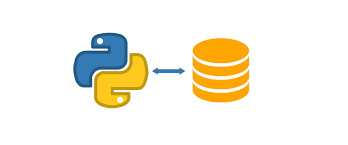
**Data Analysis with Python and SQL**



# **General İnformation**

For data practitioners to efficiently deal with data, Python and SQL are necessary tools. The initial extraction of data from relational databases using SQL queries, followed by data processing and analysis in Python using tools like Pandas, would be a typical use scenario.

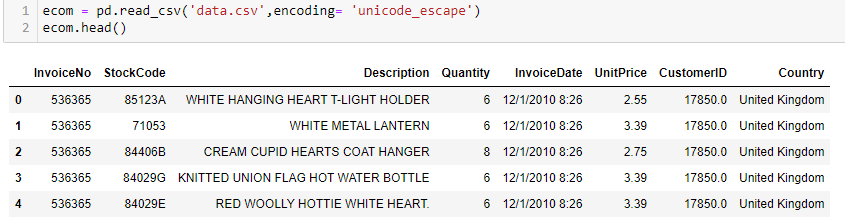
Despite its limitations as a language that is five decades old, SQL is enough to comprehend the principles of data science. Regarding field applications, SQL was not created to handle more intricate data processing and manipulation. Python is a well-documented, high-level language with a distinctive data analysis package called "Pandas," making the decision between Python and SQL a little more challenging.

Applications are made with Python, while database access and data retrieval are done with SQL. Python is used to simultaneously evaluate and change data using time-series tests, regression tests, and other calculations. The best feature of SQL is its capacity to combine data from different tables inside a single database.

In this article, we are going to analyze the data.csv file with both Pandas and SQL. The file is about buying materials. Although there are lots of SQL tools, I used MS SQL Server for this article. I loaded the “data.csv” file into the MS SQL Server database beforehand. If you're curious about how to read a file from a SQL Server database using Python code and open a file from a SQL Server database in Tableau, check out my related article [here](https://medium.com/@mevltyldz1/analyzing-data-with-python-sql-and-tableau-simultaneously-ab3154945eaa).

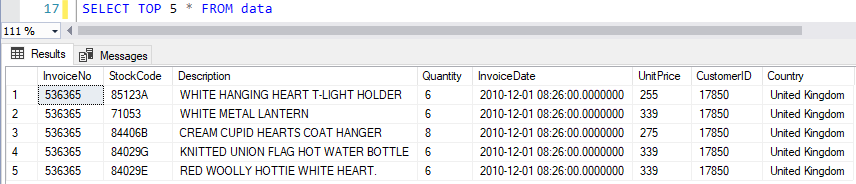
# **2. Reading the File**

First, let us read the “data.csv” file for the first 5 rows with both Pandas and SQL. To open the file with Pandas, I first use the “[pd.read\_csv](https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html#pandas.read_csv)” function, and to see the first five rows, I use the “[head()](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.head.html#pandas.DataFrame.head)” function.



**Table 1: Read File with Pandas**

To read the file with SQL, I use the “SELECT \* FROM data” query, and to see the first five rows, I use the “TOP 5” statement.



**Table 2: Read File with MS SQL Server**

This table contains information about the buying materials. The table’s columns contain: customer ID, invoice number, invoice date, stock code, description, quantity, unit price, and country. This table has 8 columns.

# **3. Beginning the Analysis**

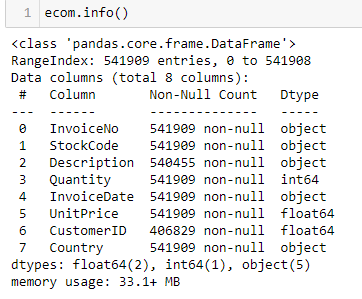
It is now time to analyze. We want to show four examples. The first one is to analyze the *“information about the data.”* The second one is *“List the country names and number of records belonging to each country.” The third one is, “How many records are there in the United Kingdom(UK)?” And the last example is about “What is the maximum total spending in the whole dataset and show the records of that spending?”*

### Example 1: Info about the data

It’s important to know the data types and record amount for each column. Hence, we look at information for that purpose. Python and SQL are used, respectively, to display the analysis findings.

#### Step 1: Use Python to display

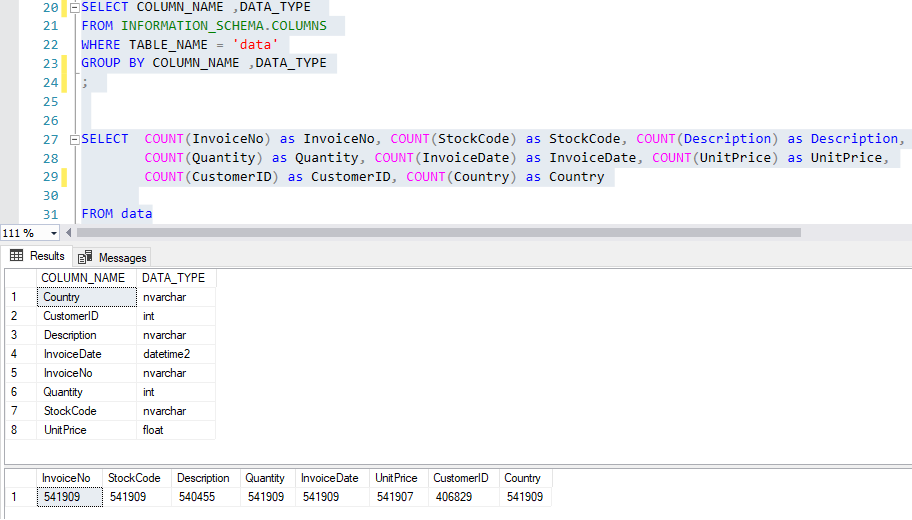
Now, if we write info(), we see the general information about the data set, like the column name, total row count, not null amount, data type, and memory usage. For example, there are 406829 CustomerIds, whose data type is float 64.



**Table 3: General Information About the Dataset with Python**

#### Step-2: Show with SQL

The "COLUMN\_NAME" statement gives the column names of the table, and the ”DATA\_TYPE” statement gives the column data types automatically. We should use the “INFORMATION\_SCHEMA.COLUMNS” statement for the table name.



**Table 4:General Information About the Dataset with SQL**

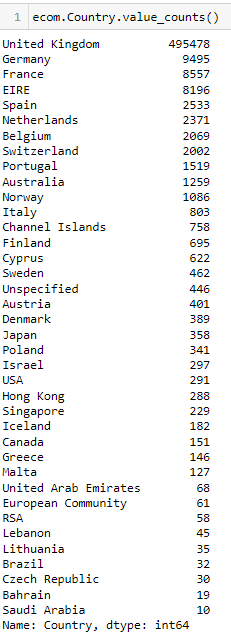
As we see, the data type “string” in Pandas equals “nvarchar” in SQL, the data type “int64” in Pandas equals “int” in SQL, the data type “float64” in Pandas equals “float” in SQL,

### Example 2: List the country names and the number of records for each country.

We wonder which country’s citizens spent more money, from the biggest to the lowest. So, we show the results of analyzing Python and SQL.

#### Step 1: Use Python to display

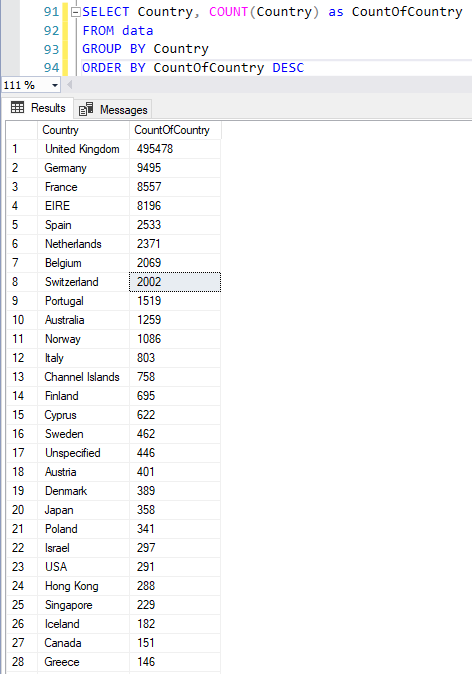
We are now using Python to analyze the query. To determine how many records there are for each nation, we create the "value counts()" statement. As a result, we can see that the United Kingdom, which is at the top of the list, has 495478 records.



**Table 5:List the Country Names and the Number of Records for Each Country with Python**

#### Step-2: Use SQL to display

We now perform a SQL analysis on the query. To see the country name, we write "country" as the column name and "COUNT(country)" as the "CountOfCountry" number of records. We see the same results with the Python results.



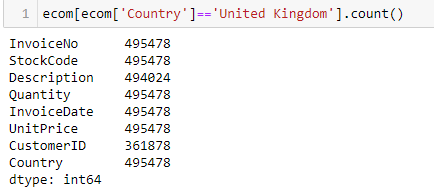
**Table 6: List the Country Names and the Number of Records for Each Country with SQL**

### Example 3: How many records for each column does the country of the United Kingdom have?

There are, as far as we are aware, 541,909 records in all. The United Kingdom has 495,478 records out of the total. It is a huge number. We want to find out more information about the UK.

#### Step-1: Use Python to display

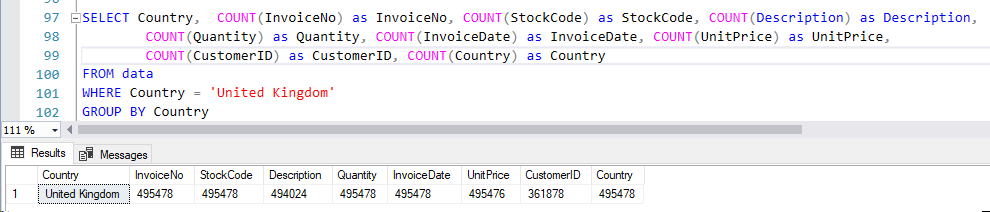
Python is now used to examine the query. The phrase "ecom[ecom['Country']=='United Kingdom'].count()" is written for the query we're trying to obtain an answer for. For instance, we see that there are 361878 different customers.



**Table 7: Records Number for Each Column of the United Kingdom with Python**

#### Step-2: Use SQL to display

#### SQL is now used to examine the query. To see the country name, we write "Country" as the column name and "COUNT(ColumnName)" as the number of records for each column. We see the same results with the Python results.



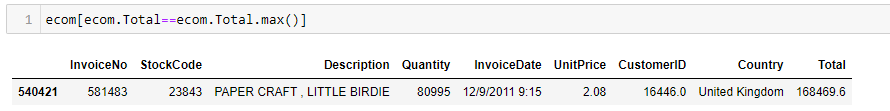
**Table 8:Records Number for Each Column of the United Kingdom with SQL**

### Example 4: What is the total maximum spending in the entire dataset, and can you show the records of that spending?

We want to learn what the total maximum spending is in the entire dataset and get details about it, like the total spending, date of the spending, amount, kind of thing bought, and country.

#### Step-1: Use Python to display

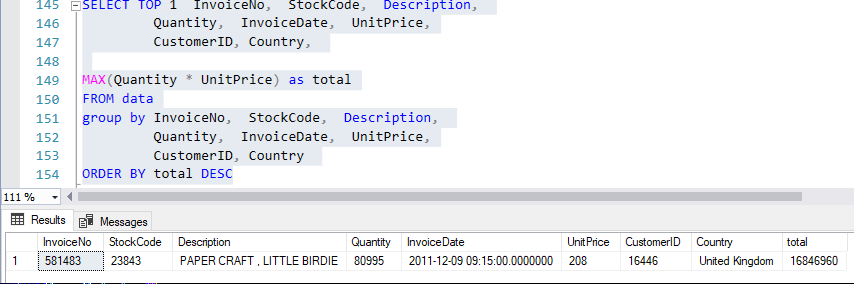
We produce the "ecom [ecom.Total==ecom.Total.max()] "statement of the query for which we want a response. The item's description is "paper craft," its quantity is 80995, its invoice date is 12/9/2011 9:15, its unit price is 2.08, its customer ID is 16446.0, its country is the United Kingdom, and its total cost is 168469.6.



**Table 9: Total Maximum Spending in the Entire Dataset with Python**

#### Step-2: Use SQL to display

We now perform a SQL analysis on the query. We formulate the search term for the issue that needs a response. The solution is then put beneath the question. The outcome is the same as Python's.



**Table 10: Total Maximum Spending in the Entire Dataset with SQL**

# **4. Conclusion**

In this article, we analyzed four questions and checked their truthiness with Pandas and SQL. We concurrently used Python and SQL to analyze the data and produced the same outcomes. Thus, we are certain that our analysis is accurate. Using this strategy, advised elbow, you can analyze any dataset with confidence in your analysis's outcomes. I hope you enjoyed reading this article. I hope this is useful for you, and I will be very happy if you follow me and support me with claps. I appreciate you reading the blog.