Práctica - Hive and Pig

Para realizar esta práctica se utilizará la VM de quickstart provista por cloudera. Se recomienda generar una carpeta (por ejemplo **env**) y correr la consola en dicha carpeta ya que las aplicaciones generan archivos que dejan en el CWD.

Ejercicios de Pig.

La consola de pig se accede mediante a el comando **pig** y a partir de ahí se pueden correr los comandos que se quieran. Alternativa para armar los scripts con más comodidad es escribirlos en un archivo y correr **pig -f <nombre_archivo>**.

1. Armemos el wordcount para pig. Para ello en la consola de pig realice los siguientes pasos:
    a. Leer el contenido de un libro:

```
book = LOAD 'hdfs:/datasets/datasets-small/books/dracula.txt' USING
PigStorage() AS (lines:chararray);
```

    b. tokenizar en palabras:

```
words = FOREACH book GENERATE  FLATTEN(TOKENIZE(lines)) as word;
```

    c. Limpiar las palabras (sacamos espacios, comillas, comas, etc):

```
wordsCleaned = FOREACH words GENERATE  TRIM(REGEX_EXTRACT(word,
'([^a-zA-Z]*)([a-zA-Z]+)([^a-zA-Z]*)', 2)) as word;
```

    d. Agrupar por palabra:

```
wordsGrouped = GROUP wordsCleaned BY word;
```

    e. Contar las apariciones cada palabra.\

```
wordsAggregated = FOREACH wordsGrouped GENERATE group as word,
COUNT(wordsCleaned);
```

    f. ordenar por cantidad

```
wordsSorted = ORDER wordsAggregated BY $1 DESC;
```

    g. Escribir la salida a archivos:

```
STORE wordsSorted INTO 'hdfs:/results/pig/word-count-sc';
```

    h. Compruebe el resultado final en la carpeta indicada en el ultimo punto.

2. Ahora realicemos la carga de la información de las medidas de temperatura. Para ello al haber campos de distinto tipo requerimos proveer a pig de un esquema.

```
month07 = LOAD 'hdfs:/datasets/datasets-med/temp-hourly/199607hourly.txt' USING
PigStorage(',') AS (wban:int, date:chararray, time:chararray,
stationType:chararray, maintenanceIndicator:chararray , skyConditions:chararray,
visibility:chararray, weatherType:chararray, dryBulbTemp:int, dewPointTemp:int,
wetBulbTemp:int,  relativeHumidity:chararray, windSpeed:float,
windDirection:chararray, windCharGusts:chararray, windChar:chararray,
stationPressure:chararray, pressureTendency:int, seaLevelPressure:int);
```

3. Para evitar tener que utilizar el esquema cada vez se puede pedir a pig que lo guarde junto al contenido:

```
STORE month07 INTO 'hdfs:/results/pig/weather-schema/' USING PigStorage(':',
'-schema');
```

4. Una vez hecho eso comprar que en la carpeta de salida se encuentra el archivo **.pig_schema**, copiarlo usando hdfs a la carpeta del dataset, e incorporar otro archivo sin indicarle el esquema:

```
month08 = LOAD 'hdfs:/datasets/datasets-med/temp-hourly/199608hourly.txt'  USING
PigStorage(',');
```

5. Filtrar el *month08* quedándose solamente con los valores tomados cuando le día estaba despejado. **Tip:** el campo skyConditions con valor CLR indica despejado (ojo que el campo tiene espacios).
6. Obtener de este último listado solamente los campos año, mes, dia, condicion del cielo y temperatura (dryBulbTemp). **Tip:** dia mes y año se obtienen haciendo substring del campo fecha.
7. Agrupar por mes y año la información obtenida.
8. Una vez agrupada obtener el promedio de la temperatura y la cantidad de mediciones de tomadas para cada grupo.
9. Guardar en un directorio la salida.

Ejercicios de HIVE

Para poder utilizar hive simplemente entrar en una consola y en el directorio que se quiera y escribir **hive** seguido de apretar **enter**. Luego de unos segundos la consola de hive estará disponible.

1. Crear la tabla words a partir de la siguiente instrucción:

```
CREATE TABLE words(word STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ' ' LINES
TERMINATED BY '\n';
```

   Lamentablemente hay que copiar y pegar en la consola y dar enter por cada línea

2. Incorporar el contenido del libro dracula.txt mediante a la siguiente instrucción:

```
LOAD DATA INPATH 'hdfs:/datasets/datasets-small/books' into table words;
```

   Un potencial problema de esta instrucción es que borra los archivos del directorio original, por si lo necesitamos en futuros ejercicios restaurarlo de la copia local nuevamente.
3. Cómo copiar y pegar por línea en la consola de hive resulta engorroso vamos a crear un archivo y ejecutar las querys desde el mismo. Entonces crear un archivo con la consulta para obtener todos los elementos de la tabla en un archivo (por ejemplo **hive.sql**). Luego salir de la consola de hive y correr:

```
$> hive -f hive.sql
```

4. Crear la consulta que haga el conteo de cuantas veces cada palabra aparece en el texto y solo muestre aquellas que aparecen más de 50 veces. **Tips:** hive tiene disponible las funciones *lower(string)* y *regexp_replace(intial, to_replace, replace_str)*.
5. Agregar a la consulta anterior que solo cuente para las palabras que comienzan con w.
   Tip: *substring(string, start, len)*.
6. Realizar el ejercicio 2 del Tutorial para importar datos desde HDFS a Hive (or Impala) e realizar unas consultas.
7. Indicar para los tokenized logs para cada url, cuantos accesos hubo para cada código de respuesta enviado (el código está en el campo code1).
8. Ahora creemos una tabla con schema para consultar los datos de las mediciones de tiempo:

```
CREATE EXTERNAL TABLE IF NOT EXISTS weather (wban INT, date STRING, time STRING,
stationType String, maintenanceIndicator String , skyConditions String,
visibility STRING, weatherType String, dryBulbTemp INT, dewPointTemp INT,
wetBulbTemp INT,  relativeHumidity STRING, windSpeed DOUBLE, windDirection
STRING, windCharGusts STRING, windChar STRING, stationPressure STRING,
pressureTendency INT, seaLevelPressure INT)
COMMENT 'Weather Table on Hive'
        PARTITIONED BY (year INT, month STRING)
        ROW FORMAT DELIMITED
        FIELDS TERMINATED BY ','
        LINES TERMINATED BY '\n'
        location '/hive/data/weather'
        tblproperties ("skip.header.line.count"="1");
```

9. Aprovechando que particionamos por fecha se puede en la inserción indicarle a HIVE a que partición corresponde la data:

```
LOAD DATA INPATH 'hdfs:/datasets/datasets-med/temp-hourly/199607hourly.txt' INTO
TABLE weather PARTITION(year=1996, month='07');
```

Revisar la estructura de directorios con la cual se crean los registros de la tabla. Esto hace muy eficiente una consulta del tipo:

```
select * from weather where month='07'
```

10. Insertar los meses 08 y 09 y realizar la consulta del máximo de la temperatura para cada mes (la temperatura la da el campo **drybulbtemp**).

11. Intenta mostrar el medidor el mes la temperatura en seco junto al máximo para el mes y la temperatura en seco junto al ranking también por mes de la medición.

```
select
   wban,
   month,
   drybulbtemp,
   max(drybulbtemp) OVER (PARTITION  BY month  ORDER BY drybulbtemp ),
   wetbulbtemp,
   rank() OVER (PARTITION  BY month  ORDER BY wetbulbtemp )
from
   weather
where
   month in (
      '08', '09'
   );
```

12. [Opcional si se hizo el ejercicio de flume de bajar tweest]]Ir a HUE, a la consola de HIVE y crear la tabla de tweets:

```
ADD JAR /home/cloudera/Downloads/hive-serdes-1.0-SNAPSHOT.jar;


CREATE EXTERNAL TABLE tweets (
           id BIGINT,
           created_at STRING,
           source STRING,
           favorited BOOLEAN,
           retweeted_status STRUCT<
             text:STRING,
             user:STRUCT<screen_name:STRING,name:STRING>,
             retweet_count:INT>,
           entities STRUCT<
             urls:ARRAY<STRUCT<expanded_url:STRING>>,
             user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>,
             hashtags:ARRAY<STRUCT<text:STRING>>>,
           text STRING,
           user STRUCT<
```

```
            screen_name:STRING,
            name:STRING,
            friends_count:INT,
            followers_count:INT,
            statuses_count:INT,
            verified:BOOLEAN,
            utc_offset:INT,
            time_zone:STRING>,
          in_reply_to_screen_name STRING
           )
           PARTITIONED BY (datehour INT)
           ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe'
           LOCATION '/user/flume/tweets';


-- En el siguiente statement cambiar la fecha de acuerdo a corresponda.
ALTER TABLE tweets ADD IF NOT EXISTS PARTITION (datehour = 2017050719)
LOCATION '/user/flume/tweets/2017/05/07/19';
```