

Curso de Especialización en  
Inteligencia Artificial y Big Data

# Big Data Aplicado

0306\_Ejemplo completo:  
HDFS, Sqoop, HUE, Hive en  
Cloudera

Javier Rojas

---

## Uso de la máquina virtual.

Descargamos e instalamos VirtualBox:

<https://www.virtualbox.org/>

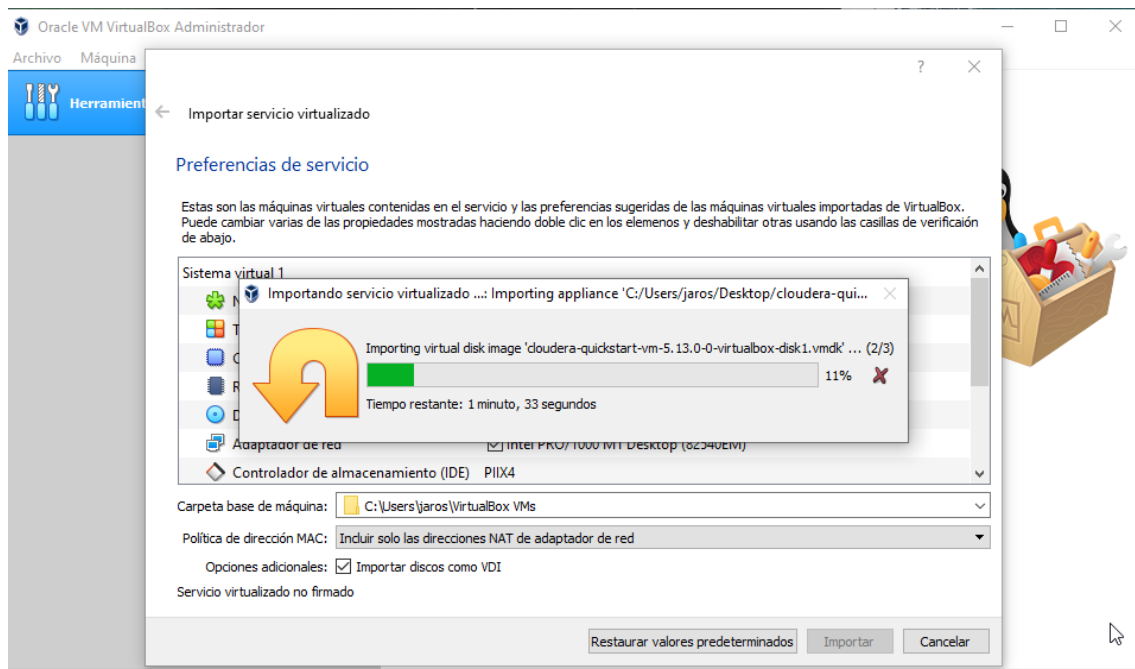
Descargamos la máquina virtual del repositorio de Cloudera:

[https://downloads.cloudera.com/demo\\_vm/virtualbox/cloudera-quickstart-vm-5.13.0-0-virtualbox.zip](https://downloads.cloudera.com/demo_vm/virtualbox/cloudera-quickstart-vm-5.13.0-0-virtualbox.zip)

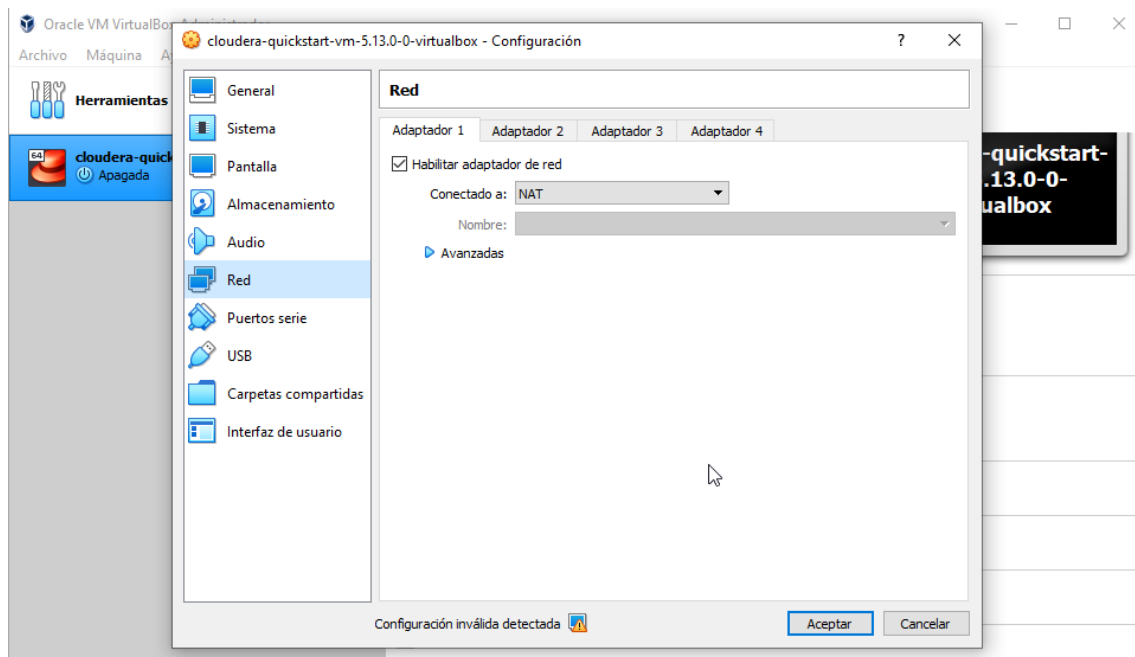
O de la carpeta compartida del módulo:

[https://drive.google.com/file/d/1PKAda9oKkbaOdVieisU62BrpsdV9dizU/view?usp=drive\\_link](https://drive.google.com/file/d/1PKAda9oKkbaOdVieisU62BrpsdV9dizU/view?usp=drive_link)

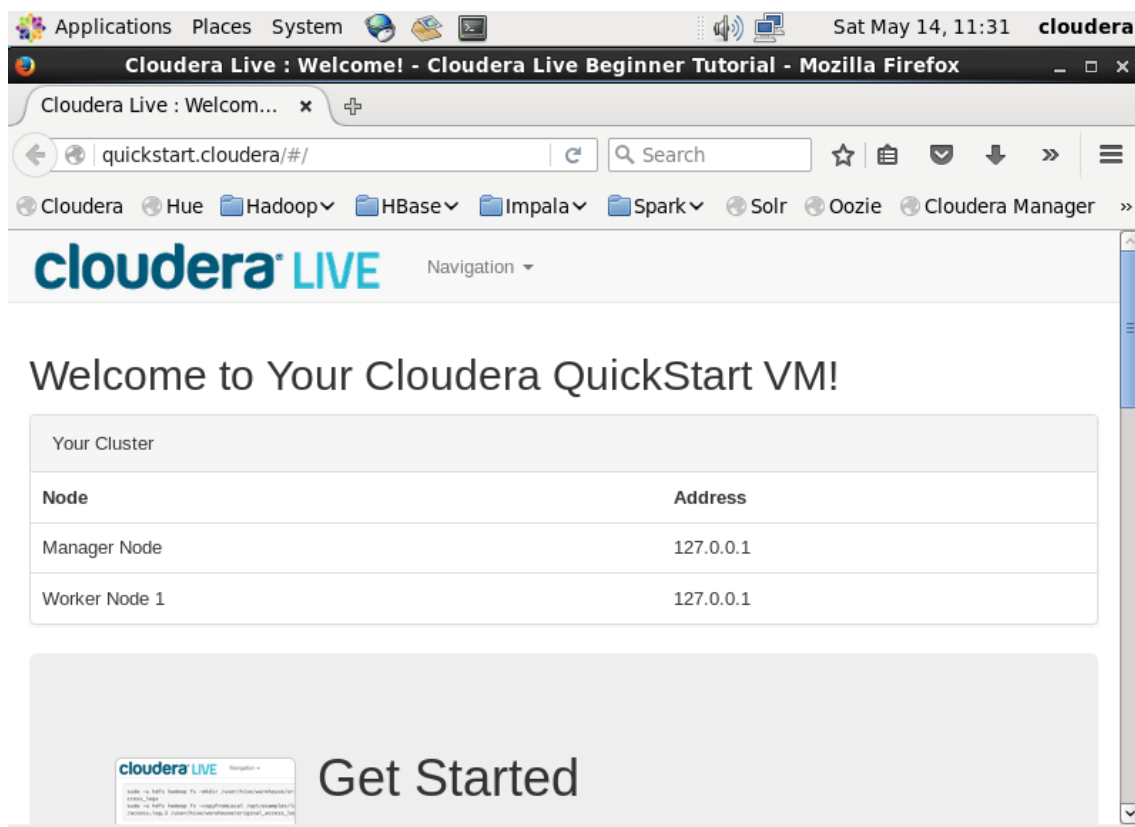
Y la importamos a VirtualBox:



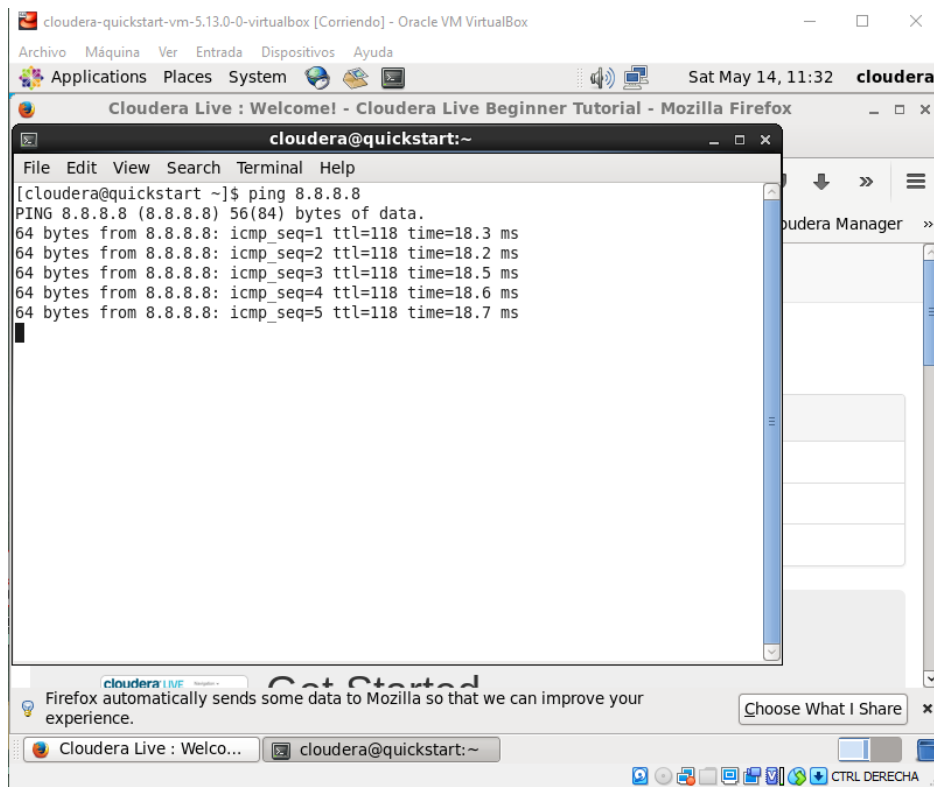
Configuramos la red:



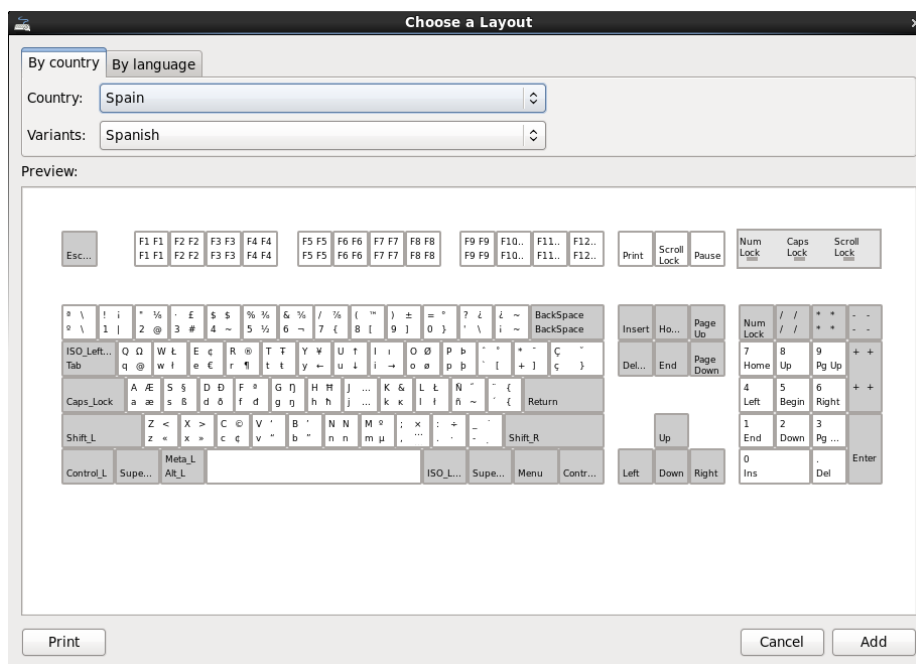
Al arrancar aparece el quikstart de Cloudera:



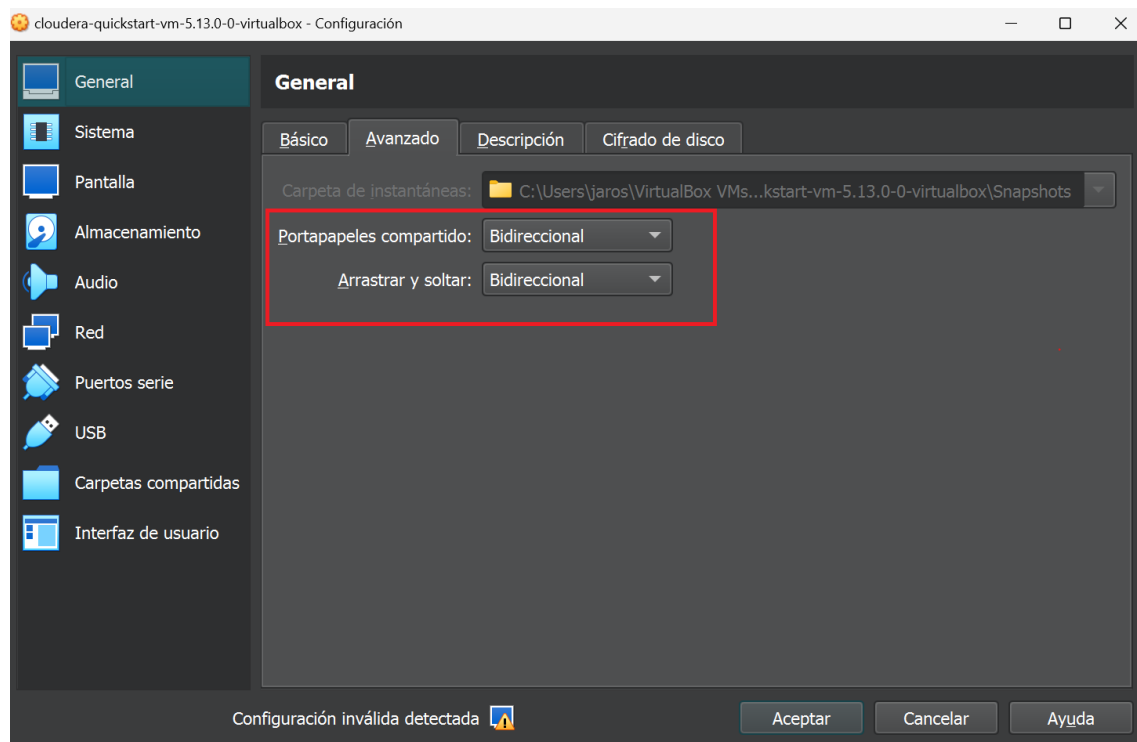
Comprobamos que tenemos red:



Cambiamos el layout del teclado a español:



Es recomendable activar el portapapeles compartido para facilitar copiar y pegar entre la máquina y el anfitrión.



Ya estamos listos para empezar.

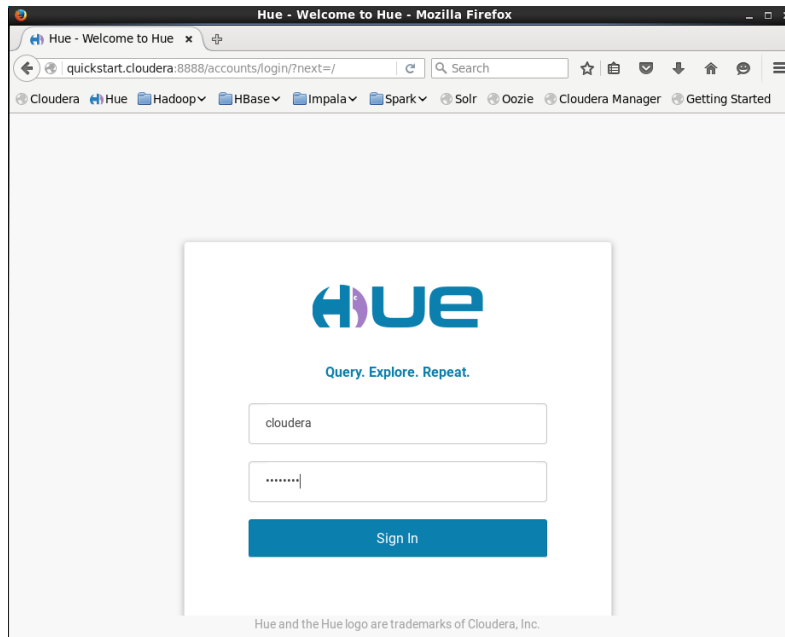
## Ingesta y consulta de datos relacionales.

Importar todas las tablas de la base de datos MySQL retail\_dba Hive.

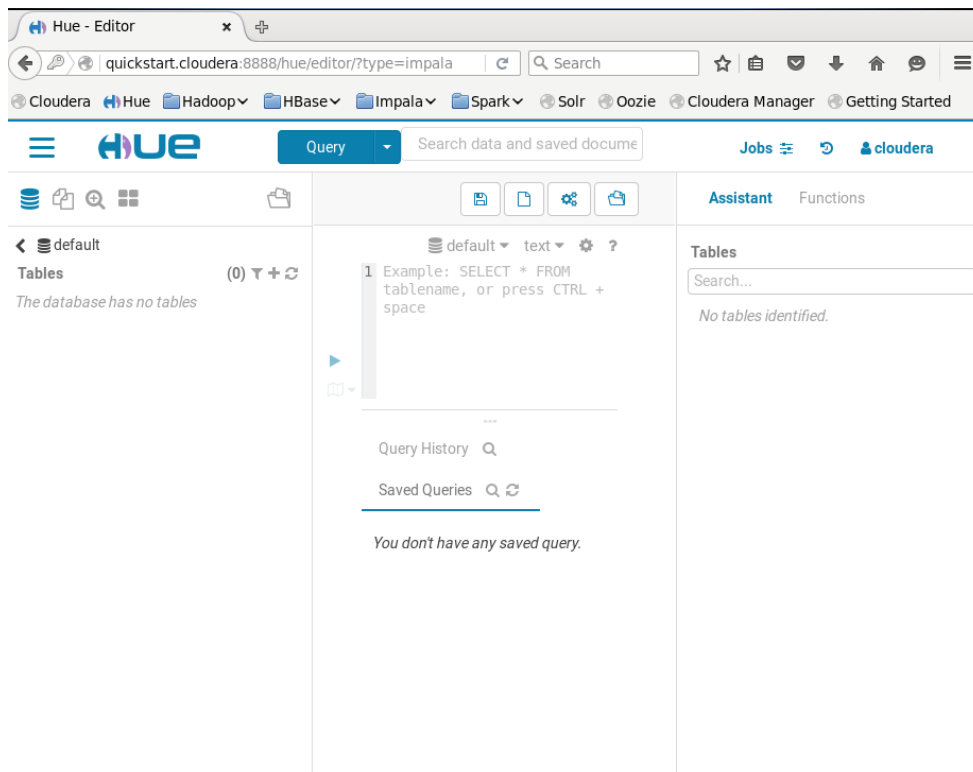
Accedemos a HUE-Hive e introducimos:

Usuario: cloudera

Password: cloudera

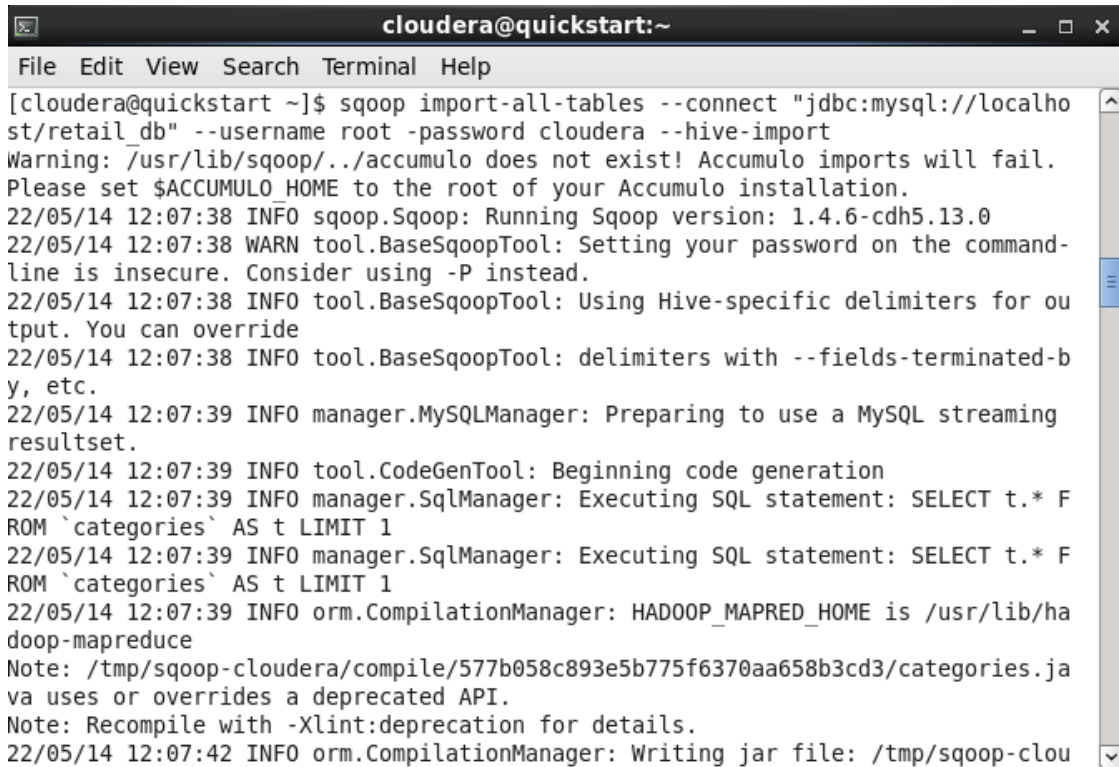


Comprobamos si hay alguna tabla y vemos que no:



Importo todas las tablas con el comando:

```
sqoop import-all-tables --connect "jdbc:mysql://localhost/retail_db" --username root -
password cloudera --hive-import
```

A screenshot of a terminal window titled 'cloudera@quickstart:~'. The terminal shows the execution of the command 'sqoop import-all-tables --connect "jdbc:mysql://localhost/retail\_db" --username root --password cloudera --hive-import'. The output includes a warning about Accumulo, version information (1.4.6-cdh5.13.0), a warning about insecure password handling, and logs from the MySQL manager and code generation tools. The command completes successfully, writing a jar file to the temporary directory.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ sqoop import-all-tables --connect "jdbc:mysql://localho  
st/retail_db" --username root --password cloudera --hive-import  
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.  
Please set $ACCUMULO_HOME to the root of your Accumulo installation.  
22/05/14 12:07:38 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0  
22/05/14 12:07:38 WARN tool.BaseSqoopTool: Setting your password on the command-  
line is insecure. Consider using -P instead.  
22/05/14 12:07:38 INFO tool.BaseSqoopTool: Using Hive-specific delimiters for ou  
tput. You can override  
22/05/14 12:07:38 INFO tool.BaseSqoopTool: delimiters with --fields-terminated-b  
y, etc.  
22/05/14 12:07:39 INFO manager.MySQLManager: Preparing to use a MySQL streaming  
resultset.  
22/05/14 12:07:39 INFO tool.CodeGenTool: Beginning code generation  
22/05/14 12:07:39 INFO manager.SqlManager: Executing SQL statement: SELECT t.* F  
ROM `categories` AS t LIMIT 1  
22/05/14 12:07:39 INFO manager.SqlManager: Executing SQL statement: SELECT t.* F  
ROM `categories` AS t LIMIT 1  
22/05/14 12:07:39 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/ha  
doop-mapreduce  
Note: /tmp/sqoop-cloudera/compile/577b058c893e5b775f6370aa658b3cd3/categories.ja  
va uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.  
22/05/14 12:07:42 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-clou
```

```
sqoop import-all-tables --connect "jdbc:mysql://localhost/retail_db" --  
username root --password cloudera --hive-import
```

Explicación de los parámetros:

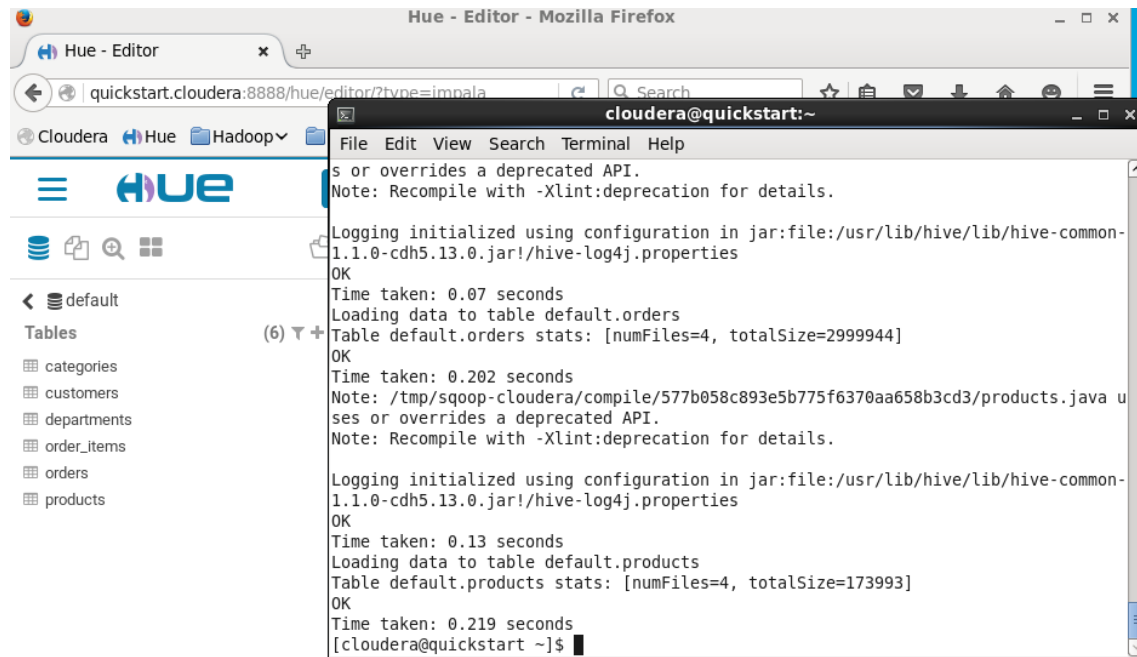
- **sqoop**: Invoca el comando Sqoop.
- **import-all-tables**: Indica que se deben importar todas las tablas de la base de datos especificada.
- **--connect "jdbc:mysql://localhost/retail\_db"**: Especifica la cadena de conexión JDBC para la base de datos fuente. En este caso, la base de datos está en un servidor MySQL ubicado en **localhost** y se llama **retail\_db**.
- **--username root**: Indica el nombre de usuario para conectarse a la base de datos MySQL. En este caso, el usuario es **root**.
- **--password cloudera**: Especifica la contraseña del usuario mencionado anteriormente (**root** en este caso).
- **--hive-import**: Indica que los datos importados deben ser almacenados en Hive, que es un sistema de almacenamiento y procesamiento de datos en Hadoop.

En resumen, este comando Sqoop se utiliza para importar todas las tablas de una base de datos MySQL llamada **retail\_db** (ubicada en **localhost**), utilizando el

usuario **root** con la contraseña **cloudera**, y los datos importados se almacenan en Hive. Esto es útil cuando se trabaja en un entorno de Big Data y se desea transferir datos desde una base de datos relacional (MySQL, en este caso) a un sistema distribuido como Hadoop, utilizando Hive como el almacén de datos.

Tras unos minutos la importación está finalizada.

Podemos ver las tablas en HUE-Hive:



## Hue y Hive

Ejecutar la siguiente consulta (en Hive en HUE-Hive): **listado de categorías de producto más populares.**

**Nota:** las categorías más populares serán aquellas que hayan vendido más productos. No tengo en cuenta si aparecen en más o menos pedidos (que según el caso también podría interesar)

Una posible solución se obtiene con siguiente consulta en HQL:

```
59.84s default text  
1 SELECT category_name, sum(order_item_quantity) as suma  
2 FROM orders, order_items, products, categories  
3 where orders.order_id = order_items.order_item_id  
4 and products.product_id = order_items.order_item_product_id  
5 and products.product_category_id = categories.category_id  
6 GROUP BY category_name  
7 ORDER BY suma desc
```



```

SELECT category_name, sum(order_item_quantity) as suma
FROM orders, order_items, products, categories
where orders.order_id = order_items.order_item_id
and products.product_id = order_items.order_item_product_id
and products.product_category_id = categories.category_id
GROUP BY category_name
ORDER BY suma desc

```

Y este el resultado:

The screenshot shows the Hue Editor interface in a Mozilla Firefox browser. The query editor contains the following SQL code:

```

1 SELECT category_name, sum(order_item_quantity) as suma
2 FROM orders, order_items, products, categories
3 where orders.order_id = order_items.order_item_id
4 and products.product_id = order_items.order_item_product_id
5 and products.product_category_id = categories.category_id
6 GROUP BY category_name
7 ORDER BY suma desc

```

The results are displayed in a table with 24 rows. The table has two columns: 'category\_name' and 'suma'.

category_name	suma
1 Cleats	29342
2 Women's Apparel	25207
3 Indoor/Outdoor Games	23401
4 Cardio Equipment	15403
5 Shop By Sport	13295
6 Men's Footwear	8735
7 Fishing	6885
8 Water Sports	6205
9 Camping & Hiking	5430
10 Electronics	4612
11 Accessories	2465
12 Golf Balls	2226

Visto más en detalle:

	category_name	suma
1	Cleats	29342
2	Women's Apparel	25207
3	Indoor/Outdoor Games	23401
4	Cardio Equipment	15403
5	Shop By Sport	13295
6	Men's Footwear	8735
7	Fishing	6885
8	Water Sports	6205
9	Camping & Hiking	5430

X-AXIS

category\_name

Y-AXIS

☒ suma

GROUP

Choose a column to...

LIMIT

Limit the number of ...

SORTING

