In [1]:
```
import os
import pandas as pd
import numpy as np
from sqlalchemy import create_engine
from sqlalchemy.types import Integer, Text, String, Float, DateTime
from datetime import datetime
```

In [2]:
```
def build_DB_URI(db_type, db_lib, user_id, password, db_name,  db_location='localhost', port='5432' ):
    '''
        A method which generates a DB_URI for SQL-Alchemey. Assumption that this will be
        used with Postgresql, however written to be generic.

        arg:

        db_type     --> the type of database, e.g 'postgres', 'mysql'

        db_lib      --> the appropriate sql-alchemy plughin for
                        db_type, e.g 'psycopg2' or 'pymysql'

        user_id     --> the user name for the database, who has
                        appropriate permissions

        password    --> the password for the db-user-id.
        db_name     --> the name of the db, e.g. 'esomeprazole'
        db_location --> the address / URL for the database. DEFAULT = localhost
        port        --> the port for the database. DEFAULT = 5432

        returns:
        db_URI     --> The URI for SQL-Alchemy of the form:
                       postgres+psycop2://user_id:password@db_location:5432/db_name

    '''

    db_URI = db_type+'+'+db_lib+'://'+user_id+':'+password+'@'+db_location+':'+port+'/'+db_name

    return db_URI
```

====

In [3]:
```
def get_db_cols(df):
    '''
    A method which converts the col-dtype from Pandas/Numpy
    to the SQLAlchemy equivelent.
    args:

    df ---> A pandas DataFrame


    returns:

    db_cols --> a dictionary with column-name as key and SQL-Alchemy
                data type as values.

    '''

    col_info = dict(df.dtypes)
    db_cols = {}
    for k in col_info:

        if col_info[k] == 'object':
            db_cols[k] = String

        elif col_info[k] == 'int64':
            db_cols[k] = Integer

        elif col_info[k] == 'float64':
            db_cols[k] = Float
        elif col_info[k] == 'string':
            db_cols[k] = String
        elif col_info[k] == 'datetime':
            db_cols[k] = DateTime
        else:
            print('Unaccounted for type:')
            print(k, col_info[k])
            return None
    return db_cols
```

====

In [4]:
```
def load_csv_file_as_df(data_file_path, file_name):
    #print('loading csv into a df', data_file_path, file_name)
    df = pd.read_csv(data_file_path+file_name)

    new_columns = [column.replace(' ', '_').lower() for column in df]
    df.columns = new_columns
    return df
```

====

In [5]:
```
def get_file_names(data_file_path, file_name_pattern):
#     print("getting the file names..")
    all_file_list = os.listdir(data_file_path)

    all_file_list.sort()
    #print(len(all_file_list))

    file_list = []

    for f in all_file_list:

        if file_name_pattern in f:
            file_list.append(f)

    return  file_list
```

====

In [65]:
```
def set_col_types(df, col_type_dict):

    print('set_col_types start')

    for col in list(df.columns):

        try:

            if col_type_dict[col] == 'string':
             #   print('got a string', col)
                df[col].fillna('', inplace=True)
                df[col] = df[col].astype(str)
                continue
            if col_type_dict[col] == 'int':
              #  print('got an int', col)
                df[col].fillna(0, inplace=True)
                df[col] = df[col].astype(int)
                continue
            if col_type_dict[col] == 'float':
               # print('got a float', col)
                df[col].fillna(0.0, inplace=True)
                df[col] = df[col].astype(float)
                continue
            if col_type_dict[col] == 'datetime':
                df[col].fillna(19990101, inplace=True)

                df[col] = pd.to_datetime(df[col], format='%Y-%m-%d').dt.strftim
e('%Y-%m-%d')

            else:
                raise TypeError('no condition for this type: ', col, col_type_d
ict[col] )
        except:
            print(col, col_type_dict[col] )
            raise

    print('set_col_types end')
    return df
```

In [7]:
```
def load_csv_data_to_db(filename_pattern_and_tablename_dict, data_file_path, co
l_type_dict):
    '''



    '''

    current_pattern = ''
    last_file = ''
    total_files = 0
    loaded_files = []
   # print('outside first for')
    # 1. iterate through list of patterns, to load all file-types into the data
base.
    for pattern in filename_pattern_and_tablename_dict.keys():

        print('complete:', current_pattern, 'total number of files:', total_fil
es)
        print('last_file', last_file)
        current_pattern = pattern
        # 2. Get the list of files from the data-folder:
        data_file_list = get_file_names(data_file_path, pattern)

        table_name = filename_pattern_and_tablename_dict[pattern]

        # 3. load data into a data frame
        file_counter = 1
       # print('in first loop, outside second.. ')
        for data_file in data_file_list:
            last_file = data_file
            if data_file not in loaded_files:
                loaded_files.append(data_file)
#             print('top of second loop.')
            df = load_csv_file_as_df(data_file_path, data_file)

            df = set_col_types(df, col_type_dict)

            # Get the columns data types from the data frame and convert
            # to SQL-Alchemy friend types.

            db_cols = get_db_cols(df)

#             if db_cols == None:

#                 print(data_file, f)
#                 return None

            if file_counter == 1:

                df.to_sql(table_name,
                                  db_engine,
                                  if_exists='replace',
                                schema='public',
                                  index=False,
                                  chunksize=1000,
                                  dtype=db_cols)
                print('if counter = 1', data_file)


            else:
                try:
                    df.to_sql(table_name,
                                      db_engine,
                                      if_exists='append',
                                      schema='public',
                                      index=False,
                                      chunksize=1000,
                                      dtype=db_cols)
                    print('going through the list.. ', data_file)
                except:
```

====

In [8]:
```
db_type = 'postgres'
db_lib = 'psycopg2'
user_id = 'bhima'
password= ''
db_name = 'openfda'

db_URI = build_DB_URI(db_type, db_lib, user_id, password, db_name)
db_engine = create_engine(db_URI, echo=False)
db_engine.connect()
connection= db_engine.connect()
```

In [67]:
```
patient_col_types ={
    'safetyreportid': 'string',
'authoritynumb': 'string',
'companynumb': 'string',
'duplicate': 'string',
'fulfillexpeditecriteria': 'int',
'occurcountry': 'string',
'patient_patientagegroup': 'string',
'patient_patientonsetage': 'float',
'patient_patientonsetageunit': 'float',
'patient_patientsex': 'int',
'patient_patientweight': 'float',
'patient_summary_narrativeincludeclinical': 'string',
'primarysource_literaturereference': 'string',
'primarysource_qualification': 'float',
'primarysource_reportercountry': 'string',
'primarysourcecountry': 'string',
'receiptdate': 'datetime',
'receiptdateformat': 'int',
'receivedate': 'datetime',
'receivedateformat': 'int',
'receiver_receiverorganization': 'string',
'receiver_receivertype': 'float',
'reporttype': 'float',
'safetyreportversion': 'float',
'sender_senderorganization': 'string',
'sender_sendertype': 'float',
'serious': 'int',
'seriousnesscongenitalanomali': 'float',
'seriousnessdeath': 'float',
'seriousnessdisabling': 'float',
'seriousnesshospitalization': 'float',
'seriousnesslifethreatening': 'float',
'seriousnessother': 'float',
'transmissiondate': 'datetime',
'transmissiondateformat': 'int'}

####
reactions_col_types = {'receiptdate': 'datetime',
'safetyreportid': 'string',
'reactionmeddrapt': 'string',
'reactionmeddraversionpt': 'float',
'reactionoutcome': 'float'}

###
drugs_col_types = {'receiptdate': 'datetime',
'safetyreportid': 'string',
'actiondrug': 'float',
'activesubstancename': 'string',
'medicinalproduct': 'string',
'openfda_md5': 'string',
'openfda_brand_name': 'string',
'openfda_generic_name': 'string',
'drugadditional': 'string',
'drugcumulativedosagenumb': 'string',
'drugcumulativedosageunit': 'string',
'drugdosageform': 'string',
'drugintervaldosagedefinition': 'string',
'drugintervaldosageunitnumb': 'string',
'drugrecurreadministration': 'string',
'drugseparatedosagenumb': 'string',
'drugstructuredosagenumb': 'string',
'drugstructuredosageunit': 'string',
'drugadministrationroute': 'string',
'drugauthorizationnumb': 'string',
'drugbatchnumb': 'string',
'drugcharacterization': 'int',
'drugdosagetext': 'string',
'drugenddate': 'datetime',
'drugenddateformat': 'float'
```

In [ ]:

In [ ]:

In [10]:
```
!ls ../Data/csv/2012-2015/2012q1_drug-event-0002-of-0002.json.*
```

```
../Data/csv/2012-2015/2012q1_drug-event-0002-of-0002.json.drug.csv
../Data/csv/2012-2015/2012q1_drug-event-0002-of-0002.json.openfda.csv
../Data/csv/2012-2015/2012q1_drug-event-0002-of-0002.json.patient.csv
../Data/csv/2012-2015/2012q1_drug-event-0002-of-0002.json.reaction.csv
```

In [11]:
```
filename_pattern_and_tablename_dict = {'openfda.csv':'open_fda'}
#{'patient.csv':'patients', 'reaction.csv':'reactions', 'openfda.csv':'open_fda
', 'drug.csv':'drugs'} #{'drug.csv':'drugs',

#location of the data files:
data_file_path = '../Data/csv/2012-2015/'

loaded_files = load_csv_data_to_db(filename_pattern_and_tablename_dict, data_fi
le_path, open_fda_col_types)
```

```
complete:  total number of files: 0
last_file
if counter = 1 2012q1_drug-event-0001-of-0002.json.openfda.csv
going through the list..  2012q1_drug-event-0002-of-0002.json.openfda.csv
going through the list..  2012q2_drug-event-0001-of-0002.json.openfda.csv
going through the list..  2012q2_drug-event-0002-of-0002.json.openfda.csv
going through the list..  2012q3_drug-event-0001-of-0002.json.openfda.csv
going through the list..  2012q3_drug-event-0002-of-0002.json.openfda.csv
going through the list..  2012q4_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2012q4_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2012q4_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2013q1_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2013q1_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2013q1_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2013q2_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2013q2_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2013q2_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2013q3_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2013q3_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2013q3_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2013q4_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2013q4_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2013q4_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2014q1_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2014q1_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2014q1_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2014q2_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2014q2_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2014q2_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2014q3_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2014q3_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2014q3_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2014q4_drug-event-0001-of-0003.json.openfda.csv
going through the list..  2014q4_drug-event-0002-of-0003.json.openfda.csv
going through the list..  2014q4_drug-event-0003-of-0003.json.openfda.csv
going through the list..  2015q1_drug-event-0001-of-0004.json.openfda.csv
going through the list..  2015q1_drug-event-0002-of-0004.json.openfda.csv
going through the list..  2015q1_drug-event-0003-of-0004.json.openfda.csv
going through the list..  2015q1_drug-event-0004-of-0004.json.openfda.csv
going through the list..  2015q2_drug-event-0001-of-0004.json.openfda.csv
going through the list..  2015q2_drug-event-0002-of-0004.json.openfda.csv
going through the list..  2015q2_drug-event-0003-of-0004.json.openfda.csv
going through the list..  2015q2_drug-event-0004-of-0004.json.openfda.csv
going through the list..  2015q3_drug-event-0001-of-0005.json.openfda.csv
going through the list..  2015q3_drug-event-0002-of-0005.json.openfda.csv
going through the list..  2015q3_drug-event-0003-of-0005.json.openfda.csv
going through the list..  2015q3_drug-event-0004-of-0005.json.openfda.csv
going through the list..  2015q3_drug-event-0005-of-0005.json.openfda.csv
going through the list..  all_other_drug-event-0001-of-0001.json.openfda.csv
```

In [17]:
```
filename_pattern_and_tablename_dict = {'patient.csv':'patients'}
#{'patient.csv':'patients', 'reaction.csv':'reactions', 'openfda.csv':'open_fda
', 'drug.csv':'drugs'} #{'drug.csv':'drugs',

#location of the data files:
data_file_path = '../Data/csv/2012-2015/'

loaded_files = load_csv_data_to_db(filename_pattern_and_tablename_dict, data_fi
le_path, patient_col_types)
```

```
complete:  total number of files: 0
last_file
```

```
/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (5,15,20) have mixed types.Specify dtype option on i
mport or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
if counter = 1 2012q1_drug-event-0001-of-0002.json.patient.csv
going through the list..  2012q1_drug-event-0002-of-0002.json.patient.csv
going through the list..  2012q2_drug-event-0001-of-0002.json.patient.csv
going through the list..  2012q2_drug-event-0002-of-0002.json.patient.csv
going through the list..  2012q3_drug-event-0001-of-0002.json.patient.csv
going through the list..  2012q3_drug-event-0002-of-0002.json.patient.csv
```

```
/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (0) have mixed types.Specify dtype option on import
or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
going through the list..  2012q4_drug-event-0001-of-0003.json.patient.csv
going through the list..  2012q4_drug-event-0002-of-0003.json.patient.csv
going through the list..  2012q4_drug-event-0003-of-0003.json.patient.csv
going through the list..  2013q1_drug-event-0001-of-0003.json.patient.csv
going through the list..  2013q1_drug-event-0002-of-0003.json.patient.csv
going through the list..  2013q1_drug-event-0003-of-0003.json.patient.csv
going through the list..  2013q2_drug-event-0001-of-0003.json.patient.csv
going through the list..  2013q2_drug-event-0002-of-0003.json.patient.csv
going through the list..  2013q2_drug-event-0003-of-0003.json.patient.csv
going through the list..  2013q3_drug-event-0001-of-0003.json.patient.csv
going through the list..  2013q3_drug-event-0002-of-0003.json.patient.csv
going through the list..  2013q3_drug-event-0003-of-0003.json.patient.csv
going through the list..  2013q4_drug-event-0001-of-0003.json.patient.csv
going through the list..  2013q4_drug-event-0002-of-0003.json.patient.csv
```

```
/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (11) have mixed types.Specify dtype option on import
or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
going through the list..  2013q4_drug-event-0003-of-0003.json.patient.csv
going through the list..  2014q1_drug-event-0001-of-0003.json.patient.csv
going through the list..  2014q1_drug-event-0002-of-0003.json.patient.csv
```

```
/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (1,11,12) have mixed types.Specify dtype option on i
mport or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
going through the list..  2014q1_drug-event-0003-of-0003.json.patient.csv
going through the list..  2014q2_drug-event-0001-of-0003.json.patient.csv
going through the list..  2014q2_drug-event-0002-of-0003.json.patient.csv
going through the list..  2014q2_drug-event-0003-of-0003.json.patient.csv
going through the list..  2014q3_drug-event-0001-of-0003.json.patient.csv
going through the list..  2014q3_drug-event-0002-of-0003.json.patient.csv
going through the list..  2014q3_drug-event-0003-of-0003.json.patient.csv
going through the list..  2014q4_drug-event-0001-of-0003.json.patient.csv
going through the list..  2014q4_drug-event-0002-of-0003.json.patient.csv
going through the list..  2014q4_drug-event-0003-of-0003.json.patient.csv
going through the list..  2015q1_drug-event-0001-of-0004.json.patient.csv
going through the list..  2015q1_drug-event-0002-of-0004.json.patient.csv
going through the list..  2015q1_drug-event-0003-of-0004.json.patient.csv
going through the list..  2015q1_drug-event-0004-of-0004.json.patient.csv
going through the list..  2015q2_drug-event-0001-of-0004.json.patient.csv
going through the list..  2015q2_drug-event-0002-of-0004.json.patient.csv
going through the list..  2015q2_drug-event-0003-of-0004.json.patient.csv
going through the list..  2015q2_drug-event-0004-of-0004.json.patient.csv
going through the list..  2015q3_drug-event-0001-of-0005.json.patient.csv
going through the list..  2015q3_drug-event-0002-of-0005.json.patient.csv
going through the list..  2015q3_drug-event-0003-of-0005.json.patient.csv
going through the list..  2015q3_drug-event-0004-of-0005.json.patient.csv
going through the list..  2015q3_drug-event-0005-of-0005.json.patient.csv
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/opt/anaconda3/lib/python3.7/site-packages/pandas/core/tools/datetimes.py in _
convert_listlike_datetimes(arg, format, name, tz, unit, errors, infer_datetime
_format, dayfirst, yearfirst, exact)
    431             try:
--> 432                 values, tz = conversion.datetime_to_datetime64(arg)
    433                 return DatetimeIndex._simple_new(values, name=name, tz
=tz)

pandas/_libs/tslibs/conversion.pyx in pandas._libs.tslibs.conversion.datetime_
to_datetime64()

TypeError: Unrecognized value type: <class 'int'>

During handling of the above exception, another exception occurred:

OutOfBoundsDatetime                       Traceback (most recent call last)
<ipython-input-16-804bda9bf992> in set_col_types(df, col_type_dict)
     23             try:
---> 24                 df[col] = pd.to_datetime(df[col], format='%Y%m%d').dt.
strftime("%Y-%m-%d")
     25             except OutOfBoundsDatetime:

/opt/anaconda3/lib/python3.7/site-packages/pandas/core/tools/datetimes.py in t
o_datetime(arg, errors, dayfirst, yearfirst, utc, format, exact, unit, infer_d
atetime_format, origin, cache)
    727         else:
--> 728             values = convert_listlike(arg._values, format)
    729             result = arg._constructor(values, index=arg.index, name=ar
g.name)

/opt/anaconda3/lib/python3.7/site-packages/pandas/core/tools/datetimes.py in _
convert_listlike_datetimes(arg, format, name, tz, unit, errors, infer_datetime
_format, dayfirst, yearfirst, exact)
    434             except (ValueError, TypeError):
--> 435                 raise e
    436

/opt/anaconda3/lib/python3.7/site-packages/pandas/core/tools/datetimes.py in _
convert_listlike_datetimes(arg, format, name, tz, unit, errors, infer_datetime
_format, dayfirst, yearfirst, exact)
    399                     result, timezones = array_strptime(
--> 400                         arg, format, exact=exact, errors=errors
    401                     )

pandas/_libs/tslibs/strptime.pyx in pandas._libs.tslibs.strptime.array_strptim
e()

pandas/_libs/tslibs/strptime.pyx in pandas._libs.tslibs.strptime.array_strptim
e()

pandas/_libs/tslibs/np_datetime.pyx in pandas._libs.tslibs.np_datetime.check_d
ts_bounds()

OutOfBoundsDatetime: Out of bounds nanosecond timestamp: 5200-09-26 00:00:00

During handling of the above exception, another exception occurred:

NameError                                 Traceback (most recent call last)
<ipython-input-17-c6fa7814c731> in <module>
      5 data_file_path = '../Data/csv/2012-2015/'
      6
----> 7 loaded_files = load_csv_data_to_db(filename_pattern_and_tablename_dic
t, data_file_path, patient_col_types)

<ipython-input-7-f33945a68b39> in load_csv_data_to_db(filename_pattern_and_tab
lename_dict, data_file_path, col_type_dict)
     31             df = load_csv_file_as_df(data_file_path, data_file)
     32
```

In [ ]:

In [21]:
```
filename_pattern_and_tablename_dict = {'reaction.csv':'reactions'}
#{'patient.csv':'patients', 'reaction.csv':'reactions', 'openfda.csv':'open_fda
', 'drug.csv':'drugs'} #{'drug.csv':'drugs',

#location of the data files:
data_file_path = '../Data/csv/2012-2015/'

loaded_files = load_csv_data_to_db(filename_pattern_and_tablename_dict, data_fi
le_path, reactions_col_types)
```

```
complete:  total number of files: 0
last_file
if counter = 1 2012q1_drug-event-0001-of-0002.json.reaction.csv
going through the list..  2012q1_drug-event-0002-of-0002.json.reaction.csv
going through the list..  2012q2_drug-event-0001-of-0002.json.reaction.csv
going through the list..  2012q2_drug-event-0002-of-0002.json.reaction.csv
going through the list..  2012q3_drug-event-0001-of-0002.json.reaction.csv
going through the list..  2012q3_drug-event-0002-of-0002.json.reaction.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (1) have mixed types.Specify dtype option on import
or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)

going through the list..  2012q4_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2012q4_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2012q4_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2013q1_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2013q1_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2013q1_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2013q2_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2013q2_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2013q2_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2013q3_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2013q3_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2013q3_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2013q4_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2013q4_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2013q4_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2014q1_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2014q1_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2014q1_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2014q2_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2014q2_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2014q2_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2014q3_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2014q3_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2014q3_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2014q4_drug-event-0001-of-0003.json.reaction.csv
going through the list..  2014q4_drug-event-0002-of-0003.json.reaction.csv
going through the list..  2014q4_drug-event-0003-of-0003.json.reaction.csv
going through the list..  2015q1_drug-event-0001-of-0004.json.reaction.csv
going through the list..  2015q1_drug-event-0002-of-0004.json.reaction.csv
going through the list..  2015q1_drug-event-0003-of-0004.json.reaction.csv
going through the list..  2015q1_drug-event-0004-of-0004.json.reaction.csv
going through the list..  2015q2_drug-event-0001-of-0004.json.reaction.csv
going through the list..  2015q2_drug-event-0002-of-0004.json.reaction.csv
going through the list..  2015q2_drug-event-0003-of-0004.json.reaction.csv
going through the list..  2015q2_drug-event-0004-of-0004.json.reaction.csv
going through the list..  2015q3_drug-event-0001-of-0005.json.reaction.csv
going through the list..  2015q3_drug-event-0002-of-0005.json.reaction.csv
going through the list..  2015q3_drug-event-0003-of-0005.json.reaction.csv
going through the list..  2015q3_drug-event-0004-of-0005.json.reaction.csv
going through the list..  2015q3_drug-event-0005-of-0005.json.reaction.csv
0         20010221
1         20010621
2         20030612
3         20020314
4         20020314
            ...
165564    20030124
165565    20030124
165566    20030124
165567    20030124
165568    20030325
Name: receiptdate, Length: 165569, dtype: int64 raised and exception, at tdate
time..
going through the list..  all_other_drug-event-0001-of-0001.json.reaction.csv
```

In [ ]:

In [68]:
```
filename_pattern_and_tablename_dict = {'drug.csv':'drugs'}
#{'patient.csv':'patients', 'reaction.csv':'reactions', 'openfda.csv':'open_fda
', 'drug.csv':'drugs'} #{'drug.csv':'drugs',

#location of the data files:
data_file_path = '../Data/csv/2012-2015/'

loaded_files = load_csv_data_to_db(filename_pattern_and_tablename_dict, data_fi
le_path, drugs_col_types)
```

```
complete:  total number of files: 0
last_file
set_col_types start
set_col_types end
if counter = 1 2012q1_drug-event-0001-of-0002.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2012q1_drug-event-0002-of-0002.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2012q2_drug-event-0001-of-0002.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2012q2_drug-event-0002-of-0002.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2012q3_drug-event-0001-of-0002.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (16) have mixed types.Specify dtype option on import
or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)

set_col_types start
set_col_types end
going through the list..  2012q3_drug-event-0002-of-0002.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (1,28) have mixed types.Specify dtype option on impo
rt or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
set_col_types start
set_col_types end
going through the list..  2012q4_drug-event-0001-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2012q4_drug-event-0002-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2012q4_drug-event-0003-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q1_drug-event-0001-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q1_drug-event-0002-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q1_drug-event-0003-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q2_drug-event-0001-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q2_drug-event-0002-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q2_drug-event-0003-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q3_drug-event-0001-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q3_drug-event-0002-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q3_drug-event-0003-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q4_drug-event-0001-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2013q4_drug-event-0002-of-0003.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (3,16,28) have mixed types.Specify dtype option on i
mport or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)

set_col_types start
set_col_types end
going through the list..  2013q4_drug-event-0003-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2014q1_drug-event-0001-of-0003.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (16,28) have mixed types.Specify dtype option on imp
ort or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)

set_col_types start
set_col_types end
going through the list..  2014q1_drug-event-0002-of-0003.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (3,28) have mixed types.Specify dtype option on impo
rt or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
set_col_types start
set_col_types end
going through the list..  2014q1_drug-event-0003-of-0003.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (19,28) have mixed types.Specify dtype option on imp
ort or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)

set_col_types start
set_col_types end
going through the list..  2014q2_drug-event-0001-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2014q2_drug-event-0002-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2014q2_drug-event-0003-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2014q3_drug-event-0001-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2014q3_drug-event-0002-of-0003.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (16,19,28) have mixed types.Specify dtype option on
import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)

set_col_types start
set_col_types end
going through the list..  2014q3_drug-event-0003-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2014q4_drug-event-0001-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2014q4_drug-event-0002-of-0003.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2014q4_drug-event-0003-of-0003.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (6,7,16,19,28) have mixed types.Specify dtype option
on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)

set_col_types start
set_col_types end
going through the list..  2015q1_drug-event-0001-of-0004.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q1_drug-event-0002-of-0004.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q1_drug-event-0003-of-0004.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q1_drug-event-0004-of-0004.json.drug.csv

/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:33
31: DtypeWarning: Columns (6,7,28) have mixed types.Specify dtype option on im
port or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
set_col_types start
set_col_types end
going through the list..  2015q2_drug-event-0001-of-0004.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q2_drug-event-0002-of-0004.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q2_drug-event-0003-of-0004.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q2_drug-event-0004-of-0004.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q3_drug-event-0001-of-0005.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q3_drug-event-0002-of-0005.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q3_drug-event-0003-of-0005.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q3_drug-event-0004-of-0005.json.drug.csv
set_col_types start
set_col_types end
going through the list..  2015q3_drug-event-0005-of-0005.json.drug.csv
set_col_types start
set_col_types end
going through the list..  all_other_drug-event-0001-of-0001.json.drug.csv
```

In [ ]:

data_file_path = '../Data/csv/SelectedData/' data_file = '2008q2_drug-event-0001-of-0002.json.openfda.csv' df = load_csv_file_as_df(data_file_path, data_file) df = clean_up_column_values(df)

!ls ../Data/csv/SelectedData/2008q2_drug-event-0001-of-0002.json.openfda.csv

for col in df.columns: print(col)

def clean_up_column_values(df, set_max_val=True, max_unique_col_vals=99): ''' A method that aims to clean up columns in a data frame. When importing data from a CSV sometimes NaN values are put in empty spaces of a column containing strings.

```
Args:
======

df                    --> the data frame which needs values to be cleaned up.

set_max_val           --> a boolean which allows the user to decide if they want to
go
                          through all unique values in a column.

max_unique_col_values --> some columns have many unique values and it would take a
                          long time to check every value, so there is an option.
```

Returns:

df --> the data frame which has been cleaned up.

```
'''

for col in list(df.columns):

    print('before', col, df[col].dtype)
    #if df[col].dtype == 'object':


    vals = pd.unique(df[col])
    print('these are the unique values:')
    print(vals)
    col_dtype = np.nan


    if set_max_val and len(vals) > max_unique_col_vals:
        continue

    for v in vals:
        if col_dtype != np.nan:
            col_dtype = type(v)
        elif col_dytpe != type(v):
            raise TypeError("There are several data-types in this column:", col, va
ls, val_type)

    if col_dtype == str:
        #print('got a string', col)
        df[col].fillna('', inplace=True)
        df[col] = df[col].astype(str)
    if col_dtype == 'int64':
        print('got an int', col)
        df[col].fillna(0, inplace=True)
        df[col] = df[col].astype(int)
    if col_dtype == 'float64':
        print('got a float', col)
        df[col].fillna(0.0, inplace=True)
        df[col] = df[col].astype(float)
return df
```

In [ ]:
```
drug_file = '2012q1_drug-event-0002-of-0002.json.drug.csv'
drugs_col_types = {}
drug_df = load_csv_file_as_df(data_file_path, drug_file)

for col in list(drug_df.columns):

    print(col,' || ', drug_df[col].dtype, ' || ', pd.unique(drug_df[col]))
    drugs_col_types[col] = drug_df[col].dtype

print('==========================')
for key in drugs_col_types:
    print("'"+key+"':", "'"+str(drugs_col_types[key])+"'")

drugs_col_types = {'receiptdate': 'datetime'
'safetyreportid': 'string'
'actiondrug': 'float'
'activesubstancename': 'string'
'medicinalproduct': 'string'
'openfda_md5': 'string'
'openfda_brand_name': 'string'
'openfda_generic_name': 'string'
'drugadditional': 'string'
'drugcumulativedosagenumb': 'string'
'drugcumulativedosageunit': 'string'
'drugdosageform': 'string'
'drugintervaldosagedefinition': 'float'
'drugintervaldosageunitnumb': 'float'
'drugrecurreadministration': 'float'
'drugseparatedosagenumb': 'float'
'drugstructuredosagenumb': 'float'
'drugstructuredosageunit': 'float'
'drugadministrationroute': 'float'
'drugauthorizationnumb': 'float'
'drugbatchnumb': 'string'
'drugcharacterization': 'int'
'drugdosagetext': 'string'
'drugenddate': 'datetime'
'drugenddateformat': 'float'
'drugindication': 'string'
'drugstartdate': 'datetime'
'drugstartdateformat': 'float'
'drugtreatmentduration': 'float'
'drugtreatmentdurationunit': 'float'}
```

In [ ]:
```
reaction_file = '2012q1_drug-event-0002-of-0002.json.reaction.csv'
reactions_col_types = {}
reaction_df = load_csv_file_as_df(data_file_path, reaction_file)

for col in list(reaction_df.columns):

    print(col,' || ', reaction_df[col].dtype, ' || ', pd.unique(reaction_df[co
l]))
    reactions_col_types[col] = reaction_df[col].dtype

print('==========================')
for key in reactions_col_types:
    print("'"+key+"':", "'"+str(reactions_col_types[key])+"'")


reaction_df

reactions_col_types ={'receiptdate': 'datetime'
'safetyreportid': 'string'
'reactionmeddrapt': 'string'
'reactionmeddraversionpt': 'float'
'reactionoutcome': 'float'}
```

In [ ]:
```
data_file_path = '../Data/csv/2012-2015/''

patient_file = '2012q1_drug-event-0002-of-0002.json.patient.csv'
patients_col_types = {}
patient_df = load_csv_file_as_df(data_file_path, patient_file)

for col in list(patient_df.columns):

    print(col,' || ', patient_df[col].dtype, ' || ', pd.unique(patient_df[co
l]))
    patients_col_types[col] = patient_df[col].dtype

print('=========================')
for key in patients_col_types:
    print("'"+key+"':", "'"+str(patients_col_types[key])+"'")

patient_df.transmissiondate.unique()

patient_col_types ={
    'safetyreportid': 'string'
'authoritynumb': 'string'
'companynumb': 'string'
'duplicate': 'string'
'fulfillexpeditecriteria': 'int'
'occurcountry': 'string'
'patient_patientagegroup': 'string'
'patient_patientonsetage': 'float'
'patient_patientonsetageunit': 'float'
'patient_patientsex': 'int'
'patient_patientweight': 'float'
'patient_summary_narrativeincludeclinical': 'string'
'primarysource_literaturereference': 'string'
'primarysource_qualification': 'float'
'primarysource_reportercountry': 'string'
'primarysourcecountry': 'string'
'receiptdate': 'datetime'
'receiptdateformat': 'int'
'receivedate': 'datetime'
'receivedateformat': 'int'
'receiver_receiverorganization': 'string'
'receiver_receivertype': 'float'
'reporttype': 'float'
'safetyreportversion': 'float'
'sender_senderorganization': 'string'
'sender_sendertype': 'float'
'serious': 'int'
'seriousnesscongenitalanomali': 'float'
'seriousnessdeath': 'float'
'seriousnessdisabling': 'float'
'seriousnesshospitalization': 'float'
'seriousnesslifethreatening': 'float'
'seriousnessother': 'float'
'transmissiondate': 'datetime'
'transmissiondateformat': 'int'}
```

In [ ]:
```
patient_df['receiptdate'] = pd.to_datetime(patient_df['receiptdate'], format='%
Y%m%d').dt.strftime("%Y-%m-%d")

patient_df[['receiptdate','receiptdate']]
```

In [70]:
```
!/opt/anaconda3/bin/conda install pandoc
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

In [ ]: