ENTREGABLE 3- EXTRAER DATOS DE API DE YOUTUBE E INSERTAR LA DATA EN UNA TABLA DE AWS REDSHIFT.

Explicación del Proyecto:

El script siguiente obtiene los 10 videos más populares (con más vistas) de YouTube en el momento de la ejecución del script. La API realiza una búsqueda de videos, ordenándolos por la cantidad de vistas (ordenados en orden descendente). Luego, obtiene los detalles de esos videos y crea un DataFrame con la información relevante, incluida la cantidad de vistas, y lo imprime.

Además, en la tabla resultante de la API le incluyo la columna "Insert_Date", que contiene el día de ejecución del script, para identificar en qué día se obtuvieron los 10 registros con los 10 videos más vistos.

Por último, esos 10 registros son insertados en la tabla "videos" dentro de mi Base de Datos de Redshift, haciendo previamente la conexión a dicha Base de Datos (y creando la tabla en primera instancia).

In [20]:
```
pip install oauth2client
```

```
Requirement already satisfied: oauth2client in c:\users\cnieto1\anaconda3\envs\dhdsbl
end2021\lib\site-packages (4.1.3)
Requirement already satisfied: httplib2>=0.9.1 in c:\users\cnieto1\anaconda3\envs\dhd
sblend2021\lib\site-packages (from oauth2client) (0.22.0)
Requirement already satisfied: pyasn1>=0.1.7 in c:\users\cnieto1\anaconda3\envs\dhdsb
lend2021\lib\site-packages (from oauth2client) (0.4.8)
Requirement already satisfied: pyasn1-modules>=0.0.5 in c:\users\cnieto1\anaconda3\en
vs\dhdsblend2021\lib\site-packages (from oauth2client) (0.2.7)
Requirement already satisfied: rsa>=3.1.4 in c:\users\cnieto1\anaconda3\envs\dhdsblen
d2021\lib\site-packages (from oauth2client) (4.8)
Requirement already satisfied: six>=1.6.1 in c:\users\cnieto1\anaconda3\envs\dhdsblen
d2021\lib\site-packages (from oauth2client) (1.16.0)
Requirement already satisfied: pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in
c:\users\cnieto1\anaconda3\envs\dhdsblend2021\lib\site-packages (from httplib2>=0.9.1
->oauth2client) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```

In [21]:
```
pip install google-api-python-client
```

```
Requirement already satisfied: google-api-python-client in c:\users\cnieto1\anaconda3
\envs\dhdsblend2021\lib\site-packages (2.101.0)Note: you may need to restart the kern
el to use updated packages.
Requirement already satisfied: httplib2<1.dev0,>=0.15.0 in c:\users\cnieto1\anaconda3
\envs\dhdsblend2021\lib\site-packages (from google-api-python-client) (0.22.0)

Requirement already satisfied: google-auth<3.0.0.dev0,>=1.19.0 in c:\users\cnieto1\an
aconda3\envs\dhdsblend2021\lib\site-packages (from google-api-python-client) (1.35.0)
Requirement already satisfied: google-auth-httplib2>=0.1.0 in c:\users\cnieto1\anacon
da3\envs\dhdsblend2021\lib\site-packages (from google-api-python-client) (0.1.1)
Requirement already satisfied: google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0.
dev0,>=1.31.5 in c:\users\cnieto1\anaconda3\envs\dhdsblend2021\lib\site-packages (fro
m google-api-python-client) (1.31.5)
Requirement already satisfied: uritemplate<5,>=3.0.1 in c:\users\cnieto1\anaconda3\en
vs\dhdsblend2021\lib\site-packages (from google-api-python-client) (4.1.1)
Requirement already satisfied: googleapis-common-protos<2.0dev,>=1.6.0 in c:\users\cn
ieto1\anaconda3\envs\dhdsblend2021\lib\site-packages (from google-api-core!=2.0.*,!=
2.1.*,!=2.2.*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (1.56.2)
Requirement already satisfied: requests<3.0.0dev,>=2.18.0 in c:\users\cnieto1\anacond
a3\envs\dhdsblend2021\lib\site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.
*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (2.27.1)
Requirement already satisfied: setuptools>=40.3.0 in c:\users\cnieto1\anaconda3\envs
\dhdsblend2021\lib\site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.
0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (68.2.2)
Requirement already satisfied: packaging>=14.3 in c:\users\cnieto1\anaconda3\envs\dhd
sblend2021\lib\site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.
0.0.dev0,>=1.31.5->google-api-python-client) (21.3)
Requirement already satisfied: six>=1.13.0 in c:\users\cnieto1\anaconda3\envs\dhdsble
nd2021\lib\site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0.
dev0,>=1.31.5->google-api-python-client) (1.16.0)
Requirement already satisfied: pytz in c:\users\cnieto1\anaconda3\envs\dhdsblend2021
\lib\site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0.dev0,>
=1.31.5->google-api-python-client) (2021.3)
Requirement already satisfied: protobuf>=3.12.0 in c:\users\cnieto1\anaconda3\envs\dh
dsblend2021\lib\site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<
3.0.0.dev0,>=1.31.5->google-api-python-client) (3.20.3)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\cnieto1\anaconda3\e
nvs\dhdsblend2021\lib\site-packages (from google-auth<3.0.0.dev0,>=1.19.0->google-api
-python-client) (4.2.4)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\cnieto1\anaconda3\en
vs\dhdsblend2021\lib\site-packages (from google-auth<3.0.0.dev0,>=1.19.0->google-api-
python-client) (0.2.7)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\cnieto1\anaconda3\envs\dhdsb
lend2021\lib\site-packages (from google-auth<3.0.0.dev0,>=1.19.0->google-api-python-c
lient) (4.8)
Requirement already satisfied: pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in
c:\users\cnieto1\anaconda3\envs\dhdsblend2021\lib\site-packages (from httplib2<1.dev
0,>=0.15.0->google-api-python-client) (3.0.4)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\cnieto1\anaconda3\env
s\dhdsblend2021\lib\site-packages (from pyasn1-modules>=0.2.1->google-auth<3.0.0.dev
0,>=1.19.0->google-api-python-client) (0.4.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\cnieto1\anaconda3\en
vs\dhdsblend2021\lib\site-packages (from requests<3.0.0dev,>=2.18.0->google-api-core!
=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (1.26.
9)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\cnieto1\anaconda3\envs
\dhdsblend2021\lib\site-packages (from requests<3.0.0dev,>=2.18.0->google-api-core!=
2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (2022.
6.15)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\cnieto1\anaconda
```

```
3\envs\dhdsblend2021\lib\site-packages (from requests<3.0.0dev,>=2.18.0->google-api-c
ore!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client)
(2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\cnieto1\anaconda3\envs\dhdsbl
end2021\lib\site-packages (from requests<3.0.0dev,>=2.18.0->google-api-core!=2.0.*,!=
2.1.*,!=2.2.*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (3.3)
```

In [22]:
```python
#import pandas as pd
import psycopg2
from psycopg2.extras import execute_values
from airflow.models import DAG, Variable
import datetime

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import smtplib
```

In [23]:
```python
from datetime import datetime  # Importa datetime antes de su uso
import pandas as pd
from googleapiclient.discovery import build
import json
```

In [24]:
```python
# Definición de Variables:

# Variables de Conexión a Redshift:
url="data-engineer-cluster.cyhh5bfevlmn.us-east-1.redshift.amazonaws.com"
data_base="data-engineer-database"
#user=Variable.get("user_redshift")                                    #esta variable fue
#pwd= Variable.get("secret_pass_redshift")                             #esta variable fue
user='christian_r_coderhouse'
pwd='3b4LjN1alG'

# Variable de Conexión a la API de Youtube:
#client_API_KEY = Variable.get("client_API_KEY")                       #esta variable fue
client_API_KEY= 'AIzaSyBXPyx2L67WhXATIaaR8yl3FJZLsXvpDIE'
```

In [25]:
```python
# Task 1:
def get_top_videos():

    # Definir tu clave de API de YouTube
    API_KEY = client_API_KEY

    # Crear una instancia del servicio de la API de YouTube
    youtube = build('youtube', 'v3', developerKey=API_KEY)

    # Función para obtener el nombre de la categoría a partir del ID
    def get_category_name(youtube, category_id):
        categories_response = youtube.videoCategories().list(
            part='snippet',
            id=category_id
        ).execute()
        if 'items' in categories_response:
            return categories_response['items'][0]['snippet']['title']
        else:
            return 'Desconocida'

    # Función para convertir la duración en formato "PT11M13S" a segundos
    def convert_duration_to_seconds(duration):
        parts = duration[2:].split('T')[-1].split('H')
```

```
        hours = int(parts[0]) if len(parts) > 1 else 0
        minutes_parts = parts[-1].split('M')
        minutes = int(minutes_parts[0]) if len(minutes_parts) > 1 else 0
        seconds_parts = minutes_parts[-1].split('S')
        seconds = int(seconds_parts[0]) if len(seconds_parts) > 1 else 0
        total_seconds = hours * 3600 + minutes * 60 + seconds
        return total_seconds

    # Obtener la fecha actual en el formato requerido por la API de YouTube
    #current_date = datetime.datetime.now().strftime('%Y-%m-%dT00:00:00Z')

    # Realizar la búsqueda de videos ordenados por vistas y limitar a 10 resultados
    search_response = youtube.search().list(
        part='id',
        maxResults=10,
        order='viewCount',
        type='video'
    ).execute()

    # Extraer los IDs de los videos obtenidos en la búsqueda
    video_ids = [item['id']['videoId'] for item in search_response['items']]

    # Obtener detalles de los videos
    videos_response = youtube.videos().list(
        part='snippet,statistics,contentDetails',
        id=','.join(video_ids)
    ).execute()

    # Crear una lista de diccionarios con la información de los videos
    video_data = []
    for video in videos_response['items']:
        video_id = video['id']
        video_info = {
            "ID_del_Video": video_id,
            "Título": video['snippet']['title'],
            "Descripción": video['snippet']['description'],
            "Canal_Propietario": video['snippet']['channelTitle'],
            "Fecha_de_Publicación": video['snippet']['publishedAt'],
            "Categoría_ID": video['snippet']['categoryId'],
            "Categoría": get_category_name(youtube, video['snippet']['categoryId']),
            "Duración_segundos": convert_duration_to_seconds(video['contentDetails']['
            "URL_del_Video": f"https://www.youtube.com/watch?v={video_id}",
            "Vistas": video['statistics']['viewCount'],
            "Likes": video['statistics'].get('likeCount', 0),
            "Dislikes": video['statistics'].get('dislikeCount', 0),
            "Favorite_Count": video['statistics'].get('favoriteCount', 0),
            "Comment_Count": video['statistics'].get('commentCount', 0),
            #"Insert_Date": current_date
        }
        video_data.append(video_info)

    # Crear un DataFrame a partir de la lista de diccionarios
    df = pd.DataFrame(video_data)

    #Hago las siguientes transformaciones a las columnas del Dataframe df:

    # Recortar la columna "Descripción" y "Título" a 301 caracteres:
    df['Descripción'] = df['Descripción'].str[:301]
    df['Título'] = df['Título'].str[:301]
```

```
    #from datetime import datetime
    #import pandas as pd

    # Convertir la columna "Fecha de Publicación" en objeto datetime y creo la columna
    df['Fecha_de_Publicación'] = pd.to_datetime(df['Fecha_de_Publicación'])
    #df['Insert_Date'] = pd.to_datetime(df['Insert_Date'])
    df['Insert_Date'] = pd.to_datetime(datetime.now().strftime('%Y-%m-%dT00:00:00Z'))

    # Formatear la columna "Fecha de Publicación" y "Insert Date" en el formato deseado
    df['Fecha_de_Publicación'] = df['Fecha_de_Publicación'].dt.strftime('%Y-%m-%d')
    df['Insert_Date'] = df['Insert_Date'].dt.strftime('%Y-%m-%d')
    df=df.to_dict()
    return(df)
```

In [26]:
```
get_top_videos()
```

```
[2023-10-07T22:29:27.103-0300] {__init__.py:49} INFO - file_cache is only supported w
ith oauth2client<4.0.0
```

Out[26]:
{'ID_del_Video': {0: '0aZ7lPQ5EXs',
  1: 'eNLjdPI9zdE',
  2: 'ebVVuJN1WFM',
  3: 'bX3S-_jUauc',
  4: 'wPNQw8naE2Q',
  5: 'aSjflT_J0Xo',
  6: '4nKcnfw9ggc',
  7: 'NPpELzyP4rw',
  8: 'XQaKFU3Fh_M',
  9: 'KATq-Ws3xtM'},
 'Título': {0: 'El Gallo y la Pata - Canciones de la Granja de Zenón 2',
  1: 'La Vaca Lola - Canciones de La Granja de Zenón 2',
  2: 'Bartolito - La Granja de Zenón 3',
  3: 'Paulo Londra ft Lenny Tavarez - Nena Maldicion (Official Video)',
  4: 'El Pollito Pío 3D - Canciones de la Granja de Zenón 2',
  5: 'Paulo Londra - Adan y Eva (Official Video)',
  6: 'Percherón - La Granja de Zenón 3',
  7: 'Paulo Londra - Tal Vez (Official Video)',
  8: 'La Gallina Turuleca - Canciones de la Granja de Zenón 1',
  9: 'Patitos Cua Cua Cua - Canciones y clásicos infantiles'},
 'Descripción': {0: '🎁 En estas Navidades, encuentra los productos de La Granja de Zenón en Amazon Store 🎁 \nPeluches ▶️ https://rebrand.ly/AmazonMXPeluchesMusicalesLGDZ\nSábanas y Mantas ▶️ https://rebrand.ly/AmazonMXJuegoDeCamaLDGZ\n\nLetra \n\nUn gallo se enamoró perdidamente\nde una pata que nadaba en la laguna\ny ese gallo ',
  1: '🎁 En estas Navidades, encuentra los productos de La Granja de Zenón en Amazon Store 🎁 \nPeluches ▶️ https://rebrand.ly/AmazonMXPeluchesMusicalesLGDZ\nSábanas y Mantas ▶️ https://rebrand.ly/AmazonMXJuegoDeCamaLDGZ\n\nLetra:\n\nLa Vaca Lola, la Vaca Lola\ntiene cabeza y tiene cola.\nLa Vaca Lola, la Vaca Lola\n',
  2: '🎁 En estas Navidades, encuentra los productos de La Granja de Zenón en Amazon Store 🎁 \nPeluches ▶️ https://rebrand.ly/AmazonMXPeluchesMusicalesLGDZ\nSábanas y Mantas ▶️ https://rebrand.ly/AmazonMXJuegoDeCamaLDGZ\n\nAmigos de España ya pueden encontrar la colección de Libros de La Granja de Zenón, dispon',
  3: 'Paulo Londra "Nena Maldicion"\n\n(Apple Music) ▶https://apple.co/3pA0oWg\n(Spotify) ▶ https://open.spotify.com/album/1pxbRsIbgUi8eA5lUzFlyi?si=yGTh7wO1RCG1U9NhWGyv8g\n\nVisit Paulo Londra online:\nhttps://www.instagram.com/Paulolondra\n\nVisit Lenny Tavarez online:\nhttps://www.instagram.com/lennytavareztm/\n\n',
  4: 'Encuentra las Marionetas de la Granja de Zenón AQUÍ: https://rebrand.ly/MarionetasLGZ y los Maxi Huevos Sorpresa de sus personajes favoritos AQUÍ: https://rebrand.ly/MaxiHuevosSorpresa \nIdeal para un lindo regalo en este Día del Niño \n¡Diviértete con La Granja de Zenón y junto a Bandai!\n\nLetra\n\nEn l',
  5: 'Paulo Londra "Adan y Eva"\n\n(Apple) ▶ https://apple.co/3pLkEV5\n(Spotify) ▶ https://open.spotify.com/album/1XBTyV1if5X9sU4IhpfH5R?si=VS1rXc94R3qbbymkR8B98g\n\nVisit Paulo Londra online:\nhttps://www.instagram.com/Paulolondra\n\nMusic Produced by OVY ON THE DRUMS:\nhttps://www.instagram.com/ovyonthedrums/\n\n',
  6: 'Ya puedes usar los filtros de La Vaca Lola en Facebook y en Instagram. Haz click en estos links: \nFacebook: https://bit.ly/FiltroLolaFB\nInstagram, según tu dispositivo: \nhttps://bit.ly/FiltroLolaInstaLite\nhttps://bit.ly/FiltroLolaInstagram\n\nTítulo Original: Perceherón\nAutor/Compositor: Dario Zamarreñ',
  7: 'Paulo Londra "Tal Vez"\n(Apple) ▶ https://apple.co/2KQarHz\n\n(Spotify) ▶ https://open.spotify.com/track/46lvmzK8wxAy66tjzXXSh0?si=UwCvYdvTRnS71f298Kaa8Q\n\n(YT Music) ▶https://music.youtube.com/watch?v=NPpELzyP4rw&feature=share\n\n\nVisit Paulo Londra online:\nhttps://www.instagram.com/Paulolondra\n\nMusic Pro',
  8: 'Letra\nYo conozco una vecina\nque ha comprado una gallina\nme parece una sardina enlatada.\n\nTiene las patas de alambre\nporque pasa mucho hambre\ny la pobre está todita desplumada.\n\nPone huevos en la sala\ny también en la cocina\npero nunca los pone en el corral.\n\n¡¡La Gallina!!\n¡¡Turuleca!!\nes un caso sing',
  9: '🎉¡Bienvenido a El Reino Infantil en Español! Activa la campanita y suscríbete a nuestro canal de YouTube para no perderte ningún vídeo - https://www.youtube.com/@ElReinoInfantil\n\n#ElReinoInfantil #Patitocuacua #cuacua #CancionesInfantiles\n\nMás contenido y diversión en nuestras redes sociales:\n😆🎶TikTo'},

```
'Canal_Propietario': {0: 'El Reino Infantil',
 1: 'El Reino Infantil',
 2: 'La Granja de Zenón',
 3: 'Paulo Londra',
 4: 'El Reino Infantil',
 5: 'Paulo Londra',
 6: 'El Reino Infantil',
 7: 'Paulo Londra',
 8: 'El Reino Infantil',
 9: 'El Reino Infantil'},
'Fecha_de_Publicación': {0: '2014-03-15',
 1: '2014-04-12',
 2: '2015-11-14',
 3: '2018-01-29',
 4: '2014-03-22',
 5: '2018-11-05',
 6: '2015-11-21',
 7: '2019-04-03',
 8: '2011-06-17',
 9: '2014-12-06'},
'Categoría_ID': {0: '10',
 1: '10',
 2: '10',
 3: '10',
 4: '10',
 5: '10',
 6: '10',
 7: '10',
 8: '10',
 9: '10'},
'Categoría': {0: 'Music',
 1: 'Music',
 2: 'Music',
 3: 'Music',
 4: 'Music',
 5: 'Music',
 6: 'Music',
 7: 'Music',
 8: 'Music',
 9: 'Music'},
'Duración_segundos': {0: 144,
 1: 144,
 2: 148,
 3: 232,
 4: 188,
 5: 261,
 6: 132,
 7: 272,
 8: 176,
 9: 117},
'URL_del_Video': {0: 'https://www.youtube.com/watch?v=0aZ7lPQ5EXs',
 1: 'https://www.youtube.com/watch?v=eNLjdPI9zdE',
 2: 'https://www.youtube.com/watch?v=ebVVuJN1WFM',
 3: 'https://www.youtube.com/watch?v=bX3S-_jUauc',
 4: 'https://www.youtube.com/watch?v=wPNQw8naE2Q',
 5: 'https://www.youtube.com/watch?v=aSjflT_J0Xo',
 6: 'https://www.youtube.com/watch?v=4nKcnfw9ggc',
 7: 'https://www.youtube.com/watch?v=NPpELzyP4rw',
 8: 'https://www.youtube.com/watch?v=XQaKFU3Fh_M',
 9: 'https://www.youtube.com/watch?v=KATq-Ws3xtM'},
```

```
 'Vistas': {0: '1955850346',
  1: '1864861747',
  2: '1294519692',
  3: '1252968961',
  4: '1238189581',
  5: '1230461273',
  6: '1137965275',
  7: '1096288919',
  8: '1043444320',
  9: '1033465785'},
 'Likes': {0: '4601114',
  1: '5194352',
  2: '2892446',
  3: '6632283',
  4: '2584070',
  5: '7313893',
  6: '2624980',
  7: '7010387',
  8: '3067245',
  9: '1645252'},
 'Dislikes': {0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0},
 'Favorite_Count': {0: '0',
  1: '0',
  2: '0',
  3: '0',
  4: '0',
  5: '0',
  6: '0',
  7: '0',
  8: '0',
  9: '0'},
 'Comment_Count': {0: '0',
  1: '0',
  2: '0',
  3: '250225',
  4: '0',
  5: '279499',
  6: '0',
  7: '212106',
  8: '0',
  9: '0'},
 'Insert_Date': {0: '2023-10-07',
  1: '2023-10-07',
  2: '2023-10-07',
  3: '2023-10-07',
  4: '2023-10-07',
  5: '2023-10-07',
  6: '2023-10-07',
  7: '2023-10-07',
  8: '2023-10-07',
  9: '2023-10-07'}}
```


In [27]:

```python
# Task 2:
def connect_to_Redshift():
    import psycopg2
    url="data-engineer-cluster.cyhh5bfevlmn.us-east-1.redshift.amazonaws.com"
    data_base="data-engineer-database"
    user="christian_r_coderhouse"

    try:
```

```
        conn = psycopg2.connect(
            host='data-engineer-cluster.cyhh5bfevlmn.us-east-1.redshift.amazonaws.com'
            dbname=data_base,
            user=user,
            password= pwd,
            port='5439'
        )
        print("Conectado a Redshift con éxito!")

    except Exception as e:
        print("No es posible conectar a Redshift")
        print(e)

    #Crear la tabla si no existe:
    with conn.cursor() as cur:
        cur.execute("""
            CREATE TABLE IF NOT EXISTS videos
            (
            Id_del_Video VARCHAR(50) primary key
            ,Título VARCHAR(350)
            ,Descripción VARCHAR(350)
            ,Canal_Propietario VARCHAR(255)
            ,Fecha_de_Publicación date
            ,Categoría_ID VARCHAR(50)
            ,Categoría VARCHAR(100)
            ,Duración_segundos INTEGER
            ,URL_del_Video NVARCHAR(500)
            ,Vistas INTEGER
            ,Likes INTEGER
            ,Dislikes INTEGER
            ,Favorite_Count INTEGER
            ,Comment_Count INTEGER
            ,Insert_Date date

            )
        """)
        conn.commit()
```

In [28]:
```
connect_to_Redshift()
```

```
Conectado a Redshift con éxito!
```

In [29]:
```
# Task 3:
def insert_data():

    import psycopg2


    conn = psycopg2.connect(
            host='data-engineer-cluster.cyhh5bfevlmn.us-east-1.redshift.amazonaws.com'
            dbname=data_base,
            user=user,
            password= pwd,
            port='5439'
        )



    data_dict = get_top_videos()
    df = pd.DataFrame(data_dict)
```

```
    #data = [(row['ID_del_Video'], row['Título'], row['Descripción'], row['Canal_Propi
    print(df)

    from psycopg2.extras import execute_values  # Añado esta línea para importar execu
    with conn.cursor() as cur:
        try:
            execute_values(
                cur,
                '''
                    INSERT INTO videos (ID_del_Video, Título, Descripción, Canal_Propi
                    VALUES %s
                    ''',
                    [tuple(row) for row in df.values],
                    #data,
                    page_size=len(df)
                )
            conn.commit()
            conn.close()
        except Exception as e:
            print("No es posible insertar datos")
            print(e)
```

In [30]:
```
insert_data()
```

```
[2023-10-07T22:29:32.162-0300] {__init__.py:49} INFO - file_cache is only supported w
ith oauth2client<4.0.0
```

	ID_del_Video	Título	Descripción	Canal_Propietario	Fecha_de_Publicación	Categoría_ID	Categoría	Duración_segundos	URL_del_Video	Vistas	Likes	Dislikes	Favorite_Count	Comment_Count	Insert_Date
0	0aZ7lPQ5EXs	El Gallo y la Pata - Canciones de la Granja de...	🎁 En estas Navidades, encuentra los productos ...	El Reino Infantil	2014-03-15	10	Music	144	https://www.youtube.com/watch?v=0aZ7lPQ5EXs	1955850346	4601114	0	0	0	2023-10-07
1	eNLjdPI9zdE	La Vaca Lola - Canciones de La Granja de Zenón 2	🎁 En estas Navidades, encuentra los productos ...	El Reino Infantil	2014-04-12	10	Music	144	https://www.youtube.com/watch?v=eNLjdPI9zdE	1864861747	5194353	0	0	0	2023-10-07
2	ebVVuJN1WFM	Bartolito - La Granja de Zenón 3	🎁 En estas Navidades, encuentra los productos ...	La Granja de Zenón	2015-11-14	10	Music	148	https://www.youtube.com/watch?v=ebVVuJN1WFM	1294519692	2892446	0	0	0	2023-10-07
3	bX3S-_jUauc	Paulo Londra ft Lenny Tavarez - Nena Maldicion...	Paulo Londra "Nena Maldicion"\n\n(Apple Music)...	Paulo Londra	2018-01-29	10	Music	232	https://www.youtube.com/watch?v=bX3S-_jUauc	1252968961	6632283	0	0	250225	2023-10-07
4	wPNQw8naE2Q	El Pollito Pío 3D - Canciones de la Granja de ...	Encuentra las Marionetas de la Granja de Zenón...	El Reino Infantil	2014-03-22	10	Music	188	https://www.youtube.com/watch?v=wPNQw8naE2Q	1238189581	2584070	0	0	0	2023-10-07
5	aSjflT_J0Xo	Paulo Londra - Adan y Eva (Official Video)	Paulo Londra "Adan y Eva"\n\n(Apple) ▶ https:...	Paulo Londra	2018-11-05	10	Music	261	https://www.youtube.com/watch?v=aSjflT_J0Xo	1230461273	7313893	0	0	279499	2023-10-07
6	4nKcnfw9ggc	Percherón - La Granja de Zenón 3	Ya puedes usar los filtros de La Vaca Lola en ...	El Reino Infantil	2015-11-21	10	Music	132	https://www.youtube.com/watch?v=4nKcnfw9ggc	1137965275	2624980	0	0	0	2023-10-07
7	NPpELzyP4rw	Paulo Londra - Tal Vez (Official Video)	Paulo Londra "Tal Vez"\n(Apple) ▶ https://appl...	Paulo Londra	2019-04-03	10	Music	272	https://www.youtube.com/watch?v=NPpELzyP4rw	1096288919	7010387	0	0	212106	2023-10-07
8	XQaKFU3Fh_M	La Gallina Turuleca - Canciones de la Granja d...	Letra\nYo conozco una vecina\nque ha comprado ...	El Reino Infantil	2011-06-17	10	Music	176	https://www.youtube.com/watch?v=XQaKFU3Fh_M	1043444320	3067245	0			
9	KATq-Ws3xtM	Patitos Cua Cua Cua - Canciones y clásicos inf...	🎉¡Bienvenido a El Reino Infantil en Español! A...	El Reino Infantil	2014-12-06	10	Music	117	https://www.youtube.com/watch?v=KATq-Ws3xtM	1033467344	1645252	0			

```
8               0             0  2023-10-07
9               0             0  2023-10-07
```

In [31]:

```python
# Task 4:
def verify_max_threshold():
    #import pandas as pd
    #import json


    # Carga los datos del archivo JSON:
    with open('config.json', 'r') as json_file:
        json_data = json.load(json_file)

    # Convierte los datos JSON en un DataFrame de pandas:
    json_df = pd.DataFrame(json_data["thresholds"]).T.reset_index()
    json_df.columns = ["Categoría", "Threshold_Min","Threshold_Max"]
    json_df_max= json_df[['Categoría','Threshold_Max']]

    # Me traigo el df que se originaba con la primera función "get_top_videos()" y lo
    dict_data=get_top_videos()
    new_df = pd.DataFrame.from_dict(dict_data)

    # Realiza el "join" entre new_df y json_df_max utilizando la columna "Categoría" c
    merged_df = new_df.merge(json_df_max, on='Categoría', how='left')


    # Convierte la columna "Vistas" al tipo de datos int64 (ya que originalmente tiene
    merged_df['Vistas'] = merged_df['Vistas'].astype('int64')


    # Itera a través de las filas del DataFrame.
    for index, row in merged_df.iterrows():
        if row['Vistas'] > row['Threshold_Max']:
            Título = row['Título']
            Threshold_Max = row['Threshold_Max']
            Vistas = row['Vistas']

            subject = f"Video {Título} is over the threshold"

            body_text = f"""
                Video '{Título}' is over the threshold.
                Max Threshold values is: {Threshold_Max}
                The Video '{Título}' reached {Vistas} Views
            """

            #Pass_Email = Variable.get("secret_pass_gmail")
            Pass_Email = 'iomo dnln ngzk slxa'
            smtp_server = 'smtp.gmail.com'
            smtp_port = 587
            sender_email = 'christian.jrivasn@gmail.com'
            password = Pass_Email

            try:
                msg = MIMEMultipart()
                msg['From'] = sender_email
                msg['To'] = sender_email
                msg['Subject'] = subject
                msg.attach(MIMEText(body_text, 'plain'))

                with smtplib.SMTP(smtp_server, smtp_port) as server:
```

```
                    server.starttls()
                    server.login(sender_email, password)
                    server.send_message(msg)

                print('El email fue enviado correctamente.')

            except Exception as exception:
                print(exception)
                print('El email no se pudo enviar.')

        #else:
            #pass  # No hace nada en el bloque "else"
            #print('No se envió ningún mail porque el Video no ha alcanzado el Thresho
```

In [32]:
```
verify_max_threshold()
```

```
[2023-10-07T22:29:34.177-0300] {__init__.py:49} INFO - file_cache is only supported w
ith oauth2client<4.0.0
El email fue enviado correctamente.
El email fue enviado correctamente.
```

In [33]:
```
# Task 5:
Pass_Email= Variable.get("secret_pass_gmail")
#Pass_Email = 'iomo dnln ngzk slxa'
smtp_server = 'smtp.gmail.com'
smtp_port = 587
sender_email = 'christian.jrivasn@gmail.com'
password = Pass_Email


def send_email():
        try:
            subject = 'Load of data of the top 10 videos with highest views in Youtube
            body_text = 'The load of data of the top 10 videos with highest views in Y

            msg = MIMEMultipart()
            msg['From'] = sender_email
            msg['To'] = sender_email
            msg['Subject'] = subject
            msg.attach(MIMEText(body_text, 'plain'))
            with smtplib.SMTP(smtp_server, smtp_port) as server:
                server.starttls()
                server.login(sender_email, password)
                server.send_message(msg)
            print('El email fue enviado correctamente.')

        except Exception as exception:
            print(exception)
            print('El email no se pudo enviar.')
```

In [34]:
```
send_email()
```

```
El email fue enviado correctamente.
```