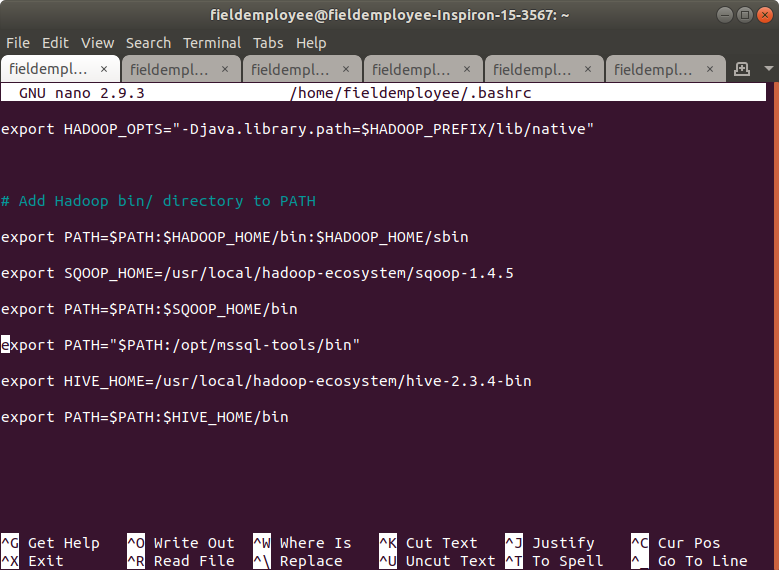
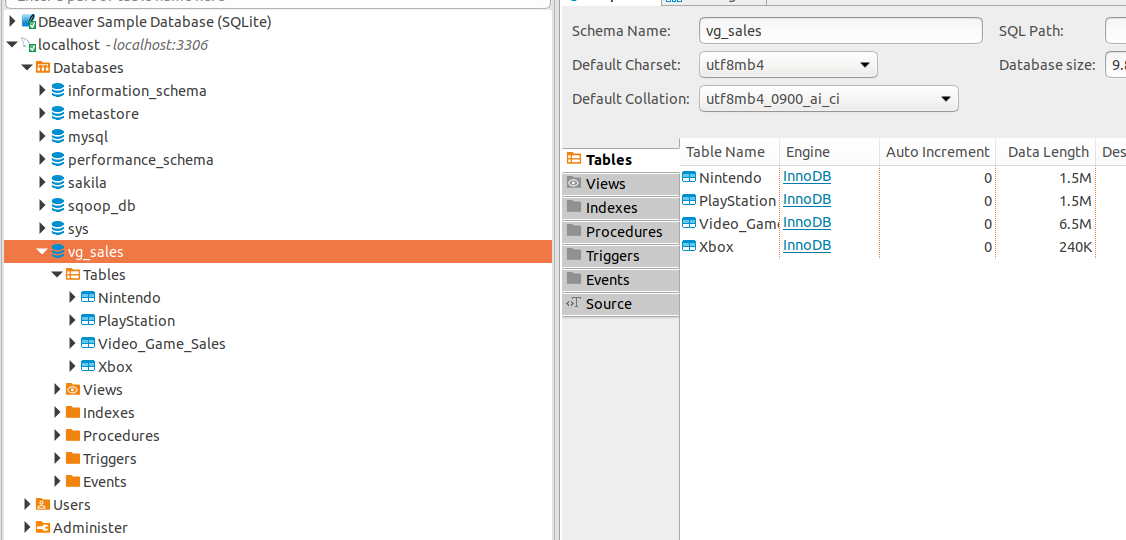
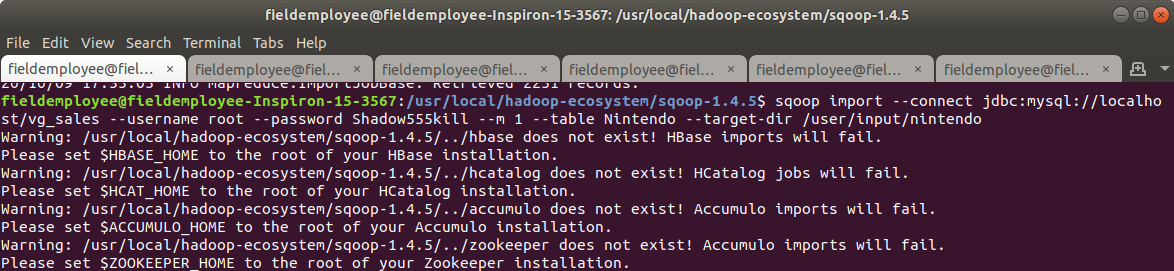
**HDFS Documentation**

What I have done during the past week or two was utilize and experiment with HDFS (Hadoop Distributed File System). I was also able to use Sqoop to connect 3 RDBM (Relational Database Management System) and import different tables with primary key to HDFS. Sqoop was a tool that needed to be installed in Hadoop so data can be simply transferred.





The first RDBM I used Sqoop for was MySQL. I had 3 tables in my database (vg\_sales) that I wanted to transfer into HDFS. The tables: “Nintendo,” “PlayStation,” and “Xbox” are displayed in DBeaver. In the terminal, I entered the Sqoop commands to import the data/tables into HDFS. 

Sqoop import – start the command to import

--connect jdbc:mysql://localhost/vg\_sales – this connects to the database that was in MySQL

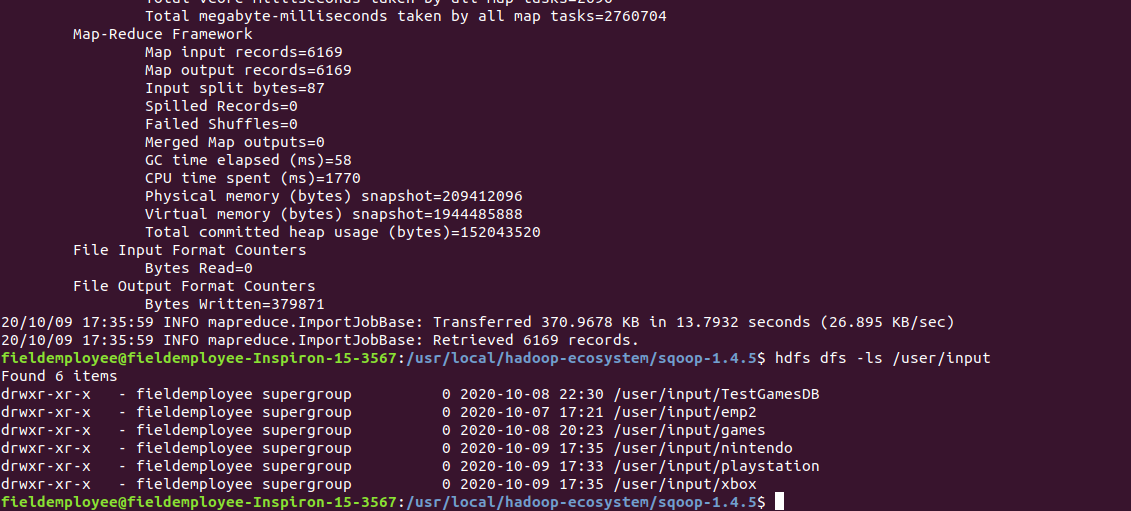
--username root – the username I have when entering the MySQL server

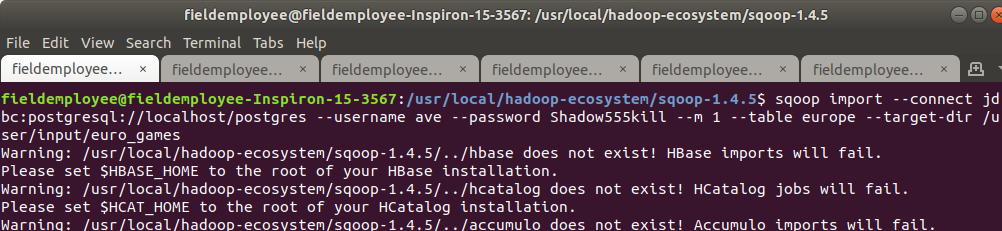
--password Shadow555kill – the password I enter when I’m trying to connect to MySQL

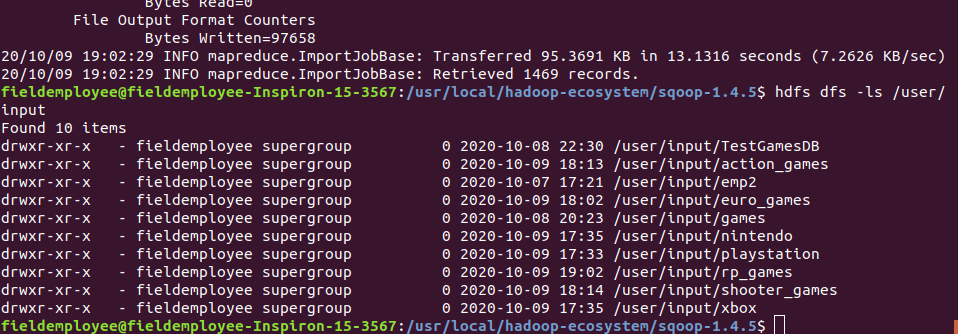
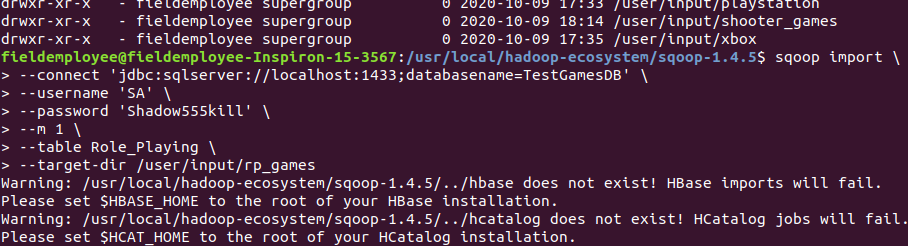
--m 1 – the number of mappers run to extract data from the source database

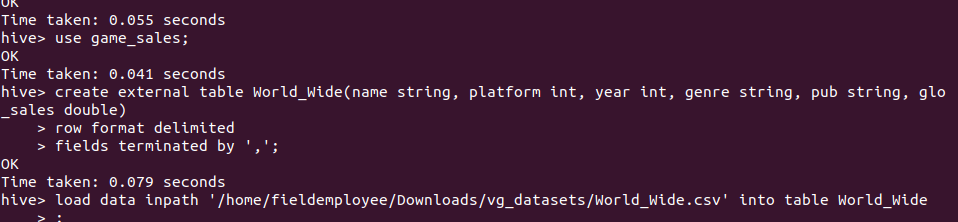
-- table Nintendo – the name of the table I want in the database

--target-dir /user/input/Nintendo – the target destination/directory I want to put the file in

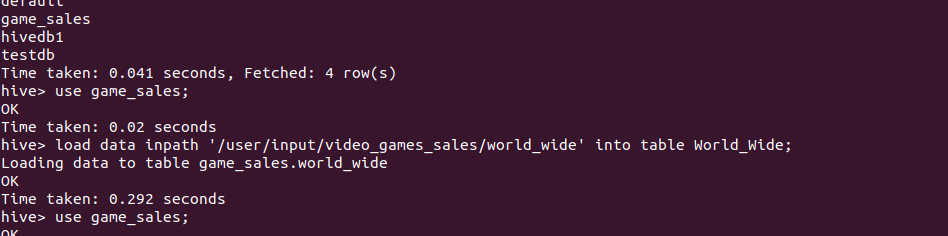


This is the output of the commands I entered. It shows the map-reduce framework and input. Then I entered “hdfs dfs -ls /user/input” to show the tables I have in the directory. Right now, my 3 tables from MySQL that I wanted are in the HDFS along with some extras I tested. Next step was to work on importing tables from PostgreSQL and MS SQL Server. 

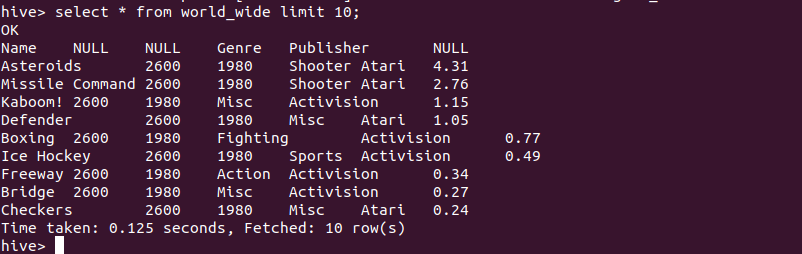
Just like with MySQL, I entered the same commands only this time with some changes. The only big change was the replacing the RDBM and of course the database “—connect jdbc:postgresql://localhost/postgres.” I also had to make sure my tables were accessible in PostgreSQL. There will be an error if “grant all privileges on all tables in schema public to user” is not entered. After that, the output was the same from when I entered Sqoop for MySQL. Finally, it was time to do the same for the database/tables in MS SQL Server.

All the 9 tables I have in the 3 RDBM are now in HDFS, now its time to work on Hive. Starting Hive required me to type “hiveserver2” in the terminal to get it running. While I was in Hive, I created a new database and an external table for that database. The external table is used to load a csv file into that table with the matching schema. The schema must match with the existing columns in the csv file otherwise an error will pop up.



The next step in the process was to load the data into the external table within Hive.

Now to select the table and show that the csv file did go into the empty external table that I created. I also set the limit to 10 to make sure the terminal wouldn’t overflow with so much data.



Now I can check on my HDFS files on the localhost. I do plan to organize all of these files and put them into one folder “video\_games\_sales.” 