
Sqoop-Hadoop

Big Data Aplicado

23/07/16 - I.E.S Fernando Wirtz

Alejandro Regueiro Ruiz

Fecha	Motivo del cambio
	Versión inicial

Índice

Título 1.....	3
Título 2.....	3
Título 3.....	3

CREAR INSTANCIA EN EL CESGA.

Para crear una maquina dentro del cloud del cesga lo que tenemos que hacer es en el menú ir a la sección de instancias dentro de computación, lanzamos la instancia y la nombramos, ponemos una pequeña descripción si queremos, seleccionamos la versión del sistema operativo que queremos usar, seleccionamos cuantos nucleos y ram queremos en la instancia, en grupos de seguridad añadimos el grupo de openssh y asignamos la clave ssh del dispositivo con el que nos vamos a conectar a la instancia del cesga.

<input type="checkbox"/>	xuedua095-alejandro-regueiro	baseos-Debian-12-v1	10.133.29.71	a1.4c8m	alejandroregueiroru	Activa	ai ¹	nova	Ninguno	Corriendo	50 minutos	Crear instantánea
--------------------------	------------------------------	---------------------	--------------	---------	---------------------	--------	-----------------	------	---------	-----------	------------	-------------------

BIGDATA-APLICADO-SQOOP-HADOOP

1. Descargo Docker, otorgo permisos a mi usuario sobre docker y descargo e inicializo los dockers de MariaDB y PostgreSQL

```
-curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh ./get-docker.sh
-sudo usermod -a -G docker $USER
-sudo su - $USER
-docker volume create datosmariadb
-docker run -p 9907:3306 --name contedor_mariadb -v
datosmariadb:/var/lib/mysql --env MARIADB_RANDOM_ROOT_PASSWORD=1 --
env MARIADB_DATABASE=demaria --env MARIADB_USER=usuariamaria --env
MARIADB_PASSWORD=DonaMaria123456 --restart unless-stopped -d
mariadb:latest
-docker run --name de-postre-sql -e POSTGRES_PASSWORD=Cl431Ns3gur4 -p
5432:5432 -p 5433:5433 -d postgres
```

```
cesgaxuser@xuequab95-alejandro-regueiro:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
373a721fae24   postgres      "docker-entrypoint.s..." 10 days ago   Up About an hour   0.0.0.0:5432-5433->5432-5433/tcp, :::5432-5433->5432-5433/tcp   de-postre-sql
c99c61fe7ba8   mariadb:late  "docker-entrypoint.s..." 10 days ago   Up About an hour   0.0.0.0:9906->3306/tcp, [::]:9906->3306/tcp                     contedor_mariadb
```

2. Importación/Exportación de Datos

2.1

Con los dockers creados voy a importar los datos de las bases de datos de world y employees en MariaDB desde la terminal local donde están alojados los dockers

Instalando el pack del cliente de MySQL en el local con los dockers:

```
-apt install mysql-client
```

Puedo usar el comando:

```
mariadb -h(host) -u(user) -p(contraseña)' < employees.sql
```

Para importar los datos de la base de datos de empleados(Para poder ejecutar desde el local al dokcer sin necesidad de meterme dentro del docker, tengo que ejecutar el comando dentro de la carpeta en la que reside employees.sql para que tenga acceso al resto de archivos y se ejecute correctamente)

Hacemos lo mismo con la base de datos de world:

```
mariadb -h(host) -u(user) -p'(contraseña)' < world.sql
```

The screenshot shows a database management tool interface. On the left, a sidebar lists the database structure, including databases like 'employees' and 'world', and tables like 'city', 'country', and 'countrylanguage'. The main window displays the 'country' table from the 'world' database. The table has the following columns: 'country' (code), 'country' (name), 'AZ Continent', 'AZ Region', '123 SurfaceArea', '123 IndepYear', and '123 Population'. The data is presented in a grid with 25 rows visible. The status bar at the bottom indicates '200 row(s) fetched - 0,052s, on 2025-01-10'.

country	country	AZ Continent	AZ Region	123 SurfaceArea	123 IndepYear	123 Population
1	ABW	Aruba	North America	Caribbean	193	[NULL]
2	AFG	Afghanistan	Asia	Southern and Central Asia	652.090	1.919
3	AGO	Angola	Africa	Central Africa	1.246.700	1.975
4	AIA	Anguilla	North America	Caribbean	96	[NULL]
5	ALB	Albania	Europe	Southern Europe	28.748	1.912
6	AND	Andorra	Europe	Southern Europe	468	1.278
7	ANT	Netherlands Antilles	North America	Caribbean	800	[NULL]
8	ARE	United Arab Emirates	Asia	Middle East	83.600	1.971
9	ARG	Argentina	South America	South America	2.780.400	1.816
10	ARM	Armenia	Asia	Middle East	29.800	1.991
11	ASM	American Samoa	Oceania	Polynesia	199	[NULL]
12	ATA	Antarctica	Antarctica	Antarctica	13.120.000	[NULL]
13	ATF	French Southern territories	Antarctica	Antarctica	7.780	[NULL]
14	ATG	Antigua and Barbuda	North America	Caribbean	442	1.981
15	AUS	Australia	Oceania	Australia and New Zealand	7.741.220	1.901
16	AUT	Austria	Europe	Western Europe	83.859	1.918
17	AZE	Azerbaijan	Asia	Middle East	86.600	1.991
18	BDI	Burundi	Africa	Eastern Africa	27.834	1.962
19	BEL	Belgium	Europe	Western Europe	30.518	1.830
20	BEN	Benin	Africa	Western Africa	112.622	1.960
21	BFA	Burkina Faso	Africa	Western Africa	274.000	1.960
22	BGD	Bangladesh	Asia	Southern and Central Asia	143.998	1.971
23	BGR	Bulgaria	Europe	Eastern Europe	110.994	1.908
24	BHR	Bahrain	Asia	Middle East	694	1.971
25	BHS	Bahamas	North America	Caribbean	13.878	1.973
26	BLZ	Belize	North America	Central America	22.966	1.981
27	BLM	British Virgin Islands	North America	Caribbean	151	[NULL]
28	BLU	Burkina Faso	Africa	Western Africa	274.000	1.960
29	BLR	Bulgaria	Europe	Eastern Europe	110.994	1.908
30	BLS	Belize	North America	Central America	22.966	1.981
31	BMA	Bermuda	North America	Caribbean	54	[NULL]
32	BOM	Bombay	Asia	Western Asia	1.213.000	1.947
33	BON	Bonaire	North America	Caribbean	291	[NULL]
34	BOT	Botswana	Africa	Southern Africa	367.000	1.966
35	BRA	Brazil	South America	South America	8.511.965	1.914
36	BRE	Breton	Europe	Western Europe	1.213.000	1.947
37	BRI	Britain	Europe	Western Europe	1.213.000	1.947
38	BRO	Bromley	Europe	Western Europe	1.213.000	1.947
39	BRS	Brazil	South America	South America	8.511.965	1.914
40	BRT	Brazil	South America	South America	8.511.965	1.914
41	BRT	Brazil	South America	South America	8.511.965	1.914
42	BRT	Brazil	South America	South America	8.511.965	1.914
43	BRT	Brazil	South America	South America	8.511.965	1.914
44	BRT	Brazil	South America	South America	8.511.965	1.914
45	BRT	Brazil	South America	South America	8.511.965	1.914
46	BRT	Brazil	South America	South America	8.511.965	1.914
47	BRT	Brazil	South America	South America	8.511.965	1.914
48	BRT	Brazil	South America	South America	8.511.965	1.914
49	BRT	Brazil	South America	South America	8.511.965	1.914
50	BRT	Brazil	South America	South America	8.511.965	1.914

Como podemos apreciar en la imagen usando dbeaver los datos de employees y world se han importado correctamente

-h(host): Los host usados para importar los datos a los docker son:

MariaDB: 172.17.0.2 por ser el primer docker creado

Postgres: 172.17.0.3

2.2

2.2.1(base de datos world)

Ahora que tenemos los datos importados en MySQL, para importar los datos en postgres vamos a hacerlo mediante 'SQOOP'

Para conectarse a hadoop usamos el comando:

```
ssh (usuario)@hadoop.cesga.es
```

Para saber si hadoop se conecta correctamente con nuestros docker vamos a usar el comando de listar tables y ver si nos devuelve las tablas pertenecientes a la base de datos

```
-sqoop list-tables --username (usuario) -P --connect
jdbc:mysql://(Ip del docker:Puerto del docker)/(base de datos)
```

```
[xuedua095@cdh61-login3 ~]$ sqoop list-tables --username root -P --connect jdbc:mysql://10.133.29.71:9906/world
Warning: /opt/cloudera/parcels/CDH-6.1.1-1.cdh6.1.1.p0.875250/bin/../lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.1.1-1.cdh6.1.1.p0.875250/jars/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.1.1-1.cdh6.1.1.p0.875250/jars/log4j-slf4j-impl-2.8.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
25/01/24 17:59:24 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7-cdh6.1.1
Enter password:
25/01/24 17:59:28 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Fri Jan 24 17:59:28 CET 2025 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertificate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate verification.
city
country
countrylanguage
[xuedua095@cdh61-login3 ~]$
```

Como podemos apreciar la conexión es correcta ya que nos muestra las tablas que están dentro de la base de datos de 'world'

Para importar las tablas de MySQL al hadoop tenemos varias opciones, que serían importar todas las tablas de la base de datos a la vez, o importarlas de una en una, yo las voy a importar todas juntas:

```
-sqoop import-all-tables --connect jdbc:mysql://10.133.29.112:9906/world --
username root --password 'iR*-`ySoKm8bq"g[W;mb`w-rW3DR%$V' --
warehouse-dir /user/xuedua095/world --num-mappers 1
```

```
[xuedua095@cdh61-login3 ~]$ hdfs dfs -ls world
Found 3 items
drwxr-xr-x - xuedua095 xunta 0 2025-01-22 21:43 world/city
drwxr-xr-x - xuedua095 xunta 0 2025-01-22 21:43 world/country
drwxr-xr-x - xuedua095 xunta 0 2025-01-22 21:44 world/countrylanguage
[xuedua095@cdh61-login3 ~]$
```

Como podemos apreciar las tablas de world se han importando correctamente

Ahora vamos a exportar las tablas de world en Postgres, la ip de conexión de la instancia del cesga puede variar dependiendo de si fue reiniciada.

Antes de exportar las tablas tenemos que asegurarnos de que la base de datos a la que vamos a exportar tenga una estructura que acepte los datos.

La primera tabla que vamos a exportar es city

```
sqoop export --connect jdbc:postgresql://10.133.29.71:5432/world --username postgres --password 1234 --table city --export-dir /user/xuedua095/world/city --input-fields-terminated-by '\000' --num-mappers 1
```

```
25/01/24 16:52:12 INFO mapreduce.Job: Job job_1708445019134_13710 running in uber mode : false
25/01/24 16:52:12 INFO mapreduce.Job: map 0% reduce 0%
25/01/24 16:52:17 INFO mapreduce.Job: map 100% reduce 0%
25/01/24 16:52:17 INFO mapreduce.Job: Job job_1708445019134_13710 completed successfully
25/01/24 16:52:17 INFO mapreduce.Job: Counters: 33
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=246873
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=18403
    HDFS: Number of bytes written=0
    HDFS: Number of read operations=4
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=0
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=1
    Rack-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=2742
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=2742
    Total vcore-milliseconds taken by all map tasks=2742
    Total megabyte-milliseconds taken by all map tasks=2807808
  Map-Reduce Framework
    Map input records=984
    Map output records=984
    Input split bytes=150
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=41
    CPU time spent (ms)=1530
    Physical memory (bytes) snapshot=342843392
    Virtual memory (bytes) snapshot=2662432768
    Total committed heap usage (bytes)=607649792
    Peak Map Physical memory (bytes)=342843392
    Peak Map Virtual memory (bytes)=2662432768
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=0
25/01/24 16:52:17 INFO mapreduce.ExportJobBase: Transferred 17.9717 KB in 13.256 seconds (1.3557 KB/sec)
25/01/24 16:52:17 INFO mapreduce.ExportJobBase: Exported 984 records.
```

Parece que exportó correctamente

Ahora vamos a exportar country

```
sqoop export --connect jdbc:postgresql://10.133.29.71:5432/world --username  
postgres --password 1234 --table country --export-dir  
/user/xuedua095/world/country --input-fields-terminated-by '\000' -num-  
mappers 1
```

```
25/01/24 17:17:35 INFO mapreduce.Job: Job job_1708445019134_13738 running in uber mode : false
25/01/24 17:17:35 INFO mapreduce.Job: map 0% reduce 0%
25/01/24 17:17:40 INFO mapreduce.Job: map 100% reduce 0%
25/01/24 17:17:40 INFO mapreduce.Job: Job job_1708445019134_13738 completed successfully
25/01/24 17:17:40 INFO mapreduce.Job: Counters: 33
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=246941
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=32286
    HDFS: Number of bytes written=0
    HDFS: Number of read operations=4
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=0
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=2762
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=2762
    Total vcore-milliseconds taken by all map tasks=2762
    Total megabyte-milliseconds taken by all map tasks=2828288
  Map-Reduce Framework
    Map input records=239
    Map output records=239
    Input split bytes=142
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=48
    CPU time spent (ms)=1290
    Physical memory (bytes) snapshot=349691904
    Virtual memory (bytes) snapshot=2662420480
    Total committed heap usage (bytes)=586678272
    Peak Map Physical memory (bytes)=349691904
    Peak Map Virtual memory (bytes)=2662420480
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=0
25/01/24 17:17:40 INFO mapreduce.ExportJobBase: Transferred 31.5293 KB in 13.2672 seconds (2.3765 KB/sec)
25/01/24 17:17:40 INFO mapreduce.ExportJobBase: Exported 239 records.
```

Y por último vamos a exportar countrylanguage

```
sqoop export --connect jdbc:postgresql://10.133.29.71:5432/world --username
postgres --password 1234 --table countrylanguage --export-dir
/user/xuedua095/world/countrylanguage --input-fields-terminated-by '\000' --
num-mappers 1
```

```
25/01/24 16:47:03 INFO mapreduce.Job: Job job_1708445019134_13706 running in uber mode : false
25/01/24 16:47:03 INFO mapreduce.Job: map 0% reduce 0%
25/01/24 16:47:08 INFO mapreduce.Job: map 100% reduce 0%
25/01/24 16:47:08 INFO mapreduce.Job: Job job_1708445019134_13706 completed successfully
25/01/24 16:47:08 INFO mapreduce.Job: Counters: 33
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=246806
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=144627
    HDFS: Number of bytes written=0
    HDFS: Number of read operations=4
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=0
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=1
    Rack-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=2860
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=2860
    Total vcore-milliseconds taken by all map tasks=2860
    Total megabyte-milliseconds taken by all map tasks=2928640
  Map-Reduce Framework
    Map input records=4079
    Map output records=4079
    Input split bytes=139
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=45
    CPU time spent (ms)=1610
    Physical memory (bytes) snapshot=347791360
    Virtual memory (bytes) snapshot=2664488960
    Total committed heap usage (bytes)=590348288
    Peak Map Physical memory (bytes)=347791360
    Peak Map Virtual memory (bytes)=2664488960
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=0
25/01/24 16:47:08 INFO mapreduce.ExportJobBase: Transferred 141.2373 KB in 13.3509 seconds (10.5789 KB/sec)
25/01/24 16:47:08 INFO mapreduce.ExportJobBase: Exported 4079 records.
```

Parece que todas las exportaciones se han hecho sin problemas, vamos a comprobar con dbeaver si en la conexión de postgres hay tablas con datos en world

The screenshot shows the DBeaver 24.3.0 interface. On the left, the 'Navegador de Bases de Datos' (Database Navigator) pane displays a tree structure of the database 'employees 2'. The 'city' table is selected under the 'public' schema. The main pane on the right shows the 'city' table's data in a grid view. The table has columns: 'id', 'name', 'countrycode', 'district', and 'population'. The data is sorted by 'id' in ascending order. The table contains 26 rows of data, including cities from Afghanistan (AFG) and the Netherlands (NLD).

	id	name	countrycode	district	population
1	1	Kabul	AFG	Kabul	1.780.000
2	2	Qandahar	AFG	Qandahar	237.500
3	3	Herat	AFG	Herat	186.800
4	4	Mazar-e-Sharif	AFG	Balkh	127.800
5	5	Amsterdam	NLD	Noord-Holland	731.200
6	6	Rotterdam	NLD	Zuid-Holland	593.321
7	7	Haag	NLD	Zuid-Holland	440.900
8	8	Utrecht	NLD	Utrecht	234.323
9	9	Eindhoven	NLD	Noord-Brabant	201.843
10	10	Tilburg	NLD	Noord-Brabant	193.238
11	11	Groningen	NLD	Groningen	172.701
12	12	Breda	NLD	Noord-Brabant	160.398
13	13	Apeldoorn	NLD	Gelderland	153.491
14	14	Nijmegen	NLD	Gelderland	152.463
15	15	Enschede	NLD	Overijssel	149.544
16	16	Haarlem	NLD	Noord-Holland	148.772
17	17	Almere	NLD	Flevoland	142.465
18	18	Arnhem	NLD	Gelderland	138.020
19	19	Zaanstad	NLD	Noord-Holland	135.621
20	20	's-Hertogenbosch	NLD	Noord-Brabant	129.170
21	21	Amersfoort	NLD	Utrecht	126.270
22	22	Maastricht	NLD	Limburg	122.087
23	23	Dordrecht	NLD	Zuid-Holland	119.811
24	24	Leiden	NLD	Zuid-Holland	117.196
25	25	Haarlemmermeer	NLD	Noord-Holland	110.722
26	26	Zoetermeer	NLD	Zuid-Holland	110.214

Mirando en dbeaver parece que funcionó y los datos se exportaron correctamente

2.2.2(employees)

Importamos las tablas de la base de datos de employees al hadoop.

```
sqoop import-all-tables --connect jdbc:mysql://10.133.29.71:9906/employees
--username root --password 'iR*-`ySoKm8bq"g[W;mb`:w-rW3DR%$V' --
warehouse-dir /user/xuedua095/employees --num-mappers 1
```

```
[xuedua095@cdh61-login3 ~]$ hdfs dfs -ls
Found 5 items
drwx----- - xuedua095 xunta      0 2025-01-23 22:00 .Trash
drwx----- - xuedua095 xunta      0 2025-01-24 18:34 .staging
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:33 employees
-rw-r--r-- 3 xuedua095 xunta     12 2025-01-10 17:45 holamundo.txt
drwxr-xr-x - xuedua095 xunta      0 2025-01-22 21:44 world
^[[A[xuedua095@cdh61-login3 ~]$ hdfs dfs -ls employees
Found 8 items
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:33 employees/current_dept_emp
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:32 employees/departments
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:33 employees/dept_emp
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:32 employees/dept_emp_latest_date
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:32 employees/dept_manager
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:33 employees/employees
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:33 employees/salaries
drwxr-xr-x - xuedua095 xunta      0 2025-01-24 18:32 employees/titles
[xuedua095@cdh61-login3 ~]$ hdfs dfs -ls employees/salaries
Found 2 items
-rw-r--r-- 3 xuedua095 xunta      0 2025-01-24 18:33 employees/salaries/_SUCCESS
-rw-r--r-- 3 xuedua095 xunta 98781181 2025-01-24 18:33 employees/salaries/part-m-00000
[xuedua095@cdh61-login3 ~]$ |
```

Como podemos apreciar, las tablas se importaron correctamente

Vamos a exportar las tablas:

salaries

```
sqoop export --connect jdbc:postgresql://10.133.29.71:5432/employees
--username postgres --password 1234 --table salaries --export-dir
/user/xuedua095/employees/salaries --input-fields-terminated-by '\000' -
num-mappers 1
```

employees

```
sqoop export --connect jdbc:postgresql://10.133.29.71:5432/employees --
username postgres --password 1234 --table employees --export-dir
/user/xuedua095/employees/employees --input-fields-terminated-by '\000' --
num-mappers 1
```

titles

```
sqoop export --connect jdbc:postgresql://10.133.29.71:5432/employees --  
username postgres --password 1234 --table titles --export-dir  
/user/xuedua095/employees/titles --input-fields-terminated-by '\000' --num-  
mappers 1
```

dept_manager

```
sqoop export --connect jdbc:postgresql://10.133.28.76:5432/employees --  
username postgres --password 1234 --table dept_manager --export-dir  
/user/xuedua095/employees/dept_manager --input-fields-terminated-by '\000'  
--num-mappers 1
```

dept_emp

```
sqoop export --connect jdbc:postgresql://10.133.28.76:5432/employees --  
username postgres --password 1234 --table dept_emp --export-dir  
/user/xuedua095/employees/dept_emp --input-fields-terminated-by '\000' --  
num-mappers 1
```

departments

```
sqoop export --connect jdbc:postgresql://10.133.28.76:5432/employees --  
username postgres --password 1234 --table departments --export-dir  
/user/xuedua095/employees/departments --input-fields-terminated-by '\000' --  
num-mappers 1
```

2.2.3(Centros)

Importar centros a mariadb

```
sqoop import-all-tables --connect jdbc:mysql://10.133.28.76:9906/Centros --  
username root --password 'iR*-\`ySoKm8bq"g[W;mb`:w-rW3DR%$V' --  
warehouse-dir /user/xuedua095/Centros --num-mappers 1
```

```
[xuedua095@cdh61-login2 ~]$ hdfs dfs -ls Centros  
Found 1 items  
drwxr-xr-x - xuedua095 xunta 0 2025-01-27 20:17 Centros/ProvJuntas  
[xuedua095@cdh61-login2 ~]$ |
```

Exportar centros a postgres

```
sqoop export --connect jdbc:postgresql://10.133.28.76:5432/centros --  
username postgres --password 1234 --table provjuntas --export-dir  
/user/xuedua095/Centros/ProvJuntas --input-fields-terminated-by '\000' -num-  
mappers 1
```

3. Un CSV 'pesado'

Importar csv a hdfs

```
[xuedua095@cdh61-login2 ~]$ hdfs dfs -put /opt/cesga/cursos/pyspark_2022/datasets/NYC_taxi_trip_records/yellow_tripdata_2018-12.csv /user/xuedua095/|
```

Exportar a dbeaver

```
sqoop export --connect jdbc:mysql://10.133.28.76:9906/employees --  
username root --password 'iR*-\`ySoKm8bq"g[W;mb`:w-rW3DR%$V' --table  
taxi_trips --export-dir /user/xuedua095/yellow_tripdata_2018-12.csv -num-  
mappers 1
```

--Checksum del CSV

```
[xuedua095@cdh61-login2 ~]$ ls
ProvJuntas.java  country.java          current_dept_emp.java  dept_emp.java
city.java        countrylanguage.java  departments.java      dept_emp_late
[xuedua095@cdh61-login2 ~]$ md5sum yellow_tripdata_2018-12.csv
93f56ae952cebe9e44dbcdc553884063  yellow_tripdata_2018-12.csv
[xuedua095@cdh61-login2 ~]$ |
```

--Número de líneas del CSV

```
[xuedua095@cdh61-login2 ~]$ wc -l yellow_tripdata_2018-12.csv
8173233 yellow_tripdata_2018-12.csv
[xuedua095@cdh61-login2 ~]$ |
```

--Count de la tabla

```
cesgaxuser@xuedua095-alejandro-regueiro:~$ mariadb -h 172.17.0.2 -u root -p -e "SELECT COUNT(*) FROM employees.taxi_trips;"
Enter password:
+-----+
| COUNT(*) |
+-----+
| 8173233 |
+-----+
cesgaxuser@xuedua095-alejandro-regueiro:~$ |
```