--------------------------------------------------------------------------------------------------------------------------------

AUTOMATION PROJECT – PARENTCONTAINER

import pandas as pd

data = pd.read\_excel("AD Dump\_License\_Asset Details.xlsx")

## - the user need to change the file name before running the code

# Main Engine of the script - for extraction

def extract\_all\_after\_ou\_clean(text):

    extracted\_values = []

    for part in reversed(text.split(",")):  # Iterate in reverse order

        if part.startswith("OU="):  # Only extract if part starts with "OU="

            extracted\_values.append(part[part.index("=") + 1:].strip("[]"))  # Extract from "=" to end, remove brackets

    return ", ".join(extracted\_values)  # Join with ", " after removing quotes

# Sub-Engine of the script - after extraction

data["extracted\_values\_clean\_3"] = data["ParentContainer"].apply(extract\_all\_after\_ou\_clean)

# Defining new column names

new\_col\_names = ['Col\_1', 'Col\_2', 'Col\_3', 'Col\_4', 'Col\_5', 'Col\_6', 'Col\_7']

# Intel picking the values as per user

max\_values = max(len(x.split(",")) for x in data["extracted\_values\_clean\_3"])

new\_col\_names = [f"Column\_{i}" for i in range(max\_values)]

data[new\_col\_names] = data["extracted\_values\_clean\_3"].str.split(",", expand = True)

# Define new column names

new\_col\_names = {

    "Column\_0": "High\_Location",

    "Column\_1": "Location\_",

    "Column\_2": "Sub\_Location",

    "Column\_3": "Department",

    "Column\_4": "Sub\_department",

    "Column\_5": "Sub\_department\_1",

    "Column\_6": "Sub\_Department\_2",

    "Column\_7": "Sub\_Department\_3"}

# Finalizing the column names

data.rename(columns=new\_col\_names, inplace = True)

# Replacing Asia

data['High\_Location'] = data.apply(lambda row: row['Sub\_Location'] if row['High\_Location'] == 'Asia' else row['High\_Location'], axis=1)

# Rplacing Asscociates

data['High\_Location'] = data.apply(lambda row: row['Sub\_Location'] if row['High\_Location'] == 'Asscociates' else row['High\_Location'], axis=1)

# Replacing USA

data['High\_Location'] = data.apply(lambda row: row['Location\_'] if row['High\_Location'] == 'USA' else row['High\_Location'], axis=1)

# Replacing Datacenter

data['High\_Location'] = data.apply(lambda row: row['Location\_'] if row['High\_Location'] == 'Datacenter' else row['High\_Location'], axis=1)

# Inverse replacement of Sri Lanka

data.loc[data['Location'] == 'Sri Lanka', 'High\_Location'] = 'Sri Lanka'

# data.loc[data['High\_Location'] == 'USA', 'High\_Location'] = data[data['High\_Location'] == 'USA']['Location']

## - this is the anchor code

# Columns to fill which are blanks - trick

data['High\_Location'] = data['High\_Location'].fillna('')

# Values to be stripped

data['High\_Location'] = data['High\_Location'].str.strip()

# Saving the final product to excel

data.to\_excel("AD-Users-2024-01-10\_updated\_test.xlsx", index=False)

--------------------------------------------------------------------------------------------------------------------------------

TRIAL – CODE for ANCHORING – (Please ignore the below)

def extract\_all\_after\_ou\_clean(text):

    extracted\_values = []

    for part in reversed(text.split(",")):  # Iterate in reverse order

        if part.startswith("OU="):  # Only extract if part starts with "OU="

            extracted\_values.append(part[part.index("=") + 1:].strip("[]"))  # Extract from "=" to end, remove brackets

    return ", ".join(extracted\_values)  # Join with ", " after removing quotes

data["extracted\_values\_clean\_3"] = data["ParentContainer"].apply(extract\_all\_after\_ou\_clean)

# Defining new column names

new\_col\_names = ['Col\_1', 'Col\_2', 'Col\_3', 'Col\_4', 'Col\_5', 'Col\_6', 'Col\_7']

max\_values = max(len(x.split(",")) for x in data["extracted\_values\_clean\_3"])

new\_col\_names = [f"Column\_{i}" for i in range(max\_values)]

data[new\_col\_names] = data["extracted\_values\_clean\_3"].str.split(",", expand = True)

# Define new column names

new\_col\_names = {

    "Column\_0": "High\_Location",

    "Column\_1": "Location",

    "Column\_2": "Sub\_Location",

    "Column\_3": "Department",

    "Column\_4": "Sub\_department",

    "Column\_5": "Sub\_department\_1",

    "Column\_6": "Sub\_Department\_2",

    "Column\_7": "Sub\_Department\_3"}

data.rename(columns=new\_col\_names, inplace = True)

filtered\_data\_1 = data.loc[(data["Licenses"] != "Not Found") & (data["PasswordExpired"] != True)]

filtered\_data\_1.loc[filtered\_data\_1['Sub\_Location'].isnull(), 'Sub\_Location'] = filtered\_data\_1['Location']

# this data is the data which fills the blank cells with values taken from Location column

columns\_to\_fill = ['High\_Location', 'Location', 'Sub\_Location', 'Department', 'Sub\_department',

                  'Sub\_department\_1', 'Sub\_department\_2', 'Sub\_department\_3']

placeholder\_value = 'na'

filtered\_data\_1['Concatenate\_Column'] = filtered\_data\_1['Location'] + '\_' + filtered\_data\_1['Sub\_Location'] + '\_' + filtered\_data\_1['Department'] + '\_' + filtered\_data\_1['Sub\_department'] + '\_'

The above is the concatenation code

columns\_to\_trim = ['High\_Location', 'Location', 'Sub\_Location', 'Department', 'Sub\_Department', 'Sub\_Department\_1',

                   'Sub\_Department\_2', 'Sub\_Department\_3', 'Concatenate\_Column']

filtered\_data\_1[columns\_to\_trim] = filtered\_data\_1[columns\_to\_trim].apply(lambda x: x.str.strip())

data.to\_excel("AD-Users-2024-01-10\_updated\_3.xlsx", index=False)

--------------------------------------------------------------------------------------------------------------------------------

columns\_to\_fill = ['High\_Location', 'Location', 'Sub\_Location', 'Department', 'Sub\_department',

                  'Sub\_department\_1', 'Sub\_department\_2', 'Sub\_department\_3']

placeholder\_value = 'na'

filtered\_data\_1[columns\_to\_fill] = filtered\_data\_1[columns\_to\_fill].fillna(placeholder\_value)

location\_filter = [' India', ' South Africa', ' Vikhroli DC', ' New Jersey', ' Sri Lanka',

       ' UAE', ' Morocco', ' Kenya', ' Nepal', ' Malaysia', ' Uganda', ' Tanzania', ' Myanmar',

       ' AUSTRALIA', ' Oman', ' ALGERIA', ' Europe', ' San Francisco', ' Kurkumbh DC', ' SouthAfrica', ' Brazil',

                   ' Sikkim Golden Cross DC', ' Goa Verna DC', ' Europe', ' SA Medpro', ' QCIL Uganda']

filtered\_data\_2 = filtered\_data\_1[filtered\_data\_1["Location"].isin(location\_filter)]

data['Sub\_Location'].unique()

location\_filter = [' India', ' South Africa', ' Vikhroli DC', ' New Jersey', ' Sri Lanka',

       ' UAE', ' Morocco', ' Kenya', ' Nepal', ' Malaysia', ' Uganda', ' Tanzania', ' Myanmar',

       ' AUSTRALIA', ' Oman', ' ALGERIA', ' Europe', ' San Francisco', ' Kurkumbh DC', ' SouthAfrica', ' Brazil',

                   ' Sikkim Golden Cross DC', ' Goa Verna DC', ' Europe', ' SA Medpro', ' QCIL Uganda']

filtered\_data\_2 = filtered\_data\_1[filtered\_data\_1["Location"].isin(location\_filter)]

filtered\_data\_1.to\_excel("AD-Users-2024-01-10\_updated\_test.xlsx", index=False)