


SparkSQL with Scala and Screenshot of the results obtained from the SparkSQL commands in Scala. :

```
0 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://localhost:4040
Spark context available as 'sc' (master = yarn, app id = application_1705123457259_0001).
Spark session available as 'spark'.
Welcome to

 version 3.0.0

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_275)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val df = spark.read.format("csv").option("header", "true").load("/data/grades.csv")
df: org.apache.spark.sql.DataFrame = [Last name: string, First name: string ... 7 more fields]

scala> df.createOrReplaceTempView("df")

scala> spark.sql("SHOW TABLES").show()
140832 [main] WARN org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.strict.managed.tables does not exist
140832 [main] WARN org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.create.as.insert.only does not exist
140890 [main] WARN org.apache.spark.sql.hive.client.HiveClientImpl - Detected HiveConf hive.execution.engine is 'tez' and will be reset to 'mr' to disable
useless hive logic

+-----+
|database|tableName|isTemporary|
+-----+
|default|grades|false|
|default|us_state|false|
|default|df|true|
+-----+

scala> spark.sql("SELECT * FROM df WHERE Final > 50").show()

+-----+
|Last name|First name|SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+
|Airpump|Andrew|223-45-6789|49|1|90|100|83|A|
|Backus|Jim|143-12-1234|48|1|97|96|97|A+|
|Elephant|Ima|456-71-9012|45|1|78|88|77|B-|
|Franklin|Benny|234-56-2890|50|1|90|80|90|B-|
+-----+
```

```
scala> spark.sql("SELECT * FROM grades").show()

+-----+
|Last name|First name|SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+
|Alfaifa|Aloyius|123-45-6789|40.0|90|100.0|83.0|45.0|D-|
|Alfred|University|123-12-1234|41.0|97|96.0|97.0|48.0|D+|
|Gerty|Gramma|567-89-0123|41.0|80|60.0|46.0|44.0|C|
|Android|Electric|087-65-4321|42.0|23|36.0|45.0|47.0|B-|
|Bumpkin|Fred|456-78-9012|43.0|78|88.0|77.0|45.0|A-|
|Rubble|Betty|234-56-7890|44.0|90|80.0|90.0|46.0|C-|
|Noshov|Cecil|345-67-8901|45.0|11|-1.0|4.0|43.0|F|
|Buff|Biff|632-79-9939|46.0|20|30.0|40.0|50.0|B+|
|Airpump|Andrew|223-45-6789|49.0|1|90.0|100.0|83.0|A|
|Backus|Jim|143-12-1234|48.0|1|97.0|96.0|97.0|A+|
|Carnivore|Art|565-89-0123|44.0|1|80.0|65.0|40.0|D+|
|Dandy|Jim|087-75-4321|47.0|1|23.0|36.0|45.0|C+|
|Elephant|Ima|456-71-9012|45.0|1|78.0|88.0|77.0|B-|
|Franklin|Benny|234-56-2890|50.0|1|90.0|80.0|90.0|B-|
|George|Boy|345-67-3901|40.0|1|11.0|-1.0|4.0|B|
|Heffalump|Harvey|632-79-9439|30.0|1|20.0|30.0|40.0|C|
+-----+
```

Screenshot of your 3 other SQL query results:

```
scala> spark.sql("SELECT * FROM df WHERE Grade = 'B-'").show()

+-----+
|Last name|First name|SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+
|Android|Electric|087-65-4321|42|23|36|45|47|B-|
|Elephant|Ima|456-71-9012|45|1|78|88|77|B-|
|Franklin|Benny|234-56-2890|50|1|90|80|90|B-|
+-----+
```

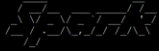
```
scala> spark.sql("SELECT * FROM df WHERE test1 > 40 and Grade = 'B-'").show()

+-----+
|Last name|First name|SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+
|Android|Electric|087-65-4321|42|23|36|45|47|B-|
|Elephant|Ima|456-71-9012|45|1|78|88|77|B-|
|Franklin|Benny|234-56-2890|50|1|90|80|90|B-|
+-----+
```

```
scala> spark.sql("SELECT * FROM df WHERE test2 = 1").show()
+-----+-----+-----+-----+-----+-----+
|Last name|First name|SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+-----+-----+-----+-----+-----+
| Airpump| Andrew|223-45-6789| 49| 1| 90| 100| 83| A|
| Backus| Jim|143-12-1234| 48| 1| 97| 96| 97| A+|
| Carnivore| Art|565-89-0123| 44| 1| 80| 60| 40| D+|
| Dandy| Jim|087-75-4321| 47| 1| 23| 36| 45| C+|
| Elephant| Ima|456-71-9012| 45| 1| 78| 88| 77| B-|
| Franklin| Benny|234-56-2890| 50| 1| 90| 80| 90| B-|
| George| Boy|345-67-3901| 40| 1| 11| -1| 4| B|
| Heffalump| Harvey|632-79-9439| 30| 1| 20| 30| 40| C|
+-----+-----+-----+-----+-----+-----+
```

Screenshot of the results obtained from the SparkSQL commands in Python :

```
scala> !quit
bash-5.0# pyspark
Python 3.7.10 (default, Mar 2 2021, 09:06:08)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/program/spark/jars/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/program/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.slf4j.impl.Log4jLoggerFactory]
0 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
844 [Thread-4] WARN org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.strict.managed.tables does not exist
844 [Thread-4] WARN org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.create.as.insert.only does not exist
Welcome to

 version 3.0.0

Using Python version 3.7.10 (default, Mar 2 2021 09:06:08)
SparkSession available as 'spark'.
>>> df = spark.read.format('csv').option('header', 'true').load('/data/grades.csv')
>>> df.show()
+-----+-----+-----+-----+-----+-----+
|Last name|First name|SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+-----+-----+-----+-----+-----+
| Alfalfa| Aloysius|123-45-6789| 40| 90| 100| 83| 49| D-|
| AlfredUniversity|123-12-1234| 41| 97| 96| 97| 48| D+|
| Gerty| Gramma|567-89-0123| 41| 80| 60| 40| 44| C|
| Android| Electric|087-65-4321| 42| 23| 36| 45| 47| B-|
| Bumpkin| Fred|456-78-9012| 43| 78| 88| 77| 45| A-|
| Rubble| Betty|234-56-7890| 44| 90| 80| 90| 46| C-|
| Noshov| Cecil|345-67-8901| 45| 11| -1| 4| 43| F|
| Buff| Biff|632-79-9939| 46| 20| 30| 40| 50| B+|
| Airpump| Andrew|223-45-6789| 49| 1| 90| 100| 83| A|
| Backus| Jim|143-12-1234| 48| 1| 97| 96| 97| A+|
| Carnivore| Art|565-89-0123| 44| 1| 80| 60| 40| D+|
| Dandy| Jim|087-75-4321| 47| 1| 23| 36| 45| C+|
| Elephant| Ima|456-71-9012| 45| 1| 78| 88| 77| B-|
| Franklin| Benny|234-56-2890| 50| 1| 90| 80| 90| B-|
| George| Boy|345-67-3901| 40| 1| 11| -1| 4| B|
| Heffalump| Harvey|632-79-9439| 30| 1| 20| 30| 40| C|
+-----+-----+-----+-----+-----+-----+
```

```
>>> df.createOrReplaceTempView('df')
>>> spark.sql('SHOW TABLES').show()
93263 [Thread-4] WARN org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.strict.managed.tables does not exist
93263 [Thread-4] WARN org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.create.as.insert.only does not exist
93287 [Thread-4] WARN org.apache.spark.sql.hive.client.HiveClientImpl - Detected HiveConf hive.execution.engine is 'tez' and will be reset to 'mr' to disable useless hive logic

+-----+-----+-----+
|database|tableName|isTemporary|
+-----+-----+-----+
| default| grades| false|
| default| us_state| false|
| | df| true|
+-----+-----+-----+

>>> spark.sql('SELECT * FROM df WHERE Final > 50').show()
+-----+-----+-----+-----+-----+-----+
|Last name|First name|SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+-----+-----+-----+-----+-----+
| Airpump| Andrew|223-45-6789| 49| | | | 100| 83| A|
| Backus| Jim|143-12-1234| 48| 1| 97| 96| 97| A+|
| Elephant| Ima|456-71-9012| 45| 1| 78| 88| 77| B-|
| Franklin| Benny|234-56-2890| 50| 1| 90| 80| 90| B-|
+-----+-----+-----+-----+-----+-----+

>>> spark.sql('SELECT * FROM grades').show()
+-----+-----+-----+-----+-----+-----+
|last name|first name|ssn|test1|test2|test3|test4|final|grade|
+-----+-----+-----+-----+-----+-----+
|Last name|First name|SSN| null| null| null| null| null|Grade|
+-----+-----+-----+-----+-----+-----+
| Alfalfa| Aloysius|123-45-6789| 40.0| 90.0| 100.0| 83.0| 49.0| D-|
| AlfredUniversity|123-12-1234| 41.0| 97.0| 96.0| 97.0| 48.0| D+|
| Gerty| Gramma|567-89-0123| 41.0| 80.0| 60.0| 40.0| 44.0| C|
| Android| Electric|087-65-4321| 42.0| 23.0| 36.0| 45.0| 47.0| B-|
| Bumpkin| Fred|456-78-9012| 43.0| 78.0| 88.0| 77.0| 45.0| A-|
| Rubble| Betty|234-56-7890| 44.0| 90.0| 80.0| 90.0| 46.0| C-|
| Noshov| Cecil|345-67-8901| 45.0| 11.0| -1.0| 4.0| 43.0| F|
| Buff| Biff|632-79-9939| 46.0| 20.0| 30.0| 40.0| 50.0| B+|
| Airpump| Andrew|223-45-6789| 49.0| 1| 90.0| 100.0| 83.0| A|
| Backus| Jim|143-12-1234| 48.0| 1| 97.0| 96.0| 97.0| A+|
| Carnivore| Art|565-89-0123| 44.0| 1| 80.0| 60.0| 40.0| D+|
| Dandy| Jim|087-75-4321| 47.0| 1| 23.0| 36.0| 45.0| C+|
| Elephant| Ima|456-71-9012| 45.0| 1| 78.0| 88.0| 77.0| B-|
| Franklin| Benny|234-56-2890| 50.0| 1| 90.0| 80.0| 90.0| B-|
| George| Boy|345-67-3901| 40.0| 1| 11.0| -1.0| 4.0| B|
| Heffalump| Harvey|632-79-9439| 30.0| 1| 20.0| 30.0| 40.0| C|
+-----+-----+-----+-----+-----+-----+
>>> |
```

Screenshot of your 3 other SQL query results:

```
>>> spark.sql('SELECT * FROM df WHERE test4 = 100.0').show()
+-----+
|Last name|First name|      SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+
| Airpump|   Andrew|223-45-6789| 49| 1| 90| 100| 83| A|
+-----+

>>> spark.sql('SELECT * FROM df WHERE test2 >= 80.0 and test3 <= 90.0').show()
+-----+
|Last name|First name|      SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+
| Gerty|   Grams|567-89-0123| 41| 80| 60| 40| 44| C|
| Rubble|   Betty|234-56-7890| 44| 90| 90| 90| 46| C-|
+-----+

>>> spark.sql('SELECT * FROM df where SSN = "143-12-1234"').show()
+-----+
|Last name|First name|      SSN|Test1|Test2|Test3|Test4|Final|Grade|
+-----+
| Backus|   Jim|143-12-1234| 48| 1| 97| 96| 97| A+|
+-----+
```

SparkSQL with Custom Data Set from week-3 :

```
val df1 = spark.read.format("text").option("header", "true").load("/data/us_state.txt")
```

```
df1.createOrReplaceTempView("df1")
```

```
spark.sql("SHOW TABLES").show()
```

```
spark.sql('SELECT * FROM us_state').show()
```

```
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
0 [main] WARN  org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://localhost:4040
Spark context available as 'sc' (master = yarn, app id = application_1705161142084_0003).
Spark session available as 'spark'.
Welcome to

Spark version 3.0.0

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_275)
Type in expressions to have them evaluated.
Type help for more information.

scala> val df1 = spark.read.format("text").option("header", "true").load("/data/us_state.txt")
df1: org.apache.spark.sql.DataFrame = [value: string]

scala> df1.createOrReplaceTempView("df1")

scala> spark.sql("SELECT * FROM us_state").show()
90096 [main] WARN  org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.strict.managed.tables does not exist
90096 [main] WARN  org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.create.as.insert.only does not exist
90126 [main] WARN  org.apache.spark.sql.hive.client.HiveClientImpl - Detected HiveConf hive.execution.engine is 'tez' and will be reset to 'mr' to disable =
seless hive logic

+-----+
|state|state_cd|capital|largest_city|population|total_area|land_area|water_area|number_of_reps|
+-----+
|Alabama|AL|Montgomery|Huntsville|5024279|52420|50645|1775|7|
|Alaska|AK|Juneau|Anchorage|733391|665384|570641|94743|1|
|Arizona|AZ|Phoenix|Phoenix|7151502|113990|113594|396|9|
|Arkansas|AR|Little Rock|Little Rock|3011524|53179|52035|1143|4|
|California|CA|Sacramento|Los Angeles|39538222|163695|155779|7916|52|
|Colorado|CO|Denver|Denver|5773714|104094|103642|452|8|
|Connecticut|CT|Hartford|Bridgeport|3605944|5543|4842|701|5|
|Delaware|DE|Dover|Wilmington|893948|2499|1949|540|1|
|Florida|FL|Tallahassee|Jacksonville|21538187|65758|53625|12133|28|
|Georgia|GA|Atlanta|Atlanta|10711908|59425|57513|1912|14|
|Hawaii|HI|Honolulu|Honolulu|1455271|10932|6423|4509|2|
|Idaho|ID|Boise|Boise|1838106|93569|82643|526|2|
|Illinois|IL|Springfield|Chicago|12812508|57914|55519|2395|17|
|Indiana|IN|Indianapolis|Indianapolis|6785528|36420|35826|593|9|
|Iowa|IA|Des Moines|Des Moines|3190368|56273|55957|416|4|
|Kansas|KS|Topeka|Wichita|2937880|82278|81759|520|4|
```

Screenshots of 1st select statement from us_state table to select all columns where state_cd = 'NE';

```
scala> spark.sql("SELECT * FROM us_state where state_cd = 'NE'").show()
+-----+
|state|state_cd|capital|largest_city|population|total_area|land_area|water_area|number_of_reps|
+-----+
|Nebraska|NE|Lincoln|Omaha|1961504|77348|76824|524|3|
+-----+
```

Screenshots of 2nd select statement from us_state table to select State, State_cd, Capital, Largest_City, Population, Number_of_Reps where Capital and Largest_City is same ;

```
scala> spark.sql("SELECT state, state_cd, capital, largest_city, population FROM us_state where capital = largest_city").show()
```

state	state_cd	capital	largest_city	population
Arizona	AZ	Phoenix	Phoenix	7151502
Arkansas	AR	Little Rock	Little Rock	3011524
Colorado	CO	Denver	Denver	5773714
Georgia	GA	Atlanta	Atlanta	10711908
Hawaii	HI	Honolulu	Honolulu	1455271
Idaho	ID	Boise	Boise	1839106
Indiana	IN	Indianapolis	Indianapolis	6785528
Iowa	IA	Des Moines	Des Moines	319369
Massachusetts	MA	Boston	Boston	7029917
Mississippi	MS	Jackson	Jackson	2961279
Ohio	OH	Columbus	Columbus	11799448
Oklahoma	OK	Oklahoma City	Oklahoma City	3959353
Rhode Island	RI	Providence	Providence	1097379
Tennessee	TN	Nashville	Nashville	6310840
Utah	UT	Salt Lake City	Salt Lake City	3271616
West Virginia	WV	Charleston	Charleston	1793716
Wyoming	WY	Cheyenne	Cheyenne	576851

Screenshots of 3rd select statement from us_state table to select all columns where Number_of_Reps >= 10;

```
scala> spark.sql("SELECT * FROM us_state where number_of_reps >= 10").show()
```

state	state_cd	capital	largest_city	population	total_area	land_area	water_area	number_of_reps
California	CA	Sacramento	Los Angeles	39538223	163695	155779	7916	52
Florida	FL	Tallahassee	Jacksonville	21538187	65758	53625	12133	28
Georgia	GA	Atlanta	Atlanta	10711908	59425	57613	1912	14
Illinois	IL	Springfield	Chicago	12912508	57914	55539	2395	17
Michigan	MI	Lansing	Detroit	10077331	96714	56539	40175	13
New Jersey	NJ	Trenton	Newark	9288994	8723	7354	1368	12
New York	NY	Albany	New York City	20201249	54555	47126	7429	26
North Carolina	NC	Raleigh	Charlotte	10439388	53819	48618	5201	14
Ohio	OH	Columbus	Columbus	11799448	44826	40861	3965	15
Pennsylvania	PA	Harrisburg	Philadelphia	13002700	46054	44743	1312	17
Texas	TX	Austin	Houston	29145505	268596	261232	7365	38
Virginia	VA	Richmond	Virginia Beach	8631393	42775	39490	3285	11
Washington	WA	Olympia	Seattle	7705281	71298	66456	4842	10