

УДАЛЯЕМ КОНТЕЙНЕР И ОБРАЗ

```
docker rm gbspark
docker rmi img-hdp-spark
docker system prune
```

Соберите находясь в той же директории

```
docker build -t img-hdp-spark .
```

Поднимаем новый контейнер из образа:

```
docker run -it --name gbspark \
```

```
-p 50090:50090 \
-p 50075:50075 \
-p 50070:50070 \
-p 8042:8042 \
-p 8088:8088 \
-p 4040:4040 \
-p 4044:4044 \
-p 8888:8888 \
--hostname localhost \
img-hdp-spark
```

Стартуем остановленный контейнер:

```
docker start -i gbspark
```

Скачиваем и распаковываем дистрибутив:

```
wget https://apache-mirror.rbc.ru/pub/apache/hive/hive-2.3.9/apache-hive-2.3.9-bin.tar.gz
tar xzf apache-hive-2.3.9-bin.tar.gz
rm apache-hive-2.3.9-bin.tar.gz
mv apache-hive-2.3.9-bin hive
```

Задаем необходимые переменные окружения:

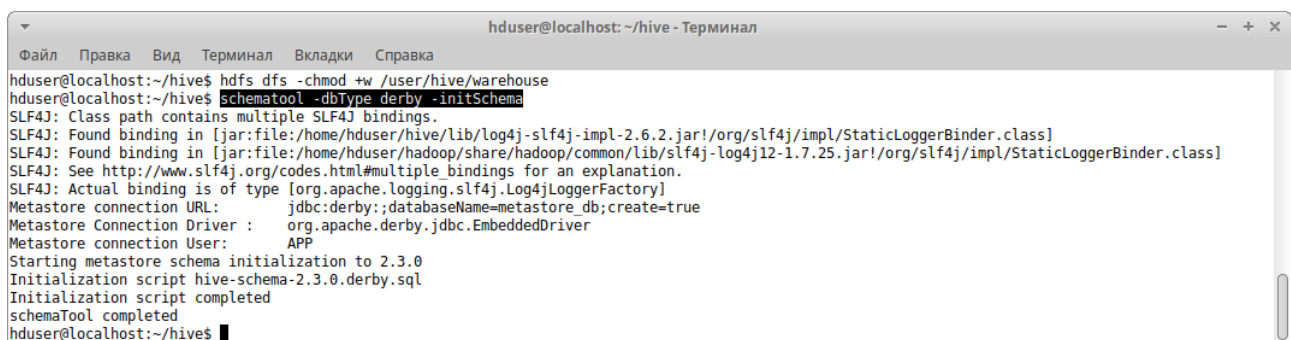
```
export HIVE_HOME=/home/hduser/hive
export PATH=$PATH:$HIVE_HOME/bin
```

Устанавливаем необходимые директории и права:

```
hdfs dfs -mkdir -p /user/hive/warehouse
hdfs dfs -chmod +w /user/hive/warehouse
```

Инициализируем метастор:

```
schematool -dbType derby -initSchema
```

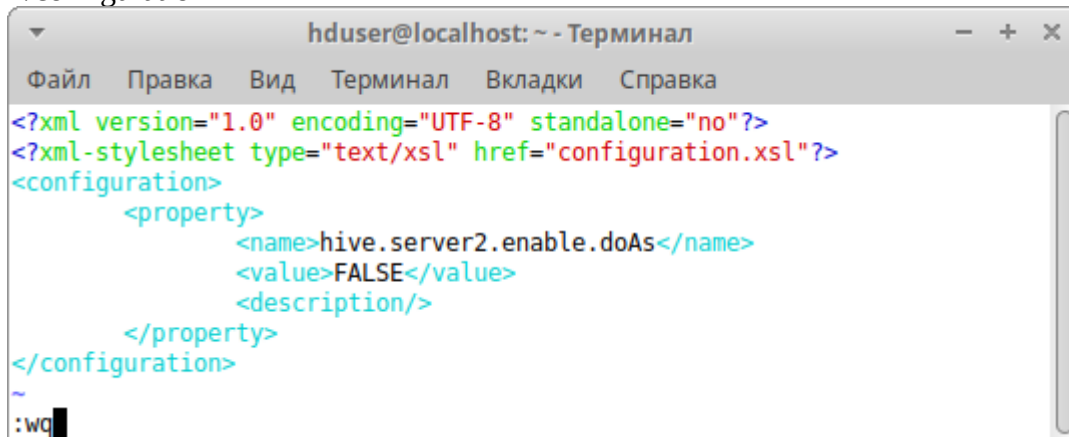


```
hduser@localhost: ~/hive - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
hduser@localhost:~/hive$ hdfs dfs -chmod +w /user/hive/warehouse
hduser@localhost:~/hive$ schematool -dbType derby -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hduser/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.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]
Metastore connection URL:      jdbc:derby;;databaseName=metastore_db;create=true
Metastore connection Driver :  org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:     APP
Starting metastore schema initialization to 2.3.0
Initialization script hive-schema-2.3.0.derby.sql
Initialization script completed
schematool completed
hduser@localhost:~/hive$
```

Создадим файл: vi ~/hive/conf/hive-site.xml и вставим:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>hive.server2.enable.doAs</name>
<value>FALSE</value>
<description/>
</property>
```

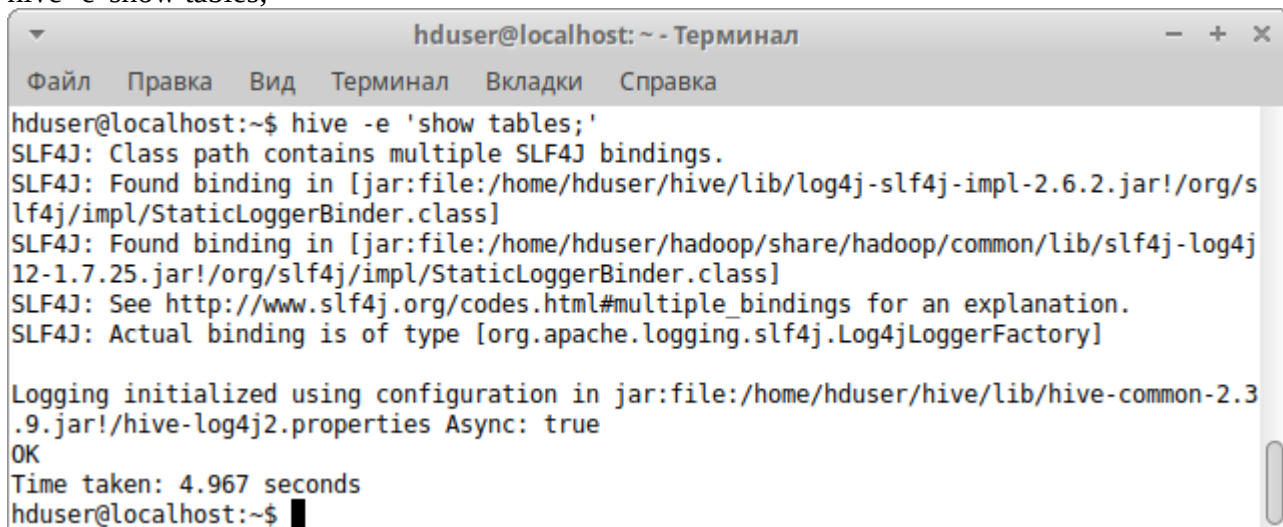
</configuration>



```
hduser@localhost: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>hive.server2.enable.doAs</name>
    <value>FALSE</value>
    <description/>
  </property>
</configuration>
~
:wq
```

Проверяем работу:

hive -e 'show tables;'



```
hduser@localhost: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
hduser@localhost:~$ hive -e 'show tables;'
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hduser/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.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]

Logging initialized using configuration in jar:file:/home/hduser/hive/lib/hive-common-2.3.9.jar!/hive-log4j2.properties Async: true
OK
Time taken: 4.967 seconds
hduser@localhost:~$
```

Запускаем в фоне Hive Server:

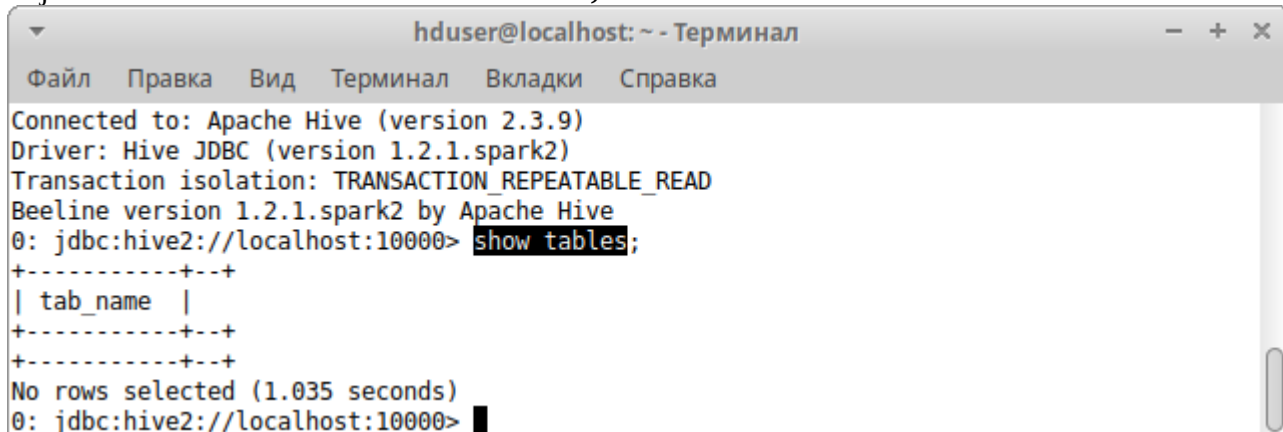
hiveserver2 &> /dev/null &

Подключаемся через beeline cli:

beeline -u jdbc:hive2://localhost:10000

Проверяем работу:

0: jdbc:hive2://localhost:10000> show tables;



```
hduser@localhost: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
Connected to: Apache Hive (version 2.3.9)
Driver: Hive JDBC (version 1.2.1.spark2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 1.2.1.spark2 by Apache Hive
0: jdbc:hive2://localhost:10000> show tables;
+-----+--+
| tab_name |
+-----+--+
+-----+--+
No rows selected (1.035 seconds)
0: jdbc:hive2://localhost:10000>
```

Выходим:

0: jdbc:hive2://localhost:10000> !q

```
hduser@localhost: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
No rows selected (1.035 seconds)
0: jdbc:hive2://localhost:10000> !q
Closing: 0: jdbc:hive2://localhost:10000
hduser@localhost:~$ █
```

УСТАНОВКА АРАСНЕ ZEPPELIN

Скачиваем и распаковываем дистрибутив:

wget https://artfiles.org/apache.org/zeppelin/zeppelin-0.10.0/zeppelin-0.10.0-bin-all.tgz

tar xzf zeppelin-0.10.0-bin-all.tgz

rm zeppelin-0.10.0-bin-all.tgz

mv zeppelin-0.10.0-bin-all zeppelin

Задаем необходимые переменные окружения: vi ~/.bashrc

export ZEPPELIN_HOME=/home/hduser/zeppelin

export PATH=\$PATH:\$ZEPPELIN_HOME/bin

НАСТРОЙКА АРАСНЕ ZEPPELIN

Создадим файл vi ~/zeppelin/conf/zeppelin-env.sh **и вставим:**

#!/bin/bash

export USE_HADOOP=true

export ZEPPELIN_ADDR=0.0.0.0

export ZEPPELIN_PORT=8888

export SPARK_HOME=/home/hduser/spark

export SPARK_APP_NAME=zeppelin-hduser

export HADOOP_CONF_DIR=/home/hduser/hadoop/etc/hadoop

```
hduser@localhost: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
#!/bin/bash
export USE_HADOOP=true
export ZEPPELIN_ADDR=0.0.0.0
export ZEPPELIN_PORT=8888
export SPARK_HOME=/home/hduser/spark
export SPARK_APP_NAME=zeppelin-hduser
export HADOOP_CONF_DIR=/home/hduser/hadoop/etc/hadoop
:wq █
```

Запускаем:

zeppelin-daemon.sh start

```
hduser@localhost: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
hduser@localhost:~$ zeppelin-daemon.sh start
Log dir doesn't exist, create /home/hduser/zeppelin/logs
Pid dir doesn't exist, create /home/hduser/zeppelin/run
Zeppelin start
[ OK ]
hduser@localhost:~$ █
```

ПРОВЕРКА

Переходим на http://localhost:8888:

НАСТРОЙКА ИНТЕРПРЕТАТОРА SPARK

spark.master yarn-cluster

Create new interpreter Interpreter Name

hive

The screenshot shows the Zeppelin web interface in a Mozilla Firefox browser. The address bar indicates the URL is `localhost:8888/#/interpreter`. The main page displays the 'Interpreters' settings for the 'hive' interpreter. A 'Create New Note' dialog box is open in the foreground.

Create New Note Dialog:

- Note Name:** ZeppelinExample
- Default Interpreter:** spark
- Text Area:** Use '/' to create folders. Example: /NoteDirA/Note1
- Create Button:** A blue button labeled 'Create'.

Interpreters Settings (hive):

Option: The interpreter will be instantiated Globally

- ☐ Connect to existing process
- ☐ Set permission

Properties:

Name	Value	Description
default.url	jdbc:hive2://localhost:10000	The URL for JDBC.
default.user		The JDBC user name
default.password		The JDBC user password
default.driver	org.apache.hive.jdbc.HiveDriver	JDBC Driver Name
default.completer.ttlInSeconds	120	Time to live sql completer in seconds (-1 to update everytime, 0 to disable update)
default.completer.schemaFilters		Comma separated schema (schema = catalog = database) filters to get metadata for completions. Supports '%' symbol is equivalent to any set of characters. (ex. prod_v_%,public%,info)
default.precode		SQL which executes while opening connection
default.statementPrecode		Runs before each run of the paragraph, in the same connection
common.max_count	1000	Max number of SQL result to display.
zeppelin.jdbc.auth.type		If auth type is needed, Example: KERBEROS
zeppelin.jdbc.auth.kerberos.proxy.enable	true	When auth type is Kerberos, enable/disable Kerberos proxy with the login user to get the connection. Default value is true.
zeppelin.jdbc.concurrent.use	false	Use parallel scheduler
zeppelin.jdbc.concurrent.max_connection	10	Number of concurrent execution
zeppelin.jdbc.keytab.location		Kerberos keytab location
zeppelin.jdbc.principal		Kerberos principal
zeppelin.jdbc.interpolation	false	Enable ZeppelinContext variable interpolation into paragraph text
zeppelin.jdbc.maxConnLifetime	-1	Maximum of connection lifetime in milliseconds. A value of zero or less means the connection has an infinite lifetime.
zeppelin.jdbc.maxRows	1000	Maximum number of rows fetched from the query.
zeppelin.jdbc.hive.timeout.threshold	60000	Timeout for hive job timeout
zeppelin.jdbc.hive.monitor.query_interval	1000	Query interval for hive statement

Dependencies:

Artifact	Exclude
org.apache.hive:hive-jdbc:2.3.8	
org.apache.hadoop:hadoop-common:2.10.1	

```
sh
python -m pip install kaggle
```

```
sh
export KAGGLE_USERNAME=igortolstikov
export KAGGLE_KEY=fb99c3ad6bf931513798d91567033035
mkdir -p /home/hduser/lego
cd /home/hduser/lego
kaggle datasets files rtatman/lego-database
kaggle datasets download rtatman/lego-database
```

```
sh
cd /home/hduser/lego
unzip lego-database.zip
rm lego-database.zip
ls -la
```

```
sh
hdfs dfs -put /home/hduser/lego /user/hduser
hdfs dfs -ls /user/hduser/lego
```

```
sh
hdfs dfs -cat /user/hduser/lego/colors.csv | head
```

```
sh
echo %table
hdfs dfs -cat /user/hduser/lego/colors.csv | \
  awk -F ',' '{ if ($4 == "${is_trans=f,f|t}") print $0 }' | \
  tr ',' '\t' | head -n ${limit=7}
```

```
sh
hdfs dfs -cat /user/hduser/lego/themes.csv | head
```

```
hive
CREATE TABLE lego_themes(id INT, name STRING, parent_id INT)
  COMMENT 'Information on lego themes'
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
  STORED AS TEXTFILE
  TBLPROPERTIES("skip.header.line.count"="1")
```

```
sh
hdfs dfs -cat /user/hduser/lego/sets.csv | head
```

```
hive
CREATE TABLE lego_sets(set_num STRING, name STRING, year INT, theme_id INT, num_parts
INT)
  COMMENT 'Information on lego themes'
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
  STORED AS TEXTFILE
  TBLPROPERTIES("skip.header.line.count"="1")
```

```
hive
show tables;
```

```
hive
LOAD DATA INPATH '/user/hduser/lego/themes.csv' INTO TABLE lego_themes;
LOAD DATA INPATH '/user/hduser/lego/sets.csv' INTO TABLE lego_sets;
```

```
hive
SELECT * FROM lego_themes LIMIT 5;
```

```
hive
SELECT year, count(*) as count FROM lego_sets GROUP BY year
```

```
sh
hdfs dfs -ls /user/hduser/lego
```

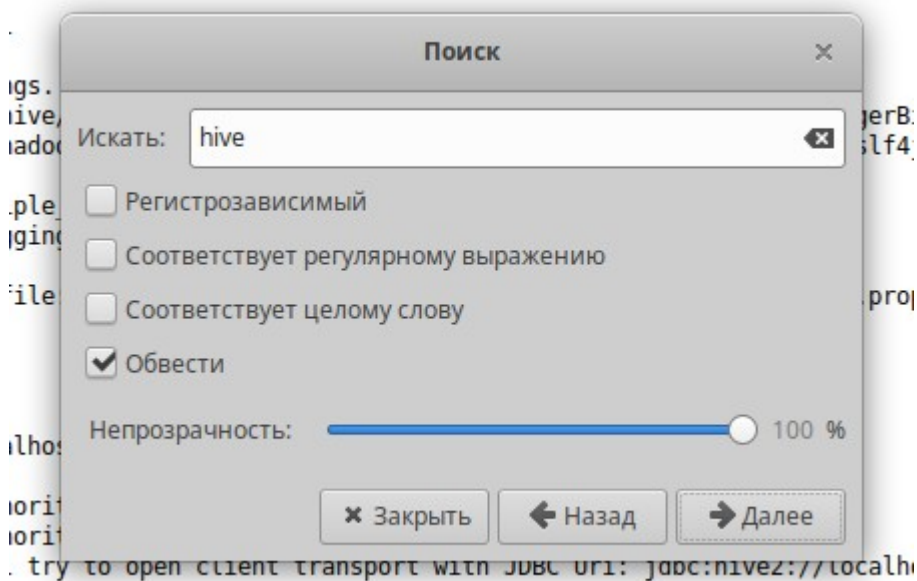
```
sh
hdfs dfs -ls /user/hive/warehouse
```

```
sh
hdfs dfs -cp /user/hive/warehouse/lego_sets/sets.csv /user/hduser/lego/
```

```
sh
hdfs dfs -cp /user/hive/warehouse/lego_themes/themes.csv /user/hduser/lego/
```

```
sh
hdfs dfs -ls /user/hduser/lego
```

```
spark.sql
SELECT * FROM csv.'/user/hduser/lego/sets.csv' LIMIT 10;
mismatched input 'FROM' expecting <EOF>(line 2, pos 9) == SQL == SELECT * FROM
csv.'/user/hduser/lego/sets.csv' LIMIT 10 -----^^^
```



```
[1] 1490
kill 1490
jps -m
```

```
hduser@localhost: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
hduser@localhost:~$ kill 1490
hduser@localhost:~$ jps -m
1920 RemoteInterpreterServer 172.17.0.2 40579 sh-shared_process :
513 SecondaryNameNode
690 ResourceManager
805 NodeManager
4406 Jps -m
1734 ZeppelinServer
328 DataNode
187 NameNode
3036 RemoteInterpreterServer 172.17.0.2 40579 hive-shared_process :
4173 ApplicationMaster --class org.apache.zeppelin.interpreter.remote.RemoteInterpreterServer --jar file:/home/hduser/zeppelin/inter
preter/spark/spark-interpreter-0.10.0.jar --arg 172.17.0.2 --arg 40579 --arg spark-shared_process --arg : --properties-file /tmp/hadoo
p-hduser/nm-local-dir/usercache/hduser/appcache/application_1637951794224_0001/container_1637951794224_0001_01_000001/_spark_conf/_
spark_conf_.properties
4270 CoarseGrainedExecutorBackend --driver-url spark://CoarseGrainedScheduler@localhost:43743 --executor-id 1 --hostname localhost --
cores 1 --app-id application_1637951794224_0001 --user-class-path file:/tmp/hadoop-hduser/nm-local-dir/usercache/hduser/appcache/appl
ication_1637951794224_0001/container_1637951794224_0001_01_000002/_app_.jar --user-class-path file:/tmp/hadoop-hduser/nm-local-dir/
usercache/hduser/appcache/application_1637951794224_0001/container_1637951794224_0001_01_000002/spark-scala-2.11-0.10.0.jar --user-cl
ass-path file:/tmp/hadoop-hduser/nm-local-dir/usercache/hduser/appcache/application_1637951794224_0001/container_1637951794224_0001_0
1_000002/zeppelin-interpreter-shaded-0.10.0.jar
[1]+  Exit 143                  hiveserver2 &> /dev/null
hduser@localhost:~$
```

```
mismatched input 'FROM' expecting <EOF>(line 2, pos 9)
== SQL ==
SELECT * FROM csv.'/user/hduser/lego/sets.csv' LIMIT 10
-----^^^
```

```
spark.sql
SELECT _c2 as year, count(*) as count FROM csv.'/user/hduser/lego/sets.csv' LIMIT 10;
```

```
mismatched input 'FROM' expecting <EOF>(line 2, pos 38) == SQL == SELECT _c2 as year,
count(*) as count FROM csv.'/user/hduser/lego/sets.csv' LIMIT 10
```

```
spark
val a=1
val b=2
val c= a+b
```

```
spark
val a=1
val b=2
val c= a+b
```

```
val df = sqlContext
    .read
    .format("csv")
    .option("header", "true")
    .option("interSchema", "true")
    .load("/user/hduser/lego/sets.csv")
```

```
df.printSchema()
```

```
df.show()
```

```
df.groupBy("year").count.collect
```

```
val rows = df.groupBy("year").count.collect
```

```
println("%table")
```

```
println("year\tcount")
```

```
rows.map{ row => s"${row.getInt(0)}\t${row.getLong(1)}" }.map(println)
```

```
val rows = df.groupBy("year").count.collect
```

```
val data = rows.map(row => getInt(0) + "\t" + row.getLong(1))
```

```
println("%table\n" + "year\tcount\n" + data.mkString("\n"))
```

```
<console>:27: error: not found: value getInt val data = rows.map(row => getInt(0) + "\t" +  
row.getLong(1)) ^ <console>:31: error: value mkString is not a member of Array[Nothing]
```

```
println("%table\n" + "year\tcount\n" + data.mkString("\n"))
```

```
spark.sql
```

```
SELECT year, Count(*) as count FROM lego_sets GROUP BY year
```

```
Table or view not found: lego_sets; line 2 pos 36
```