Antonio Gallego Peñalver

Examen BDA1 - Tarea práctica

Ejercicio 1. Preparación de los datos

- Crea un directorio “examen” en el home del usuario y sitúate en él.
- Descarga los ficheros de las siguientes URLs (wget enlace)
- Descomprime el primer fichero

```
hadoop@myubuntu:~/examen$ dir
city_temperature.csv  city_temperature.csv.tar.gz  olive.csv  palm.csv  sunflowerseed.csv
```

- En el sistema de archivos HDFS del clúster, crea un directorio “/examen”
- Copia los ficheros descargados al directorio “/examen” del clúster hdfs

```
hadoop@myubuntu:~/examen$ hdfs dfs -mkdir /examen
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
hadoop@myubuntu:~/examen$ hdfs dfs -ls /
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Found 5 items
drwxr-xr-x   - hadoop supergroup          0 2024-09-14 16:10 /curso
drwxr-xr-x   - hadoop supergroup          0 2024-12-04 17:33 /examen
drwxrwxrwx   - hadoop supergroup          0 2024-09-15 15:22 /logs
drwxrwxr-x   - hadoop supergroup          0 2024-09-15 14:29 /tmp
drwxr-xr-x   - hadoop supergroup          0 2024-09-15 00:30 /user
```

```
hadoop@myubuntu:~/examen$ hdfs dfs -put ~/examen/city_temperature.csv /examen
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
hadoop@myubuntu:~/examen$ hdfs dfs -put ~/examen/olive.csv /examen
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
hadoop@myubuntu:~/examen$ hdfs dfs -put ~/examen/palm.csv /examen
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
hadoop@myubuntu:~/examen$ hdfs dfs -put ~/examen/sunflowerseed.csv /examen
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
hadoop@myubuntu:~/examen$ hdfs dfs -ls /examen
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Found 4 items
-rw-r--r--   1 hadoop supergroup  140600832 2024-12-04 17:37 /examen/city_temperature.csv
-rw-r--r--   1 hadoop supergroup      86251 2024-12-04 17:37 /examen/olive.csv
-rw-r--r--   1 hadoop supergroup     428679 2024-12-04 17:37 /examen/palm.csv
-rw-r--r--   1 hadoop supergroup     346199 2024-12-04 17:37 /examen/sunflowerseed.csv
```

Ejercicio 2. Tablas externas en Hive

- Crea una carpeta “/examen/oil” en el clúster hdfs y mueve el fichero “olive.csv” a dicha carpeta.

```
hadoop@myubuntu:~/examen$ hdfs dfs -mkdir /examen/oil
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
hadoop@myubuntu:~/examen$ hdfs dfs -ls /examen
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Found 5 items
-rw-r--r--   1 hadoop supergroup  140600832 2024-12-04 17:37 /examen/city_temperature.csv
drwxr-xr-x   - hadoop supergroup          0 2024-12-04 17:46 /examen/oil
-rw-r--r--   1 hadoop supergroup      86251 2024-12-04 17:37 /examen/olive.csv
-rw-r--r--   1 hadoop supergroup     428679 2024-12-04 17:37 /examen/palm.csv
-rw-r--r--   1 hadoop supergroup     346199 2024-12-04 17:37 /examen/sunflowerseed.csv
```

```
hadoop@myubuntu:~/examen$ hdfs dfs -mv /examen/olive.csv /examen/oil/
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
hadoop@myubuntu:~/examen$ hdfs dfs -ls /examen/oil
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Found 1 items
-rw-r--r--   1 hadoop supergroup      86251 2024-12-04 17:37 /examen/oil/olive.csv
```

- Crea una tabla externa sobre el directorio “oil”, con las columnas que tiene el fichero. Sustituye el texto marcado por los valores adecuados.

```
hadoop@myubuntu:~/examen$ hadoop@myubuntu:~/examen$ hive
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Beeline version 4.0.0 by Apache Hive
beeline>
hadoop@myubuntu:~/examen$ hiveserver2 &
[1] 6774
hadoop@myubuntu:~/examen$ Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
2024-12-04 18:35:45: Starting HiveServer2
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Hive Session ID = 1ccd0e07-e5b0-499b-9b44-a8f72399eca5
Hive Session ID = 32d42b58-8603-47c1-a667-030aabbcb03e
```

```
CREATE EXTERNAL TABLE oil_prod (
  country STRING,
  year INT,
  Beginning_Stocks DOUBLE,
  Domestic_Consumption DOUBLE,
  Ending_Stocks DOUBLE,
  Exports DOUBLE,
  Feed_Waste DOUBLE,
  Food_Use DOUBLE,
  Imports DOUBLE,
  Industrial DOUBLE,
  Production DOUBLE,
  Total_Distribution DOUBLE,
  Total_Supply DOUBLE
)
ROW FORMAT SERDE
'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  "separatorChar" = ",",
  "quoteChar" = "\"",
  "escapeChar" = "\\"
)
STORED AS TEXTFILE
LOCATION '/examen/oil'
TBLPROPERTIES ("skip.header.line.count"="1");
```

- Realiza las siguientes consultas y captura pantalla de los resultados de cada una, incluyéndolas en el fichero de respuesta:
    - Muestra las 10 primeras filas de la tabla

```
1 SELECT * FROM oil_prod LIMIT 10;
```

	oil_prod.country	oil_prod.year	oil_prod.beginning_stoc
1	Algeria	1964	0
2	Algeria	1965	0
3	Algeria	1966	0
4	Algeria	1967	0
5	Algeria	1968	0
6	Algeria	1969	0
7	Algeria	1970	0
8	Algeria	1971	0
9	Algeria	1972	2
10	Algeria	1973	2

    - Cuenta el número total de registros

        No he sido capaz de hacerlo, la consulta alcanzaba los 8 minutos incluso con LIMIT 50.
    - Obtén país y año de máxima producción (pista: 2 sentencias sql)

        No he sido capaz de hacerlo, la consulta alcanzaba los 8 minutos.

- Mueve los ficheros “palm.csv” y “sunflowerseed.csv” al directorio “/examen/oil” en el clúster hdfs, y repite las consultas del punto 3, guardando las capturas de pantalla.

```
hadoop@myubuntu:~/examen$ hdfs dfs -mv /examen/palm.csv /examen/oil/
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
hadoop@myubuntu:~/examen$ hdfs dfs -mv /examen/sunflowerseed.csv /examen/oil/
hadoop@myubuntu:~/examen$ hdfs dfs -ls /examen/oil
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Found 3 items
-rwxr-xr-x   1 hadoop supergroup      86251 2024-12-04 17:37 /examen/oil/olive_copy_1.csv
-rw-r--r--   1 hadoop supergroup     428679 2024-12-04 17:37 /examen/oil/palm.csv
-rw-r--r--   1 hadoop supergroup     346199 2024-12-04 17:37 /examen/oil/sunflowerseed.csv
```

Ejercicio 3. Tablas temporales, formato parquet y particiones

- Crea una tabla temporal, añade propiedad para que se salte la primera línea de cabecera y carga los datos del fichero “city_temperature.csv”.

```
CREATE EXTERNAL TABLE staging_temperatures (
  Region STRING,
  Country STRING,
  State STRING,
  City STRING,
  Month INT,
  Day INT,
  Year INT,
  AvgTemperature FLOAT
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  "separatorChar" = ",",
  "quoteChar" = "\"",
  "escapeChar" = "\\"
)
STORED AS TEXTFILE
LOCATION '/examen/oil'
TBLPROPERTIES ("skip.header.line.count"="1");
```

- Realiza dos consultas para comprobar los datos:
    - 100 primeras filas.

```
SELECT * FROM staging_temperatures LIMIT 100;
```

	staging_temperatures.region	staging_temperatures.coun
1	Algeria	1964
2	Algeria	1965
3	Algeria	1966
4	Algeria	1967
5	Algeria	1968
6	Algeria	1969
7	Algeria	1970
8	Algeria	1971
9	Algeria	1972
10	Algeria	1973
11	Algeria	1974
12	Algeria	1975
13	Algeria	1976
14	Algeria	1977
15	Algeria	1978

consulta1.csv

    - 10 primeras filas con valores de AvgTemperature negativos.

```
SELECT *
FROM staging_temperatures
WHERE AvgTemperature < 0
LIMIT 10;
```

No he sido capaz de hacerlo, la consulta me sale vacía

- Comprueba los ficheros almacenados en el warehouse de Hive. Captura pantalla e inclúyela en el fichero a entregar.

```
hadoop@myubuntu:~/examen$ hdfs dfs -ls /user/hive/warehouse
Picked up _JAVA_OPTIONS: -Djava.io.tmpdir=/tmp
Found 1 items
drwxr-xr-x   - hadoop hadoop          0 2024-09-15 15:32 /user/hive/warehouse/employees
```

- Crea una tabla en formato parquet sobre los datos de la tabla temporal, filtrando aquellos que tengan valores incorrectos (temperatura <> -99)

```
CREATE TABLE temperatures
STORED AS PARQUET
AS
SELECT *
FROM staging_temperatures
WHERE AvgTemperature <> -99;
```

No he sido capaz de hacerlo, la consulta alcanzaba los 8 minutos.

- Para la región 'Europe' y el año 2015, muestra las temperaturas mayores de 85, incluyendo ciudad, mes, día y temperatura. Obtén el plan de ejecución de la consulta y captura la pantalla de resultado donde se muestra el número de registros leídos antes de filtrar, e inclúyela en el fichero a entregar.

```
SELECT City, Month, Day, AvgTemperature
FROM staging_temperatures
WHERE Region = 'Europe'
  AND Year = 2015
  AND AvgTemperature > 85;
```

```
EXPLAIN SELECT City, Month, Day, AvgTemperature
FROM staging_temperatures
WHERE Region = 'Europe'
  AND Year = 2015
  AND AvgTemperature > 85;
```

	Explain
1	STAGE DEPENDENCIES:
2	Stage-0 is a root stage
3	
4	STAGE PLANS:
5	Stage: Stage-0
6	Fetch Operator
7	limit: -1
8	Processor Tree:
9	TableScan
10	alias: staging_temperatures
11	filterExpr: ((region = 'Europe') and (UDFToDouble(year) = 2015.
12	Statistics: Num rows: 872 Data size: 905280 Basic stats: COM

consulta2.csv

- Crea una tabla en formato parquet, particionada por country y year. Activa antes las particiones.

```
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;
```

```
✔ Success.
```

```
CREATE TABLE temperaturesbyRegionYear (
  Country STRING,
  State STRING,
  City STRING,
  Month INT,
  Day INT,
  AvgTemperature FLOAT
)
PARTITIONED BY (Region STRING, Year INT)
STORED AS PARQUET;
```

```
STORED AS PARQUET
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hadoop_20241204205019_ab0
0baf0-32ac-44e0-a383-2041543cc429); Time taken: 0.354 seconds
...
✔ Success.
```

- Inserta los datos de la tabla temperaturas a partir de 2010 (por la limitación de memoria de nuestra máquina virtual).

```
INSERT INTO TABLE temperaturesbyRegionYear PARTITION (Region, Year)
SELECT Country, State, City, Month, Day, AvgTemperature, Region, Yea
FROM staging_temperatures
WHERE Region IS NOT NULL
  AND Year IS NOT NULL
  AND Year > 2010;
```

No he sido capaz de hacerlo, la consulta alcanzaba los 8 minutos.