

## Sqoop. Práctica 3: Exportación de datos desde HDFS a mySql

La herramienta **export** exporta un conjunto de archivos de HDFS a un RDBMS.

La tabla de destino ya debe existir en la base de datos.

Los archivos de entrada son leídos y procesados en un conjunto de registros acorde con los delimitadores especificados por el usuario.

La operación predeterminada es ejecutar un conjunto de sentencias INSERT que inyectan los registros en la base de datos.

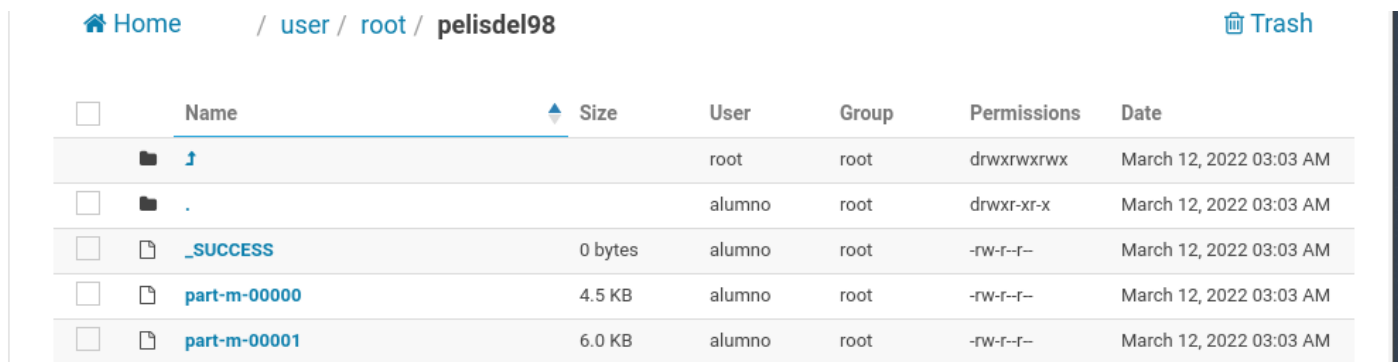
### Importación de los datos a HDFS

El primer paso a realizar es traer a HDFS los datos que exportaremos posteriormente. Como en prácticas anteriores, utilizamos la tabla *movie* que ya conocemos. En este caso, filtramos los datos por películas que son posteriores a 1998, y los registros resultantes se guardan en formato de texto.

```
sqoop import --connect jdbc:mysql://node1/movielens --username root --password hadoop123 \  
--target-dir /user/root/pelisd98 --table movie --where "year > 1998" \  
--fields-terminated-by ';' -m 2 --driver com.mysql.jdbc.Driver
```

### Comprobamos los resultados en HDFS

Usando la consola de HUE verificamos que el directorio */user/root/pelisd98* se ha generado con los archivos que guardan los registros obtenidos de la importación.








<input type="checkbox"/>	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	 f		root	root	drwxrwxrwx	March 12, 2022 03:03 AM
<input type="checkbox"/>	 .		alumno	root	drwxr-xr-x	March 12, 2022 03:03 AM
<input type="checkbox"/>	 _SUCCESS	0 bytes	alumno	root	-rw-r--r--	March 12, 2022 03:03 AM
<input type="checkbox"/>	 part-m-00000	4.5 KB	alumno	root	-rw-r--r--	March 12, 2022 03:03 AM
<input type="checkbox"/>	 part-m-00001	6.0 KB	alumno	root	-rw-r--r--	March 12, 2022 03:03 AM

Figure 1: Archivos resultantes de la importación

Al visualizar el contenido de uno de ellos, se comprueba que cada campo de cada fila está delimitado con el carácter ';'. Es importante resaltar que este delimitador debe ser único para no tener conflictos con los datos de los registros.

File Browser

Back
Home

Page 1 to 2 of 2

Edit file
Refresh
View as binary
Download

Last modified  
03/12/2022  
12:03 PM  
User  
alumno  
Group

/ user / root / pelisdel98 / part-m-00000

2564;Empty Mirror, The;1999
2566;Doug's 1st Movie;1999
2567;EDtv;1999
2568;Mod Squad, The;1999
2570;Walk on the Moon, A;1999
2571;Matrix, The;1999
2572;10 Things I Hate About You;1999
2574;Out-of-Towners, The;1999
2580;Go;1999
2581;Never Been Kissed;1999
2583;Cookie's Fortune;1999
2584;Foolish;1999
2586;Goodbye, Lover;1999
2587;Life;1999

Figure 2: Uso de delimitadores en los archivos

## Creación de la tabla en mySql

La tabla de destino debe existir en el sistema gestor de bases de datos, y los tipos de datos de los campos deben coincidir con los que se van a exportar.

Al intentar insertar un registro que no coincida con la definición de la tabla en la BD relacional, el proceso de exportación fallará finalizando este proceso.

### Creamos la tabla destino movie98

Como los datos son equivalentes a los de la tabla *movie* se crea otra tabla *movie98* con los mismos tipos de datos. Para ver qué base de datos estamos usando en mySql podemos usar la función *select database()*

```
mysql> select database();
```

```
mysql> create table movie98 (id INT NOT NULL PRIMARY KEY, nombre VARCHAR(75), ANIO INT);
```

```
mysql> desc movie98;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   | PRI | NULL    |       |
| nombre | varchar(75)   | YES  |     | NULL    |       |
| ANIO  | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Figure 3: Descripción de la tabla que recoge los datos exportados

## Exportación desde HDFS a la tabla *movie98*

El proceso de exportación es equivalente a la importación, siendo más exigente en cuanto a la inserción / actualización de los datos en la tabla de BD relacional. Hay que tener en cuenta el formato de los archivos (texto o serializado), campos a insertar, delimitadores de los campos, claves repetidas o tipos de datos entre otros aspectos.

La sentencia a utilizar es

```
sqoop export --connect jdbc:mysql://node1/movielens
--username root --password hadoop123
--table movie98 --input-fields-terminated-by ';'
--export-dir /user/root/pelisd98
--update-mode allowinsert
-m 1
```

Analizamos los parámetros utilizados:

- **-connect jdbc:mysql://node1/movielens** : Base de datos a conectar
- **-table movie98** : tabla donde se alojan los datos exportados desde HDFS
- **-input-fields-terminated-by ‘;’** : carácter delimitador de los campos en los ficheros de HDFS
- **-export-dir /user/root/pelisd98** : ruta en hdfs donde están los datos a exportar
- **-update-mode allowinsert** : modo de actualización, en este caso permite inserciones (otro es *updateonly*)

## Comprobación del resultado

Desde la consola de mysql se puede ver el número de registros importados en la tabla *movie98*. Debieran coincidir con el número de registros que se filtraron de la tabla *movie*

```
mysql> select * from movie98;
Empty set (0.01 sec)

mysql> select count(*) from movie98;
+-----+
| count(*) |
+-----+
|      418 |
+-----+
1 row in set (0.00 sec)

mysql> select count(*) from movie where year > 1998;
+-----+
| count(*) |
+-----+
|      418 |
+-----+
1 row in set (0.00 sec)
```

Figure 4: Comprobación de la exportación