iesgrancapitan-CEIABD-BDA / **ud5-practica-1-hadoop-hive-Dansara…**

Code Issues Pull requests Actions Projects Wiki Security

main ud5-practica-1-hadoop-hive-Dansarasix-DML / **Readme.md**

Dansarasix-DML update1 b1c12f0 · 5 months ago

`200 lines (127 loc) · 6.26 KB`

Preview Code Blame Raw

Big Data Aplicado

UD 5 - Apache Hadoop - Hive

Práctica 1 Hive - *(RA5075.1 / CE.1d)*

> Para los siguientes ejercicios, copia el comando y/o haz una captura de pantalla donde se muestre el resultado de cada acción
>
> Debes entregar los correspondientes comandos y capturas.
>
> Recuerda que tienes que tener correctamente configurado Apache Hadoop (HDFS, MapReduce, Yarn y Tez). Si no es así, consulta la documentación del módulo
>
> **RECOMENDACIÓN**: Clona las máquinas virtuales donde estás trabajando la asignatura para realizar esta tarea. También puedes hacerla usando la máquina pseudo distribuida correctamente configurada

1. La práctica consiste en utilizar otro sistema de Base de Datos diferente al mostrado en la documentación del módulo (`derby`).

2. Las diferentes opciones son las siguientes (las puedes consultar en la documentación oficial de Hive)

```
derby|mysql|postgres|oracle|mssql
```

3. Tienes que configurar también el acceso de usuarios a la `dbType` que elijas.

4. Realiza alguna consulta y muestra los resultados tanto en terminal con `beeline` como la WebUI de `hiveserver2`

5. Puedes ayudarte del siguiente recurso indicado en nuestra documentación oficial

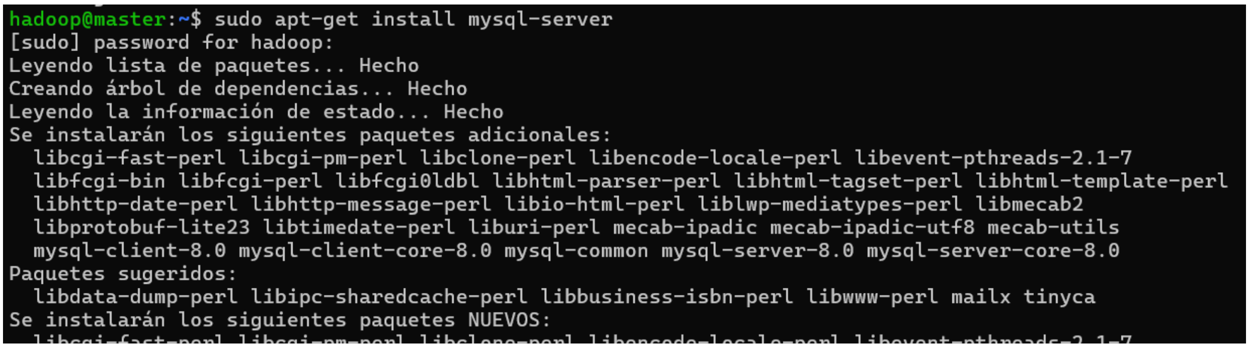
1. Introducción

En esta práctica se instalará otro sistema de bases de datos que no sea `derby` . En mi caso será `MySQL` .

2. Instalación

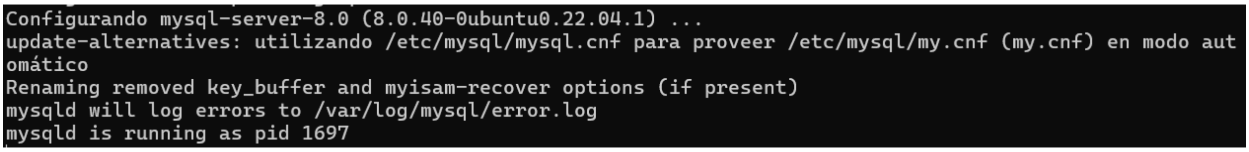
Primero debemos instalar `mysql-server` .

```
sudo apt-get install mysql-server
```



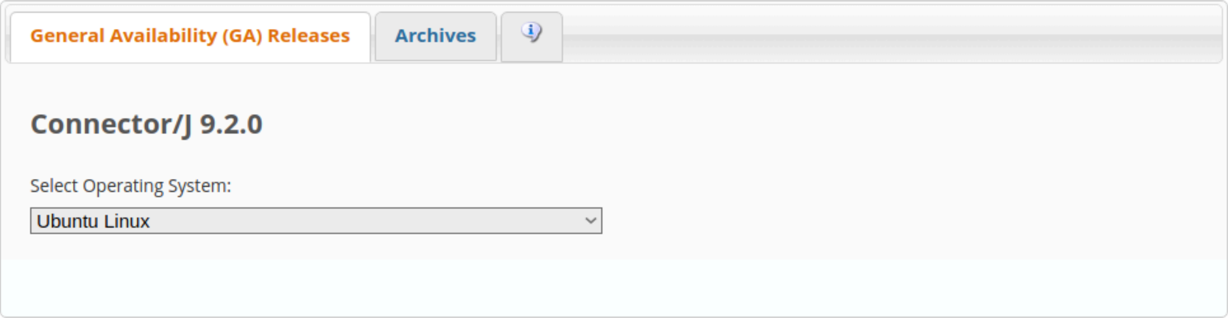
```
hadoop@master:~$ sudo apt-get install mysql-server
[sudo] password for hadoop:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7
  libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2
  libprotobuf-lite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Paquetes sugeridos:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
Se instalarán los siguientes paquetes NUEVOS:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7
```

Vemos que tras su instalación, se ejecuta automáticamente.



```
Configurando mysql-server-8.0 (8.0.40-0ubuntu0.22.04.1) ...
update-alternatives: utilizando /etc/mysql/mysql.cnf para proveer /etc/mysql/my.cnf (my.cnf) en modo aut
omático
Renaming removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 1697
```

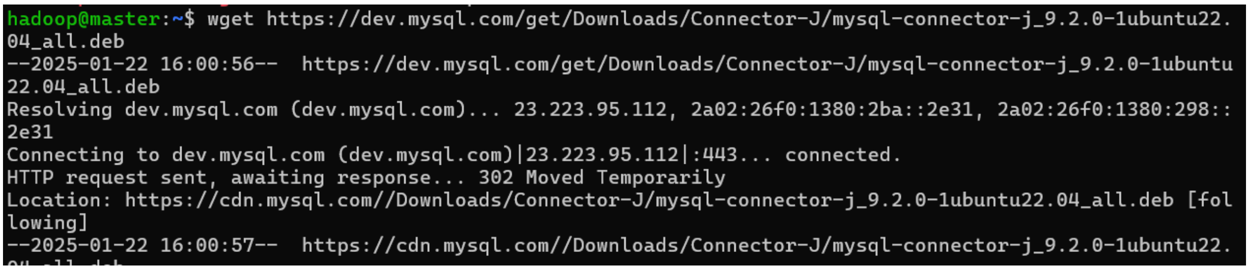
Lo siguiente será instalar el conector de `JAVA` para `MySQL` . Sin embargo, ya no se puede instalar con `sudo apt` si no que primero lo descargamos de su página y luego lo instalamos. He usado la versión 9.2.0.



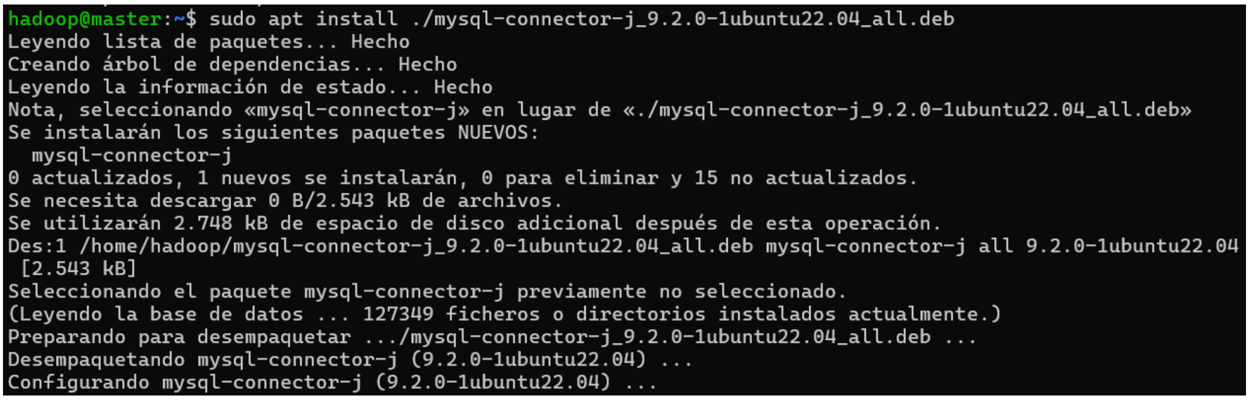
Connector/J 9.2.0

Select Operating System: Ubuntu Linux

```
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j_9.2.0-1ubuntu22.04_all.deb

sudo apt install ./mysql-connector-j_9.2.0-1ubuntu22.04_all.deb
```



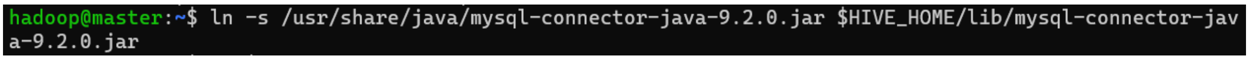
```
hadoop@master:~$ wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j_9.2.0-1ubuntu22.
04_all.deb
--2025-01-22 16:00:56--  https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j_9.2.0-1ubuntu
22.04_all.deb
Resolving dev.mysql.com (dev.mysql.com)... 23.223.95.112, 2a02:26f0:1380:2ba::2e31, 2a02:26f0:1380:298::
2e31
Connecting to dev.mysql.com (dev.mysql.com)|23.223.95.112|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-j_9.2.0-1ubuntu22.04_all.deb [fol
lowing]
--2025-01-22 16:00:57--  https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-j_9.2.0-1ubuntu22.
```



```
hadoop@master:~$ sudo apt install ./mysql-connector-j_9.2.0-1ubuntu22.04_all.deb
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Nota, seleccionando «mysql-connector-j» en lugar de «./mysql-connector-j_9.2.0-1ubuntu22.04_all.deb»
Se instalarán los siguientes paquetes NUEVOS:
  mysql-connector-j
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 15 no actualizados.
Se necesita descargar 0 B/2.543 kB de archivos.
Se utilizarán 2.748 kB de espacio de disco adicional después de esta operación.
Des:1 /home/hadoop/mysql-connector-j_9.2.0-1ubuntu22.04_all.deb mysql-connector-j all 9.2.0-1ubuntu22.04
 [2.543 kB]
Seleccionando el paquete mysql-connector-j previamente no seleccionado.
(Leyendo la base de datos ... 127349 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../mysql-connector-j_9.2.0-1ubuntu22.04_all.deb ...
Desempaquetando mysql-connector-j (9.2.0-1ubuntu22.04) ...
Configurando mysql-connector-j (9.2.0-1ubuntu22.04) ...
```

Por último, hacemos un enlace simbólico al `.jar` a `$HIVE_HOME/lib/`.

```
ln -s /usr/share/java/mysql-connector-java-9.2.0.jar $HIVE_HOME/lib/mysql-connector-java-9.2.0.jar
```

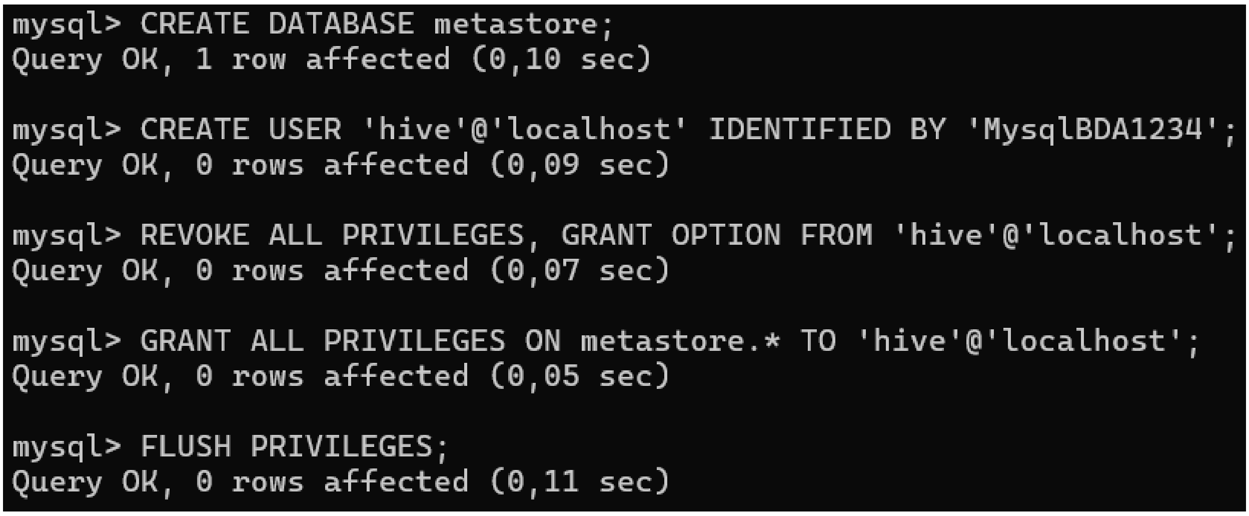


```
hadoop@master:~$ ln -s /usr/share/java/mysql-connector-java-9.2.0.jar $HIVE_HOME/lib/mysql-connector-jav
a-9.2.0.jar
```

3. Configuración

Lo siguiente será configurar un usuario en `MySQL` que `HIVE` usará para conectarse a su base de datos. Previavemente crearemos su base de datos para darle todos los permisos a dicho usuario.

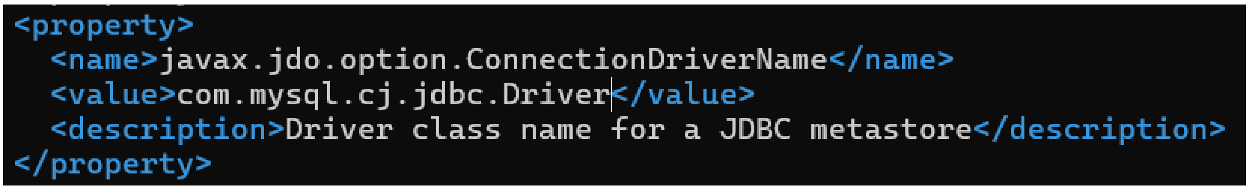
```
CREATE DATABASE metastore;

CREATE USER 'hive'@'localhost' IDENTIFIED BY 'MysqlBDA1234';

REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'hive'@'localhost';

GRANT ALL PRIVILEGES ON metastore.* TO 'hive'@'localhost';

FLUSH PRIVILEGES;
```



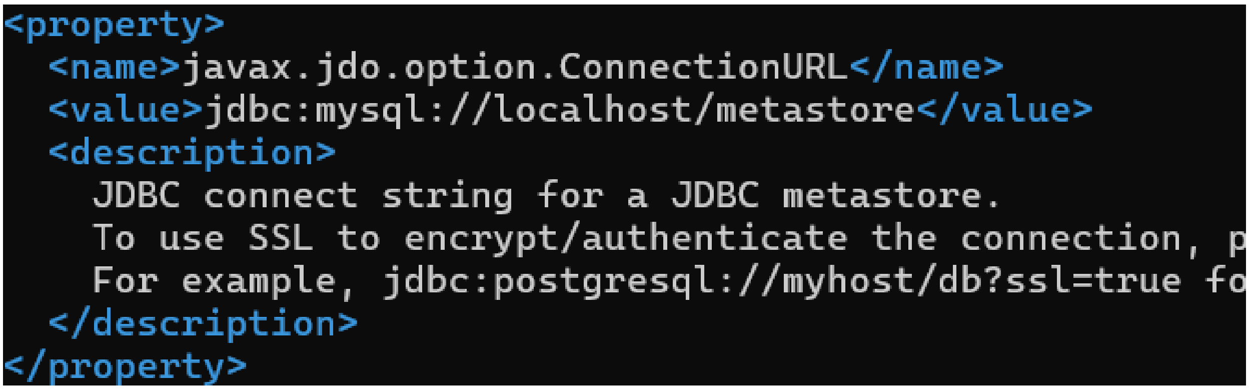
```
mysql> CREATE DATABASE metastore;
Query OK, 1 row affected (0,10 sec)

mysql> CREATE USER 'hive'@'localhost' IDENTIFIED BY 'MysqlBDA1234';
Query OK, 0 rows affected (0,09 sec)

mysql> REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'hive'@'localhost';
Query OK, 0 rows affected (0,07 sec)

mysql> GRANT ALL PRIVILEGES ON metastore.* TO 'hive'@'localhost';
Query OK, 0 rows affected (0,05 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,11 sec)
```

Luego iremos al archivo `hive-site.xml` y modificaremos las siguientes propiedades:

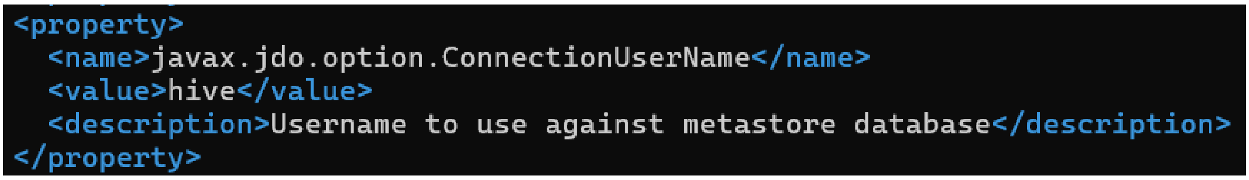
```
<property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true</
    <description>JDBC connect string for a JDBC metastore</description>
</property>

 <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <description>Driver class name for a JDBC metastore</description>
 </property>

 <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hive</value>
    <description>username to use against metastore database</description>
 </property>

 <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>MysqlBDA1234</value>
    <description>password to use against metastore database</description>
 </property>

 <property>
    <name>hive.metastore.uris</name>
    <value>thrift://localhost:9083</value>
 </property>
```



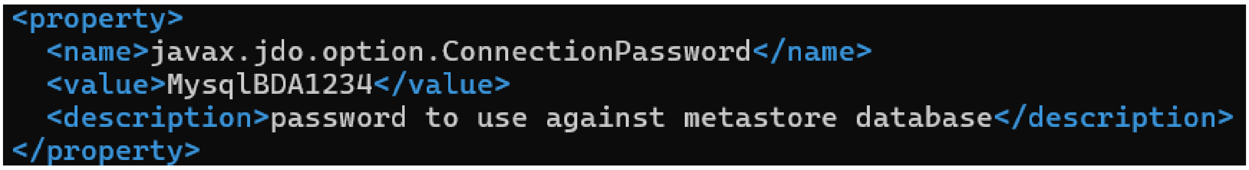
```
<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>com.mysql.cj.jdbc.Driver</value>
  <description>Driver class name for a JDBC metastore</description>
</property>
```



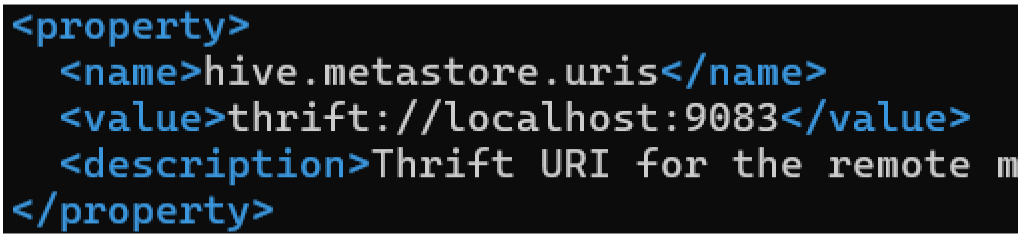
```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://localhost/metastore</value>
  <description>
    JDBC connect string for a JDBC metastore.
    To use SSL to encrypt/authenticate the connection, p
    For example, jdbc:postgresql://myhost/db?ssl=true fo
  </description>
</property>
```



```
<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>hive</value>
  <description>Username to use against metastore database</description>
</property>
```



```
<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>MysqlBDA1234</value>
  <description>password to use against metastore database</description>
</property>
```



```
<property>
  <name>hive.metastore.uris</name>
  <value>thrift://localhost:9083</value>
  <description>Thrift URI for the remote m
</property>
```

Debemos tener en cuenta la advertencia:

> ⓘ **IMPORTANT**
>
> For MySQL 8, set the `javax.jdo.option.ConnectionDriverName` property to `com.mysql.cj.jdbc.Driver`. The `com.mysql.jdbc.Driver` is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver`. However, the driver is automatically registered via the Service Provider Interface, so manual loading of the driver class is generally unnecessary.

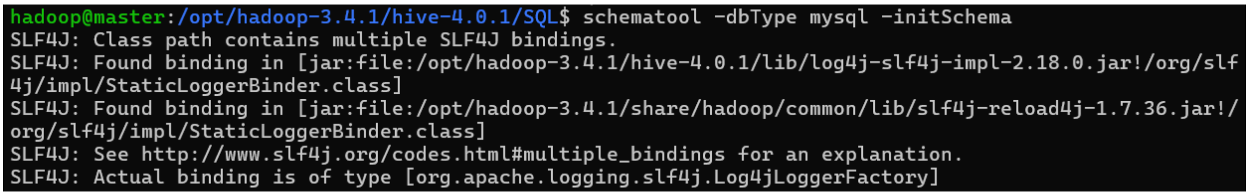
El puero servirá para que `MySQL` escuche por dicho puerto mientras se realizan las consultas por `HIVE` .

Una vez hemos guardado la configuración. Paramos los servicios y los iniciamos de nuevo.

```
stop-all.sh
start-dfs.sh
start-yarn.sh
```

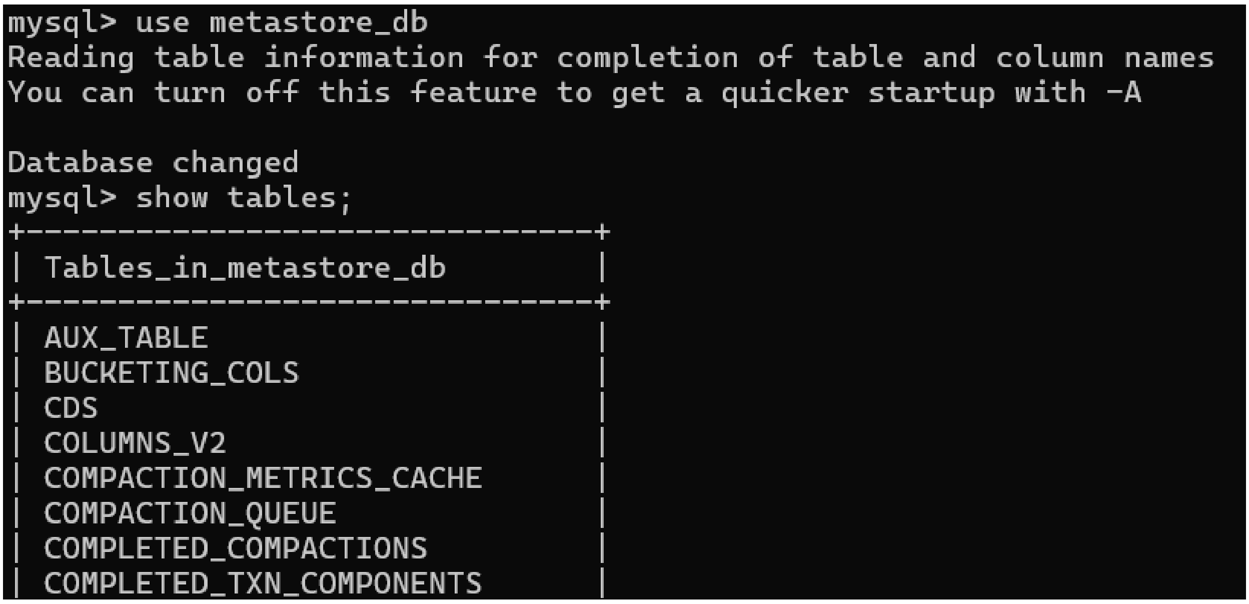
Por último, lanzamos el `schematool` pero con `MySQL` .

```
schematool -dbType mysql -initSchema
```



```
hadoop@master:/opt/hadoop-3.4.1/hive-4.0.1/SQL$ schematool -dbType mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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.apache.logging.slf4j.Log4jLoggerFactory]
```

Comprobamos que en `MySQL` la base de datos ha cambiado.

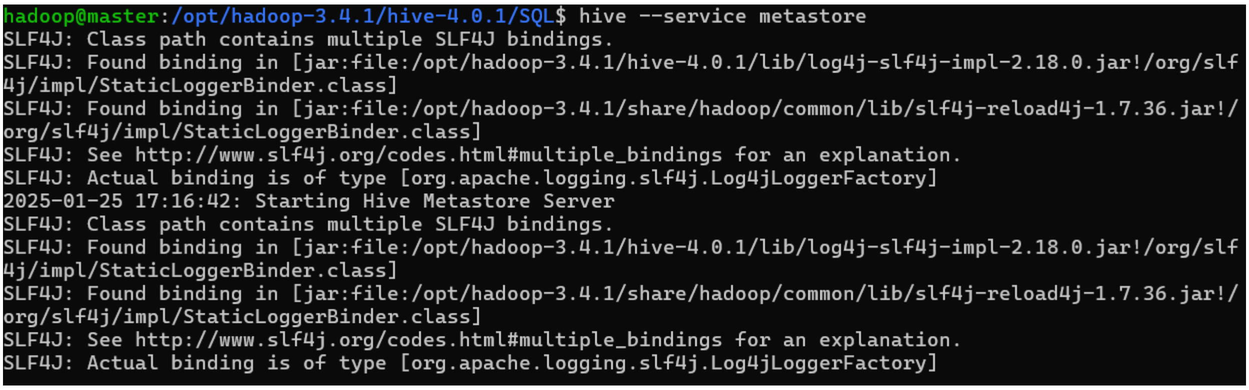


```
mysql> use metastore_db
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------------------+
| Tables_in_metastore_db        |
+-------------------------------+
| AUX_TABLE                     |
| BUCKETING_COLS                |
| CDS                           |
| COLUMNS_V2                    |
| COMPACTION_METRICS_CACHE      |
| COMPACTION_QUEUE              |
| COMPLETED_COMPACTIONS         |
| COMPLETED_TXN_COMPONENTS      |
```

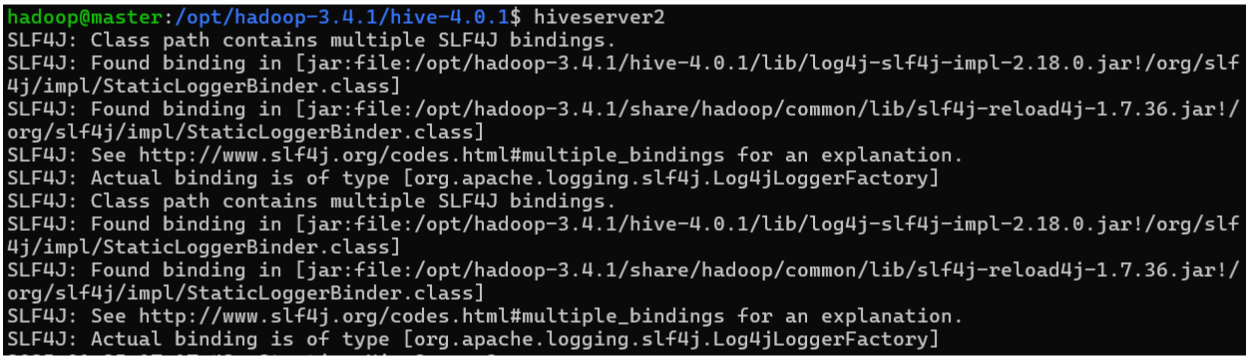
4. Ejecución

Para ejecutar `HIVE` primero debemos ejectar el `metastore` de `MySQL` .

```
hive --service metastore

hiveserver2
```



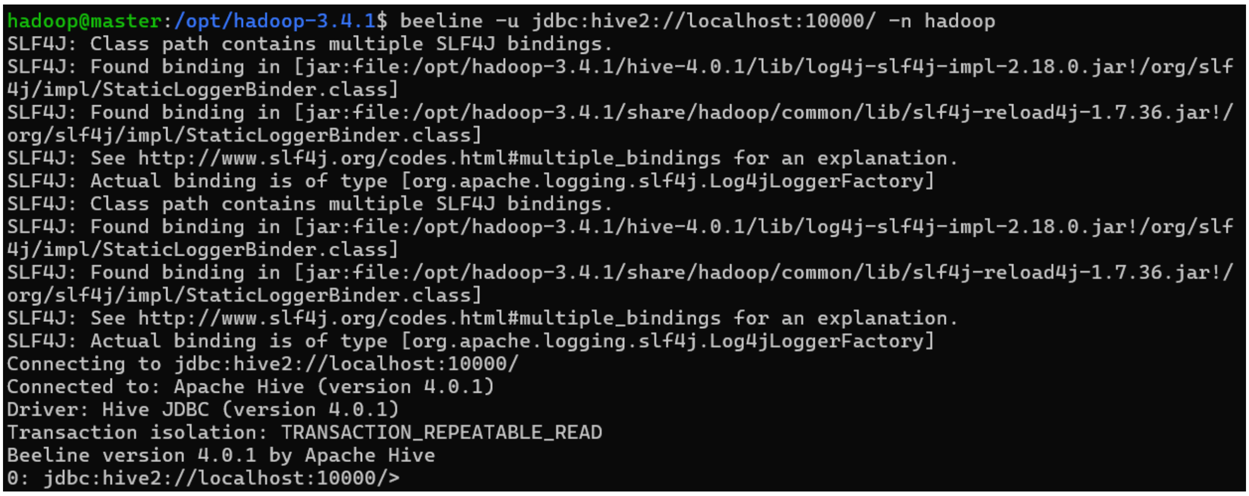
```
hadoop@master:/opt/hadoop-3.4.1/hive-4.0.1/SQL$ hive --service metastore
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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.apache.logging.slf4j.Log4jLoggerFactory]
2025-01-25 17:16:42: Starting Hive Metastore Server
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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.apache.logging.slf4j.Log4jLoggerFactory]
```



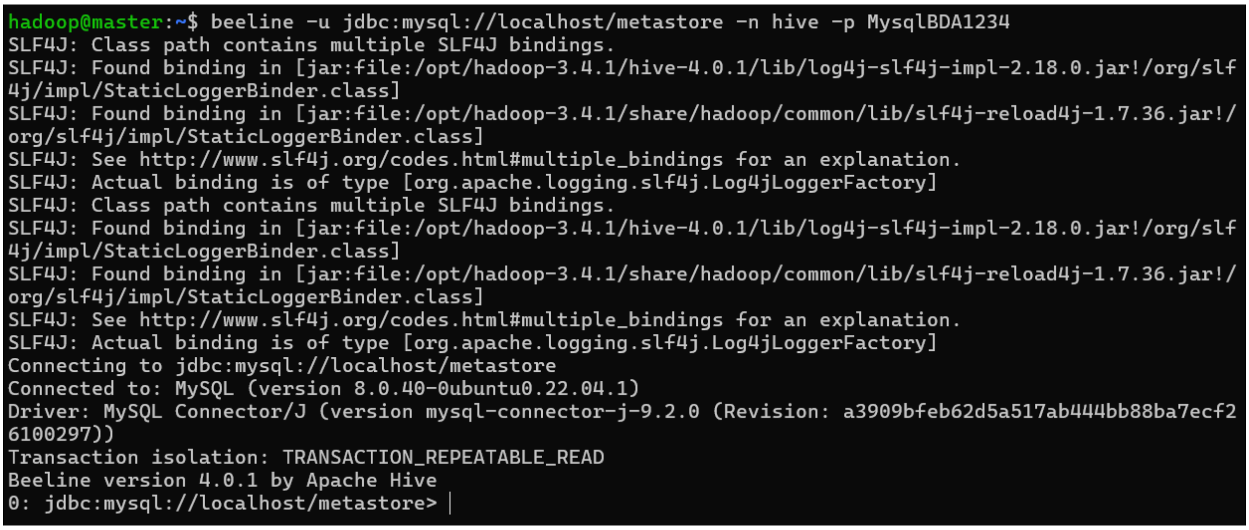
```
hadoop@master:/opt/hadoop-3.4.1/hive-4.0.1$ hiveserver2
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/
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.apache.logging.slf4j.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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.apache.logging.slf4j.Log4jLoggerFactory]
```

Una vez se han lanzado los dos servicios **por separado**, entramos en `beeline`. Podemos entrar tanto en `HIVE` como en `MySQL`.

```
beeline -u jdbc:hive2://localhost:10000/ -n hadoop # Acceso a HIVE

beeline -u jdbc:mysql://localhost/metastore -n hive -p MysqlBDA1234 # Acceso a MySQL
```

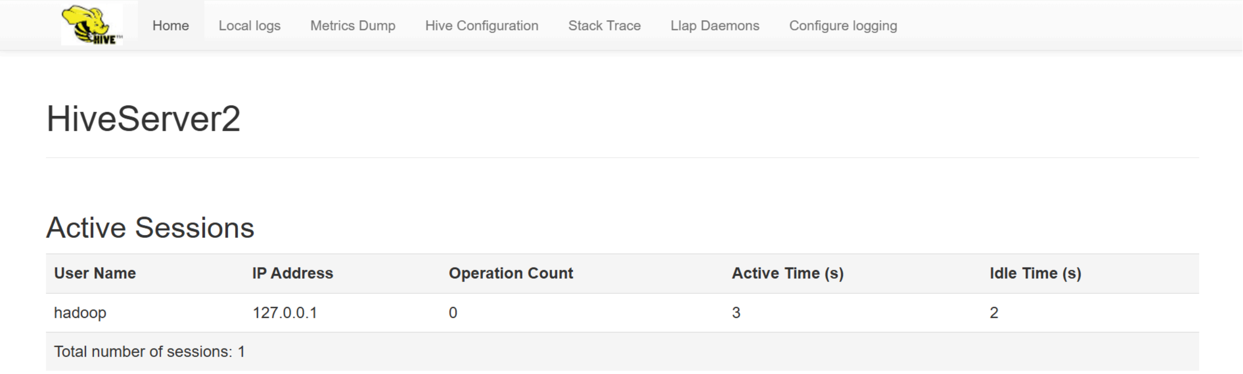


```
hadoop@master:/opt/hadoop-3.4.1$ beeline -u jdbc:hive2://localhost:10000/ -n hadoop
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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.apache.logging.slf4j.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000/
Connected to: Apache Hive (version 4.0.1)
Driver: Hive JDBC (version 4.0.1)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 4.0.1 by Apache Hive
0: jdbc:hive2://localhost:10000/>
```



```
hadoop@master:~$ beeline -u jdbc:mysql://localhost/metastore -n hive -p MysqlBDA1234
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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.apache.logging.slf4j.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/hive-4.0.1/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-3.4.1/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:mysql://localhost/metastore
Connected to: MySQL (version 8.0.40-0ubuntu0.22.04.1)
Driver: MySQL Connector/J (version mysql-connector-j-9.2.0 (Revision: a3909bfeb62d5a517ab444bb88ba7ecf2
6100297))
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 4.0.1 by Apache Hive
0: jdbc:mysql://localhost/metastore>
```

También podemos acceder desde el navegador por el puerto `10002`.



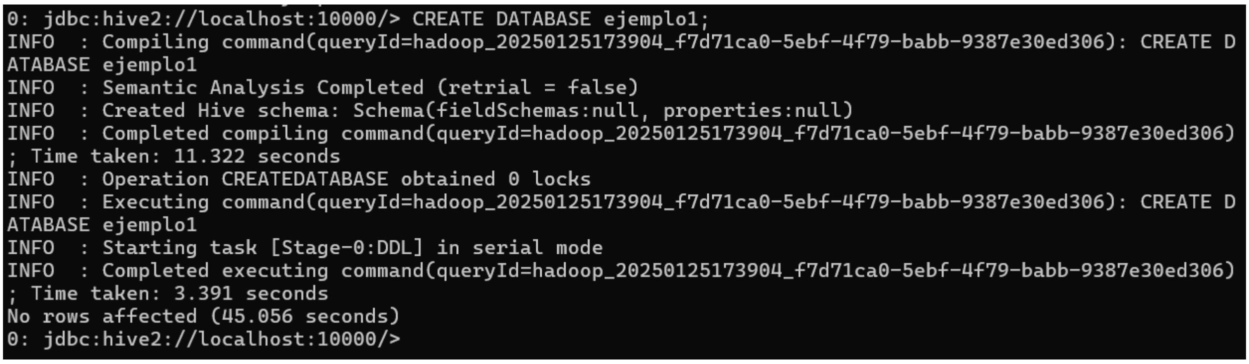
HiveServer2

Active Sessions

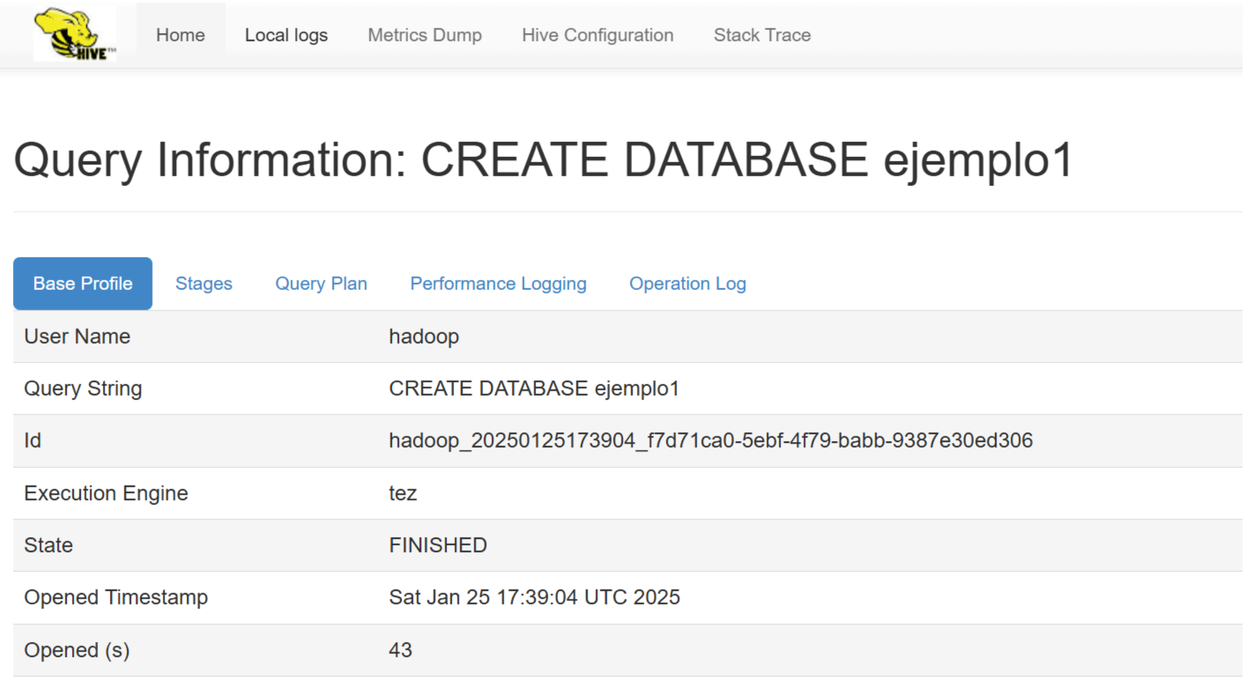
User Name	IP Address	Operation Count	Active Time (s)	Idle Time (s)
hadoop	127.0.0.1	0	3	2
Total number of sessions: 1				

5. Ejemplos

Ahora que tenemos `HIVE` unido a `MySQL`, aquí se muestran unos ejemplos realizados.



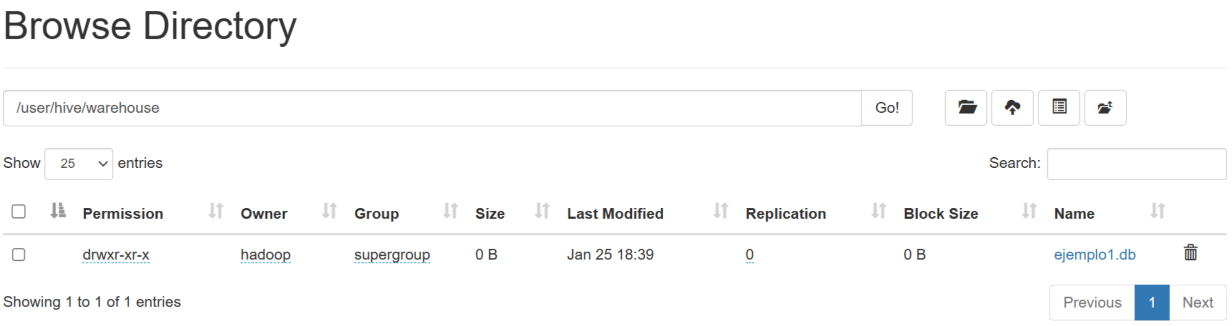
```
0: jdbc:hive2://localhost:10000/> CREATE DATABASE ejemplo1;
INFO  : Compiling command(queryId=hadoop_20250125173904_f7d71ca0-5ebf-4f79-babb-9387e30ed306): CREATE D
ATABASE ejemplo1
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hadoop_20250125173904_f7d71ca0-5ebf-4f79-babb-9387e30ed306)
; Time taken: 11.322 seconds
INFO  : Operation CREATEDATABASE obtained 0 locks
INFO  : Executing command(queryId=hadoop_20250125173904_f7d71ca0-5ebf-4f79-babb-9387e30ed306): CREATE D
ATABASE ejemplo1
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hadoop_20250125173904_f7d71ca0-5ebf-4f79-babb-9387e30ed306)
; Time taken: 3.391 seconds
No rows affected (45.056 seconds)
0: jdbc:hive2://localhost:10000/>
```



Query Information: CREATE DATABASE ejemplo1

User Name	hadoop
Query String	CREATE DATABASE ejemplo1
Id	hadoop_20250125173904_f7d71ca0-5ebf-4f79-babb-9387e30ed306
Execution Engine	tez
State	FINISHED
Opened Timestamp	Sat Jan 25 17:39:04 UTC 2025
Opened (s)	43

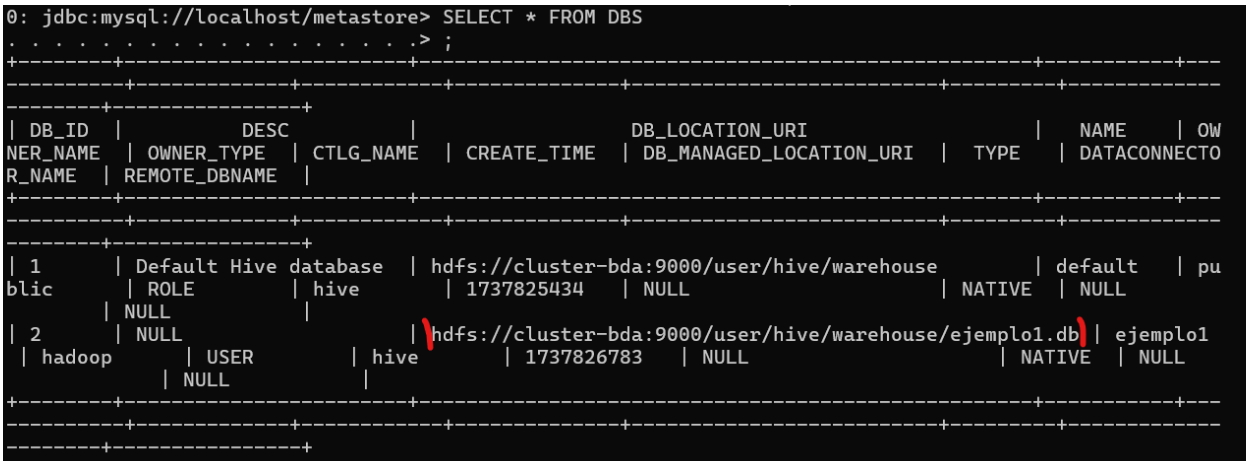
Aquí vemos que se guarda en `HDFS` también.



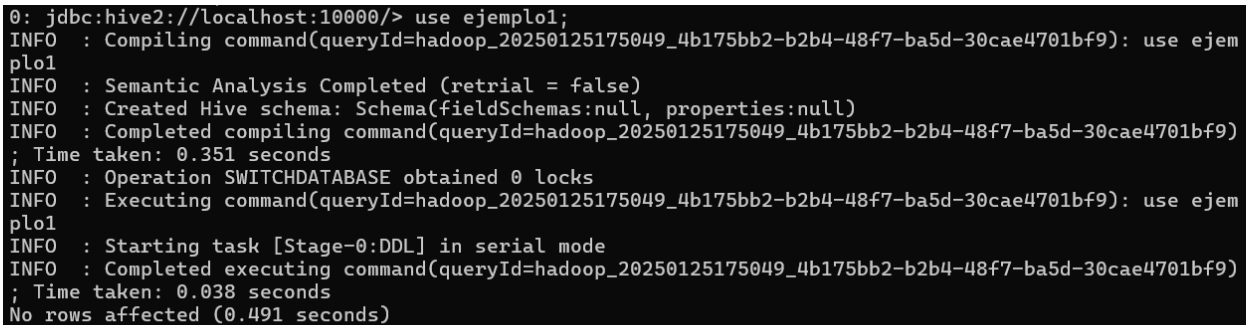
☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	drwxr-xr-x	hadoop	supergroup	0 B	Jan 25 18:39	0	0 B	ejemplo1.db

Showing 1 to 1 of 1 entries

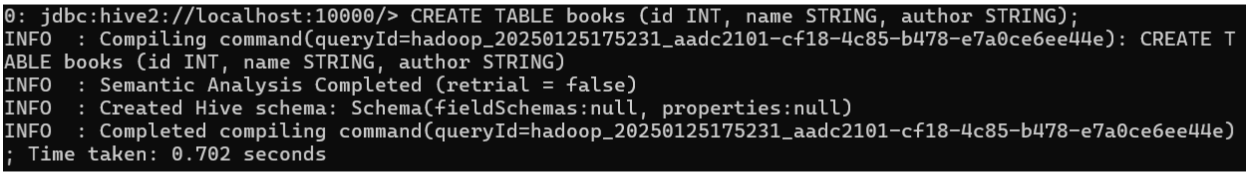
La base de datos de `MySQL` guardará en las distintas tablas las consultas realizadas.



```
0: jdbc:mysql://localhost/metastore> SELECT * FROM DBS
. . . . . . . . . . . . . . . . . .> ;
+--------+-----------------------+----------------------------------------------------+----------+---
---------+-------------+------------+--------------+--------------------------+---------+------------
--------+----------------+
| DB_ID  |         DESC          |                  DB_LOCATION_URI                   |   NAME   | OW
NER_NAME  | OWNER_TYPE  | CTLG_NAME  | CREATE_TIME  | DB_MANAGED_LOCATION_URI  |  TYPE   | DATACONNECTO
R_NAME  | REMOTE_DBNAME  |
+--------+-----------------------+----------------------------------------------------+----------+---
---------+-------------+------------+--------------+--------------------------+---------+------------
--------+----------------+
| 1      | Default Hive database | hdfs://cluster-bda:9000/user/hive/warehouse        | default  | pu
blic      | ROLE        | hive       | 1737825434   | NULL                     | NATIVE  | NULL
         | NULL           |
| 2      | NULL                  | hdfs://cluster-bda:9000/user/hive/warehouse/ejemplo1.db | ejemplo1
 | hadoop      | USER          | hive         | 1737826783    | NULL                        | NATIVE  | NULL
          | NULL           |
+--------+-----------------------+----------------------------------------------------+----------+---
---------+-------------+------------+--------------+--------------------------+---------+------------
--------+----------------+
```



```
0: jdbc:hive2://localhost:10000/> use ejemplo1;
INFO  : Compiling command(queryId=hadoop_20250125175049_4b175bb2-b2b4-48f7-ba5d-30cae4701bf9): use ejem
plo1
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hadoop_20250125175049_4b175bb2-b2b4-48f7-ba5d-30cae4701bf9)
; Time taken: 0.351 seconds
INFO  : Operation SWITCHDATABASE obtained 0 locks
INFO  : Executing command(queryId=hadoop_20250125175049_4b175bb2-b2b4-48f7-ba5d-30cae4701bf9): use ejem
plo1
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hadoop_20250125175049_4b175bb2-b2b4-48f7-ba5d-30cae4701bf9)
; Time taken: 0.038 seconds
No rows affected (0.491 seconds)
```



```
0: jdbc:hive2://localhost:10000/> CREATE TABLE books (id INT, name STRING, author STRING);
INFO  : Compiling command(queryId=hadoop_20250125175231_aadc2101-cf18-4c85-b478-e7a0ce6ee44e): CREATE T
ABLE books (id INT, name STRING, author STRING)
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hadoop_20250125175231_aadc2101-cf18-4c85-b478-e7a0ce6ee44e)
; Time taken: 0.702 seconds
```