# Problem #1

Companies often have two names: legal name and a DBA name, which means "Doing Business As". We need to split legal and DBA name.

Write a Python function clean\_names(raw\_names) that gets a list of raw names, processes it and returns the list of pairs equal to CLEANED\_NAME\_PAIRS list.

RAW\_NAMES = [

    'SPV  Inc., DBA:   Super  Company',

    'Michael Forsky LLC d.b.a F/B Burgers .',

    '\*\*\* Youthful You Aesthetics \*\*\*',

    'Aruna Indika (dba. NGXess)',

    'Diot SA,  -  D. B. A.   \*Diot-Technologies\*',

    'PERFECT PRIVACY, LLC, d-b-a Perfection,',

    'PostgreSQL DB Analytics',

    '/JAYE INC/',

    ' ETABLISSEMENTS  SCHEPENS /D.B.A./ ETS\_SCHEPENS',

    'DUIKERSTRAINING OOSTENDE | D.B.A.:  D.T.O. '

]

CLEANED\_NAME\_PAIRS = [

    ('SPV Inc',                  'Super Company'),

    ('Michael Forsky LLC',       'F/B Burgers'),

    ('Youthful You Aesthetics',  None),

    ('Aruna Indika',             'NGXess'),

    ('Diot SA',                  'Diot-Technologies'),

    ('PERFECT PRIVACY, LLC',     'Perfection'),

    ('PostgreSQL DB Analytics',  None),

    ('JAYE INC',                 None),

    ('ETABLISSEMENTS SCHEPENS',  'ETS SCHEPENS'),

    ('DUIKERSTRAINING OOSTENDE', 'D.T.O'),

]

def clean\_names(raw\_names):

    import re

try:

result\_list = []

#this part is to compile the regular expression for legal name or dba name once it has been split

pattern = re.compile(r'\w[/.\w\s///,-]\*\w')

#iterate over each string in RAW\_NAMES list

for cstring in raw\_names:

#this part is to split legal name and dba name if we get dba like pattern

splits = re.split(r'\W\*[dD]\W\*[bB]\W\*[aA]\W\W\*', cstring)

#this part is if re.split results in two strings which means dba like pattern found

if len(splits) > 1:

#below part is to search legal\_name

clean\_legal\_name = pattern.search(splits[0])

#below part is to actually get legal name

legal\_name = clean\_legal\_name.group()

#we just join the words in legal name by a space

legal\_name = ' '.join(legal\_name.split())

#below part is to search dba name

clean\_dba\_name = pattern.search(splits[1])

#below part is to actually get dba name

dba\_name = clean\_dba\_name.group()

#we just join the words in dba name by a space

dba\_name = ' '.join(dba\_name.split())

#below part is to create a tuple and replace underscore with space

result\_tuple = (legal\_name.replace("\_"," "),dba\_name.replace("\_"," "))

#this part is if re.split results one strings which means dba like pattern not found

else:

#same steps like above

clean\_legal\_name = pattern.search(splits[0])

legal\_name = clean\_legal\_name.group()

legal\_name = ' '.join(legal\_name.split())

result\_tuple = (legal\_name.replace("\_"," "),None)

#below we just append the tupple to the actual result list

result\_list.append(result\_tuple)

return result\_list

except Exception as e:

print(str(e))

assert clean\_names(RAW\_NAMES) == CLEANED\_NAME\_PAIRS