

**LAB7****IMPORT y EXPORT con SQUOOP****PARTE 1**

En este laboratorio vamos a evaluar el uso de Apache Sqoop para trabajar con bases de datos e incorporar contenido de bases de datos relacionales dentro de un sistema HDFS.

1. Vamos a utilizar la instalación local de MySQL, que se encuentra en la máquina virtual de Cloudera y, en este caso, lo que haremos es abrir una terminal para comprobar la información de ciertas tablas.

```
$ mysql -u root -p  
Enter password: cloudera
```

```
[cloudera@quickstart ~]$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 18  
Server version: 5.1.73 Source distribution  
  
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> █
```

Muestre las bases de datos contenidas en MySQL: show databases;

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
  
mysql> show databases  
-> ;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| cm |  
| firehose |  
| hue |  
| metastore |  
| mysql |  
| nav |  
| navms |  
| oozie |  
| retail_db |  
| rman |  
| sentry |  
+-----+  
12 rows in set (0.02 sec)
```

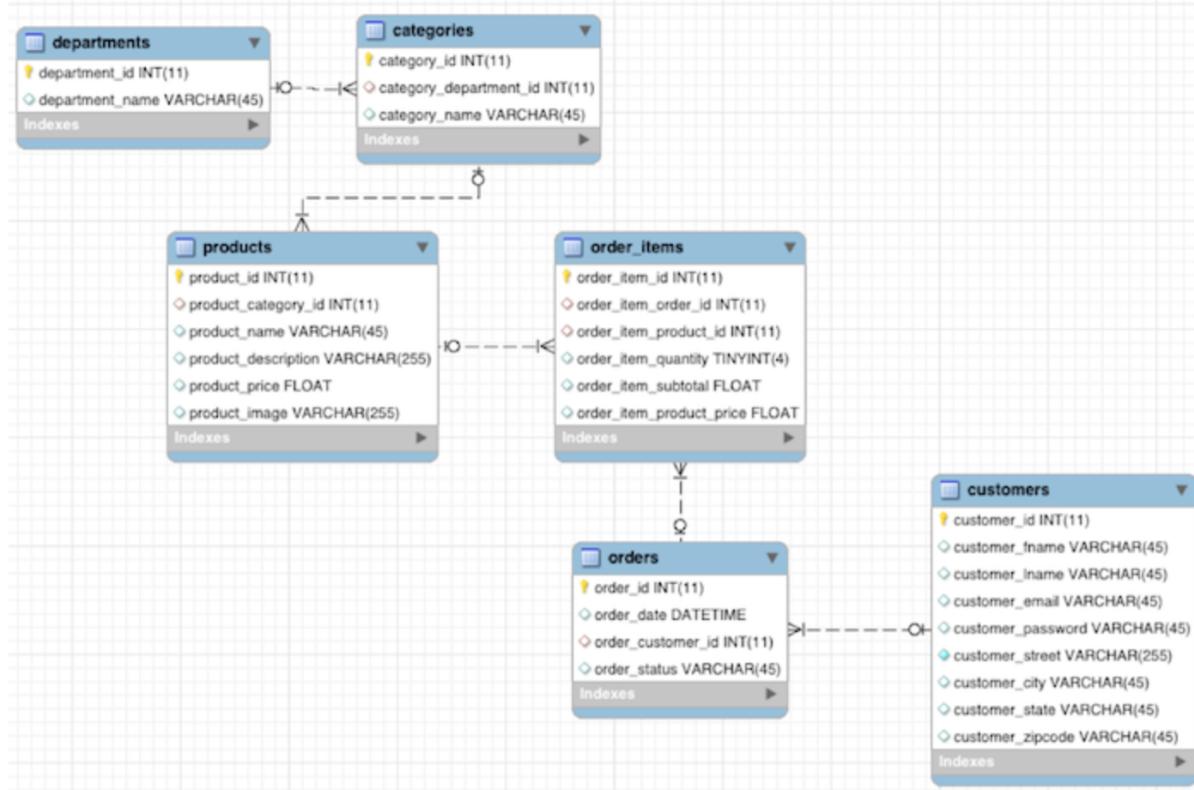
2. Seleccionemos la bdd con la que vamos a trabajar

```
mysql> use retail_db
```

```
mysql> use retail_db
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> ■
```

Familiarízate con el modelo de datos de retail\_db:



Dada la bdd, podemos mirar las tablas que contiene:

```
mysql> show tables;
+-----+
| Tables_in_retail_db |
+-----+
| categories
| customers
| departments
| order_items
| orders
| products
+-----+
6 rows in set (0.00 sec)

mysql> ■
```

3. Podemos hacer varias consultas y verificar qué datos existen en esta base de datos sobre customers y categories (ambas son dos tablas de nuestra base relacional). Pero primero familiarízate con la estructura que tienen estas dos tablas. Recuerda que el lenguaje de consulta que ocupamos para interactuar con MySQL es SQL :)

```
mysql> describe customers;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_id | int(11) | NO | PRI | NULL | auto_increment |
| customer_fname | varchar(45) | NO | | NULL |
| customer_lname | varchar(45) | NO | | NULL |
| customer_email | varchar(45) | NO | | NULL |
| customer_password | varchar(45) | NO | | NULL |
| customer_street | varchar(255) | NO | | NULL |
| customer_city | varchar(45) | NO | | NULL |
| customer_state | varchar(45) | NO | | NULL |
| customer_zipcode | varchar(45) | NO | | NULL |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql> describe categories;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| category_id | int(11) | NO | PRI | NULL | auto_increment |
| category_department_id | int(11) | NO | | NULL |
| category_name | varchar(45) | NO | | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> 
```

mysql> select \* from categories;

cloudera@quickstart:~		
File Edit View Search Terminal Help		
mysql> select * from categories		
> ;		
category_id	category_department_id	category_name
1	2	Football
2	2	Soccer
3	2	Baseball & Softball
4	2	Basketball
5	2	Lacrosse
6	2	Tennis & Racquet
7	2	Hockey
8	2	More Sports
9	3	Cardio Equipment
10	3	Strength Training
11	3	Fitness Accessories
12	3	Boxing & MMA
13	3	Electronics
14	3	Yoga & Pilates
15	3	Training by Sport
16	3	As Seen on TV!
17	4	Cleats
18	4	Men's Footwear
19	4	Women's Footwear
20	4	Kids' Footwear
21	4	Featured Shops
22	4	Accessories
23	5	Men's Apparel
24	5	Women's Apparel
25	5	Boys' Apparel
26	5	Girls' Apparel
27	5	Accessories

mysql> select \* from customers;

00725				
12431	Mary	Rios	XXXXXXXX	XXXXXXXX
1221	Cinder Pines		Kaneohe	HI
96744				
12432	Angela	Smith	XXXXXXXX	XXXXXXXX
1525	Jagged Barn Highlands		Caguas	PR
00725				
12433	Benjamin	Garcia	XXXXXXXX	XXXXXXXX
5459	Noble Brook Landing		Levittown	NY
11756				
12434	Mary	Mills	XXXXXXXX	XXXXXXXX
9720	Colonial Parade		Caguas	PR
00725				
12435	Laura	Horton	XXXXXXXX	XXXXXXXX
5736	Honey Downs		Summerville	SC
29483				

12435 rows in set (0.03 sec)

mysql>

4. Abre **otra terminal** para trabajar aquí con Sqoop (trabajaremos con dos terminales, una para mysql y otra para sqoop) y comprueba qué versión está instalada:  
 \$ sqoop version

```

cloudera@quickstart:~$ sqoop version
Warning: /usr/lib/sqoop/..../accumulo does not exist! Accumulo im
Please set $ACCUMULO_HOME to the root of your Accumulo installa
23/10/31 16:47:07 INFO sqoop.Sqoop: Running Sqoop version: 1.4.
Sqoop 1.4.6-cdh5.13.0
git commit id
Compiled by jenkins on Wed Oct 4 11:04:44 PDT 2017
[cloudera@quickstart ~]$ 
```

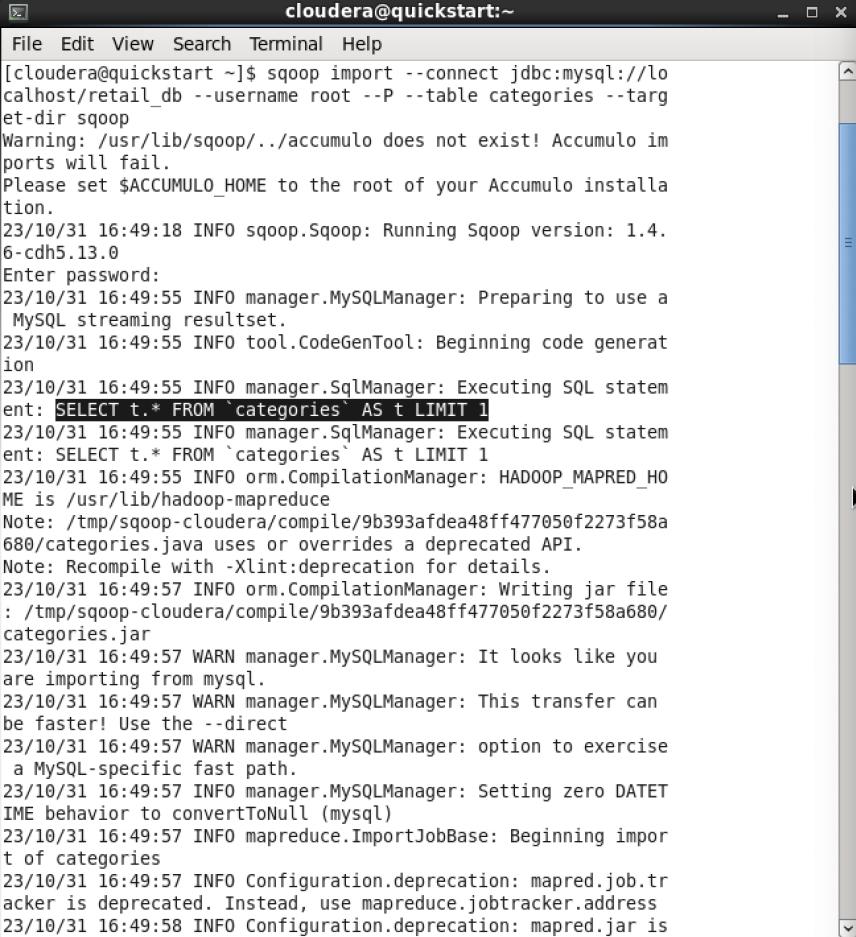
5. Lo que vamos a hacer es una inserción de datos sobre categorías relacionadas con productos en un comercio electrónico dentro de Sqoop. En otras palabras, pasaremos la data de la tabla categories de la bdd relacional a HDFS gracias a Sqoop.

Para esto debemos importar llamando al comando "import".

El comando "import" lo que hace es conectarse a la base de datos, en este caso MySQL, y configuraremos como conector de entrada un "jdbc", indicándole:

- el motor de la base de datos que es MySQL y está en el ordenador local (Centos),
- el lugar donde está disponible la base datos (que como hemos dicho es local),
- el nombre de la base de datos que es retail\_db
- el "usuario", en este caso es root y
- el "password" que es *cloudera pero lo pide después*,
- la tabla con la que vamos a trabajar
- el directorio en hdfs donde queremos generar, que le pondremos el nombre de sqoop

```
$ sqoop import --connect jdbc:mysql://localhost/retail_db --username root --P --table
categories --target-dir sqoop
```

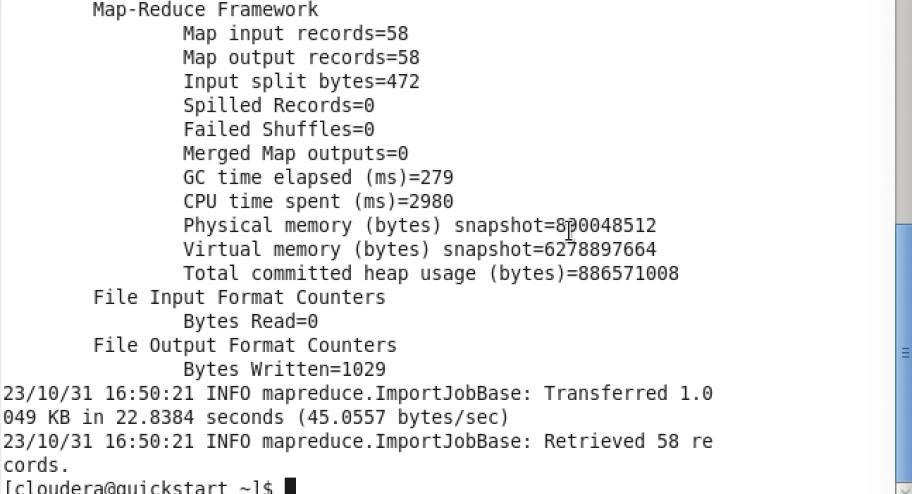


```

cloudera@quickstart:~$ sqoop import --connect jdbc:mysql://localhost/retail_db --username root --password --table categories --target-dir sqoop
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/10/31 16:49:18 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
23/10/31 16:49:55 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/10/31 16:49:55 INFO tool.CodeGenTool: Beginning code generation
23/10/31 16:49:55 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `categories` AS t LIMIT 1
23/10/31 16:49:55 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `categories` AS t LIMIT 1
23/10/31 16:49:55 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/9b393afdea48ff477050f2273f58a680/categories.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
23/10/31 16:49:57 INFO orm.CompilationManager: Writing jar file : /tmp/sqoop-cloudera/compile/9b393afdea48ff477050f2273f58a680/categories.jar
23/10/31 16:49:57 WARN manager.MySQLManager: It looks like you are importing from mysql.
23/10/31 16:49:57 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
23/10/31 16:49:57 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
23/10/31 16:49:57 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
23/10/31 16:49:57 INFO mapreduce.ImportJobBase: Beginning import of categories
23/10/31 16:49:57 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
23/10/31 16:49:58 INFO Configuration.deprecation: mapred.jar is

```

Esto ha disparado toda una serie de trabajos de forma interna y al final, lo que tenemos es una serie de registros, en este caso son 58 registros de la tabla original que se han copiado en HDFS.



```

Map-Reduce Framework
  Map input records=58
  Map output records=58
  Input split bytes=472
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=279
  CPU time spent (ms)=2980
  Physical memory (bytes) snapshot=830048512
  Virtual memory (bytes) snapshot=6278897664
  Total committed heap usage (bytes)=886571008
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=1029
23/10/31 16:50:21 INFO mapreduce.ImportJobBase: Transferred 1.049 KB in 22.8384 seconds (45.0557 bytes/sec)
23/10/31 16:50:21 INFO mapreduce.ImportJobBase: Retrieved 58 records.
[cloudera@quickstart ~]$ 

```

NOTA: Fíjate que se hace un "select" de la tabla de categories. En esta ejecución se están copiando todos los datos a un directorio HDFS dentro del ecosistema Hadoop (ver vía browser en Hadoop-HDFS NameNode – Utilities – Browse the file system – user/cloudera/sqoop).

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	cloudera	cloudera	0 B	Mon Jun 28 09:21:25 2021	1	128 MB	_SUCCESS
-rw-r--r--	cloudera	cloudera	271 B	Mon Jun 28 09:21:20 2021	1	128 MB	part-m-00000
-rw-r--r--	cloudera	cloudera	263 B	Mon Jun 28 09:21:24 2021	1	128 MB	part-m-00001
-rw-r--r--	cloudera	cloudera	266 B	Mon Jun 28 09:21:24 2021	1	128 MB	part-m-00002
-rw-r--r--	cloudera	cloudera	229 B	Mon Jun 28 09:21:24 2021	1	128 MB	part-m-00003

6. Ahora vamos a comprobar que el directorio de salida, que se llama sqoop, existe. Lo que hacemos es visualizar que la carpeta existe, y que dentro de esa carpeta existen ficheros.

\$ hdfs dfs -ls sqoop

```
[cloudera@quickstart ~]$ hdfs dfs -ls sqoop
Found 5 items
-rw-r--r-- 1 cloudera cloudera      0 2023-10-31 16:50 sqoop/_SUCCESS
-rw-r--r-- 1 cloudera cloudera  271 2023-10-31 16:50 sqoop/part-m-00000
-rw-r--r-- 1 cloudera cloudera  263 2023-10-31 16:50 sqoop/part-m-00001
-rw-r--r-- 1 cloudera cloudera  266 2023-10-31 16:50 sqoop/part-m-00002
-rw-r--r-- 1 cloudera cloudera  229 2023-10-31 16:50 sqoop/part-m-00003
```

Vemos que se tienen cuatro archivos de datos donde cada uno de ellos es una parte de lo que se tiene en la tabla categories de la base de datos en MySQL.

7. Ahora, lo que queremos hacer es comprobar, por ejemplo, qué hay dentro de uno de esos archivos de datos.

Entonces utilizamos el comando "cat" para visualizar el contenido de uno de esos archivos:

\$ hdfs dfs -cat sqoop/part-m-00000

```
[cloudera@quickstart ~]$ hdfs dfs -cat sqoop/part-m-00000
1,2,Football
2,2,Soccer
3,2,Baseball & Softball
4,2,Basketball
5,2,Lacrosse
6,2,Tennis & Racquet
7,2,Hockey
8,2,More Sports
9,3,Cardio Equipment
10,3,Strength Training
11,3,Fitness Accessories
12,3,Boxing & MMA
13,3,Electronics
14,3,Yoga & Pilates
15,3,Training by Sport
```

Vemos que en esta partición hay 15 categorías distintas. Así mismo, la última partición tiene parte de esas categorías tomadas desde la base de mysql:

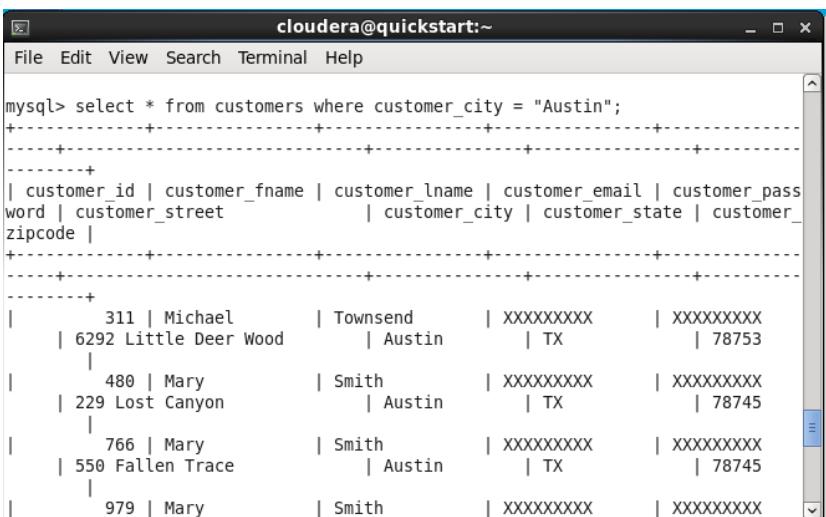
```
[cloudera@quickstart ~]$ hdfs dfs -cat sqoop/part-m-00003
44,7,Hunting & Shooting
45,7,Fishing
46,7,Indoor/Outdoor Games
47,7,Boating
48,7,Water Sports
49,8,MLB
50,8,NFL
51,8,NHL
52,8,NBA
53,8,NCAA
54,8,MLS
55,8,International Soccer
56,8,World Cup Shop
57,8,MLB Players
58,8,NFL Players
[cloudera@quickstart ~]$
```

8. Vamos a hacer otra prueba y vamos a importar datos de MySQL donde los datos tienen una restricción, usaremos la tabla customers, donde los datos van a ser filtrados por una condición dada, en este caso, sólo nos interesan aquellos clientes que vengan de la ciudad Austin.

```
mysql> describe customers;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| customer_id | int(11) | NO | PRI | NULL | auto_increment |
| customer_fname | varchar(45) | NO | | NULL | |
| customer_lname | varchar(45) | NO | | NULL | |
| customer_email | varchar(45) | NO | | NULL | |
| customer_password | varchar(45) | NO | | NULL | |
| customer_street | varchar(255) | NO | | NULL | |
| customer_city | varchar(45) | NO | | NULL | |
| customer_state | varchar(45) | NO | | NULL | |
| customer_zipcode | varchar(45) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
```

\*\* Primero miremos en la bdd relacional:

```
mysql> select * from customers where customer_city = "Austin";
```



```
cloudera@quickstart:~$ mysql> select * from customers where customer_city = "Austin";
+-----+-----+-----+-----+-----+-----+-----+-----+
| customer_id | customer_fname | customer_lname | customer_email | customer_password | customer_street | customer_city | customer_state | customer_zipcode |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 311 | Michael | Townsend | XXXXXXXXX | XXXXXXXXX | 6292 Little Deer Wood | Austin | TX | 78753 |
| 480 | Mary | Smith | XXXXXXXXX | XXXXXXXXX | 229 Lost Canyon | Austin | TX | 78745 |
| 766 | Mary | Smith | XXXXXXXXX | XXXXXXXXX | 550 Fallen Trace | Austin | TX | 78745 |
| 979 | Mary | Smith | XXXXXXXXX | XXXXXXXXX |
```

En este caso son 25 clientes.

9. Queremos importar sólo el resultado de esa consulta dentro de HDFS usando Sqoop, entonces realizamos un "import", llamamos al "conector" de base de datos que es el "jdbc MySQL", le decimos que la base de datos está en local,

- le damos el nombre de la bdd,
- el nombre del usuario y P
- la tabla con la que queremos trabajar
- el número de **mappers** que correrán para extraer los datos desde la bdd
- la carpeta (con el nombre de Austin) donde queremos guardar los resultados.
- la cláusula **where** para especificar que queremos aquellos clientes cuya ciudad sea Austin".

**NOTA:** Recuerde que pide el "password", que es *cloudera*.

```
$ sqoop import --connect jdbc:mysql://localhost/retail_db --username root --P --table
customers --m 1 --target-dir Austin --where "customer_city='Austin'"
```

```
cloudera@quickstart:~$ sqoop import --connect jdbc:mysql://localhost/retail_db --username root --P --table customers --m 1 --target-dir Austin --where "customer_city='Austin'"
Warning: /usr/lib/sqoop/..../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/10/31 16:58:20 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
23/10/31 16:58:25 INFO manager.MySQLManager: Preparing to use a MySQL streaming
resultset.
23/10/31 16:58:25 INFO tool.CodeGenTool: Beginning code generation
23/10/31 16:58:26 INFO manager.SqlManager: Executing SQL statement: SELECT t./*
FROM `customers` AS t LIMIT 1
23/10/31 16:58:26 INFO manager.SqlManager: Executing SQL statement: SELECT t./*
FROM `customers` AS t LIMIT 1
23/10/31 16:58:26 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/h
adoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/ec1616ad9c3f96cf34e6c013ed558e82/customers.ja
va uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
23/10/31 16:58:27 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-clo
uder/compile/ec1616ad9c3f96cf34e6c013ed558e82/customers.jar
23/10/31 16:58:27 WARN manager.MySQLManager: It looks like you are importing fr
om mysql.
23/10/31 16:58:27 WARN manager.MySQLManager: This transfer can be faster! Use t
he --direct
23/10/31 16:58:27 WARN manager.MySQLManager: option to exercise a MySQL-specifi
c fast path.
23/10/31 16:58:27 INFO manager.MySQLManager: Setting zero DATETIME behavior to
convertToNull (mysql)
23/10/31 16:58:27 INFO mapreduce.ImportJobBase: Beginning import of customers
23/10/31 16:58:27 INFO Configuration.deprecation: mapred.job.tracker is depreca
ted. Instead, use mapreduce.jobtracker.address
23/10/31 16:58:28 INFO Configuration.deprecation: mapred.jar is deprecated. Ins
tead, use mapreduce.job.jar
23/10/31 16:58:28 INFO Configuration.deprecation: mapred.map.tasks is depreca
d. Instead, use mapreduce.job.maps
23/10/31 16:58:28 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.
```

En este caso, la consulta es mucho más específica. Al ejecutar esto, de nuevo se generan una serie de trabajos que son transparentes para el usuario (nosotros no tenemos que programar), entonces se están creando jobs Hadoop de forma transparente, y al final se insertan estos registros en un archivo llamado Austin en HDFS. Note que son 25 registros encontrados.



```

cloudera@quickstart:~ 
File Edit View Search Terminal Help
File System Counters
  FILE: Number of bytes read=0
  FILE: Number of bytes written=171779
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=87
  HDFS: Number of bytes written=1873
  HDFS: Number of read operations=4
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
Job Counters
  Launched map tasks=1
  Other local map tasks=1
  Total time spent by all maps in occupied slots (ms)=2954
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=2954
  Total vcore-milliseconds taken by all map tasks=2954
  Total megabyte-milliseconds taken by all map tasks=3024896
Map-Reduce Framework
  Map input records=25
  Map output records=25
  Input split bytes=87
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=20
  CPU time spent (ms)=680
  Physical memory (bytes) snapshot=228995072
  Virtual memory (bytes) snapshot=1571516416
  Total committed heap usage (bytes)=221249536
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=1873
23/10/31 16:58:43 INFO mapreduce.ImportJobBase: Transferred 1.8291 KB in 14.938
3 seconds (125.3828 bytes/sec)
23/10/31 16:58:43 INFO mapreduce.ImportJobBase: Retrieved 25 records.
[cloudera@quickstart ~]$ 

```

10. Vamos a comprobar que se guardó en HDFS. Vamos a la carpeta Austin y ahí debería haber la partición generada con los resultados de clientes provenientes de Austin.

```

$ hdfs dfs -ls Austin
[cloudera@quickstart ~]$ hdfs dfs -ls Austin
Found 2 items
-rw-r--r-- 1 cloudera cloudera      0 2023-10-31 16:58 Austin/_SUCCESS
-rw-r--r-- 1 cloudera cloudera 1873 2023-10-31 16:58 Austin/part-m-00000

```

11. En este caso sólo hay un archivo, es el "part-m-00000", y vamos a comprobar qué contiene ese archivo de texto "part-m-00000":

```
$ hdfs dfs -cat Austin/part-m-00000
```

```
[cloudera@quickstart ~]$ hdfs dfs -cat Austin/part-m-00000
311,Michael,Townsend,XXXXXXXX,XXXXXXXX,6292 Little Deer Wood,Austin,TX,78753
480,Mary,Smith,XXXXXXXX,XXXXXXXX,229 Lost Canyon,Austin,TX,78745
766,Mary,Smith,XXXXXXXX,XXXXXXXX,550 Fallen Trace,Austin,TX,78745
979,Mary,Smith,XXXXXXXX,XXXXXXXX,9995 High Robin Pointe,Austin,TX,78745
996,Amanda,Smith,XXXXXXXX,XXXXXXXX,7007 Golden Edge,Austin,TX,78753
1028,Virginia,Sanders,XXXXXXXX,XXXXXXXX,1801 Jagged Dale Park,Austin,TX,78704
1702,Mary,Smith,XXXXXXXX,XXXXXXXX,67 Cotton Mountain Terrace,Austin,TX,78745
2100,Mary,Smith,XXXXXXXX,XXXXXXXX,1870 Stony Prairie Bend,Austin,TX,78753
2130,Mary,Smith,XXXXXXXX,XXXXXXXX,6523 Rustic Beacon Landing,Austin,TX,78745
2175,Mary,Smith,XXXXXXXX,XXXXXXXX,1230 Gentle Isle,Austin,TX,78704
2308,Mary,Morgan,XXXXXXXX,XXXXXXXX,3796 Rustic Autoroute,Austin,TX,78753
3489,Larry,Waller,XXXXXXXX,XXXXXXXX,5352 Silver Acres,Austin,TX,78704
4054,Mary,Smith,XXXXXXXX,XXXXXXXX,1661 Gentle Rabbit Turnabout,Austin,TX,78753
4508,Emma,Calderon,XXXXXXXX,XXXXXXXX,1859 Sunny Quay,Austin,TX,78704
4868,Andrea,Marks,XXXXXXXX,XXXXXXXX,4900 Heather Elk Row,Austin,TX,78753
4956,Mary,Smith,XXXXXXXX,XXXXXXXX,5913 Umber Orchard,Austin,TX,78745
5908,Emily,Smith,XXXXXXXX,XXXXXXXX,8767 Thunder Corners,Austin,TX,78753
6075,Mary,Smith,XXXXXXXX,XXXXXXXX,4546 Golden Hills Ridge,Austin,TX,78753
6839,Bobby,Bean,XXXXXXXX,XXXXXXXX,1538 Old Autumn Island,Austin,TX,78753
7476,Benjamin,Cole,XXXXXXXX,XXXXXXXX,9136 Emerald Oak Maze,Austin,TX,78704
7763,Philip,Smith,XXXXXXXX,XXXXXXXX,5270 Quaking Autoroute,Austin,TX,78745
9623,Mary,Davila,XXXXXXXX,XXXXXXXX,355 Amber Elk Abbey,Austin,TX,78704
12207,Jose,Smith,XXXXXXXX,XXXXXXXX,8473 Crystal Hickory Path,Austin,TX,78745
12302,Donald,Sampson,XXXXXXXX,XXXXXXXX,8550 Red Oak Bank,Austin,TX,78753
12381,Mary,Olsen,XXXXXXXX,XXXXXXXX,1208 Cotton Bluff Carrefour,Austin,TX,78704
[cloudera@quickstart ~]$
```

Vemos que hay datos sobre esos clientes y todos ellos están en la ciudad de Austin, ( antepenúltimo campo).

## PARTE 2

En esta siguiente parte del laboratorio es hacer lo contrario, es decir, exportar datos de Sqoop, de HDFS hacia una base de datos de MySQL.

1. En este caso, utilizamos MySQL, y creamos una tabla “temp” dentro de la base de datos "retail\_db", (bdd de ejemplo que hay en MySQL).

```
$ mysql -u root -p
Enter password: cloudera
```

```
mysql> use retail_db
```

Antes de crear la tabla temporal miremos qué tablas tiene retail\_db:

```
mysql> SHOW FULL TABLES FROM retail_db;
+-----+-----+
| Tables_in_retail_db | Table_type |
+-----+-----+
| categories         | BASE TABLE |
| customers          | BASE TABLE |
| departments         | BASE TABLE |
| order_items         | BASE TABLE |
| orders              | BASE TABLE |
| products             | BASE TABLE |
+-----+-----+
6 rows in set (0.00 sec)

mysql> ■
```

La tabla "temp" es temporal y tendremos tres campos o atributos:

- el primero, es el identificador que es un entero,
- el segundo es una categoría que también es un entero,
- y el tercero es un nombre de esa categoría que es un "string" de tamaño 30.

```
mysql> CREATE TABLE temp (id INT NOT NULL PRIMARY KEY, cat INT, name
VARCHAR(30));
mysql> CREATE TABLE temp (id INT NOT NULL PRIMARY KEY, cat INT, name VARCHAR(30));
Query OK, 0 rows affected (0.03 sec)

mysql> ■
```

Estamos definiendo ahora la tabla donde queremos guardar los datos que están en HDFS, queremos convertir datos de HDFS a base de datos relacional y guardarlos en esta tabla temporal (**Haga un SELECT de la tabla temp para que mire que está vacía**).

2. Una vez que está la tabla creada, **vamos a Sqoop** y lo que hacemos es una exportación de datos de Sqoop.

Exportar significa extraer datos que están en HDFS, y llevarlos a la base de datos relacional, que en este caso es MySQL. Para esto

- llamamos al jdbc de MySQL que es local,
- llamamos a la base retail\_db
- llamamos al mismo usuario, y password
- la tabla destino que es la tabla "temp" y que acabamos de crear (tenemos solo el esquema, sin datos aún),
- y la *carpeta llamada sqoop* donde están los datos que queremos exportar (aquí tenemos las 4 partes de datos sobre categorías).

```
$ sqoop export --connect jdbc:mysql://localhost/retail_db --username root --P --table
temp --export-dir sqoop
```

```
[cloudera@quickstart ~]$ sqoop export --connect jdbc:mysql://localhost/retail_db
--username root --P --table temp --export-dir sqoop
Warning: /usr/lib/sqoop/..../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/07/01 16:34:24 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.12.0
Enter password: ■
```

Vemos que se han exportado 58 registros:

```
Map-Reduce Framework
  Map input records=58
  Map output records=58
  Input split bytes=823
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=185
  CPU time spent (ms)=2120
  Physical memory (bytes) snapshot=832086016
  Virtual memory (bytes) snapshot=6260862976
  Total committed heap usage (bytes)=886571008
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=0
23/10/31 17:07:21 INFO mapreduce.ExportJobBase: Transferred 2.2207 KB in 22.334
6 seconds (101.815 bytes/sec)
23/10/31 17:07:21 INFO mapreduce.ExportJobBase: Exported 58 records.
[cloudera@quickstart ~]$ ■
```

3. Lo que haremos luego, una vez que esos trabajos Hadoop se han ejecutado, es comprobar que en MySQL se han insertado los datos:

```
mysql> select * from temp
-> ;
+---+---+---+
| id | cat | name
+---+---+---+
| 1  | 2   | Football
| 2  | 2   | Soccer
| 3  | 2   | Baseball & Softball
| 4  | 2   | Basketball
| 5  | 2   | Lacrosse
| 6  | 2   | Tennis & Racquet
| 7  | 2   | Hockey
| 8  | 2   | More Sports
| 9  | 3   | Cardio Equipment
| 10 | 3   | Strength Training
| 11 | 3   | Fitness Accessories
| 12 | 3   | Boxing & MMA
| 13 | 3   | Electronics
| 14 | 3   | Yoga & Pilates
| 15 | 3   | Training by Sport
| 16 | 3   | As Seen on TV!
| 17 | 4   | Teats
| 18 | 4   | Men's Footwear
| 19 | 4   | Women's Footwear
| 20 | 4   | Kids' Footwear
| 21 | 4   | Featured Shops
| 22 | 4   | Accessories
| 23 | 5   | Men's Apparel
| 24 | 5   | Women's Apparel
| 25 | 5   | Boys' Apparel
| 26 | 5   | Girls' Apparel
| 27 | 5   | Accessories
| 28 | 5   | Top Brands
| 29 | 5   | Shop By Sport
| 30 | 6   | Men's Golf Clubs
| 31 | 6   | Women's Golf Clubs
| 32 | 6   | Golf Apparel
| 33 | 6   | Golf Shoes
| 34 | 6   | Golf Bags & Carts
| 35 | 6   | Golf Gloves
| 36 | 6   | Golf Balls
| 37 | 6   | Electronics
| 38 | 6   | Kids' Golf Clubs
| 39 | 6   | Tennis Shop
| 40 | 6   | Accessories
| 41 | 6   | Trade In
| 42 | 7   | Bike & Skate Shop
| 43 | 7   | Camping & Hiking
| 44 | 7   | Hunting & Shooting
| 45 | 7   | Fishing
| 46 | 7   | Indoor/Outdoor Games
| 47 | 7   | Boating
| 48 | 7   | Water Sports
| 49 | 8   | MLB
| 50 | 8   | NFL
| 51 | 8   | NHL
| 52 | 8   | NBA
| 53 | 8   | NCAA
| 54 | 8   | MLS
| 55 | 8   | International Soccer
| 56 | 8   | World Cup Shop
| 57 | 8   | MLB Players
| 58 | 8   | NFL Players
+---+---+---+
58 rows in set (0.00 sec)

mysql> ■
```

4. Una vez que ha concluido su trabajo puede eliminar directorios en HDFS y la tabla temp de la bdd de MySQL

\$ hdfs dfs -rm Austin

```
cloudera@quickstart:~
```

File Edit View Search Terminal Help

```
[cloudera@quickstart ~]$ hdfs dfs -ls
Found 2 items
drwxr-xr-x  - cloudera cloudera      0 2023-10-31 16:58 Austin
drwxr-xr-x  - cloudera cloudera      0 2023-10-31 16:50 sqoop
[cloudera@quickstart ~]$ hdfs dfs -rm Austin
rm: `Austin': Is a directory
[cloudera@quickstart ~]$ hdfs dfs -rm -r Austin
Deleted Austin
[cloudera@quickstart ~]$ hdfs dfs -rm -r sqoop
Deleted sqoop
[cloudera@quickstart ~]$ ■
```

```
mysql> DROP TABLE temp;■
```

## PARTE 3: TRABAJO GRUPAL

1. SQL Practice: **Qué hace la siguiente sentencia de SQL?** Ejecútela en mysql.

```
SELECT c.customer_fname, c.customer_lname, c.customer_state, o.order_date,
o.order_status
FROM customers c
JOIN orders o
ON(c.customer_id = o.order_customer_id)
WHERE c.customer_state = 'TX'
LIMIT 20;
```

2. En este segundo ejercicio, se va a realizar una importación de datos desde MySQL a Hadoop usando la herramienta Sqoop. Vamos a usar la tabla "products" de la base de datos "retail\_db".

El objetivo del ejercicio es escribir en un directorio de HDFS todos los elementos de la tabla "products". Para resolver este ejercicio, tienes que importar todos los datos de la tabla "products" en una nueva carpeta llamada **productos** dentro de HDFS.

Revisa el mensaje final del job Hadoop que inserta los datos en HDFS y contesta la siguiente pregunta: **¿Cuántos registros ha generado la tarea Map del este job?** Debe buscar el número de registros indicado por "Map output records=" **¿Cuántos ficheros HDFS se generaron en la carpeta productos?**

3. Importa los registros de la tabla products de la base retail\_db al sistema HDFS en un nuevo directorio de resultados, siempre y cuando se cumpla que esos productos tienen un precio mayor a 800\$. Usa 3 mappers. Revisa el/los archivos HDFS generados (su contenido) y responde: **¿cuál es el nombre del producto más caro?** ¿Esto coincide con lo que se tiene almacenado en MySQL (haga la consulta SQL respectiva para hallar el valor y comparar)?

NOTA: se pueden ejecutar consultas SQL al momento de hacer un sqoop import, un ejemplo es:

```
$ sqoop import --connect jdbc:mysql://localhost/retail_db --username root --P --query
'SELECT * FROM products WHERE product_price>800 AND $CONDITIONS
ORDER BY product_price DESC' --split-by products.product_id --m 3 --target-dir
productos_filtro1
```

Sin embargo, las 3 particiones generadas con los resultados, tendrán internamente los datos ordenados, **revise esta salida generada.**

Guía:

```
mysql> select product_name,product_price  from products where product_price = (select max(product_price)from products);
+-----+-----+
| product_name      | product_price |
+-----+-----+
| SOLE E35 Elliptical |      1999.99 |
+-----+-----+
1 row in set (0.01 sec)
```

```
mysql> select * from products order by product_price desc limit 3;
+-----+-----+-----+-----+
| product_id | product_category_id | product_name      | product_description |
| product_price | product_image          |                   |
+-----+-----+-----+-----+
|      208 |          10 | SOLE E35 Elliptical |                   |
| 1999.99 | http://images.acmesports.sports/SOLE+E35+Elliptical |
|      66 |          4 | SOLE F85 Treadmill |                   |
| 1799.99 | http://images.acmesports.sports/SOLE+F85+Treadmill |
|     199 |          10 | SOLE F85 Treadmill |                   |
| 1799.99 | http://images.acmesports.sports/SOLE+F85+Treadmill |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> █
```

\*\* Para mayor detalle: <https://sqoop.apache.org/docs/1.4.6/SqoopUserGuide.html>