Clase 8

Consigna: Por cada ejercicio, escribir el código y agregar una captura de pantalla del resultado obtenido.

Diccionario de datos:
https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020?select=results.csv

1. Crear la siguientes tablas externas en la base de datos f1 en hive:
    a. driver_results (driver_forename, driver_surname, driver_nationality, points)
    b. constructor_results (constructorRef, cons_name, cons_nationality, url, points)

Creación base f1

```
hive> create database f1;
OK
Time taken: 0.698 seconds
```

Creación tabla driver_results en base f1

```
hive> create external table f1.driver_results(driver_forename string, driver_surname string,
driver_nationality string, points int)
    > comment 'Driver results in F1'
    > row format delimited
    > fields terminated by ','
    > stored as textfile
    > location '/tables/f1/driver_results' ;
```

```
hive> create external table f1.driver_results(driver_forename string, driver_surname string, driver_nationality string,
points int)
    > comment 'Driver results in F1'
    > row format delimited
    > fields terminated by ','
    > stored as textfile
    > location '/table/f1/driver_results' ;
OK
Time taken: 0.181 seconds
```

Creación tabla constructor_results en base f1

```
hive> create external table f1.constructor_results(constructorRef string, cons_name string,
cons_nationality string, url string, points int)
    > comment 'Constructor results in F1'
    > row format delimited
    > fields terminated by ','
    > stored as textfile
    > location '/tables/f1/constructor_results' ;
```

```
hive> create external table f1.constructor_results(constructorRef string, cons_name string, cons_nationality string, url
 string, points int)
    > comment 'Constructor results in F1'
    > row format delimited
    > fields terminated by ','
    > stored as textfile
    > location '/table/f1/constructor_results' ;
OK
Time taken: 0.057 seconds
```

2. En Hive, mostrar el esquema de driver_results y constructor_results

```
hive> describe driver_results;
OK
driver_forename         string
driver_surname          string
driver_nationality      string
points                  int
Time taken: 0.039 seconds, Fetched: 4 row(s)
```

```
hive> describe constructor_results;
OK
constructorref          string
cons_name               string
cons_nationality        string
url                     string
points                  int
Time taken: 0.033 seconds, Fetched: 5 row(s)
```

3. Crear un archivo .bash que permita descargar los archivos mencionados abajo e ingestarlos en HDFS:
results.csv https://data-engineer-edvai-public.s3.amazonaws.com/results.csv
drivers.csv https://data-engineer-edvai-public.s3.amazonaws.com/drivers.csv
constructors.csv https://data-engineer-edvai-public.s3.amazonaws.com/constructors.csv
races.csv https://data-engineer-edvai-public.s3.amazonaws.com/races.csv

Script ingest_f1.sh

```
#Descarga de archivo results.csv
wget -P /home/hadoop/landing
https://data-engineer-edvai-public.s3.amazonaws.com/results.csv

#Descarga de archivo drivers.csv
wget -P /home/hadoop/landing
https://data-engineer-edvai-public.s3.amazonaws.com/drivers.csv

#Descarga de archivo constructors.csv
wget -P /home/hadoop/landing
https://data-engineer-edvai-public.s3.amazonaws.com/constructors.csv

#Descarga de archivo races.csv
```

wget -P /home/hadoop/landing https://data-engineer-edvai-public.s3.amazonaws.com/races.csv

#Copia de archivos desde carpeta Landing al HDFS.
/home/hadoop/hadoop/bin/hdfs dfs -put /home/hadoop/landing/results.csv /ingest
/home/hadoop/hadoop/bin/hdfs dfs -put /home/hadoop/landing/drivers.csv /ingest
/home/hadoop/hadoop/bin/hdfs dfs -put /home/hadoop/landing/constructors.csv /ingest
/home/hadoop/hadoop/bin/hdfs dfs -put /home/hadoop/landing/races.csv /ingest

#Borrado de archivos de carpeta Landing
rm /home/hadoop/landing/results.csv
rm /home/hadoop/landing/drivers.csv
rm /home/hadoop/landing/constructors.csv
rm /home/hadoop/landing/races.csv

```
hadoop@5f1a3da9dadf:~/scripts$ ls -rtl ingest_f1.sh
-rw-rw-r-- 1 hadoop hadoop 1074 Oct 20 22:30 ingest_f1.sh
```

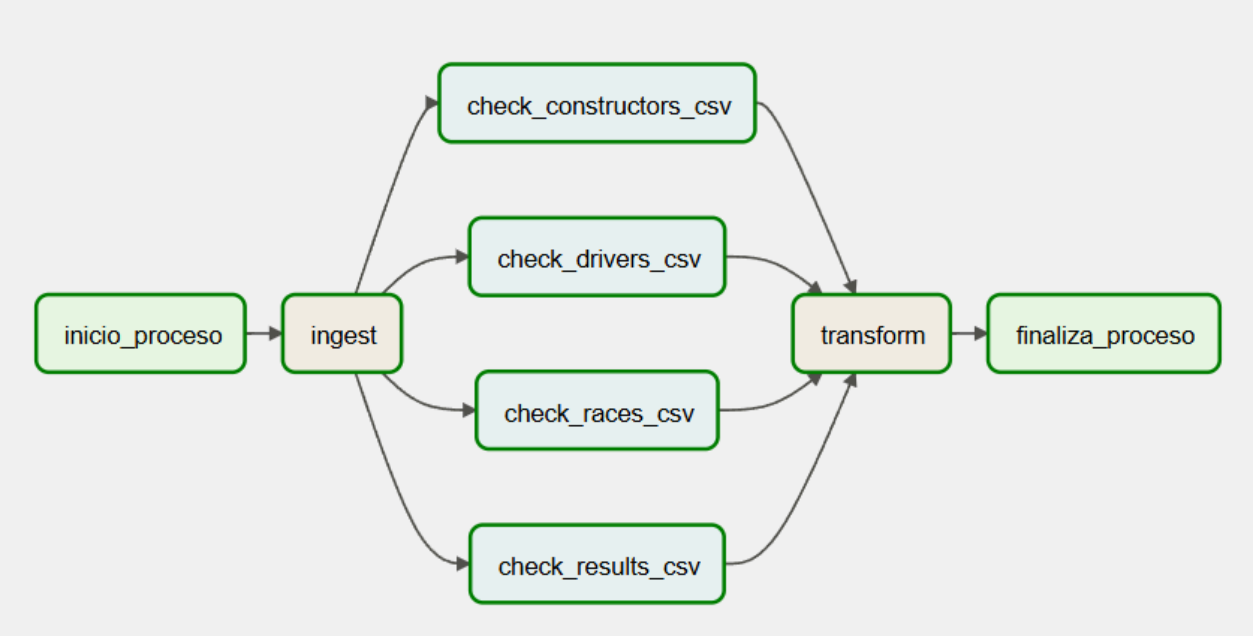
4. Generar un archivo .py que permita, mediante Spark:
    a. insertar en la tabla driver_results los corredores con mayor cantidad de puntos en la historia.
    b. insertar en la tabla constructor_result quienes obtuvieron más puntos en el Spanish Grand Prix en el año 1991

Archivo de transformación "t_clase8.py"

```
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext('local')
spark = SparkSession(sc)

from pyspark.sql import HiveContext
hc = HiveContext(sc)

##Leyendo archivos csv de carpeta /ingest

df_res =
spark.read.option("header","true").csv("hdfs://172.17.0.2:9000/ingest/results.csv")
df_drs =
spark.read.option("header","true").csv("hdfs://172.17.0.2:9000/ingest/drivers.csv")
df_cos =
spark.read.option("header","true").csv("hdfs://172.17.0.2:9000/ingest/constructors.csv")
df_ras = spark.read.option("header","true").csv("hdfs://172.17.0.2:9000/ingest/races.csv")

#Crea las vistas para el punto A del ejercicio sobre tabla driver_results

df_res.createOrReplaceTempView("results_vista")
df_drs.createOrReplaceTempView("drivers_vista")

#Filtro columnas para el punto A

df_driver_filtro = spark.sql("select cast(a.forename as string) as driver_forename,
```

```
cast(a.surname as string) as driver_surname, cast(a.nationality as string) as
driver_nationality, cast(sum(b.points)as int) as points from drivers_vista a inner join
results_vista b on a.driverId = b.driverId group by
driver_forename,driver_surname,a.nationality order by points desc")

#Crea las vistas para el punto B del ejercicio sobre tabla driver_results

df_cos.createOrReplaceTempView("constructor_vista")
df_ras.createOrReplaceTempView("races_vista")

#Filtro columnas para el punto B

df_construc_filtro = spark.sql("select cast(a.constructorRef as string) as constructorRef,
cast(a.name as string) as cons_name, cast(a.nationality as string) as cons_nationality,
cast(a.url as string), cast(sum(b.points) as int) as points from constructor_vista a inner
join results_vista b on a.constructorId = b.constructorId inner join races_vista c on
b.raceId = c.raceId where c.name like '%Spanish Grand Prix%' and c.year = 1991 group
by constructorRef,cons_name,cons_nationality,a.url order by points desc")

#Hago la vista para insertar punto A

df_driver_filtro.createOrReplaceTempView("driver_insert")

#Hago la vista para insertar punto B

df_construc_filtro.createOrReplaceTempView("construc_insert")

#Hace load en HIVE para punto A

hc.sql("insert into f1.driver_results select * from driver_insert")

#Hace load en HIVE para punto B

hc.sql("insert into f1.constructor_results select * from construc_insert")
```

5. Realizar un proceso automático en Airflow que orqueste los archivos creados en los puntos 3 y 4. Correrlo y mostrar una captura de pantalla (del DAG y del resultado en la base de datos)



El DAG:

```
from datetime import timedelta
from airflow import DAG
from airflow.providers.apache.hdfs.sensors.web_hdfs import WebHdfsSensor
from airflow.operators.bash import BashOperator
from airflow.operators.dummy import DummyOperator
from airflow.utils.dates import days_ago

args = {
    'owner': 'airflow',
}

# Configuracion del HdfsSensor
HDFS_CONN_ID = 'webhdfs_default'
HDFS_FILE_PATH = '/ingest'

with DAG(
    dag_id='ingest_clase8',
    default_args=args,
    schedule_interval='0 0 * * *',
    start_date=days_ago(2),
    dagrun_timeout=timedelta(minutes=60),
    tags=['ingest', 'transform'],
    params={"example_key": "example_value"},
) as dag:

    inicio_proceso = DummyOperator(
        task_id="inicio_proceso",
    )

    finaliza_proceso = DummyOperator(
        task_id='finaliza_proceso',
    )
```

```
    ingest = BashOperator(
        task_id='ingest',
        bash_command='/usr/bin/sh /home/hadoop/scripts/ingest_f1.sh ',
    )

# Sensores para cada archivo individual
    check_results = WebHdfsSensor(
        task_id='check_results_csv',
        filepath=f'{HDFS_FILE_PATH}/results.csv',
        webhdfs_conn_id=HDFS_CONN_ID, poke_interval=10, timeout=600,
    )
    check_constructors = WebHdfsSensor(
        task_id='check_constructors_csv',
        filepath=f'{HDFS_FILE_PATH}/constructors.csv',
        webhdfs_conn_id=HDFS_CONN_ID, poke_interval=10, timeout=600,
    )
    check_races = WebHdfsSensor(
        task_id='check_races_csv',
        filepath=f'{HDFS_FILE_PATH}/races.csv',
        webhdfs_conn_id=HDFS_CONN_ID, poke_interval=10, timeout=600,
    )
    check_drivers = WebHdfsSensor(
        task_id='check_drivers_csv',
        filepath=f'{HDFS_FILE_PATH}/drivers.csv',
        webhdfs_conn_id=HDFS_CONN_ID, poke_interval=10, timeout=600,
    )


    transform = BashOperator(
        task_id='transform',
        bash_command='ssh hadoop@172.17.0.2
   /home/hadoop/spark/bin/spark-submit --files
   /home/hadoop/hive/conf/hive-site.xml /home/hadoop/scripts/t_clase8.py ',
    )


   # El flujo de tareas
    inicio_proceso >> ingest

    # Todos los sensores esperan a que termine la ingestion
    ingest >> [check_results, check_constructors, check_races,
   check_drivers]

    # La transformacion espera a que TODOS los sensores finalicen
    [check_results, check_constructors, check_races, check_drivers] >>
   transform

    transform >> finaliza_proceso


if __name__ == "__main__":
    dag.cli()
```

Hive: tabla drivers_results

En spark había quedado así el dataframe:

```
>>> df_driver_filtro.show(20)
+---------------+-------------+------------------+------+
|driver_forename|driver_surname|driver_nationality|points|
+---------------+-------------+------------------+------+
|          Lewis|      Hamilton|           British|  4820|
|      Sebastian|        Vettel|            German|  3098|
|            Max|    Verstappen|             Dutch|  2912|
|       Fernando|       Alonso|           Spanish|  2329|
|           Kimi|     Räikkönen|           Finnish|  1873|
|        Valtteri|       Bottas|           Finnish|  1788|
|           Nico|      Rosberg|            German|  1594|
|         Sergio|        Pérez|           Mexican|  1585|
|        Michael|    Schumacher|            German|  1566|
|        Charles|      Leclerc|         Monegasque|  1363|
|         Daniel|     Ricciardo|         Australian|  1320|
|         Jenson|       Button|           British|  1235|
|         Carlos|        Sainz|           Spanish|  1203|
```

En hive:

```
hive> set hive.resultset.use.unique.column.names=false
    > ;
hive> select * from driver_results limit 10;
OK
Lewis   Hamilton        British 4820
Sebastian       Vettel  German  3098
Max     Verstappen      Dutch   2912
Fernando        Alonso  Spanish 2329
Kimi    Räikkönen       Finnish 1873
Valtteri        Bottas  Finnish 1788
Nico    Rosberg German  1594
Sergio  Pérez   Mexican 1585
Michael Schumacher      German  1566
Charles Leclerc Monegasque      1363
```

Hive: tabla constructor_results

En spark había quedado así el dataframe:

```
>>> df_construc_filtro.show()
+--------------+------------+---------------+--------------------+------+
|constructorRef|   cons_name|cons_nationality|                 url|points|
+--------------+------------+---------------+--------------------+------+
|      williams|    Williams|        British|http://en.wikiped...|    14|
|       ferrari|     Ferrari|        Italian|http://en.wikiped...|     9|
|       mclaren|     McLaren|        British|http://en.wikiped...|     2|
|      benetton|    Benetton|        Italian|http://en.wikiped...|     1|
|     fondmetal|   Fondmetal|        Italian|http://en.wikiped...|     0|
|       tyrrell|     Tyrrell|        British|http://en.wikiped...|     0|
```

En hive:

```
hive> set hive.resultset.use.unique.column.names=false
    > ;
hive> select * from constructor_results;
OK
williams        Williams        British http://en.wikipedia.org/wiki/Williams_Grand_Prix_Engineering    14
ferrari Ferrari Italian http://en.wikipedia.org/wiki/Scuderia_Ferrari   9
mclaren McLaren British http://en.wikipedia.org/wiki/McLaren    2
benetton        Benetton        Italian http://en.wikipedia.org/wiki/Benetton_Formula   1
fondmetal       Fondmetal       Italian http://en.wikipedia.org/wiki/Fondmetal  0
leyton  Leyton House    British http://en.wikipedia.org/wiki/Leyton_House       0
minardi Minardi Italian http://en.wikipedia.org/wiki/Minardi    0
tyrrell Tyrrell British http://en.wikipedia.org/wiki/Tyrrell_Racing     0
brabham Brabham British http://en.wikipedia.org/wiki/Brabham    0
lola    Lola    British http://en.wikipedia.org/wiki/MasterCard_Lola    0
ligier  Ligier  French  http://en.wikipedia.org/wiki/Ligier     0
ags     AGS     French  http://en.wikipedia.org/wiki/Automobiles_Gonfaronnaises_Sportives       0
dallara Dallara Italian http://en.wikipedia.org/wiki/Dallara    0
team_lotus      Team Lotus      British http://en.wikipedia.org/wiki/Team_Lotus 0
```