Insertion des ProductAttribute dans la table productAttribute

Import des bibliothèques

Entrée [ ]:
```
from lxml import etree
import pandas as pd
import mysql.connector
from mysql.connector import Error
import re
import json
```

Parsage du catalogue Attribute

Entrée [ ]:
```
# à modifier avant chaque traitement d'un nouveau fichier XML
refPath = 'unzipped_files/cat-attributes-fr-25-11-2019_11-41-09'

xtree = etree.parse(refPath)
xroot = xtree.getroot()
```

Entrée [ ]:
```
for child in list(xroot):
    print(child.tag)
```

Génération d'un dataframe dfProdAtt

Entrée [ ]:
```
df_cols = ["productId", "attributes"]
rows=[]
```

Entrée [ ]:
```
for att in xroot.findall('products'):
    #print(att.tag)
    for i,att1 in enumerate(att.getchildren()):
        #print(att1.attrib['productId'])
        b = att1.getchildren()
        for i,att2 in enumerate(b):
            c = att2.getchildren()
            row = []
            for i,y in enumerate(c):
               # row = []
                row.append({"id": y.attrib['id'], "value": y.attrib['value']})
            row_json = json.dumps(row, ensure_ascii=False)
            rows.append({"productId": str(att1.attrib['productId']),
                         "attributes": (row_json)})

dfProdAtt = pd.DataFrame(rows, columns=df_cols)
dfProdAtt.sort_values(['attributes'], ascending=False)
```

NETTOYAGE DU DATAFRAME¶

Valeurs de productId dans XML cat

Entrée [ ]:
```
# à modifier avant chaque traitement d'un nouveau fichier XML
refPath1 = 'unzipped_files/cat-ref-FR3787ED_2019-11-22 11.30.32-22-11-2019_11-48-2

xtree1 = etree.parse(refPath1)
xroot1 = xtree1.getroot()
```

Entrée [ ]:
```
df_cols1 = ["productId"]
rows1 = []
for products in xroot1.iter('products'):
    for a,ec in enumerate(products.getchildren()):
        rows1.append(ec.attrib['productId'])
print(len(rows1))
```

Recherche des doublons productId

Entrée [ ]:
```
rows2 = []
doublons = []
for i in rows1:
    if i not in rows2:
        rows2.append(i)
    else:
        doublons.append(i)
print(len(rows2))
print(len(doublons))
```

On supprime les doublons

Entrée [ ]:
```
rows2 = list(set(rows2) - set(doublons))
print(len(rows2))
```

Comparaison des valeurs de productId entre df et Product

Entrée [ ]:
```
integrite = pd.Series(dfProdAtt['productId'].isin(rows2))
integrite
```

Entrée [ ]:
```
#Insertion de la série integrité dans dfImage
dfProdAtt['Integrité'] = integrite
dfProdAtt.sort_values(['Integrité'], ascending=False)
```

Entrée [ ]:
```
dfProdAtt['Integrité'].value_counts()
```

Entrée [ ]:
```
resultat = dfProdAtt.groupby('productId').nunique()
resultat.shape
```

Entrée [ ]:
```
dfProdAtt.attributes.replace("'"," ", regex=True, inplace=True)
dfProdAtt.attributes.replace("\\\\","", regex=True, inplace=True)
dfProdAtt.attributes.replace('1/2"',"1/2", regex=True, inplace=True)
dfProdAtt.attributes.replace('3/4"',"3/4", regex=True, inplace=True)
dfProdAtt.attributes.replace('3/8"',"3/8", regex=True, inplace=True)
dfProdAtt.attributes.replace('1/4"',"1/4", regex=True, inplace=True)
```

Entrée [ ]:

```
dfProdAtt.attributes.replace('1/2}','1/2"}', regex=True, inplace=True)
dfProdAtt.attributes.replace('3/4}','3/4"}', regex=True, inplace=True)
dfProdAtt.attributes.replace('3/8}','3/8"}', regex=True, inplace=True)
dfProdAtt.attributes.replace('1/4}','1/4"}', regex=True, inplace=True)
dfProdAtt.attributes.replace('18.90"','18.90', regex=True, inplace=True)
dfProdAtt.attributes.replace('17.72"','17.72', regex=True, inplace=True)
```

Insertion des données du dataframe dans la table Attribute

Entrée [ ]:

```
connection_config = {
                    'host':"localhost",
                    'port': 3308,
                    'database': 'bihr_db',
                    'user': 'BASTIER',
                    'passwd': "DA2019",
                    #'autocommit': True
                    }
```

Entrée [ ]:

```
try:
    connection = mysql.connector.connect(**connection_config)

    for i in range(dfProdAtt.shape[0]):
        prodAtt = dfProdAtt.iloc[i]
        print(i)
        if prodAtt['Integrité'] == True :
            print(prodAtt['Integrité'])
            print(prodAtt['attributes'])
            ProdAttInsertQuery = """INSERT INTO productattribute (productId, attri
                               VALUES
                               ("""+ """'""" + str(prodAtt['productId']) + """','"""+
            cursor = connection.cursor()
            result = cursor.execute(ProdAttInsertQuery)
        #print(prodAtt['productId'])

    connection.commit()
    print("Insertion datas in ProdAtt table successful ")
    cursor.close()

except mysql.connector.Error as error:
    print("Failed to insert datas in ProdAtt table : {}".format(error))

finally:
    if (connection.is_connected()):
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

Entrée [ ]:

```

```