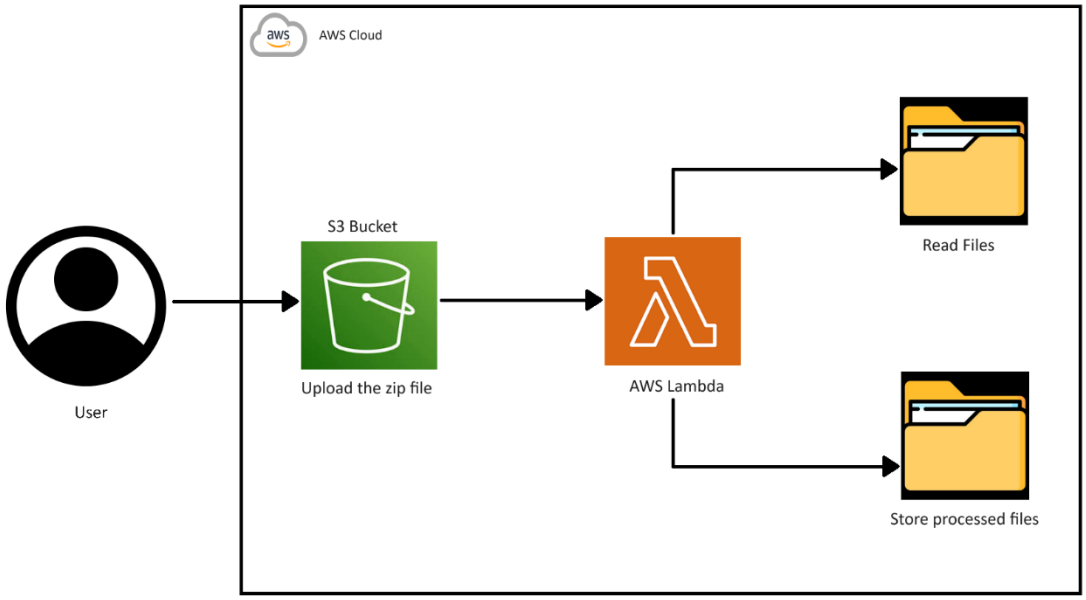
Task 2 : Python and Lambda

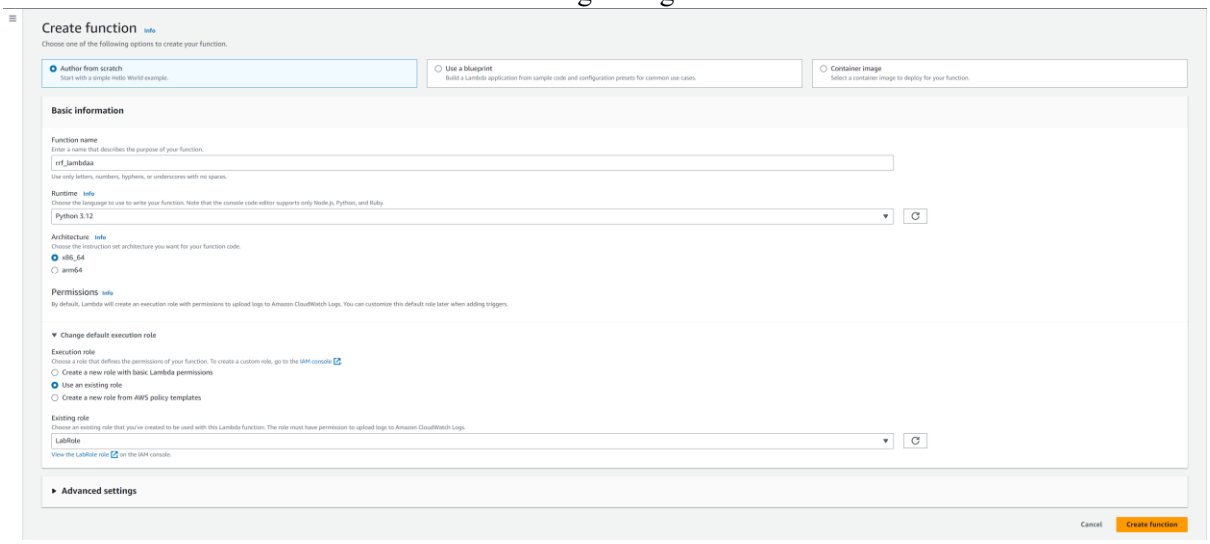
Work with RXNORM file,

1. Scrap the latest RXNORM file from NLM webpage
2. Download the latest RXNORM file with api_key
3. Create a log file for the downloaded file
4. While uploading the file to S3, create CloudWatch log file
5. Add header into each rff from RXNORM.xlsx
6. Add CODE_SET & VERSION_MONTH column with default values RxNorm and version month from downloaded filename
7. Convert dates into YYYY-MM-DD
8. Save files as txt delimited by comma(,)c
9. Validate row_count between original and converted files

Cloud Architecture of this Task

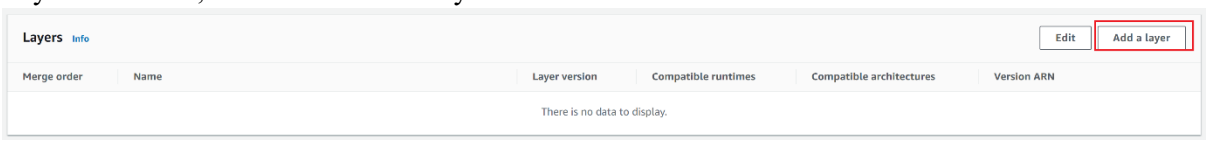


First we create a Lambda Function with following configurations:

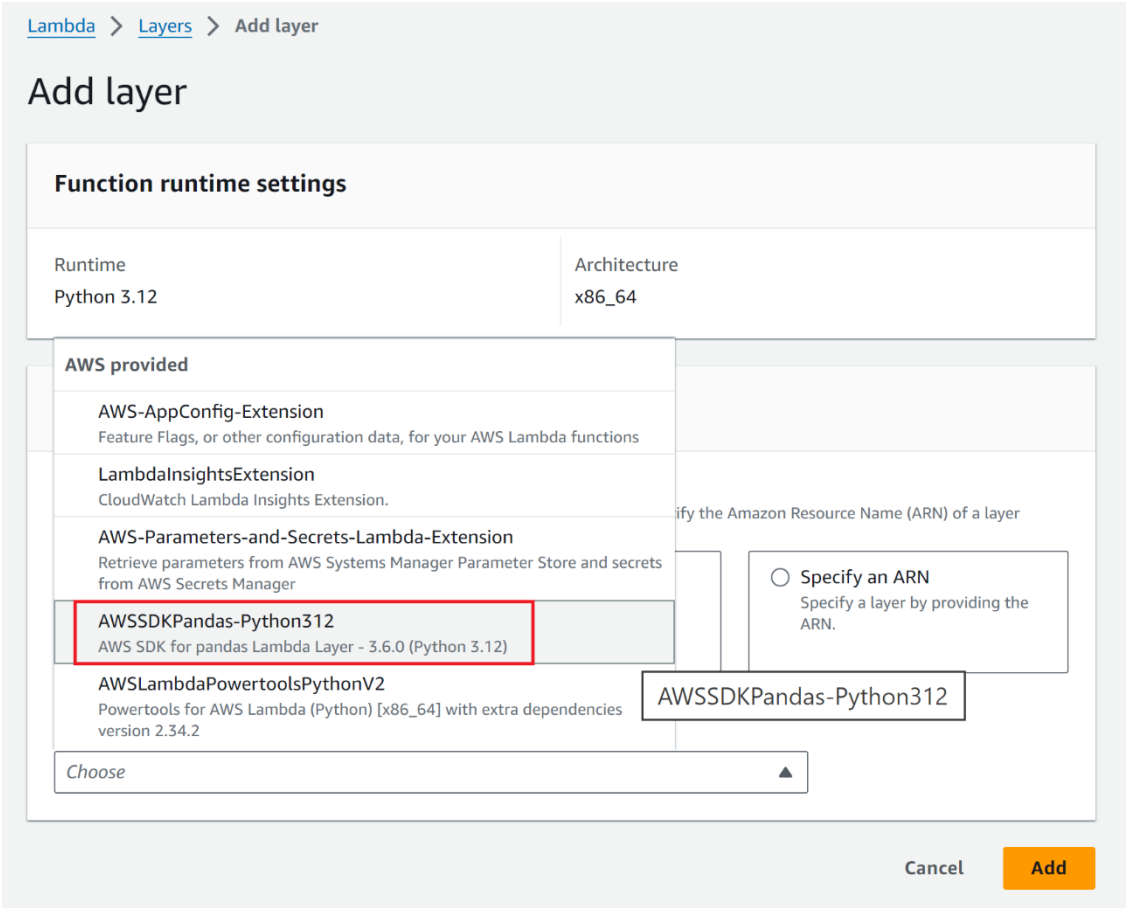


Secondly, we need to add Layers to our Lambda.

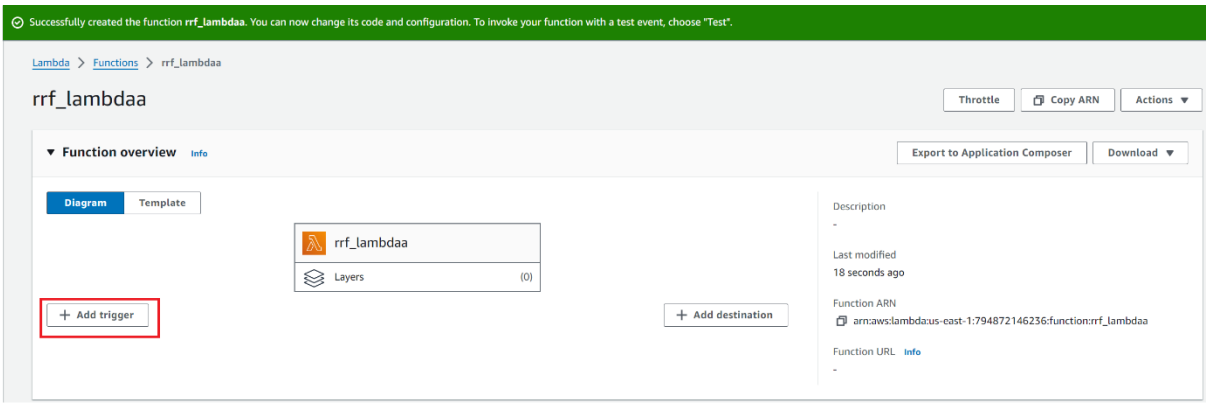
Open our newly created Lambda function, and scroll down. In the bottom of the page, we can see Layers tab. Here, we need to Add a Layer.



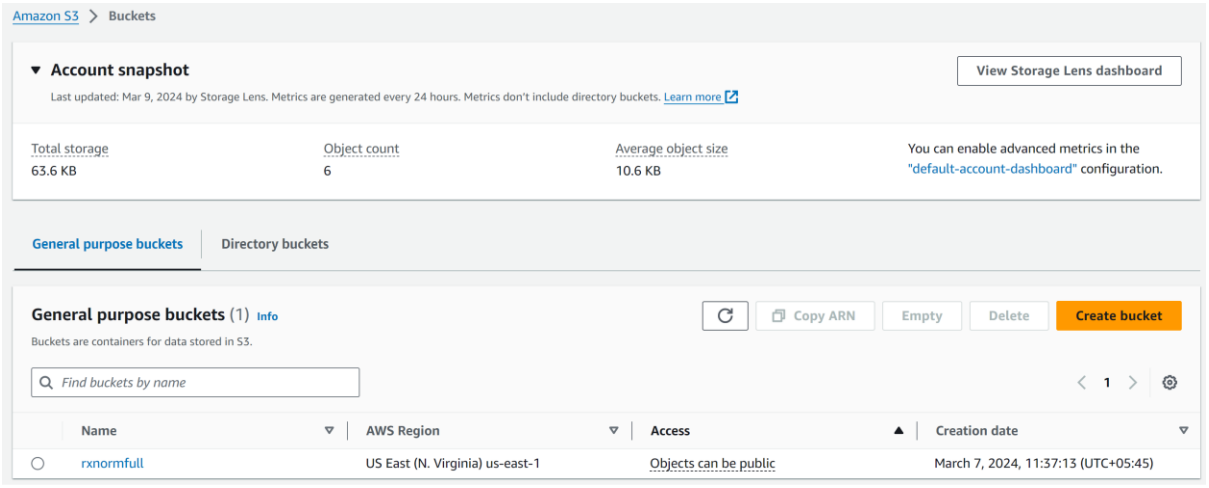
Here, select AWSSDKPandas-Python312 and click Add. Also choose Version as 4.



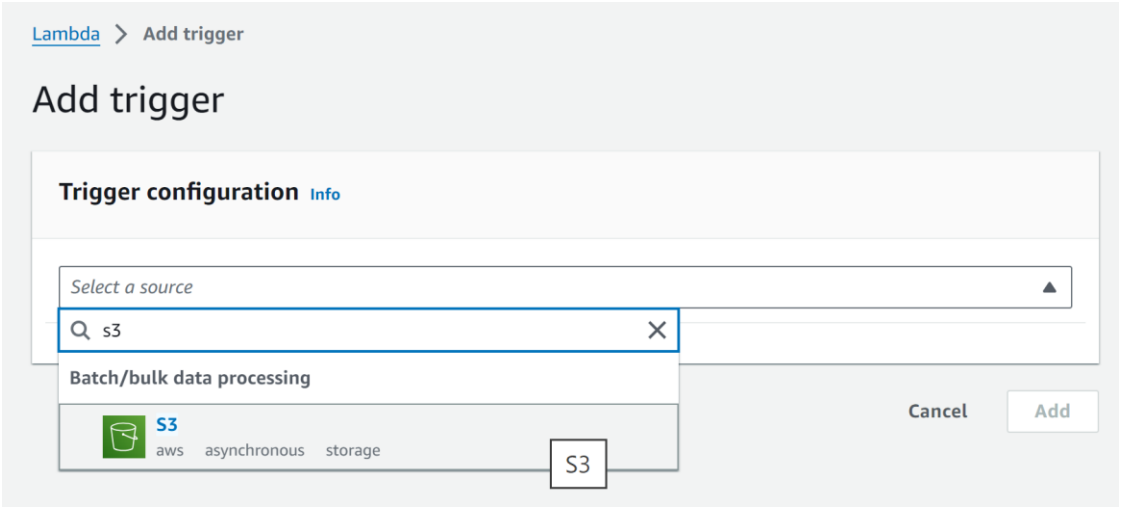
Next step, is to add a Trigger to our Lambda

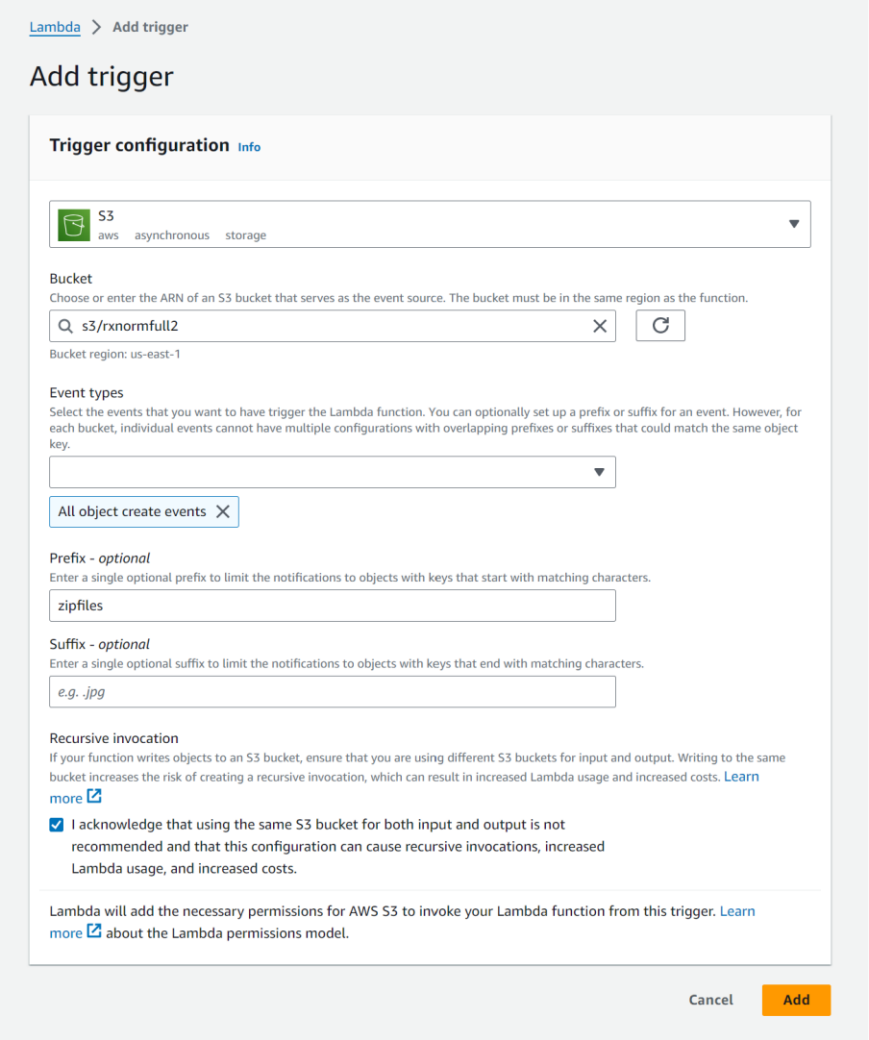


We need to create S3 bucket for this. I have created a bucket.

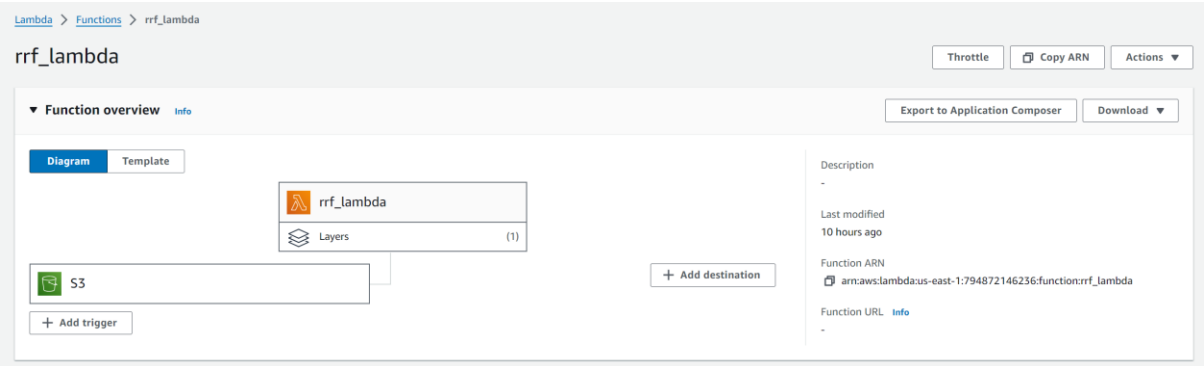


Now go back to Lambda and Select S3 in Trigger Configuration

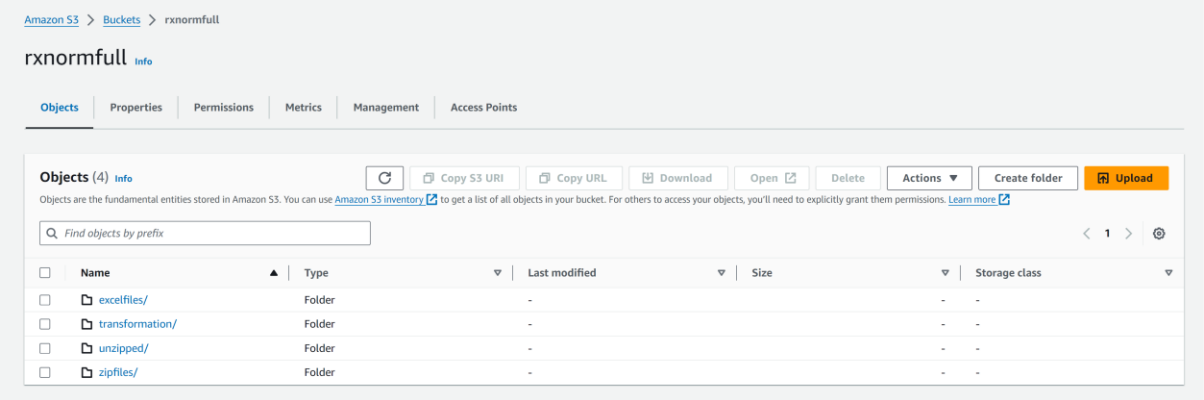




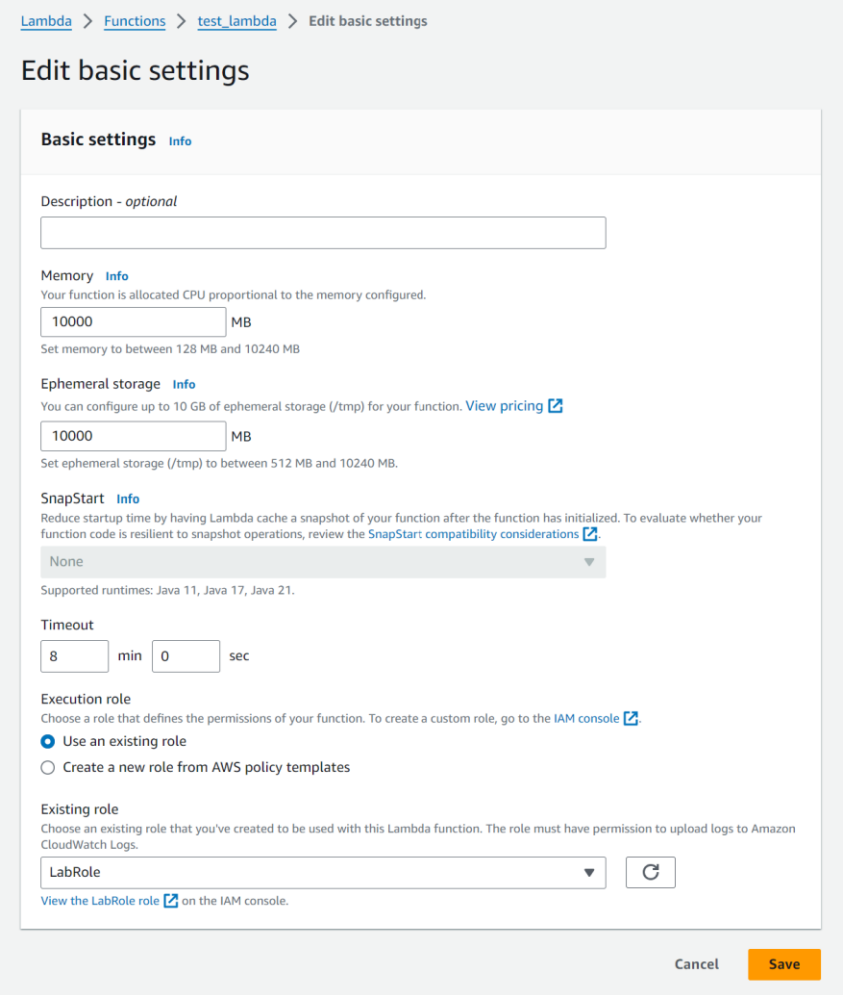
Now this is how our Lambda looks



In our bucket, two folders i.e, excelfiles/ and zipfiles/ are pre-created whereas transformations/ and unzippedfiles/ are created after the lambda is trigger and files are read



We have to



We have to import the following:

lambda_function × Environment Var ×

```
1  import boto3
2  import zipfile
3  import io
4  import os
5  import pandas as pd
6  import json
7  import openpyxl
8  from io import BytesIO
9
```

excel_headers holds the headers to be assigned to the rrf files

---

The following function (read_excel_from_s3) reads the excel files and updates the header

```
def read_excel_from_s3(bucket):
    try:
        folder_path = 'excelfiles/'
        excel_file_name = 'RxNorm_Header.xlsx'
        key = folder_path + excel_file_name

        # Download the Excel file to the /tmp directory
        local_excel_file = '/tmp/RxNorm_Header.xlsx'
        s3.download_file(bucket, key, local_excel_file)

        # Check if the Excel file exists
        if os.path.exists(local_excel_file):
            print(f"Excel file downloaded to: {local_excel_file}")



         # Read the Excel file into an ExcelFile object
        excel_file = pd.ExcelFile(local_excel_file)

           # Get the sheet names
        sheet_names = excel_file.sheet_names
        print("Sheet names:", sheet_names)

        for sheets in sheet_names:
            # Read the data from the sheet into a DataFrame
            sheets_data = excel_file.parse(sheets,header=None)
            headers_data = sheets_data.iloc[:, 0].tolist()
            excel_headers[sheets] = headers_data
        print(f'excel_headers dictionary for sheet {sheet_names[0]}: {excel_headers[sheet_names[0]]}')
```

52:9 Python Spaces: 4

The following function incorporate the Code Set and Version Mont: The Code Set has been designated as RXNORM, and the Version Month has been derived from the zip file name, specifically RxNorm_full_02052024.zip. As a result, the Version Month is identified as 2024-05-02.

Secondly, apply_header_to_rrf adds header to rrf files.

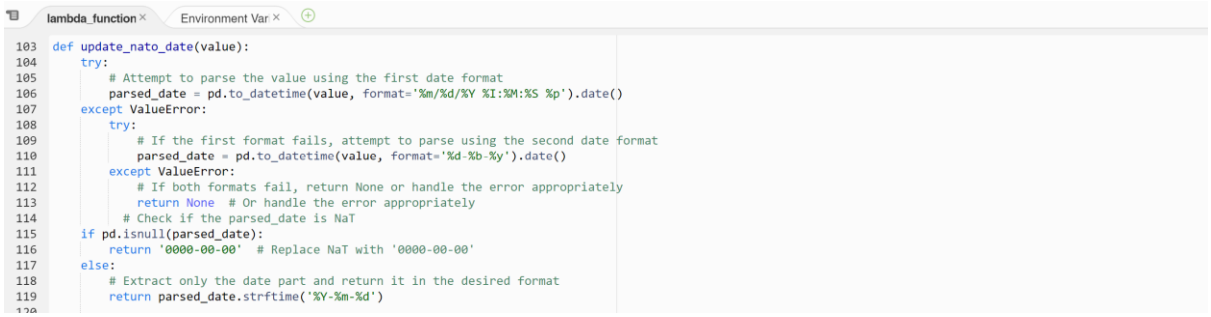
```
def code_set_and_version_month(zip_filename, rrf_df):
    try:
        # Convert the zip data to a string to extract information
        version_month = os.path.splitext(zip_filename)[0].split('_')[-1]

        # Convert version month to a more readable format
        version_month = pd.to_datetime(version_month, format='%m%d%Y').strftime('%Y-%m-%d')
        print(f"Version month: {version_month}")

        # Add 'Code Set' and 'Version Month' columns to the DataFrame
        rrf_df['Code Set'] = 'RxNorm'
        rrf_df['Version Month'] = version_month
    except Exception as e:
        print(f"Error occurred while extracting version month: {e}")

    return rrf_df


def apply_header_to_rrf(file_name, rrf_df):
    # Check if the corresponding Excel sheet exists
    if file_name in excel_headers:
        # Get the headers from the Excel sheet
        excel_headers_list = excel_headers[file_name]
        excel_headers_list = [header for header in excel_headers_list if header != 'SVER']
        # Take names from the excel header list up to the length of the split DataFrame
        excel_headers_list = excel_headers_list[:len(rrf_df.columns)]
        # Set the correct header for the DataFrame
        rrf_df.columns = excel_headers_list
    return rrf_df

```

45:5 Python Spaces: 4

The following function is employed to transform dates into the specified format. In the case of RXNSAB, the SVER column obtains its value by extracting the year component from the VSTART column, as illustrated below:

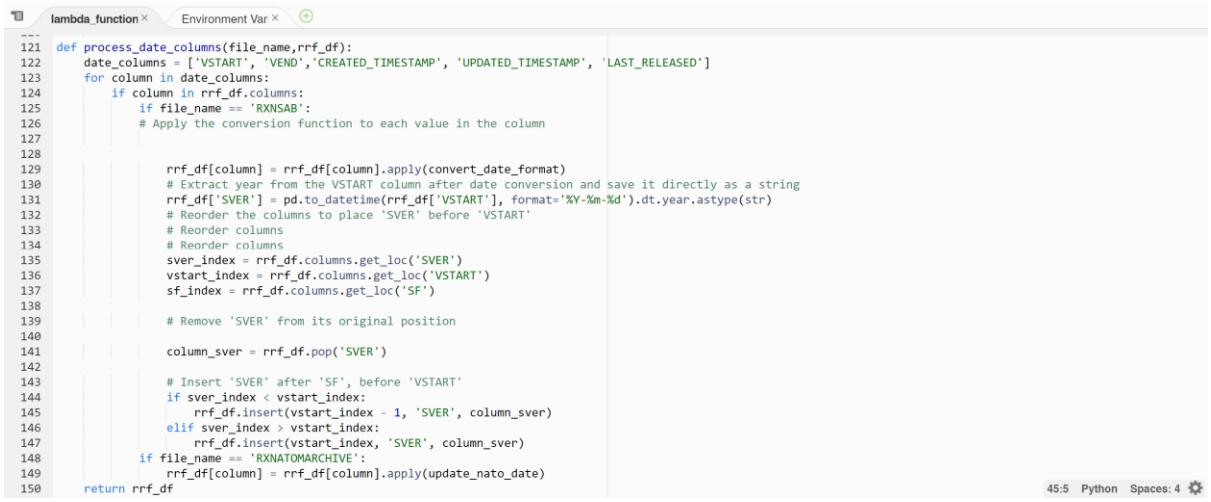

```
def convert_date_format(value):

    try:
        # Try to parse the value into datetime format
        parsed_date = pd.to_datetime(value, format='%Y_%m_%d').date()
        # Extract only the date part
        return parsed_date.strftime('%Y-%m-%d')
    except ValueError:
        if value == '2020':
            return '2020-01-01'
        elif value == '5.0_2024_01_04':
            # Remove the float value and parse the remaining string

            return convert_date_format('2024_01_04')
        elif value == '2020AA':
            return '2024-01-02'
        elif value == '20AA_240205F':
            return '2024-02-05'
        else:
            return value
```

The following function handles different date formats of RXNATOMARCHIVE as below:




```
def update_nato_date(value):
    try:
        # Attempt to parse the value using the first date format
        parsed_date = pd.to_datetime(value, format='%m/%d/%Y %I:%M:%S %p').date()
    except ValueError:
        try:
            # If the first format fails, attempt to parse using the second date format
            parsed_date = pd.to_datetime(value, format='%d-%b-%y').date()
        except ValueError:
            # If both formats fail, return None or handle the error appropriately
            return None  # Or handle the error appropriately
      # Check if the parsed_date is NaT
    if pd.isnull(parsed_date):
        return '0000-00-00'  # Replace NaT with '0000-00-00'
    else:
        # Extract only the date part and return it in the desired format
        return parsed_date.strftime('%Y-%m-%d')
```

The following function processes date columns




```
def process_date_columns(file_name,rrf_df):
    date_columns = ['VSTART', 'VEND','CREATED_TIMESTAMP', 'UPDATED_TIMESTAMP', 'LAST_RELEASED']
    for column in date_columns:
        if column in rrf_df.columns:
            if file_name == 'RXNSAB':
            # Apply the conversion function to each value in the column


                rrf_df[column] = rrf_df[column].apply(convert_date_format)
                # Extract year from the VSTART column after date conversion and save it directly as a string
                rrf_df['SVER'] = pd.to_datetime(rrf_df['VSTART'], format='%Y-%m-%d').dt.year.astype(str)
                # Reorder the columns to place 'SVER' before 'VSTART'
                # Reorder columns
                # Reorder columns
                sver_index = rrf_df.columns.get_loc('SVER')
                vstart_index = rrf_df.columns.get_loc('VSTART')
                sf_index = rrf_df.columns.get_loc('SF')

                # Remove 'SVER' from its original position

                column_sver = rrf_df.pop('SVER')

                # Insert 'SVER' after 'SF', before 'VSTART'
                if sver_index < vstart_index:
                    rrf_df.insert(vstart_index - 1, 'SVER', column_sver)
                elif sver_index > vstart_index:
                    rrf_df.insert(vstart_index, 'SVER', column_sver)
            if file_name == 'RXNATOMARCHIVE':
                rrf_df[column] = rrf_df[column].apply(update_nato_date)
    return rrf_df
```

The following code saves the converted file to .txt

lambda_function × Environment Var ×

```
def save_as_txt_file(rrf_df, file_name, bucket_name):
    # Construct the filename for the output text file
    transformation_folder = 'transformation/'

    # Convert DataFrame to CSV format in memory
    csv_buffer = io.StringIO()
    rrf_df.to_csv(csv_buffer, sep='|', index=False)

    # Upload the CSV buffer to S3
    s3_key = transformation_folder + file_name + '.txt'
    s3.put_object(Bucket=bucket_name, Key=s3_key, Body=csv_buffer.getvalue())

    print(f"Transformed data saved to: s3://{bucket_name}/{s3_key}")   #
```

The primary function where the preceding functions are invoked: Here, zip archives are retrieved from the correct directory, and the final pipe delimiter is disregarded by segmentation.

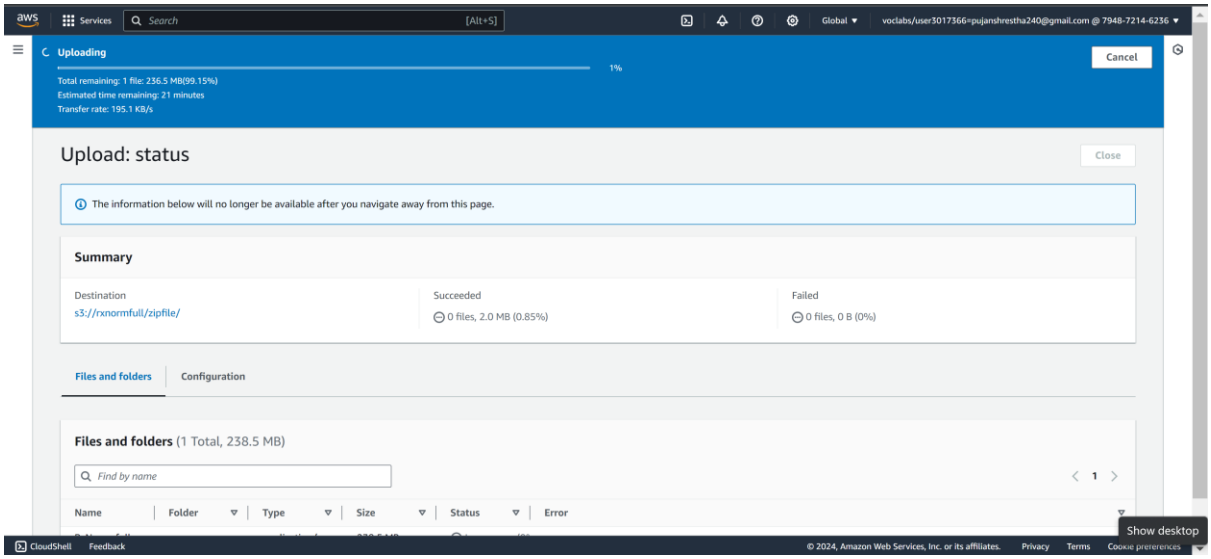
lambda_function × Environment Var ×

```
def read_and_relocate_rrf_files(s3, bucket, key):
    try:
        zip_response = s3.get_object(Bucket=bucket, Key=key)
        zip_data = zip_response['Body'].read()
        zip_filename = os.path.basename(key)

        # Wrap the zip data in a BytesIO object
        zip_file = BytesIO(zip_data)

        file_path = 'test_full_02052024/rrf'

        with zipfile.ZipFile(zip_file, 'r') as zip_ref:
            for file_info in zip_ref.infolist():
                if file_info.filename.startswith(file_path) and not file_info.filename.endswith('/'):
                    filename = os.path.basename(file_info.filename)
                    print(f"The {filename} is read from zip file.")

                    with zip_ref.open(file_info) as source_file:
                        file_content = source_file.read().decode('utf-8')
                        if file_content.endswith('|'):
                            file_content = file_content[:-1]
                        file_content_io = io.StringIO(file_content)
                        rrf_df = pd.read_csv(file_content_io, delimiter='|', header=None)
                        rrf_df = rrf_df.iloc[:, :-1]
                        print(f"Row count before transformation: {rrf_df.shape[0]}")

                        file_name = os.path.splitext(filename)[0]
                        apply_header_to_rrf(file_name, rrf_df)
                        rrf_df = process_date_columns(file_name, rrf_df)
                        code_set_and_version_month(zip_filename, rrf_df)

                        print(f"Row count of {file_name} after transformation: {rrf_df.shape[0]}")

                        pd.set_option('display.max_columns', None)
                        print(rrf_df.head(5))

                        # Save the transformed DataFrame to a text file
                        save_as_txt_file(rrf_df, file_name, bucket)

                        # Upload the unzipped file to the 'unzipped' folder
                        unzipped_key = 'unzipped/' + filename
                        s3.put_object(Bucket=bucket, Key=unzipped_key, Body=file_content)
                        print(f"Unzipped file saved to: s3://{bucket}/{unzipped_key}")
    except Exception as e:
        print(f"Error occurred: {e}")
```

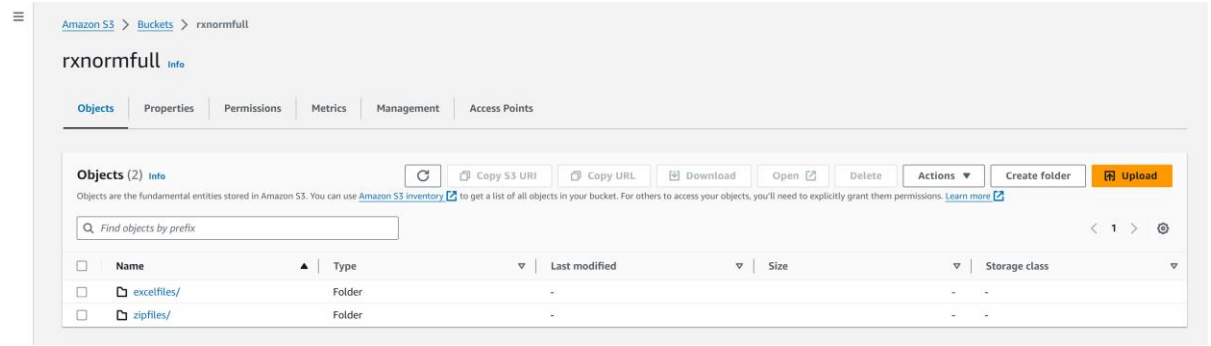
The following is the lambda_handler

```
def lambda_handler(event, context):
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = event['Records'][0]['s3']['object']['key']


    # This is the function that relocate the rrf files from zip file
    read_excel_from_s3(bucket)
    read_and_relocate_rrf_files(s3,bucket,key)
```

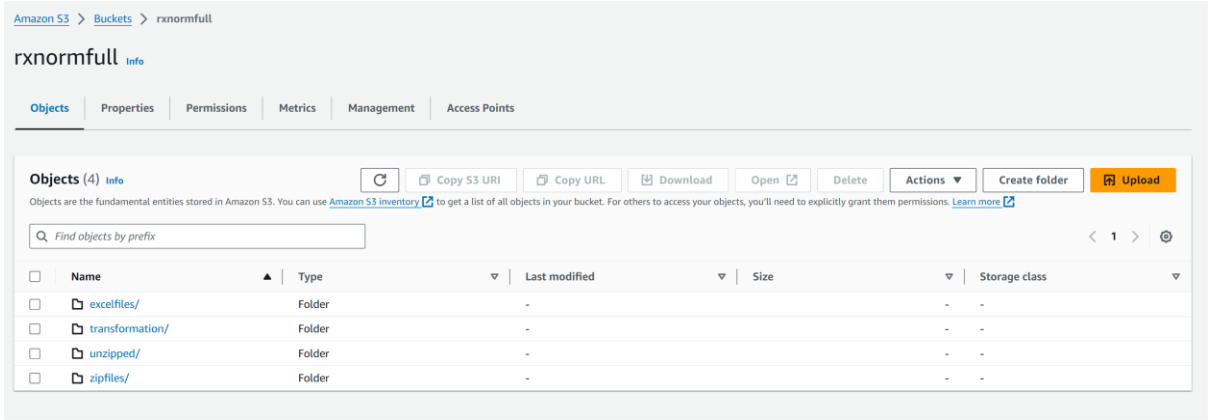
Uploading zip file



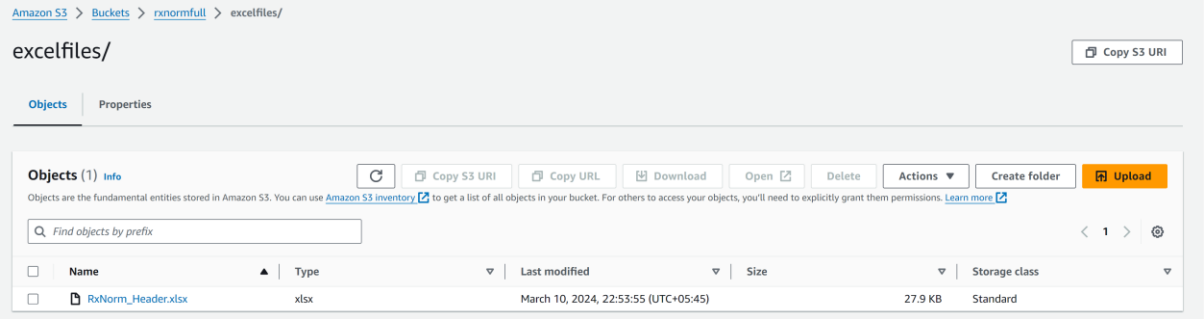
Initial rxnorm



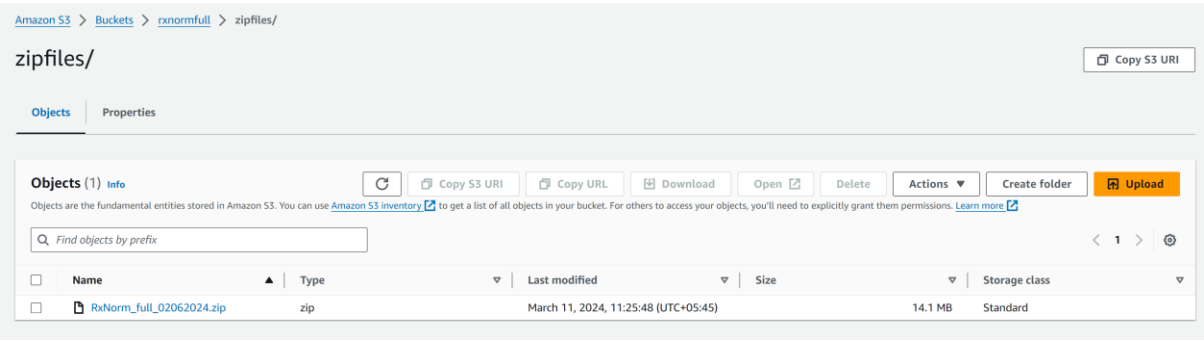
After lambda is triggered:



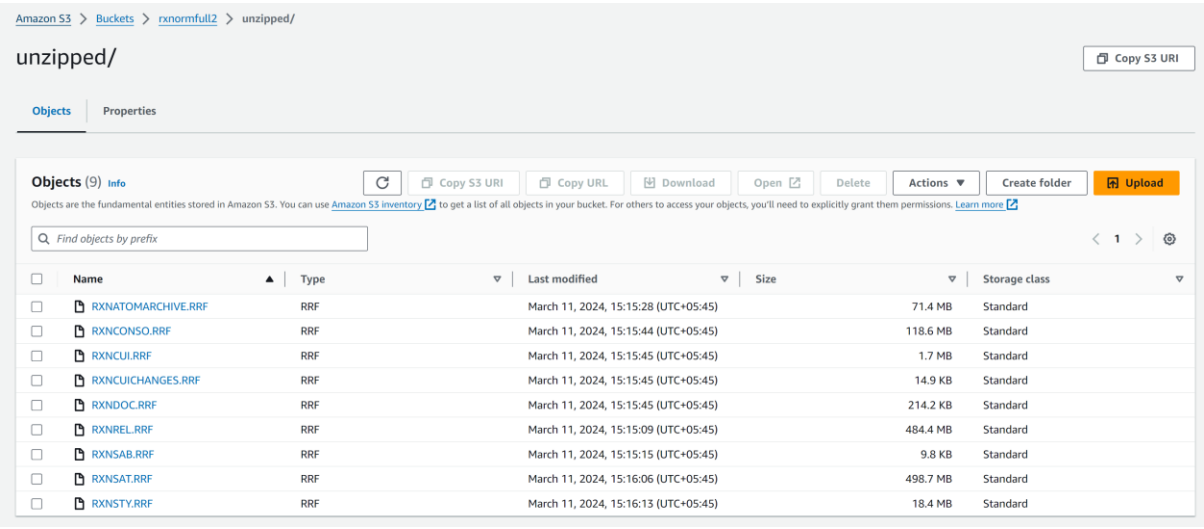
excelfiles/



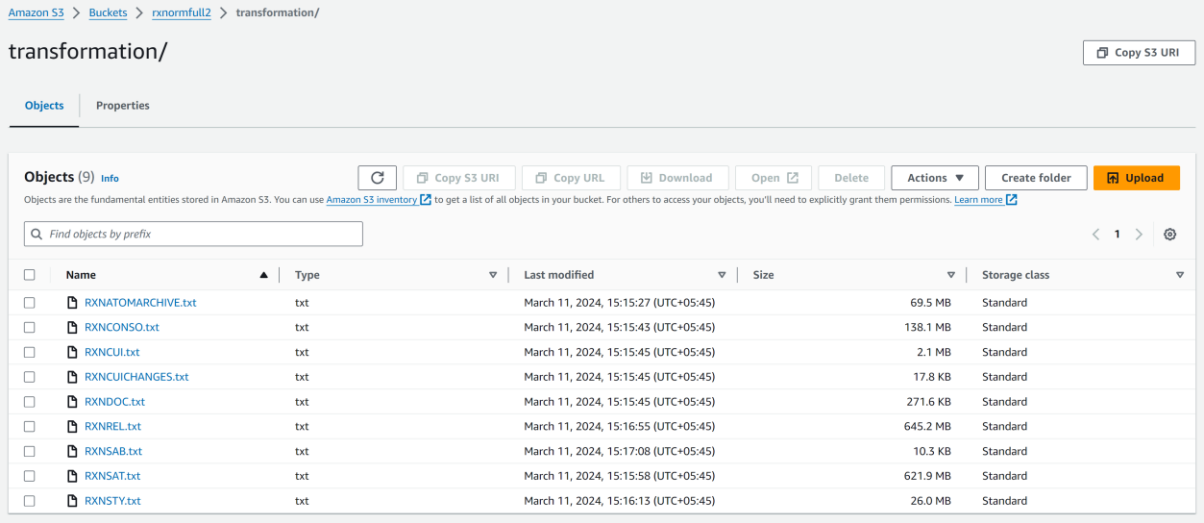
zippedfiles/



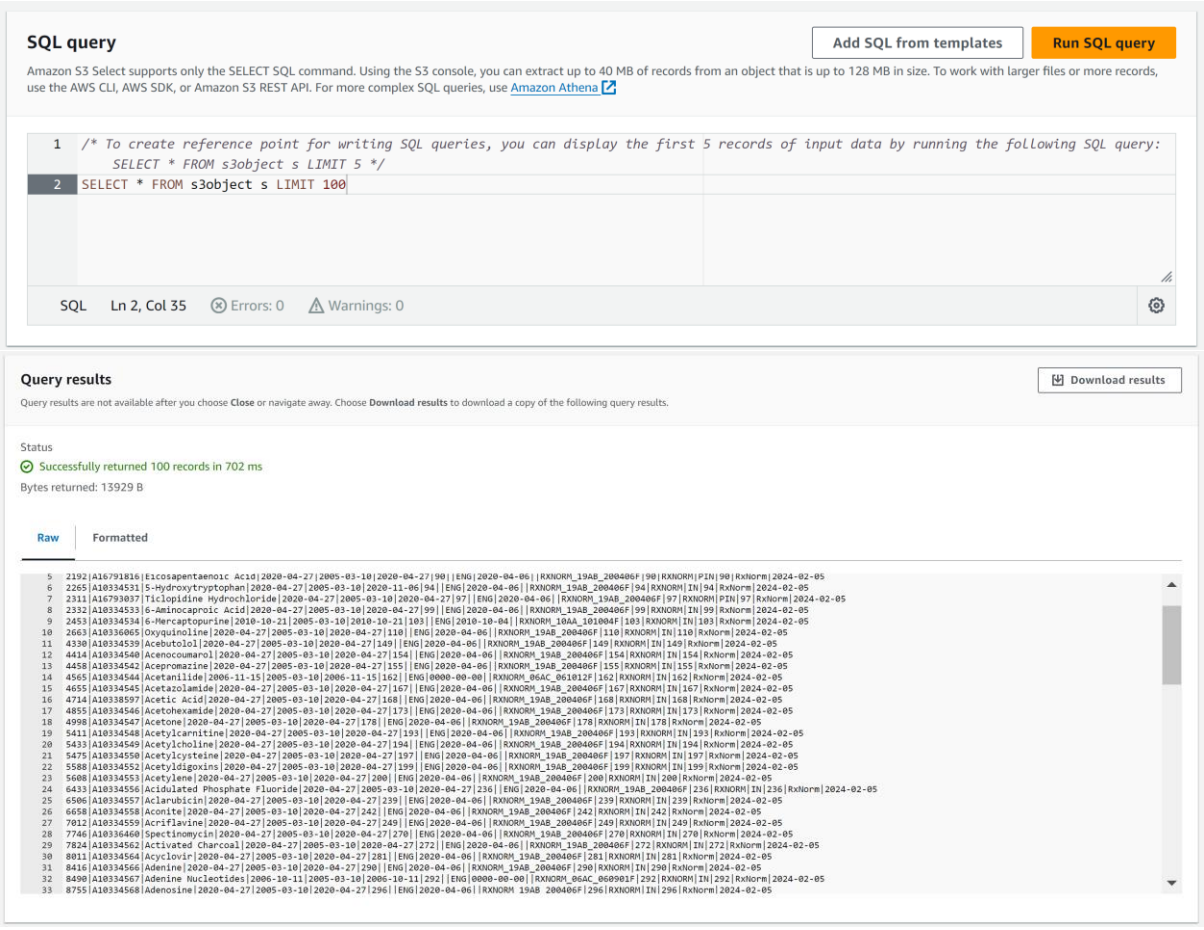
Unzippedfiles/



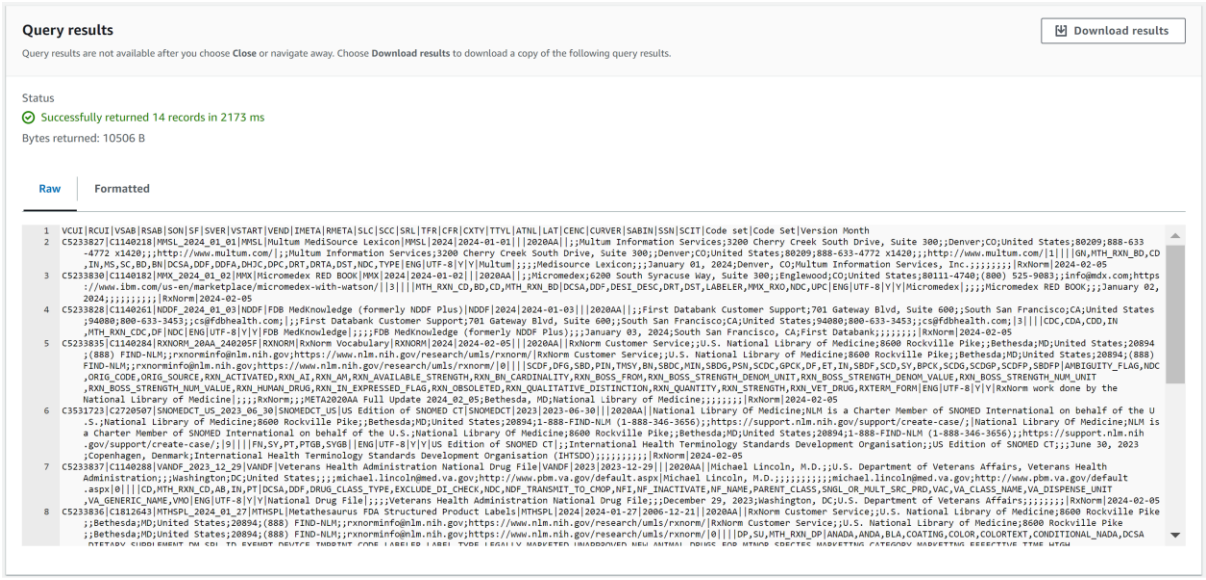
Transformations/



Here is the RXNATOMARCHIVE.rrf files after transformation of dates in the txt format



Here is the RXNSAB.rrf files transformed after adding SVER column and changing the date format.



Finally, we validate the CloudWatch logs:

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

Filter events | Clear

Timestamp	Message
	There are older events to load. *Load more.*
2024-03-11T15:15:23.511+05:45	1 2024-02-05
2024-03-11T15:15:23.511+05:45	2 2024-02-05
2024-03-11T15:15:23.511+05:45	3 2024-02-05
2024-03-11T15:15:23.511+05:45	4 2024-02-05
2024-03-11T15:15:27.321+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNATOMARCHIVE.txt
2024-03-11T15:15:28.005+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNATOMARCHIVE.RRF
2024-03-11T15:15:28.005+05:45	The RXNCONSO.RRF is read from zip file.
2024-03-11T15:15:33.125+05:45	Row count before transformation: 1133065
2024-03-11T15:15:33.125+05:45	Version month: 2024-02-05
2024-03-11T15:15:33.137+05:45	Row count of RXNCONSO after transformation: 1133065
2024-03-11T15:15:33.141+05:45	RXCUI LAT TS LUI STT SUI ISPREF RXAUI SAUI SCUI SDUI \
2024-03-11T15:15:33.141+05:45	0 3 ENG NaN NaN NaN NaN NaN 8717795 NaN 58488005 NaN
2024-03-11T15:15:33.141+05:45	1 3 ENG NaN NaN NaN NaN NaN 8717796 NaN 58488005 NaN
2024-03-11T15:15:33.141+05:45	2 3 ENG NaN NaN NaN NaN NaN 8717808 NaN 58488005 NaN
2024-03-11T15:15:33.141+05:45	3 3 ENG NaN NaN NaN NaN NaN 8718164 NaN 58488005 NaN
2024-03-11T15:15:33.141+05:45	4 19 ENG NaN NaN NaN NaN NaN 10794494 NaN 112116001 NaN

▶	2024-03-11T15:15:33.141+05:45	SAB TTY CODE STR \
▶	2024-03-11T15:15:33.141+05:45	0 SNOMEDCT_US PT 58488005 1,4-alpha-Glucan branching enzyme
▶	2024-03-11T15:15:33.141+05:45	1 SNOMEDCT_US FN 58488005 1,4-alpha-Glucan branching enzyme (substance)
▶	2024-03-11T15:15:33.141+05:45	2 SNOMEDCT_US SY 58488005 Amylo-(1,4,6)-transglycosylase
▶	2024-03-11T15:15:33.141+05:45	3 SNOMEDCT_US SY 58488005 Branching enzyme
▶	2024-03-11T15:15:33.141+05:45	4 SNOMEDCT_US SY 112116001 17-hydrocorticosteroid
▶	2024-03-11T15:15:33.141+05:45	SRL SUPPRESS CVF Code Set Version Month
▶	2024-03-11T15:15:33.141+05:45	0 NaN N NaN RxNorm 2024-02-05
▶	2024-03-11T15:15:33.141+05:45	1 NaN N NaN RxNorm 2024-02-05
▶	2024-03-11T15:15:33.141+05:45	2 NaN N NaN RxNorm 2024-02-05
▶	2024-03-11T15:15:33.141+05:45	3 NaN N NaN RxNorm 2024-02-05
▶	2024-03-11T15:15:33.141+05:45	4 NaN N NaN RxNorm 2024-02-05
▶	2024-03-11T15:15:42.936+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNCONSO.txt
▶	2024-03-11T15:15:43.948+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNCONSO.RRF
▶	2024-03-11T15:15:43.948+05:45	The RXNCUICHANGES.RRF is read from zip file.
▶	2024-03-11T15:15:44.041+05:45	Row count before transformation: 153
▶	2024-03-11T15:15:44.042+05:45	Version month: 2024-02-05
▶	2024-03-11T15:15:44.042+05:45	Row count of RXNCUICHANGES after transformation: 153
▶	2024-03-11T15:15:44.044+05:45	RXAUI CODE SAB TTY STR OLD_RXCUI NEW_RXCUI Code Set Version Month
▶	2024-03-11T15:15:44.044+05:45	0 NaN NaN RXNORM NaN NaN 1653118 2672916 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.044+05:45	1 NaN NaN RXNORM NaN NaN 1653119 2672917 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.044+05:45	2 NaN NaN RXNORM NaN NaN 1806685 2672352 RxNorm 2024-02-05

▶	2024-03-11T15:15:44.044+05:45	3 NaN NaN RXNORM NaN NaN 1806938 2673304 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.044+05:45	4 NaN NaN RXNORM NaN NaN 2647628 2672353 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.081+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNCUICHANGES.txt
▶	2024-03-11T15:15:44.133+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNCUICHANGES.RRF
▶	2024-03-11T15:15:44.133+05:45	The RXNCUI.RRF is read from zip file.
▶	2024-03-11T15:15:44.159+05:45	Row count before transformation: 30046
▶	2024-03-11T15:15:44.160+05:45	Version month: 2024-02-05
▶	2024-03-11T15:15:44.160+05:45	Row count of RXNCUI after transformation: 30046
▶	2024-03-11T15:15:44.162+05:45	CUI1 VER_START VER_END Cardinality CUI2 \
▶	2024-03-11T15:15:44.162+05:45	0 89904 RXNORM_17AA_170905F RXNORM_20AA_221003F 1 89904
▶	2024-03-11T15:15:44.162+05:45	1 91669 RXNORM_17AA_170905F RXNORM_18AB_181203F 1 91669
▶	2024-03-11T15:15:44.162+05:45	2 91714 RXNORM_10AA_100802F RXNORM_10AA_100802F 1 1000992
▶	2024-03-11T15:15:44.162+05:45	3 91811 RXNORM_04AC_050210F RXNORM_09AA_090706F 1 857082
▶	2024-03-11T15:15:44.162+05:45	4 91812 RXNORM_04AC_050210F RXNORM_09AA_090706F 1 857087
▶	2024-03-11T15:15:44.162+05:45	Code Set Version Month
▶	2024-03-11T15:15:44.162+05:45	0 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.162+05:45	1 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.162+05:45	2 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.162+05:45	3 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.162+05:45	4 RxNorm 2024-02-05
▶	2024-03-11T15:15:44.363+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNCUI.txt
▶	2024-03-11T15:15:44.479+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNCUI.RRF

▶	2024-03-11T15:15:44.479+05:45	The RXNDOC.RRF is read from zip file.
▶	2024-03-11T15:15:44.486+05:45	Row count before transformation: 3445
▶	2024-03-11T15:15:44.487+05:45	Version month: 2024-02-05
▶	2024-03-11T15:15:44.487+05:45	Row count of RXNDOC after transformation: 3445
▶	2024-03-11T15:15:44.489+05:45	KEY VALUE TYPE \
▶	2024-03-11T15:15:44.489+05:45	0 ATN AAL_TERM expanded_form
▶	2024-03-11T15:15:44.489+05:45	1 ATN ACCEPTABILITYID expanded_form
▶	2024-03-11T15:15:44.489+05:45	2 ATN ACCEPTED_THERAPEUTIC_USE_FOR expanded_form
▶	2024-03-11T15:15:44.489+05:45	3 ATN ACTIVE expanded_form
▶	2024-03-11T15:15:44.489+05:45	4 ATN ADDED_MEANING expanded_form
▶	2024-03-11T15:15:44.489+05:45	EXPL Code Set Version Month
▶	2024-03-11T15:15:44.489+05:45	0 AAL term RxNorm 2024-02-05
▶	2024-03-11T15:15:44.489+05:45	1 Acceptability Id RxNorm 2024-02-05
▶	2024-03-11T15:15:44.489+05:45	2 Accepted therapeutic use for RxNorm 2024-02-05
▶	2024-03-11T15:15:44.489+05:45	3 Active RxNorm 2024-02-05
▶	2024-03-11T15:15:44.489+05:45	4 Additional descriptive information RxNorm 2024-02-05
▶	2024-03-11T15:15:44.587+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNDOC.txt
▶	2024-03-11T15:15:44.652+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNDOC.RRF
▶	2024-03-11T15:15:44.652+05:45	The RXNREL.RRF is read from zip file.
▶	2024-03-11T15:16:01.421+05:45	Row count before transformation: 7154306
▶	2024-03-11T15:16:01.422+05:45	Version month: 2024-02-05
▶	2024-03-11T15:16:01.512+05:45	Row count of RXNREL after transformation: 7154306

▶	2024-03-11T15:16:01.517+05:45	RXCUI1 RXAUI1 STYPE1 REL RXCUI2 RXAUI2 STYPE2 \
▶	2024-03-11T15:16:01.517+05:45	0 NaN 828.0 AUI RO NaN 788773.0 AUI
▶	2024-03-11T15:16:01.517+05:45	1 NaN 828.0 AUI RO NaN 790716.0 AUI
▶	2024-03-11T15:16:01.517+05:45	2 NaN 829.0 AUI SY NaN 828.0 AUI
▶	2024-03-11T15:16:01.517+05:45	3 NaN 945.0 AUI RO NaN 2369805.0 AUI
▶	2024-03-11T15:16:01.517+05:45	4 NaN 950.0 AUI RO NaN 9251787.0 AUI
▶	2024-03-11T15:16:01.517+05:45	RELA RUI SRUI SAB SL RG \
▶	2024-03-11T15:16:01.517+05:45	0 has_ingredient 5066189.0 NaN MMSL NaN NaN
▶	2024-03-11T15:16:01.517+05:45	1 has_ingredient 3823969.0 NaN MMSL NaN NaN
▶	2024-03-11T15:16:01.517+05:45	2 NaN NaN NaN RXNORM NaN NaN
▶	2024-03-11T15:16:01.517+05:45	3 has_ingredient 4306227.0 NaN MMSL NaN NaN
▶	2024-03-11T15:16:01.517+05:45	4 has_basis_of_strength_substance 116141227.0 NaN SNOMEDCT_US NaN NaN
▶	2024-03-11T15:16:01.517+05:45	DIR SUPPRESS CVF Code Set Version Month
▶	2024-03-11T15:16:01.517+05:45	0 NaN NaN NaN RxNorm 2024-02-05
▶	2024-03-11T15:16:01.517+05:45	1 NaN NaN NaN RxNorm 2024-02-05
▶	2024-03-11T15:16:01.517+05:45	2 NaN NaN NaN RxNorm 2024-02-05
▶	2024-03-11T15:16:01.517+05:45	3 NaN NaN NaN RxNorm 2024-02-05
▶	2024-03-11T15:16:01.517+05:45	4 NaN NaN NaN RxNorm 2024-02-05
▶	2024-03-11T15:17:01.034+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNREL.txt
▶	2024-03-11T15:17:07.625+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNREL.RRF
▶	2024-03-11T15:17:07.625+05:45	The RXNSAB.RRF is read from zip file.

▶	2024-03-11T15:17:07.854+05:45	Row count before transformation: 13
▶	2024-03-11T15:17:07.861+05:45	Version month: 2024-02-05
▶	2024-03-11T15:17:07.861+05:45	Row count of RXNSAB after transformation: 13
▶	2024-03-11T15:17:07.866+05:45	VCUI RCUI VSAB RSAB \
▶	2024-03-11T15:17:07.866+05:45	0 C5233827 C1140218 MMSL_2024_01_01 MMSL
▶	2024-03-11T15:17:07.866+05:45	1 C5233830 C1140182 MMX_2024_01_02 MMX
▶	2024-03-11T15:17:07.866+05:45	2 C5233828 C1140261 NDDF_2024_01_03 NDDF
▶	2024-03-11T15:17:07.866+05:45	3 C5233835 C1140284 RXNORM_20AA_240205F RXNORM
▶	2024-03-11T15:17:07.866+05:45	4 C3531723 C2720507 SNOMEDCT_US_2023_06_30 SNOMEDCT_US
▶	2024-03-11T15:17:07.866+05:45	SON SF SVER VSTART VEND \
▶	2024-03-11T15:17:07.866+05:45	0 Multum MediSource Lexicon MMSL 2024 2024-01-01 NaN
▶	2024-03-11T15:17:07.866+05:45	1 Micromedex RED BOOK MMX 2024 2024-01-02 NaN
▶	2024-03-11T15:17:07.866+05:45	2 FDB MedKnowledge (formerly NDDF Plus) NDDF 2024 2024-01-03 NaN
▶	2024-03-11T15:17:07.866+05:45	3 RxNorm Vocabulary RXNORM 2024 2024-02-05 NaN
▶	2024-03-11T15:17:07.866+05:45	4 US Edition of SNOMED CT SNOMEDCT 2023 2023-06-30 NaN
▶	2024-03-11T15:17:07.866+05:45	IMETA RMETA SLC SCC \
▶	2024-03-11T15:17:07.866+05:45	0 NaN 2020AA NaN ;;Multum Information Services;3200 Cherry Cree...
▶	2024-03-11T15:17:07.866+05:45	1 NaN 2020AA NaN ;;Micromedex;6200 South Syracuse Way, Suite 30...
▶	2024-03-11T15:17:07.866+05:45	2 NaN 2020AA NaN ;;First Databank Customer Support;701 Gateway ...
▶	2024-03-11T15:17:07.866+05:45	3 NaN 2020AA NaN RxNorm Customer Service;;U.S. National Library...
▶	2024-03-11T15:17:07.866+05:45	4 NaN 2020AA NaN National Library Of Medicine;NLM is a Charter ...
▶	2024-03-11T15:17:07.866+05:45	SRL TFR CFR CXTY TTYL \

▶	2024-03-11T15:17:07.866+05:45	0 ;;Multum Information Services;3200 Cherry Cree... 1 NaN NaN NaN
▶	2024-03-11T15:17:07.866+05:45	1 NaN 3 NaN NaN NaN
▶	2024-03-11T15:17:07.866+05:45	2 ;;First Databank Customer Support;701 Gateway ... 3 NaN NaN NaN
▶	2024-03-11T15:17:07.866+05:45	3 RxNorm Customer Service;;U.S. National Library... 0 NaN NaN NaN
▶	2024-03-11T15:17:07.866+05:45	4 National Library Of Medicine;NLM is a Charter ... 9 NaN NaN NaN
▶	2024-03-11T15:17:07.866+05:45	ATNL \
▶	2024-03-11T15:17:07.866+05:45	0 GN,MTH_RXN_BD,CD,IN,MS,SC,BD,BN
▶	2024-03-11T15:17:07.866+05:45	1 MTH_RXN_CD,BD,CD,MTH_RXN_BD
▶	2024-03-11T15:17:07.866+05:45	2 CDC,CDA,CDD,IN,MTH_RXN_CDC,DF
▶	2024-03-11T15:17:07.866+05:45	3 SCDF,DFG,SBD,PIN,TMSY,BN,SBDC,MIN,SBDG,PSN,SCD...
▶	2024-03-11T15:17:07.866+05:45	4 FN,SY,PT,PTGB,SYGB
▶	2024-03-11T15:17:07.866+05:45	LAT CENC CURVER SABIN SSN \
▶	2024-03-11T15:17:07.866+05:45	0 DCSA,DDF,DDFA,DHJC,DPC,DRT,DRTA,DST,NDC,TYPE ENG UTF-8 Y Y
▶	2024-03-11T15:17:07.866+05:45	1 DCSA,DDF,DESI_DESC,DRT,DST,LABELER,MMX_RXO,NDC... ENG UTF-8 Y Y
▶	2024-03-11T15:17:07.866+05:45	2 NDC ENG UTF-8 Y Y
▶	2024-03-11T15:17:07.866+05:45	3 AMBIGUITY_FLAG,NDC,ORIG_CODE,ORIG_SOURCE,RXN_A... ENG UTF-8 Y Y
▶	2024-03-11T15:17:07.866+05:45	4 NaN ENG UTF-8 Y Y
▶	2024-03-11T15:17:07.866+05:45	SCIT \
▶	2024-03-11T15:17:07.866+05:45	0 Multum
▶	2024-03-11T15:17:07.866+05:45	1 Micromedex
▶	2024-03-11T15:17:07.866+05:45	2 FDB MedKnowledge
▶	2024-03-11T15:17:07.866+05:45	3 RxNorm work done by the National Library of Me...

▶	2024-03-11T15:17:07.866+05:45	4 US Edition of SNOMED CT
▶	2024-03-11T15:17:07.866+05:45	Code set Code Set Version Month
▶	2024-03-11T15:17:07.866+05:45	0 ;;;;Medisource Lexicon;;;January 01, 2024;Denv... RxNorm 2024-02-05
▶	2024-03-11T15:17:07.866+05:45	1 ;;;;Micromedex RED BOOK;;;January 02, 2024;;;;... RxNorm 2024-02-05
▶	2024-03-11T15:17:07.866+05:45	2 ;;;;FDB MedKnowledge (formerly NDDF Plus);;;Ja... RxNorm 2024-02-05
▶	2024-03-11T15:17:07.866+05:45	3 ;;;;RxNorm;;;META2020AA Full Update 2024_02_05... RxNorm 2024-02-05
▶	2024-03-11T15:17:07.866+05:45	4 ;;International Health Terminology Standards D... RxNorm 2024-02-05
▶	2024-03-11T15:17:07.895+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNSAB.txt
▶	2024-03-11T15:17:07.948+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNSAB.RRF
▶	2024-03-11T15:17:07.948+05:45	The RXNSAT.RRF is read from zip file.
▶	2024-03-11T15:17:24.932+05:45	Row count before transformation: 7222404
▶	2024-03-11T15:17:24.932+05:45	Version month: 2024-02-05
▶	2024-03-11T15:17:25.023+05:45	Row count of RXNSAT after transformation: 7222404
▶	2024-03-11T15:17:25.026+05:45	RXCUI LUI SUI RXAUI STYPE CODE ATUI SATUI ATN \
▶	2024-03-11T15:17:25.026+05:45	0 38 NaN NaN 829 AUI 38 NaN NaN RXN_BN_CARDINALITY
▶	2024-03-11T15:17:25.026+05:45	1 44 NaN NaN 955 AUI d01411 NaN NaN DPC
▶	2024-03-11T15:17:25.026+05:45	2 44 NaN NaN 2798745 AUI NR7O1405Q9 NaN NaN SPL_SET_ID
▶	2024-03-11T15:17:25.026+05:45	3 44 NaN NaN 2982613 AUI NR7O1405Q9 NaN NaN SPL_SET_ID
▶	2024-03-11T15:17:25.026+05:45	4 44 NaN NaN 2982613 AUI NR7O1405Q9 NaN NaN SPL_SET_ID
▶	2024-03-11T15:17:25.026+05:45	SAB ATV SUPPRESS CVF Code Set \
▶	2024-03-11T15:17:25.026+05:45	0 RXNORM single N 4096.0 RxNorm
▶	2024-03-11T15:17:25.026+05:45	1 MMSL N N NaN RxNorm

2024-03-11T15:17:25.026+05:45	2 MTHSPL 2a2a526f-636b-7920-4261-6c626f612a2a N 4096.0 RxNorm
2024-03-11T15:17:25.026+05:45	3 MTHSPL 2e8eebc3-ea75-4c57-bfe1-2c5dc2c37806 N 4096.0 RxNorm
2024-03-11T15:17:25.026+05:45	4 MTHSPL 49fdfbc3-69f8-4706-9d11-bf60e4e7811b N 4096.0 RxNorm
2024-03-11T15:17:25.026+05:45	Version Month
2024-03-11T15:17:25.026+05:45	0 2024-02-05
2024-03-11T15:17:25.026+05:45	1 2024-02-05
2024-03-11T15:17:25.026+05:45	2 2024-02-05
2024-03-11T15:17:25.026+05:45	3 2024-02-05
2024-03-11T15:17:25.026+05:45	4 2024-02-05
2024-03-11T15:18:11.032+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNSAT.txt
2024-03-11T15:18:17.738+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNSAT.RRF
2024-03-11T15:18:17.738+05:45	The RXNSTY.RRF is read from zip file.
2024-03-11T15:18:18.462+05:45	Row count before transformation: 461874
2024-03-11T15:18:18.462+05:45	Version month: 2024-02-05
2024-03-11T15:18:18.467+05:45	Row count of RXNSTY after transformation: 461874
2024-03-11T15:18:18.470+05:45	RXCUI TUI STN STY ATUI CVF \
2024-03-11T15:18:18.470+05:45	0 3 T126 A1.4.1.1.3.3 Enzyme NaN NaN
2024-03-11T15:18:18.470+05:45	1 3 T116 A1.4.1.2.1.7 Amino Acid, Peptide, or Protein NaN NaN
2024-03-11T15:18:18.470+05:45	2 19 T125 A1.4.1.1.3.2 Hormone NaN NaN
2024-03-11T15:18:18.470+05:45	3 19 T109 A1.4.1.2.1 Organic Chemical NaN NaN
2024-03-11T15:18:18.470+05:45	4 21 T125 A1.4.1.1.3.2 Hormone NaN NaN
2024-03-11T15:18:18.470+05:45	Code Set Version Month
2024-03-11T15:18:18.470+05:45	0 RxNorm 2024-02-05
2024-03-11T15:18:18.470+05:45	1 RxNorm 2024-02-05
2024-03-11T15:18:18.470+05:45	2 RxNorm 2024-02-05
2024-03-11T15:18:18.470+05:45	3 RxNorm 2024-02-05
2024-03-11T15:18:18.470+05:45	4 RxNorm 2024-02-05
2024-03-11T15:18:20.072+05:45	Transformed data saved to: s3://rxnormfull2/transformation/RXNSTY.txt
2024-03-11T15:18:20.391+05:45	Unzipped file saved to: s3://rxnormfull2/unzipped/RXNSTY.RRF
2024-03-11T15:18:20.414+05:45	END RequestId: 69ed8773-2ad3-4253-8698-c202a72f4ed9
2024-03-11T15:18:20.414+05:45	REPORT RequestId: 69ed8773-2ad3-4253-8698-c202a72f4ed9 Duration: 289681.64 ms Billed Duration: 289682 ms Memory Size: 10000 MB Max Memory Used: 7672 MB Init Duration: 2877.71 ms
	No newer events at this moment. *Auto retry paused.* ***Resume***