

Reporte de Desarrollo

Introducción

Este reporte describe el proceso de desarrollo de un script en Python que lee, procesa, y almacena los datos desde un archivo JSON que contiene los tweets. El objetivo del script es verificar que el contenido del archivo siga una estructura adecuada y también leer los tweets almacenados en el archivo

Estructura del script

Conexion a la base de datos

```
connection = mysql.connector.connect(  
    host='localhost',  
    port='33060',  
    database='nao',  
    user='root',  
    password='secret',  
    charset='utf8mb4'  
)  
cursor = connection.cursor()
```

creacion de tablas desde un script

```
with open('crearTabla.sql', 'r') as sql_file:  
    sql_script = sql_file.read()  
sql_statements = sql_script.split(';')  
# Ejecutar cada sentencia SQL  
for statement in sql_statements:  
    # Ignorar líneas vacías  
    if statement.strip():  
        cursor.execute(statement)  
        print(f"Ejecutando: {statement.strip()}")  
  
# Confirmar los cambios  
connection.commit()
```

```
CREATE TABLE nao.tweet (  
    `id` int(11) NOT NULL AUTO_INCREMENT,
```

```

        id_tweet varchar(30) NOT NULL,
        texto varchar(1000) NULL,
        usuario varchar(100) NULL,
        fecha DATETIME NULL,
        retweets NUMERIC NULL,
        favoritos NUMERIC NULL,
        CONSTRAINT tweet_pk PRIMARY KEY (id)
    ) charset utf8mb4;

CREATE TABLE `hashtag` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `hashtags` varchar(500) DEFAULT NULL,
    `id_tweets` int(11) DEFAULT NULL,
    PRIMARY KEY (`id`),
    KEY `hashtag_tweet_FK` (`id_tweets`),
    CONSTRAINT `hashtag_tweet_FK` FOREIGN KEY (`id_tweets`) REFERENCES
    `tweet` (`id`)
) charset utf8mb4;

```

se realiza la insercion en las tablas

```

insert_query_tweet = '''
    INSERT INTO nao.tweet
    (id_tweet, texto, usuario, fecha, retweets, favoritos)
    VALUES (%s, %s, %s, %s, %s, %s);
'''

insert_query_hash = '''
    INSERT INTO nao.hashtag
    ( hashtags, id_tweets)
    VALUES ( %s, %s);
'''

with open('../tweets_extraction.json', 'r') as archivo:
    datos = json.load(archivo)

for record in datos:
    values = (
        record['id'],

```

```
        record['texto'],
        record['usuario'],
        convert_iso_to_mysql_format(record['fecha']),
        record['retweets'],
        record['favoritos'],
    )

    cursor.execute(insert_query_tweet, values)
    id_tweet = cursor.lastrowid
    if len(record['hashtags']):
        for hash in record['hashtags']:
            val = (
                hash,
                id_tweet,
            )
            cursor.execute(insert_query_hash, val)

connection.commit()
```

Conclusiones

El script desarrollado permite leer y almacenar un archivo JSON de manera eficiente utilizando la librería jsonschema y mysqlconector, creando 2 tablas para la inserción de la información