EJERCICIOS DE ETL CON PYTHON

Indicaciones

- Modo de entrega: Individual, en aula de MIND HUB, Entrega clase 11.1
- Plazo: Hasta las 20:00 del 18 de abril
- Adjuntos: Este documento, más archivos PY/IPYNB y SQL que correspondan

Grupo N°:

Integrantes:

- Felipe Bravo Espinosa.
- Carlos Schoenfeldt.
- Camilo Valenzuela.
- Angel Paillalef.

Código desarrollo

EXTRACTION 1. Importar las dependencias

```
import pandas as pd
import datetime
import sqlite3
from IPython.display import display
```

EXTRACTION 2. Obtener y mostrar tablas de ambos archivos con formato csv "all_years_03.csv" y "all_years_pm25.csv", renombrando cada dataframe df1 y df2.

```
#Cargar dataframes
df1 = pd.read_csv("all_years_o3.csv", sep=";")
df2 = pd.read_csv("all_years_pm25.csv", sep=";")

#mostrar dataframes
display(df1.head())
display(df2.head())
```

TRANSFORM 1. Eliminar las últimas 3 columnas, cuyo campo son: "min…", "max…", "median…" del archivo "all_years_03.csv" o df1

```
drop_df_1 = df_1.drop(['min (ppb)', 'max (ppb)', 'median (ppb)'], axis= 1)
```

TRANSFORM 2. Renombrar la columna "count" de la tabla "all_years_03.csv" o df1

```
#Renombrar la columna "count" de la tabla ''all_years_03.csv'' o df1
rename_df_1 = drop_df_1.rename(columns = {'count': 'Count_03'})
```

TRANSFORM 3. Eliminar la columna " specie" de la tabla "all_years_03.csv" o df1

```
#Eliminar la columna '' specie'' de la tabla ''all_years_03.csv'' o df1
new_df_1 = rename_df_1.drop(columns = ['Specie'])
```

TRANSFORM 4. Eliminar las últimas 3 columnas, cuyo campo son: "min…", "max…", "median…" de la tabla "all_years_pm25.csv" o df2.

```
#Eliminar las últimas 3 columnas, cuyo campo son: 'min…', 'max…', 'median…' de la tabla
'all_years_pm25.csv' o df2.

drop_df_2 = df_2.drop(['min (ug/m3)', 'max (ug/m3)', 'median (ug/m3)'], axis= 1)
```

TRANSFORM 5. Renombrar la columna "count" de la tabla "all_years_pm25.csv" o df2.

```
rename_df_2 = drop_df_2.rename(columns = {'count': 'Count_pm25'})
```

TRANSFORM 6. Eliminar la columna " specie" de la tabla"all_years_pm25.csv" o df2.

```
#Eliminar la columna '' specie'' de la tabla''all_years_pm25.csv'' o df2.
new_df_2 = rename_df_2.drop(columns = ['Specie'])
```

TRANSFORM 7. Unir (merge) el df1 y df2 por medio de los campos "date", "country" y "city", renombrar por new_df y mostrar en consola

```
new_df = new_df_1.merge(new_df_2, how = 'right')
```

LOAD 1. Crear tabla en etl_projct2

```
sqlite3.connect('database.db')
```

LOAD 2. Cargar bancos de datos a sqliteonline.com

```
conn = sqlite3.connect('database.db')
c = conn.cursor()
```

LOAD 3. Confirmar tablas

```
c.execute('''
          CREATE TABLE IF NOT EXISTS merge_counts
          (
              [id_mc] INTEGER PRIMARY KEY,
              [Date] TEXT,
              [Country] VARCHAR,
              [City] VARCHAR,
              [Count_03] INTEGER,
              [Count_pm25] INTEGER)''')

conn.commit()
```

LOAD 4. Cargar dataframe

```
new_df.to_sql(name='merge_counts', con=conn, if_exists='append', index=False)
conn.close()
```

LOAD 5. Mostrar los datos añadidos a la tabla

```
con = sqlite3.connect('database.db')
cur = con.cursor()
cur.execute('''SELECT * FROM merge_counts''')

rows = cur.fetchall()
for row in rows:
    print(row)
```