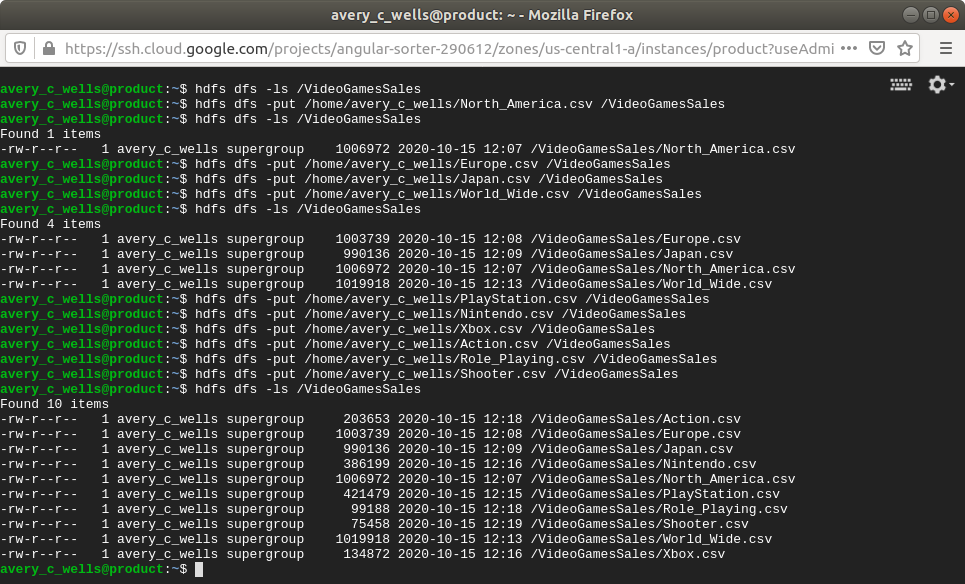
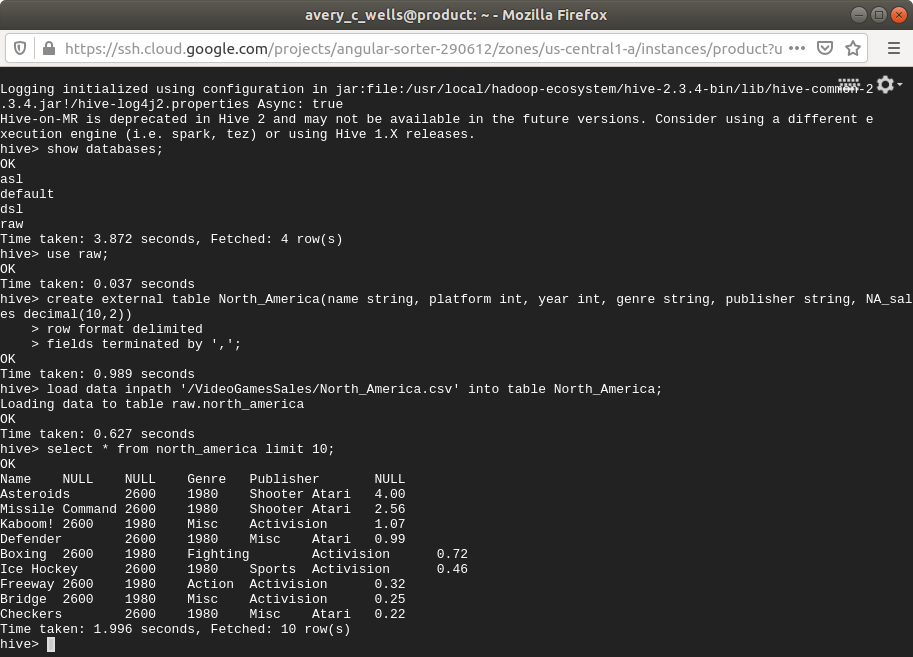
**Hadoop Overview**

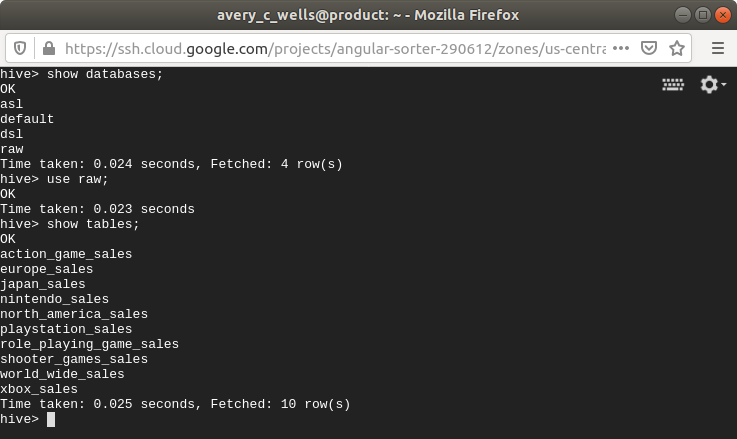
This process was done in the Google Cloud Platform using my production instances. Before getting to the internal and external tables in Hive, I had to place my csv files in HDFS. They are all in one folder “/VideoGamesSales” so it can be easier to look for them.



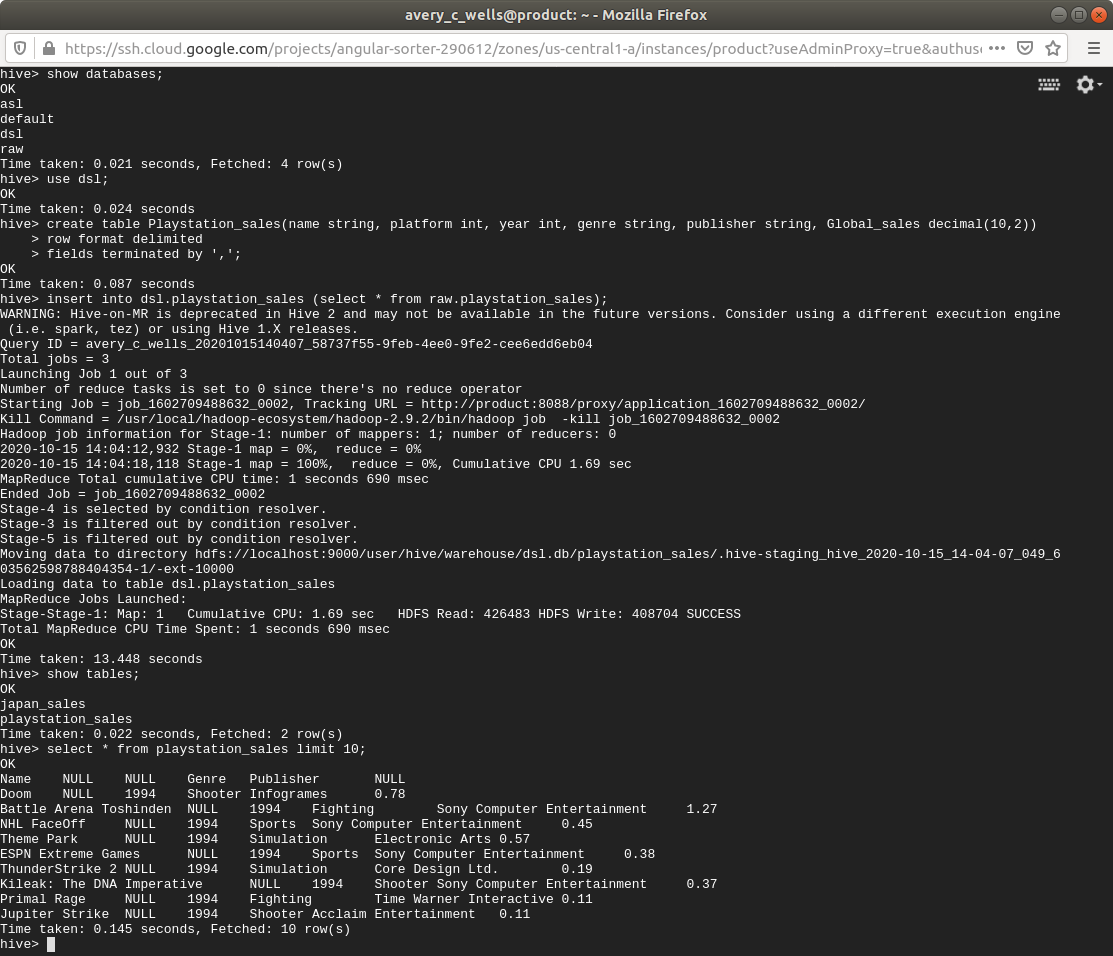
Next for me was to transfer those files/tables into Hive. I had to create an external table in the RAW database in Hive so all the tables can be in one place. Getting the right schema to match the table was important when moving tables to another database. Even though some of the tables had the same schema, some of them had them in different order so it was key to not misplace them.



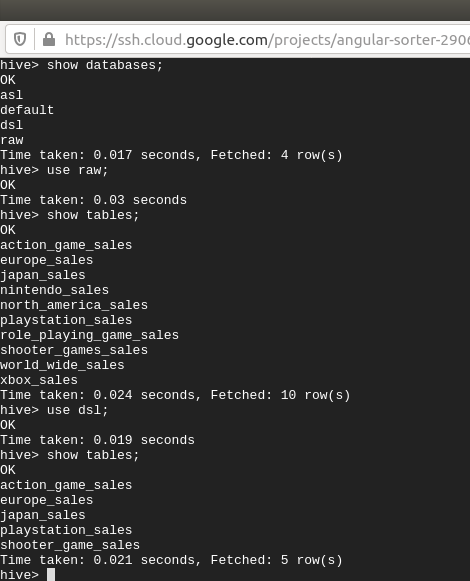
I did rename all my tables so they can be described better.



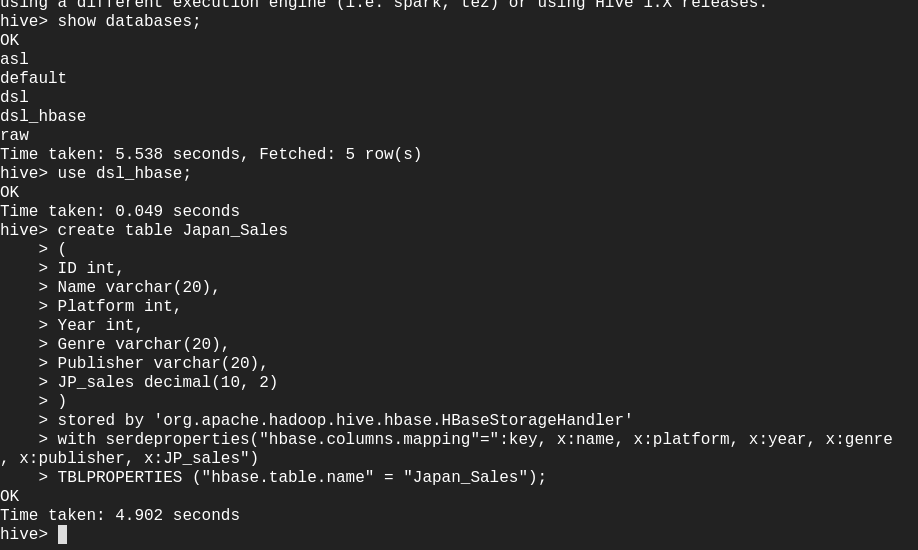
Now it was time for the internal tables. Thee internal tables were to be placed in the DSL database. I only selected 5 of the tables from the RAW database. The command for creating an internal table were just the same as external except that there was no need to type “internal” in the command. Inserting the tables were also different from external. The command would be: “insert into dsl.table (select \* from raw.table);”.

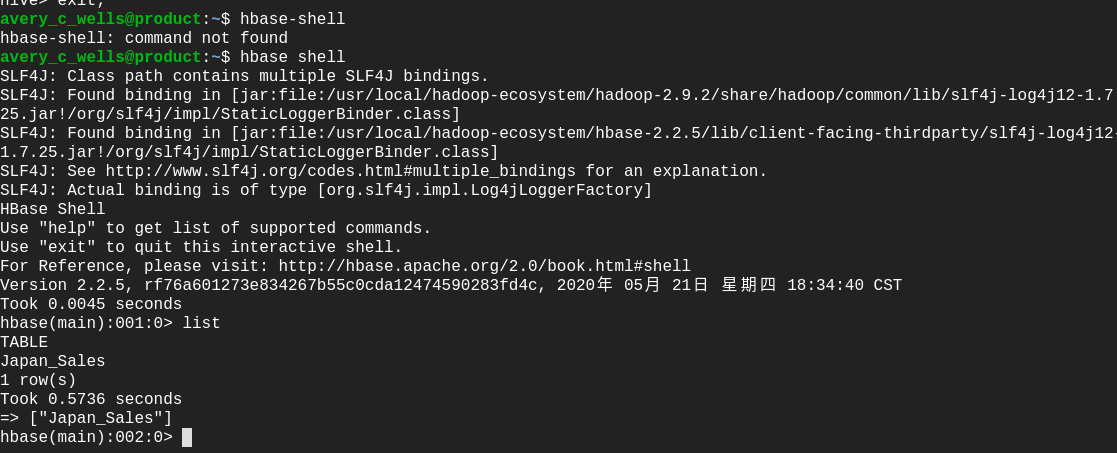


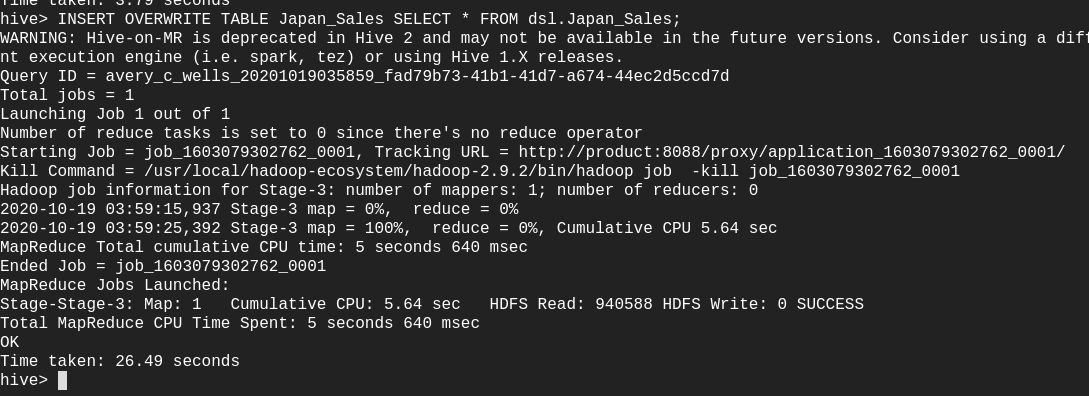
Below, both the RAW and DSL databases are showing their respected tables.

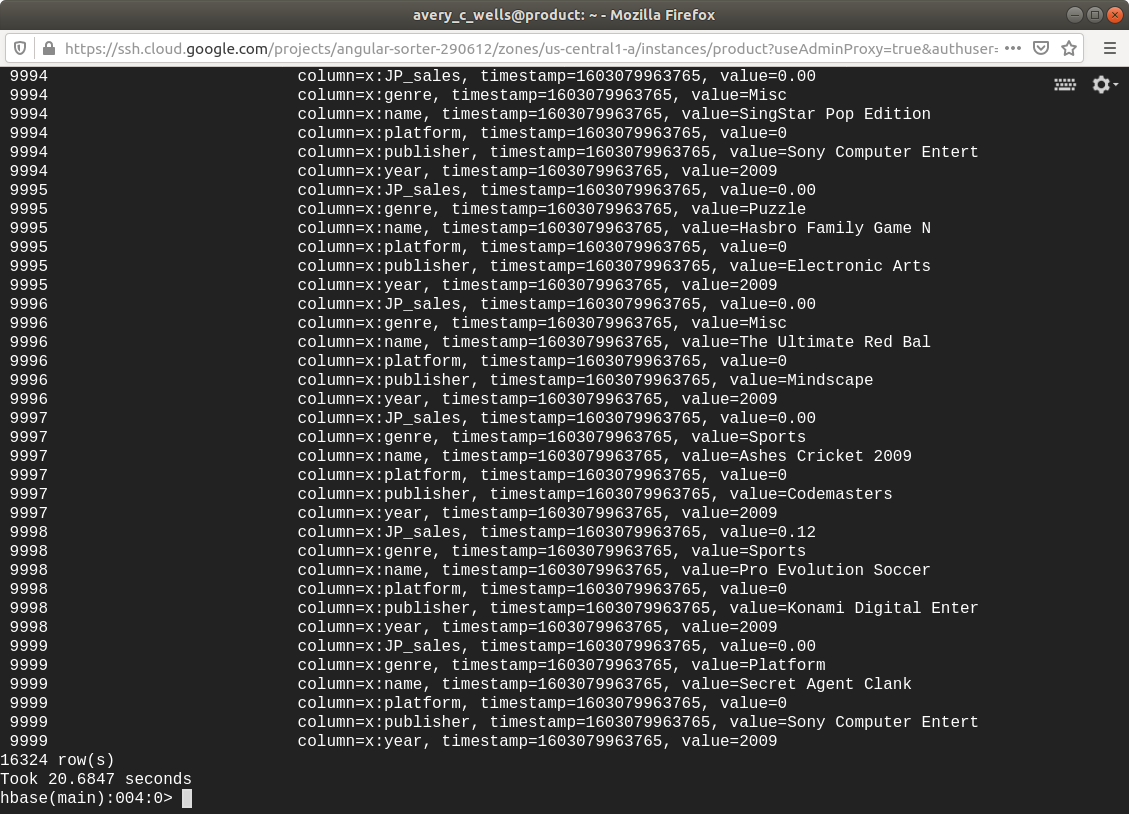


**Update:**

The last task to do was to import my tables from Hive to HBase. I created a new database “dsl\_hbase” so I would not have to use the other databases. The process starts with creating a table like the with external and internal. I added “ID int” for that is my primary key and is also important in order to get this importing done. I also replaced my data type string with “varchar(20)” so the database can read this specific data type. Then it was time to call out the location of hbase ‘org…Handler’ and call out the properties/columns I already have in my previous tables.



The table is now in HBase but the data is not in it yet. I need to enter the Insert Overwrite command in order to let map reduce take its course and import the data in my table.

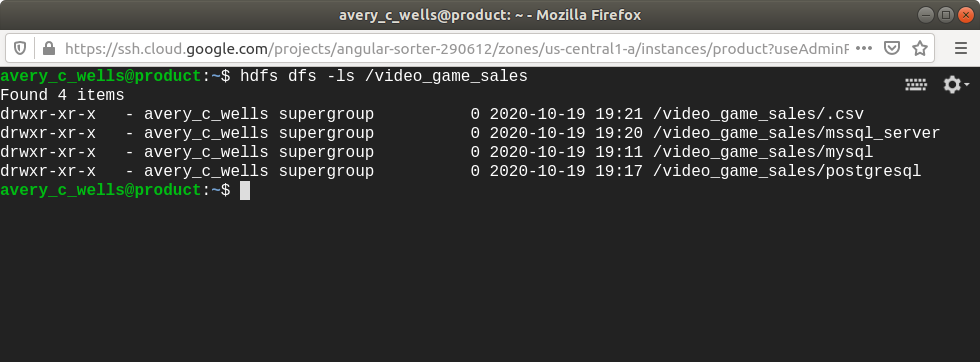


The table display all of the data in a really fast pace and even though there is a lot of data, it still took around 20 seconds to load every row that my table had. Now I can put the rest of my tables in HBase and finish my objective.

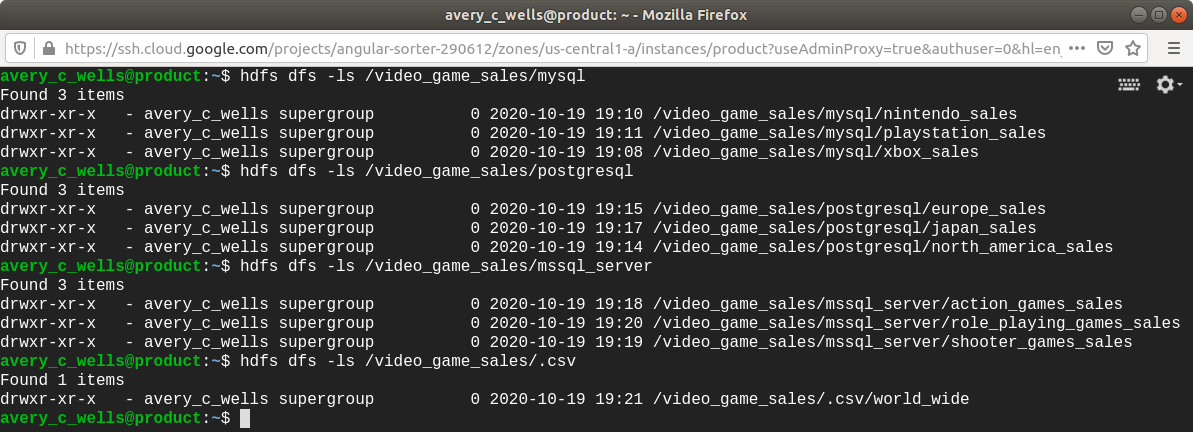
In conclusion, I was able to use the tools/elements of Hadoop to get a regular csv file from one database to another and display its tables in different ways. I can also say that I used Sqoop through 3 RDBMS, HDFS, external/internal tables in Hive, and HBase to complete the objective.

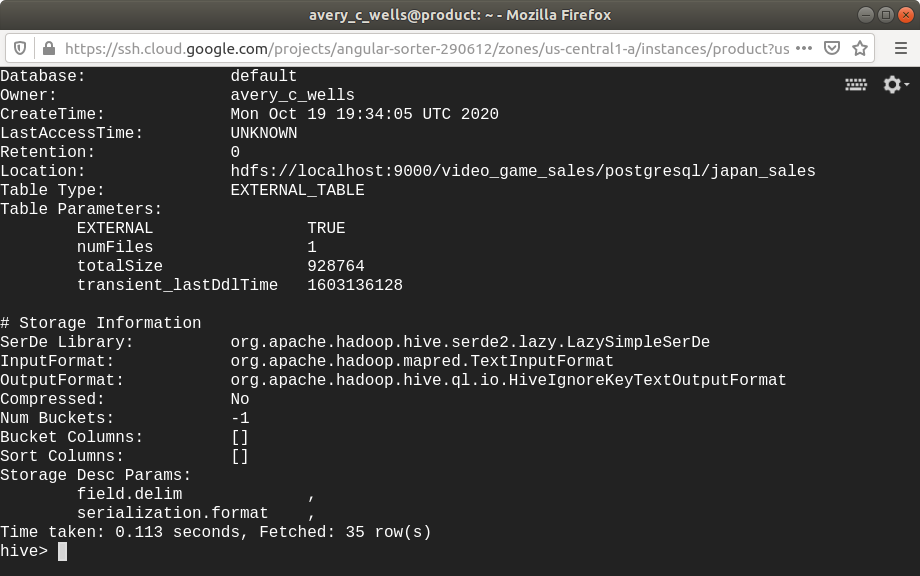
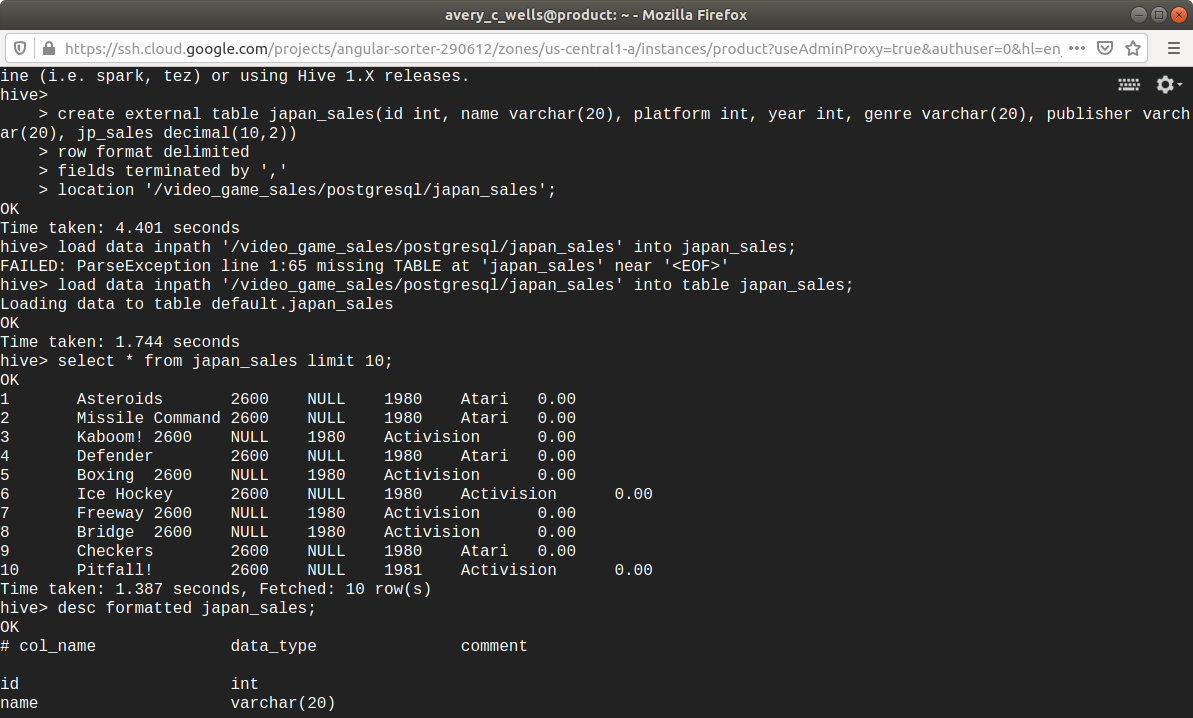
**Update 2:**

I reorganized the directories based on their RDBMs.

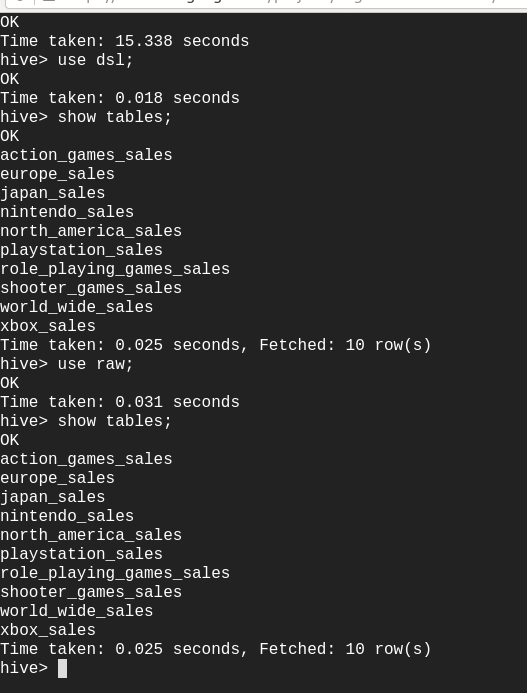


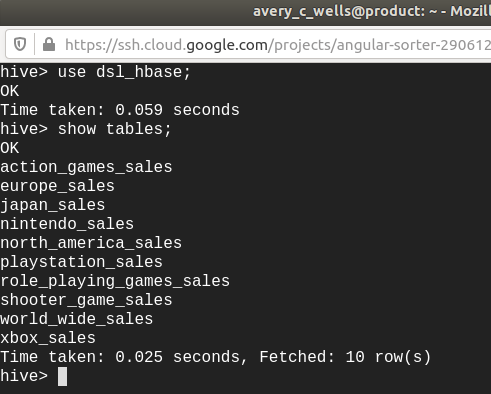
Each table that came from their original RDBM source are now sorted in their directory. There are also 10 tables total now as opposed to 5.





The locations for the tables are now shown and correct so it can follow its original path.



An updated version of raw and dsl database now includes all 10 tables; as well as dsl\_hbase.  
Finally, HBase now has all of the tables displayed.