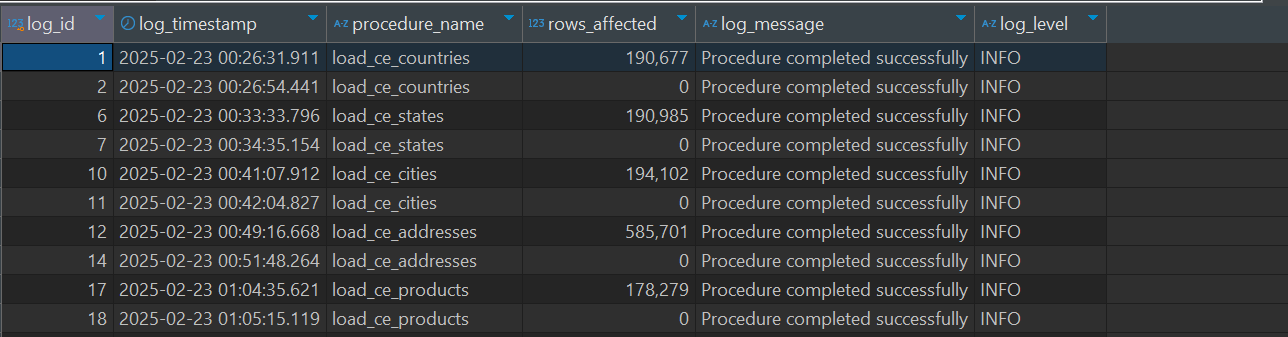
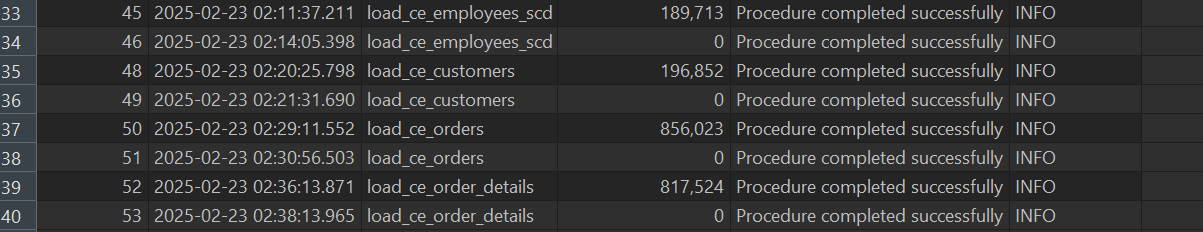
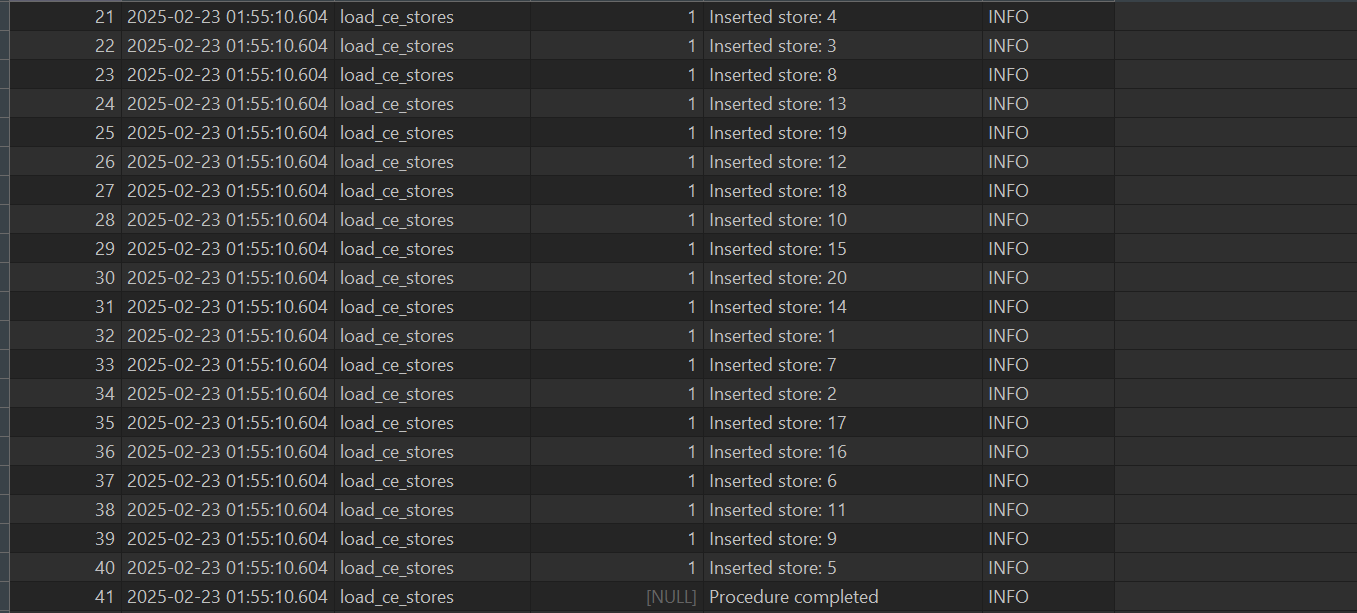
Here I will show the screenshots that code works correctly. In the sql file after each procedure I wrote procedure calls and select clauses for the tables to check If I was doing well. I was running that code 2 times to check if it was correct so in the screenshots there will be 2 rows for each table.(ce\_stores is an exception because I used for loop while inserting the data and after each insert I was calling the procedure which adds row in the log table, I have 20 stores so there is 20 rows for stores in log table)

Table ce\_countries:



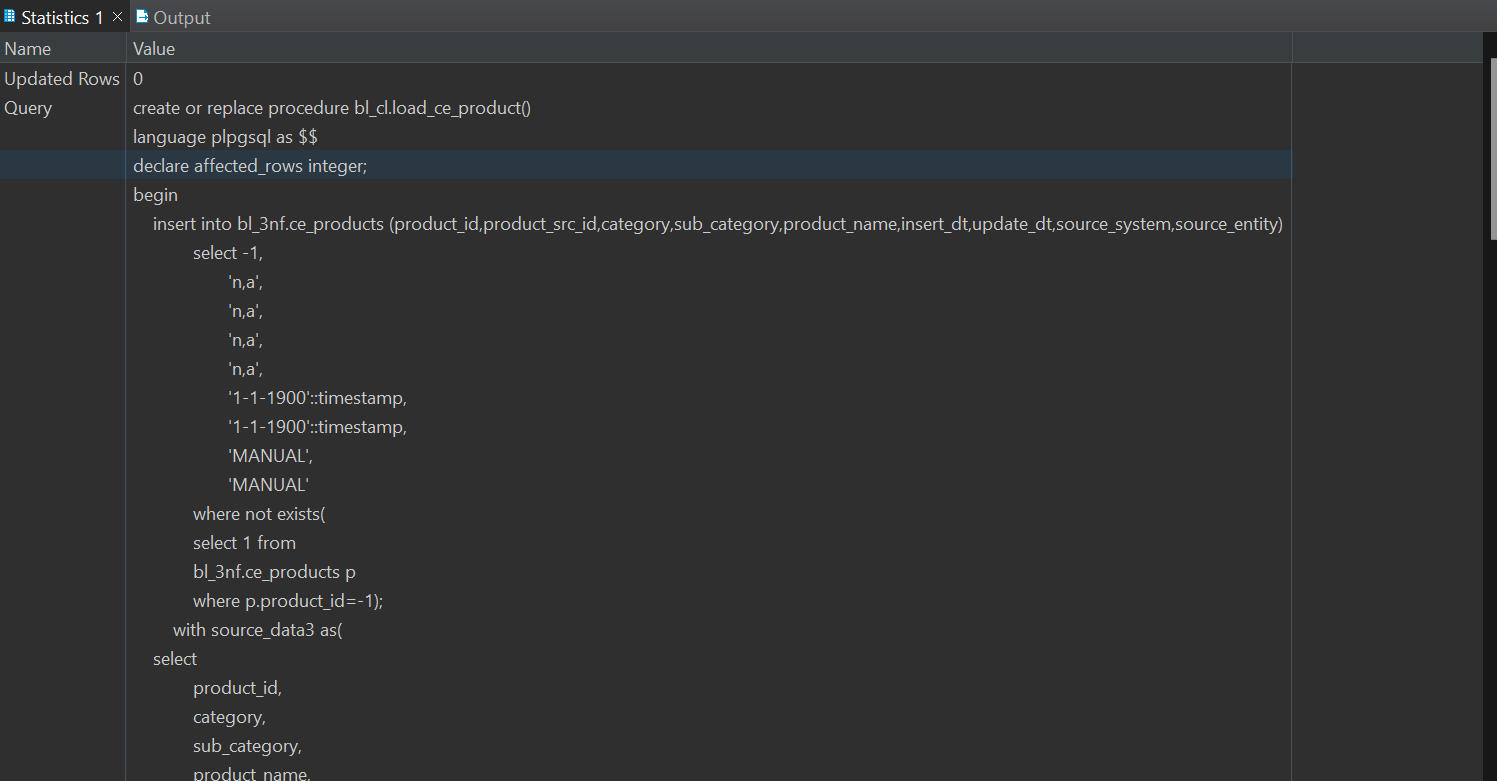


And this is the output for ce\_stores:

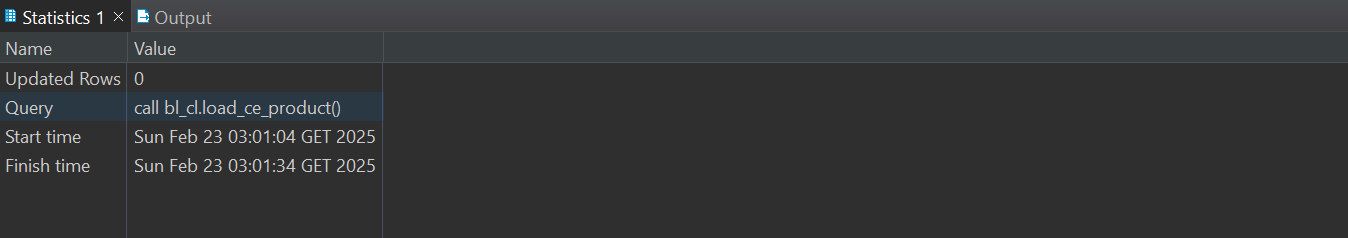


I didn’t take screenshots for all the procedures but each of them were showing 0 rows updated. I will just upload it for the ce\_products but it is the same for every table.

This is procedure creation:



And this is call:



In the code there are 2 ways of how I am loading and inserting the data.

First is a single procedure in that procedure I just use simple insert for default row and then after that I use query from the task 6,since it was inserting already correctly I just used it.

In the function I declare affected\_rows variable

For the counting purposes. After insert is done and every row is in the table affected\_rows becomes the quantity of that rows. Then I call the procedure which inserts all that information in log table. when I call that procedure at second time affected\_rows will be zero since I handle duplicated values and that information goes to log table as well. With the help of this I can prove that I handled that task 😊

Second logic for inserting and loading data goes like this. At first I create function which return type is table and I return all the columns that I need from the source data. Output of the table is everything, that I needed for my table.

After that I create procedure and declare 2 variables. First one again is affected\_rows and second one is rec which type is record. I need it for the for loop. Logic goes like this-if this record is in the table that our function returns and it is not already added in the table that we are inserting, then add it. And I call the procedure for log table for every loop iteration so it adds information one by one and not all at once.

I wanted to use for loop for the small table.because for big data it would take so much time and it tried avoiding that. I just have 20 stores in my dataset so that’s why I used this logic on only this table and not anywhere else.