Informe sobre el Script "Reportes Vacios" con SQLAlchemy y PostgreSQL

**Resumen**

El script "Reportes Vacios" se ha desarrollado en Python y utiliza SQLAlchemy para conectarse a una base de datos PostgreSQL. Su función principal es identificar y generar informes sobre fechas faltantes en un conjunto de datos específico. A continuación, se proporciona una descripción detallada del script y cómo funciona, adicional se realizó un script para poder eliminar archivos .csv si se lo requiere.

**Objetivo**

El objetivo principal del script es encontrar fechas faltantes en un conjunto de datos de estaciones meteorológicas y crear informes CSV que muestren estas fechas faltantes para cada estación y combinación de parámetros.

**Estructura del Script**

El script se divide en las siguientes secciones:

1. \*\*Configuración Inicial\*\*: En esta sección, se importan las bibliotecas necesarias, se establece una conexión a la base de datos PostgreSQL y se definen algunas variables iniciales.

2. \*\*Consulta SQL\*\*: Se define una consulta SQL que recupera datos de la base de datos PostgreSQL. La consulta se une a tres tablas diferentes ("data\_m01", "copa" y "vta\_estaciones\_todos") para obtener información relevante.

3. \*\*Carga de Datos y Preprocesamiento\*\*: Utilizando Pandas y SQLAlchemy, se ejecuta la consulta SQL y se carga el resultado en un DataFrame. Además, se convierte la columna de fecha ("fecha\_faltante") en un formato datetime.

4. \*\*Iteración a través de Estaciones y Combinaciones\*\*: El script itera a través de las estaciones y combinaciones de parámetros únicos encontrados en los datos.

5. \*\*Detección de Fechas Faltantes\*\*: Para cada estación y combinación de parámetros, el script compara las fechas en los datos con un rango de fechas completo. Las fechas que faltan se registran en un nuevo DataFrame llamado "missing\_dates\_df".

6. \*\*Generación de Informes\*\*: Si se encuentran fechas faltantes para una estación y combinación de parámetros, se genera un informe en formato CSV. Los informes se almacenan en un directorio separado para cada estación.

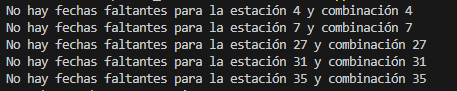


Figura 1: se ejecuta el script si no hay fechas faltantes en el terminal imprime no hay fechas faltantes.

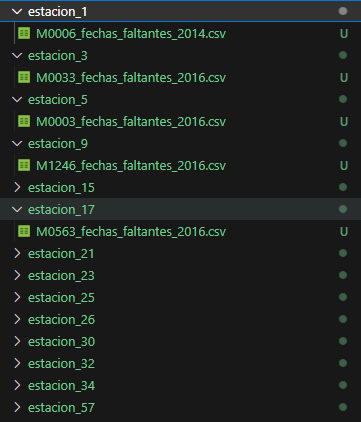


Figura 2: vista de la estructura, como se crean las carpetas por estación y dentro de ellas las estaciones con su respectivo formato.

7. \*\*Optimizaciones Futuras\*\*: Se menciona que el script se puede mejorar para reducir el tiempo de ejecución, especialmente cuando se trabaja con grandes conjuntos de datos.

**Bibliotecas utilizadas**

SQLAlchemy: SQLAlchemy es una biblioteca de Python que proporciona una API de alto nivel para interactuar con bases de datos relacionales. Se utiliza para crear y administrar conexiones a bases de datos, generar consultas SQL y cargar datos en un DataFrame de Pandas.

pandas: pandas es una biblioteca de análisis de datos en Python que proporciona estructuras de datos flexibles para trabajar con datos tabulares, como DataFrames. Se utiliza para cargar datos desde una base de datos, realizar operaciones de manipulación y análisis de datos, y generar informes en formato CSV.

os: El módulo os es parte de la biblioteca estándar de Python y se utiliza para interactuar con el sistema operativo. En este script, se utiliza para crear directorios si no existen y administrar rutas de archivo.

coneccion\_postgres: Aunque no se muestra en el script proporcionado, es una importación personalizada que se utiliza para obtener la cadena de conexión a la base de datos PostgreSQL. Esta importación es específica de tu implementación y se utiliza para mantener la información de conexión separada del código principal.

**Descripción del script eliminarcsv:**

Este script se encarga de limpiar un directorio que contiene carpetas de estaciones, donde cada carpeta contiene archivos CSV. El objetivo principal es eliminar todos los archivos CSV y, después de eliminar los archivos, intentar eliminar las carpetas de estación vacías.

**Funcionamiento:**

Define el directorio base donde se encuentran las carpetas de estaciones.

Obtiene una lista de todas las carpetas de estaciones en el directorio base.

Itera a través de cada carpeta de estación:

a. Obtiene una lista de archivos CSV dentro de la carpeta de estación.

b. Elimina cada archivo CSV encontrado.

c. Intenta eliminar la carpeta de estación después de eliminar los archivos CSV.

d. Si no puede eliminar la carpeta de estación (porque aún contiene archivos), muestra un mensaje de error.

**Uso Principal:**

El script se ejecuta como función principal cuando se llama desde otro script o desde la línea de comandos. Al ejecutarse, realiza la limpieza de archivos y carpetas según la lógica descrita anteriormente.

**Consideraciones de Uso:**

Este script es útil cuando se necesita eliminar archivos CSV antiguos o no deseados en un directorio de estaciones, como parte de la administración de datos.

**Mejoras Potenciales del script Reportes vacíos**

Para reducir el tiempo de ejecución, se pueden considerar las siguientes mejoras:

1. \*\*Optimización de Consultas SQL\*\*: Evaluar si las consultas SQL se pueden optimizar para reducir el tiempo de respuesta de la base de datos.

2. \*\*Caché de Datos\*\*: Almacenar en caché los datos consultados previamente para evitar consultas repetitivas si los datos son estáticos.

3. \*\*Uso de Bibliotecas Específicas\*\*: Explorar bibliotecas y herramientas específicas para el análisis de series de tiempo, como Pandas, que pueden ofrecer mejoras de rendimiento.

Estas mejoras potenciales pueden ayudar a optimizar el script y hacer que sea más eficiente al trabajar con grandes cantidades de datos.

**Resultados dentro de cada archivo csv.**

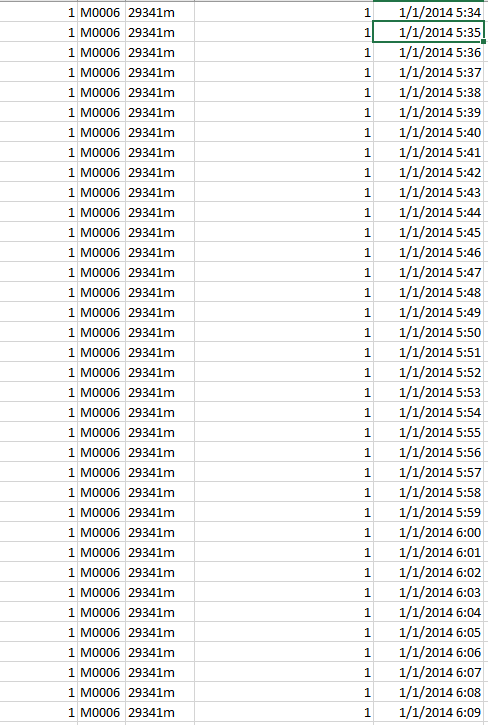


Figura 3: visualización de los datos obtenidos de la estación 1

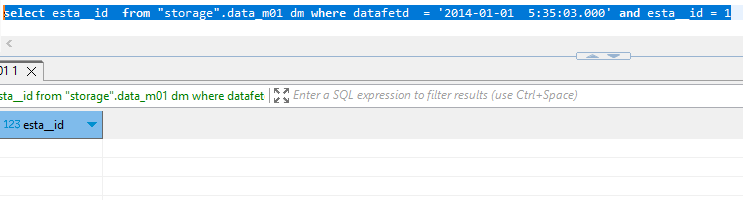
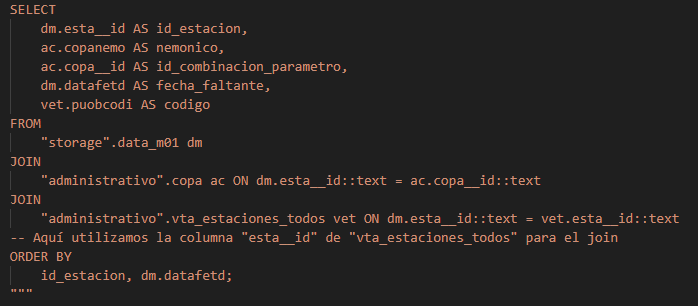


Figura 4: comprobación al ejecutar la consulta para ver si la fecha existe.

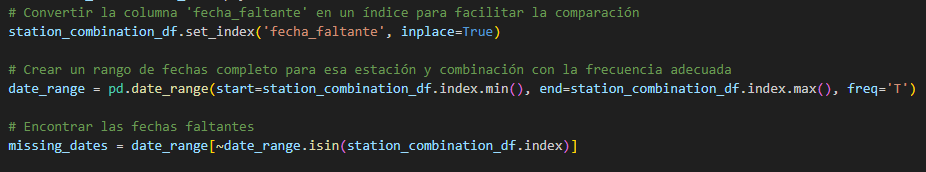
**Conclusiones:**

* La adaptación del script "Reportes Vacíos" para trabajar con una conexión a MongoDB requerirá cambios significativos en la configuración de la conexión y en la forma en que se realizan las consultas de datos. Sin embargo, la lógica principal para detectar fechas faltantes y generar informes se mantendrá relativamente similar.
* El script "Reportes Vacíos" es útil para identificar y generar informes sobre fechas faltantes en datos de estaciones meteorológicas. Facilita la detección de problemas en la recopilación de datos y permite tomar medidas para llenar los vacíos de información. Sin embargo, se sugiere explorar optimizaciones adicionales para manejar conjuntos de datos más grandes de manera eficiente.

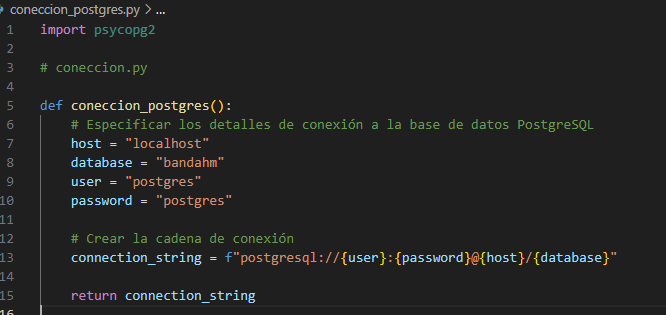
**Anexos:**



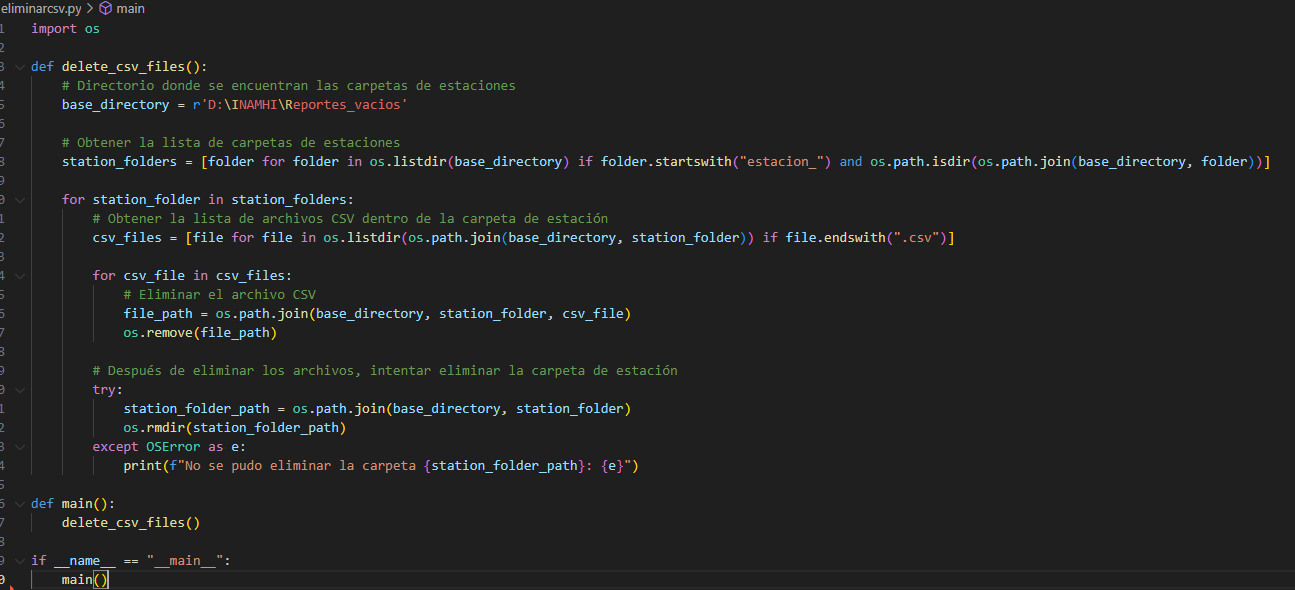
Anexo 1: SQL de la consulta a la base de datos Postgres



Anexo 2: Cálculo para encontrar las fechas faltantes



Anexo 3: conexión a Postgres.



Anexo 4: script eliminar csv.