Deployment Simplified

Problem/pain: scrolling through many many emails to find the right email with the related attachment (data) for a Power Bi report. The file is sent over to us as a zipped file. The manual act of locating the email, downloading it, and extracting the excel file out of the zipped folder is not difficult or too timely. However, these small steps do rack up time. If this manual process takes let's say ~10 minutes a day then it sucks up one hour of business time weekly.

Solution:

- create a special folder in your inbox
- write a rule to automatically isolate the emails that send over the Deployment data to the special folder
- write a python script that can open the email message from the special folder + download the attachment + unzips the file + save the file in the destination path

Benefit:

- streamlined
- systematic
- accurate
- saves business time

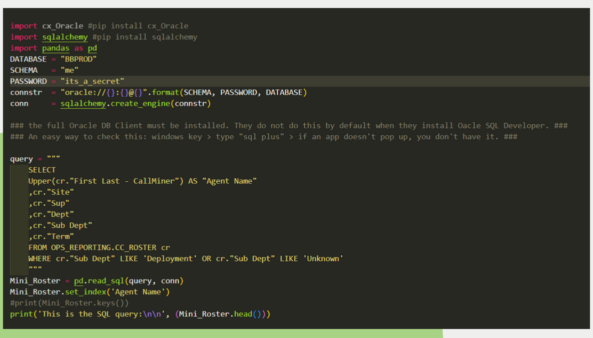
Scroll to see the phase two!

Deployment Simplified

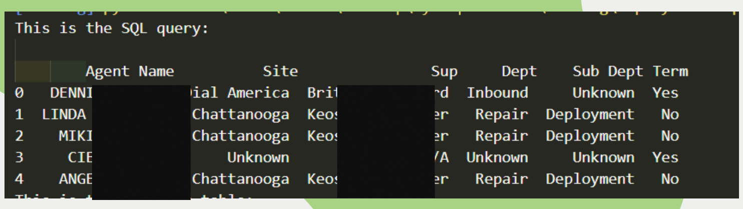
This is the phase two of a new ETL process for the Deployment Report. Currently the report is inefficient. It is slow as the data fed into the report is heavy. Also the cleaning process is done in the Power Query of the Power Bi . To speed up the reporting, we need to automate the data wrangling process.

Step 1

Query a mini roster that will later drive the merge with another table. The report is only concerned with a sub group of the whole contact center.

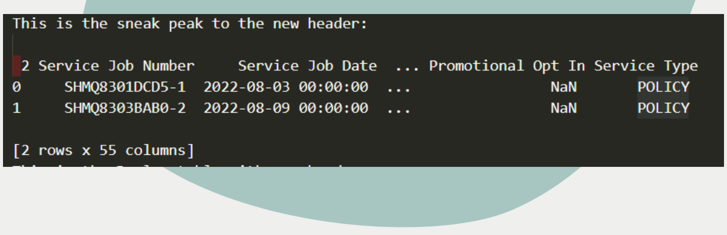


```
import cx_Oracle #pip install cx_Oracle
import sqlalchemy #pip install sqlalchemy
import pandas as pd
DATABASE = "BBPROD"
SCHEMA   = "me"
PASSWORD = "its_a_secret"
connstr  = "oracle://{}:{}@{}".format(SCHEMA, PASSWORD, DATABASE)
conn     = sqlalchemy.create_engine(connstr)

### the full Oracle DB Client must be installed. They do not do this by default when they install Oacle SQL Developer. ###
### An easy way to check this: windows key > type "sql plus" > if an app doesn't pop up, you don't have it. ###

query = """
    SELECT
    Upper(cr."First Last - CallMiner") AS "Agent Name"
    ,cr."Site"
    ,cr."Sup"
    ,cr."Dept"
    ,cr."Sub Dept"
    ,cr."Term"
    FROM OPS_REPORTING.CC_ROSTER cr
    WHERE cr."Sub Dept" LIKE 'Deployment' OR cr."Sub Dept" LIKE 'Unknown'
    """
Mini_Roster = pd.read_sql(query, conn)
Mini_Roster.set_index('Agent Name')
#print(Mini_Roster.keys())
print('This is the SQL query:\n\n', (Mini_Roster.head()))
```



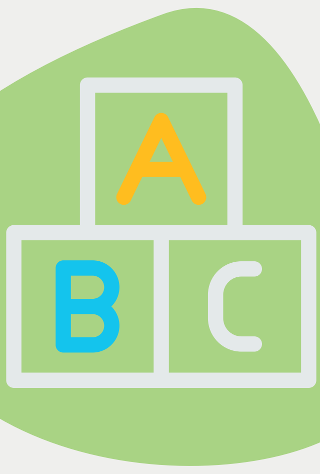
```
This is the SQL query:

       Agent Name           Site                 Sup      Dept    Sub Dept Term
0   DENNI[redacted]ial America   Brit[redacted]rd  Inbound     Unknown  Yes
1  LINDA            Chattanooga   Keos          er   Repair  Deployment   No
2    MIKI           Chattanooga   Keos          er   Repair  Deployment   No
3     CIE               Unknown                 /A  Unknown     Unknown  Yes
4    ANGE           Chattanooga   Keos          er   Repair  Deployment   No
```

Step 2



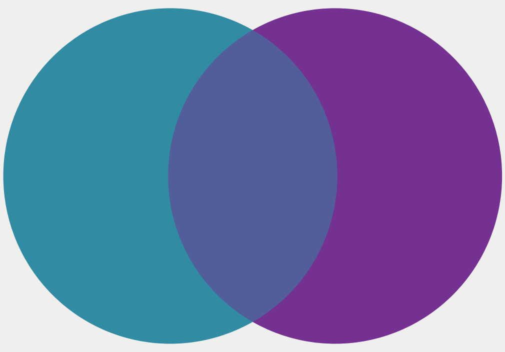
```
This is the sneak peak to the new header:

2 Service Job Number      Service Job Date  ... Promotional Opt In Service Type
0      SHMQ8301DCD5-1  2022-08-03 00:00:00  ...                NaN       POLICY
1      SHMQ8303BAB0-2  2022-08-09 00:00:00  ...                NaN       POLICY

[2 rows x 55 columns]
```

Import the unzipped file as a result of the first step in the Deployment Simplification. Then delete blank and irrelevant rows. Lastly resetting the header row to the correct row.



Pre-merging

Prior to merging I need to identify a relationship between the roster and the unzipped file which is the *agents' names*. In order for a successful join I'll need to standardize the names by using the .upper() for both tables.

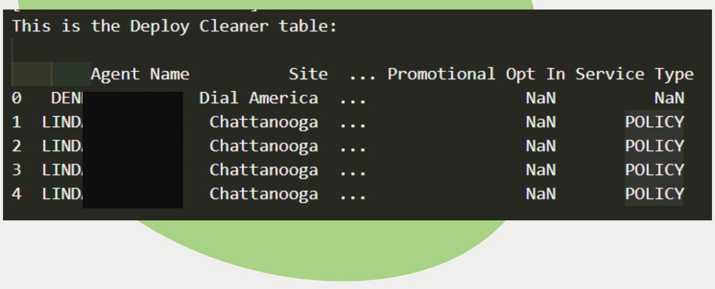


Merging choice

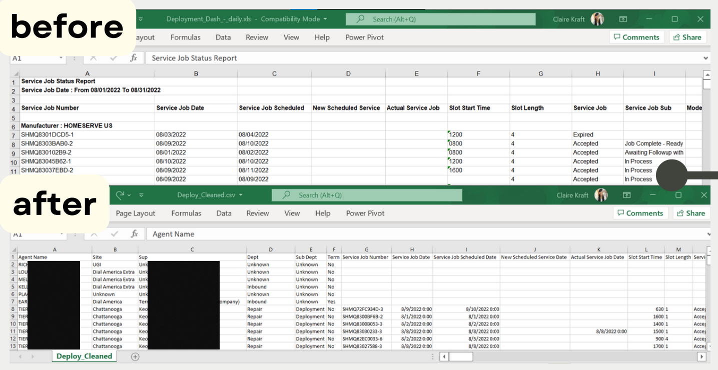
Left join on the mini roster. Again the report only cares about the Deployment agents. Unknown agents are kept as well as an error check mechanism. It's an indicator to the leadership whether a unknown agent's profile ought to be updated in the roster.

Step 3

Merge mini roster with the unzipped file. This new table will be significantly cleaner and smaller than the original table that fed the Deployment report.



```
This is the Deploy Cleaner table:

      Agent Name          Site  ... Promotional Opt In Service Type
0   DEN[redacted] Dial America  ...                NaN          NaN
1  LIND          Chattanooga  ...                NaN       POLICY
2  LIND          Chattanooga  ...                NaN       POLICY
3  LIND          Chattanooga  ...                NaN       POLICY
4  LIND          Chattanooga  ...                NaN       POLICY
```



before

Service Job Status Report

	A	B	C	D	E	F	G	H	I	
1	Service Job Status Report									
2	Service Job Date : From 08/01/2022 To 08/31/2022									
3										
4	Service Job Number	Service Job Date	Service Job Scheduled	New Scheduled Service	Actual Service Job	Slot Start Time	Slot Length	Service Job	Service Job Sub	Mode
5										
6	Manufacturer : HOMESERVE US									
7	SHMQ8301DCD5-1	08/03/2022	08/04/2022			1200	4	Expired		
8	SHMQ8303BAB0-2	08/09/2022	08/10/2022			0800	4	Accepted	Job Complete - Ready	
9	SHMQ830102B9-2	08/01/2022	08/02/2022			0800	4	Accepted	Awaiting Followup with	
10	SHMQ83045B62-1	08/10/2022	08/10/2022			1200	4	Accepted	In Process	
11	SHMQ83037EBD-2	08/09/2022	08/11/2022			1600	4	Accepted	In Process	
		08/09/2022	08/09/2022				4	Accepted	In Process	

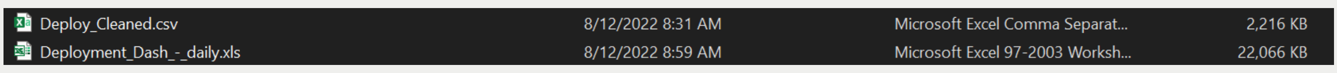
after

Agent Name

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	Agent Name	Site	Sup	Dept	Sub Dept	Term	Service Job Number	Service Job Date	Service Job Scheduled Date	New Scheduled Service Date	Actual Service Job Date	Slot Start Time	Slot Length	Servi
2	RIC[redacted]	UGI	Unk[redacted]	Unknown	Unknown	No								
3	LOU[redacted]	Dial America Extra	Unk[redacted]	Unknown	Unknown	No								
4	MEL[redacted]	Dial America Extra	Unk[redacted]	Unknown	Unknown	No								
5	KEL[redacted]	Dial America Extra	Unk[redacted]	Inbound	Unknown	No								
6	PLA[redacted]	Unknown	Unk[redacted]	Unknown	Unknown	No								
7	EAR[redacted]	Dial America	Ter[redacted]ompany)	Inbound	Unknown	Yes								
8	TIER[redacted]	Chattanooga	Keo[redacted]	Repair	Deployment	No	SHMQ72FC934D-3	8/9/2022 0:00	8/10/2022 0:00			630	1	Acce
9	TIER[redacted]	Chattanooga	Keo[redacted]	Repair	Deployment	No	SHMQ8300BF6B-2	8/1/2022 0:00	8/1/2022 0:00			1600	1	Acce
10	TIER[redacted]	Chattanooga	Keo[redacted]	Repair	Deployment	No	SHMQ8300B053-3	8/2/2022 0:00	8/2/2022 0:00			1400	1	Acce
11	TIER[redacted]	Chattanooga	Keo[redacted]	Repair	Deployment	No	SHMQ83030233-3	8/8/2022 0:00	8/8/2022 0:00		8/8/2022 0:00	1500	1	Acce
12	TIER[redacted]	Chattanooga	Keo[redacted]	Repair	Deployment	No	SHMQ62EC0033-6	8/2/2022 0:00	8/5/2022 0:00			900	4	Acce
13	TIER[redacted]	Chattanooga	Keo[redacted]	Repair	Deployment	No	SHMQ83027588-3	8/8/2022 0:00	8/8/2022 0:00			1700	1	Acce

Step 4

Once cleaned and merged. The output is saved as a csv called "Deploy_Cleaned". The PowerBi just needs to be repointed to the Deploy_Cleaned file and refreshed.



```
Deploy_Cleaned.csv            8/12/2022 8:31 AM    Microsoft Excel Comma Separat...     2,216 KB
Deployment_Dash_-_daily.xls   8/12/2022 8:59 AM    Microsoft Excel 97-2003 Worksh...   22,066 KB
```

after
before