In order to import a csv file to an existing table in a PostgreSQL database, the column names in the table and in the csv file have to be matching. To adjust the columns names to the prevalent naming conventions and make the data more readable, I’ve renamed some of them. The main issue was that column names were preceded with the name of the entity they related to, i.e. *seller\_city* in the file relating to sellers.

***I’ve eventually erased “” in all csv files, due to inconsistency between the first row and the rest.***

The list of changes in the files:

1. *olist\_sellers\_dataset*
   1. seller\_id 🡪 id
   2. seller\_zip\_code\_prefix 🡪 zip\_code
   3. seller\_city 🡪 city
   4. seller\_state 🡪 state
2. *olist\_products\_dataset*
   1. product\_id 🡪 id
   2. product\_photos\_qty 🡪 photos\_qty
   3. product\_weight\_g 🡪 weight\_g
   4. product\_length\_cm 🡪 length\_cm
   5. product\_width\_cm 🡪 width\_cm
   6. product\_height\_cm 🡪 height\_cm
3. *olist\_orders\_dataset*
   1. order\_id 🡪 id
   2. order\_status 🡪 status
   3. order\_purchase\_timestamp 🡪 purchase\_timestamp
   4. order\_approved\_at 🡪 approved\_at
   5. order\_delivered\_carrier\_date 🡪 delivered\_carrier\_date
   6. order\_delivered\_customer\_date 🡪 delivered\_customer\_date
   7. order\_estimated\_delivery\_date 🡪 estimated\_delivery\_date
4. *olist\_order\_reviews\_dataset*
   1. review\_id 🡪 id
   2. review\_score 🡪 score
   3. review\_comment\_title 🡪 comment\_title
   4. review\_comment\_message 🡪 comment\_message
   5. review\_creation\_date 🡪 creation\_date
   6. review\_answer\_timestamp 🡪 answer\_timestamp
5. *olist\_order\_payments\_dataset*
   1. payment\_sequential 🡪 sequential
   2. payment\_type 🡪 type
   3. payment\_installments 🡪 installments
   4. payment\_value 🡪 valu
6. *olist\_geolocation\_dataset*
   1. geolocation\_zip\_code\_prefix 🡪 zip\_code
   2. geolocation\_lat 🡪 lat
   3. geolocation\_lng 🡪 lng
   4. geolocation\_city 🡪 city
   5. geolocation\_state 🡪 state
7. *olist\_customers\_dataset*
   1. customer\_id 🡪 customer\_order\_id
   2. customer\_unique\_id 🡪 id
   3. customer\_zip\_code\_prefix 🡪 zip\_code
   4. customer\_city 🡪 city
   5. customer\_state 🡪 state
8. olist\_order\_items\_dataset
   1. order\_item\_id 🡪 items\_number *(such a name is more suitable for the data the variable is representing)*

I’ve also decided to remove columns that I’ve found to be undoubtedly unnecessary, to save space and computational power. I’m aware of the importance of keeping various data even if it seems irrelevant but information gained from the erased columns can be easily retrieved by simple SQL functions (name lengths of other variables, for instance).

It could have been done with the use of the DROP command, however, since I was going to rename then in the original file anyway, erasing them instead was a better option.

The erased columns:

* product\_name\_length
* product\_description\_length

File of origin: *olist\_products\_dataset*

Furthermore, all of the csv files except *product\_category\_name\_translation* had the variable names inside “” quotation marks. Most of them don’t have to be erased since the delimiter can be set as ”*,”*. That doesn’t solve the problem with the quotation mark at the end of the last column name, though, so they are removed from each table. Oddly, none of the tables had a quotation marks for the first variable. Due to this fact, I’ve added a quotation mark at the end of the first variable’s name, so that the delimiter “,” works correctly.