M03-DC02 HBase

EJERCICIO EVALUABLE HBase

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

Periodo académico: **2019 – 2020**

Autor/es: **CARLOS ALFONSEL JAÉN**

1. ENUNCIADO

Se requiere la entrega de un documento PDF con las soluciones y comentarios propuestos de cada ejercicio.

1. Levanta la máquina virtual de **Cloudera Quickstart**.

2. Ejecuta en la shell de **HBase** una consulta que cree una tabla llamada "sqoopTest" con una columnFamily llamada "cfTest".

3. Importa la base de datos "customers" de mysql incluida dentro de la máquina virtual ayudándote del comando "sqoop-import" a la tabla creada en HBase en el ejercicio anterior con su columnFamily a través del campo "customer_id" de la tabla de origen. Divídela por el campo customer_id con 8 mappers. Datos necesarios:

- Driver: com.mysql.jdbc.Driver
- Connect: jdbc:mysql://quickstart:3306/retail_db
- Username: retail_dba
- Password: cloudera

4. Haz un scan de la tabla. ¿Cuánto tiempo tarda en ejecutar ese comando?.

5. Realiza otro "scan" de la tabla limitando el resultado a 3 rows y después otro scan filtrando por una row key de comienzo, 4000, y otra row key de final, 4003. ¿Cuántas filas salen?. Detecta los posibles "customer_id" mapeados a row_key de la tabla en HBase. ¿Qué observas?. ¿Cuál es la ordenación?.

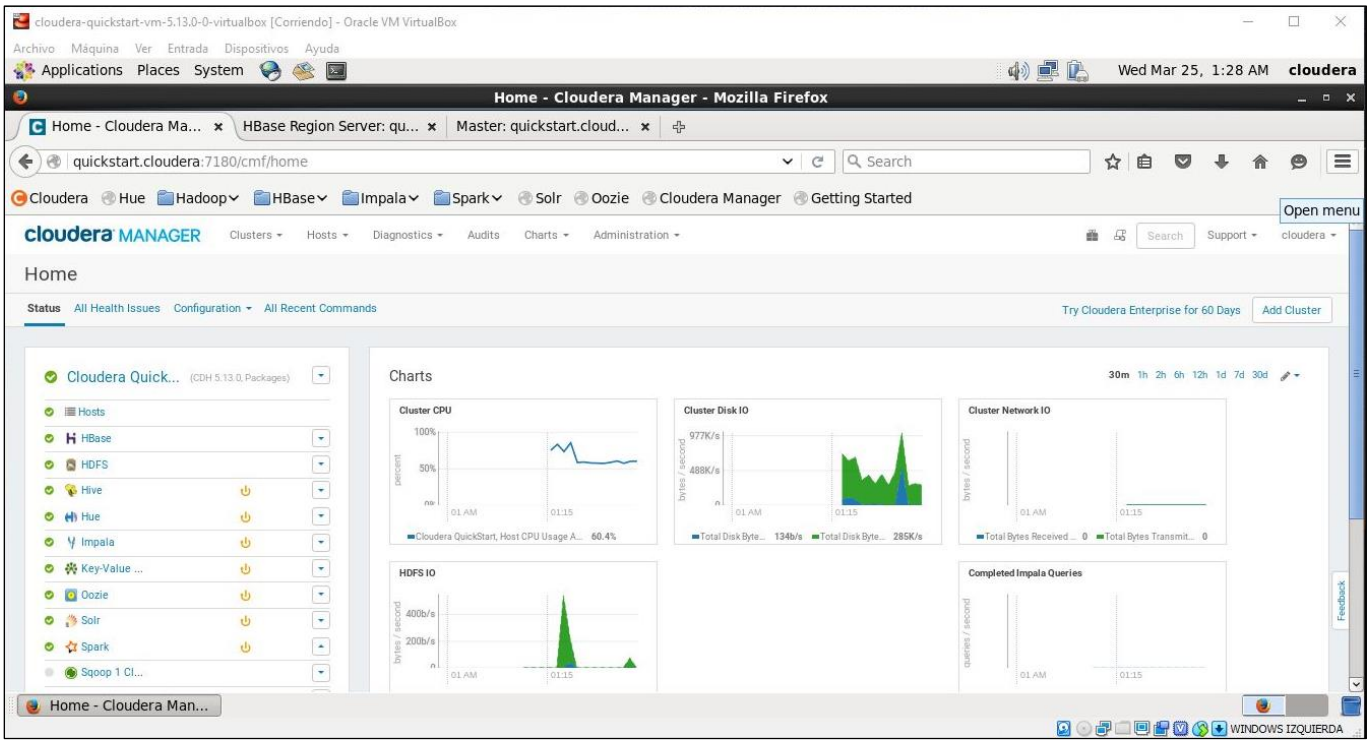
6. Modifica la tabla creada con 5 número de versiones y 1Mb (1048576 bytes) de tamaño de bloque.

7. Deshabilita la tabla "sqoopTest" y realiza un scan sobre la misma tabla. ¿Qué ocurre?.

8. Vuelve a habilitar la tabla y realiza un conteo de filas, ¿cuántas filas hay? Haz una consulta sobre el último row key. ¿Qué columnFamily tiene? ¿Cuántos qualifier? Indica sus valores.

2. SOLUCIONES

2.1. Levanta la Máquina Virtual de Cloudera Quickstart.



2.2. Ejecuta en la shell de HBase una consulta que cree una tabla llamada "sqoopTest" con una columnFamily llamada "cfTest".

```
[cloudera@quickstart ~]$ sudo hbase shell
```

```
hbase(main):008:0> create 'sqoopTest', {NAME => 'cfTest'}
```

```
hbase(main):008:0> create 'sqoopTest', {NAME => 'cfTest'}
0 row(s) in 1.2550 seconds

=> Hbase::Table - sqoopTest
```

```
hbase(main):006:0> describe 'sqoopTest'
```

```
hbase(main):006:0> describe 'sqoopTest'
Table sqoopTest is ENABLED
sqoopTest
COLUMN FAMILIES DESCRIPTION
{NAME => 'cfTest', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', T
TL => 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
1 row(s) in 0.0440 seconds
```

2.3. Importa la base de datos "customers" de mysql incluida dentro de la máquina virtual ayudándote del comando "sqoop-import" a la tabla creada en HBase en el ejercicio anterior con su columnFamily a través del campo "customer_id" de la tabla de origen. Divídela por el campo customer_id con 8 mappers. Datos necesarios:

- **Driver:** com.mysql.jdbc.Driver
- **Connect:** jdbc:mysql://quickstart:3306/retail_db
- **Username:** retail_dba
- **Password:** cloudera

¡IMPORTANTE!: el siguiente comando hay que lanzarlo desde un terminal de Linux, no desde la shell de HBase.

```
[cloudera@quickstart ~]$ sudo sqoop import \
> --connect jdbc:mysql://quickstart:3306/retail_db \
> --driver com.mysql.jdbc.Driver \
> --username retail_dba \
> --password cloudera \
> --table customers \
> --hbase-table sqoopTest \
> --hbase-row-key customer_id \
> --column-family cfTest \
> --split-by customer_id -m 8
```

```
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=8
                Other local map tasks=8
                Total time spent by all maps in occupied slots (ms)=247775232
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=483936
                Total vcore-milliseconds taken by all map tasks=483936
                Total megabyte-milliseconds taken by all map tasks=247775232
        Map-Reduce Framework
                Map input records=12435
                Map output records=12435
                Input split bytes=945
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=63532
                CPU time spent (ms)=83640
                Physical memory (bytes) snapshot=1030418432
                Virtual memory (bytes) snapshot=5931978752
                Total committed heap usage (bytes)=403177472
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
20/03/25 06:03:24 INFO mapreduce.ImportJobBase: Transferred 0 bytes in 276.5825 seconds (0 bytes/sec)
20/03/25 06:03:24 INFO mapreduce.ImportJobBase: Retrieved 12435 records.
[cloudera@quickstart ~]$
```

Como puede apreciarse en la captura anterior, ha tardado 276,5825 segundos en transferir 12.435 registros.

2.4. Haz un scan de la tabla. ¿Cuánto tiempo tarda en ejecutar ese comando?.

```
hbase(main):001:0> scan 'sqoopTest'
```

Según puede observarse en la siguiente captura, el tiempo empleado en escanear la tabla ha sido de 15.638 segundos.

```
9996                                    column=cfTest:customer_street, timestamp=1585141400600, value=3155 Burning Grove
9996                                    column=cfTest:customer_zipcode, timestamp=1585141400600, value=90631
9997                                    column=cfTest:customer_city, timestamp=1585141400600, value=Irving
9997                                    column=cfTest:customer_email, timestamp=1585141400600, value=XXXXXXXXX
9997                                    column=cfTest:customer_fname, timestamp=1585141400600, value=John
9997                                    column=cfTest:customer_lname, timestamp=1585141400600, value=Fuentes
9997                                    column=cfTest:customer_password, timestamp=1585141400600, value=XXXXXXXXX
9997                                    column=cfTest:customer_state, timestamp=1585141400600, value=TX
9997                                    column=cfTest:customer_street, timestamp=1585141400600, value=3200 Sunny Grove Jetty
9997                                    column=cfTest:customer_zipcode, timestamp=1585141400600, value=75061
9998                                    column=cfTest:customer_city, timestamp=1585141400600, value=Caguas
9998                                    column=cfTest:customer_email, timestamp=1585141400600, value=XXXXXXXXX
9998                                    column=cfTest:customer_fname, timestamp=1585141400600, value=David
9998                                    column=cfTest:customer_lname, timestamp=1585141400600, value=Conrad
9998                                    column=cfTest:customer_password, timestamp=1585141400600, value=XXXXXXXXX
9998                                    column=cfTest:customer_state, timestamp=1585141400600, value=PR
9998                                    column=cfTest:customer_street, timestamp=1585141400600, value=4725 Harvest Heights
9998                                    column=cfTest:customer_zipcode, timestamp=1585141400600, value=00725
9999                                    column=cfTest:customer_city, timestamp=1585141400600, value=Albuquerque
9999                                    column=cfTest:customer_email, timestamp=1585141400600, value=XXXXXXXXX
9999                                    column=cfTest:customer_fname, timestamp=1585141400600, value=Susan
9999                                    column=cfTest:customer_lname, timestamp=1585141400600, value=Smith
9999                                    column=cfTest:customer_password, timestamp=1585141400600, value=XXXXXXXXX
9999                                    column=cfTest:customer_state, timestamp=1585141400600, value=NM
9999                                    column=cfTest:customer_street, timestamp=1585141400600, value=5238 Sunny Walk
9999                                    column=cfTest:customer_zipcode, timestamp=1585141400600, value=87112
12435 row(s) in 15.6380 seconds
```

2.5. Realiza otro "scan" de la tabla limitando el resultado a 3 rows y después otro scan filtrando por una row key de comienzo, 4000, y otra row key de final, 4003. ¿Cuántas filas salen?. Detecta los posibles "customer_id" mapeados a row_key de la tabla en HBase. ¿Qué observas?. ¿Cuál es la ordenación?.

```
hbase(main):002:0> scan 'sqoopTest', {LIMIT => 3}
```

```
hbase(main):002:0> scan 'sqoopTest', {LIMIT => 3}
ROW                                   COLUMN+CELL
 1                                    column=cfTest:customer_city, timestamp=1585141337609, value=Brownsville
 1                                    column=cfTest:customer_email, timestamp=1585141337609, value=XXXXXXXXX
 1                                    column=cfTest:customer_fname, timestamp=1585141337609, value=Richard
 1                                    column=cfTest:customer_lname, timestamp=1585141337609, value=Hernandez
 1                                    column=cfTest:customer_password, timestamp=1585141337609, value=XXXXXXXXX
 1                                    column=cfTest:customer_state, timestamp=1585141337609, value=TX
 1                                    column=cfTest:customer_street, timestamp=1585141337609, value=6303 Heather Plaza
 1                                    column=cfTest:customer_zipcode, timestamp=1585141337609, value=78521
 10                                   column=cfTest:customer_city, timestamp=1585141337609, value=Stafford
 10                                   column=cfTest:customer_email, timestamp=1585141337609, value=XXXXXXXXX
 10                                   column=cfTest:customer_fname, timestamp=1585141337609, value=Melissa
 10                                   column=cfTest:customer_lname, timestamp=1585141337609, value=Smith
 10                                   column=cfTest:customer_password, timestamp=1585141337609, value=XXXXXXXXX
 10                                   column=cfTest:customer_state, timestamp=1585141337609, value=VA
 10                                   column=cfTest:customer_street, timestamp=1585141337609, value=8598 Harvest Beacon Plaza
 10                                   column=cfTest:customer_zipcode, timestamp=1585141337609, value=22554
 100                                  column=cfTest:customer_city, timestamp=1585141337609, value=Caguas
 100                                  column=cfTest:customer_email, timestamp=1585141337609, value=XXXXXXXXX
 100                                  column=cfTest:customer_fname, timestamp=1585141337609, value=George
 100                                  column=cfTest:customer_lname, timestamp=1585141337609, value=Barrett
 100                                  column=cfTest:customer_password, timestamp=1585141337609, value=XXXXXXXXX
 100                                  column=cfTest:customer_state, timestamp=1585141337609, value=PR
 100                                  column=cfTest:customer_street, timestamp=1585141337609, value=4110 Silent Pointe
 100                                  column=cfTest:customer_zipcode, timestamp=1585141337609, value=00725
3 row(s) in 0.0380 seconds
```

```
hbase(main):004:0> scan 'sqoopTest', {STARTROW => '4000', STOPROW => '4003'}
```

```
hbase(main):004:0> scan 'sqoopTest', {STARTROW => '4000', STOPROW => '4003'}
ROW                                   COLUMN+CELL
 4000                                 column=cfTest:customer_city, timestamp=1585141364370, value=Memphis
 4000                                 column=cfTest:customer_email, timestamp=1585141364370, value=XXXXXXXXX
 4000                                 column=cfTest:customer_fname, timestamp=1585141364370, value=Mary
 4000                                 column=cfTest:customer_lname, timestamp=1585141364370, value=Ford
 4000                                 column=cfTest:customer_password, timestamp=1585141364370, value=XXXXXXXXX
 4000                                 column=cfTest:customer_state, timestamp=1585141364370, value=TN
 4000                                 column=cfTest:customer_street, timestamp=1585141364370, value=620 Red River Trail
 4000                                 column=cfTest:customer_zipcode, timestamp=1585141364370, value=38109
 4001                                 column=cfTest:customer_city, timestamp=1585141364370, value=Caguas
 4001                                 column=cfTest:customer_email, timestamp=1585141364370, value=XXXXXXXXX
 4001                                 column=cfTest:customer_fname, timestamp=1585141364370, value=Maria
 4001                                 column=cfTest:customer_lname, timestamp=1585141364370, value=Nixon
 4001                                 column=cfTest:customer_password, timestamp=1585141364370, value=XXXXXXXXX
 4001                                 column=cfTest:customer_state, timestamp=1585141364370, value=PR
 4001                                 column=cfTest:customer_street, timestamp=1585141364370, value=2517 Shady Branch Avenue
 4001                                 column=cfTest:customer_zipcode, timestamp=1585141364370, value=00725
 4002                                 column=cfTest:customer_city, timestamp=1585141364370, value=Columbus
 4002                                 column=cfTest:customer_email, timestamp=1585141364370, value=XXXXXXXXX
 4002                                 column=cfTest:customer_fname, timestamp=1585141364370, value=Mary
 4002                                 column=cfTest:customer_lname, timestamp=1585141364370, value=Smith
 4002                                 column=cfTest:customer_password, timestamp=1585141364370, value=XXXXXXXXX
 4002                                 column=cfTest:customer_state, timestamp=1585141364370, value=GA
 4002                                 column=cfTest:customer_street, timestamp=1585141364370, value=4320 Iron Highlands
 4002                                 column=cfTest:customer_zipcode, timestamp=1585141364370, value=31907
3 row(s) in 0.0150 seconds
```

De nuevo salen 3 filas, incluyendo los **customer_id** 4000, 4001 y 4002. En todas estas consultas (en la primera se aprecia mejor) la ordenación se realiza alfabéticamente, no numéricamente, es decir, el orden es 1, 10, 100, 1000, 11, 110, …, 9998, 9999; en vez de 1, 2, 3, 4, …, 9998, 9999.

2.6. Modifica la tabla creada con 5 número de versiones y 1Mb (1048576 bytes) de tamaño de bloque.

```
hbase(main):005:0> alter 'sqoopTest', {NAME => 'cfTest', VERSIONS =>
5, BLOCKSIZE => 1048576}
```

```
hbase(main):006:0> describe 'sqoopTest'
```

```
hbase(main):005:0> alter 'sqoopTest', {NAME => 'cfTest', VERSIONS => 5, BLOCKSIZE => 1048576}
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 2.7310 seconds

hbase(main):006:0> describe 'sqoopTest'
Table sqoopTest is ENABLED
sqoopTest
COLUMN FAMILIES DESCRIPTION
{NAME => 'cfTest', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '5', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL =
> 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '1048576', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
1 row(s) in 0.1610 seconds
```

2.7. Deshabilita la tabla "sqoopTest" y realiza un scan sobre la misma tabla. ¿Qué ocurre?.

```
hbase(main):007:0> disable 'sqoopTest'

hbase(main):008:0> scan 'sqoopTest'
```

```
hbase(main):007:0> disable 'sqoopTest'
0 row(s) in 2.4720 seconds

hbase(main):008:0> scan 'sqoopTest'
ROW                                           COLUMN+CELL

ERROR: sqoopTest is disabled.
```

```
ERROR: sqoopTest is disabled.
```

2.8. Vuelve a habilitar la tabla y realiza un conteo de filas, ¿cuántas filas hay?. Haz una consulta sobre el último row key. ¿Qué columnFamily tiene?. ¿Cuántos qualifier?. Indica sus valores.

```
hbase(main):009:0> enable 'sqoopTest'

hbase(main):010:0> count 'sqoopTest'
```

```
hbase(main):009:0> enable 'sqoopTest'
0 row(s) in 1.4280 seconds

hbase(main):010:0> count 'sqoopTest'
Current count: 1000, row: 10898
Current count: 2000, row: 11798
Current count: 3000, row: 1505
Current count: 4000, row: 2405
Current count: 5000, row: 3305
Current count: 6000, row: 4205
Current count: 7000, row: 5105
Current count: 8000, row: 6005
Current count: 9000, row: 6906
Current count: 10000, row: 7806
Current count: 11000, row: 8706
Current count: 12000, row: 9606
12435 row(s) in 2.2900 seconds

=> 12435
```

Según puede observarse en la captura del ejercicio 2.4, el último row key es el 9999.

`hbase(main):012:0>` **`get 'sqoopTest', 9999`**

```
hbase(main):012:0> get 'sqoopTest', 9999
COLUMN                                CELL
 cfTest:customer_city                 timestamp=1585141400600, value=Albuquerque
 cfTest:customer_email                timestamp=1585141400600, value=XXXXXXXXX
 cfTest:customer_fname                timestamp=1585141400600, value=Susan
 cfTest:customer_lname                timestamp=1585141400600, value=Smith
 cfTest:customer_password             timestamp=1585141400600, value=XXXXXXXXX
 cfTest:customer_state                timestamp=1585141400600, value=NM
 cfTest:customer_street               timestamp=1585141400600, value=5238 Sunny Walk
 cfTest:customer_zipcode              timestamp=1585141400600, value=87112
8 row(s) in 0.1380 seconds
```

El columnFamily es cfTest, con 8 qualifiers diferenciados:

- customer_city = Alburquerque
- customer_email = XXXXXXXXX
- customer_fname = Susan
- customer_lname = Smith
- customer_password = XXXXXXXXX
- customer_state = NM
- customer_street = 5238 Sunny Walk
- customer_zipcode = 87112