**CRUD I: FINDING DOCUMENTS**

**Browsing and Selecting Collections**

MongoDB is one of the easiest databases to get started with! MongoDB can easily be run in a terminal using the [MongoDB Shell](https://www.mongodb.com/docs/mongodb-shell/?utm_campaign=academia_partners&utm_source=codecademy&utm_medium=referral) (mongosh for short). Throughout this course, we will be providing you with your very own mongosh shell via a terminal. Now - before we can get into making fancy queries on our data, one of the first things we will have to do is navigate around our database instances. MongoDB allows us to store multiple databases inside of a single running instance.

For example, imagine we are a freelance developer using MongoDB to manage the data for multiple different projects: an e-commerce shop, a social media application, and a portfolio website. To compartmentalize our data, we can create a separate database for each project.

With all these databases in our MongoDB instance, how exactly would we choose and navigate around them? Fortunately, MongoDB offers us some handy commands to easily see a list of all our databases, switch databases, and confirm which database we are currently using.

First, let’s list all of our existing databases for our freelance projects. To see all of our databases, we can run the command show dbs. This will output a list of all the databases in our current instance and the disk space each takes up. Here is what it might look like:

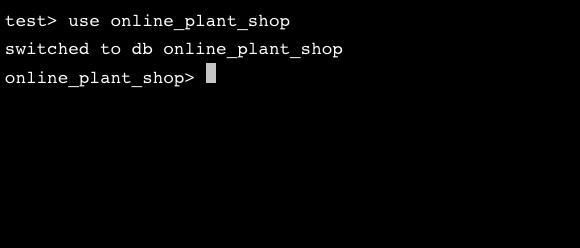
online\_plant\_shop       73.7 KiB  
plant\_lovers\_meet       55.7 MiB  
my\_portfolio\_site       9.57 MiB  
admin                    340 KiB  
local                   1.37 GiB  
config                 12.00 KiB

Looking at the example output above, notice three unique databases: admin, [config](https://www.mongodb.com/docs/manual/reference/config-database/?utm_campaign=academia_partners&utm_source=codecademy&utm_medium=referral), and [local](https://www.mongodb.com/docs/manual/reference/local-database/?utm_campaign=academia_partners&utm_source=codecademy&utm_medium=referral). These databases are included by MongoDB to help configure our instance. In addition, we have our three databases for each of our freelance projects.

*Note: We won’t be working with the admin, config, and local databases throughout this course, but feel free to explore them on your own!*

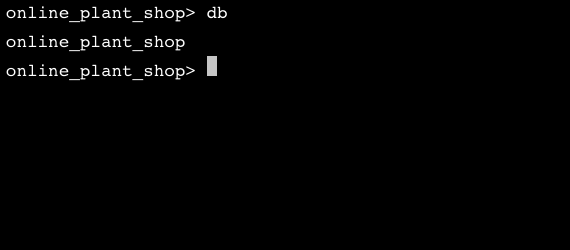
Now that we have a full list of our databases in our MongoDB instance, we will need to choose the specific one we want to work with. To navigate to a particular database, we can run the use <db> command. For example, if we wanted to use our e-commerce database, we’d run use online\_plant\_shop. This would place us inside our online\_plant\_shop database, where we have the option to view and manage all of its collections. It’s important to note, that if the database we specify does not exist, MongoDB will create it, and place us inside of that database.

Here is what our terminal might look like:



Notice that the terminal will list the current database we are in before a > symbol. When we switch databases, we should see the name of the database we switched into displayed there instead. In this case, we can see the prompt changed from test> to online\_plant\_shop>.

If at any point we lose track of what database we are in, we can orient ourselves by running the command, db. This will output the name of the database we are currently using. It would look like this:



Now that we have covered the basics, let’s practice navigating a MongoDB instance!

**Instructions**

**1.**

Let’s get familiar with our environment and orient ourselves by seeing what databases currently exist in our database instance.

Use the appropriate MongoDB command to see a list of all the current databases.

To check your commands for each task, use the **Check Work** button.

*Note: Every exercise in this course will have the MongoDB Shell connected to a test database when it first loads.*

Checkpoint 2 Passed

Hint

You can use the show dbs command to see a list of all your databases.

**2.**

Throughout this course, we will be working closely with the restaurants database. Navigate to the restaurants database in your MongoDB Shell.

Checkpoint 3 Passed

Hint

You can use the use <db> command to navigate to a specific database.

**3.**

Before moving on, use the appropriate MongoDB command to confirm that you are in the restaurants database.

Checkpoint 4 Passed

Hint

You can use the db command to see which database you are actively using. The output should return restaurants.



**Introduction to Querying**

In the world of databases, [persistence](https://en.wikipedia.org/wiki/Persistent_data) describes a database’s ability to store data that is stable and enduring. There are four essential functions that a persistent database must be able to perform: create new data entries, and read, update and delete existing entries. We can summarize these four operations with the acronym CRUD.

In this lesson, we’ll focus on the R in CRUD, reading data. So - how exactly do we start to read data from our MongoDB database? Well, in order to read data, we must first **query** the database. Querying is the process by which we request data from the database. The most common way to query data in MongoDB is to use the [.find()](https://www.mongodb.com/docs/manual/reference/method/db.collection.find/?utm_campaign=academia_partners&utm_source=codecademy&utm_medium=referral) method. Let’s take a look at the syntax:

db.<collection>.find()

Notice the .find() method must be called on a specific collection. When we call .find() without arguments, it will match all of the documents in the specified collection. If our query is successful, MongoDB will return a [**cursor**](https://www.mongodb.com/docs/v5.3/tutorial/iterate-a-cursor/), an object that points to the documents matched by our query. Because our queries could potentially match large numbers of documents, MongoDB uses cursors to return our results in batches.

In other words, when we query collections using the .find() method, MongoDB will return up to the first set of matching documents. If we want to see the next batch of documents, we use the it keyword (short for iterate).

Now, let’s practice using the .find() method!

### Instructions

**1.**

Inside the restaurants database, there is a collection called listingsAndReviews.

Connect to the restaurants database, and then query the listingsAndReviews collection to get familiar with the documents it stores.

After running the command, be sure to hit the **Check Work** button!

Checkpoint 2 Passed

Hint

Remember, you can query a collection using the following syntax:

db.<collection>.find();

To see a list of all collections in a database, you can type the command show collections.

Need another hint?

Your query should look as follows:

db.listingsAndReviews.find()

**2.**

The cursor only returned the first batch of documents. Iterate through the cursor to see the next batch of documents.

Checkpoint 3 Passed

Hint

You can use the it command to iterate through your cursor.

