**Loading**

After finalizing the tables, an engine was created in order to load the tables to ElephantSQL. This was done using a connection string. In order to hide the username and password for the connection, the string was saved into a separate file named config. In addition, a gitignore file was created in the root folder to ignore the config file in the checkpoints. For most of the tables, a df.to\_sql() was performed with distinguishing names, a connection to the engine, parameter if\_exists='append' and index=True. For the large tables such as sales, prices range, price changes, and product ecommerce, datasets were loaded in SQL using psycopg2 and StringIO. Due to the performance of sqlalchemy being slow with very big datasets, this led to a search for a more efficient way to load the tables. By using StringIO, the data frames could be stored in csv in memory and read in chunks into PostgreSQL using COPY STDIN. Based on time estimates, this was at least 20 times faster for the very long tables (originally 80+ million rows). After all the tables were loaded into the database, a primary key was added to various tables that had an integer IDs as a column. With the final tables loaded, a final check using the table names function was run to ensure that the tables were successfully added to the database. Lastly, the engine was closed to avoid any errors or conflicts.

