DataBricks Pyspark Pipeline Generator: Load up a CSV file into unity catalog bronze tables make some transformations and then store the results into a silver table.

Prerequisites
Before we begin, make sure you have the following libraries installed in your Python environment:

Step 1: Normal PySpark Imports and import parameters from pipeline
These are always needed at the top of our file

```
import pyspark.sql.functions as F
from pyspark.sql import Window
from delta import DeltaTable
from functools import partials
from datetime import datetime, timezone
import common

params = {"catalog:"", "schema":""}
common.notebook_params(params)
for k,v in params.items()
    globals()[k]=v
print(params)
del params

readstream_config = partial(common.readstream,
    source_path="s3a://josephs_bucket",
    databricks_catalog=catalog, # set using globals()
    databricks_schema=schema,
    product="Patients pipeline",
    path_and_filename={source_path}/{csv_name}
    )
```

Step 2.a: Import the CSV data
According to databricks best practices we should use the databricks autoloader to automatically load up the CSV into a unity catalog table. Fortunately, a brilliant sub-genius has already made this simple for us. Do this for all of the CSV's we want to import before going to the next step.

```
csv_{CSV_FILENAME} = readstream("{CSV_FILENAME}")
```
Replace CSV_FILENAME of the files you want to import

Step 2.b: Wait for all the CSV's to load
The above step with run fully async, we now need to wait until all CSV files are loaded. Do this for all of the CSV files we loaded from the above step.

```
csv_{CSV_FILENAME}.awaitTermination()
bronze_{CSV_FILENAME} = spark.read.table("bronze_{CSV_FILENAME}")
```
Replace CSV_FILENAME of the files you want to import
bronze_{CSV_FILENAME} is a Dataframe with our data

Step 3: Transformations
We want to make all the data on our pipelines to be HIPAA Safe Harbor compliant; de-identify or remove the 18 specific patient identifiers from health data so it's no longer considered Protected Health Information (PHI) using encryption to be complaint with Safe Harbor provision in cybersecurity (HR 7898). The 18 identifiers are: names, geo data, dates, phone, fax, email address, SSN, MRN, license numbers, behicle identifiers, license plates, web urls, ip addresses, fingerprints, voices, photos.

We want to remove the data within PII column specified above. We can do it with the following code.

```
bronze_{CSV_FILENAME} = bronze_{CSV_FILENAME}.withColumn("{COLUMN_NAME}", F.lit(None))
```
Replace COLUMN_NAME with the column you want within the Dataframe.
Replace CSV_FILENAME of the files you want to import

Step 4: Saving the transformed data in a silver table
We will keep it simple in saving data. Just save it and overwrite old data.

```
bronze_{CSV_FILENAME}.write.option("clusterBy.auto", "true").saveAsTable("silver_{CSV_FILENAME}}")
```
Replace CSV_FILENAME of the files you want to import