**1. Download and install Mongodb**

<https://www.mongodb.com/download-center?_ga=2.124132303.491637010.1526431934-983194859.1526431933#enterprise>

Select Windows x64

Click Download

In the dialog box select MSI installer

This will copy a file to your local machine named : mongodb-win32-x86\_64-enterprise-windows-64-3.6.4-signed.msi

Double click this file to launch the installer

Select all defaults. Installation level choose "Complete" installation

Installer will run and complete.

Dialog may come up for Mongo Compass, click through that for now.

More info:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-enterprise-on-windows/>

**2. Start & Run Mongodb Enterprise**

open a windows command prompt

run cmd.exe from the Start menu - type into the text box

In the command shell...

Create data directory for mongodbmd - this creates a directory at c:\data\db

md \data\db

Run the MongoDb Enterprise in command shell

"C:\Program Files\MongoDB\Server\3.6\bin\mongod.exe"

look for message "[initandlisten] waiting for connections on port 27017"

**3. Connect to Mongo DB and create a document collection**

Open up a second command shell

run the Mongo Shell...

"C:\Program Files\MongoDB\Server\3.6\bin\mongo.exe"

display the current collection

db

Create new connection, or switch to existing collection

use customers

**4. Insert Data into the new customers collection**

insert a record into the customers database

db.customers.insertOne( { "name" : "Eric Conway", "email" : "conway.eric@gmail.com" })

Insert another:

db.customers.insertOne( { "name" : "Crissy Conway", "email" : "crisconway@gmail.com" })

retrieve all records in the customers table

db.customers.find()

insert a few more customers. Notice that one also includes a phone number....

db.customers.insertMany([ {"name" : "Kara Conway", "email" : "kbc917@gmail.com"}, {"name" : "Jessica Conway", "email" : "jlconway113@gmail.com", "cell" : "440-769-7757"} ])

**5. Run queries to pull data out of the document collection**

Select all records from the customer collection....

db.customers.find()

Select all records from the customer collection matching a given email...

db.customers.find( { email : "conway.eric@gmail.com" } )

Create an index to allow one to search for strings within the email field

db.customers.createIndex( { email: "text" } )

Now that index is created you can search the indexed email field for a given search string...

db.customers.find( { $text: { $search: "gmail" } } )

**6. Investigate command shell commands...**

type the following, then hit the Tab key....

db.customers. (hit TAB key)

This will list all the possible commands. Try some out....

db.customers.count() - count records in the collection

db.customers.deleteOne( {name: "Eric Conway"}) - delete a record

**7. Explore MongoDB Compass GUI (Graphical User Interface)**

Open MongoDB Compass from the Start menu

In the Connect to Host form: Hostname=localhost, Port=27017. Hit CONNECT.

Click on the Collections named Customers

Click on the name: Customers and you should see a list of the records you inserted into the collection.

Under VIEW switch between LIST (JSON view) and TABLE (Table view).

Use this gui for all CRUD functionality (Create, Retrieve, Update, Delete)

CREATE: Insert a record y clicking the Insert Document button. Enter Key:Value pairs for name, email, cell. You can even add a new field like 'home phone' and give it a value. Click insert and you'll see it show up in the List or Table view.

RETRIEVE: Run a query. In the Filter box enter a search string like {name: "Kara Conway"} and click Find.

UPDATE: Edit any value in Table view by clicking in the cell, modifying, and click Update button.

DELETE: Delete any row from Table view by selecting and clicking on the Trashcan icon. Click Delete button.

Create a new collection in the My Cluster panel on the left. Hover over an existing collection name (hover over customers), look for the + button and click to create new collection. I created a collection called 'accounts' to represent bank accounts. It has a type (e.g. checking or savings) and a balance, and a customer field. The customer field would be the \_id from the customer collection. Example record I inserted into accounts:

type : "checking",  
balance : "1250.00",  
customer : "5afb8bc8e0ad819dbe70af5f"

(The customer value is the id for Crissy Conway in my database, thus it links or 'relates' this account record to her customer record).