Let’s say that I have been assigned by a Senior Data Engineer to assist in creating a Staging server as part of the Data management workflow for an online startup store. The Senior Data Engineer has provided me with some steps to guide me in the creation of the staging area. He has also requested for screenshots showing the results of the steps that I have taken to arrive at the staging server that you have created.

1 – Starting the database. For running the PostgreSQL service, there are 2 commands that we have to commit. Sudo service postgresql start & sudo -u postgres psql. After that, we will see the postgres=# in the command line, which means that we are in postgres environment, and we are using postgres user account.

Text

Description automatically generated

2- Create a database. The database name is bill\_data\_wh. \l command shows us the names of all the databases in the system. This way we can confirm the database creation.

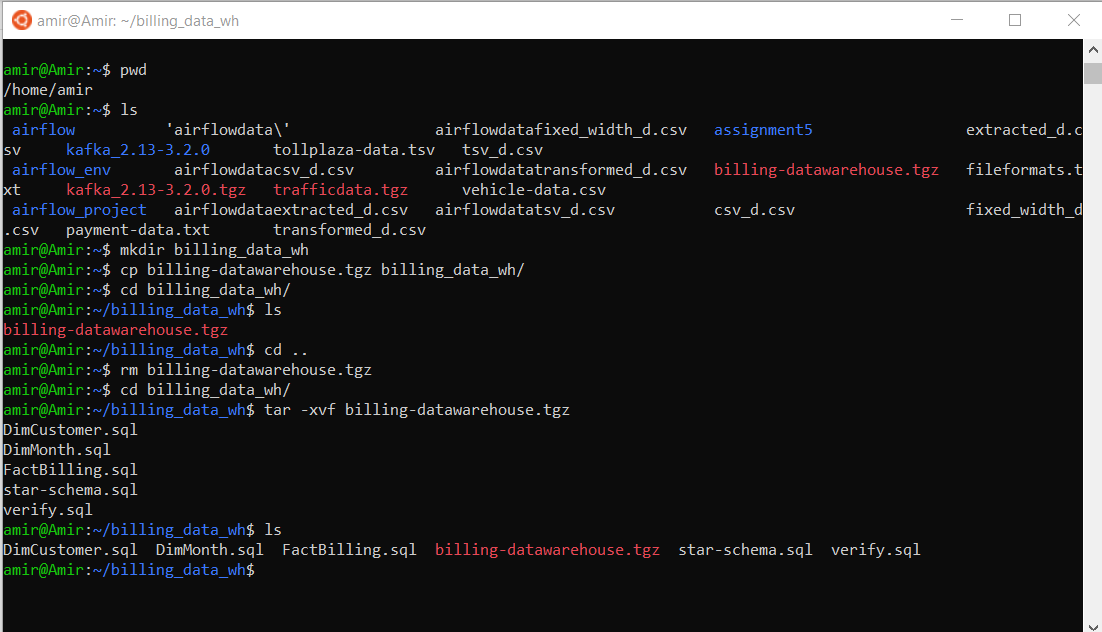
A picture containing text

Description automatically generated

3- Download the sql files for creating schemas and insertion intor the tables. For this step, I opened another linux command line (terminal) and use the wget command to download a rar file from it.

Text

Description automatically generated

4 & 5 – (4) I have to unzip it. So using the tar -xvf command, I try to unzip it. But first I need to copy this file to a separate directory, in order not to make mess with the current directory. So, for this purpose, first I check the current directory, which is home. Then I list out the items in the home directory. Then I make a separate directory for the tgz file, and move it from home to its own directory. Then in the same directory I unzip it. (5) The last line represents the 5 sql files. 

6 – Creating the Schema

6-1 – First table: FactBuilding. This table has 4 columns, one serial data type and three integers.

But at the bigenning I have to connect to the database. \c bill\_data\_wh command establishes the connection.

Text

Description automatically generated

6-2 – Second table: DimMonth. This table contains all the information of the month related to the monthid column of the FactBilling

Text

Description automatically generated

6-3 – Third table: DimCustomer. Likewise the previous table, this table holds all the information regarding the customerid column of the FactBilling table.

**Text

Description automatically generated**

6-4 – Altering the FactBilling table. In this step, I add a foreign key to the table. Foreign key is the key to relate to another tables’ primary key. This way the relational databases work. So in this picture, I make two foreign keys in factbilling, the first one is customerid, which is going to reference the same customerid in DimMonth table, and the second one is monthid, and it is going to be referred by the monthid of the DimMonth table.

Text

Description automatically generated

7 – Loading the dimension into the DimCustomer table. As there are roughly 1000 rows of data to enter into the table, it is good to have a command that reads the “.sql” file, and inserts the pre-written rows into the table. So for this purpose, the command is \i ‘destination\_of\_the\_sql\_file’. I use this command to insert into the table.

Graphical user interface, text

Description automatically generated

BTW, next picture is the .sql file that is already inserted (just 23 row od 1000 data rows):

Text

Description automatically generated

8 – The same process for the DimMonth table.

Text

Description automatically generated

The next picture shows some data rows of the sql file.

Graphical user interface, text

Description automatically generated

9 – The same step for FactBilling table happens. I have to mention the this third table took to long to be inserted. 132000 of data rows, it is huge.

Text

Description automatically generated

Again, just a glance to the sql file:

Text

Description automatically generated

10 – Verifying the Data Insertion. In the next picture I represent the sql file which shows the command for requesting the count of data for each table.

Text

Description automatically generated

And by running it the same way as insertion process, I will have this output:

Text

Description automatically generated

So this shows that all of the data have ben inserted into their tables, safe and sound.