حل الطلب الأول:

مقدمة:

في البداية قمنا بمعرفة أو اكتشاف ما بداخل ملف (Excel) لكي نعرف ما يجب علينا فعله ولكي نعرف عدد الأعمدة وعدد الصفوف ولكي نعرف إذا كان هناك صفوف فارغة أو أعمدة فارغة أو هل يوجد تكرار في الصفوف وعلى حسب ماذا نريد من البيانات نستطيع معرفة أي الأعمدة هي المهمة وأيها أقل أهمية وأي عمود يؤثر على باقي الأعمدة وأي عمود يجب أن لا يكون فارغ وإلا سوف يكون هناك تضارب في البيانات فيما بعد عندما نقوم بتحليل البيانات، لذلك بناءاً على كل هذه التساؤلات يمكننا معرفة كيف سوف نتصرف عندما نبدأ بتنظيف البيانات.

الأدوات المستخدمة:

قمت باستخدام (Jupiter) باستخدام لغة بايثون.

المكتبات المستخدمة:

(Pandas)

بداية اكتشاف البيانات:

استيراد المكتبات:

```
# Import the library of Pandas.
import pandas as pd
```

حيث نقوم باستيراد المكتبة عبر أمر (import) ثم كتابة اسم المكتبة ولقد استخدمنا (as) لكي أستطيع وضع اختصار للمكتبة واسميته (pd) كما هو متعارف عليه، حيث استطيع استخدام المكتبة فيما بعد عن طريق اسم الاختصار (pd).

ثم قمنا بهذا الأمر لكي نقوم بقراءة ملف (Excel):

```
# Read The path of Excel file and save it in a variable called (df) --> it means DataFrame.
df = pd.read_excel(r"C:\Users\Classic\Desktop\data_all.xltx", engine="openpyxl")
```

حيث تم استخدام مكتبة (Pandas) ثم استخدام الدالة (read_excel) وثم تم وضع حرف (r) قبل علامتي التنصيص المزدوجة لكي يتم معرفة أن هذا مسار (path) للقراءة لأن داخل المسار يوجد (\) ولا نريد أن تحدث مشكلات لذلك تم وضع حرف (r) بمعنى للقراءة فقط ، ولقد وضعت متغير (df) كناية عن (DataFrame) حيث أقوم بحفظ الأمر داخل هذا المتغير لكي أستطيع استخدام هذا المتغير فيما بعد عندما أريده.

إن لاحقة ملف (Excel) هي (xltx) ولكي أحولها إلى لاحقة (csv) يجب استخدام هذا الأمر:

```
# Transform suffix from xltx to csv.
df.to_csv(r"C:\Users\Classic\Desktop\data_all.csv", index=False, encoding="utf-8-sig")
```

حيث بواسطة المتغير (df) قمت باستخدامه واستخدام عليه دالة التحويل(to_csv) وكذلك هنا قمت بوضع حرف (r) لكي أفهم البرنامج أن هذا السطر للقراءة، واستخدمت (index = False) لكي لا يظهر عامود فهرسة للجدول أو ل (Dataframe) حيث أن (DataFrame) هو عبارة عن أكثر من عمود وأكثر من صف، واستخدمت (encoding="utf-8-sig") لكي لا تحدث مشاكل إذا كان هناك بيانات مكتوبة باللغة العربية.

لكي نعرف كم عدد السطور والأعمدة داخل الملف استخدمت الأمر التالي:

```
# Know the numbers of rows and columns.
df.shape
```

```
(259917, 10)
```

حيث من خلال استخدام المتغير (df) ثم الأمر (shape) ظهرت لنا نتيجة بعدد الأسطر بما فيها سطر العناوين وعدد الأعمدة حيث الخانة الأولى هي لعدد الصفوف والخانة الثانية هي لعدد الأعمدة.

ولكي نعرف أسماء الأعمدة نستخدم الأمر التالي:

```
df.columns
```

```
Index(['account_id', 'SourceSystem', 'activity_date', 'who_id',
       'opportunity_id', 'opportunity_stage', 'is_lead', 'types', 'Country',
       'solution'],
      dtype='object')
```

وإذا أردنا معرفة تفاصيل أكثر عن البيانات الموجودة داخل الملف نستخدم الأمر التالي:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 259917 entries, 0 to 259916
Data columns (total 10 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   account_id         259917 non-null  object
 1   SourceSystem       259917 non-null  object
 2   activity_date      205516 non-null  datetime64[ns]
 3   who_id             238294 non-null  object
 4   opportunity_id     45849 non-null   object
 5   opportunity_stage  259901 non-null  object
 6   is_lead            259901 non-null  float64
 7   types              259901 non-null  object
 8   Country            259730 non-null  object
 9   solution           259901 non-null  object
dtypes: datetime64[ns](1), float64(1), object(8)
memory usage: 19.8+ MB
```

حيث تظهر النتيجة عدد الصفوف وعدد الأعمدة وأسماء الأعمدة وأنواع البيانات و عدد الصفوف التي ليست فارغة، ومن خلال هذا الأمر نلاحظ عدة نقاط تفيدنا عند إجراء تنظيف البيانات ومنها مثلا نلاحظ في العمود (account_id) لا يحتوي على صفوف فارغة وهو يعتبر أهم عمود في الجدول لأنه يحدد كل عميل (customer).

يوجد أمر من خلاله أستطيع معرفة إذا كان هناك صفوف فارغة وكم عدد الصفوف الفارغة في كل عمود:

```
df.isna().sum()
```

```
account_id              0
SourceSystem            0
activity_date       54401
who_id              21623
opportunity_id     214068
opportunity_stage       16
is_lead                 16
types                   16
Country                187
solution                16
dtype: int64
```

حيث إن الأمر (isna()) يعيد (True or Flase) كنتيجة ولكن عندما استخدمنا معها الأمر (sum) استطعنا جمع عدد الصفوف الفارغة في كل عمود.

بدء تنظيف البيانات:

العمود (account_id):

هذا العمود هو الذي يعبّر عن العملاء لذلك هذا العمود جداً مهم،لا يحتوي على بيانات فارغة واكتشفت هذه المعلومة من خلال الأوامر السابقة ، ولمعرفة باقي التفاصيل إذا كان هذا العمود يحتوي على نصوص كبيرة وغير متناسقة او صغيرة ولكي نعرف كذلك ماذا يوجد كقيم فريدة داخل هذا العمود قمت بعمل عدة أوامر ساعدتني على معرفة تساؤلاتي،

حيث من خلال الأمر التالي:

```
df['account_id'].unique()
```

```
array(['0010L00001hVmFhQAK', '0010L00001hVxd6QAC', '0010L00001hVyJQQA0',
       ..., '001b0000040UoFUAA0', '001b00000424wsfAAA',
       '001b000004275jgAAA'], dtype=object)
```

استطعت معرفة محتويات العمود.

ومن خلال هذا الأمر التالي:

```
lengths = df['account_id'].astype(str).str.len()
([lengths.min(), lengths.median(), lengths.max()])
```

```
[18, 18.0, 18]
```

استطعت معرفة عدد قيم او طول النصوص الكبيرة والصغيرة والمتوسطة، و وجدت أن جميعها متساوي الطول، لذلك اعتقد أن هذا العمود لا يحتاج إلى أي تنظيف آخر.

العمود (SourceSystem):

```
df['SourceSystem'].unique()
```

```
array(['SFDC_US', 'SFDC_ROW', 'SFDC_BEANWORKS', 'SFDC_CXM', 'SFDC_PP',
       'SFDC_GLOBAL'], dtype=object)
```

```
df['SourceSystem'].isna().sum()
```

```
np.int64(0)
```

عندما قمت باكتشاف ما هي البيانات الفريدة الموجودة داخله وهل داخل هذا العمود يوجد أي صف فارغ، وجدت أنه لا يوجد بيانات فارغة وليس هناك حاجة لأي تنظيف للبيانات داخل هذا العمود.

العمود (activity_date):

لكي أعرف البيانات الموجودة داخل هذا العمود قمت بتنفيذ هذا الأمر:

```
df['activity_date'].unique()
```

```
<DatetimeArray>
['2022-07-25 00:00:00', '2023-02-08 00:00:00', '2023-02-14 00:00:00',
 '2023-02-20 00:00:00', '2023-03-16 00:00:00', '2023-03-22 00:00:00',
 '2023-12-04 00:00:00', '2024-02-12 00:00:00', '2024-02-22 00:00:00',
 '2024-04-01 00:00:00',
 ...
 '2024-09-26 22:00:00', '2024-10-14 22:00:00', '2024-11-25 23:00:00',
 '2023-04-28 00:00:00', '2025-04-16 22:00:00', '2025-05-18 22:00:00',
 '2024-10-10 22:00:00', '2024-10-20 22:00:00', '2024-11-27 23:00:00',
 '2025-02-04 23:00:00']
Length: 4719, dtype: datetime64[ns]
```

إن هذا العمود مهم جداً لأن رحلة العميل الأقصر تتطلب وجود تاريخ محدد لذلك وجود الصفوف الفارغة تجعل البيانات متضاربة ولا يمكننا إجراء تحليل دقيق كذلك وجود أي أخطاء من تنسيق وأمور أخرى في هذا العمود سوف يضر في إجراء عملية التحليل لذلك بدايةً سوف أقوم بإجراء عدة خطوات للتأكد من سلامة وتنظيف هذا العمود المهم.

أولاً سوف أنفذ الأمر التالي لمعرفة عدد الصفوف الفارغة في هذا العمود:

```
df['activity_date'].isna().sum()
```

```
np.int64(54401)
```

ولكن لن أقوم بإجراء أي حذف للصفوف الفارغة حالياً لكي لا يحدث تضارب أو أخطاء كثيرة حيث مبدئياً يجب تحديد الأولوية وهي تنظيف الأعمدة جميعها ومن ثم بعد الانتهاء أستطيع أن أحذف الصفوف الفارغة بدون حدوث أي مشكلات مثل حذف صفوف مهمة وضياع الفائدة من البيانات في وقت التحليل، طبعاً وصلت إلى هذه النتيجة عندما قمت بتنفيذ الحذف للصفوف في البداية اكتشف ان عدد الصفوف المحذوفة أكثر من عدد الصفوف المحذوفة بعد إجراء عملية التنظيف لكامل الأعمدة لذلك توصلت إلى فكرة تحديد الأولويات لكي أحافظ على البيانات المهمة قدر المستطاع،

تكملة للخطوات على هذا العمود يجب أن أقوم بتحويل نوع العمود إلى تاريخ لكي لا يحدث مشكلات فيما بعد ويكون هذا العمود متناسق من حيث النوع، لذلك من خلال الأمر التالي نفذت عملية التحويل:

```
df['activity_date'] = pd.to_datetime(df['activity_date'], errors = 'coerce')
```

وهنا في نهاية الأمر وضعت قيمة (errors = 'coerce') لكي إذا حدث أي أخطاء تظهر هذه القيمة المحددة،

وعندما أقوم بتنفيذ الأمر التالي أستطيع رؤية النتيجة بعد تغيير نوع البيانات:

```
df['activity_date'].unique()
```

```
array([datetime.date(2022, 7, 25), datetime.date(2023, 2, 8),
       datetime.date(2023, 2, 14), ..., datetime.date(2018, 5, 28),
       datetime.date(2016, 7, 29), datetime.date(2023, 4, 28)],
      dtype=object)
```

حيث تم تحويل التاريخ إلى وقت بدلا من ظهوره ك(string) ومن هنا أجد أنه بعد الانتهاء من التنسيق يجب علي القيام بترتيب هذا العمود مع العمود الذي يعبّر عن العميل (account_id) لكي تظهر كافة الاعمدة على أساس هذا الترتيب لكي يتم كذلك معرفة كل نشاطات العميل مع تاريخ كل نشاط ويتم ذلك عن طريق الأمر التالي:

```
# Arrange the columns by account_id and activity_date columns.
df = df.sort_values(by=['account_id', 'activity_date'], ascending = True)
```

بعد تنفيذ الأمر السابق سوف نرى النتيجة كالتالي:

```
# Show DataFrame after arranging the columns.
df
```

	account_id	SourceSystem	activity_date	who_id	opportunity_id	opportunity_stage	is_lead	types	Country	solution
0	0010L00001hVmFhQAK	SFDC_US	2022-07-25	0030L00001vlbHLQAY	NaN	no_opp	1.0	Follow Up	US	MRS
1	0010L00001hVmFhQAK	SFDC_US	2023-02-08	0034X00002xZlQtQAK	NaN	no_opp	1.0	Inbound Call	US	MRS
2	0010L00001hVmFhQAK	SFDC_US	2023-02-14	0030L00001vlbHLQAY	NaN	no_opp	1.0	Inbound Call	US	MRS
3	0010L00001hVmFhQAK	SFDC_US	2023-02-20	0030L00001vlbHLQAY	NaN	no_opp	2.0	Inbound Call	US	MRS
4	0010L00001hVmFhQAK	SFDC_US	2023-03-16	0034X00003GOUrFQAX	NaN	no_opp	1.0	Inbound Call	US	MRS
...	...	...	...	...	...	...	...	...	...	...
259912	001b00000424wsfAAA	SFDC_ROW	2025-05-15	0030X00002gQd41QAC	NaN	no_opp	1.0	Call	FR	MRS
259913	001b00000424wsfAAA	SFDC_ROW	2025-05-15	0030X00002gQd41QAC	NaN	no_opp	2.0	Call	FR	MRS
259914	001b00000424wsfAAA	SFDC_ROW	2025-07-03	003SZ00000bQOHfYAO	NaN	no_opp	1.0	Email	FR	MRS
259915	001b000004275jgAAA	SFDC_ROW	2022-11-16	0036700003ycNNIAA2	NaN	no_opp	1.0	Call	FR	MRS
259916	001b000004275jgAAA	SFDC_ROW	2022-11-17	003b00000206ARZAA2	NaN	no_opp	1.0	Call	FR	MRS

259917 rows × 10 columns

نلاحظ أن كل عميل يتم عرض نشاطاته كافةً مرتبة ترتيباً تصاعدياً حسب عمود (activity_date).

العمود (Who_id):

في هذا العمود أستطيع معرفة البيانات الموجودة داخله وكم عدد الصفوف الفارغة عبر الأمرين التاليين:

```
df['who_id'].unique()
```

```
array(['0030L00001vlbHLQAY', '0034X00002xZlQtQAK', '0034X00003GOUrFQAX',
       ..., '003SZ00000bQOHfYAO', '0036700003ycNNIAA2',
       '003b00000206ARZAA2'], dtype=object)
```

```
df['who_id'].isna().sum()
```

```
np.int64(21623)
```

في هذا العمود وجود الصفوف الفارغة ليس مشكلة لأن هذا العمود أقل أهمية مقارنةً بالأعمدة الأخرى المهمة، لذلك وجود بيانات فارغة لن يحدث أي مشكلة فيما بعد.

العمود (opportunity_id):

كذلك هذا العمود لا يختلف كثيراً من حيث الأهمية عن عمود (who_id) وكذلك بهذا العمود قمت بتنفيذ الأمرين التاليين:

```
df['opportunity_id'].unique()
```

```
array([nan, '0060y000019Fk7PAAS', '0060y00001F66cvAAB', ...,
       '0060X00000ei1t9QAA', '0066700000znDK2AAM', '0066700000wUXg9AAG'],
      dtype=object)
```

```
df['opportunity_id'].isna().sum()
```

```
np.int64(214068)
```

العمود (opportunity_stage):

هذا العمود مهم لما نريد عمله بعد عملية التنظيف لذلك مبدئياً سوف أقوم بمعرفة البيانات الفريدة لهذا العمود عبر الأمر التالي:

```
df['opportunity_stage'].unique()
```

```
array(['no_opp', 'Won', 'Lost', 'Diagnose', 'Access', 'Discovery',
       'Negotiate', 'Delivery', 'Design', '01 - New',
       'SDS - 05 - Negotiating', 'Implemented', '2 - Validation',
       '8 - Disqualified', 'Stage 2: Qualified Renewal',
       '1 - Qualificationâ€‹', '3 - Design',
       'Stage 5: Procurement/Negotiation',
       'Stage 8: â,¬0 Contract Change', '4 - Negotiate', nan,
       '5 - Procurement', '05 - Negotiating', '03 - Qualified',
       '04 - Offer'], dtype=object)
```

من النتيجة لاحظت وجود قيم فارغة و عدم وجود تنسيق لذلك سوف أقوم باستبدال القيم الفارغة ب (UNKOWN) وكذلك سوف أجعل جميع البيانات موحدة بحروف كبيرة وخالية من الفراغات الزائدة من خلال الأمرين التاليين:

```
df['opportunity_stage'] = df['opportunity_stage'].replace(['nan', None, ''], 'UNKNOWN')
df['opportunity_stage'] = df['opportunity_stage'].astype(str).str.strip().str.upper()
```

ولكي نرى النتيجة نستخدم الأمر التالي:

```
df['opportunity_stage'].unique()
```

```
array(['NO_OPP', 'WON', 'LOST', 'DIAGNOSE', 'ACCESS', 'DISCOVERY',
       'NEGOTIATE', 'DELIVERY', 'DESIGN', '01 - NEW',
       'SDS - 05 - NEGOTIATING', 'IMPLEMENTED', '2 - VALIDATION',
       '8 - DISQUALIFIED', 'STAGE 2: QUALIFIED RENEWAL',
       '1 - QUALIFICATIONÂ€‹', '3 - DESIGN',
       'STAGE 5: PROCUREMENT/NEGOTIATION',
       'STAGE 8: Â,¬0 CONTRACT CHANGE', '4 - NEGOTIATE', 'UNKNOWN',
       '5 - PROCUREMENT', '05 - NEGOTIATING', '03 - QUALIFIED',
       '04 - OFFER'], dtype=object)
```

حيث تم توحيد شكل أو حجم البيانات ونلاحظ مشكلة جمالية بسبب خطئ بالترميز عند قيمتين من البيانات الفريدة ألا وهما:

```
'1 - QUALIFICATIONÂ€‹'

'STAGE 8: Â,¬0 CONTRACT CHANGE'
```

حيث نستطيع حل هذه المشكلة الجمالية عبر الأمر التالي:

```
df['opportunity_stage'] = df['opportunity_stage'].replace({
    '1 - QUALIFICATIONÂ€‹': '1 - QUALIFICATION',
    'STAGE 8: Â,¬0 CONTRACT CHANGE': 'STAGE 8: CONTRACT CHANGE'
})
```

والنتيجة النهائية سوف نراها عبر الأمر التالي:

```
df['opportunity_stage'].unique()
```

```
array(['NO_OPP', 'WON', 'LOST', 'DIAGNOSE', 'ACCESS', 'DISCOVERY',
       'NEGOTIATE', 'DELIVERY', 'DESIGN', '01 - NEW',
       'SDS - 05 - NEGOTIATING', 'IMPLEMENTED', '2 - VALIDATION',
       '8 - DISQUALIFIED', 'STAGE 2: QUALIFIED RENEWAL',
       '1 - QUALIFICATION', '3 - DESIGN',
       'STAGE 5: PROCUREMENT/NEGOTIATION', 'STAGE 8: CONTRACT CHANGE',
       '4 - NEGOTIATE', 'UNKNOWN', '5 - PROCUREMENT', '05 - NEGOTIATING',
       '03 - QUALIFIED', '04 - OFFER'], dtype=object)
```

وبهذا نكون قد نظفنا العمود بدون أن نقوم بحذف شيء لأن هذا المود مهم جدا فيما بعد.

العمود (is_lead):

```
df['is_lead'].unique()
```

```
array([ 1.,  2., nan])
```

نجد أنه يحتوي على أرقام وعلى صفوف فارغة لذلك ممكن أن نستبدل الصفوف الفارغة بقيمة (0) عبر الأمر التالي:

```
df['is_lead'] = df['is_lead'].fillna(0)
```

حيث من خلال هذا الأمر وكأنني أقول للكود قم بملئ أي بيانات فارغة بهذا العمود بقيمة (0)، ونرى النتيجة النهائية من خلال الأمر التالي:

```
df['is_lead'].isna().sum()
```

```
np.int64(0)
```

حيث لم يعد هناك وجود لبيانات فارغة داخل هذا العمود.

العمود (types):

```
df['types'].unique()
```

```
array(['Follow Up', 'Inbound Call', 'Meeting', 'Review', 'Email',
       'On-Site', 'Demo', 'Call', 'Discovery', 'Outbound Call',
       '2nd Appointment', '1st Appointment', nan], dtype=object)
```

نلاحظ أنه لا يوجد مشكلة بتنسيق البيانات ولكن هناك بيانات فارغة وكذلك هناك قيمتين يبدأن برقم لذلك ممكن أن نغيرهم جمالياً عبر الأمر التالي:

```
df['types'] = df['types'].replace({
    '1st Appointment':'First Appointment',
    '2nd Appointment': 'Second Appointment'
})
```

وتصبح النتيجة كالتالي:

```
df['types'].unique()
```

```
array(['Follow Up', 'Inbound Call', 'Meeting', 'Review', 'Email',
       'On-Site', 'Demo', 'Call', 'Discovery', 'Outbound Call',
       'Second Appointment', 'First Appointment', nan], dtype=object)
```

وبالنسبة للصفوف الفارغة فسوف أتعامل معها بعد الانتهاء من تنظيف كل الأعمدة.

العمود (Country):

```
df['Country'].unique()
```

```
array(['US', 'FR', 'Martinique', 'Guyane Fran¡§aise', nan, 'CA',
       'Thailand', 'Brazil', 'DE', 'Bulgaria', 'Ireland', 'Nicaragua',
       'Mexico', 'Spain', 'BE', 'Italy', 'Netherlands', 'Switzerland',
       'Poland', 'UK', 'Slovakia', 'Czech Republic', 'Austria',
       'Hong Kong', 'China', 'Japan', 'Taiwan', 'Australia',
       'Cayman Islands', 'India', 'New Zealand', 'Korea, Republic of',
       'Saudi Arabia', 'Malaysia', 'United Arab Emirates', 'Hungary',
       'Singapore', 'Latvia', 'Argentina', 'Israel', 'Turkey', 'Peru',
       'South Africa', 'El Salvador', '[NULL]', 'NL', 'AT', 'IT', 'DK',
       'LU', 'CH', 'IE', 'Denmark', 'R¡©union', 'Guadeloupe'],
      dtype=object)
```

نجد أن داخله بيانات مكررة حيث هناك أسماء دول مذكورة نفسها بصيغتين مثل دولة (Italy) مذكورة كذلك هكذا (IT) وغيرها من الدول وكذلك نجد وجود قيم فارغة وقيم ([NULL]) وقيم فيها رموز لذلك قمت بتنظيف هذه القيم وتوحيد الصيغ عبر الأمر التالي:

```
df['Country'] = df['Country'].replace(
    {
        '[NULL]':'Unknown',
         None: 'Unknown',
        'Guyane Fran¡§aise':'French Guiana',
        'R¡©union':'Reunion',
        'NL':'Netherlands',
        'AT':'Austria',
        'CH':'Switzerland',
        'IT':'Italy',
        'DK':'Denmark',
        'IE':'Ireland',
        'US': 'United States',
        'FR': 'France',
        'CA': 'Canada',
        'DE': 'Germany',
        'BE': 'Belgium',
        'UK': 'United Kingdom',
        'LU': 'Luxembourg',
        'Israel':'Palestine'
    }
)
```

ونرى النتيجة من خلال الأمر التالي:

```
df['Country'].unique()
```

```
array(['United States', 'France', 'Martinique', 'French Guiana',
       'Unknown', 'Canada', 'Thailand', 'Brazil', 'Germany', 'Bulgaria',
       'Ireland', 'Nicaragua', 'Mexico', 'Spain', 'Belgium', 'Italy',
       'Netherlands', 'Switzerland', 'Poland', 'United Kingdom',
       'Slovakia', 'Czech Republic', 'Austria', 'Hong Kong', 'China',
       'Japan', 'Taiwan', 'Australia', 'Cayman Islands', 'India',
       'New Zealand', 'Korea, Republic of', 'Saudi Arabia', 'Malaysia',
       'United Arab Emirates', 'Hungary', 'Singapore', 'Latvia',
       'Argentina', 'Palestine', 'Turkey', 'Peru', 'South Africa',
       'El Salvador', 'Denmark', 'Luxembourg', 'Reunion', 'Guadeloupe'],
      dtype=object)
```

أستطيع معرفة عدد الأعمدة الفارغة في هذا العمود التي قمت بتعويضها بقيمة (Unknown) باستخدام الأمر التالي نستطيع معرفة عدد الأعمدة الفارغة :

```
df[df['Country'] == 'Unknown'].shape[0]
```

```
301
```

إن عمود (Country) مهم جدا وبدونه لا أستطيع إنجاز عدة مهمات لذلك يجب حذف كل الصفوف الفارغة داخله لكن سأقوم بهذا الأمر بعد الانتهاء من تنظيف آخر عمود.

العمود (solution):

```
df['solution'].unique()
```

```
array(['MRS', 'Digital', 'PLS', nan], dtype=object)
```

إن بياناته لا تحتاج إلى تنظيف لأنها منسقة بشكل جيد ولكن داخله يوجد صفوف فارغة ، ولمعرفة عدد الصفوف الفارغة نستخدم الأمر التالي:

```
df['solution'].isna().sum()
```

```
np.int64(16)
```

حذف الصفوف المكررة:

أولاً حذف الصفوف المكررة لكل الأعمدة:

```
df = df.drop_duplicates(subset = ['account_id', 'SourceSystem', 'activity_date', 'who_id',
       'opportunity_id', 'opportunity_stage', 'is_lead', 'types', 'Country',
       'solution'], keep = 'first')
```

معرفة نتيجة الحذف كم أصبحت أعداد الصفوف الفارغة في أهم الأعمدة التي يؤثر داخلها وجود صفوف فارغة، وذلك عبر الأوامر التالية:

```
df['solution'].isna().sum()
```

```
np.int64(14)
```

```
df['types'].isna().sum()
```

```
np.int64(14)
```

```
df['activity_date'].isna().sum()
```

```
np.int64(12171)
```

```
df[df['Country'] == 'Unknown'].shape[0]
```

```
298
```

ثانياً حذف الصفوف المكررة لأهم الأعمدة:

```
df = df.drop_duplicates(subset = ['account_id', 'activity_date', 'types', 'Country',
       'solution'], keep = 'first')
```

معرفة كم أصبحت أعداد الصفوف المكررة لأهم الأعمدة:

```
df['solution'].isna().sum()
```

```
np.int64(14)
```

```
df['types'].isna().sum()
```

```
np.int64(14)
```

```
df['activity_date'].isna().sum()
```

```
np.int64(753)
```

```
df[df['Country'] == 'Unknown'].shape[0]
```

```
275
```

ثالثاً حذف الصفوف المكررة للأعمدة المهمة هذه:

```
df = df.drop_duplicates(subset = ['account_id', 'activity_date', 'types'], keep = 'first')
```

ويظهر بالنتيجة أن قيمة الصفوف الفارغة لعمود (activity_date) قد تغير ونقص كما نرى:

```
df['activity_date'].isna().sum()
```

```
np.int64(737)
```

و أخيراً حذف الصفوف الفارغة لعمود (activity_date):

```
df = df.dropna(subset=['activity_date'])
```

ثم نقوم بحذف الصفوف الفارغة للعمود (Country):

```
df = df.drop(df[df['Country'] == 'Unknown'].index)
```

وبعد حذف كل الصفوف الفارغة في هذا العمود وجدت أن كلا عامودي (solution) و (types) لم تعد تحتوي على صفوف فارغة:

```
df['solution'].isna().sum()
```

```
np.int64(0)
```

```
df['types'].isna().sum()
```

```
np.int64(0)
```

وفي النهاية نجد أن عدد الصفوف قد تغير بعد إجراء كافة عمليات التنظيف على هذا الملف:

```
# Show the numbers of rows after cleaning data.
df.shape
```

```
(154638, 10)
```

عدد الصفوف المحذوفة:

حيث تم حذف (105278) صف.

ولكي أستطيع رؤية ملف الاكسيل بعد التنظيف استخدم الأمر التالي:

```
# Transform this file into Excel file
df.to_excel("data_clean.xlsx", index= False)
```