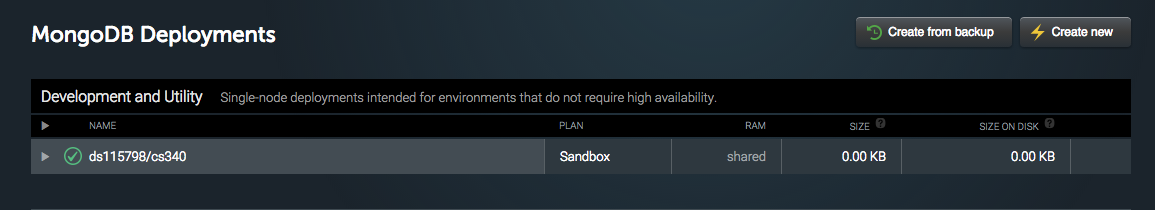
**Set up mLab account**

mLab is a website that offers free hosting for MongoDB databases. We will use their service to set up a simple database.

1. Go to http://www.mlab.com in a separate window.
2. Click the blue button labeled "Sign Up" in the upper right corner.
3. Fill out the form that appears.
   * "Account name" should be globally unique, so pick something likely to be unique, such as your Rhodes username.
   * "Username" can be anything you want, but I'd use your Rhodes username again.
4. After you sign up, you will need to check your email and verify your account.

**Create a MongoDB deployment**

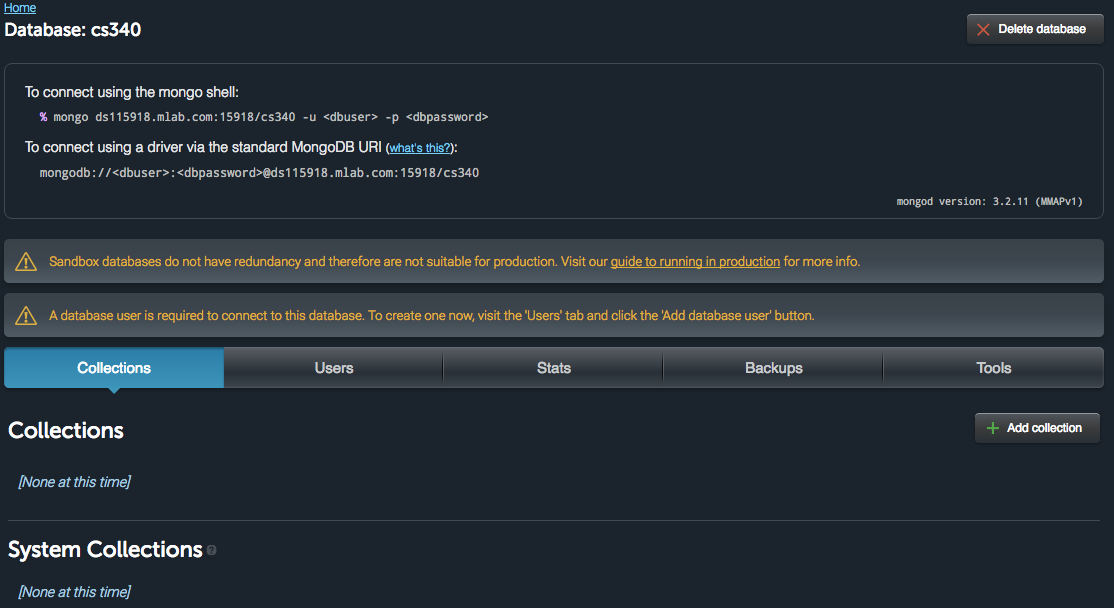
A “deployment” is what mLab calls the combination of a MongoDB database and all the infrastructure to host it.

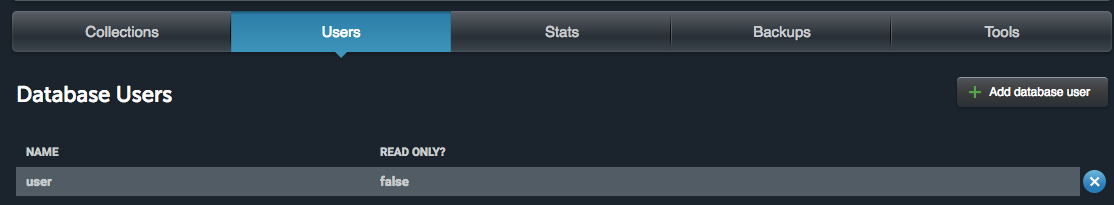
1. From the main mLab webpage, look under “MongoDB Deployements” and choose “Create new.”
2. Leave “Cloud provider” alone (Amazon is fine).
3. Under "Plan," choose "Single-node," then look under “Standard line” to select "Sandbox" (this is the FREE option).
4. Under “Database Name”, give your new database the name "cs340" (or use something else, but this tutorial will use "cs340").
5. Click "Create new MongoDB deployment."
6. You should see the following after a quick wait:  
   

**Create a database user**

Each MongoDB database has its own set of usernames and passwords that are used to connect to the database. This is typical of most large-scale database systems, including Oracle, Microsoft SQL Server, PostgreSQL, and MySQL. SQLite is atypical in that it doesn’t have usernames and passwords.

The usernames and passwords used to connect to a Mongo database are different than the ones used to connect to the mLab website. Each user is only unique within a database, so we can safely use a simple user name and password without worrying about security. We will use “user” and “pass” as our credentials.

1. Click on the cs340 database that you just created. Click on the green checkmark or the name of the database or anywhere in the gray box area other than the gray right-pointing triangle. You will be taken to a new webpage as seen here:  
   
2. Click on the “Users” tab next to “Collections.”
3. Click “Add database user.”
4. Set the username to “user” and the password to “pass”.
5. Click “Create.”
6. You should see this:



**Download the MongoChef client**

1. Like SQLite can be interacted with through a text-based command line interface or through a graphical interface, so can MongoDB. We will use a client called MongoChef.
2. Download it here: <http://3t.io/mongochef/download/core/platform/>
3. Install MongoChef and open it.
4. Click the big screen “Connect” button: 
5. In the Connection Manager, click “New Connection.”
6. You will need the following information from the mLab website. Find the text on the mLab website that looks like this:

To connect using the mongo shell:

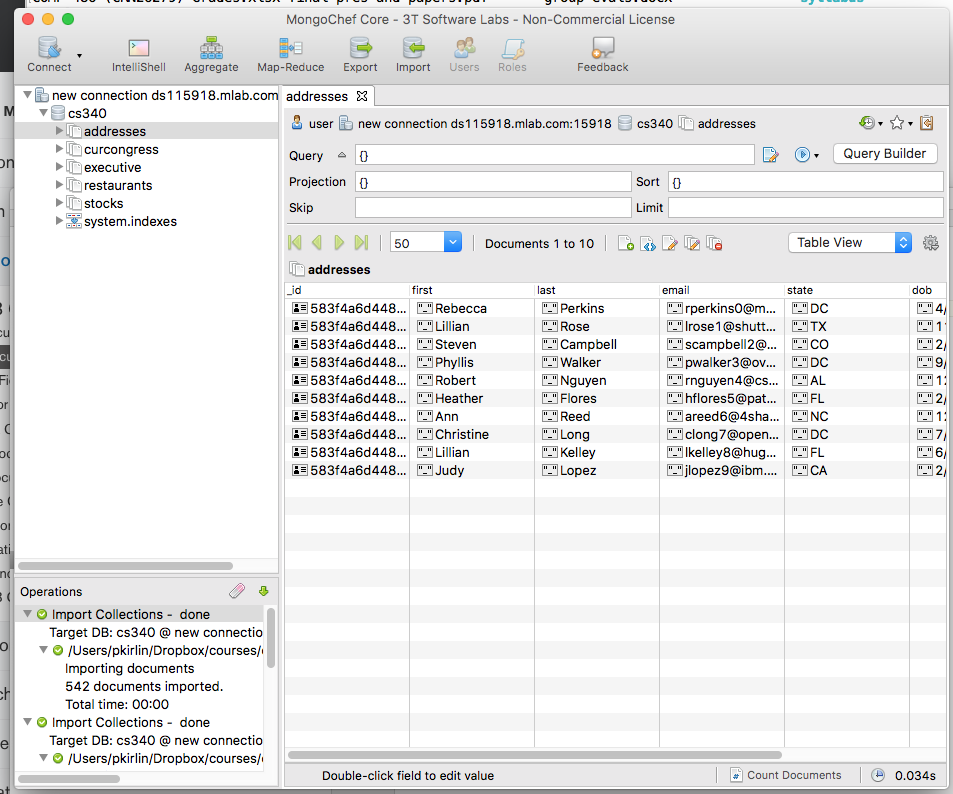
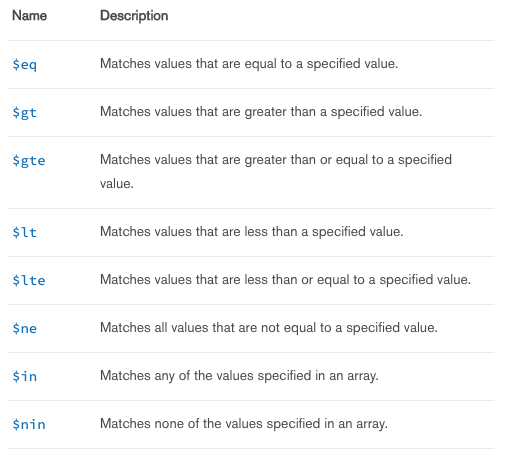
mongo ds115918.mlab.com:15918/cs340 -u <dbuser> -p <dbpassword>

**Load data into database**

We will create a small MongoDB collection of documents containing demographic information for ten people.

1. Click on the cs340 database that you just created (click on the name of it or in the gray area). You will be taken to a new webpage.
2. Under “Collections,” you will see “None at this time.” Click “Add collection” on the right side of the page. Name the collection “addresses.”
3. Click on the “Users” tab which is to the right of “Collections.” Click “Add database user.” I recommend using the same username and password as when you signed up for the site. (Or just use “user” and “password” --- these are unique to your individual database, so you can pick insecure things.)

**Familiarize yourself with the addresses collection**

1. To search a mongo collection, you use a JSON document.
2. Double-click on the “addresses” collection under “cs340” and you should see this screen:  
     
   
3. Under “Query”, put the following:   
   **{"state": "DC"}**
4. Press the blue circle with the triangle inside of it to the right (looks like a “play” button) and the query runs. You should see THREE results.
5. Try switching between Table View, Tree View, and JSON View.
6. By default, MongoDB assumes equality when you write {key: value} in a query. To get another kind of comparison, you can use the following query operators:
7. 
8. Here’s how they work:
   1. Find all documents where the children field is greater than 1:  
      **{**"children"**:** **{**$gt**:** 1**}}**
   2. Find all people who don’t live in DC:  
      **{**"state"**:** **{**$ne**:** "DC"**}}**
   3. Find all people who live in a certain set of states:  
      **{**"state"**:** **{**$in**:** **[**"DC"**,** "TX"**,** "NC"**]}}**
9. Now we will learn how to use AND and OR in MongoDB.
10. To combine two queries with AND, just use a comma:  
      
    **{**"children"**:** 0**,** "state"**:** "DC"**}**
11. To combine queries with an OR, use the $or operator, which takes an array of documents:  
    **{$or : [ {**"children"**:** 0}**,** {"state"**:** "DC"**} ] }**
12. Here is a query that finds all people with greater than 2 children or live in a certain set of states:  
    **{** $or **:**   
     **[**   
     **{**"children"**:** **{**$gt**:** 1**}},**   
     **{**"state"**:** **{**$in**:** **[**"DC"**,** "TX"**,** "NC"**]}}**   
     **]**   
    **}**

**STOP AND ANSWER LAB QUESTIONS PART A!**

1. You can use “dot notation” to search embedded documents. Take a look at the “executive” collection. This contains JSON information for all US presidents and vice presidents.  
     
   Find all presidents: {"terms.type": "prez"}  
   Find all vice presidents: {"terms.type": "viceprez"}

Find all democratic presidents: {"terms.type": "prez", "terms.party": "Democratic"}

Find all republican VPs: {"terms.type": "viceprez", "terms.party": "Republican"}  
  
Notice how the above are all queries on elements from ARRAYS. When you use the dot notation with arrays, you are asking that AT LEAST ONE element of the array matches the query.

**STOP AND ANSWER LAB QUESTIONS PART B!**