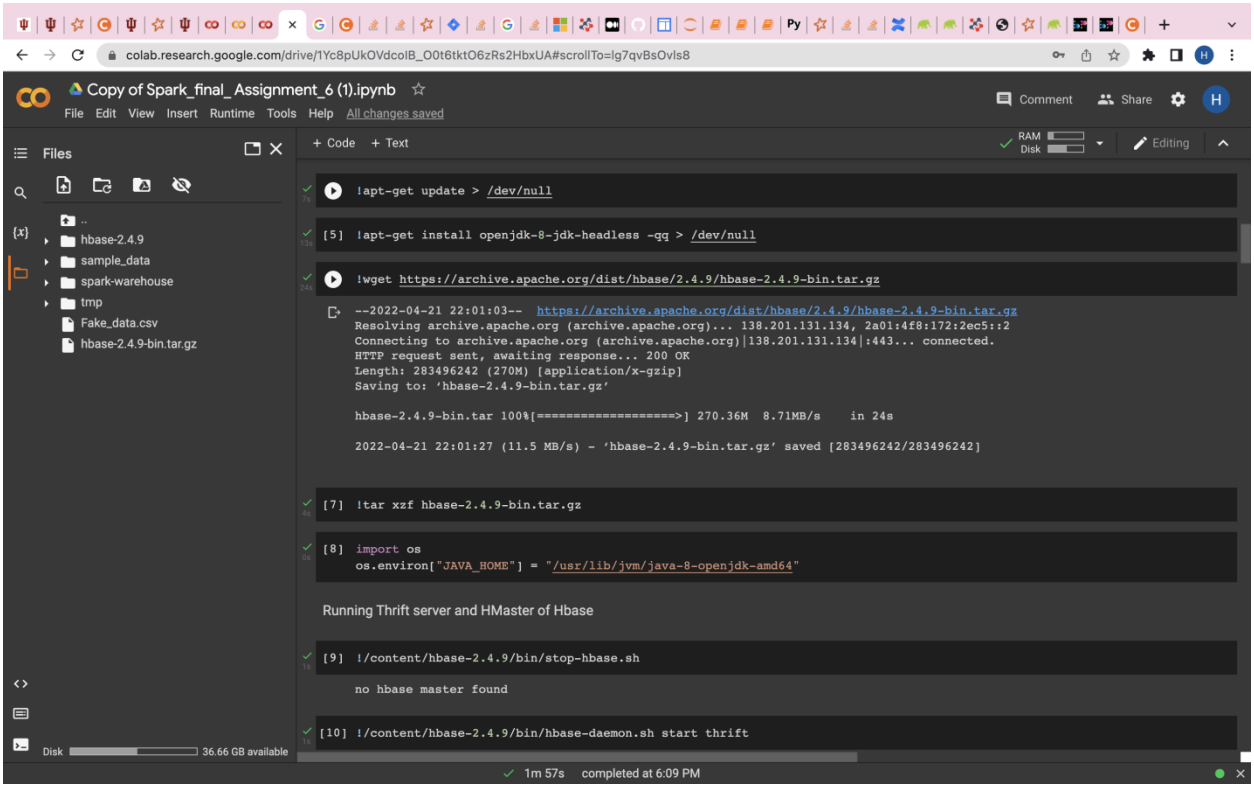
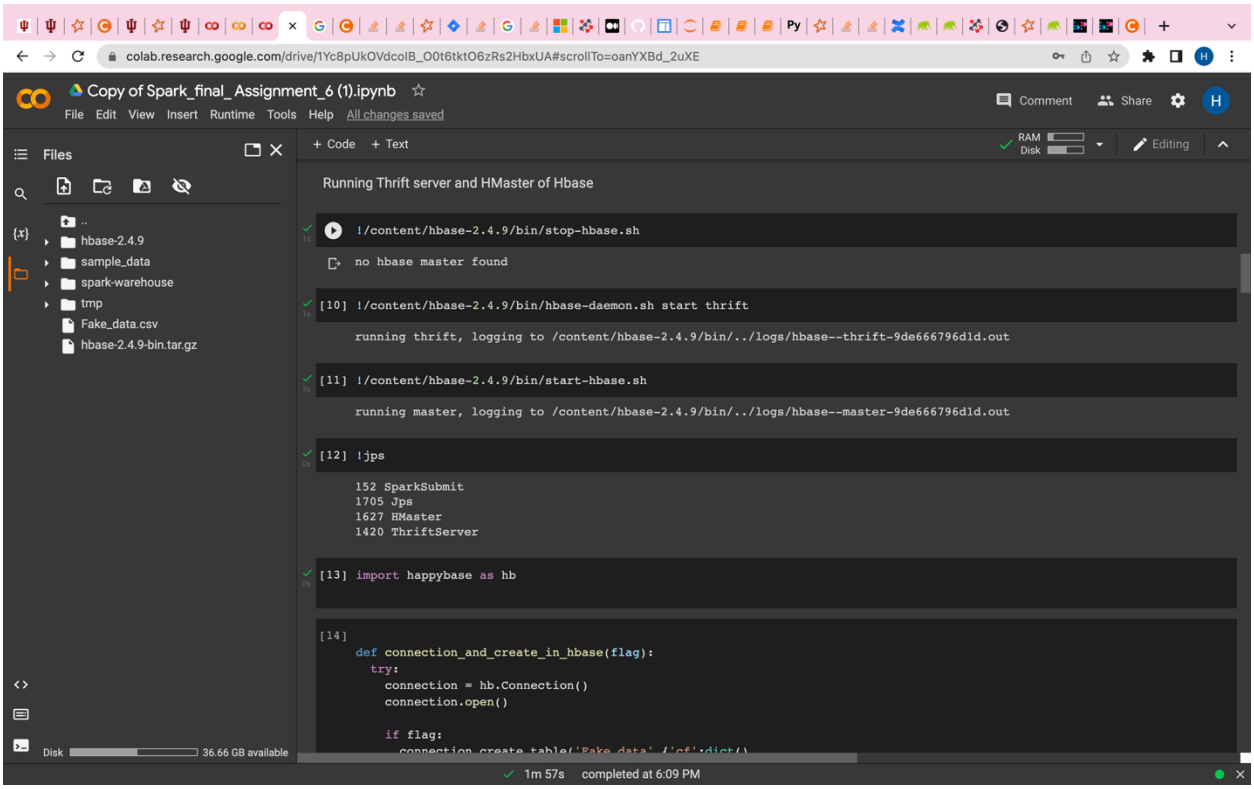
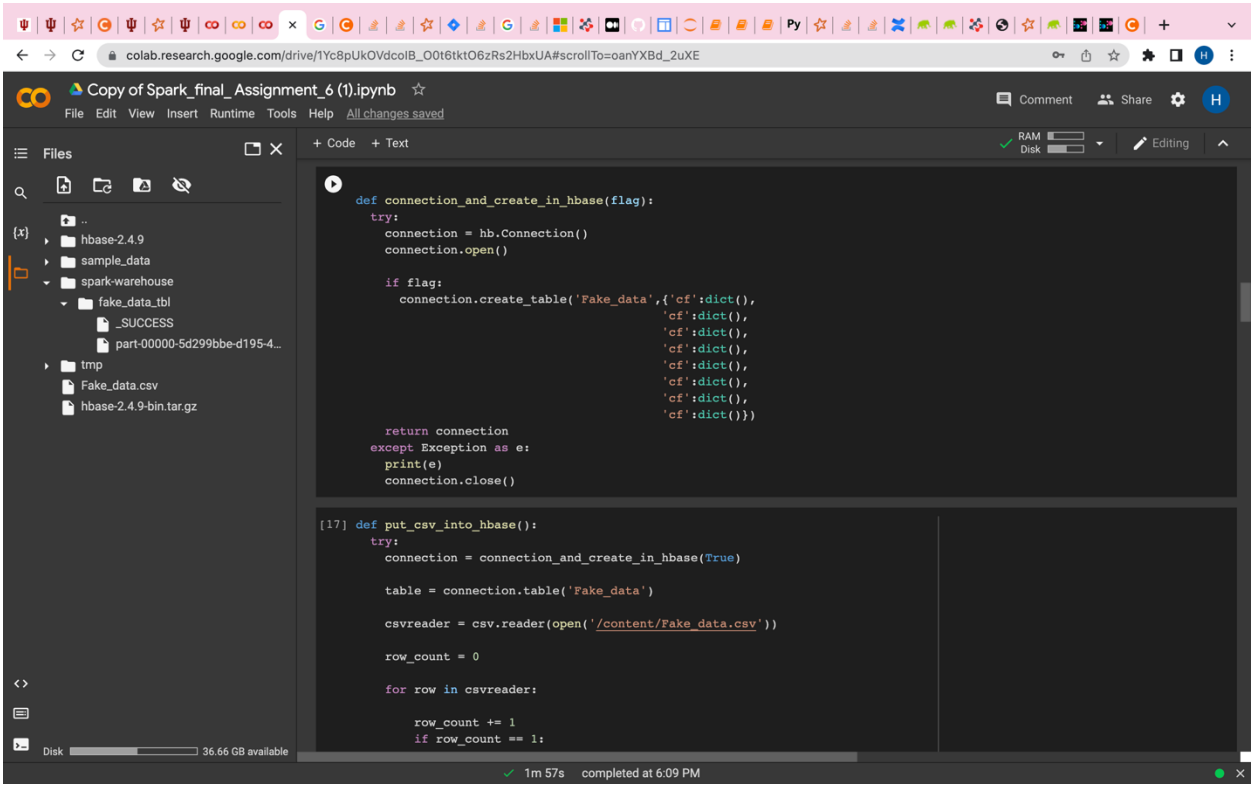
Setting up Java Home system variable and installing JDK and HBase



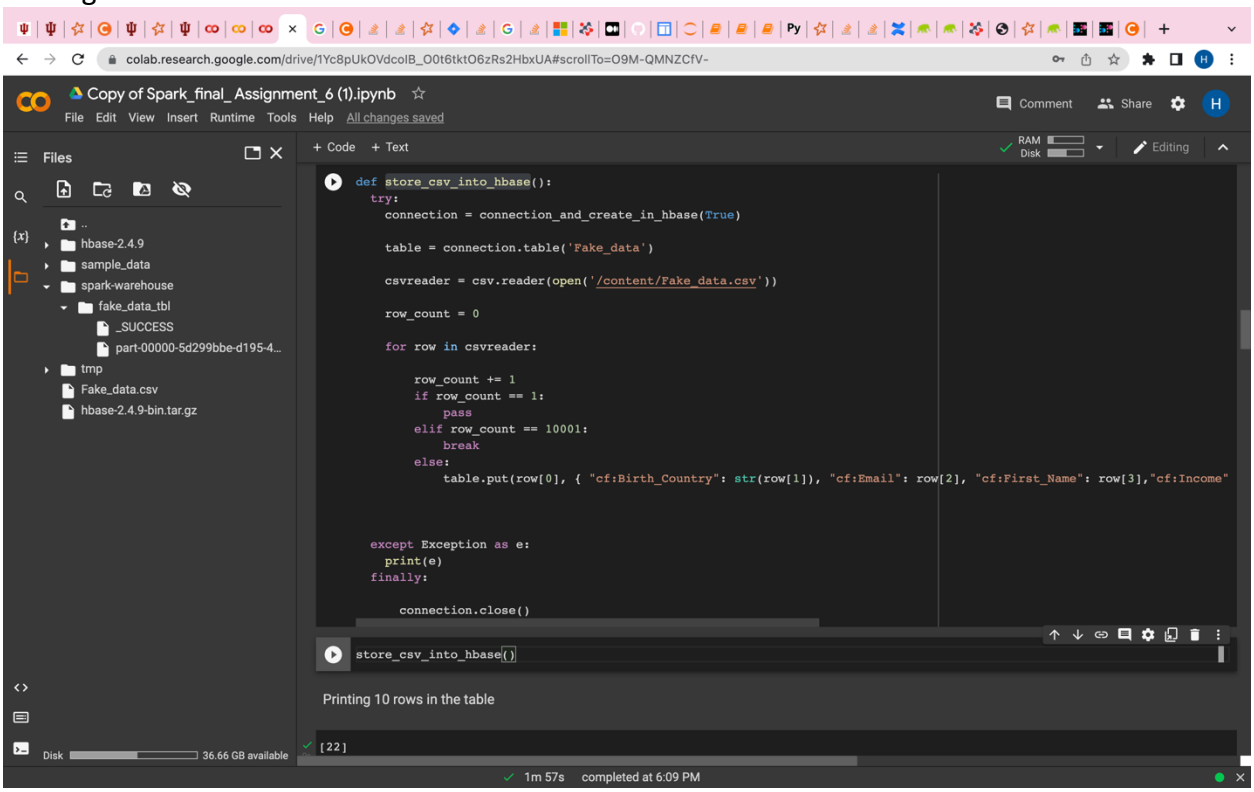
Running Thrift server and HMaster of Hbase. Checking the daemons which are running and installing Happybase



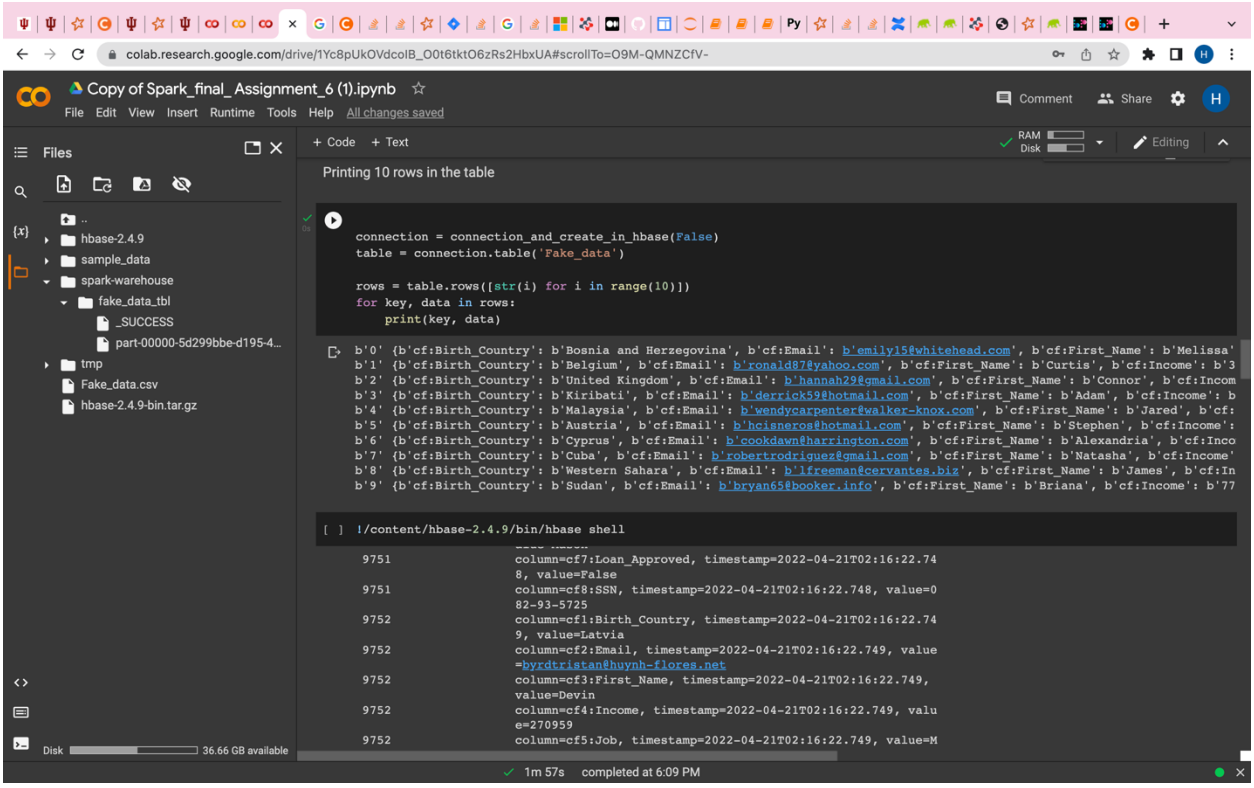
Connecting to Hbase and Creating a Table named Fake_data with column family schema



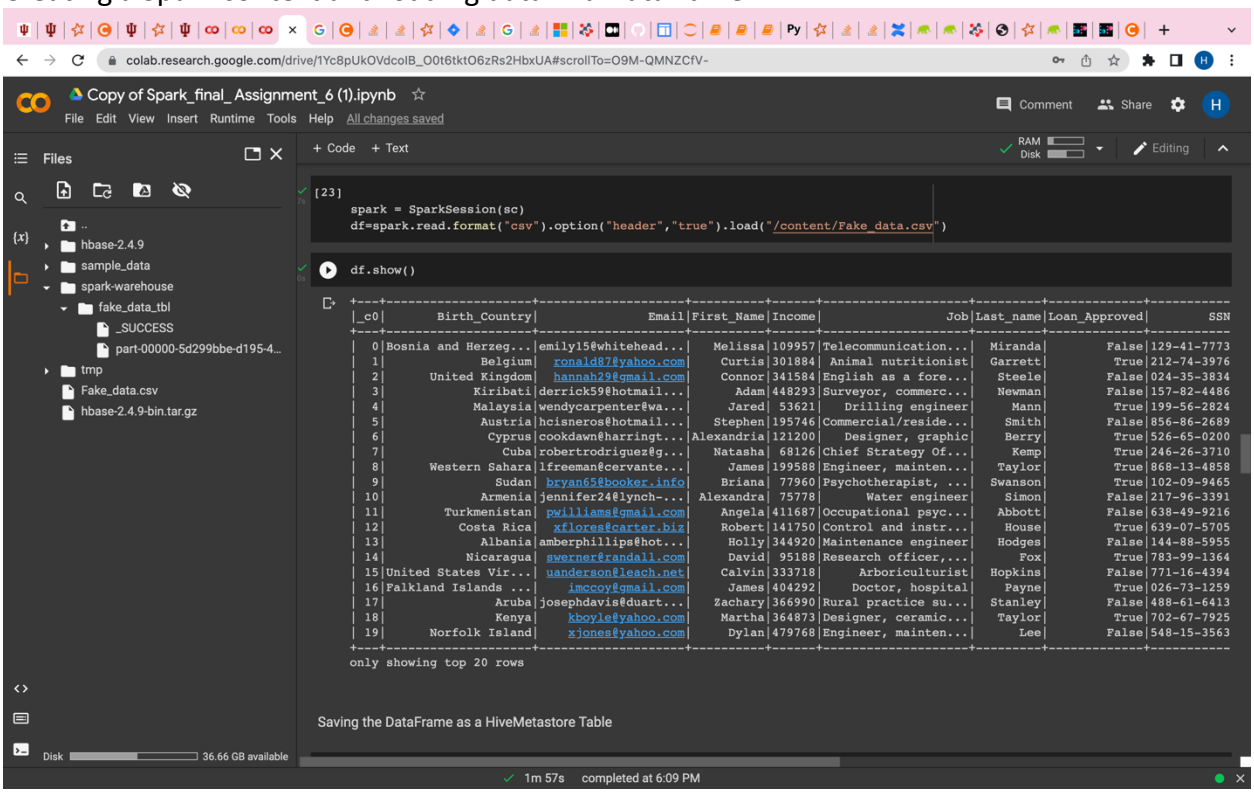
Putting the CSV of Fake data into Hbase Table



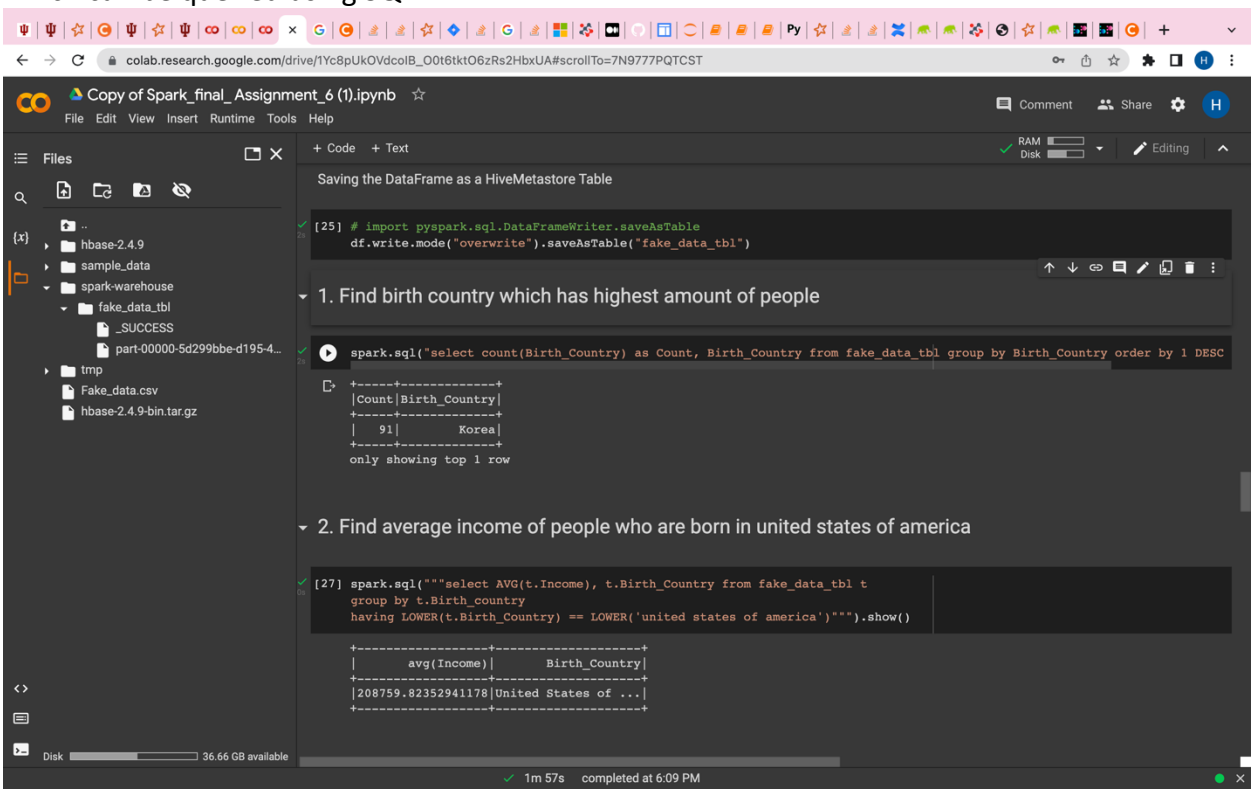
Printing a few rows inserted into the Hbase Table using HappyBase and scanning all the rows of that Table through Hbase Shell

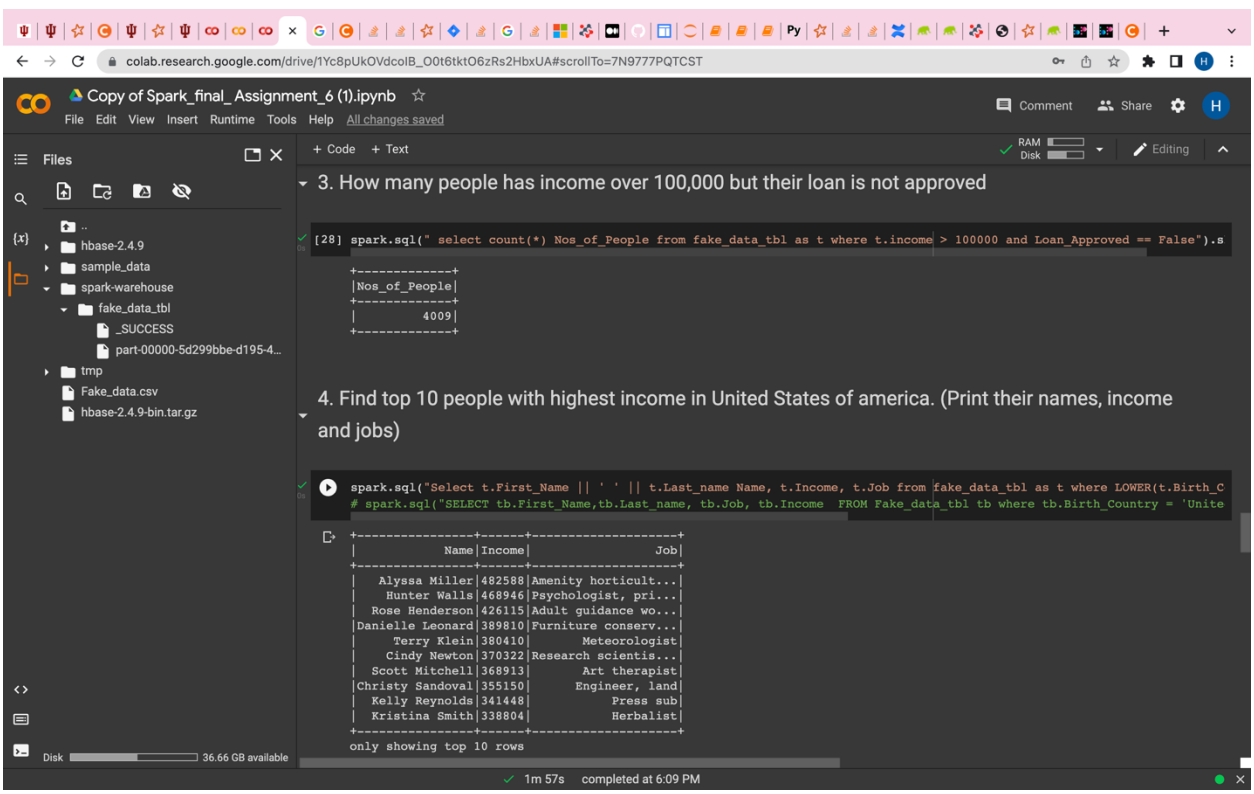


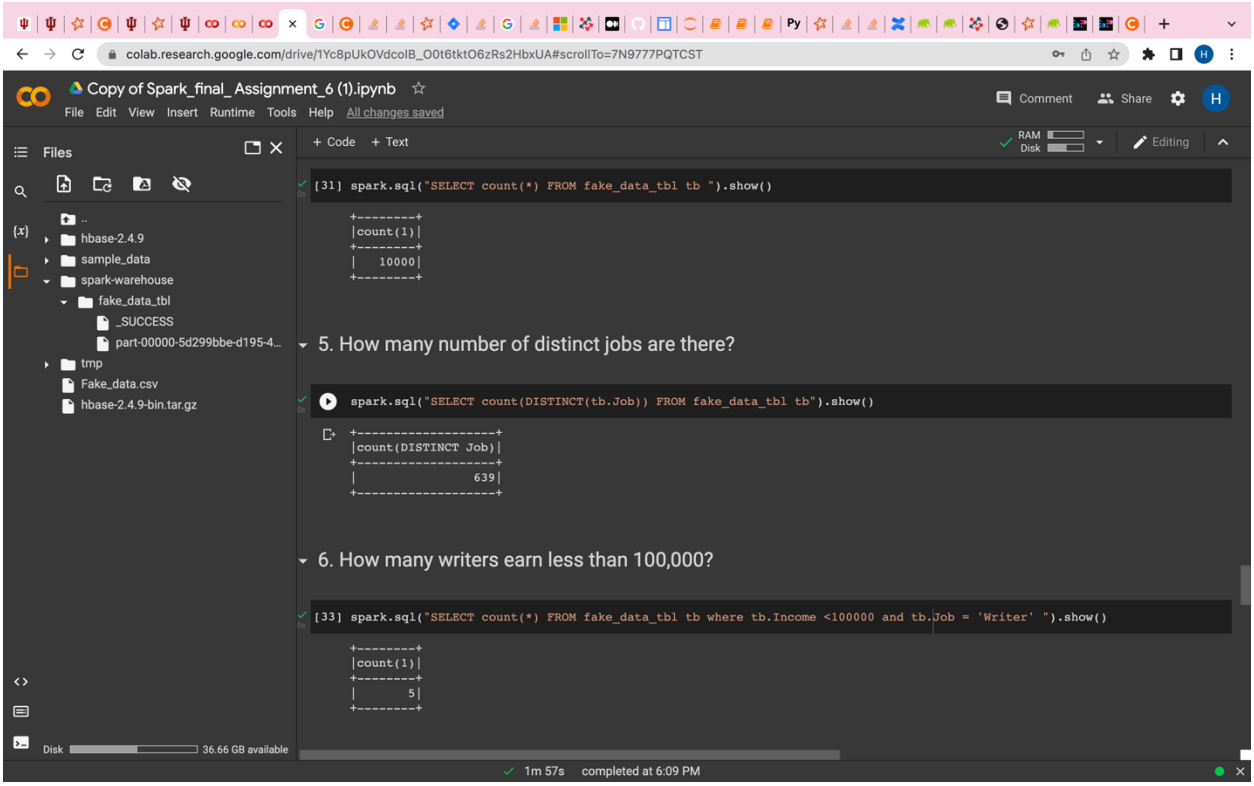
Creating a Spark context and loading data in a Dataframe



Querying the Hbase Table is difficult so I had to save the dataframe into HiveMetastore Table which can be queried using SQL







References:
https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.sql.DataFrameWriter.saveAsTable.html#pyspark.sql.DataFrameWriter.saveAsTable

https://spark.apache.org/docs/2.2.0/sql-programming-guide.html#:~:text=Spark%20SQL%20is%20a%20Spark,information%20to%20perform%20extra%20optimizations

https://happybase.readthedocs.io/en/latest/user.html

https://happybase.readthedocs.io/en/latest/api.html

https://sparkbyexamples.com/apache-hbase-tutorial/

https://www.youtube.com/watch?v=CsYaTIUvez0

https://www.rittmanmead.com/blog/2015/05/loading-updating-and-deleting-from-hbase-tables-using-hiveql-and-python/

Appendix:

I tried creating an External Table in Hive which would be associated with Hbase Table. However, I was facing issues with the StorageHandler class, which I wasn't able to resolve.