Ejercicio 4- consumo de datos de google analytics para análisis de tendencia de un sitio web:

La información se procesa a día cerrado, La semana actual siempre presenta información parcial a menos que se este procesando el ultimo dia de esta.

# Consumo de información

## Paso 1 - Elección de herramienta

* Orquestador Apache Airflow (Seguir pasos de configuración en el link)
  + Se pueden usar los Operadores como se indica en el link:

# [airflow.providers.google.marketing\_platform.operators.analytics](https://airflow.apache.org/docs/apache-airflow-providers-google/stable/_api/airflow/providers/google/marketing_platform/operators/analytics/index.html#module-airflow.providers.google.marketing_platform.operators.analytics)

* + Se puede generar una conexión jdbc y descargar un csv (Opción seleccionada en el presente documento)
    - https://www.cdata.com/kb/tech/ganalytics-jdbc-apache-airflow.rst

## 

## Paso 2 - Generación de conexión airflow

Utilizaria el diseñador de cadenas de conexión integrado en el controlador JDBC de Google Analytics: [source](https://www.cdata.com/kb/tech/ganalytics-jdbc-apache-airflow.rst#viewSource)

*java -jar cdata.jdbc.googleanalytics.jar*

Obtendrìa la licencia para el controlador.

Una vez realizado esto generaría la conexión en airflow completando los parametros de esta:

Id. de conexión : nombre de la conexión, es decir: googleanalytics\_jdbc

Tipo de conexión : Conexión JDBC

URL de conexión : la URL de conexión JDBC de arriba, es decir: jdbc:googleanalytics:RTK=5246...;Profile=MyProfile;InitiateOAuth=GETANDREFRESH)

Clase de controlador : cdata.jdbc.googleanalytics.GoogleAnalyticsDriver

Ruta del controlador : RUTA/TO/cdata.jdbc.googleanalytics.jar

## Paso 3 - Generar un DAG (tarea de consumo)

Para extraer la información tanto para generar un archivo csv o guardar en un dataframe temporal utilizaria un código similar al siguiente para extracción de .

import time

from datetime import datetime

from airflow.decorators import dag, task

from airflow.providers.jdbc.hooks.jdbc import JdbcHook

import pandas as pd

# Declare Dag

@dag(dag\_id="google analytics\_hook", schedule\_interval="0 10 \* \* \*", start\_date=datetime(2022,2,15), catchup=False, tags=['load\_csv'])

# Define Dag Function

def extract\_and\_load():

# Define tasks

@task()

def jdbc\_extract():

try:

hook = JdbcHook(jdbc\_conn\_id="jdbc")

sql = """ select \* from Account """

df = hook.get\_pandas\_df(sql)

df.to\_csv("/{some\_file\_path}/{name\_of\_csv}.csv",header=False, index=False, quoting=1)

# print(df.head())

print(df)

tbl\_dict = df.to\_dict('dict')

**return** tbl\_dict

except Exception as e:

print("Data extract error: " + str(e))

jdbc\_extract()

sf\_extract\_and\_load = extract\_and\_load()

# Carga de información:

## Paso 1 - Generar la conexión a bigquery desde airflow:

Se obtiene el keyfile json de acceso y se carga la conexión.

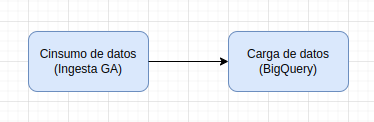
## paso - 2 - Load de los datos en tabla de stage:

Se siguen los pasos para la carga de información espe

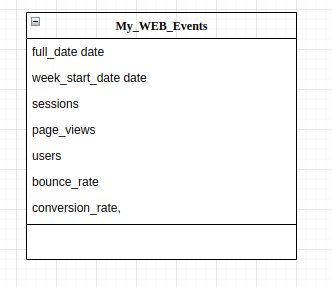
# [airflow.providers.google.cloud.hooks.bigquery](https://airflow.apache.org/docs/apache-airflow-providers-google/2.0.0/_api/airflow/providers/google/cloud/hooks/bigquery/index.html#module-airflow.providers.google.cloud.hooks.bigquery)

## Paso -3 Integración

se integra esta tarea de carga como tarea relativa a la generada en “Consumo de información” el agregado de columnas calculadas como “week\_start\_date” se hace en el dataframe de carga on the fly



# Modelo de datos



# Query:

SELECT

week\_start\_date,

sessions,

page\_views,

users,

bounce\_rate,

conversion\_rate,

sessions/LAG(sessions,1, NULL) OVER( ORDER BY week\_start\_date) change\_rate\_sessions,

page\_views/LAG(page\_views,1, NULL) OVER( ORDER BY week\_start\_date) change\_rate\_views,

users/LAG(users,1, NULL) OVER( ORDER BY week\_start\_date) change\_rate\_users,

bounce\_rate/LAG(bounce\_rate,1, NULL) OVER( ORDER BY week\_start\_date) change\_rate\_bounce\_rate,

conversion\_rate/LAG(conversion\_rate,1, NULL) OVER( ORDER BY week\_start\_date) change\_rate\_conversion\_rate

FROM production.My\_Web\_Events