

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt # Визуализирует данные
import seaborn as sns # Визуализация данных
import os
import shutil
```

```
df=pd.read_excel("Остатки товаров v2.3.xlsx")
```

```
df=df.drop(labels=[0])
print(df.head(10))
```

| | Номенклатура | 1.Основной склад "ВентЭл" | \ |
|----|---|---------------------------|---|
| 1 | Материалы | 42000 | |
| 2 | Комплектующие для компьютера | NaN | |
| 3 | Термоэтикетка 58*40 ЭКО (1 шт из рулона 700 шт) | 42000 | |
| 4 | Термоэтикетки 58*40 ЭКО, 700 шт в ролике | NaN | |
| 5 | Товары | 41728 | |
| 6 | Вентиляторы | 40987 | |
| 7 | .ЕВМРАРСТ | 4859 | |
| 8 | 02002-4-1021 Соединительный провод с разъемом | NaN | |
| 9 | 09414-2-4039 Решетка защитная | 1 | |
| 10 | 09418-2-4039 Решетка защитная | 8 | |

| | 2.СТОК | 3.Резерв | 4.В пути | 5.БРАК | 6.Офис | 7.Недовоз | \ |
|----|--------|----------|----------|--------|--------|-----------|---|
| 1 | NaN | NaN | 62 | NaN | NaN | NaN | |
| 2 | NaN | NaN | 2 | NaN | NaN | NaN | |
| 3 | NaN | NaN | NaN | NaN | NaN | NaN | |
| 4 | NaN | NaN | 60 | NaN | NaN | NaN | |
| 5 | 945 | 1215 | 15326 | 26 | 1 | 44 | |
| 6 | 945 | 1035 | 15323 | 26 | 1 | 44 | |
| 7 | 942 | 69 | 163 | 1 | NaN | 5 | |
| 8 | 4 | NaN | NaN | NaN | NaN | NaN | |
| 9 | NaN | NaN | NaN | NaN | NaN | NaN | |
| 10 | NaN | NaN | NaN | NaN | NaN | NaN | |

| | 8.Талдом Склад 000 "ВентЭл" | Итого |
|----|-----------------------------|-------|
| 1 | NaN | 42062 |
| 2 | NaN | 2 |
| 3 | NaN | 42000 |
| 4 | NaN | 60 |
| 5 | 3090 | 62375 |
| 6 | 3090 | 61451 |
| 7 | NaN | 6039 |
| 8 | NaN | 4 |
| 9 | NaN | 1 |
| 10 | NaN | 8 |

```
def cutter_df(word,df_old):
    df_new=df_old.copy()
    df_check=df_old[word].notna() # Возвращает bool-тип если объекты ненулевые
    for i in range(1,len(df_check)):
        if df_check[i]:
            delta=df_old['Итого'][i]-df_old[word][i]
            #print(delta)
            df_new['Итого'][i]=delta
    df_new=df_new.drop(columns=word)
    return df_new.copy()

df_travelled=cutter_df('4.В пути ',df)
df_travelled=cutter_df('7.Недовоз',df_travelled)
'Оставим только самые необходимые столбцы'
df_travelled=df_travelled.drop(columns=['1.Основной склад
"ВентЭл"', '2.СТОК', '3.Резерв', '5.БРАК', '6.Офис', '8.Талдом Склад 000 "ВентЭл"'])

df_travelled

# df_travelled=df.copy()
# df_check=df['4.В пути '].notna() # Возвращает bool-тип если объекты ненулевые
# for i in range(1,len(df_check)):
#     if df_check[i]:
#         #print(df['Итого'][i],df['4.В пути '][i])
#         delta=df['Итого'][i]-df['4.В пути '][i]
#         #print(delta)
#         df_travelled['Итого'][i]=delta

# df_travelled=df_travelled.drop(columns='4.В пути ')
# df_travelled
```

| | Номенклатура | Итого |
|------|---|--------|
| 1 | Материалы | 42000 |
| 2 | Комплектующие для компьютера | 0 |
| 3 | Термоэтикетка 58*40 ЭКО (1 шт из рулона 700 шт) | 42000 |
| 4 | Термоэтикетки 58*40 ЭКО, 700 шт в ролике | 0 |
| 5 | Товары | 47005 |
| ... | ... | ... |
| 1318 | Обслуживание 1С | 0 |
| 1319 | Транспортные услуги за доставку груза | 0 |
| 1320 | Услуги по организации экспресс доставки | 0 |
| 1321 | Услуги по перевозке и хранению грузов | 0 |
| 1322 | Итого | 104575 |

1322 rows × 2 columns

```
df_travelled=df_travelled[df_travelled['Итого']>0]
df_travelled
```

| | Номенклатура | Итого |
|------|---|--------|
| 1 | Материалы | 42000 |
| 3 | Термоэтикетка 58*40 ЭКО (1 шт из рулона 700 шт) | 42000 |
| 5 | Товары | 47005 |
| 6 | Вентиляторы | 46084 |
| 7 | .EBMPAPST | 5871 |
| ... | ... | ... |
| 1306 | YWF-K92-4E-42B Электродвигатель (YWF4E-92/42B-K) | 9 |
| 1307 | АИР 100 S2 4кВт 3000об/мин IM3081 (B5) Электро... | 1 |
| 1308 | Энерал-Центр (Воронин 5%) | 1 |
| 1309 | АИР 90L2 3.0кВт 3000об/мин 1081 Электродвигатель | 1 |
| 1322 | Итого | 104575 |

1187 rows × 2 columns

```
#Ctrl+H
#([\r\n]*)(\r?\n)?
#"$1", $2

word_list=["Материалы",
"Товары",
"Вентиляторы",
".EBMPAPST",
".NICOTRA ( Богданов)",
".NMB",
".Spal (Богданов)",
"Запчасти",
"Осевые 12V - А-поток (от радиатора)",
"Осевые 12V - S-поток (на радиатор)",
"Осевые 24V - А-поток (от радиатора)",
"Осевые 24V - S-поток (на радиатор)",
"Радиальные 12V",
"Радиальные 24V",
".SUNON (Богданов)",
".WEIGUANG",
"Т - ФЛАНЦЕВЫЙ",
"аналог W1G",
"Двигатели",
"КОМПАКТНЫЕ",
"Мотор-колеса",
"220 Вольт",
"24 Вольта",
"380 Вольт",
"НАСТЕННАЯ ПАНЕЛЬ",
"200",
"250",
"300",
"350",
"400",
"450",
"500",
"550",
```

"630",
"710",
"800",
"910",
"РАДИАЛЬНЫЕ",
"РЕШЕТКА В-поток (нагнетание)",
"200В",
"250В",
"300В",
"315",
"330В",
"350В",
"400В",
"420В",
"450В",
"500В",
"550В",
"560В",
"600В",
"630В",
"РЕШЕТКА S-поток (всасывание)",
"200S",
"250S",
"300S",
"315S",
"330S",
"350S",
"400S",
"420",
"450S",
"500S",
"550S",
"560S",
"600S",
"630S",
"710",
"800S",
"Тангенциальные",
".Ziehl-Abegg",
"Старые",
"УКРАИНА",
".Агровент-М (Воронин)",
".Тепловенткомплект (Воронин)",
"Bahcivan (Богданов)",
"ComeFri",
"DUNLI (Корнилов)",
"FANDIS/Провенто (БОГДАНОВ)",
"Fans-tech",
"HUAWEI",
"Jamicon (Воронин)",
"JASON",
"KRUBO",
"LFT",
"LONGWELL",
"оконные",
"MEANSOON",
"MES",
"MINXIN",
"Multi-Wing",

```

"OSTBERG (Богданов)",
"SANHE",
"Systemair (Воронин)",
"TIDAR",
"Vilmann",
"VORTICE",
"WISTRO",
"ВВФ (Воронин)",
"Вентиляционные аксессуары (Чеботарев)",
"ИОЛЛА (БОГДАНОВ)",
"Климат-смарт",
"КОРФ",
"Радойл YWF (Богданов)",
"РОВЕН",
"Судовые вентиляторы",
"ЭЛРЕ",
"ЯЛКА",
"Владимир",
"Все для Картошки Воронин 1 %",
"Разукomплектация (комплектация) по договору, контракту, тендеру",
"Фильтры",
"Фильтры Подольск (отв. Корнилов 5%)",
"Электродвигатели",
"Энерал-Центр (Воронин 5%)",
"Услуги",
"Услуги ИП Слесарев В.Д.",
"Услуги сторонних организаций",
"Итого",
"VORTICE",
".NMB ",
"",]
for word in word_list:
    df_travelled=df_travelled[df_travelled['Номенклатура']!=word]

# Рассмотрим пример что всё работает. Для этого рассмотрим мотор-колесо, где 24 товара было
недовезено
df_travelled[df_travelled['Номенклатура']=='CF280B-2E-AC0 Мотор-колесо MES']

```

| | Номенклатура | Итого |
|------|--------------------------------|-------|
| 1104 | CF280B-2E-AC0 Мотор-колесо MES | 34 |

```

'Переименуем для удобства'
df_travelled=df_travelled.rename(columns={'Номенклатура':'Товар','Итого':'Кол-во'})
'Сбросим нумерацию индексов'
df_travelled = df_travelled.reset_index(drop=True)
df_travelled

```

| | Товар | Кол-во |
|---|---|--------|
| 0 | Термоэтикетка 58*40 ЭКО (1 шт из рулона 700 шт) | 42000 |
| 1 | 02002-4-1021 Соединительный провод с разъемом | 4 |
| 2 | 09414-2-4039 Решетка защитная | 1 |
| 3 | 09418-2-4039 Решетка защитная | 8 |
| 4 | 09500-2-4039 Защитная решетка | 1 |

| | Товар | Кол-во |
|------|---|--------|
| ... | ... | ... |
| 1071 | YWF-K102-4E-60B (11мм) Электродвигатель | 2 |
| 1072 | YWF-K92-4E-35B Электродвигатель | 5 |
| 1073 | YWF-K92-4E-42B Электродвигатель (YWF4E-92/42B-K) | 9 |
| 1074 | AIP 100 S2 4кВт 3000об/мин IM3081 (B5) Электро... | 1 |
| 1075 | AIP 90L2 3.0кВт 3000об\мин 1081 Электродвигатель | 1 |

1076 rows × 2 columns

```

from zipfile import ZipFile
files = os.listdir()
excels = list(filter(lambda x: x.endswith('.xlsx'), files))
excels = list(filter(lambda x: x.startswith('остатки'), excels))
assert len(excels)==1 , 'В корневой папке содержится больше чем одна таблица с названием,
начинающемся на (остатки)'
second_excel=excels[0]
print(second_excel)

name_file='Остатки (измененный формат).xlsx'

# Создаем временную папку
tmp_folder = '/tmp/convert_wrong_excel/'
os.makedirs(tmp_folder, exist_ok=True)

# Распаковываем excel как zip в нашу временную папку
with ZipFile(second_excel) as excel_container:
    excel_container.extractall(tmp_folder)

# Переименовываем файл с неверным названием
wrong_file_path = os.path.join(tmp_folder, 'xl', 'SharedStrings.xml')
correct_file_path = os.path.join(tmp_folder, 'xl', 'sharedStrings.xml')
os.rename(wrong_file_path, correct_file_path)

# Запаковываем excel обратно в zip и переименовываем в исходный файл
shutil.make_archive('yourfile', 'zip', tmp_folder)
second_excel=name_file

files_find=list(filter(lambda x: x.startswith('Остатки (измененный формат)'), files))
if len(files_find)>0:
    os.remove(second_excel)

os.rename('yourfile.zip', second_excel)

df_second=pd.read_excel(second_excel)
df_second=df_second.drop(columns=['Unnamed: 1','Unnamed: 2','Unnamed: 4'])
df_second=df_second.rename(columns={'Unnamed: 0':'Товар','Unnamed: 3':'Кол-во'})
row = df_second[df_second['Товар'] == 'Итого'].index.tolist()[0]
df_second=df_second.iloc[:row]
df_second=df_second.drop(labels=[0,1,2,3])
print(df_second)

os.remove('Остатки (измененный формат).xlsx')

```

остатки 231222.xlsx

| | Товар | Кол-во |
|------|--|--------|
| 4 | 001-B53-03S 24V OR RPA3VCV Spal Вентилятор | 2 |
| 5 | 02-122-1 | 2 |
| 6 | 004-B42-28D 24V OR RPA3VCB Spal Вентилятор | 19 |
| 7 | 02-101-1 | 19 |
| 8 | 004-B43-28S 24V Spal Вентилятор | 18 |
| ... | ... | ... |
| 7599 | 06-042-1 | 6 |
| 7600 | Электропривод BLF24.1 | 1 |
| 7601 | 02-331-1 | 1 |
| 7602 | Электроштабелер ETV 214 № 91040737 | 1 |
| 7603 | 02-361-1 | 1 |

[7600 rows x 2 columns]

```
word_list=['Итого',
'Ячейка',
'Номенклатура']
for word in word_list:
    df_second=df_second[df_second['Товар']!=word]

df_second
```

| | Товар | Кол-во |
|------|--|--------|
| 4 | 001-B53-03S 24V OR RPA3VCV Spal Вентилятор | 2 |
| 5 | 02-122-1 | 2 |
| 6 | 004-B42-28D 24V OR RPA3VCB Spal Вентилятор | 19 |
| 7 | 02-101-1 | 19 |
| 8 | 004-B43-28S 24V Spal Вентилятор | 18 |
| ... | ... | ... |
| 7599 | 06-042-1 | 6 |
| 7600 | Электропривод BLF24.1 | 1 |
| 7601 | 02-331-1 | 1 |
| 7602 | Электроштабелер ETV 214 № 91040737 | 1 |
| 7603 | 02-361-1 | 1 |

7600 rows × 2 columns

```
num_holder=[]
for a1 in range(10):
    for a2 in range(10):
        if a1==0 or (a1==1 and a2<2) or a1==9:
            for a3 in range(10):
                for a4 in range(10):
                    for a5 in range(10):
                        place=str(a1)+str(a2)+'-'+str(a3)+str(a4)+str(a5)
                        if place == '07-409':
                            a6=9
```

```

        else:
            a6=1
            place=place+'-'+str(a6)
            num_holder.append(place)

similar=list(set(num_holder) & set(df_second['Товар']))
for word in similar:
    df_second=df_second[df_second['Товар']!=word]

```

```
df_second.head(20)
```

| | Товар | Кол-во |
|----|---|--------|
| 4 | 001-B53-03S 24V OR RPA3VCV Spal Вентилятор | 2 |
| 6 | 004-B42-28D 24V OR RPA3VCB Spal Вентилятор | 19 |
| 8 | 004-B43-28S 24V Spal Вентилятор | 18 |
| 10 | 006-A39-22 - 3-х скорост. 12V Spal Вентилятор | 34 |
| 12 | 006-A45-22 12V RPA3VCV Spal Вентилятор | 81 |
| 15 | 006-B39-22 24V RPA3VCV Spal Вентилятор | 62 |
| 17 | 006-B45-22 24V RPA3V(2+2)CV вентилятор центроб... | 81 |
| 20 | 006-A45-22 12V Spal Вентилятор | 1 |
| 22 | 007-A56-32D Вентилятор Spal | 2 |
| 24 | 007-B56-32D 24V RA(1)4VCB Вентилятор центробеж... | 43 |
| 27 | 008-A100-93D Вентилятор Spal | 21 |
| 29 | 008-A45-02 Вентилятор Spal | 12 |
| 31 | 008-B100-93D 24V GR RPA3VCB Вентилятор | 40 |
| 33 | 008-B100-93D 24V Вентилятор | 49 |
| 35 | 008-B45-02 24V GR RPA3VCV Вентилятор | 1 |
| 37 | 008-B45/2C-02 24V GR RA3VCV Spal Вентилятор | 59 |
| 40 | 009-A70-74D 12V GR CB Spal Вентилятор | 26 |
| 43 | 009-B70-74D 24V GR RPA3VCB Spal Вентилятор | 9 |
| 45 | 010-B70-74D 24V GR RPA 3VCB Вентилятор Spal | 209 |
| 48 | 011-B40-22 24V RPA3VCV S.E. Вентилятор центроб... | 41 |

```

from zipfile import ZipFile
files = os.listdir()
print(files)
excels = list(filter(lambda x: x.endswith('.xlsx'), files))
excels = list(filter(lambda x: x.startswith('Элре'), excels))
#assert len(excels)==1 , 'В корневой папке содержится больше чем одна таблица с названием,
начинающемся на (элре)'
elre_excel=excels[0]
print(elre_excel)

word_list=["Итого",
"Инвентарь и хозяйственные принадлежности",

```


"Товары",
"Вентиляторы",
"Beijing Henry Mechanical and Electrical Equipment Co.,Ltd",
"Comefri",
"EBMPAPST",
"FANS-TECH ELECTRIC",
"HANGZHOU MEANSOON VENTILATION CO.,LTD.",
"LONGWELL ELECTRIC TECHNOLOGY CO., LTD",
"Nicotra-Gebhardt",
"SHANDONG YUYUN SANHE MACHINERY CO.,LTD.",
"WEIGUANG ELECTRONIC CO.,LTD.",
"Ziehl-Abegg",
"Ziehl-abegg Elmotech LLC Kiev",
"Подшипники",
"!!!ЗПК",
"3200,3300 ZPK",
"!!Линейные_SAMICK, ArtNC, HIWIN, Exxelin, THK, BOSH",
"ArtNC",
"HIWIN",
"КАРЕТКИ",
"Каретки TECHNIX",
"рельсы направляющие HIWIN",
"SAMICK",
"BVC-R / АПП",
"Шариковые радиальные",
"BECO",
"BHTS высокотемпературные",
"BSS нерж.сталь",
"ELRE нерж. минитюрные",
"NMB",
"Корпусные нерж.",
"BSS пластик корпуса IBV-IBU-LFD-LDI-BECO",
"корпуса НЕРЖ",
"крышки",
"подшипники нерж. корпусные",
"узлы в сборе с нерж подшипником",
"ELRE",
"FARO",
"Комбинированные ролики",
"GE. SI. SA шарниры и пш скольжения",
"BECO_GE. SI. SA шарниры и пш скольжения",
"MTM_GE. SI. SA шарниры и пш скольжения",
"ZPK_GE. SI. SA шарниры и пш скольжения",
"НСВ EXPO MAKINA",
"722. SD КОРПУСА РТИ",
"ДОБАВИТЬ компл. УПЛ. НСВ",
"Hecht Kugellager GmbH&Co KG",
"MTM (Poland) / CX (Complex Poland)",
"NTN-SNR",
"Конические",
"Смазка",
"NTN-SNR-NSK-RHP-FYN-KOYO-NACHI-ORS",
"Радиально-упорные шариковые",
"Упорные",
"Шариковые радиальные",
"шариковые радиальные FBJ",
"РТИ",
"Обгонные муфты",
"Подшипники",

"RENK",
"SICHUAN MIGHTY MACHINERY CO. LTD.",
"SKF",
"Конические",
"Принадлежности",
"SNL 22200,22300 Подшипники",
"1200,2200,4200",
"MTM_SNL 22200,22300 Подшипники",
"NSK_Сферические роликовые",
"PTI_SNL 22200,22300 Подшипники",
"STEYR_SNL 22200,22300 Подшипники",
"ZPK_SNL 22200,22300 Подшипники",
"SNL ВТУЛКИ ЗАКРЕПИТЕЛЬНЫЕ",
"SNL КОРПУСА",
"НЕСНТ",
"MTM в компл с кольцами и уплотнениями",
"PTI",
"SLZ",
"SNL 200",
"ZPK в компл с кольцами и уплотнениями",
"ЗПУ Абсолютшар",
"Комплекующие к корпусам PTI",
"МЕДВЕДЬ",
"ТЕХНИКС Разъемные корпуса и компл.",
"кольца фикс.",
"крышки",
"STC-Steyr Wälzlager Deutschland GmbH Германия",
"TECHNIX",
"Линейные: втулки, валы, ШВП, винты, опоры",
"ZPK шарик втулки",
"MTM шарик втулки",
"Опорные ролики KR. KRV. KRE",
"ZPK Опорные ролики KR. KRV. KRE",
"Трансмиссия",
"Шарнирный наконечник",
"UC UK UCP UFL UCF UFL",
"ASAHI",
"PTI - UC",
"SLZ_UC UK",
"TECHNIX Корпусные UCP UCF UCFL UCT UCFC",
"UC_FBJ",
"ZPK_UC UK",
"корпуса штампованные FBJ. NN. SNR",
"Корпусные NTN-SNR-NSK-RHP-FYN-KOYO-NACHI-ORS",
"Узлы в сборе USFD.SY.FY_ZPK.PTI",
"PTI Ремни, уплотнения",
"Цепи и звездочки",
"ТУС",
"Цепи",
"Продажи Дудкина Наталья Анатольевна",
"Веза",
"Продажи электродвигатели, редукторы",
"Belimo Привода (Верба 5%)",
"Wistro (Замарин 5%)",
"Насосы",
"Преобразователи (Замарин 5%)",
"INNOVERT (Замарин 5%)",
"Ziehl-Abegg (Соловьев 5%)",
"Редукторы",

"Bonfiglioli (Замарин 5%)",
"Brevini (Замарин 5%)",
"NORD (Вербма 5%)",
"МехПривод-ТК-NMRV, RC, KA, FA (Замарин 5%)",
"ПРОМСИТЕХ (Замарин 5%)",
"IRW Червячный INNORED",
"025",
"030",
"040",
"050",
"063",
"075",
"090",
"Q Червячный квадратный INNOVARI",
"Q45",
"Q75",
"Проставки",
"Соосные INNOVARI",
"Червячный круглый INNOVARI",
"030",
"050",
"085",
"СИТИ РУС-ТРАМЕС, СИТИ (Замарин 5%)",
"ЭЛКОМ (Замарин 5%)",
"Тормоза (Замарин 5%)",
"ABLE (Замарин 5%)",
"CANTONI, ЕМА-ЕЛФА (Замарин 5%)",
"Выпрямители",
"Электродвигатели",
"ABB (Вербма 5%)",
"ABLE (Замарин 5%)",
"100",
"112",
"132",
"160",
"180",
"200",
"225",
"250",
"280",
"56",
"63",
"71",
"80",
"90",
"Незав. вентиляция",
"Однофазные",
"С тормозом",
"Cantoni Group (Замарин 5%)",
"ГАМАК (Замарин 5%)",
"INNOVARI, RED Промситех (Замарин 5%)",
"INNORED Китай",
"INNOVARI ELK",
"OD Взрывники",
"незав. вент.",
"Однофазные",
"С тормозом",
"Mosca Motori (Соловьев 5%)",
"Siemens (Замарин 5%)",

```

"УМЕР Румыния (Верба 2,5%)",
"WEG (Замарин 5%)",
"W21 AL",
"WEIGUANG ELECTRONIC (Замарин 5%)",
"ДАР АДЧР (Замарин 5%)",
"ИП Тимофеев В.А. (Замарин 5%)",
"КЗЭД (Замарин 5%)",
"Могилев-ВентЭл-Запад",
"4ВР",
"АИР",
"АИС",
"НПО МЭЗ, Кюгель (Могилев) (Верба 5%)",
"Практик (Замарин 5%)",
"112",
"132",
"160",
"56",
"63",
"71",
"80",
"90",
"АИС (DIN)",
"Однофазные",
"СЗЭМО (Замарин 5%)",
"АИС",
"АИР",
"Уралэлектро (Соловьев 5%)",
"Электромонтаж (Замарин 5%)",
"Элком (Замарин 5%)",
"Услуги"]

df_elre=pd.read_excel(elre_excel)

df_elre=df_elre.drop(labels=[0])
df_elre=cutter_df('4.В пути ',df_elre)
df_elre=df_elre[df_elre['Итого']>0]
df_elre=df_elre.rename(columns={'Номенклатура':'Товар','Итого':'Кол-во'})
for word in word_list:
    df_elre=df_elre[df_elre['Товар']!=word]
df_elre=df_elre.reset_index(drop=True)
df_elre=df_elre.drop(columns=['1.Основной склад','2.РЕЗЕРВ','Склад Талдом'])
print(df_elre.head(10))

```

```

['app.py', 'db', 'db_watch.ipynb', 'pngwing.com (2).png', 'pngwing.com-_2_.ico', 'wtf.ipynb',
'Несовпадения баз данных.xlsx', 'Остатки (измененный формат).xlsx', 'остатки 231222.xlsx', 'Остатки
товаров v2.3.xlsx', 'Элре 2023.xlsx']

```

Элре 2023.xlsx

| | Товар | Кол-во |
|---|--|--------|
| 0 | 4715MS-23T-B5A-D00 Вентилятор осевой компакты... | 6 |
| 1 | 09415-2-4039 Защитная решетка | 8 |
| 2 | 19117-2-4039 Решетка защитная | 1 |
| 3 | 4656 N (9274014139 VUC0119YQHCS) Вентилятор | 3 |
| 4 | LZ 30-4 Решетка защитная | 3 |
| 5 | LZ 37 Защитная решетка | 7 |
| 6 | LZ-28-1 (9920028001) Решетка защитная | 2 |
| 7 | M4E074-EI15-01 Электродвигатель | 1 |

| | | |
|---|---|---|
| 8 | S6E630-AN01-01/F01 Вентилятор | 1 |
| 9 | JA0925H2BON-T(клемма) 220V (92x92x25) В(подшип... | 1 |

```
#Проверяем одну гипотезу (никак не влияет на код, но полезно)
word='VA09-AP12/C-54A 12V Вентилятор осевой 280 мм'
qq=df_elre[df_elre['Товар'] == word]
print(len(qq))
word2='YWF4D-200B-92/15-G Электровентилятор осевой'
qq=df_elre[df_elre['Товар'] == word2]
print(len(qq))
```

```
1
0
```

```
#Поиск совпадений в двух БД

def similar_finder(df_1s,df_sklad,df_secondary,dic,name_dic):
    dic[name_dic] = pd.DataFrame(columns=["Наименование","Кол-во 1С","Кол-во Склад","Разница","Примечание"])
    similar=list(set(df_1s['Товар']) & set(df_sklad['Товар']))
    for word in similar:
        row_sklad = df_sklad[df_sklad['Товар'] == word].index.tolist()[0]
        row_1s = df_1s[df_1s['Товар'] == word].index.tolist()[0]
        conclusion=""
        if len(df_secondary[df_secondary['Товар'] == word]) >0:
            row_secondary= df_secondary[df_secondary['Товар'] == word].index.tolist()[0]
            count_secondary=df_secondary['Кол-во'][row_secondary]
            conclusion=f"{count_secondary} найдено в базе другой фирмы"
        count1=df_1s['Кол-во'][row_1s]
        count2=df_sklad['Кол-во'][row_sklad]
        if count1-count2!=0:
            dic2=pd.DataFrame({"Наименование":[word],"Кол-во 1С":[count1],"Кол-во Склад":
            [count2],"Разница":[count1-count2],"Примечание":[conclusion] })
            dic[name_dic]=dic[name_dic].append(dic2, ignore_index= True)

    return dic
```

```
#Поиск совпадений в двух БД 1С

def similar_finder_1s(df_1s,df_2s,dic,name_dic):
    dic[name_dic] = pd.DataFrame(columns=["Наименование","Кол-во 1С Вентэл","Кол-во 1С Элре","Сумма"])
    similar=list(set(df_1s['Товар']) & set(df_2s['Товар']))
    for word in similar:
        row_2s = df_2s[df_2s['Товар'] == word].index.tolist()[0]
        row_1s = df_1s[df_1s['Товар'] == word].index.tolist()[0]
        count1=df_1s['Кол-во'][row_1s]
        count2=df_2s['Кол-во'][row_2s]
        dic2=pd.DataFrame({"Наименование":[word],"Кол-во 1С Вентэл":[count1],"Кол-во 1С Элре":
        [count2],"Сумма":[count1+count2] })
        dic[name_dic]=dic[name_dic].append(dic2, ignore_index= True)

    return dic
```

```
#Поиск одинаковых товаров в одной и той же базе
dic = {}
```

```

dic2 = {}

def mirrors_df(df,dic,name_dic,):
    from collections import defaultdict

    D = defaultdict(list)

    for i,item in enumerate(df['Товар']):
        D[item].append(i)
    D = {k:v for k,v in D.items() if len(v)>1}
    print(D)
    if len(D)>0:
        dic[name_dic] = pd.DataFrame(columns=["Наименование", "Суммарное Кол-во"])
        for name, numbers in D.items():
            summ=df['Кол-во'][numbers[0]]+df['Кол-во'][numbers[1]]
            df['Кол-во'][numbers[1]]=summ
            df=df.drop(labels=numbers[0])
            #print(f"Было\n{df['Товар'][numbers[1]]} Кол-во: {df['Кол-во'][numbers[1]]}")
            dic2=pd.DataFrame({"Наименование":[name], "Суммарное Кол-во":[summ] })
            dic[name_dic]=dic[name_dic].append(dic2, ignore_index= True)
            #print(dic)

    df=df.reset_index(drop=True)
    return df,dic

```

```

# similar=list(set(df_travelled['Товар']) - set(df_elre['Товар']))
# print(similar)

```

```

def empty_df(df_1s,df_2s,dic,name_dic,variant=1,df_last=0):
    if variant==1:
        title="Кол-во 1С ВентЭл"
    if variant==2:
        title="Кол-во 1С Элре"
    if variant==3:
        title="Кол-во в Складской базе"
    dic[name_dic] = pd.DataFrame(columns=["Наименование",title])
    if variant==1 or variant==2:
        notsimilar=list(set(df_1s['Товар']) - set(df_2s['Товар']))
    else:
        notsimilar_1=list(set(df_1s['Товар']) - set(df_2s['Товар']))
        notsimilar=list(set(notsimilar_1) - set(df_last['Товар']))
    for word in notsimilar:
        row_1s = df_1s[df_1s['Товар'] == word].index.tolist()[0]
        count1=df_1s['Кол-во'][row_1s]
        dic2=pd.DataFrame({"Наименование":[word],title:[count1] })
        dic[name_dic]=dic[name_dic].append(dic2, ignore_index= True)

    return dic

```

```

import warnings
warnings.filterwarnings('ignore') # Добавил тк без этого будет выводиться надпись, что в будущем
метод pd.append не будет использоваться

'Дополнительно Сбросим нумерацию индексов'
df_travelled = df_travelled.reset_index(drop=True)
df_second=df_second.reset_index(drop=True)

```

```

df_elre=df_elre.reset_index(drop=True)

dic['Отсутствие ВентЭл'] = pd.DataFrame(columns=["Наименование","Кол-во 1С","Кол-во Склад"])
dic['Отсутствие Элре'] = pd.DataFrame(columns=["Наименование","Кол-во 1С","Кол-во Склад"])
#dic['Несовпадения ВентЭл2'] = pd.DataFrame(columns=["Наименование","Кол-во 1С","Кол-во
Склад","Разница"])

df_travelled,dic = mirrors_df(df_travelled,dic,'Повторы в ВентЭл')
df_elre,dic = mirrors_df(df_elre,dic,'Повторы в Элре')

dic=similar_finder(df_travelled,df_second,df_elre,dic,'Несовпадения ВентЭл')
dic=similar_finder(df_elre,df_second,df_travelled,dic,'Несовпадения Элре')
dic=similar_finder_1s(df_travelled,df_elre,dic,'Совпадение в Элре и ВентЭл')

dic=empty_df(df_travelled,df_second,dic,"Отсутствие ВентЭл",variant=1)
dic=empty_df(df_elre,df_second,dic,"Отсутствие Элре",variant=2)
dic=empty_df(df_second,df_travelled,dic,"Отсутствие Склад",variant=3,df_last=df_elre)
# for i in range(len(df_travelled)):
#     checker=0
#     name1=df_travelled['Товар'][i]
#     for j in range(len(df_second)):
#         name2=df_second['Товар'][j]
#         if name1 == name2:
#             checker=1
#             break
#     if checker==0:
#         count1=df_travelled['Кол-во'][i]
#         dic2=pd.DataFrame({"Наименование":[name1],"Кол-во 1С":[count1],"Кол-во Склад":[''] })
#         dic['Отсутствие ВентЭл']=dic['Отсутствие ВентЭл'].append(dic2, ignore_index= True)

# for j in range(len(df_second)):
#     checker=0
#     name2=df_second['Товар'][j]
#     for i in range(len(df_travelled)):
#         name1=df_travelled['Товар'][i]
#         if name1 == name2:
#             checker=1
#             break
#     if checker==0:
#         count2=df_second['Кол-во'][j]
#         dic2=pd.DataFrame({"Наименование":[name2],"Кол-во 1С":[''], "Кол-во Склад":[count2] })
#         dic['Отсутствие']=dic['Отсутствие'].append(dic2, ignore_index= True)

from openpyxl import Workbook
# Добавляем библиотеку для записи в Excel с изменением названий листов
with pd.ExcelWriter('Несовпадения баз данных.xlsx') as writer:
    for name, df in dic.items():
        sheet_name=str(name)
        df.to_excel(writer, sheet_name=sheet_name)

```

```

{'YWF4E-630B-137/70-В Электровентилятор осевой': [535, 536], 'JF0925B2H-R(JF0925B2H-001C066R) 24V
(92x92x25) В(подшипник)Jamicon вентилятор с фишкой': [838, 839]}
{'KB3068 ОР (МТМ) Закрытый линейный подшипник': [1303, 1304]}

```

