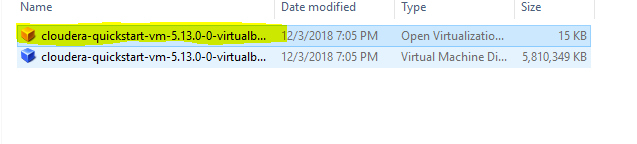
Ryan Sikora

RJ Podeschi

IS470 – Business Intelligence and Big Data

Cloudera Lab Documentation

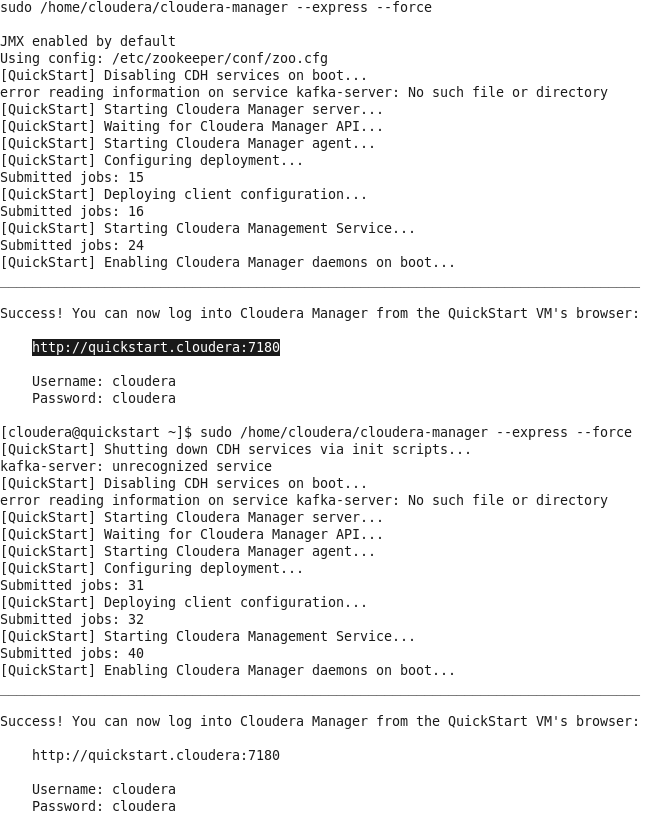
First, I opened Cloudera’s virtualization file. When opening it, the user is prompted to set the specifications of the virtual machine. I changed the RAM from 4GBs to 8 GBs and set the processors from 1 to 2.

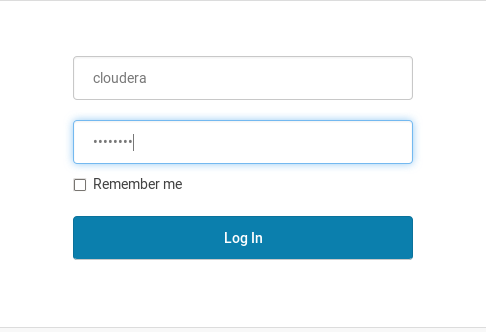


From there, I tried opening the Launch Cloudera Express application to get to the cloudera manager. However, it did not launch right away because Cloudera Express wants to have more than one node (the amount of computers in the Hadoop Cluster). I was able to bypass this inconvenience by opening the terminal and typing the command below:

sudo /home/cloudera/cloudera-manager --express --force

into the terminal, and from there I was given a link, username, and password to login Cloudera Manager. One of the problems I ran into while trying to run this command is that Cloudera Express does not give you the full command, so when I first tried running the code, it asked me to specify between Cloudera Manager Express and Couldera Manager Enterprise. From there I had to retype the code to specify that it was the express edition.





After logging in, I needed to make sure to start all the required applications needed to complete the assignment, they are:

Apache Impala: This is used to query data

Apache Hive: Structures the storage

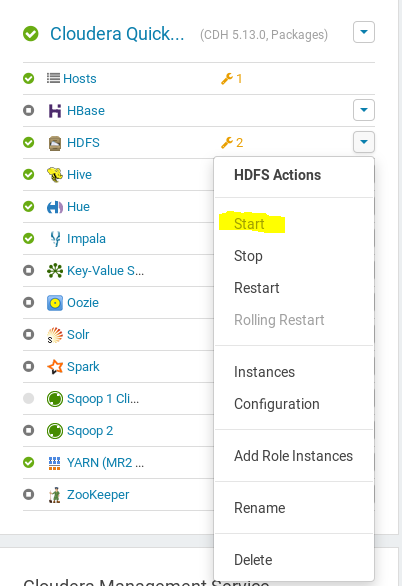
HUE: Allows the end user to input queries

HDFS: The distributed data storage

YARN: The framework used for processing by Hive

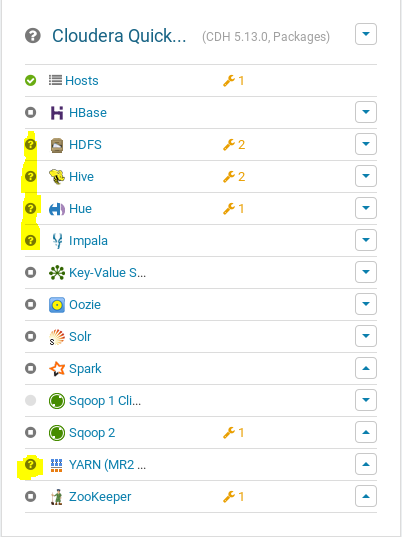
To do this, I clicked on the arrow to the far right, clicked start, and let the applications start running.

Some of the challenges to this were that after you finished starting the application, the status bar would change from green (like it should), to yellow, and then to red, meaning that there are problems with the application. Although this could be stressful to look at, all you needed to do is let the application sit for a little before trying to start the next applications. The status should come back to normal eventually.

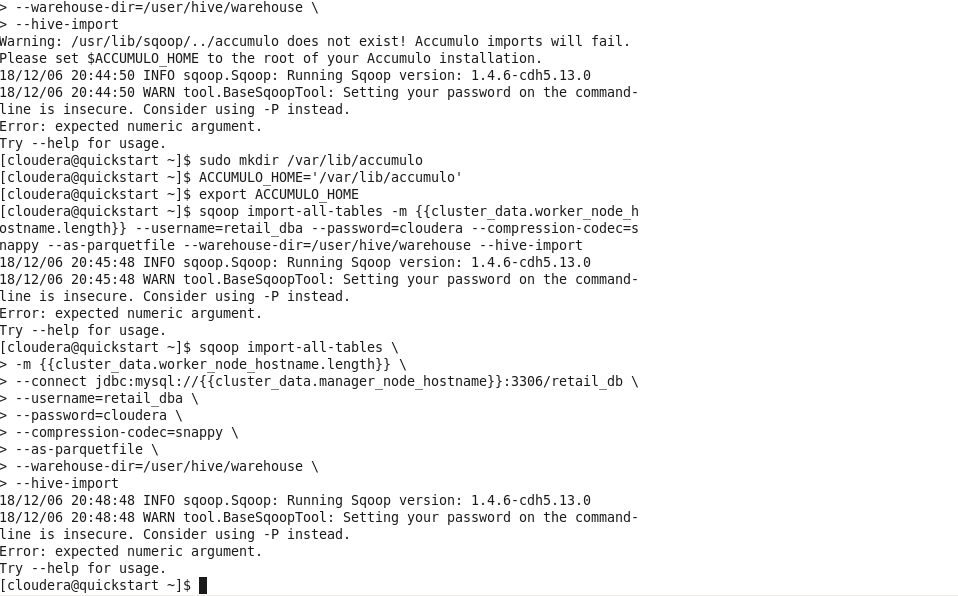


The next step is to take structured data from a relational database and transfer it into the Hadoop Filing System to use for analysis. Apache Sqoop is a tool that allows users to do this. I tried running the Sqoop commands by typing them in as they were given in the tutorial, however, I would get an error saying Accumulo imports would fail. In the Lab Assignment Details, there were a set of commands that install a package that removes this error. At first, when I tried running these commands, my Sqoop command would still not run. Because of this this, I tried restarting my virtual machine to see if that would help.

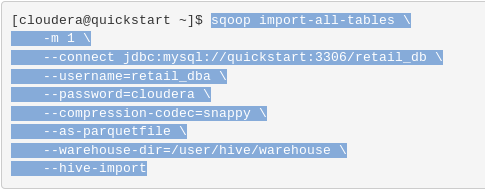
Unfortunately, that did not help, and it also changed all the application statuses from “healthy” to unknown health.

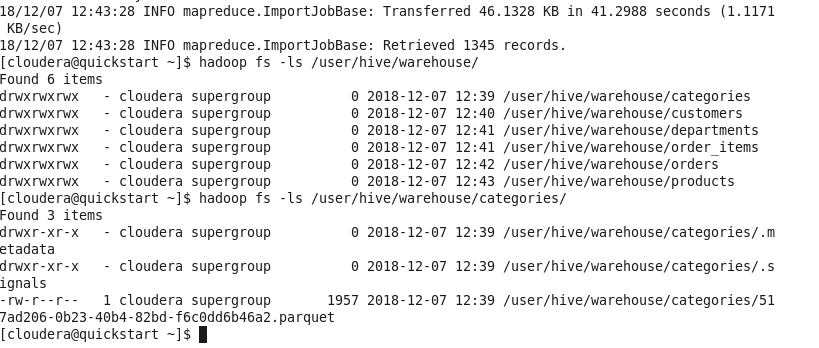


So then, I closed all apps to rest, shutdown the Cloudera virtual machine, went to the virtual machine settings and changed the RAM from 8GBs to 10GBs, reopened the virtual machine, opened all apps again, slowly, tried rerunning the Sqoop and Acumulo commands, with no positive results.

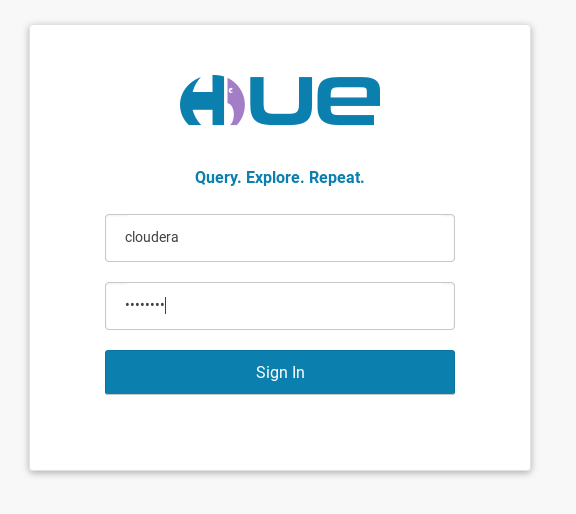


Finally, after going to the tutorial section on the Cloudera homepages, and copying and pasting the Sqoop code into the terminal, the Sqoop command finally ran. The Sqoop call uses MapReduce to pull and sort the data.

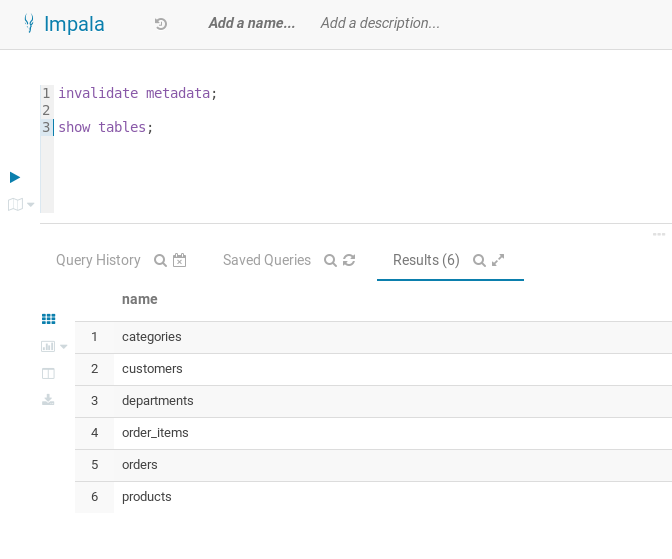




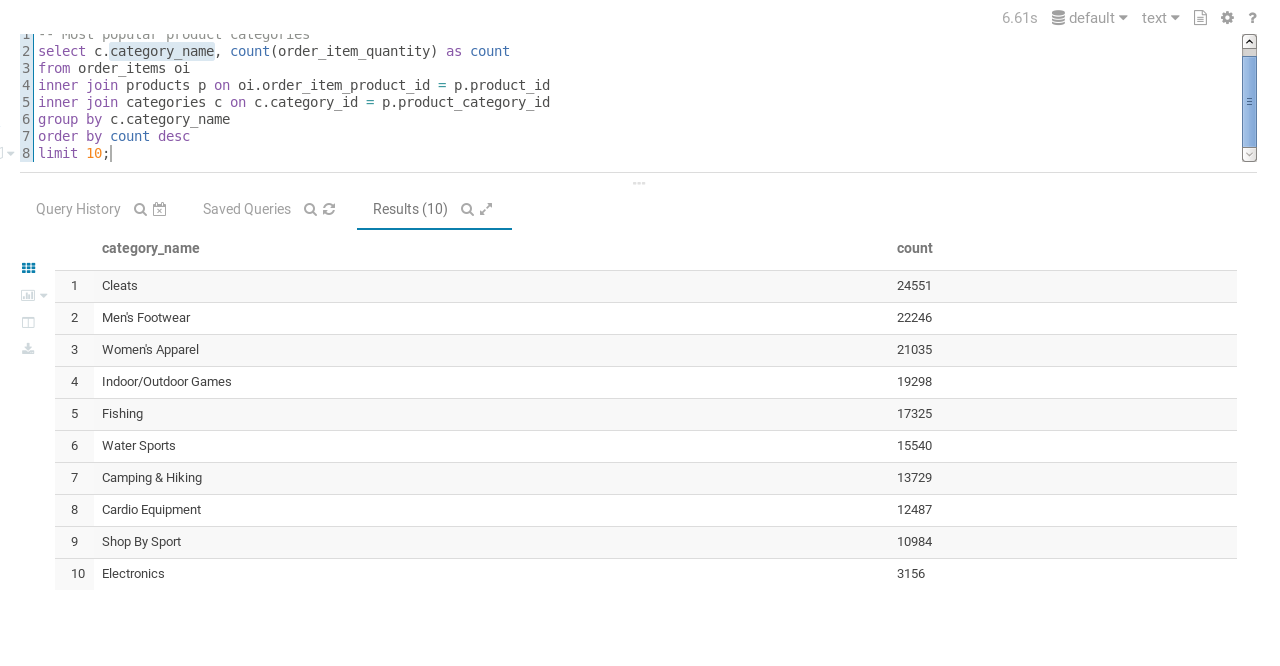
After checking that the data files (with the commands above), I went to the Hue login and used the username: cloudera and password: cloudera

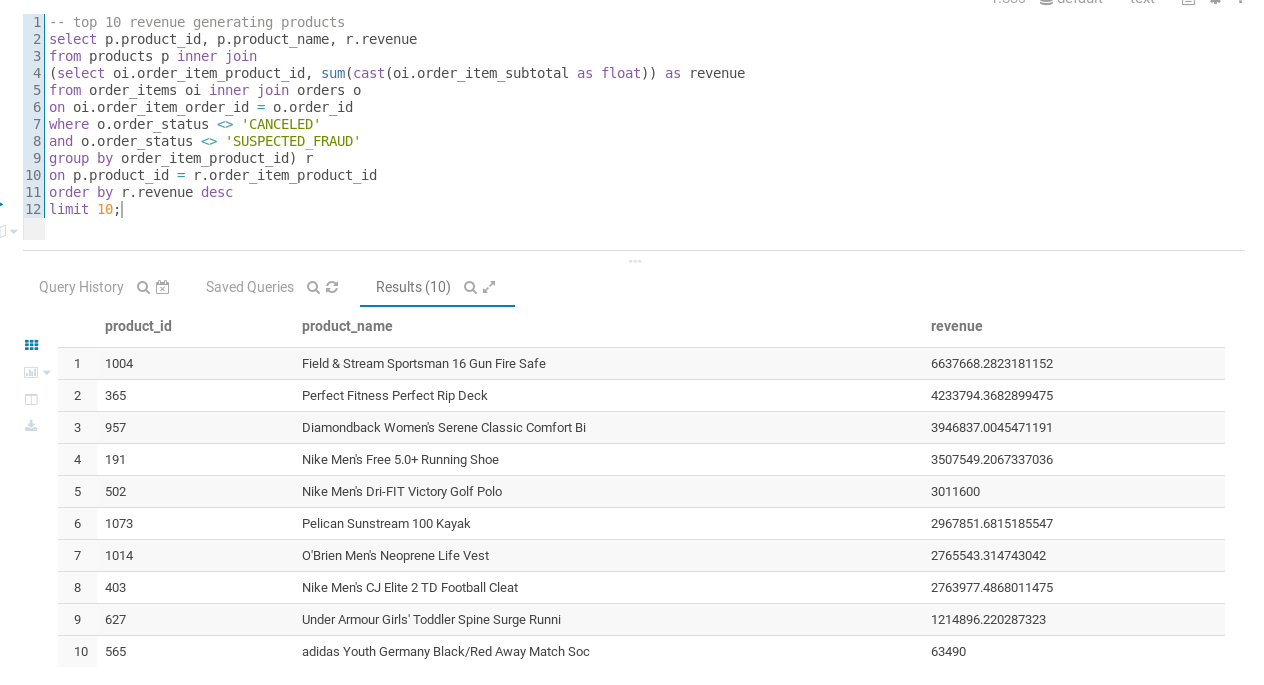


From here, I went to Hue’s Impala app, so I could pull information from the tables we imported in the last steps. First, I used the query “invalidate metadata;” because we need to tell the metadata it is out of date. This will allow us to query the data. To check if the data is still actually there, we use the “show tables;” query. One mistake I had run the show table query before even doing the invalidate metadata query This was because I had my cursor at the end of the show tables query and the semicolon shows a split between two queries. I had to have my cursor at the end of the invalidate metadata query for it to run.



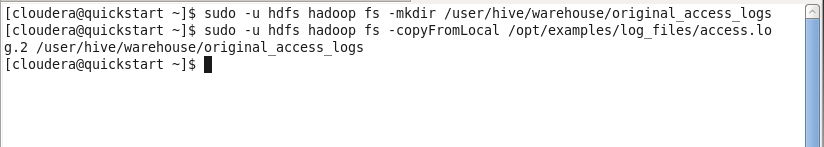
Finally, I copy and pasted the queries provided into Impala so I can find out top 10 most popular product categories and top 10 revenue generating products (as shown below).





It’s important to note the difference between Hive and Impala in these examples. Hive turns SQL queries into MapReduce jobs whereas Impala is used to query the data itself.

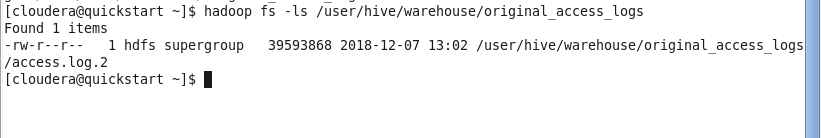
To start exercise 2, we need to bulk upload some pre-loaded data into the Hadoop Filing System. We will use the commands shown below to import the data.



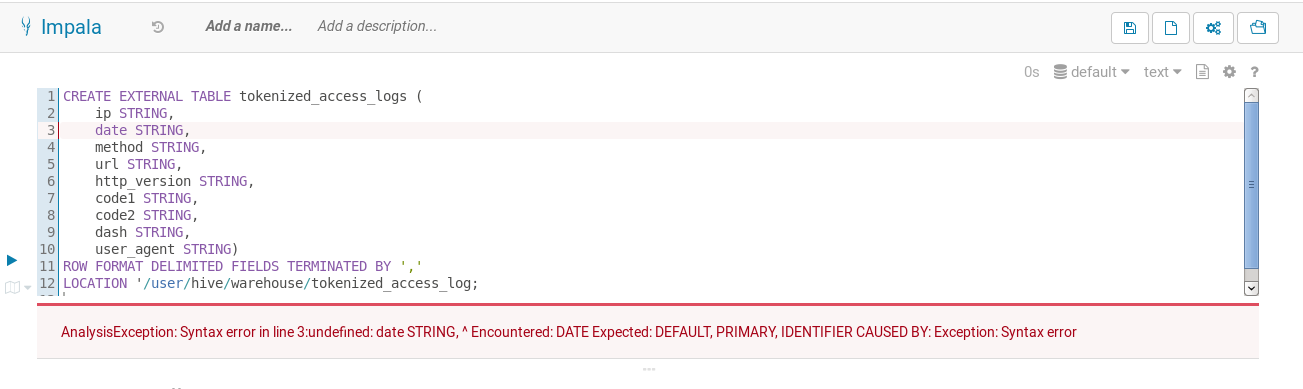
We then use the command

Hadoop fs -ls /user/hive/warehouse/original\_access\_logs

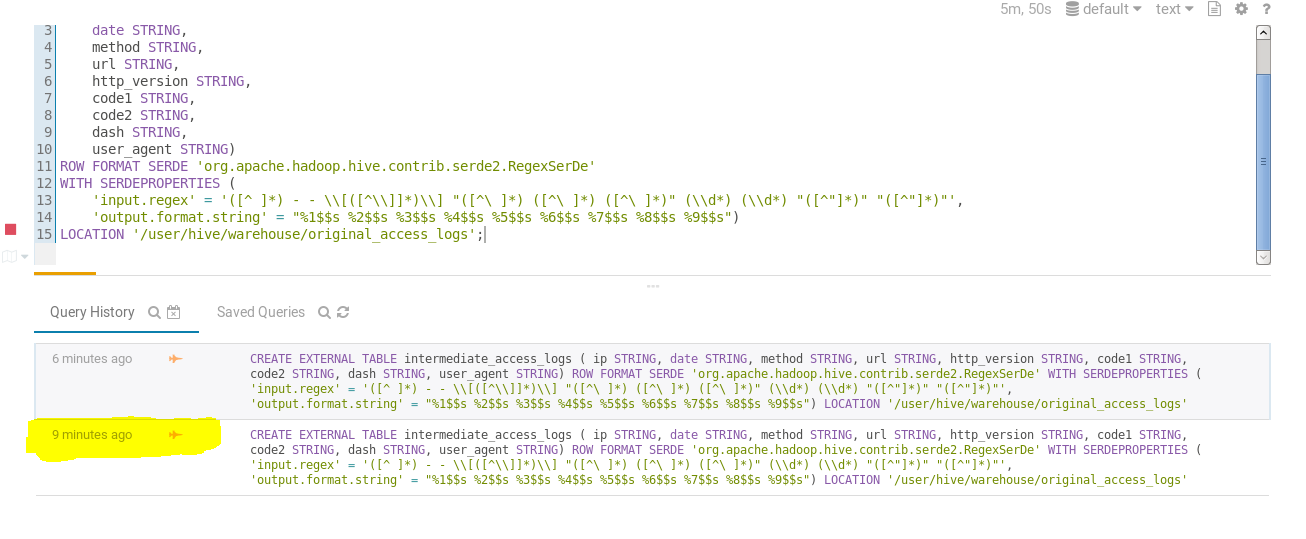
to show that our data is imported.



Now that the data is on Cloudera, we now use Hive to create the tables. Insert the following code into the Hive editor to create the table. One mistake I had made doing this is forgetting to switch from Impala app that I used in the last tutorial to Hive. If you don’t do this, you get errors because Impala is not used to create tables, just querying data.



Once I had gotten into Hive and tried making the tables, the query had just kept running trying to make the tables without much progress. As shown below, even after 9 minutes with the virtual machine having 10 GB of RAM allocated to it, there was no positive progress. Therefore, I was not able to continue with exercise 2.



**Last Step**

1. I think I had learned the most about the different ways to troubleshoot and try to fix things using different terminal commands and messing around with the virtual machine settings. This lab went anything but smoothly, so really being patient with the virtual machine, knowing different commands to safely restart the VM, bypassing the Cloudera Express restrictions, making sure to shut off all applications before restarting, etc. By the end of the lab, I was able to do restart the VM and bypass the Cloudera Express restrictions without looking at the instructions because I had done it so much.
2. I think from what I had done so far it would be cool to do the rest of the exercises, as long as I had more freedom to try writing queries by myself. Doing SQL exercises have always been some of my favorite activities in my IS classes, so if the rest of the tutorials had similar aspects, it would enjoy learning more about the Cloudera system.
3. The tutorial helped me learn the differences between what each application was able to do. For example, I learned that Sqoop is used to pull data into Hadoop, Hive is used for creating tables, and Impala is used to query data. Each application has an important part to play, and when putting them all together, we were able to accomplish our goals.

Ryan, this was a valiant effort and you went through a lot of troubleshooting steps to get as far as you did. Kudos to you for persevering and seeking guidance from others. You did a thorough job of explaining what you went through, the steps included, but also defined the tools you were using and their specific use-cases. This was an important element to the lab exercise that was overlooked by many. You receive full credit for the lab. 35/35