Ejercicios de Sqoop

Importar una tabla en MySQL a Hive

- **Creación de la tabla en MySQL**

1. Probamos la conexión con mysql

Contraseña (cloudera) facilitada por mi compañero Miguel Angel

```
[cloudera@quickstart ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
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.
```

2. Vemos las bbdd que contiene

Vemos las tablas que contiene:

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
+--------------------+
12 rows in set (0.03 sec)
```

3. Creamos en MYSQL la table que queremos importar en hive

```
mysql> CREATE DATABASE testdb;
Query OK, 1 row affected (0.00 sec)
```

4. Creamos una tabla con datos que luego importaremos a hive mediante sqoop

```
mysql> USE testdb;
Database changed
```

```
mysql> CREATE TABLE table_prueba (
    -> name VARCHAR(50),
    -> edad INT
    -> );
Query OK, 0 rows affected (0.01 sec)
```

5. comprobamos que se ha creado

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
| testdb             |
+--------------------+
13 rows in set (0.00 sec)
```

```
mysql> DESCRIBE table_prueba;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(50) | YES  |     | NULL    |       |
| edad  | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

6. Importamos algunas filas

```
mysql> INSERT INTO table_prueba (name, edad) VALUES ('John Doe', 30), ('Jane Smi
th', 25), ('Emily Davis', 22);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

7. Comprobamos que los datos se han insertado en la tabla

```
mysql> SELECT * FROM table_prueba;
+-------------+------+
| name        | edad |
+-------------+------+
| John Doe    |   30 |
| Jane Smith  |   25 |
| Emily Davis |   22 |
+-------------+------+
3 rows in set (0.00 sec)
```

- **<u>Creación de la tabla en Hive</u>**

Creamos la tabla en hive donde se importarán los datos que acabamos de crear

1. Accedemos a hive

```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
```

2. Creamos una base de datos para esta prueba y accedemos a ella

```
hive> CREATE DATABASE prueba_scoop;
OK
Time taken: 1.575 seconds
hive> USE prueba_scoop;
OK
Time taken: 0.064 seconds
```

3. Comprobamos que está en el warehouse de hive 1

Warehouse: lugar donde Hive guarda los datos de las tablas que se crean y administran a través de las consultas HiveQL.

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/hive/warehouse
Found 6 items
drwxrwxrwx   - cloudera supergroup          0 2024-07-11 10:52 /user/hive/wareho
use/ejerciciosmasterdb.db
drwxrwxrwx   - cloudera supergroup          0 2024-07-11 14:05 /user/hive/wareho
use/empleados
drwxrwxrwx   - cloudera supergroup          0 2024-07-12 09:04 /user/hive/wareho
use/pokemongeneration
drwxrwxrwx   - cloudera supergroup          0 2024-07-12 09:49 /user/hive/wareho
use/pokemongenerationtype
drwxrwxrwx   - cloudera supergroup          0 2024-07-12 09:00 /user/hive/wareho
use/pokemonstats
drwxrwxrwx   - cloudera supergroup          0 2024-07-15 13:45 /user/hive/wareho
use/prueba scoop.db
```

Se ve que esta el último.

4. Creamos la estructura de la table que contendrá los datos importados desde mysql con sqoop

```
hive> USE prueba_scoop;
OK
Time taken: 0.295 seconds
hive> CREATE TABLE IF NOT EXISTS table_prueba_scoop (
    > name STRING,
    > edad INT
    > );
OK
Time taken: 0.34 seconds
```

5. Comprobamos que se ha creado con éxito

```
hive> DESCRIBE table_prueba_scoop;
OK
col_name        data_type       comment
name                    string
edad                    int
Time taken: 0.102 seconds, Fetched: 2 row(s)
```

- **Importamos la tabla con SQOOP**

1. Dado que la "bbdd" Accumulo no está configurada, abrimos un Shell y ejecutamos los siguientes comandos para evitar warnings molestos.

Apache Accumulo es una base de datos distribuida y escalable, construida sobre Apache Hadoop y basada en los principios de diseño del sistema Bigtable de Google. Accumulo se desarrolló principalmente para el almacenamiento y la gestión eficiente de grandes volúmenes de datos, particularmente diseñada para aplicaciones que requieren almacenamiento y recuperación rápida de datos estructurados, como la indexación de texto completo, el procesamiento de gráficos y la gestión de registros de datos.

```
[cloudera@quickstart ~]$ sudo mkdir /var/lib/accumulo
[cloudera@quickstart ~]$ ACCUMULO_HOME='/var/lib/accumulo'
[cloudera@quickstart ~]$ export ACCUMULO_HOME
```

2. En un Shell escribimos lo siguiente para ver que sqoop está conectado con nuestro mysql:

Va bien con el siguiente comando:

sqoop list-databases --connect jdbc:mysql://localhost/testdb \
--username root --password cloudera

```
[cloudera@quickstart ~]$ sqoop list-databases --connect jdbc:mysql://localhost \
> --username root --password cloudera
24/07/15 14:09:28 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
24/07/15 14:09:28 WARN tool.BaseSqoopTool: Setting your password on the command-
line is insecure. Consider using -P instead.
24/07/15 14:09:29 INFO manager.MySQLManager: Preparing to use a MySQL streaming
resultset.
information_schema
cm
firehose
hue
metastore
mysql
nav
navms
oozie
retail_db
rman
sentry
testdb
```

Vemos que existe la conexión

3. Ahora listamos la tabla "table_prueba" de la bbdd "pruebadb" que hemos creado en MySQL

```
[cloudera@quickstart ~]$ sqoop list-databases --connect jdbc:mysql://localhost/t
estdb \
> --username root --password cloudera
24/07/15 14:15:48 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
24/07/15 14:15:48 WARN tool.BaseSqoopTool: Setting your password on the command-
line is insecure. Consider using -P instead.
24/07/15 14:15:48 INFO manager.MySQLManager: Preparing to use a MySQL streaming
resultset.
information_schema
cm
firehose
hue
metastore
mysql
nav
navms
oozie
retail_db
rman
sentry
testdb
```

4. Usando los argumentos de importación hive mostrados en las slides del curso, importar la tabla creada en Mysql en la estructura creada en hive. Usar como conector (jdbc:mysql://localhost/bbddMysql) y un solo mapper.

```
[cloudera@quickstart ~]$ sqoop import \
> --connect jdbc:mysql://localhost/testdb \
> --username root \
> --password cloudera \
> --table table_prueba \
> --hive-import \
> --hive-overwrite \
> --hive-table prueba_scoop.table_prueba_scoop \
> -m 1
```

aqui pillamos la tabla a importar

nombre de la bbdd . nombre de la tabla
numero de mappers a utilizar

```
Table prueba_scoop.table_prueba_scoop stats: [numFiles=1, numRows=0, totalSize=4
1, rawDataSize=0]
OK
Time taken: 0.804 seconds
```

Comprobamos en hive:

```
hive> SELECT * FROM table_prueba_scoop;
OK
table_prueba_scoop.name table_prueba_scoop.edad
John Doe        30
Jane Smith      25
Emily Davis     22
Time taken: 0.372 seconds, Fetched: 3 row(s)
```

Se puede ver que la inserción ha ido bien.