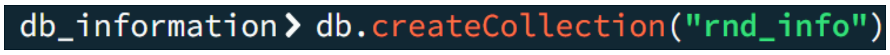
Section 6.1.1 Collection Creation



```
db_information❯ db.createCollection("prv_info")
```



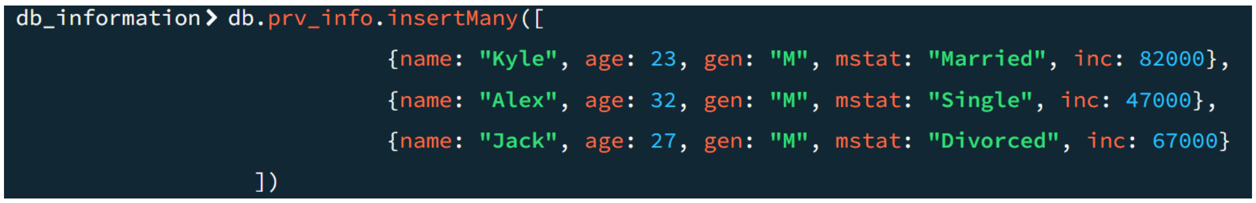
```
db_information❯ db.createCollection("pub_info"
```



```
db_information❯ db.createCollection("rnd_info")
```

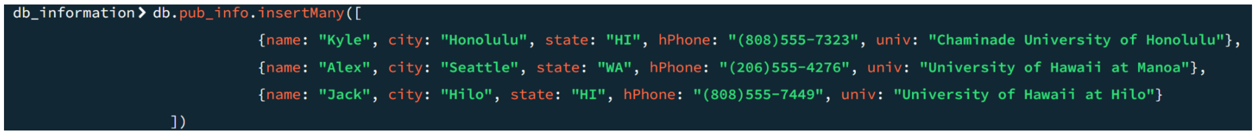
Section 6.1.2 Data Insertion

Collection 1:



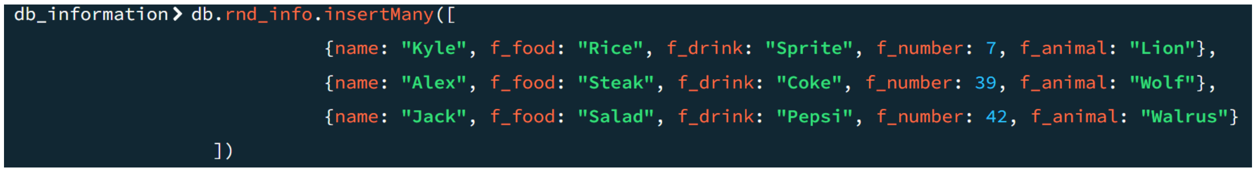
```
db_information❯ db.prv_info.insertMany([
                           {name: "Kyle", age: 23, gen: "M", mstat: "Married", inc: 82000},
                           {name: "Alex", age: 32, gen: "M", mstat: "Single", inc: 47000},
                           {name: "Jack", age: 27, gen: "M", mstat: "Divorced", inc: 67000}
                 ])
```

Collection 2:



```
db_information❯ db.pub_info.insertMany([
                        {name: "Kyle", city: "Honolulu", state: "HI", hPhone: "(808)555-7323", univ: "Chaminade University of Honolulu"},
                        {name: "Alex", city: "Seattle", state: "WA", hPhone: "(206)555-4276", univ: "University of Hawaii at Manoa"},
                        {name: "Jack", city: "Hilo", state: "HI", hPhone: "(808)555-7449", univ: "University of Hawaii at Hilo"}
                ])
```

Collection 3:



```
db_information❯ db.rnd_info.insertMany([
                        {name: "Kyle", f_food: "Rice", f_drink: "Sprite", f_number: 7, f_animal: "Lion"},
                        {name: "Alex", f_food: "Steak", f_drink: "Coke", f_number: 39, f_animal: "Wolf"},
                        {name: "Jack", f_food: "Salad", f_drink: "Pepsi", f_number: 42, f_animal: "Walrus"}
                ])
```

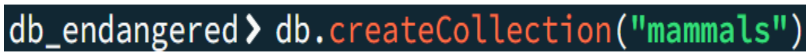
Section 6.1.3 Querying

As long as you get the queries to work, that is good enough. If you are having difficulties, just refer to the previous section on how to do a few basic queries.

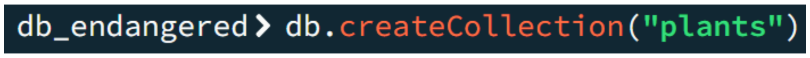
Section 6.2 Answers to Code Case 2

Create a database of your own choosing. The naming scheme doesn't matter for these code cases, but should be something you are comfortable with doing repetitively.

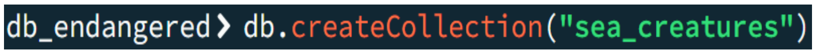
Section 6.1.1 Collection Creation



```
db_endangered❯ db.createCollection("mammals")
```



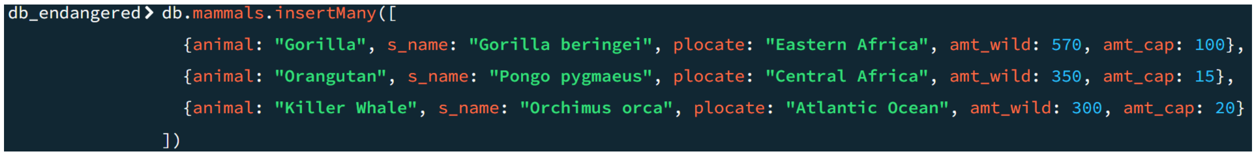
```
db_endangered❯ db.createCollection("plants")
```



```
db_endangered❯ db.createCollection("sea_creatures")
```

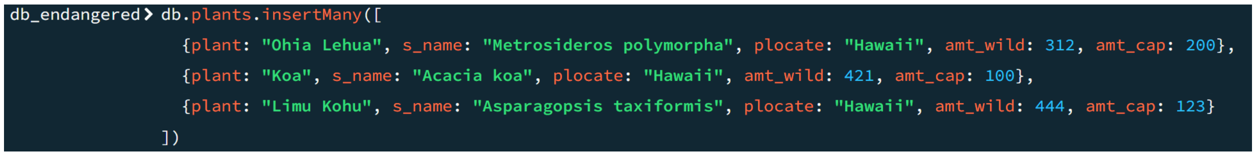
Section 6.1.2 Data Insertion

Collection 1:



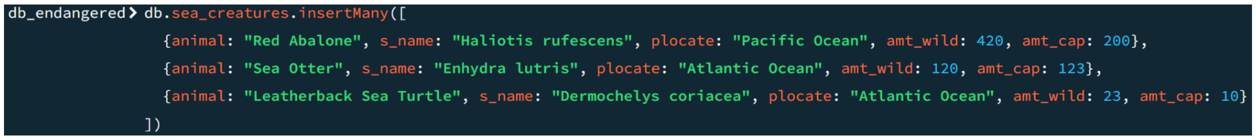
```
db_endangered❯ db.mammals.insertMany([
        {animal: "Gorilla", s_name: "Gorilla beringei", plocate: "Eastern Africa", amt_wild: 570, amt_cap: 100},
        {animal: "Orangutan", s_name: "Pongo pygmaeus", plocate: "Central Africa", amt_wild: 350, amt_cap: 15},
        {animal: "Killer Whale", s_name: "Orchimus orca", plocate: "Atlantic Ocean", amt_wild: 300, amt_cap: 20}
    ])
```

Collection 2:



```
db_endangered❯ db.plants.insertMany([
        {plant: "Ohia Lehua", s_name: "Metrosideros polymorpha", plocate: "Hawaii", amt_wild: 312, amt_cap: 200},
        {plant: "Koa", s_name: "Acacia koa", plocate: "Hawaii", amt_wild: 421, amt_cap: 100},
        {plant: "Limu Kohu", s_name: "Asparagopsis taxiformis", plocate: "Hawaii", amt_wild: 444, amt_cap: 123}
    ])
```

Collection 3:



```
db_endangered❯ db.sea_creatures.insertMany([
            {animal: "Red Abalone", s_name: "Haliotis rufescens", plocate: "Pacific Ocean", amt_wild: 420, amt_cap: 200},
            {animal: "Sea Otter", s_name: "Enhydra lutris", plocate: "Atlantic Ocean", amt_wild: 120, amt_cap: 123},
            {animal: "Leatherback Sea Turtle", s_name: "Dermochelys coriacea", plocate: "Atlantic Ocean", amt_wild: 23, amt_cap: 10}
        ])
```

Section 6.2.3 Querying

As long as you get the queries to work, that is good enough. If you are having difficulties, just refer to the previous section on how to do a few basic queries.

7 Summary

Now that you have completed this guide, you are proficient at the most basic functions of *MongoDB*. This paper detailed a short but comprehensive guide to installing and using basic features of MongoDB for both Mac and PC users. A tutorial on some of the basic uses of *MongoDB* and two code cases were used for demonstrative purposes and to familiarize new users with terms and features they have access to.

References

Comparing the Differences - MongoDB vs MySQL. MongoDB. (n.d.). https://www.mongodb.com/compare/mongodb-mysql

Gillis, A. S., & Botelho, B. (2023, March 7). *What is mongodb? features and how it works – techtarget definition*. Data Management. https://www.techtarget.com/searchdatamanagement/definition/MongoDB

JSON and BSON. MongoDB. (n.d.-b). https://www.mongodb.com/json-and-bson

What is a Database Schema?. SolarWinds. (n.d.). https://www.solarwinds.com/resources/it-glossary/database-schema

What is NoSQL?. MongoDB. (2023). https://www.mongodb.com/nosql-explained