After Downloading the PubmedBase files from the Pubmed site, a detailed study on XML files were carried. After a Study of 10 XML files, made the conclusion that all the files follow the same structure.

To read the XML file & understand the structure use the below mention code.

%fs head /mnt/pubmed2023/downloadurl/pubmed23n1166.xml

After understanding the structure the xml file was split to multiple datasets.

df\_xml1 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "Article") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

#display(df\_xml1)

df\_xml2 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "ChemicalList") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml2)

df\_xml3 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "CitationSubset") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml3)

df\_xml4 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "CoiStatement") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml4)

df\_xml5 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "CommentsCorrectionsList") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml5)

df\_xml6 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "DateCompleted") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml6)

df\_xml7 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "DateRevised") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml7)

df\_xml8 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "InvestigatorList") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml8)

df\_xml9 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "KeywordList") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml9)

df\_xml10 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "MedlineJournalInfo") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml10)

df\_xml11 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "MeshHeadingList") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml11)

df\_xml12 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "OtherAbstract") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml12)

df\_xml13 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "PMID") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml13)

df\_xml14 = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "SupplMeshList") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

display(df\_xml14)

After a detailed study of the 14 Data sets 2 datasets were used for the final solution to pull the necessary details needed for the Power BI.

Here is the final code used to combine the datasets

from pyspark.sql import functions as f

df\_xml\_countrynumber = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "Article") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

df\_xml\_countrynumber = df\_xml\_countrynumber.select('ArticleDate.Year', 'ArticleDate.Month', 'ArticleDate.Day', 'ArticleTitle', 'AuthorList.Author.LastName', 'AuthorList.Author.ForeName', "Journal.ISSN.true")

df\_xml\_countrynumber = df\_xml\_countrynumber.withColumnRenamed('true', 'Country\_Id')

df\_xml\_countrynumber = df\_xml\_countrynumber.withColumn('Date', f.concat(f.col('Year'), f.lit('-'), f.col('Month'), f.lit('-'), f.col('Day')))

df\_xml\_countrynumber = df\_xml\_countrynumber.select('ArticleTitle', 'LastName', 'ForeName', 'Date', 'Country\_Id' )

display(df\_xml\_countrynumber)

df\_xml\_country = spark.read \

.format("com.databricks.spark.xml") \

.option("rootTag", "PubmedArticleSet.PubmedArticle.MedlineCitation") \

.option("rowTag", "MedlineJournalInfo") \

.option("valueTag", True) \

.load('/mnt/pubmed2023/downloadurl/pubmed23n1166.xml')

#df\_xml\_country = df\_xml\_country.select('Journal.Country', 'ArticleTitle')

display(df\_xml\_country)

# To get the Country name

Final = df\_xml\_countrynumber.join(df\_xml\_country, df\_xml\_countrynumber.Country\_Id == df\_xml\_country.ISSNLinking, 'left')

# Final = Final.limit(1)

# display(Final)

from pyspark.sql.functions import \*

import pandas as pd

Final\_pandas = Final.toPandas()

# Final\_pandas = Final\_pandas.set\_index(['ArticleTitle']).apply(pd.Series.explode).reset\_index()

Final\_pandas = Final\_pandas.apply(lambda x: x.explode())

Final\_conv = spark.createDataFrame(Final\_pandas)

Final\_conv = Final\_conv.withColumn("Full\_name", concat(col("ForeName"), lit("-"), col("LastName")))

display(Final\_conv)

The same data Excel file is enclosed.