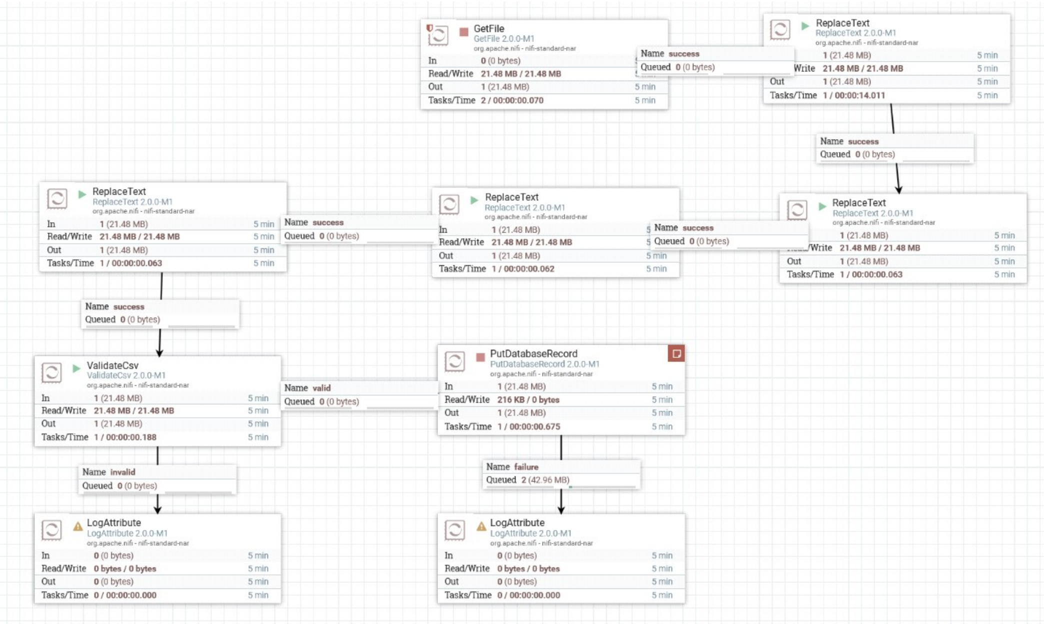
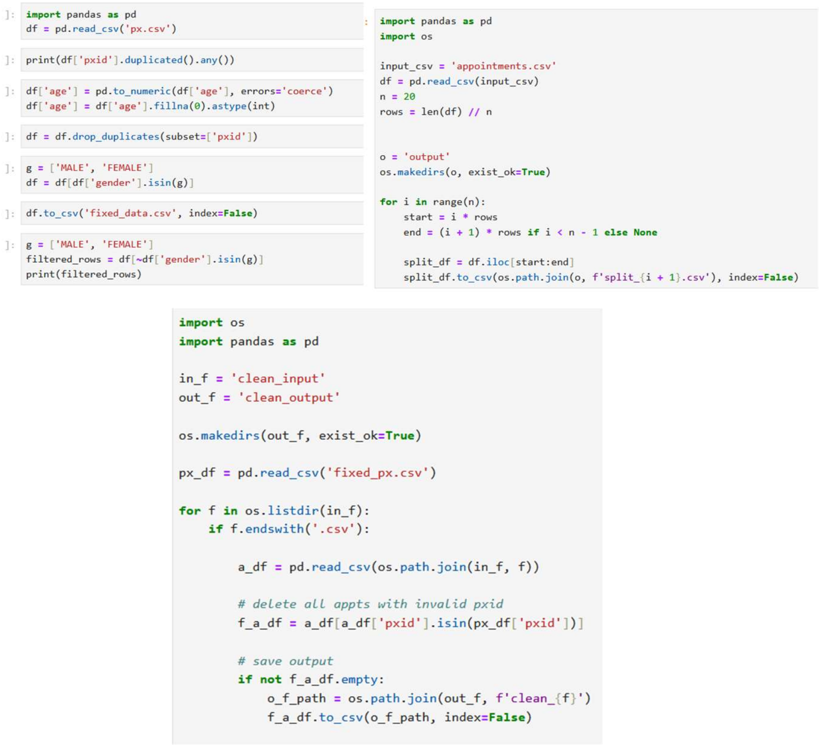


Figure 1. Apache Nifi Pipeline

This pipeline was the primary pipeline used to extract, clean, validate, and then load all 4 CSV files. It is only possible to have a pipeline as simple as this one due to the CSV structure matching the structure of our database warehouse. We also chose to not use an intermediary database to load from for this very reason – the intermediary database would have had the exact same structure as our DBW. As such, we chose not to.



```
import pandas as pd
df = pd.read_csv('px.csv')

print(df['pxid'].duplicated().any())

df['age'] = pd.to_numeric(df['age'], errors='coerce')
df['age'] = df['age'].fillna(0).astype(int)

df = df.drop_duplicates(subset=['pxid'])

g = ['MALE', 'FEMALE']
df = df[df['gender'].isin(g)]

df.to_csv('fixed_data.csv', index=False)

g = ['MALE', 'FEMALE']
filtered_rows = df[~df['gender'].isin(g)]
print(filtered_rows)
```

```
import pandas as pd
import os

input_csv = 'appointments.csv'
df = pd.read_csv(input_csv)
n = 20
rows = len(df) // n


o = 'output'
os.makedirs(o, exist_ok=True)

for i in range(n):
    start = i * rows
    end = (i + 1) * rows if i < n - 1 else None

    split_df = df.iloc[start:end]
    split_df.to_csv(os.path.join(o, f'split_{i + 1}.csv'), index=False)
```

```
import os
import pandas as pd

in_f = 'clean_input'
out_f = 'clean_output'

os.makedirs(out_f, exist_ok=True)

px_df = pd.read_csv('fixed_px.csv')

for f in os.listdir(in_f):
    if f.endswith('.csv'):

        a_df = pd.read_csv(os.path.join(in_f, f))

        # delete all appts with invalid pxid
        f_a_df = a_df[a_df['pxid'].isin(px_df['pxid'])]

        # save output
        if not f_a_df.empty:
            o_f_path = os.path.join(out_f, f'clean_{f}')
            f_a_df.to_csv(o_f_path, index=False)
```

Figure 2. Jupyter Notebook Pandas Cleaning Pipeline

As mentioned in our report, our machines were unable to handle the size of appointments, as we were unable to properly extract it in Nifi without crashing. As such, we were forced to use alternative means to clean the files with unusable data and split them into more easily processable files. Namely, both px and appointments were filled with issues, such as duplicate IDs and appointments containing vast amounts of entries that did not contain a corresponding px_id. This would cause a foreign key exception error in Nifi, as such, we simply deleted all entries that did not have a corresponding px_id. After all cleaning, data was extracted, cleaned, and loaded in the Nifi pipeline pictured in Figure 1.