

DC-02 TP-02 HIVE

DC-02 TRABAJO PRÁCTICO 2 HIVE

Programa: **Máster Executive en Big Data, Cloud & Analytics**

Periodo académico: **2019 – 2020**

Autor/es: **CARLOS ALFONSEL JAÉN**

1. DESCRIPCIÓN:

Cada trimestre se publica por parte del INE los datos de la EPA (Encuesta de Población Activa). En este enlace, podéis encontrar los datos (en diferentes formatos) y la descripción de los campos que se contienen:

https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176918&menu=resultados&secc=1254736030639&idp=1254735976595

Vamos a repasar algunos ejemplos vistos en clase a partir de este dataset. Partiendo de los datos de la encuesta para el 2º Trimestre del 2019, usaremos el formato CSV que podéis encontrar en el ZIP descargado. Entregar en un PDF los pasos, comandos y los pantallazos del proceso realizado contra el clúster de Dataproc de Google Cloud.

2. SE PIDE:

2.1. Descargar los ficheros de la encuesta (2019 2º Trimestre).

2.2. Subir a HDFS.

De los ficheros descargados, subir el csv a la carpeta `/tmp/data/`

2.3. Crear una external table en Hive a partir de este archivo.

Para los tipos de datos podéis ver la definición de los campos en la Excel que podéis ver en el enlace del INE. Los campos serán de tipo String o Int. Comprobar que tenemos la tabla creada en Hive.

2.4. Hacer alguna consulta sencilla:

- Obtener la descripción detallada de la tabla `epa` (describe formatted).
- Obtener el número total de registros que tiene la encuesta
(https://www.w3schools.com/sql/sql_count_avg_sum.asp).
- Obtener el número total de encuestados por ccaa
(https://www.w3schools.com/sql/sql_groupby.asp).

2.5. Para subir nota:

En clase hemos introducido el concepto de particiones y buckets. También hemos visto en la práctica un ejemplo de crear particiones. Se pide buscar información y ejemplos para entender los tipos de particiones (dinámicas y estáticas) (<http://www.hadoopinsight.com/blog/hive/hive-static-vs-dynamic-partition/>). Una vez entendido, se pide:

- a) A partir de la tabla *epa* creada, se pide además crear una tabla particionada dinámicamente *epa_particionada* y repetir las consultas vistas en el punto 4.
- b) Si quisiéramos tener una gran tabla en Hive con todas las EPAs por año, cómo podríamos hacer esa carga de manera eficiente: con particiones dinámicas o estáticas. Justificar la respuesta.

3. SOLUCIÓN:

En primer lugar, creamos el clúster en Google Cloud Platform:

```
> gcloud beta dataproc clusters create dc02-tp02-cluster --enable-component-gateway --region Europe-west3 --subnet default --zone Europe-west3-a --master-machine-type n1-standard-2 --master-boot-disk-size 50 --num-workers 3 --worker-machine-type n1-standard-2 --worker-boot-disk-size 50 --image-version 1.3-deb9 --project m01-dc02-hadoop
```

Accedemos vía SSH a la consola del nodo maestro del clúster y subimos el archivo CSV descargado con la EPA del segundo trimestre de 2019 (ver adjunto *EPA_2019T2.rar*), al que previamente hemos quitado la primera línea que contiene las cabeceras de los campos (**Figura 01**).

A continuación, subimos el archivo a la carpeta */tmp/data* del HDFS del clúster:

```
> hadoop fs -put EPA_2019T2.csv /tmp/data/epa_2019t2.csv
```

Comprobamos que el archivo se ha alojado correctamente en nuestro sistema de almacenamiento de ficheros (**Figura 02**):

```
> hadoop fs -ls /tmp/data
```

Desde la consola se arranca hive y se crea una tabla externa a partir del archivo subido a HDFS (**Figura 03**):

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS epa (  
  > ciclo STRING,  
  > ccaa STRING,  
  > prov STRING,  
  > nivel STRING,  
  > edad5 STRING,  
  > relpp1 STRING,  
  > sexo1 STRING,  
  > ncony STRING,  
  > npadre STRING,
```

```
> nmadre STRING,
> eciv1 STRING,
> nac1 STRING,
> nforma STRING,
> ocup1 STRING,
> act1 STRING,
> situ STRING
> )

> COMMENT 'Tabla EPA Segundo Trimestre 2019'
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY '\t'
> LINES TERMINATED BY '\n'
> STORED AS TEXTFILE
> LOCATION '/tmp/data';
```

cajaenh@dc02-tp02-cluster-m: ~ - Mozilla Firefox

<https://ssh.cloud.google.com/projects/m01-dc02-hadoop/zones/europe-west3-ε>

Connected, host fingerprint: ssh-rsa 0 23:27:A5:F5:BA:44:29:86:04:08:F6:BD:71:CF:82:A9:64:60:C6:A9:56:BC:52:FD:0A:9B:5C:1E:2C:C3:7A:0C
Linux dc02-tp02-cluster-m 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3+deb9u1 (2019-09-20) x86_64

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

cajaenh@dc02-tp02-cluster-m:~\$

Transferencia de archivos

Close

EPA_2019T2.csv

Finalizada

Destino de los archivos subidos: /home/cajaenh

Figura 1

```

cajaenh@dc02-tp02-cluster-m: ~ - Mozilla Firefox
https://ssh.cloud.google.com/projects/m01-dc02-hadoop/zones/europe-west3-...

Connected, host fingerprint: ssh-rsa 0 23:27:A5:F5:BA:44:29:86:04:08:F6:BD:71:CF
:82:A9:64:60:C6:A9:56:BC:52:FD:0A:9B:5C:1E:2C:C3:7A:0C
Linux dc02-tp02-cluster-m 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3+deb9u1 (2019-09
-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
cajaenh@dc02-tp02-cluster-m:~$ hadoop fs -ls /tmp
Found 2 items
drwxrwxrwt - hdfs hadoop 0 2019-12-09 16:20 /tmp/hadoop-yarn
drwx-wx-wx - hive hadoop 0 2019-12-09 16:21 /tmp/hive
cajaenh@dc02-tp02-cluster-m:~$ hadoop fs -mkdir /tmp/data
cajaenh@dc02-tp02-cluster-m:~$ hadoop fs -put epa_2019t2.csv /tmp/data/epa_2019t2.csv
put: `epa_2019t2.csv': No such file or directory
cajaenh@dc02-tp02-cluster-m:~$ ls -la
total 70200
drwxr-xr-x 3 cajaenh cajaenh 4096 Dec 9 16:37 .
drwxr-xr-x 3 root root 4096 Dec 9 16:25 ..
-rw-r--r-- 1 cajaenh cajaenh 220 May 15 2017 .bash_logout
-rw-r--r-- 1 cajaenh cajaenh 3526 May 15 2017 .bashrc
-rw-r--r-- 1 cajaenh cajaenh 71852496 Dec 9 16:37 EPA_2019T2.csv
-rw-r--r-- 1 cajaenh cajaenh 675 May 15 2017 .profile
drwx----- 2 cajaenh cajaenh 4096 Dec 9 16:28 .ssh
cajaenh@dc02-tp02-cluster-m:~$ hadoop fs -put EPA_2019T2.csv /tmp/data/epa_2019t2.csv
cajaenh@dc02-tp02-cluster-m:~$ hadoop fs -ls /tmp/data
Found 1 items
-rw-r--r-- 2 cajaenh hadoop 71852496 2019-12-09 16:44 /tmp/data/epa_2019t2.csv
cajaenh@dc02-tp02-cluster-m:~$

```

Figura 2

La tabla externa *epa*, por sencillez, se ha creado únicamente con 16 campos (columnas) en vez de los 93 que existen según la definición que puede consultarse en el archivo adjunto *dr_EPA_2005.xlsx*: Período de Referencia (ciclo), Comunidad Autónoma (ccaa), Provincia (prov), Nivel del Registro (nivel), Edad (edad5), Relación con la Persona de Referencia (relpp1), Sexo (sexo1), Número de Orden del Cónyuge (ncony), Número de Orden del Padre (npadre), Número de Orden de la Madre (nmadre), Estado Civil (eciv1), Nacionalidad (nac1), Nivel de Estudios (nforma), Ocupación (ocup1), Actividad Profesional (act1) y Situación Laboral (situ).

```

cajaenh@dc02-tp02-cluster-m: ~ - Mozilla Firefox
https://ssh.cloud.google.com/projects/m01-dc02-hadoop/zones/europe-west3-...

-rw-r--r-- 1 cajaenh cajaenh 71852496 Dec  9 16:37 EPA_2019T2.csv
-rw-r--r-- 1 cajaenh cajaenh  675 May 15 2017 .profile
drwx----- 2 cajaenh cajaenh  4096 Dec  9 16:28 .ssh
cajaenh@dc02-tp02-cluster-m:~$ hadoop fs -put EPA_2019T2.csv /tmp/data/epa_2019t2.csv

cajaenh@dc02-tp02-cluster-m:~$ hadoop fs -ls /tmp/data
Found 1 items
-rw-r--r-- 2 cajaenh hadoop  71852496 2019-12-09 16:44 /tmp/data/epa_2019t2.csv
cajaenh@dc02-tp02-cluster-m:~$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
hive> CREATE EXTERNAL TABLE IF NOT EXISTS epa (
  > ciclo STRING,
  > ccaa STRING,
  > prov STRING,
  > nivel STRING,
  > edad5 STRING,
  > relppi STRING,
  > sexo1 STRING,
  > ncony STRING,
  > npadre STRING,
  > nmadre STRING,
  > eciv1 STRING,
  > nacl STRING,
  > nforma STRING,
  > ocup1 STRING,
  > act1 STRING,
  > situ STRING
  > )
  > COMMENT 'Tabla EPA Segundo Trimestre 2019'
  > ROW FORMAT DELIMITED
  > FIELDS TERMINATED BY '\t'
  > LINES TERMINATED BY '\n'
  > STORED AS TEXTFILE
  > LOCATION '/tmp/data';
OK
Time taken: 1.116 seconds
hive> show tables;
OK
epa
Time taken: 0.115 seconds, Fetched: 1 row(s)
hive>
  
```

Figura 3

A continuación, hacemos las consultas pedidas en el enunciado sobre la tabla creada:

- Para obtener una descripción detallada de la tabla (**Figura 04**):

```
hive> describe formatted epa;
```



```

cajaenh@dc02-tp02-cluster-m: ~ - Mozilla Firefox
https://ssh.cloud.google.com/projects/m01-dc02-hadoop/zones/europe-west3-ε
nivel          string
edad5          string
relpp1         string
sexo1          string
ncony          string
npadre         string
nmadre         string
eciv1          string
nac1           string
nforma         string
ocup1          string
act1           string
situ           string

# Detailed Table Information
Database:      default
Owner:         cajaenh
CreateTime:    Mon Dec 09 16:53:30 UTC 2019
LastAccessTime: UNKNOWN
Retention:     0
Location:      hdfs://dc02-tp02-cluster-m/tmp/data
Table Type:    EXTERNAL_TABLE
Table Parameters:
  EXTERNAL      TRUE
  comment       Tabla EPA Segundo Trimestre 2019
  numFiles      1
  totalSize     71852496
  transient_lastDdlTime 1575910410

# Storage Information
SerDe Library: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:   org.apache.hadoop.mapred.TextInputFormat
OutputFormat:  org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:    No
Num Buckets:   -1
Bucket Columns: []
Sort Columns:  []
Storage Desc Params:
  field.delim      \t
  line.delim       \n
  serialization.format \t
Time taken: 0.186 seconds, Fetched: 46 row(s)
hive>

```

Figura 4

- Para obtener el número total de registros de la encuesta (**Figura 05**):

```
hive> select count(ciclo) from epa;
```

Puede observarse en el pantallazo adjunto que el número de registros de la encuesta es **164.764**.

```

cajaenh@dc02-tp02-cluster-m: ~ - Mozilla Firefox
https://ssh.cloud.google.com/projects/m01-dc02-hadoop/zones/europe-west3-ε ...
field.delim      \t
line.delim       \n
serialization.format \t
Time taken: 0.186 seconds, Fetched: 46 row(s)
hive> select count(ciclo) from epa;
Query ID = cajaenh_20191209165508_5e94add5-852c-4a29-be3e-696ceb01242a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1575908379445_0001)
)
OK
164764
Time taken: 21.318 seconds, Fetched: 1 row(s)
hive> select ccaa, count(ciclo) from epa group by ccaa;
Query ID = cajaenh_20191209165605_b7b2ade9-b6d7-49a4-b7fc-bf6989c1775b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1575908379445_0001)
)
OK
"01"      26873
"02"      7173
"03"      4747
"04"      4177
"05"      6744
"06"      3732
"07"     15479
"08"     11277
"09"     16663
"10"     12364
"11"      5561
"12"     19509
"13"      9271
"14"      5336
"15"      3848
"16"      7758
"17"      2832
"51"        693
"52"        727
Time taken: 12.504 seconds, Fetched: 19 row(s)
hive>

```

Figura 5

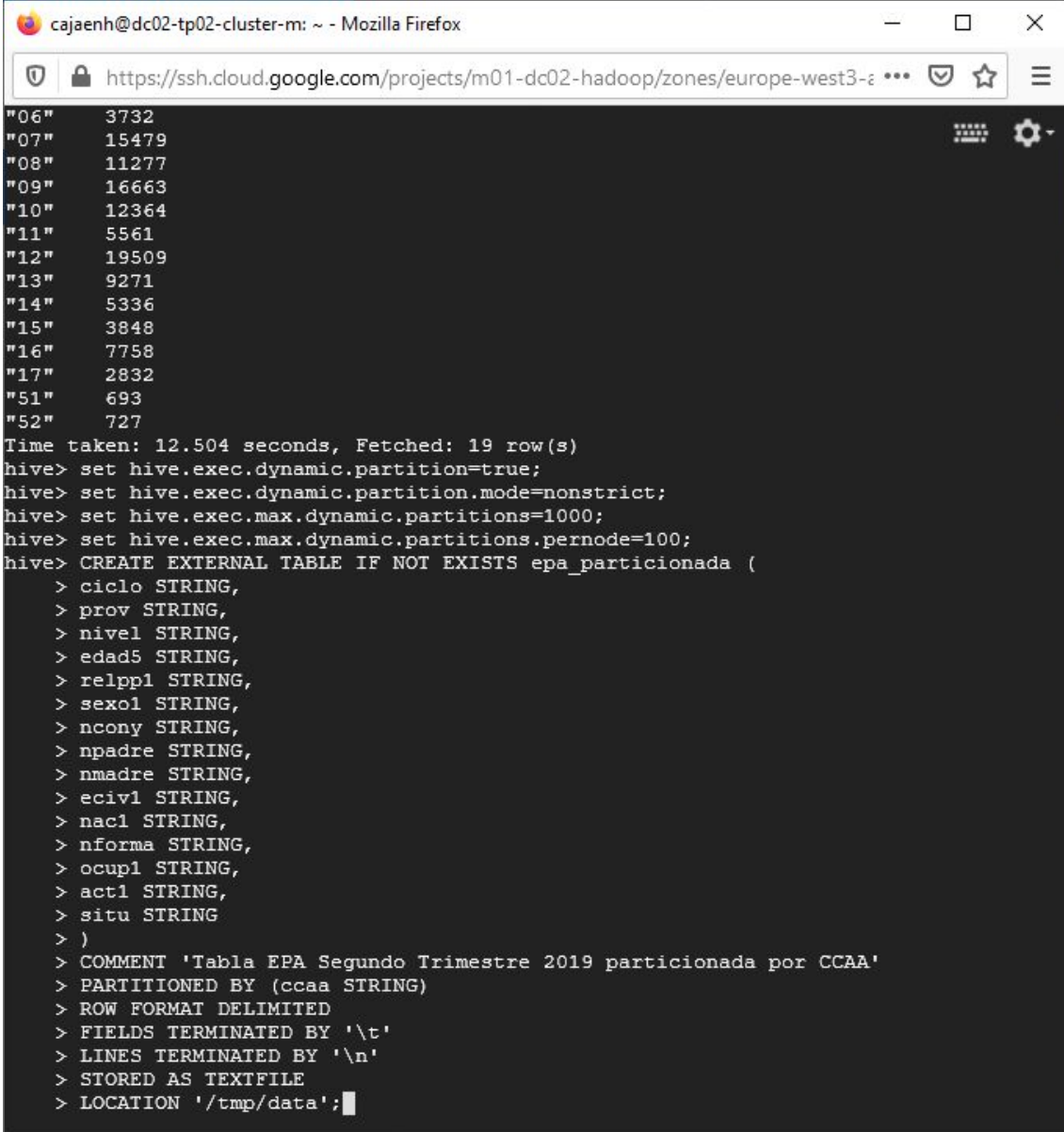
- Para obtener el número total de encuestados por Comunidad Autónoma (Figura 05):

```
hive> select ccaa, count(ciclo) from epa group by ccaa;
```

Es importante señalar que se ha comprobado que la suma de los encuestados por CC.AA. coincide con el número total de registros de la tabla (164.764). además, también se puede comprobar que los resultados son 19: 17 por cada una de las CC.AA. y dos más por Ceuta y Melilla (campos 51 y 52).

A continuación, se va a crear una tabla particionada dinámicamente por Comunidad Autónoma y llamada *epa_particionada* (Figura 06):


```
hive> set hive.exec.dynamic.partition=true;
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> set hive.exec.max.dynamic.partitions=1000;
hive> set hive.exec.max.dynamic.partitions.pernode=100;
```



```
cajaenh@dc02-tp02-cluster-m: ~ - Mozilla Firefox
https://ssh.cloud.google.com/projects/m01-dc02-hadoop/zones/europe-west3-ε ...
"06" 3732
"07" 15479
"08" 11277
"09" 16663
"10" 12364
"11" 5561
"12" 19509
"13" 9271
"14" 5336
"15" 3848
"16" 7758
"17" 2832
"51" 693
"52" 727
Time taken: 12.504 seconds, Fetched: 19 row(s)
hive> set hive.exec.dynamic.partition=true;
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> set hive.exec.max.dynamic.partitions=1000;
hive> set hive.exec.max.dynamic.partitions.pernode=100;
hive> CREATE EXTERNAL TABLE IF NOT EXISTS epa_particionada (
  > ciclo STRING,
  > prov STRING,
  > nivel STRING,
  > edad5 STRING,
  > relpp1 STRING,
  > sexo1 STRING,
  > ncony STRING,
  > npadre STRING,
  > nmadre STRING,
  > eciv1 STRING,
  > nac1 STRING,
  > nforma STRING,
  > ocup1 STRING,
  > act1 STRING,
  > situ STRING
  > )
  > COMMENT 'Tabla EPA Segundo Trimestre 2019 particionada por CCAA'
  > PARTITIONED BY (ccaa STRING)
  > ROW FORMAT DELIMITED
  > FIELDS TERMINATED BY '\t'
  > LINES TERMINATED BY '\n'
  > STORED AS TEXTFILE
  > LOCATION '/tmp/data';
```

Figura 6

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS epa_particionada (
  > ciclo STRING,
  > prov STRING,
  > nivel STRING,
  > edad5 STRING,
  > relpp1 STRING,
  > sexo1 STRING,
  > ncony STRING,
```

```
> npadre STRING,
> nmadre STRING,
> eciv1 STRING,
> nac1 STRING,
> nforma STRING,
> ocup1 STRING,
> act1 STRING,
> situ STRING
> )
> COMMENT 'Tabla EPA Segundo Trimestre 2019 particionada por CCAA'
> PARTITIONED BY (ccaa STRING)
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY '\t'
> LINES TERMINATED BY '\n'
> STORED AS TEXTFILE
> LOCATION '/tmp/data';
```

Para rellenar la nueva tabla particionada dinámicamente:

```
hive> INSERT INTO TABLE epa_particionada PARTITION(ccaa)
> SELECT
ciclo,ccaa,prov,nivel,edad5,relpp1,sexo1,ncony,npadre,nmadre,eciv1,nac
1,nforma,ocup1,act1,situ FROM epa;
```

Para ver las particiones que se han creado (**Figura 7**):

```
hive> SHOW PARTITIONS epa_particionada:
```

Y podemos hacer las mismas consultas vistas en el punto anterior (**Figura 8**):

```
hive> describe formatted epa_particionada;
hive> select count(ciclo) from epa_particionada;
hive> select ccaa, count(ciclo) from epa_particionada group by ccaa;
```

Es importante indicar que el número de registros que se obtienen con la segunda consulta sobre la tabla particionada dinámicamente sigue siendo el mismo (164.764), como puede apreciarse en los pantallazos adjuntos.

```

cajaenh@dc02-tp02-cluster-m: ~ - Mozilla Firefox
https://ssh.cloud.google.com/projects/m01-dc02-hadoop/zones/europe-west3-ε ...
"187" "16" "01" "00012" "1" "01" "45" "1" "1" "02" "00"
00" " " "1" "20" " " " " " " " " " "
Time taken: 0.268 seconds, Fetched: 25 row(s)
hive> select * from epa_particionada limit 10;
OK
"187" "16" "01" "00001" "1" "01" "35" "1" "1" "02" "00"
00" " " "1" "01" " " " " " " " " " "
"187" "16" "01" "00001" "1" "02" "35" "2" "6" "01" "00"
00" " " "1" "01" " " " " " " " " " "
"187" "16" "01" "00001" "2" "03" "00" "3" "6" "00" "01"
02" " " " " "01" " " " " " " " " "
"187" "16" "01" "00002" "1" "01" "45" "1" "6" "00" "00"
00" " " "4" "01" " " " " " " " " "
"187" "16" "01" "00002" "1" "02" "20" "3" "1" "00" "00"
01" " " "1" "01" " " " " " " " " "
"187" "16" "01" "00002" "1" "03" "16" "3" "6" "00" "00"
01" " " "1" "01" " " " " " " " " "
"187" "16" "01" "00003" "1" "01" "45" "1" "1" "02" "00"
00" " " "2" "01" " " " " " " " " "
"187" "16" "01" "00003" "2" "03" "10" "3" "1" "00" "01"
02" " " " " "01" " " " " " " " "
"187" "16" "01" "00003" "2" "04" "05" "3" "1" "00" "01"
02" " " " " "01" " " " " " " " "
"187" "16" "01" "00004" "1" "01" "45" "1" "1" "02" "00"
00" " " "1" "48" " " " " " " " "
Time taken: 0.277 seconds, Fetched: 10 row(s)
hive> SHOW PARTITIONS epa_particionada;
OK
ccaa=22 22
ccaa=2210022
ccaa=2211522
ccaa=2212522
ccaa=2212822
ccaa=2220022
ccaa=2230022
ccaa=2231022
ccaa=2235022
ccaa=2240022
ccaa=2241022
ccaa=2242022
ccaa=2250022
Time taken: 0.116 seconds, Fetched: 13 row(s)
hive>

```

Figura 7

```

cajaenh@dc02-tp02-cluster-m: ~ - Mozilla Firefox
https://ssh.cloud.google.com/projects/m01-dc02-hadoop/zones/europe-west3-ε ...
Num Buckets: -1
Bucket Columns: []
Sort Columns: []
Storage Desc Params:
    field.delim \t
    line.delim \n
    serialization.format \t
Time taken: 0.143 seconds, Fetched: 54 row(s)
hive> select count(ciclo) from epa_particionada;
Query ID = cajaenh_20191209170529_8303cac0-ec7e-42c3-9290-cb194bac6a98
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1575908379445_0002)

OK
164764
Time taken: 24.497 seconds, Fetched: 1 row(s)
hive> select ccaa, count(ciclo) from epa_particionada group by ccaa;
Query ID = cajaenh_20191209170635_9cc98295-40c4-4448-8e14-ad7500326142
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1575908379445_0002)

OK
" " 150513
"100" 773
"115" 1938
"125" 167
"128" 1659
"200" 2721
"300" 71
"310" 1309
"350" 4870
"400" 300
"410" 36
"420" 395
"500" 12
Time taken: 12.764 seconds, Fetched: 13 row(s)
hive>

```

Figura 8

Por último, a la pregunta de qué sería más conveniente desde un punto de vista de eficiencia, si particiones dinámicas o estáticas para una carga masiva en Hive de todas las Encuestas de Población Activa ordenadas por año, la respuesta es **DINÁMICAS**. La razón es que, en caso contrario, usando particiones estáticas, habría que ejecutar N cargas, una por cada trimestre existente, en vez de hacer una única carga particionada por Ciclo de Referencia.

Volviendo al ejemplo desarrollado en el apartado anterior, si la tabla se hubiera creado de forma estática, habría que haber ejecutado el comando INSERT tantas veces como valores distintos tuviera el campo por el quisiéramos particionar: en nuestro ejemplo, en el que hemos creado la tabla particionada por CC.AA., habría que haberlo ejecutado 19 veces.

Partición dinámica:

```
hive> INSERT INTO TABLE epa_particionada PARTITION(ccaa)  
> SELECT  
ciclo,ccaa,prov,nivel,edad5,relpp1,sexo1,ncony,npadre,nmadre,eciv1,nac  
1,nforma,ocup1,act1,situ FROM epa;
```

Partición estática:

```
hive> INSERT INTO TABLE epa_particionada PARTITION(ccaa='01')  
> SELECT  
ciclo,ccaa,prov,nivel,edad5,relpp1,sexo1,ncony,npadre,nmadre,eciv1,nac  
1,nforma,ocup1,act1,situ  
> FROM epa WHERE ccaa='01';
```

```
hive> INSERT INTO TABLE epa_particionada PARTITION(ccaa='02')  
> SELECT  
ciclo,ccaa,prov,nivel,edad5,relpp1,sexo1,ncony,npadre,nmadre,eciv1,nac  
1,nforma,ocup1,act1,situ  
> FROM epa WHERE ccaa='02';
```

...

```
hive> INSERT INTO TABLE epa_particionada PARTITION(ccaa='17')  
> SELECT  
ciclo,ccaa,prov,nivel,edad5,relpp1,sexo1,ncony,npadre,nmadre,eciv1,nac  
1,nforma,ocup1,act1,situ  
> FROM epa WHERE ccaa='17';
```

```
hive> INSERT INTO TABLE epa_particionada PARTITION(ccaa='51')  
> SELECT  
ciclo,ccaa,prov,nivel,edad5,relpp1,sexo1,ncony,npadre,nmadre,eciv1,nac  
1,nforma,ocup1,act1,situ  
> FROM epa WHERE ccaa='51';
```

```
hive> INSERT INTO TABLE epa_particionada PARTITION(ccaa='52')  
> SELECT  
ciclo,ccaa,prov,nivel,edad5,relpp1,sexo1,ncony,npadre,nmadre,eciv1,nac  
1,nforma,ocup1,act1,situ  
> FROM epa WHERE ccaa='52';
```