Ejercicios de PIG

Parte 1

Copiar en local file sistema de la MV el fichero datos_pig.txt en la ruta /home/cloudera/ejercicios/pig y abrir el fichero para revisar su contenido.

Copiamos el archivo…

```
[cloudera@quickstart ~]$ cp ~/Desktop/PIG/datos_pig.txt /home/cloudera/ejercicio
s/pig/
```

Comprobamos que esta el contenido, mediante cat datos_pig.txt

```
media   C4      05/31/2013      23:59:53        audioexpert.example.com 0       1
06      NETHERLANDS     BOTTOM
holiday C2      05/31/2013      23:59:54        salestiger.example.com  0       1
20      USA     TOP
```

1. Arranca el Shell de Pig en modo local.

Arrancamos en modo local:

```
[cloudera@quickstart ~]$ pig -x local
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell)
.
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more in
fo.
```

```
2024-07-15 08:16:53,284 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt>
```

grunt indica que estamos en la shell de pig y podemos ejecutar comandos de pig latin.

La ventaja del modo local es que hadoop NO está en funcionamiento, por lo que es ideal para pruebas y desarrollo en máquina unica.

2. Carga los datos en pig en una variable llamada “data”. Los nombres de las columnas deben ser (key, campana, fecha, tiempo, display, accion, cpc, pais, lugar). Los tipos de las columnas deben ser chararray excepto acction y cpc que son int.

```
grunt> data = LOAD '/home/cloudera/ejercicios/pig/datos_pig.txt'
>>        USING PigStorage(',')
>>        AS (key: chararray, campana: chararray, fecha: chararray, tiempo: char
array, display: chararray, accion: int, cpc: int, pais: chararray, lugar: charar
ray);
```

3. Usa el comando DESCRIBE para ver el esquema de la variable “data”.

```
grunt> DESCRIBE data;
data: {key: chararray,campana: chararray,fecha: chararray,tiempo: chararray,displ
ay: chararray,accion: int,cpc: int,pais: chararray,lugar: chararray}
```

4. Selecciona las filas de “data” que provengan de USA.

```
grunt> usa_data = FILTER data by pais == 'USA';
grunt> DUMP usa_data;
2024-07-15 08:29:19,207 [main] INFO  org.apache.pig.tools.pigstats.ScriptState -
Pig features used in the script: FILTER
2024-07-15 08:29:19,243 [main] INFO  org.apache.pig.newplan.logical.optimizer.Log
icalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEa
chColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimize
r, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer,
PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter], RULES
_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
```

5. Listar los datos que contengan en su key el sufijo surf.

```
grunt> surf_data = FILTER data BY key MATCHES'.*surf$';
grunt> DUMP surf_data;
2024-07-15 08:30:45,952 [main] INFO  org.apache.pig.tools.pigstats.ScriptState -
Pig features used in the script: FILTER
2024-07-15 08:30:45,953 [main] INFO  org.apache.pig.newplan.logical.optimizer.Log
icalPlanOptimizer - {RULES ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEa
```

6. Crear una variable llamada “ordenado” que contenga las columnas de data en el siguiente orden: (campaña, fecha, tiempo, key, display, lugar, accion, cpc).

```
grunt> ordenado = FOREACH data GENERATE campana, fecha, tiempo, key, display, lug
ar, accion, cpc;
```

7. Guarda el contenido de la variable “ordenado” en una carpeta en el local file system de tu MV llamada resultado en la ruta /home/cloudera/ejercicios/pig

```
grunt> STORE ordenado INTO '/home/cloudera/ejercicios/pig/resultado' USING PigSto
rage(',');
2024-07-15 08:33:56,557 [main] INFO  org.apache.pig.tools.pigstats.ScriptState -
Pig features used in the script: UNKNOWN
```

8. Comprobar el contenido de la carpeta

```
[cloudera@quickstart ~]$ ls /home/cloudera/ejercicios/pig/resultado/
part-m-00000  _SUCCESS
[cloudera@quickstart ~]$ cat /home/cloudera/ejercicios/pig/resultado/part-m-0000
0
```

Nos da los resultados bien:

```
120     USA     SIDE,,,,
,,,train        A5      05/31/2013      23:59:24        bitpress.example.com   0
120     USA     TOP,,,,
,,,NEWS D5      05/31/2013      23:59:27        datasnap.example.com    0      1
02      USA     SIDE,,,,
,,,bargain      A2      05/31/2013      23:59:32        masterbaker.example.com0
110     USA     BOTTOM,,,,
,,,travel       B5      05/31/2013      23:59:45        pcexpert.example.com   0
116     USA     INLINE,,,,
,,,small        B1      05/31/2013      23:59:48        dealmonkey.example.com 0
120     NETHERLANDS     TOP,,,,
,,,small        D8      05/31/2013      23:59:52        burritofinder.example.co
m       0       118     USA     INLINE,,,,
,,,media        C4      05/31/2013      23:59:53        audioexpert.example.com0
106     NETHERLANDS     BOTTOM,,,,
,,,holiday      C2      05/31/2013      23:59:54        salestiger.example.com 0
120     USA     TOP,,,,
```

Parte 2

Usa el archivo “estudiantes”, comprueba el formato que tienen los datos dentro de este archivo.

```
[cloudera@quickstart ~]$ cat ~/Desktop/PIG/estudiantes.txt
001,Ana,Rojas,674625333,Madrid,Fisica      ,10
002,Marcos,Sanchez,654323442,Sevilla,Matematicas      ,9
003,Rajesh,Khanna,689653222,Valencia,Fisica      ,7
004,Juan,Agarwa,654789888,Barcelona,Biologia      ,8
005,Pedro,Martinez,687987444,Alicante,Derecho      ,7
006,Patricia,Lopez,677888221,Malaga,Informatica      ,9
007,Lucia,Mery,677855221,Malaga,Informatica      ,7
008,Belen,BUrgos,674645333,Madrid,Fisica      ,10
009,Vicente,Babacar,644323442,Sevilla,Matematicas      ,9
010,Krull,Khanfa,689654222,Valencia,Fisica      ,7
011,Pepe,Gomez,654749888,Barcelona,Biologia      ,8
012,Teresa,Ramirez,647987444,Alicante,Derecho      ,7
013,Lucas,Fishra,677848221,Malaga,Informatica      ,9
014,Sara,Mahra,677855224,Malaga,Informatica      ,7
```

1. Realiza la carga en la variable estudiantes teniendo en cuenta el formato de los datos del fichero.

Conecto a pig y cargo los datos en la variable estudiantes:

```
grunt> estudiantes = LOAD '/home/cloudera/Desktop/PIG/estudiantes.txt'
>>                USING PigStorage(',')
>>                AS (id: chararray, nombre: chararray, apellido: chararray, tel
efono: chararray, ciudad: chararray, grado: chararray, nota: int);
```

2. Lista los datos con el nombre y apellido en una misma columna y todas las columnas que consideres sin espacios en blanco.

```
grunt> estudiantes_format = FOREACH estudiantes GENERATE
>>                      CONCAT(nombre, ' ', apellido) as nombre_completo,
>>                      TRIM(telefono) as telefono,
>>                      TRIM(ciudad) as ciudad,
>>                      TRIM(grado) as grado,
>>                      nota;
```

3. Guarda el resultado en un nuevo archivo con ‘:’ como delimitador.

```
grunt> STORE estudiantes_format INTO '/home/cloudera/ejercicios/pig/estudiantes_
format' USING PigStorage(':');
```

4. Lista los datos que sean de Madrid.

```
grunt> madrid_estudiantes = FILTER estudiantes_format BY ciudad == 'Madrid';
DUMP madrid_estudiantes;
```

```
ne.util.MapRedUtil - Total input paths to process : 1
(Ana Rojas,674625333,Madrid,Fisica,10)
(Belen BUrgos,674645333,Madrid,Fisica,10)
```

5. Muestra la media agrupada por grado.

```
grunt> grado_media = GROUP estudiantes_format BY grado;
```

```
grunt> grado_avg = FOREACH grado_media GENERATE group AS grado, AVG(estudiantes_
format.nota) AS media_nota;
```

```
grunt> DUMP grado_avg;
```

```
ne.util.MapRedUtil - Total input paths to process : 1
(Fisica,8.5)
(Derecho,7.0)
(Biologia,8.0)
(Informatica,8.0)
(Matematicas,9.0)
```

6. Muestra el número de estudiantes de cada ciudad.

```
grunt> ciudad_estudiantes = GROUP estudiantes_format BY ciudad;
ciudad_count = FOREACH ciudad_estudiantes GENERATE group AS ciudad, COUNT(estudi
antes_format) AS num_estudiantes;
DUMP ciudad_count;
```

```
2024-07-15 09:02:06,497 [main] INFO  org.apache.pig.ba
ne.util.MapRedUtil - Total input paths to process : 1
(Madrid,2)
(Malaga,4)
(Sevilla,2)
(Alicante,2)
(Valencia,2)
(Barcelona,2)
```

7. El nombre del alumno o alumnos de cada ciudad con mayor nota media.

```
grunt> ciudad_grupo = GROUP estudiantes_format BY ciudad;
max_nota = FOREACH ciudad_grupo {
    ordenado_por_nota = ORDER estudiantes_format BY nota DESC;
    primero = LIMIT ordenado_por_nota 1;
    GENERATE group AS ciudad, FLATTEN(primero.nombre_completo) AS nombre_complet
o, MAX(ordenado_por_nota.nota) AS nota;
};
DUMP max nota;
```

```
ne.util.MapRedUtil - Total input paths to process : 1
(Alicante,Teresa Ramirez,7)
(Barcelona,Pepe Gomez,8)
(Madrid,Belen BUrgos,10)
(Malaga,Patricia Lopez,9)
(Sevilla,Marcos Sanchez,9)
(Valencia,Krull Khanfa,7)
```