



# SONY AIESEC TASK DEBRIEF

Luiz C. E. Amoras Jr.  
Junior Data Science Analyst Application

## 1 - SQL Task - Description

1.1 - SQL Task - Solution Problem #01

1.3 - SQL Task - Solution Problem #02

1.3 - SQL Task - Solution Problem #03

## 2 - Python Task - Description

2.1 - Python Task - Solution Problem



# SQL TASK

--

# 1 - SQL Task - Description

---

## 1. Prepare tables and populate with Dummy data:

- Company table fields (CompanyID, CompanyName)
- Customer belongs to Company fields (CustomerID, CustomerName, CompanyID)
- Customer has Net Sales fields (Customer ID, Quarter [data like 2019Q1], Amount)

## 2. Make Selection that will show the Net Seales for Company ID 1.

## 3. Create pivot that will show Customers as rows, quarters as columns and Net Sales Amounts as Values as table valued function, parameter is Company ID.

# 1.1 - SQL Task - Solution Problem #01

---

## 1. Prepare tables and populate with Dummy data:

Create a new Data Base or drop if it already exist



Create tables: Company, Customer and Customer Net Sales



Define tables setting



Insert dummy data into each table

# 1.1 - SQL Task - Solution Problem #01

[0] Input:

The screenshot shows the MySQL Workbench interface. The 'Script\_1 - Create tables' window contains the following SQL script:

```
1 # 1. Prepare tables
2 # - Company table fields (CompanyID, CompanyName) + populate with Dummy data
3 # - Customer belongs to Company fields (CustomerID, CustomerName, CompanyID) + populate with Dummy data
4 # - Customer has Net Sales fields (CustomerID, Quarter, Amount) + populate with Dummy data Quarter like 2019Q1
5
6 • DROP DATABASE IF EXISTS SALES_DB;
7 • CREATE DATABASE IF NOT EXISTS SALES_DB;
8 • USE SALES_DB;
9
10 • CREATE TABLE COMPANY (
11     COMPANY_ID INT AUTO_INCREMENT NOT NULL,
12     COMPANY_NAME VARCHAR(100) NOT NULL,
13     PRIMARY KEY (COMPANY_ID),
14     UNIQUE KEY (COMPANY_NAME)
15 );
16
17 • CREATE TABLE CUSTOMER (
18     CUSTOMER_ID INT AUTO_INCREMENT NOT NULL,
```

The 'Output' window at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
2	18:47:48	CREATE DATABASE IF NOT EXISTS SALES_DB	1 row(s) affected	0.093 sec
3	18:47:48	USE SALES_DB	0 row(s) affected	0.015 sec
4	18:47:48	CREATE TABLE COMPANY (COMPANY_ID INT AUTO_INCREMENT NOT NULL, COMPANY_NAME VA...	0 row(s) affected	0.641 sec
5	18:47:48	CREATE TABLE CUSTOMER (CUSTOMER_ID INT AUTO_INCREMENT NOT NULL, CUSTOMER_NAME...	0 row(s) affected	0.797 sec
6	18:47:49	CREATE TABLE CUSTOMER_NETSALE (CUSTOMER_ID INT, YEAR_QUARTER VARCHAR(100) NOT ...	0 row(s) affected	0.812 sec
7	18:47:50	INSERT INTO COMPANY (COMPANY_NAME) VALUES ('Et Eiam Laoreet LLP'), ('Vulputate Incorporated'), (A...	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0	0.093 sec
8	18:47:50	INSERT INTO CUSTOMER (CUSTOMER_NAME, COMPANY_ID) VALUES ('Colton X. Gould', '9'), ('Baxter D. ...	60 row(s) affected Records: 60 Duplicates: 0 Warnings: 0	0.125 sec
9	18:47:50	INSERT INTO CUSTOMER_NETSALE (CUSTOMER_ID, YEAR_QUARTER, AMOUNT) VALUES ('1', '2019Q...	1000 row(s) affected Records: 1000 Duplicates: 0 Warnings: 0	0.641 sec
10	18:50:36	DROP DATABASE 'sales_db'	3 row(s) affected	1.234 sec



# 1.1 - SQL Task - Solution Problem #01

[1] Output:

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view with 'sales\_db' selected and highlighted by a red rectangle. The main editor shows a SQL script titled 'Script\_1 - Create tables'. The script includes comments for table preparation and population, followed by SQL commands to drop and create the 'SALES\_DB' database, and then create tables 'COMPANY' and 'CUSTOMER'. The 'Output' pane at the bottom shows the execution results, including the time taken for each action and the number of rows affected.

**SQL Script:**

```
1 # 1. Prepare tables
2 # - Company table fields (CompanyID, CompanyName) + populate with Dummy data
3 # - Customer belongs to Company fields (CustomerID, CustomerName, CompanyID) + populate with Dummy data
4 # - Customer has Net Sales fields (CustomerID, Quarter, Amount) + populate with Dummy data Quarter like 2019Q1
5
6 DROP DATABASE IF EXISTS SALES_DB;
7 CREATE DATABASE IF NOT EXISTS SALES_DB;
8 USE SALES_DB;
9
10 CREATE TABLE COMPANY (
11     COMPANY_ID INT AUTO_INCREMENT NOT NULL,
12     COMPANY_NAME VARCHAR(100) NOT NULL,
13     PRIMARY KEY (COMPANY_ID),
14     UNIQUE KEY (COMPANY_NAME)
15 );
16
17 CREATE TABLE CUSTOMER (
18     CUSTOMER_ID INT AUTO_INCREMENT NOT NULL,
```

**Output:**

#	Time	Action	Message	Duration / Fetch
1	18:47:44	DROP DATABASE IF EXISTS SALES_DB	3 row(s) affected	3.406 sec
2	18:47:48	CREATE DATABASE IF NOT EXISTS SALES_DB	1 row(s) affected	0.093 sec
3	18:47:48	USE SALES_DB	0 row(s) affected	0.015 sec
4	18:47:48	CREATE TABLE COMPANY (COMPANY_ID INT AUTO_INCREMENT NOT NULL, COMPANY_NAME VARCHAR(100) NOT NULL, PRIMARY KEY (COMPANY_ID), UNIQUE KEY (COMPANY_NAME))	0 row(s) affected	0.641 sec
5	18:47:48	CREATE TABLE CUSTOMER (CUSTOMER_ID INT AUTO_INCREMENT NOT NULL, CUSTOMER_NAME VARCHAR(100) NOT NULL, PRIMARY KEY (CUSTOMER_ID), UNIQUE KEY (CUSTOMER_NAME))	0 row(s) affected	0.797 sec
6	18:47:49	CREATE TABLE CUSTOMER_NETSALE (CUSTOMER_ID INT, YEAR_QUARTER VARCHAR(100) NOT NULL, NETSALE DECIMAL(10,2) NOT NULL, PRIMARY KEY (CUSTOMER_ID, YEAR_QUARTER))	0 row(s) affected	0.812 sec
7	18:47:50	INSERT INTO COMPANY (COMPANY_NAME) VALUES ('Elk Eiam Laoreet LLP'), ('Vulpurate Incorporated'), ('Ant...	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0	0.093 sec
8	18:47:50	INSERT INTO CUSTOMER (CUSTOMER_NAME, COMPANY_ID) VALUES ('Colton X. Gould', '9'), ('Baxter D. No...	60 row(s) affected Records: 60 Duplicates: 0 Warnings: 0	0.125 sec
9	18:47:50	INSERT INTO CUSTOMER_NETSALE (CUSTOMER_ID, YEAR_QUARTER, AMOUNT) VALUES ('1', '2019Q1', ...	1000 row(s) affected Records: 1000 Duplicates: 0 Warnings: 0	0.641 sec

## 1.2 - SQL Task - Solution Problem #02

2. Make Selection that will show the Net Seales for Company ID 1.

Select columns from joined tables

Left join tables by Company\_ID and Customer\_ID columns

Define Company\_ID equal to 1 as condition

Group by columns



# 1.2 - SQL Task - Solution Problem #02

[0] Input:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'sales\_db' expanded. The main editor window is titled 'Script\_2 - NetSale ID 1' and contains the following SQL query:

```
1 # 2. Make Selection that will show the Net Sales for Company ID 1
2
3 SELECT
4     B.COMPANY_ID,
5     B.COMPANY_NAME,
6     A.CUSTOMER_ID,
7     B.CUSTOMER_NAME,
8     A.YEAR_QUARTER,
9     A.AMOUNT
10
11 FROM CUSTOMER_NETSALE AS A
12 LEFT JOIN (
13     SELECT CT.CUSTOMER_ID, CT.CUSTOMER_NAME, CT.COMPANY_ID, CO.COMPANY_NAME
14     FROM CUSTOMER AS CT
15     LEFT JOIN (SELECT * FROM COMPANY) AS CO
16     ON CT.COMPANY_ID = CO.COMPANY_ID
17 ) AS B
18 ON A.CUSTOMER_ID = B.CUSTOMER_ID
19
20 WHERE B.COMPANY_ID IN (1)
21
22 GROUP BY
23     B.COMPANY_ID,
24     B.COMPANY_NAME,
25     A.CUSTOMER_ID,
```

The bottom of the window shows the 'Output' tab with 'Action Output' selected. The 'Object Info' and 'Session' tabs are also visible at the bottom left.

# 1.2 - SQL Task - Solution Problem #02

[1] Output:

The screenshot shows the MySQL Workbench interface. The 'Script\_2 - NetSale ID 1' tab is active, displaying a SQL query. The query is as follows:

```
1 # 2. Make Selection that will show the Net Sales for Company ID 1
2
3 SELECT
4     B.COMPANY_ID,
5     B.COMPANY_NAME,
6     A.CUSTOMER_ID,
7     B.CUSTOMER_NAME,
8     A.YEAR_QUARTER,
9     A.AMOUNT
10
11 FROM CUSTOMER_NETSALE AS A
12 LEFT JOIN (
13     SELECT CT.CUSTOMER_ID, CT.CUSTOMER_NAME, CT.COMPANY_ID, CO.COMPANY_NAME
```

The 'Result Grid' is visible, showing 65 rows of data. The columns are: COMPANY\_ID, COMPANY\_NAME, CUSTOMER\_ID, CUSTOMER\_NAME, YEAR\_QUARTER, and AMOUNT. The data is grouped by company and customer, showing sales across different quarters.

COMPANY_ID	COMPANY_NAME	CUSTOMER_ID	CUSTOMER_NAME	YEAR_QUARTER	AMOUNT
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2018Q1	1643.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2018Q1	1432.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2018Q2	1995.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2018Q2	1159.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2018Q3	1227.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2018Q3	602.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2018Q3	529.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2018Q4	831.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2019Q1	749.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2019Q1	1757.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2019Q2	496.00
1	Elit Etiam Laoreet LLP	8	Ilana G. Murray	2019Q2	1588.00

The 'Output' pane at the bottom shows the execution details: 1 19:00:21 SELECT B.COMPANY\_ID, B.COMPANY\_NAME, A.CUSTOMER\_ID, B.CUSTOMER\_NAME, A.YEAR\_QUARTER... 65 row(s) returned. Duration / Fetch: 0.000 sec / 0.000 sec.

## 1.3 - SQL Task - Algorithm Problem #03

3. Create pivot that will show Customers as rows, quarters as columns and Net Sales Amounts as Values as table valued function, parameter is Company ID.

Select columns from joined tables

Create Sum columns for each quarter information available

Left join tables by Company\_ID and Customer\_ID columns

Group by columns

# 1.3 - SQL Task - Solution Problem #03

[0] Input:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'employees', 'sales\_db', and 'sys' databases. The 'sales\_db' database is selected, showing its tables, views, stored procedures, and functions. The main editor window is titled 'Script\_3 - Pivot Table' and contains the following SQL query:

```
1 3. Create pivot that will show Customers as rows, quarters as columns and Net Sales Amounts as Values as table valued function, parameter is Company ID
2
3 * SELECT
4 B.COMPANY_ID,
5 B.COMPANY_NAME,
6 A.CUSTOMER_ID,
7 B.CUSTOMER_NAME,
8
9 SUM(CASE WHEN YEAR_QUARTER IN ('2018Q1') THEN AMOUNT ELSE 0 END) AS Q1_2018,
10 SUM(CASE WHEN YEAR_QUARTER IN ('2018Q2') THEN AMOUNT ELSE 0 END) AS Q2_2018,
11 SUM(CASE WHEN YEAR_QUARTER IN ('2018Q3') THEN AMOUNT ELSE 0 END) AS Q3_2018,
12 SUM(CASE WHEN YEAR_QUARTER IN ('2018Q4') THEN AMOUNT ELSE 0 END) AS Q4_2018,
13 SUM(CASE WHEN YEAR_QUARTER IN ('2019Q1') THEN AMOUNT ELSE 0 END) AS Q1_2019,
14 SUM(CASE WHEN YEAR_QUARTER IN ('2019Q2') THEN AMOUNT ELSE 0 END) AS Q2_2019,
15 SUM(CASE WHEN YEAR_QUARTER IN ('2019Q3') THEN AMOUNT ELSE 0 END) AS Q3_2019,
16 SUM(CASE WHEN YEAR_QUARTER IN ('2019Q4') THEN AMOUNT ELSE 0 END) AS Q4_2019
17
18 FROM CUSTOMER_NETSALE AS A
19 LEFT JOIN (
20 SELECT CT.CUSTOMER_ID, CT.CUSTOMER_NAME, CT.COMPANY_ID, CO.COMPANY_NAME
21 FROM CUSTOMER AS CT
22 LEFT JOIN (SELECT * FROM COMPANY) AS CO
23 ON CT.COMPANY_ID = CO.COMPANY_ID
24 ) AS B
25 ON A.CUSTOMER_ID = B.CUSTOMER_ID
26
```

The bottom of the interface shows the 'Output' tab with 'Action Output' selected. The output table has columns: #, Time, Action, Message, and Duration / Fetch. The 'No object selected' message is visible in the left sidebar.

# 1.3 - SQL Task - Solution Problem #03

## [1] Output:

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Script\_1 - Create tables Script\_2 - NetSale ID 1 Script\_3 - Pivot Table x

SCHMAS

Filter objects

- employees
- sales\_db
  - Tables
  - Views
  - Stored Procedures
- sys

Administration Schemas

Information: No object selected

Object Info Session

Script\_3 - Pivot Table

```
# 3. Create pivot that will show Customers as rows, quarters as columns and Net Sales Amounts as Values as table valued function, parameter is Company ID

SELECT
  B.COMPANY_ID,
  B.COMPANY_NAME,
  A.CUSTOMER_ID,
  B.CUSTOMER_NAME,
  SUM(CASE WHEN YEAR_QUARTER IN ('2018Q1') THEN AMOUNT ELSE 0 END) AS Q1_2018,
  SUM(CASE WHEN YEAR_QUARTER IN ('2018Q2') THEN AMOUNT ELSE 0 END) AS Q2_2018,
  SUM(CASE WHEN YEAR_QUARTER IN ('2018Q3') THEN AMOUNT ELSE 0 END) AS Q3_2018,
```

Result 1

COMPANY_ID	COMPANY_NAME	CUSTOMER_ID	CUSTOMER_NAME	Q1_2018	Q2_2018	Q3_2018	Q4_2018	Q1_2019	Q2_2019	Q3_2019	Q4_2019
8	Elit Etiam Laoreet LLP	8	Illana G. Murray	3075.00	3154.00	2358.00	831.00	2506.00	2084.00	6983.00	3920.00
11	Elit Etiam Laoreet LLP	11	Marvin R. Bauer	2501.00	1132.00	463.00	1253.00	5116.00	2360.00	5030.00	0.00
39	Elit Etiam Laoreet LLP	39	Stella D. Hernandez	1348.00	1505.00	2814.00	977.00	3326.00	2412.00	2409.00	1016.00
45	Elit Etiam Laoreet LLP	45	Hanna J. Klein	1804.00	1591.00	1978.00	2638.00	3117.00	2176.00	663.00	2788.00
2	Vulputate Incorporated	2	Baxter D. Norton	903.00	3185.00	3031.00	0.00	3871.00	4667.00	4793.00	7076.00
7	Vulputate Incorporated	7	Rae E. Bush	4108.00	0.00	0.00	833.00	0.00	2573.00	1084.00	0.00
15	Vulputate Incorporated	15	Winter W. Curtis	1920.00	855.00	690.00	2444.00	4877.00	0.00	535.00	719.00
16	Vulputate Incorporated	16	Serena C. Summers	0.00	0.00	6017.00	0.00	1354.00	0.00	4740.00	2451.00
41	Vulputate Incorporated	41	Tamekah S. Mayo	1464.00	3428.00	2684.00	7689.00	1973.00	2635.00	1668.00	0.00
46	Vulputate Incorporated	46	Germane N. Lott	2594.00	2286.00	3760.00	3180.00	3094.00	986.00	5010.00	1448.00
48	Vulputate Incorporated	48	Kenyon B. Cline	0.00	1588.00	1692.00	2707.00	966.00	0.00	0.00	1021.00
56	Vulputate Incorporated	56	Joelle V. Cardenas	706.00	4212.00	1883.00	1181.00	1935.00	3324.00	2541.00	4067.00
31	Ante Ipsum Foundation	31	Keane Q. Carr	0.00	1400.00	3699.00	2368.00	1611.00	954.00	3570.00	0.00
32	Ante Ipsum Foundation	32	Melinda O. Young	4300.00	492.00	3554.00	3592.00	0.00	1845.00	765.00	2988.00

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	19:08:19	SELECT B.COMPANY_ID, B.COMPANY_NAME, A.CUSTOMER_ID, B.CUSTOMER_NAME, SUM(CASE WH...	60 row(s) returned	0.031 sec / 0.000 sec



# PYTHON TASK

--

## 2 - Python Task

**Task description:** Design a VBA/Python automation which would create a report for the business according to the below details.

**Source Files**(There is an automatic daily process creating refreshed source files each day with updated data):

- *Source 1: BW\_CPFR\_OUTLET\_HU10.XLS (Sell Out Data)*
- *Source 2: BW\_SDS\_HU10.XLS (Net Sales Data)*

**Final Report File:**

- Top5NS\_Report.xlsm (to be created daily)

**Show data as values only (no formula reference to the datasheets) presenting the following:**

1. List the top 5 SELL OUT NS generating materials of each week available in the source file. Also show the %Ratio they represent of the total NS of that week. Please use 0 (zero) Average NS wherever the data is not available for the material. Key between Source 1 and Source 2 file is MATERIAL.
2. Which are the top 5 NS generating customers of each week in the last 3 closed weeks? (excluding the last - partial - week).

The automation should run daily in English environment.

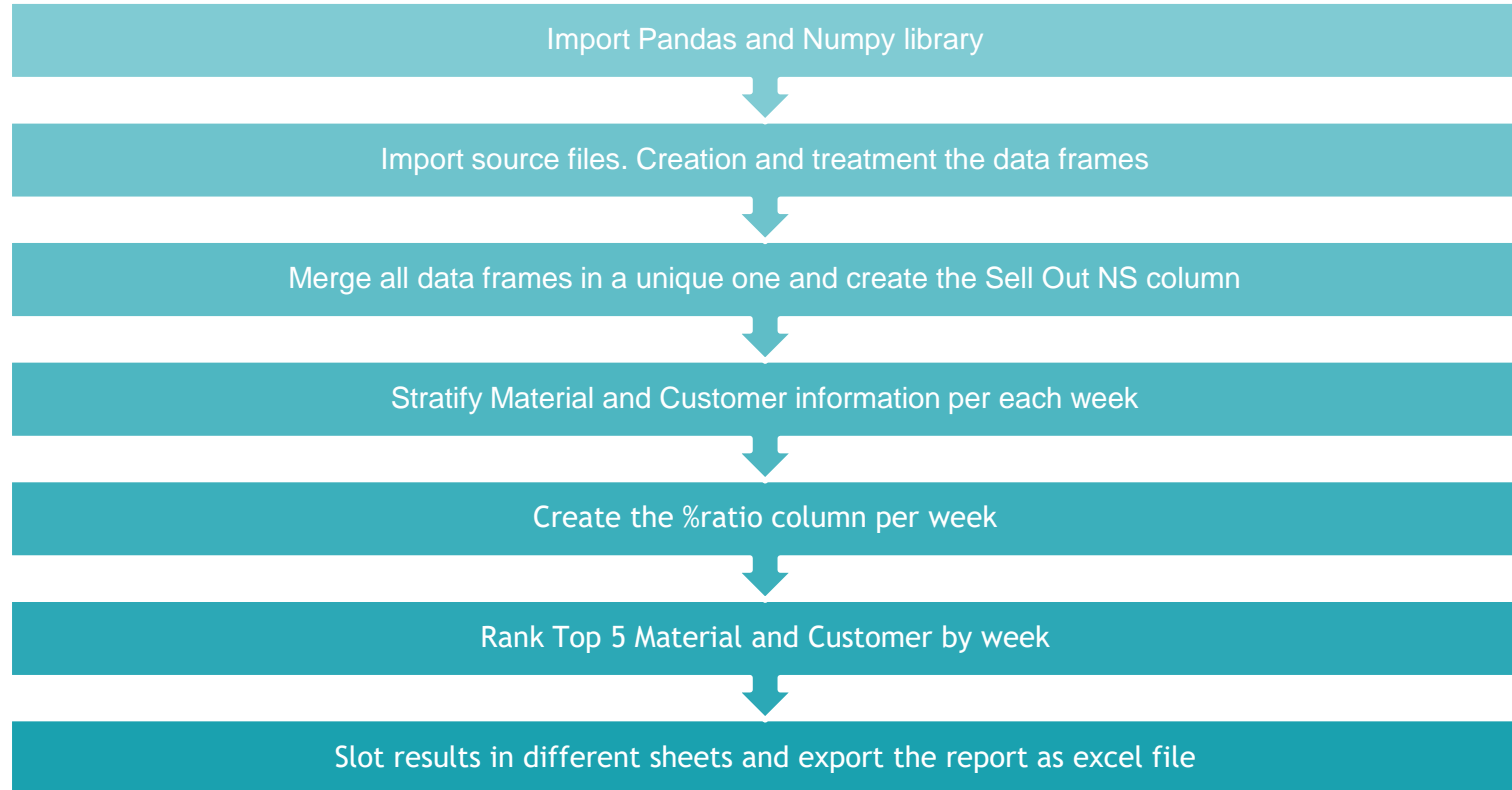
Materials = products

NS = Net Sales (P5 Net Sales EUR)

[Average NS] = this should be calculated from available NS data<sup>15</sup>



## 2.1 - Python Task - Solution Problem



## 2.1 - Python Task - Solution Problem

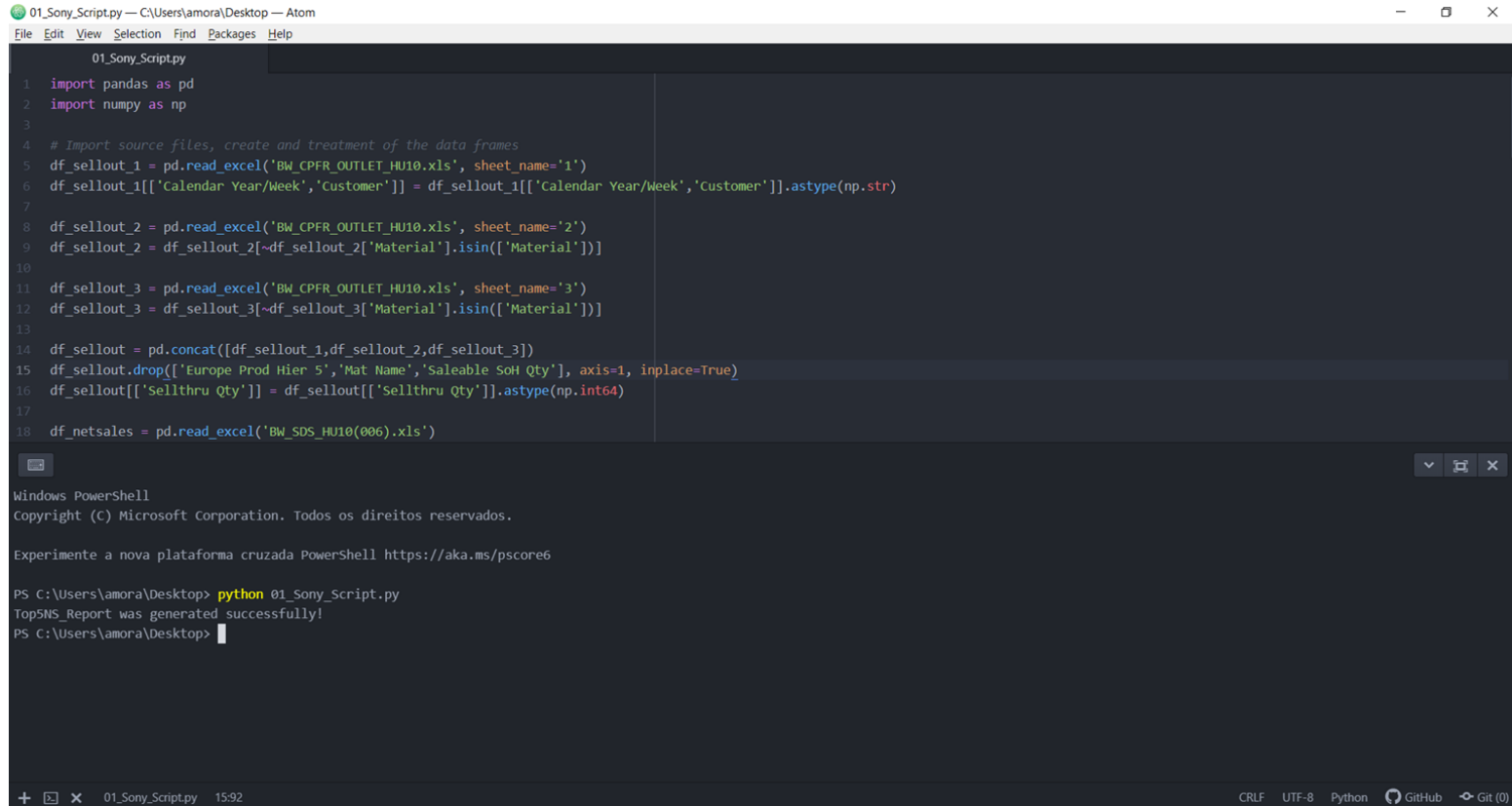
[0] Input:

```
01_Sony_Script.py — C:\Users\amora\Desktop — Atom
File Edit View Selection Find Packages Help

01_Sony_Script.py
1 import pandas as pd
2 import numpy as np
3
4 # Import source files, create and treatment of the data frames
5 df_sellout_1 = pd.read_excel('BW_CPFR_OUTLET_HU10.xls', sheet_name='1')
6 df_sellout_1[['Calendar Year/Week', 'Customer']] = df_sellout_1[['Calendar Year/Week', 'Customer']].astype(np.str)
7
8 df_sellout_2 = pd.read_excel('BW_CPFR_OUTLET_HU10.xls', sheet_name='2')
9 df_sellout_2 = df_sellout_2[~df_sellout_2['Material'].isin(['Material'])]
10
11 df_sellout_3 = pd.read_excel('BW_CPFR_OUTLET_HU10.xls', sheet_name='3')
12 df_sellout_3 = df_sellout_3[~df_sellout_3['Material'].isin(['Material'])]
13
14 df_sellout = pd.concat([df_sellout_1, df_sellout_2, df_sellout_3])
15 df_sellout.drop(['Europe Prod Hier 5', 'Mat Name', 'Saleable SoH Qty'], axis=1, inplace=True)
16 df_sellout[['Sellthru Qty']] = df_sellout[['Sellthru Qty']].astype(np.int64)
17
18 df_netsales = pd.read_excel('BW_SDS_HU10(006).xls')
19 df_netsales.columns = df_netsales.iloc[0]
20 df_netsales.drop(df_netsales.index[1], axis=0, inplace=True)
21 df_netsales.drop(['Calendar Year/Month', 'Europe Prod Hier 3', 'Sales Quantity', 'PS Net Sales LC', 'Material Name'], axis=1, inplace=True)
22 df_netsales.rename(columns={'PS Net Sales EUR': 'NetSales'}, inplace=True)
23 df_netsales[['NetSales']] = df_netsales[['NetSales']].astype(np.float64)
24
25 # Merge data frames into a unique and create the Sell Out NS column
26 df_source = (pd.merge(df_netsales, df_sellout, on=['Calendar Year/Week', 'Material', 'Customer'], how='outer'))
27 df_source = df_source.replace(np.nan, 0, int=True)
28 df_source = (df_source.groupby(['Calendar Year/Week', 'Customer', 'Material'], as_index=False)
29               .agg({'Sellthru Qty': 'sum', 'NetSales': 'mean'})
30               .rename(columns={'NetSales': 'NS Average'})))
31 n_column = df_source['Calendar Year/Week'].str.split('.', n=1, expand=True)
32 df_source['Week'] = n_column[0]
33 df_source['Year'] = n_column[1]
34 df_source.drop(['Calendar Year/Week'], axis=1, inplace=True)
35 df_source = df_source.reindex(columns=['Year', 'Week', 'Customer', 'Material', 'NS Average', 'Sellthru Qty'])
```

## 2.1 - Python Task - Solution Problem

### [1] Output:



The screenshot displays a code editor window titled "01\_Sony\_Script.py" with the following Python code:

```
1 import pandas as pd
2 import numpy as np
3
4 # Import source files, create and treatment of the data frames
5 df_sellout_1 = pd.read_excel('BW_CPFR_OUTLET_HU10.xls', sheet_name='1')
6 df_sellout_1[['Calendar Year/Week', 'Customer']] = df_sellout_1[['Calendar Year/Week', 'Customer']].astype(np.str)
7
8 df_sellout_2 = pd.read_excel('BW_CPFR_OUTLET_HU10.xls', sheet_name='2')
9 df_sellout_2 = df_sellout_2[~df_sellout_2['Material'].isin(['Material'])]
10
11 df_sellout_3 = pd.read_excel('BW_CPFR_OUTLET_HU10.xls', sheet_name='3')
12 df_sellout_3 = df_sellout_3[~df_sellout_3['Material'].isin(['Material'])]
13
14 df_sellout = pd.concat([df_sellout_1, df_sellout_2, df_sellout_3])
15 df_sellout.drop(['Europe Prod Hier 5', 'Mat Name', 'Saleable SoH Qty'], axis=1, inplace=True)
16 df_sellout[['Sellthru Qty']] = df_sellout[['Sellthru Qty']].astype(np.int64)
17
18 df_netsales = pd.read_excel('BW_SDS_HU10(006).xls')
```

Below the code editor is a Windows PowerShell terminal window. It shows the command to run the script and the successful output:

```
PS C:\Users\amora\Desktop> python 01_Sony_Script.py
TopSNS_Report was generated successfully!
PS C:\Users\amora\Desktop>
```

The status bar at the bottom of the application shows the file name "01\_Sony\_Script.py", the time "15:52", and various icons for encoding (CRLF, UTF-8), language (Python), and version control (GitHub, Git).

## 2.1 - Python Task - Solution Problem

[2] Output:

AutoSave Off Top5NS\_R

File Home Insert Page Layout Formulas Data Review View Help Search

Clipboard Font Alignment Number

Rank

Rank	Year	Week	Customer	NS Average	sellthru	Qt	Sell Out	NS	% Ratio
1	2013	39	290011488	616,07	8	4928,56	39,15309		
2	2013	39	290011488	157,29	15	2359,35	18,74297		
3	2013	39	290010501	691,71	3	2075,13	16,48509		
4	2013	39	290011488	137,87	11	1516,57	12,04782		
5	2013	39	290011488	69,32	15	1039,8	8,2603		

Material\_W39 Customer\_W39 Material\_W38 Customer\_W38 Material\_W37 Customer\_W37

AutoSave Off Top5NS\_Repo

File Home Insert Page Layout Formulas Data Review View Help Search

Clipboard Font Alignment Number

Rank

Rank	Year	Week	Material	NS Average	sellthru	Qt	Sell Out	NS	% Ratio
1	1	2013	39 MDREX37	616,07	8	4928,56	39,15309		
3	2	2013	39 SVF1521A	157,29	15	2359,35	18,74297		
4	3	2013	39 NEX3NLW	691,71	3	2075,13	16,48509		
5	4	2013	39 MDRE9LP	137,87	11	1516,57	12,04782		
6	5	2013	39 MDRZX10	69,32	15	1039,8	8,2603		

Material\_W39 Customer\_W39 Material\_W38 Customer\_W38 Material\_W37 Customer\_W37

**SONY**

make.believe