

# Máster en Big Data

---

## Tecnologías de Almacenamiento

### 10. Hands-On: Importar datos con Sqoop

Presentado por: Jose David Angulo y Albert Ripoll

# Índice

1. Introducción3
2. Entorno3
3. Explorar los datos origen3
4. Uso de Sqoop4

# 1. Introducción

El objetivo de este Hands-On es ver un ejemplo de uso de Sqoop para importar datos de una base de datos MySQL

## 2. Entorno

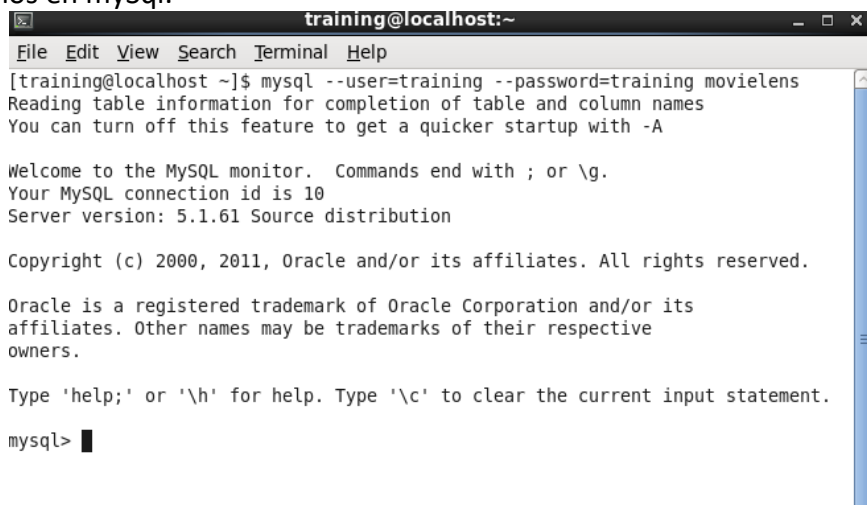
Para este Hands On, utilizaremos la máquina virtual desplegada en Hands-On anteriores llamada Developer\_Hadoop y todo será ejecutado vía shell

## 3. Explorar los datos origen

Lo primero que haremos es login en MySQL para ejecutar las instrucciones mediante Shell. Los datos que exploraremos son los relativos a base de datos movielens. Por todo ello nos conectaremos mediante el comando:

```
mysql --user=training --password=training movielens
```

Entramos en mySql.



```
training@localhost:~  
File Edit View Search Terminal Help  
[training@localhost ~]$ mysql --user=training --password=training movielens  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 10  
Server version: 5.1.61 Source distribution  
  
Copyright (c) 2000, 2011, 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>
```

- a) Revisa la estructura de la tabla movie y observa un ejemplo de datos de 5 registros.

Para revisar la estructura podemos usar

```
Descr movie;
```

O también

```
SHOW COLUMNS FROM movie;
```

ambos comandos son equivalentes

```
mysql> SHOW COLUMNS FROM movie;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	0	
name	char(75)	YES		NULL	
year	smallint(6)	YES		NULL	

3 rows in set (0.00 sec)

```
mysql> desc movie;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	0	
name	char(75)	YES		NULL	
year	smallint(6)	YES		NULL	

3 rows in set (0.00 sec)

Para observar un ejemplo de datos de 5 registros usamos:

```
SELECT * FROM movie limit 5;
```

```
mysql> SELECT * FROM movie limit 5;
```

id	name	year
1	Toy Story	1995
2	Jumanji	1995
3	Grumpier Old Men	1995
4	Waiting to Exhale	1995
5	Father of the Bride Part II	1995

5 rows in set (0.00 sec)

```
mysql> █
```

b) Revisa la estructura de la tabla movierating y observa un ejemplo de datos de 5 registros

Se puede usar

```
SHOW COLUMNS FROM movierating;
```

O también

```
desc movierating;
```

```
mysql> SHOW COLUMNS FROM movierating;
```

Field	Type	Null	Key	Default	Extra
userid	int(11)	NO	PRI	NULL	
movieid	int(11)	NO	PRI	NULL	
rating	tinyint(4)	NO		NULL	

3 rows in set (0.00 sec)

```
mysql> desc movierating;
```

Field	Type	Null	Key	Default	Extra
userid	int(11)	NO	PRI	NULL	
movieid	int(11)	NO	PRI	NULL	
rating	tinyint(4)	NO		NULL	

3 rows in set (0.00 sec)

Para observar un ejemplo de datos de 5 registros usamos:

```
select * from movierating limit 5;
```

```
mysql> select * from movierating limit 5;
```

userid	movieid	rating
1	1193	5
1	661	3
1	914	3
1	3408	4
1	2355	5

5 rows in set (0.01 sec)

```
mysql> █
```

## 4. Uso de Sqoop

### a) Muestra la ayuda de Sqoop

Para mostrar la ayuda de Sqoop, hemos salido de mysql y se volvió a ejecutar la terminal antes de mandar el siguiente comando.

#### Sqoop help

```
[training@localhost ~]$ sqoop help
usage: sqoop COMMAND [ARGS]
```

```
Available commands:
codegen          Generate code to interact with database records
create-hive-table Import a table definition into Hive
eval            Evaluate a SQL statement and display the results
export          Export an HDFS directory to a database table
help            List available commands
import          Import a table from a database to HDFS
import-all-tables Import tables from a database to HDFS
job             Work with saved jobs
list-databases  List available databases on a server
list-tables     List available tables in a database
merge          Merge results of incremental imports
metastore       Run a standalone Sqoop metastore
version         Display version information
```

See 'sqoop help COMMAND' for information on a specific command.

### b) Lista las bases de datos de MySQL desde Sqoop

```
sqoop list-databases --connect jdbc:mysql://localhost:3306/movielens --username training -
--password training
```

```
[training@localhost ~]$ sqoop list-databases --connect jdbc:mysql://localhost:3306/movielens --username training --password training
24/05/02 15:45:47 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
24/05/02 15:45:47 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
information_schema
dualcore
hue
metastore
movielens
mysql
test
training
[training@localhost ~]$ █
```

### c) Lista las tablas de la base de datos movielens

```
sqoop list-tables \
--connect jdbc:mysql://localhost:3306/movielens \
--username training \
--password training
```

```
[training@localhost ~]$ sqoop list-tables \
> --connect jdbc:mysql://localhost:3306/movielens \
> --username training \
> --password training
24/05/02 15:51:00 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
24/05/02 15:51:00 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
genre
movie
moviegenre
movierating
occupation
user
```

### d) Importar la tabla movie y movierating

- Con movie.

importamos sqoop  
conectamos a mysql al movilens con el usuario y password  
pedimos la tabla movie localizada en user/hadoop/movielens/movie  
y indicamos a sqoop que procese solo una sola tarea (un único mapper) con m1.  
El comando es el siguiente:

```
sqoop import \  
--connect jdbc:mysql://localhost:3306/movielens \  
--username training \  
--password training \  
--table movie \  
--target-dir /user/hadoop/movielens/movie \  
-m 1  
--fields-terminated-by ";"
```

```
[training@localhost ~]$ hdfs dfs -rm -r /user/hadoop/movielens/movie  
Deleted /user/hadoop/movielens/movie  
[training@localhost ~]$ sqoop import \  
> --connect jdbc:mysql://localhost:3306/movielens \  
> --username training \  
> --password training \  
> --table movie \  
> --target-dir /user/hadoop/movielens/movie \  
> --fields-terminated-by ";" \  
> -m 1
```

- Con movierating.

```
sqoop import \  
--connect jdbc:mysql://localhost:3306/movielens \  
--username training \  
--password training \  
--table movierating \  
--target-dir /user/hadoop/movielens/movierating \  
-m 1 \  
--fields-terminated-by ";"
```

```
[training@localhost ~]$ hdfs dfs -ls /user/hadoop/movielens/movie;
Found 3 items
-rw-r--r-- 1 training supergroup          0 2024-05-02 16:06 /user/hadoop/movielens/movie/_SUCCESS
drwxrwxrwx - training supergroup          0 2024-05-02 16:06 /user/hadoop/movielens/movie/_logs
-rw-r--r-- 1 training supergroup    102052 2024-05-02 16:06 /user/hadoop/movielens/movie/part-m-00000
[training@localhost ~]$ hdfs dfs -ls /user/hadoop/movielens/movierating
[training@localhost ~]$ hdfs dfs -ls /user/hadoop/movielens/movierating;
[training@localhost ~]$ hdfs dfs -ls /user/hadoop/movielens/movierating;
[training@localhost ~]$ sqoop import \
> --connect jdbc:mysql://localhost:3306/movielens \
> --username training \
> --password training \
> --table movierating \
> --target-dir /user/hadoop/movielens/movierating \
> -m 1 \
> --fields-terminated-by ";"
```

e) Lista los archivos importados. Según la nomenclatura de los archivos resultantes, ¿Que característica tiene el Job de Sqoop que se ha ejecutado?

- Con movie.

```
hdfs dfs -ls /user/hadoop/movielens/movie;
```

```
[training@localhost ~]$ hdfs dfs -ls /user/hadoop/movielens/movie;
Found 3 items
-rw-r--r-- 1 training supergroup          0 2024-05-02 16:06 /user/hadoop/movielens/movie/_SUCCESS
drwxrwxrwx - training supergroup          0 2024-05-02 16:06 /user/hadoop/movielens/movie/_logs
-rw-r--r-- 1 training supergroup    102052 2024-05-02 16:06 /user/hadoop/movielens/movie/part-m-00000
[training@localhost ~]$ █
```

- Con movierating.

```
hdfs dfs -ls /user/hadoop/movielens/movierating;
```

```
[training@localhost ~]$ hdfs dfs -ls /user/hadoop/movielens/movierating;
Found 3 items
-rw-r--r-- 1 training supergroup          0 2024-05-06 14:26 /user/hadoop/movielens/movierating/_SUCCESS
drwxrwxrwx - training supergroup          0 2024-05-06 14:26 /user/hadoop/movielens/movierating/_logs
-rw-r--r-- 1 training supergroup    11553408 2024-05-06 14:26 /user/hadoop/movielens/movierating/part-m-00000
[training@localhost ~]$ █
```

El sufijo de los archivos resultantes es `part-m-00000`. Esto indica que solo se ha generado una única partición de datos ya que se ha ejecutado un único mapper. Anteriormente al escribir `-m1` hemos usado un mapper.

Aparece 1 porque nosotros comandamos para 1 pero por defecto lo normal sería que fuesen 4.

f) Hacer un export de la misma tabla.

```
sqoop export \
--connect jdbc:mysql://localhost:3306/movielens \
--username training \
--password training \
--table movierating \
--export-dir /user/hadoop/movielens/movierating \
-m 1
```

```
[training@localhost ~]$ hadoop fs -ls /user/hadoop/movielens/movierating
Found 3 items
-rw-r--r-- 1 training supergroup          0 2024-05-06 14:26 /user/hadoop/movielens/movierating/_SUCCESS
drwxrwxrwx - training supergroup          0 2024-05-06 14:26 /user/hadoop/movielens/movierating/_logs
-rw-r--r-- 1 training supergroup 11553408 2024-05-06 14:26 /user/hadoop/movielens/movierating/part-m-00000
[training@localhost ~]$ sqoop export \
> --connect jdbc:mysql://localhost:3306/movielens \
> --username training \
> --password training \
> --table movierating \
> --export-dir /user/hadoop/movielens/movierating \
> -m 1
```

g) Mirar que este en la base de datos.

**mysql -u training -p movielens**

Enter password: training

```
[training@localhost ~]$ mysql -u training -p movielens
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.1.61 Source distribution

Copyright (c) 2000, 2011, 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.
```

- Con movie.

**SELECT \* FROM movie LIMIT 10;**

```
mysql> SELECT * FROM movie LIMIT 10;
+-----+-----+-----+
| id | name                                | year |
+-----+-----+-----+
| 1  | Toy Story                          | 1995 |
| 2  | Jumanji                           | 1995 |
| 3  | Grumpier Old Men                   | 1995 |
| 4  | Waiting to Exhale                  | 1995 |
| 5  | Father of the Bride Part II       | 1995 |
| 6  | Heat                               | 1995 |
| 7  | Sabrina                           | 1995 |
| 8  | Tom and Huck                       | 1995 |
| 9  | Sudden Death                      | 1995 |
| 10 | GoldenEye                         | 1995 |
+-----+-----+-----+
10 rows in set (0.00 sec)
```



- Con movierating.

**SELECT \* FROM movierating LIMIT 10;**

```
mysql> SELECT * FROM movierating LIMIT 10;
```

userid	movieid	rating
1	1193	5
1	661	3
1	914	3
1	3408	4
1	2355	5
1	1197	3
1	1287	5
1	2804	5
1	594	4
1	919	4

10 rows in set (0.00 sec)

h) Exportar los datos a una nueva base de datos en el servidor de MySQL.

**CREATE DATABASE albert;**

```
mysql> CREATE DATABASE albert;
```

Query OK, 1 row affected (0.00 sec)

**Use albert**

Des de allí especificamos que queremos usar la base de datos albert, antes de crear las tablas ya que si no especificamos use albert, vamos a crear las tablas en la base de datos que estemos, que en este caso seria movilens.

```
mysql> use albert
```

Database changed

Ahora creamos las tablas.

**CREATE TABLE albert\_table (id INT(11) NOT NULL PRIMARY KEY DEFAULT 0,name CHAR(75),year SMALLINT(6));**

```
mysql> CREATE TABLE albert_table (id INT(11) NOT NULL PRIMARY KEY DEFAULT 0,name CHAR(75),year SMALLINT(6));
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SHOW COLUMNS FROM albert_table;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	0	
name	char(75)	YES		NULL	
year	smallint(6)	YES		NULL	

3 rows in set (0.00 sec)

```
CREATE TABLE albert_table2 (
  userid INT(11) NOT NULL,
  movieid INT(11) NOT NULL,
  rating TINYINT(4) NOT NULL,
  PRIMARY KEY (userid, movieid)
);
```

```
mysql> CREATE TABLE albert_table2 (
  ->   userid INT(11) NOT NULL,
  ->   movieid INT(11) NOT NULL,
  ->   rating TINYINT(4) NOT NULL,
  ->   PRIMARY KEY (userid, movieid)
  -> );
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> SHOW COLUMNS FROM albert_table2;
```

Field	Type	Null	Key	Default	Extra
userid	int(11)	NO	PRI	NULL	
movieid	int(11)	NO	PRI	NULL	
rating	tinyint(4)	NO		NULL	

3 rows in set (0.00 sec)

i) Explorar las tablas de la nueva base de datos para comprobar que ha funcionado.

- Exportamos los datos de movie a la base de datos albert en la tabla albert\_table

```
sqoop export \
--connect jdbc:mysql://localhost:3306/albert\
--username training \
--password training \
--table albert_table \
--export-dir /user/hadoop/movielens/movie \
-m 1
```

```
sqoop export --connect jdbc:mysql://localhost:3306/albert--username training --password
training --table albert_table --export-dir /user/hadoop/movielens/movie
```

- Exportamos los datos de movierating a la base de datos albert en la tabla albert\_table2

```
sqoop export \  
--connect jdbc:mysql://localhost:3306/albert\  
--username training \  
--password training \  
--table albert_table2 \  
--export-dir /user/hadoop/movielens/movierating \  
-m 1
```

```
sqoop export --connect jdbc:mysql://localhost:3306/albert--username training --password  
training --table albert_table2 --export-dir /user/hadoop/movielens/movie
```