

T2.8: Sqoop

Big Data Aplicado

24/01/2025 – ÍES Fernando Wirtz

Constantin Madalin Ismana

Fecha	Motivo del cambio
24/01/2025	Versión inicial
24/01/2025	Creación del docker
24/01/2025	Instalación de MariaDB
24/01/2025	Importar Base de World y Employees
24/01/2025	Instalación de PostgreSQL
24/01/2025	Importación de World y Employees con Sqoop
27/01/2025	Importación de la Base de Centros Educativos
27/01/2025	Exportación de la Base de Centros Educativos con Sqoop
27/01/2025	Finalización del Cuarto Paso

Sumario

Primera parte: Creación da máquina Virtual e Dockers.....2

Instalación de MariaDB..... 4

Instalación de PostgreSQL.....7

Segunda parte: Importar os datos de proba..... 10

Importación de World y Employees y Centros Educativos con Sqoop..... 17

Cuarta parte: Un CSV "pesado".....22

Primera parte: Creación de máquina Virtual e Dockers

Creación del docker:

Primer Paso vamos a crear nuestra instancia en el Cesga que sería la siguiente:

☐ [xuedua088-cmi-Sqoop-NO-BORRAR](#) [baseos-Debian-12-v1](#) 10.133.27.96 [a1.4c8m](#) cmi

Después de crearla nos vamos a conectar a ella con el siguiente comando:

ssh cesgaxuser@XX.XXX.XX.XX-- → Donde están las X hay que poner la IP que nos dan cuando creas la instancia

```
(base) PS C:\Users\constantin.madalin.i> ssh cesgaxuser@10.133.27.96
```

Cuando entremos en la instancia vamos a ejecutar los siguientes comandos para tener la máquina actualizada para que no nos salten fallos:

sudo apt update- → Este comando sirve para actualizar los paquetes

sudo apt -y dist-upgrade- → Aquí actualizamos las distribuciones que tenemos

sudo apt -y install curl- → Aquí instalamos la herramienta curl

Después vamos a ejecutar el script recomendado de la página de docker para obtenerlo y ejecutarlo.

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

```
sudo sh ./get-docker.sh
```

Después de Ejecutar los dos comandos vamos a añadir nuestro usuario al grupo docker para así evitar usar siempre sudo cuando vallamos a instalar algo que sería con el siguiente comando:

```
sudo usermod -a -G docker $USER
```

Después probamos si tenemos el docker instalado ejecutando el siguiente comando:

```
docker run hello-world
```

Y nos tiene que salir esto:

```
57243f8bcab5  hello-world  "/hello"  10 days ago  Exited (0) 10 days ago
               reverent_jemison
```

Instalación de MariaDB

Ahora vamos a instalar MariaDB, para instalar mariadb vamos a crear el volumen datosmariadb con el siguiente comando:

```
docker volumen create datosmariadb
```

Después de crear el volumen vamos a crear la base de datos con su respectado usuario que sería usuariamaria con la siguiente contraseña DonaMaria123456 con la siguiente Base de datos que sería demaria y la contraseña del root aleatoria, también el nombre del contenedor donde estará nuestra base de datos será contedor_mariadb

```
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
```

Para ver nuestra contraseña del root vamos a ejecutar el siguiente comando:

```
docker logs contedor_mariadb
```

Que nos saldría todo esto:

```
2025-01-13 20:00:26+00:00 [Note] [Entrypoint]: Temporary server started.
2025-01-13 20:00:26+00:00 [Note] [Entrypoint]: GENERATED ROOT PASSWORD: z)az->8`3M_,3K|+bZZ}HC0}$V7sDwI~
2025-01-13 20:00:26+00:00 [Note] [Entrypoint]: Creating database demaria
2025-01-13 20:00:26+00:00 [Note] [Entrypoint]: Creating user usuariamaria
```

Vamos a ver si tenemos la base creada, para eso vamos a conectarnos con el usuario ROOT y su contraseña a mariadb:

```
(base) cesgaxuser@xuedua088-cmi-sqoop:~$ docker exec -it contedor_mariadb mariadb -uroot -p'z)az->8`3M_,3K|+bZZ}HC0}$V7sDwI~'
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 11.6.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Ahora vamos a hacer:

show databases;

Aquí nos saldría todas las bases de datos que tenemos:

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| World    |
| demaria  |
| employees|
| information_schema |
| mysql    |
| performance_schema |
| sys      |
| test_db  |
| world    |
+-----+
```

Después de ver las bases vamos a seleccionar una con use database; Donde pone database ponemos el nombre de la base de datos que queremos usar

```
MariaDB [(none)]> use demaria;
Database changed
MariaDB [demaria]>
```

Después de entrar ejecutamos el comando show tables; este comando sirve para ver las tablas que tenemos en la base de datos.

Por ejemplo, en esta base de datos no tenemos nada. Pero entro en una para ver como ejecuta este comando.

```
MariaDB [demaria]> show tables;  
Empty set (0.000 sec)  
  
MariaDB [demaria]>
```

```
MariaDB [demaria]> use world;  
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  
MariaDB [world]> show tables;  
+-----+  
| Tables_in_world |  
+-----+  
| city             |  
| country          |  
| countrylanguage |  
+-----+  
3 rows in set (0.000 sec)  
  
MariaDB [world]>
```

Después de tener nuestra Base de datos con nuestras tablas vamos a crear un usuario con los siguientes comandos:

CREATE USER 'usuario-a-crear'@'%' IDENTIFIED BY 'contrasinal-abc123.'; → Aquí vamos a crear el usuario con la contraseña que queremos

GRANT ALL PRIVILEGES ON base-de-datos. * TO 'usuario-a-crear'@'%'; - → Aquí vamos a dar privilegios a nuestra base de datos

FLUSH PRIVILEGES; - → Aquí vamos a ejecutar para que refresque los privilegios para que los coja

Instalación de PostgreSQL

Ahora vamos a instalar PostgreSQL para eso vamos a ejecutar el siguiente comando:

```
docker run --name de-postre-sql -e POSTGRES_PASSWORD=Cl431Ns3gur4 \
-p 5432:5432 -p 5433:5433 -d postgres
```

Aquí estamos poniendo el nombre del contenedor postre-sql con la siguiente contraseña Cl431Ns3gur4 con el siguiente nombre que sería postgres

```
(base) cesgaxuser@xuedua088-cmi-sqoop:~$ docker run --name de-postre-sql -e POSTGRES_PASSWORD=Cl431Ns3gur4 \
-p 5432:5432 -p 5433:5433 -d postgres
```

Ahora vamos a entrar en una base de datos y hacemos los siguientes comandos:

\l → Asomar bases de datos

\c postgres → Seleccionar base de datos a utilizar

\dt → Visualizar tablas

CREATE DATABASE sobremesa; → Crear base de datos

CREATE USER lambon WITH PASSWORD 'FI4nD3C4f3'; → Crear un usuario

GRANT ALL PRIVILEGES ON DATABASE sobremesa to lambon; → Dar permisos sobre a BBDD a usuario

\c sobremesa → Conectamos a base de datos:

GRANT ALL ON SCHEMA public TO lambon; → Damos permiso a esquema public:

docker exec -it de-postre-sql \

psql postgresql://lambon:FI4nD3C4f3@localhost/sobremesa → Conectar con el usuario, contraseña de e BBDD creadas


```

postgres=# \l
postgres=# \c wo1rd
connection to server on socket "/var/run/postgresql/.s.PGSQL.5432" failed: FATAL: database "wo1rd" does not exist
Previous connection kept
postgres=# \c world
You are now connected to database "world" as user "postgres".
world=# \dt
Did not find any relations.
world=# \c employees
You are now connected to database "employees" as user "postgres".
employees=# \dt
Did not find any relations.
employees=# \l

```

Name	Owner	Encoding	Locale	Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
centros	postgres	UTF8	libc		en_US.utf8	en_US.utf8			
employees	postgres	UTF8	libc		en_US.utf8	en_US.utf8			=Tc/postgres + postgres=CTc/postgres+ mada=CTc/postgres
postgres	postgres	UTF8	libc		en_US.utf8	en_US.utf8			
sobremesa	postgres	UTF8	libc		en_US.utf8	en_US.utf8			=Tc/postgres + postgres=CTc/postgres+ made=CTc/postgres
template0	postgres	UTF8	libc		en_US.utf8	en_US.utf8			=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	libc		en_US.utf8	en_US.utf8			=c/postgres + postgres=CTc/postgres
test_db	postgres	UTF8	libc		en_US.utf8	en_US.utf8			
world	postgres	UTF8	libc		en_US.utf8	en_US.utf8			

(8 rows)

```

employees=#

```

Conectarse con Postgres con DBeaver

Configuración de la conexión "employees"

Ajustes de conexión
PostgreSQL ajustes de conexión

PostgreSQL

Ajustes de conexión

- Inicialización
- Comandos de shell
- Identificación de cliente
- Transacciones
- General
- Metadatos
- Errores y timeouts
- Data Transfer
- Data Editor
- Editor SQL

General PostgreSQL Driver properties SSH SSL

+ Network configurations...

Server

Connect by: ☒ Host ☐ URL

URL: jdbc:postgresql://10.133.27.96:5432/employees

Host: 10.133.27.96 Port: 5432

Database: employees ☐ Show all databases

Authentication

Authentication: Database Native

Nombre de usuario: postgres

Contraseña: ☒ Save password

Advanced

Session role: Local Client: PostgreSQL Binaries

Ahora vamos a importar la base de datos de World

```
(base) cesgaxuser@xuedua888 ~$ cat sqoop:~$ docker cp /home/cesgaxuser/world-db/world.sql e2cd2b0ec8e1:/world.sql
Successfully copied 4080kB to e2cd2b0ec8e1:/world.sql
(base) cesgaxuser@xuedua888 ~$ docker exec -it e2cd2b0ec8e1 bash
root@e2cd2b0ec8e1:/# ls
bin boot dev docker-entrypoint-initdb.d employees_postgres.sql employees.sql etc home lib lib64 media mnt opt proc root run sbin srv sys test_db tmp usr var world world.sql
root@e2cd2b0ec8e1:/# source world.sql
bash: world.sql: line 1: syntax error near unexpected token '('
bash: world.sql: line 1: '- MySQL dump 10.13 Distrib 8.0.19, for osx10.14 (x86_64)'
root@e2cd2b0ec8e1:/# cat world.sql
-- MySQL dump 10.13 Distrib 8.0.19, for osx10.14 (x86_64)
--
-- Host: 127.0.0.1 Database: world
-- Server version      8.0.19-debug

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

Y después la comprobamos si esta con los comandos de arriba y listo.

Segunda parte: Importar os datos de proba

Importar Base de World y Employees y Centros Educativos

Ahora vamos a Importar una Base de Datos que sería World, Employees y Centros educativos a nuestro MariaDB.

“Aquí me faltaría meter centros educativos, pero lo metí al final en el postgres”

Para obtener las Bases de datos nos vamos a las siguientes rutas para obtener el enlace:

<https://downloads.mysql.com/docs/world-db.tar.gz>

https://github.com/datacharmer/test_db

```
(base) cesgaxuser@xuedua088-cmi-sqoop:~$ wget https://github.com/datacharmer/test_db/releases/download/v1.0.7/test_db-1.0.7.tar.gz
```

Para obtenerlos dentro instalamos con “apt-get install wget” y ponemos el enlace para obtener los archivos

```
(base) cesgaxuser@xuedua088-cmi-sqoop:~$ ls
CosasdeClase      get-docker.sh  test_db          world-db
employees-installation.html  miniconda3    test_db-1.0.7.tar.gz  world-db.tar.gz
(base) cesgaxuser@xuedua088-cmi-sqoop:~$
```

Descomprimos el archivo con el siguiente comando

`tar -xvf employees_db-full-1.0.7.tar.gz ->` **Este comando sirve para descomprimir el archivo .tar**

```
(base) cesgaxuser@xuedua088-cmi-sqoop:~$ tar xvf test_db-1.0.7.tar.gz
```

Hacemos esto con los dos archivos test_db-1.0.1.tar.gz y World-db.tar.gz para que nos salgas las carpetas de test_db y World-de

Después vamos a importar la base con el siguiente comando:

```
docker exec -i contedor_mariadb mariadb -uroot -p'z)az->8`3M_,3K|+bZZ}HCO}$V7sDwl~' world <
//home/cesgaxuser/world-db/world.sql
```

Después de ejecutarlo vamos a ver si está dentro:

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| World    |
| demaria  |
| employees|
| information_schema |
| mysql    |
| performance_schema |
| sys      |
| test_db  |
| world    |
+-----+
```

```
MariaDB [demaria]> use world;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

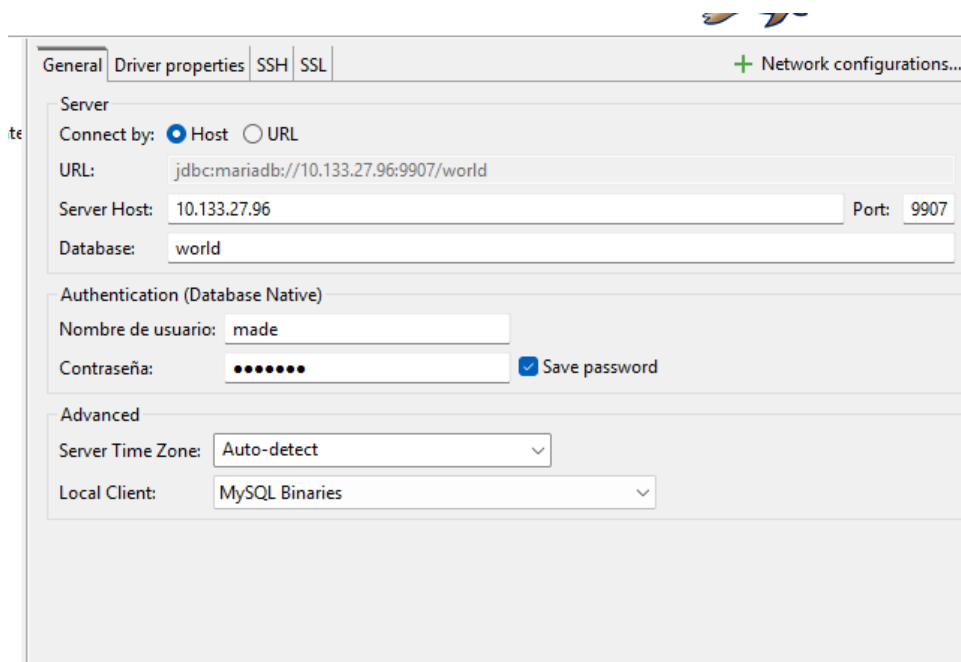
Database changed
MariaDB [world]> show tables;
+-----+
| Tables_in_world |
+-----+
| city             |
| country          |
| countrylanguage  |
+-----+
3 rows in set (0.000 sec)

MariaDB [world]>
```

Hacemos lo mismo para la base de datos de Employees, pero para ejecutar el comando tenemos que estar dentro de la carpeta de Test_db y ejecutar el siguiente comando:

```
(base) cesgaxuser@xuedua088-cmi-sqoop:~$ mariadb -h172.17.0.2 -uroot -p'z)az->8`3M_,3K|+bZZ}HCO}$V7sDwI~' < employees.sql
```

Ahora nos vamos a conectarnos con el DBeaver nuestras dos Bases de datos de MariaDB.



En la siguiente foto tenemos lo siguiente:

Server Host: que sería nuestra Ip de la instancia

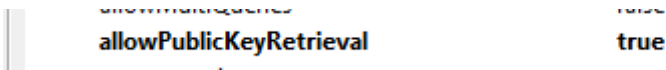
Database: El nombre de la base de datos que vamos a conectarnos

Nombre de Usuario: El nombre que vamos a conectarnos, cual creamos anteriormente.

Contraseña: La contraseña del usuario que creamos

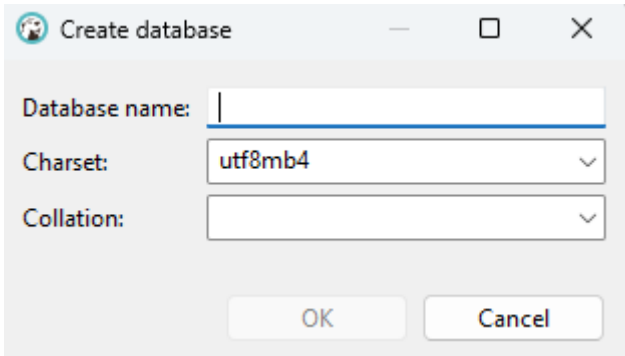
Port: El puerto por donde nos vamos a conectarnos

Nos faltaría cambiar un parámetro en Driver Properties que sería la siguiente opción:

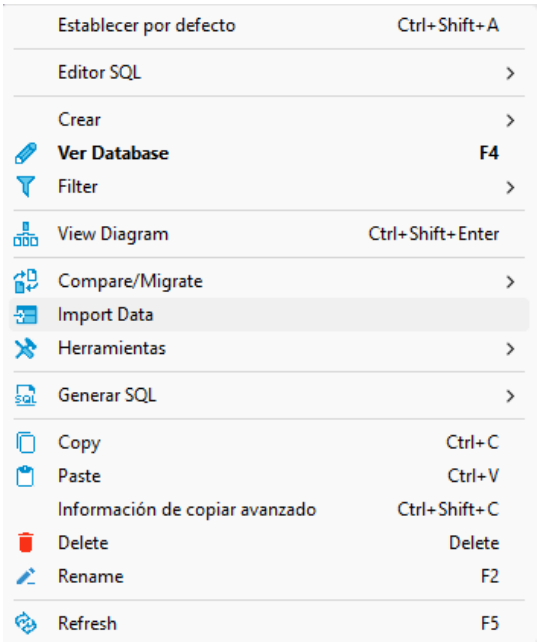


Esto lo hacemos con las dos Bases de datos con MariaDB dependiendo el usuario que tengamos creado cambiando eso y el puerto.

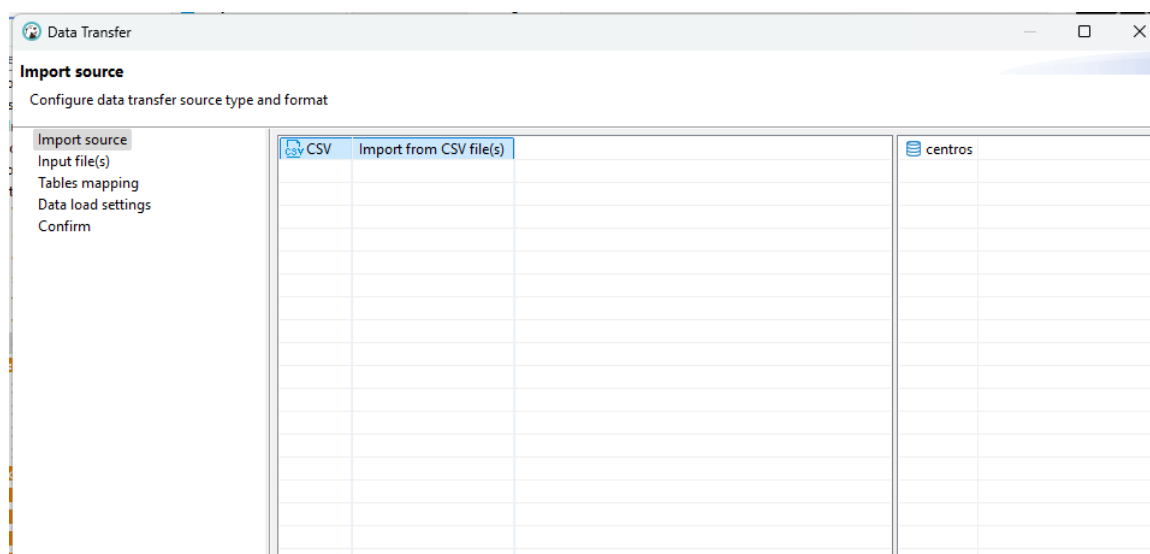
Ahora vamos a crear una Base de datos para meter los csv de los centros:



Aquí vamos a poner centros y le damos a Ok

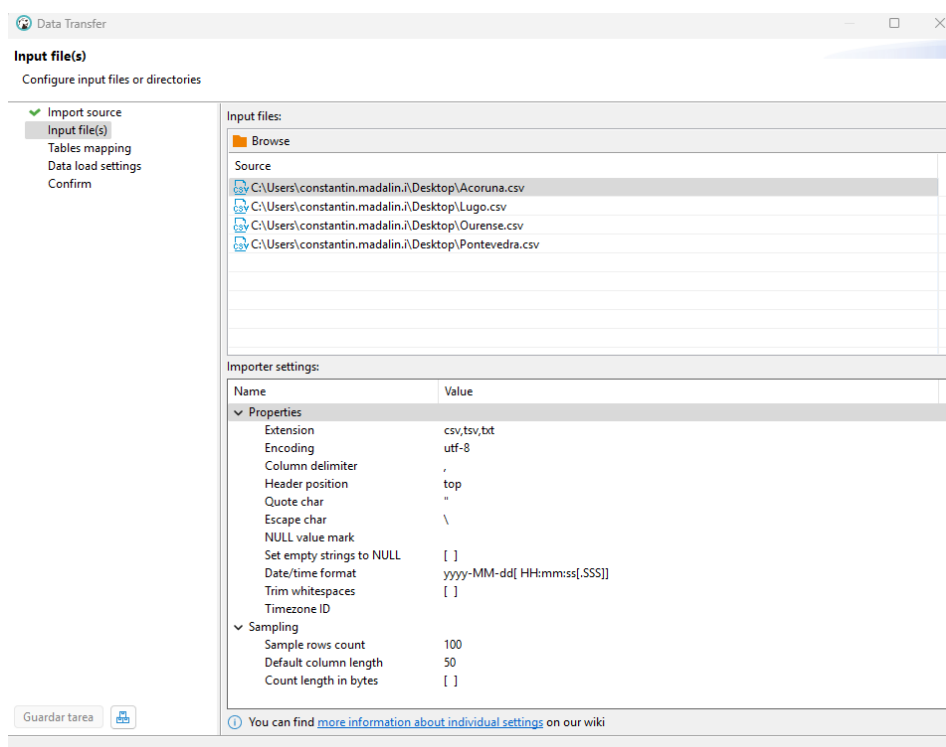


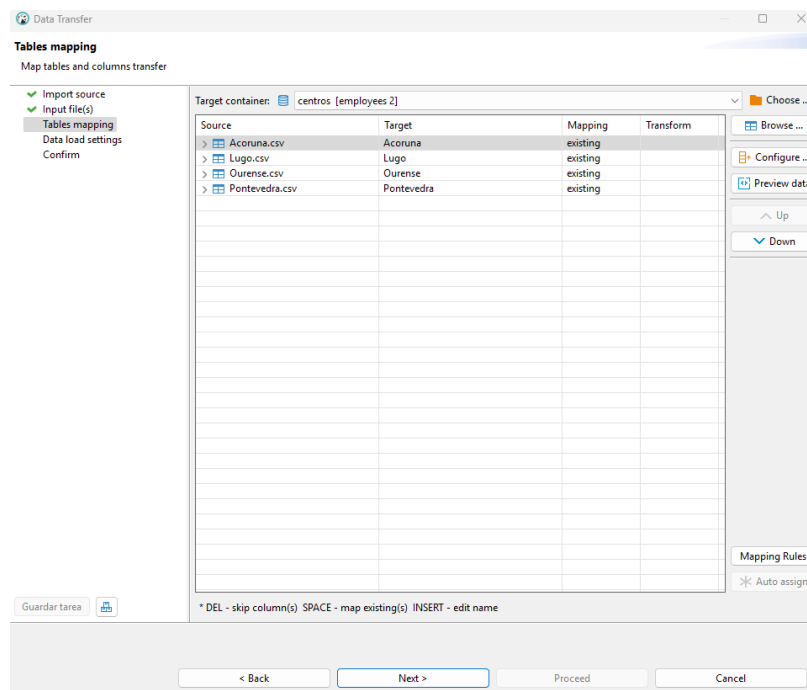
En esta opción seleccionamos Import Database



Nos saldría esta Opción y le damos a siguiente

Escogemos los csv que tenemos y los ponemos ahí y le damos a siguiente





T2.8: Sqoop

Big Data Aplicado

Le damos todo a siguiente y si nos sale algún fallo le damos a skip all y ya tendríamos los csv importados con los datos:

centros					
Tables					
Acoruna	16K				
Lugo	16K				
Ourense	16K				
Pontevedra	16K				
Views					
Indexes					
Procedures					
Packages					
Sequences					
Triggers					
Events					
Users					
Administer					
System Info					

Column Name	#	Data Type	Not Null
123 Código	1	int(11)	[]
A-Z Nome	2	varchar(50)	[]
A-Z Enderezo	3	varchar(50)	[]
A-Z Concello	4	varchar(50)	[]
A-Z Provincia	5	varchar(50)	[]
123 Cód. postal	6	int(11)	[]
A-Z Teléfono	7	varchar(50)	[]
A-Z Tipo de centro	8	varchar(50)	[]
123 COORDENADA_X	9	double	[]
123 COORDENADA_Y	10	double	[]
A-Z TITULARIDADE	11	varchar(50)	[]
A-Z ENSINO_CONCERTADO	12	varchar(50)	[]
A-Z DEPENDENTE	13	varchar(50)	[]

Acoruna	Lugo	Ourense	Pontevedra	7
Propiedades	Datos	Diagrama ER		
employees 2 Databases centros				
Acoruna	Enter a SQL expression to filter results (use Ctrl+Sp)			
Grilla	123 Código	A-Z Nome	A-Z Enderezo	
1	15.000.016	CEIP Plurilingüe Sa	Campo da Feira s/	15000
2	15.000.107	CEIP A Maía	Rúa Peregrina, s/n	
3	15.000.132	CPR Plurilingüe Al	Lugar Biduido de /	
4	15.000.338	CPI As Mirandas	Avenida Celso Emi	
5	15.000.363	CEIP Ponte dos Br	Avenida Arsenio Ig	
6	15.000.545	EMUSPR Presto Viv	Rúa Posse, 37, 1º	
7	15.000.569	CEIP San Xosé Obr	Travesa de Meiras,	
8	15.000.600	CPR Plurilingüe Nt	Rúa Carmen 22	
9	15.000.612	CEIP de Arzúa	Rúa Baltasar Parda	
10	15.001.033	CFEA de Guisamo	Lugar de Bos 14	
11	15.001.070	CPI de Cruz do Sar	Rúa A Senra 33	
12	15.001.112	CPR Plurilingüe Nu	Avenida de Castela	
13	15.001.124	CEIP Francisco Val	Rúa Carregal 4	
14	15.001.136	IES Francisco Agui	Avenida Da Coruña	
15	15.001.148	IES As Mariñas	Avenida da Coruña	
16	15.001.151	EMUSPR de Vedra	Santa Cruz de Riba	
17	15.001.239	CPI Armando Cota	Campo do Rosario	
18	15.001.367	CEIP Plurilingüe de	Rúa Exipto 108	
19	15.001.471	CEIP Plurilingüe Pr	Rúa Praia Xardin s/	
20	15.001.616	CEIP de Escarabote	Lampón (Santiago	
21	15.001.665	CPI Plurilingüe Ant	O Forte	
22	15.001.744	EEl dos Ánxeles	Rúa da Fonte 1	
23	15.001.811	EEl de Sabaxáns	Bastavales (San Xu	
24	15.001.847	CEIP Plurilingüe de	Paseo Pedrouzos s	
25	15.001.926	CPI As Revoltas	Lugar A Carballa 1	
26	15.002.025	CEIP Plurilingüe El	Rúa Barreiro 14	
27	15.002.050	CEIP O Areal	Rúa Eugenio Lope	
28	15.002.062	CEIP de Camelle	O Areal, 13 (Came	
29	15.002.086	CEIP Plurilingüe de	Rúa O Allo s/n	
30	15.002.165	CEIP Plurilingüe W	Rúa Wenceslao Fei	
31	15.002.566	CPR Plurilingüe Ar	Estrada RAZO Km	
32	15.002.578	CEIP Fogar	Rúa Vila de Ordes :	
33	15.002.581	IES Alfredo Brañas	Rúa Vila de Corcuk	

Importación de World y Employees y Centros Educativos con Sqoop

Vamos a crear primero la carpeta de world

```
[xuedua088@cdh61-login5 ~]$ hdfs dfs -mkdir world
```

Vamos a importar la base con las tablas de world con el siguiente comando:

```
sqoop import --username root --password 'z)az->8`3M_,3K|+bZZ}HCO}$V7sDwl~' --connect  
jdbc:mysql://10.133.27.96:9907/world --table city --target-dir /user/xuedua088/world/city --num-mappers 1
```

```
sqoop import --username root --password 'z)az->8`3M_,3K|+bZZ}HCO}$V7sDwl~' --connect  
jdbc:mysql://10.133.27.96:9907/world --table country --target-dir /user/xuedua088/world/country --num-mappers 1
```

```
sqoop import --username root --password 'z)az->8`3M_,3K|+bZZ}HCO}$V7sDwl~' --connect  
jdbc:mysql://10.133.27.96:9907/world --table countrylanguage --target-dir  
/user/xuedua088/world/countrylanguage --num-mappers 1
```

Después de ejecutar el comando nos saldrá todo esto con todas las 3 tablas que tenemos:

```

File System Counters
  FILE: Number of bytes read=0
  FILE: Number of bytes written=246534
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=85
  HDFS: Number of bytes written=144485
  HDFS: Number of read operations=6
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
  HDFS: Number of bytes read erasure-coded=0
Job Counters
  Launched map tasks=1
  Other local map tasks=1
  Total time spent by all maps in occupied slots (ms)=3093
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=3093
  Total vcore-milliseconds taken by all map tasks=3093
  Total megabyte-milliseconds taken by all map tasks=3167232
Map-Reduce Framework
  Map input records=4079
  Map output records=4079
  Input split bytes=85
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=58
  CPU time spent (ms)=2320
  Physical memory (bytes) snapshot=358981632
  Virtual memory (bytes) snapshot=2671296512
  Total committed heap usage (bytes)=584056832
  Peak Map Physical memory (bytes)=358981632
  Peak Map Virtual memory (bytes)=2671296512
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=144485
25/01/24 18:33:52 INFO mapreduce.ImportJobBase: Transferred 141.0986 KB in 13.6298 seconds (10.3522 KB/sec)
25/01/24 18:33:52 INFO mapreduce.ImportJobBase: Retrieved 4079 records.

```

Después de hacer los comandos nos tienen que salir esto:

```

[xuedua088@cdh61-login5 ~]$ hdfs dfs -ls world
Found 3 items
drwxr-xr-x  - xuedua088 xunta          0 2025-01-24 18:33 world/city
drwxr-xr-x  - xuedua088 xunta          0 2025-01-22 21:12 world/country
drwxr-xr-x  - xuedua088 xunta          0 2025-01-22 21:13 world/countrylanguage
[xuedua088@cdh61-login5 ~]$ |

```

```

[xuedua088@cdh61-login5 ~]$ hdfs dfs -ls world
Found 3 items
drwxr-xr-x  - xuedua088 xunta          0 2025-01-24 18:33 world/city
drwxr-xr-x  - xuedua088 xunta          0 2025-01-22 21:12 world/country
drwxr-xr-x  - xuedua088 xunta          0 2025-01-22 21:13 world/countrylanguage
[xuedua088@cdh61-login5 ~]$ hdfs dfs -ls world/city
Found 2 items
-rw-r--r--  3 xuedua088 xunta          0 2025-01-24 18:33 world/city/_SUCCESS
-rw-r--r--  3 xuedua088 xunta    144485 2025-01-24 18:33 world/city/part-m-00000
[xuedua088@cdh61-login5 ~]$ hdfs dfs -ls world/country
Found 2 items
-rw-r--r--  3 xuedua088 xunta          0 2025-01-22 21:12 world/country/_SUCCESS
-rw-r--r--  3 xuedua088 xunta    32141 2025-01-22 21:12 world/country/part-m-00000
[xuedua088@cdh61-login5 ~]$ hdfs dfs -ls world/countrylanguage
Found 2 items
-rw-r--r--  3 xuedua088 xunta          0 2025-01-22 21:13 world/countrylanguage/_SUCCESS
-rw-r--r--  3 xuedua088 xunta    18250 2025-01-22 21:13 world/countrylanguage/part-m-00000
[xuedua088@cdh61-login5 ~]$

```

Ahora vamos a hacer lo mismo con la Base de datos de Employees, pero esta vez vamos a hacer todas las tablas de un comando que sería el siguiente comando:

```
sqoop import-all-tables --username root --password 'z)az->8`3M_,3K|+bZZ}HCO}$V7sDwl~' --connect jdbc:mysql://10.133.28.241:9907/employees --warehouse-dir /user/xuedua088/employees --num-mappers 1 --autoreset-to-one-mapper
```

Ahora vamos a Export las Dos bases de datos a PostgreSQL

Vamos a hacer 3 veces con las tablas que tenemos en world

```
[xuedua088@cdh61-login5 ~]$ sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/world --table city --export-dir /user/xuedua088/world/city --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/world --table city --export-dir /user/xuedua088/world/city --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/world --table country --export-dir /user/xuedua088/world/country --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/world --table countrylanguage --export-dir /user/xuedua088/world/countrylanguage --input-fields-terminated-by '\000' --num-mappers 1
```

Ahora vamos con la otra base de datos que serían estos comandos:

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/employees --table departments --export-dir /user/xuedua088/employees/departments --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect  
jdbc:postgresql://10.133.27.96:5432/employees --table dept_emp --export-dir  
/user/xuedua088/employees/dept_emp --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect  
jdbc:postgresql://10.133.27.96:5432/employees --table dept_manager --export-dir  
/user/xuedua088/employees/dept_manager --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect  
jdbc:postgresql://10.133.27.96:5432/employees --table employees --export-dir  
/user/xuedua088/employees/employees --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect  
jdbc:postgresql://10.133.27.96:5432/employees --table salaries --export-dir  
/user/xuedua088/employees/salaries --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect  
jdbc:postgresql://10.133.27.96:5432/employees --table titles --export-dir /user/xuedua088/employees/titles --  
input-fields-terminated-by '\000' --num-mappers 1
```

Al acabar esto vamos a importar todas las tablas con el siguiente comando:

```
sqoop import-all-tables --username root --password 'z)az->8`3M_,3K|+bZZ}HCO}$V7sDwl~' --connect jdbc:mysql://10.133.27.96:9907/centros --warehouse-dir /user/xuedua088/centros --num-mappers 1 --autoreset-to-one-mapper
```

Y nos quedaría lo siguiente en el HDFS:

```
[xuedua088@cdh61-login6 ~]$ hdfs dfs -ls
Found 6 items
drwx----- - xuedua088 xunta      0 2025-01-25 19:00 .Trash
drwx----- - xuedua088 xunta      0 2025-01-27 16:45 .staging
drwxr-xr-x - xuedua088 xunta      0 2025-01-27 16:44 centros
drwxr-xr-x - xuedua088 xunta      0 2025-01-22 22:03 employees
-rw-r--r--  3 xuedua088 xunta     11 2025-01-10 17:45 holamundo.txt
drwxr-xr-x - xuedua088 xunta      0 2025-01-24 18:33 world
[xuedua088@cdh61-login6 ~]$ hdfs dfs -ls centros
Found 4 items
drwxr-xr-x - xuedua088 xunta      0 2025-01-27 16:44 centros/Acoruna
drwxr-xr-x - xuedua088 xunta      0 2025-01-27 16:44 centros/Lugo
drwxr-xr-x - xuedua088 xunta      0 2025-01-27 16:44 centros/Ourense
drwxr-xr-x - xuedua088 xunta      0 2025-01-27 16:44 centros/Pontevedra
[xuedua088@cdh61-login6 ~]$ hdfs dfs -ls centros/Acoruna
Found 2 items
-rw-r--r--  3 xuedua088 xunta      0 2025-01-27 16:44 centros/Acoruna/_SUCCESS
-rw-r--r--  3 xuedua088 xunta    74935 2025-01-27 16:44 centros/Acoruna/part-m-00000
```

Y ahora la vamos a exportar con el siguiente comando:

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/centros --table Acoruna --export-dir /user/xuedua088/centros/Acoruna --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/centros --table Lugo --export-dir /user/xuedua088/centros/Lugo --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/centros --table Ourense --export-dir /user/xuedua088/centros/Ourense --input-fields-terminated-by '\000' --num-mappers 1
```

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect jdbc:postgresql://10.133.27.96:5432/centros --table Pontevedra --export-dir /user/xuedua088/centros/Pontevedra --input-fields-terminated-by '\000' --num-mappers 1
```

Cuarta parte: Un CSV "pesado"

Ahora vamos a cargar un CSV Pesado que seria los siguientes pasos:

Primero vamos a copiar el archivo a nuestro HDFS con el siguiente comando:

Después de copiar el archivo, vamos a comprobar las tablas que vamos a tener para crearlas en el DBeaver con el siguiente comando:

Después de sacarlas nos vamos al DBeaver y las creamos:

```
CREATE TABLE public."taxi" (  
  VendorID text,  
  tpep_pickup_datetime text,  
  tpep_dropoff_datetime text,  
  passenger_count text,  
  trip_distance text,  
  RatecodeID text,  
  store_and_fwd_flag text,  
  PULocationID text,  
  DOLocationID text,  
  payment_type text,  
  fare_amount text,  
  extra text,  
  mta_tax text,  
  tip_amount text,  
  tolls_amount text,  
  improvement_surcharge text,  
  total_amount text  
);
```

Después de crear las tablas ya podemos hacer el export con el siguiente comando:

```
sqoop export --username postgres --password 'Cl431Ns3gur4' --connect  
jdbc:postgresql://10.133.27.96:5432/centros --table taxi --export-dir /user/xuedua088/taxis/all.csv --num-  
mappers 1
```

Después de hacer todo hacemos un Checksum do CSV para obtener el identificador:

```
[xuedua088@cdh61-login6 ~]$ md5sum all.csv  
93f56ae952cebe9e44dbcdc553884063 all.csv
```

Ahora vamos a mirar o número de líneas do CSV e un COUNT(*) da tabla exportada con el siguiente comando:

```
[xuedua088@cdh61-login6 ~]$ wc -l yellow_tripdata_2018-12.csv  
8173233 yellow_tripdata_2018-12.csv  
[xuedua088@cdh61-login6 ~]$
```

```
centros=# select count(*) from taxi;  
count  
-----  
8173233  
(1 row)  
  
centros=#
```