DSC 650

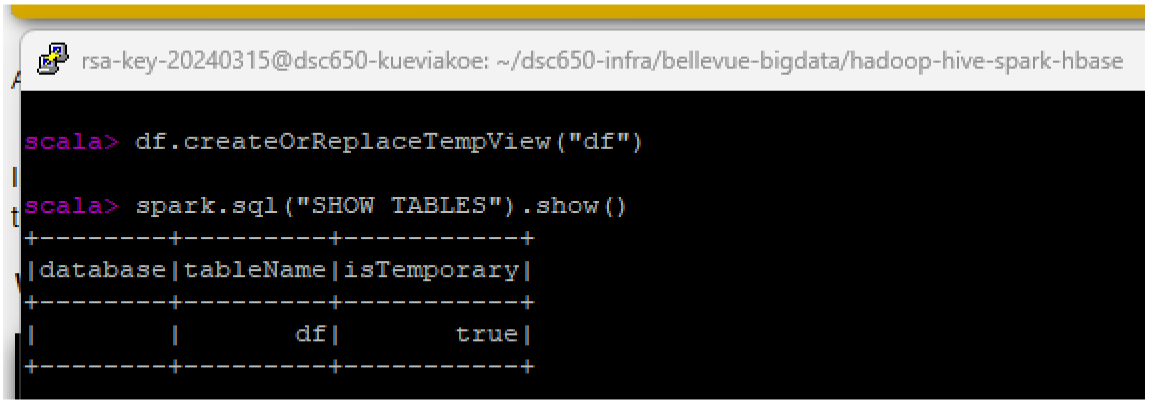
Week 5 Assignment

Eyram Kueviakoe

April 9, 2024

Screenshot of the results obtained from the SparkSQL commands in Scala

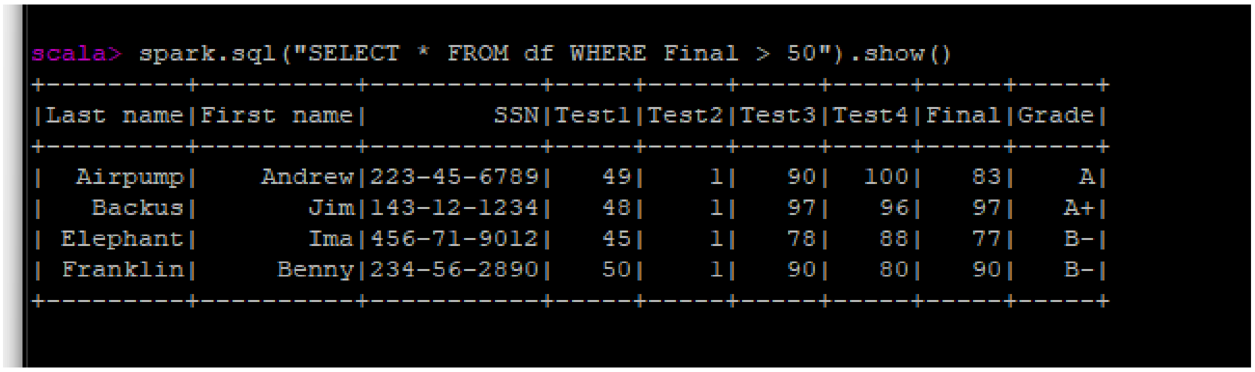
spark.sql("SHOW TABLES").show()



rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase

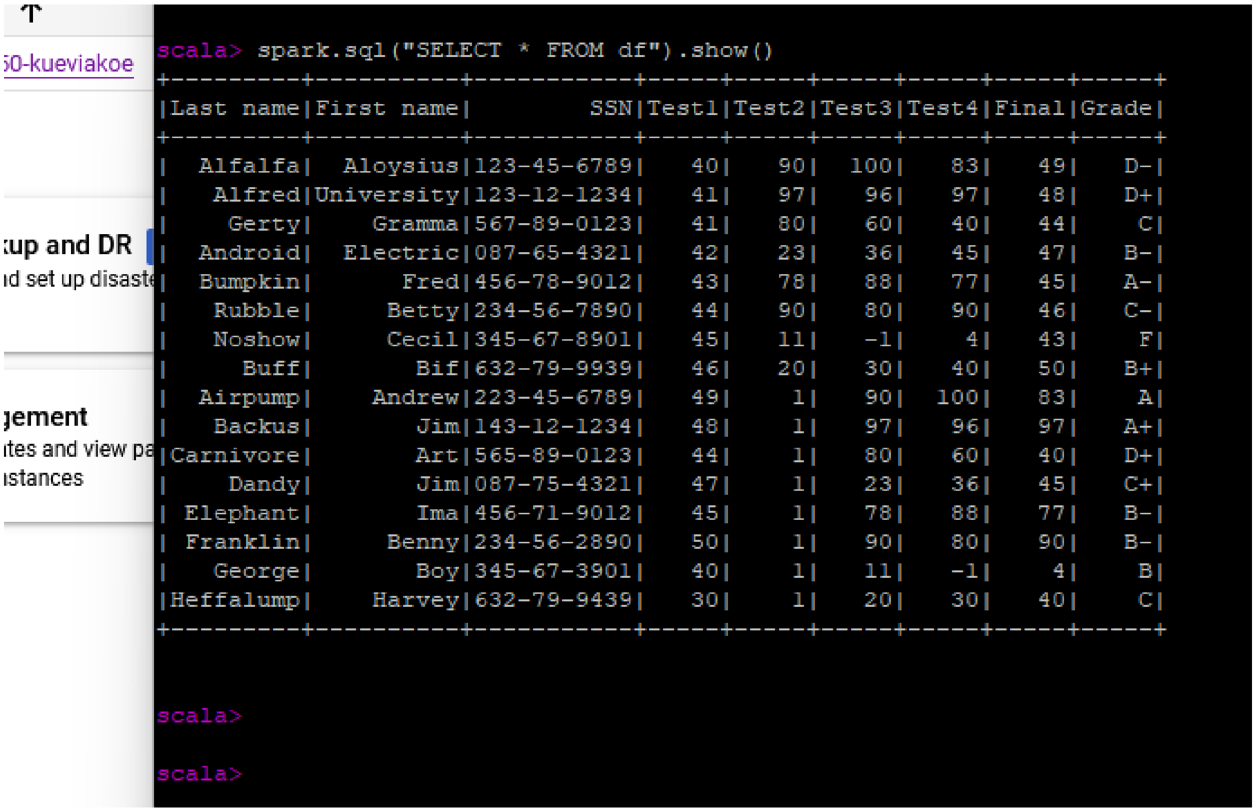
```
scala> df.createOrReplaceTempView("df")

scala> spark.sql("SHOW TABLES").show()
+--------+---------+-----------+
|database|tableName|isTemporary|
+--------+---------+-----------+
|        |       df|       true|
+--------+---------+-----------+
```

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



```
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-|
+---------+----------+-----------+-----+-----+-----+-----+-----+-----+
```

*spark.sql("SELECT * FROM grades").show()*

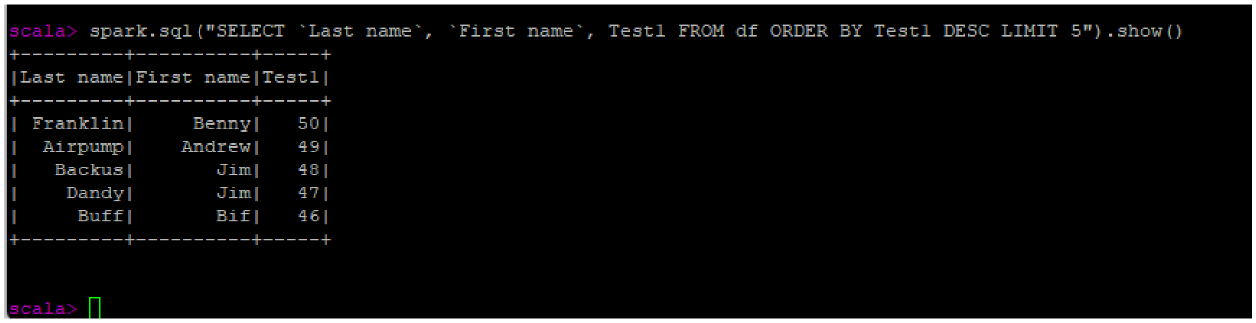


```
scala> spark.sql("SELECT * FROM df").show()
+---------+----------+-----------+-----+-----+-----+-----+-----+-----+
|Last name|First name|        SSN|Test1|Test2|Test3|Test4|Final|Grade|
+---------+----------+-----------+-----+-----+-----+-----+-----+-----+
|  Alfalfa|  Aloysius|123-45-6789|   40|   90|  100|   83|   49|   D-|
|   Alfred|University|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-|
|   Noshow|     Cecil|345-67-8901|   45|   11|   -1|    4|   43|    F|
|     Buff|       Bif|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|
+---------+----------+-----------+-----+-----+-----+-----+-----+-----+


scala>

scala>
```

Screenshot of your 3 other SQL query results

Query 1: Top 5 students with the highest scores in Test 1

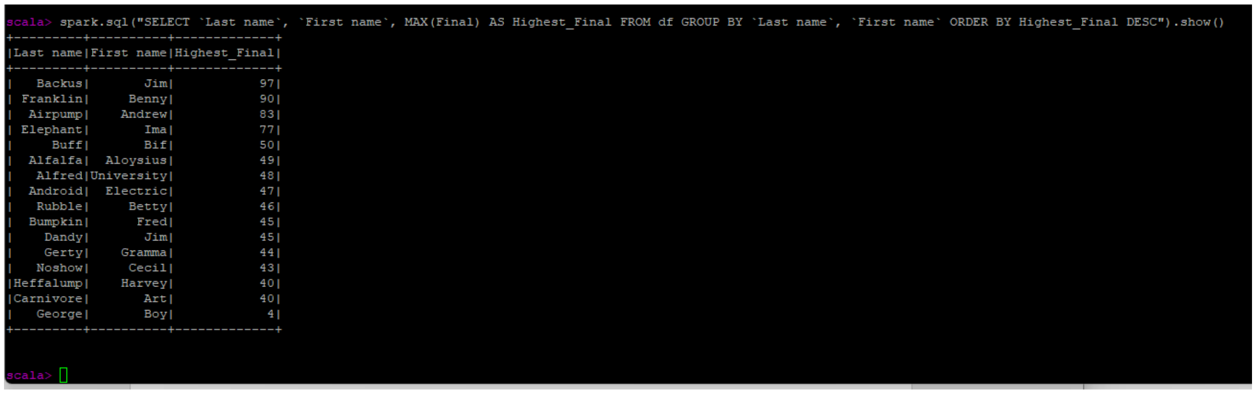
spark.sql("SELECT `Last name`, `First name`, Test1 FROM df ORDER BY Test1 DESC LIMIT 5").show()



```
scala> spark.sql("SELECT `Last name`, `First name`, Test1 FROM df ORDER BY Test1 DESC LIMIT 5").show()
+---------+----------+-----+
|Last name|First name|Test1|
+---------+----------+-----+
| Franklin|     Benny|   50|
|  Airpump|    Andrew|   49|
|   Backus|       Jim|   48|
|    Dandy|       Jim|   47|
|     Buff|       Bif|   46|
+---------+----------+-----+


scala>
```

Query 2: Students with highest Final exam score

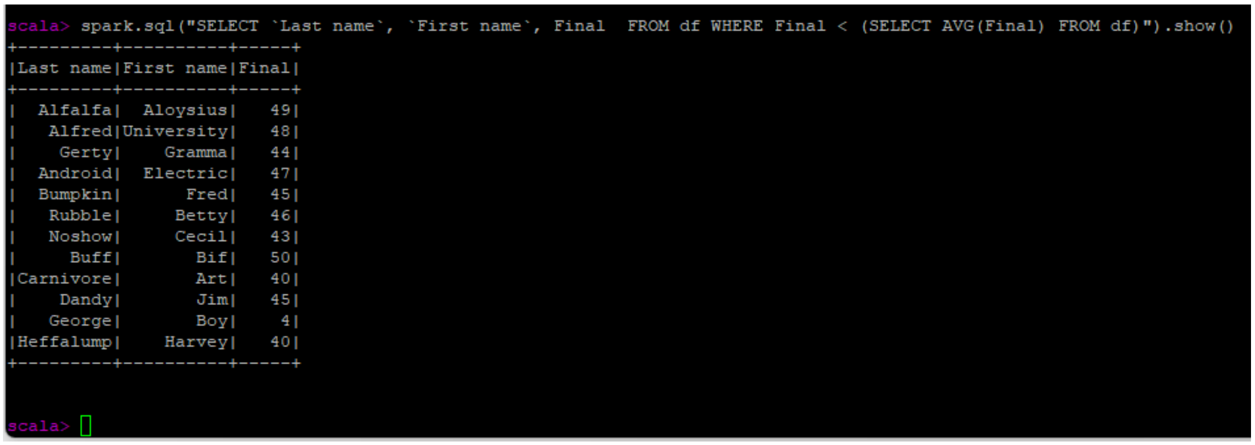
spark.sql("SELECT `Last name`, `First name`, MAX(Final) AS Highest_Final FROM df GROUP BY `Last name`, `First name` ORDER BY Highest_Final DESC").show()



```
scala> spark.sql("SELECT `Last name`, `First name`, MAX(Final) AS Highest_Final FROM df GROUP BY `Last name`, `First name` ORDER BY Highest_Final DESC").show()
+---------+----------+-------------+
|Last name|First name|Highest_Final|
+---------+----------+-------------+
|   Backus|       Jim|           97|
| Franklin|     Benny|           90|
|  Airpump|    Andrew|           83|
| Elephant|       Ima|           77|
|     Buff|       Bif|           50|
|  Alfalfa|  Aloysius|           49|
|   Alfred|University|           48|
|  Android|  Electric|           47|
|   Rubble|     Betty|           46|
|  Bumpkin|      Fred|           45|
|    Dandy|       Jim|           45|
|    Gerty|    Gramma|           44|
|   Noshow|     Cecil|           43|
|Heffalump|    Harvey|           40|
|Carnivore|       Art|           40|
|   George|       Boy|            4|
+---------+----------+-------------+


scala> 
```

Query 3: List of students who scored less than the average final exam score

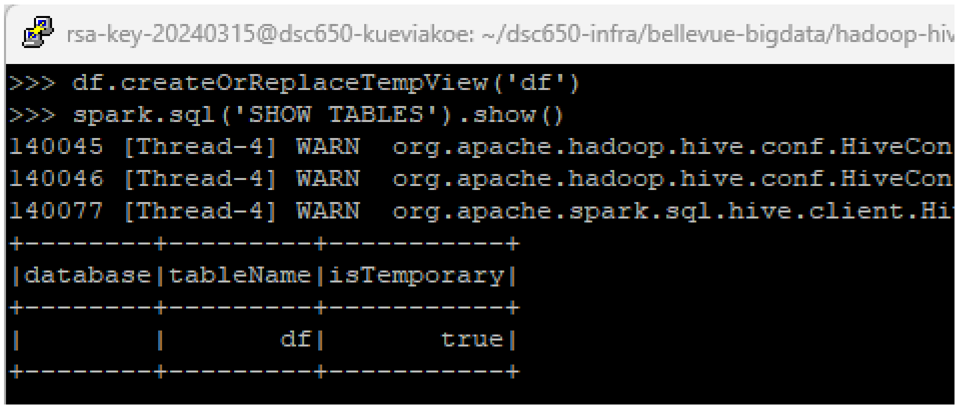
spark.sql("SELECT `Last name`, `First name`, Final FROM df WHERE Final < (SELECT AVG(Final) FROM df)").show()



```
scala> spark.sql("SELECT `Last name`, `First name`, Final  FROM df WHERE Final < (SELECT AVG(Final) FROM df)").show()
+---------+----------+-----+
|Last name|First name|Final|
+---------+----------+-----+
|  Alfalfa|  Aloysius|   49|
|   Alfred|University|   48|
|    Gerty|    Gramma|   44|
|  Android|  Electric|   47|
|  Bumpkin|      Fred|   45|
|   Rubble|     Betty|   46|
|   Noshow|     Cecil|   43|
|     Buff|       Bif|   50|
|Carnivore|       Art|   40|
|    Dandy|       Jim|   45|
|   George|       Boy|    4|
|Heffalump|    Harvey|   40|
+---------+----------+-----+


scala> 
```

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

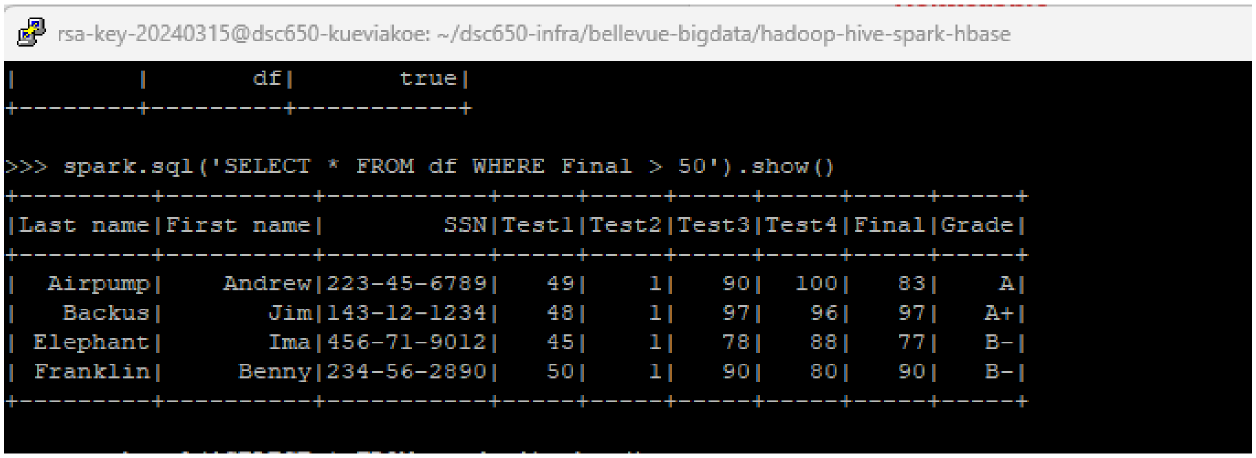
spark.sql('SHOW TABLES').show()



rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hiv

```
>>> df.createOrReplaceTempView('df')
>>> spark.sql('SHOW TABLES').show()
140045 [Thread-4] WARN  org.apache.hadoop.hive.conf.HiveCon
140046 [Thread-4] WARN  org.apache.hadoop.hive.conf.HiveCon
140077 [Thread-4] WARN  org.apache.spark.sql.hive.client.Hi
+--------+---------+-----------+
|database|tableName|isTemporary|
+--------+---------+-----------+
|        |       df|       true|
+--------+---------+-----------+
```

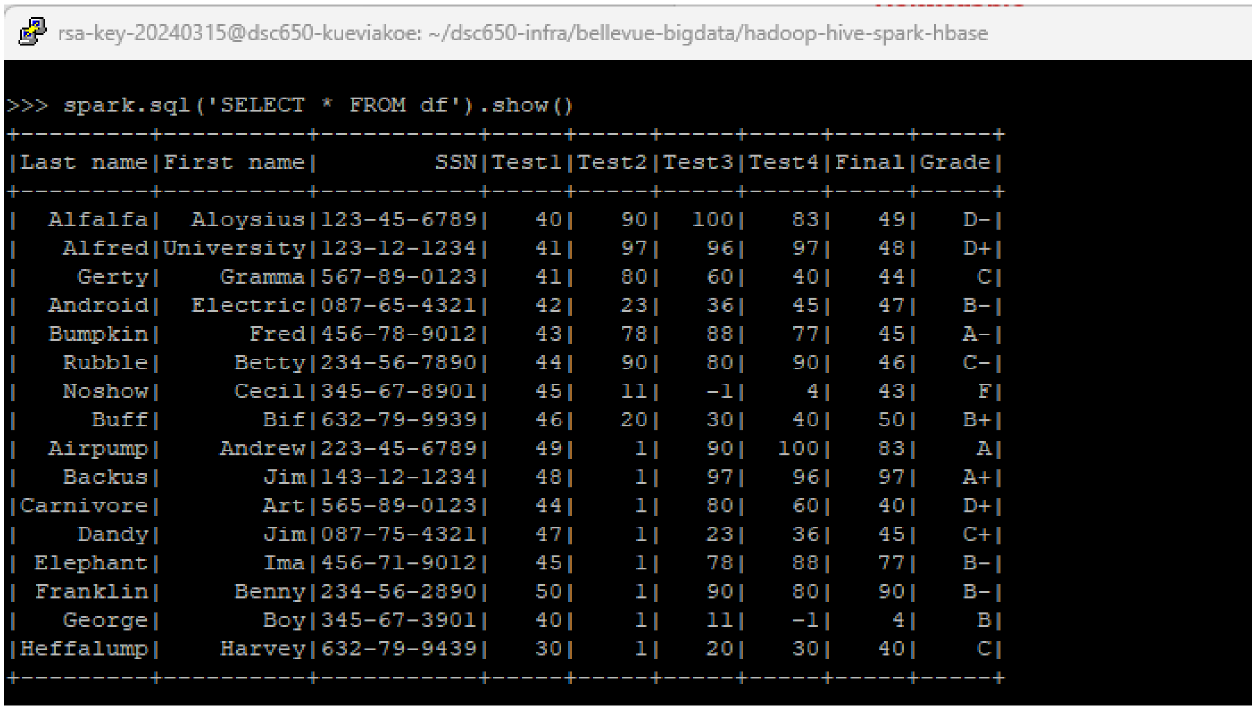
*spark.sql('SELECT * FROM df WHERE Final > 50').show()*



rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase

```
|        |       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|    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-|
+---------+----------+-----------+-----+-----+-----+-----+-----+-----+
```

*spark.sql('SELECT * FROM df').show()*



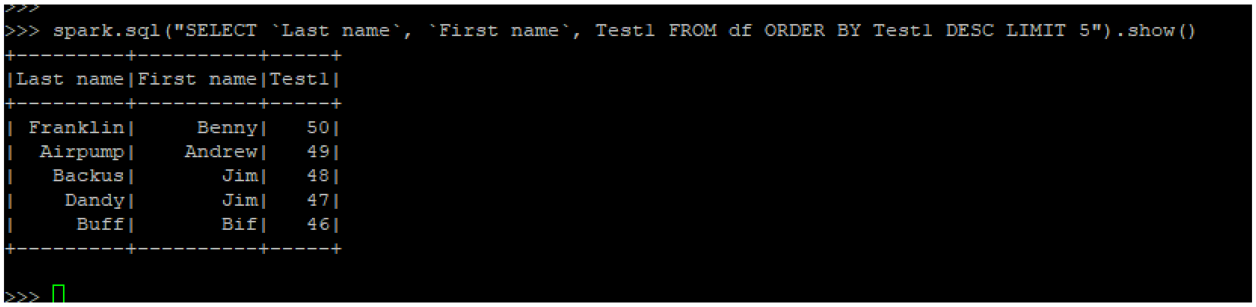
rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase

```
>>> spark.sql('SELECT * FROM df').show()
+---------+----------+-----------+-----+-----+-----+-----+-----+-----+
|Last name|First name|        SSN|Test1|Test2|Test3|Test4|Final|Grade|
+---------+----------+-----------+-----+-----+-----+-----+-----+-----+
|  Alfalfa|  Aloysius|123-45-6789|   40|   90|  100|   83|   49|   D-|
|   Alfred|University|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-|
|   Noshow|     Cecil|345-67-8901|   45|   11|   -1|    4|   43|    F|
|     Buff|       Bif|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|
+---------+----------+-----------+-----+-----+-----+-----+-----+-----+
```

Run 3 other SQL queries in the PySpark Shell

Query 1: **Top 5 students with the highest scores in Test 1**

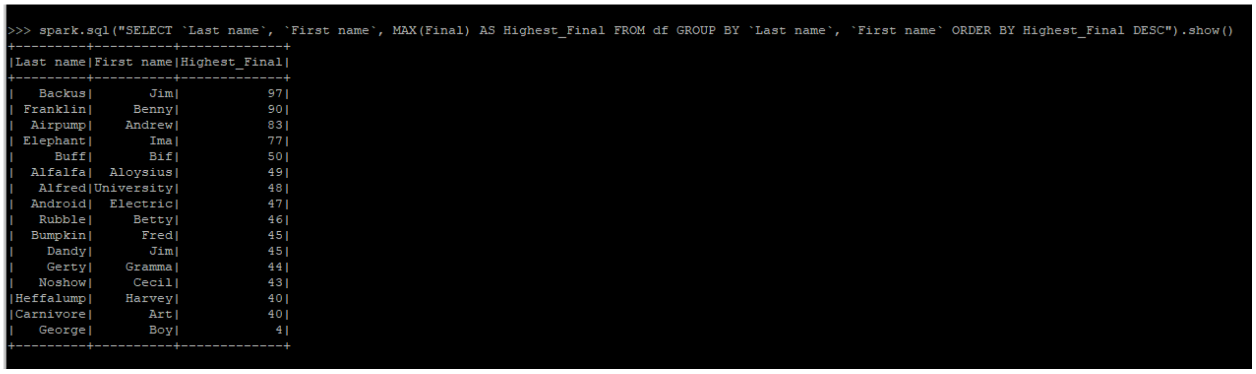
spark.sql("SELECT `Last name`, `First name`, Test1 FROM df ORDER BY Test1 DESC LIMIT 5").show()



```
>>>
>>> spark.sql("SELECT `Last name`, `First name`, Test1 FROM df ORDER BY Test1 DESC LIMIT 5").show()
+---------+----------+-----+
|Last name|First name|Test1|
+---------+----------+-----+
| Franklin|     Benny|   50|
|  Airpump|    Andrew|   49|
|   Backus|       Jim|   48|
|    Dandy|       Jim|   47|
|     Buff|       Bif|   46|
+---------+----------+-----+

>>>
```

Query 2: **Students with highest Final exam score**

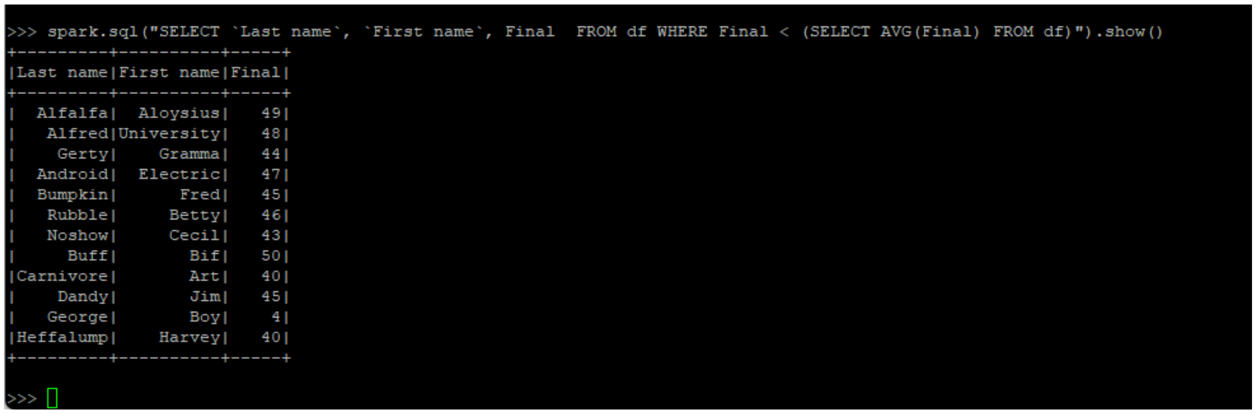
spark.sql("SELECT `Last name`, `First name`, MAX(Final) AS Highest_Final FROM df GROUP BY `Last name`, `First name` ORDER BY Highest_Final DESC").show()



```
>>> spark.sql("SELECT `Last name`, `First name`, MAX(Final) AS Highest_Final FROM df GROUP BY `Last name`, `First name` ORDER BY Highest_Final DESC").show()
+---------+----------+-------------+
|Last name|First name|Highest_Final|
+---------+----------+-------------+
|   Backus|       Jim|           97|
| Franklin|     Benny|           90|
|  Airpump|    Andrew|           83|
| Elephant|       Ima|           77|
|     Buff|       Bif|           50|
|  Alfalfa|  Aloysius|           49|
|   Alfred|University|           48|
|  Android|  Electric|           47|
|   Rubble|     Betty|           46|
|  Bumpkin|      Fred|           45|
|    Dandy|       Jim|           45|
|    Gerty|    Gramma|           44|
|   Noshow|     Cecil|           43|
|Heffalump|    Harvey|           40|
|Carnivore|       Art|           40|
|   George|       Boy|            4|
+---------+----------+-------------+
```

Query 3: List of students who scored less than the average final exam score

spark.sql("SELECT `Last name`, `First name`, Final FROM df WHERE Final < (SELECT AVG(Final) FROM df)").show()



```
>>> spark.sql("SELECT `Last name`, `First name`, Final  FROM df WHERE Final < (SELECT AVG(Final) FROM df)").show()
+---------+----------+-----+
|Last name|First name|Final|
+---------+----------+-----+
|  Alfalfa|  Aloysius|   49|
|   Alfred|University|   48|
|    Gerty|    Gramma|   44|
|  Android|  Electric|   47|
|  Bumpkin|      Fred|   45|
|   Rubble|     Betty|   46|
|   Noshow|     Cecil|   43|
|     Buff|       Bif|   50|
|Carnivore|       Art|   40|
|    Dandy|       Jim|   45|
|   George|       Boy|    4|
|Heffalump|    Harvey|   40|
+---------+----------+-----+

>>> 
```

3- SparkSQL with custom data set

Our dataset from assignment 3 is world_pop_data.csv.

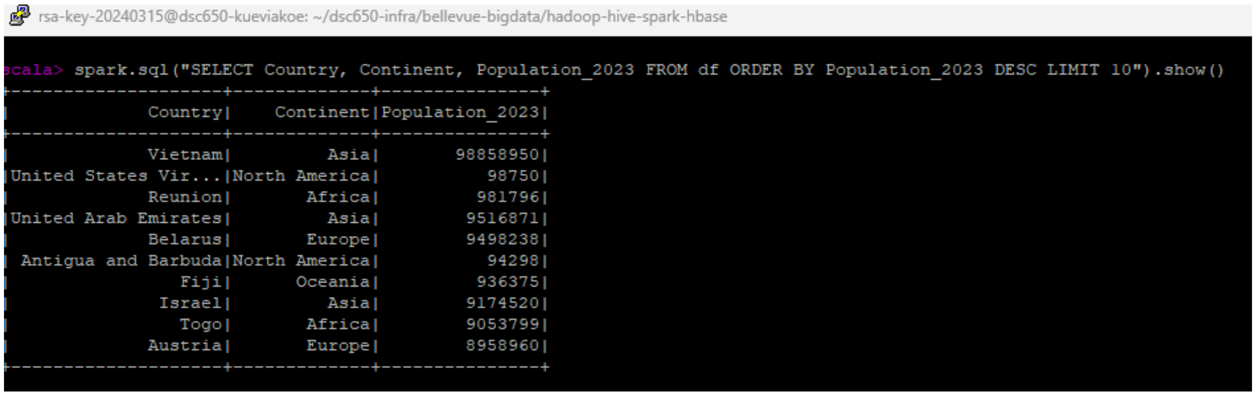
Loading data into Spark

val df = spark.read.format("csv").option("header", "true").load("/data/world_pop_data.csv")

df.createOrReplaceTempView("df")

Query 1: 10 most populated countries in 2023

spark.sql("SELECT Country, Continent, Population_2023 FROM df ORDER BY Population_2023 DESC LIMIT 10").show()

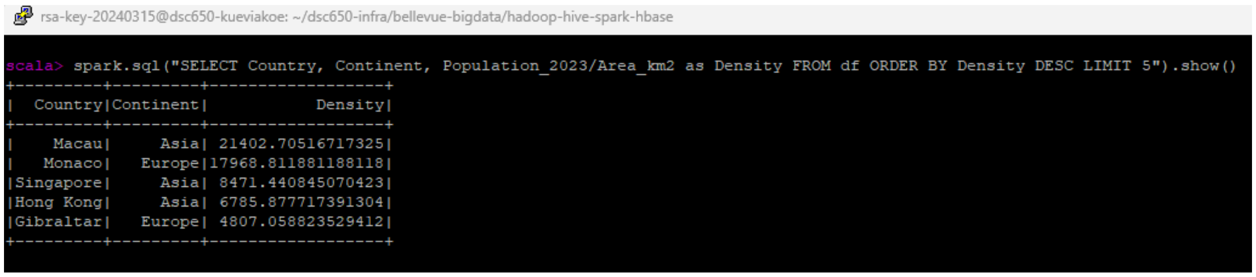


rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase

```
scala> spark.sql("SELECT Country, Continent, Population_2023 FROM df ORDER BY Population_2023 DESC LIMIT 10").show()
+--------------------+-------------+---------------+
|             Country|    Continent|Population_2023|
+--------------------+-------------+---------------+
|             Vietnam|         Asia|       98858950|
|United States Vir...|North America|          98750|
|             Reunion|       Africa|         981796|
|United Arab Emirates|         Asia|        9516871|
|             Belarus|       Europe|        9498238|
| Antigua and Barbuda|North America|          94298|
|                Fiji|      Oceania|         936375|
|              Israel|         Asia|        9174520|
|                Togo|       Africa|        9053799|
|             Austria|       Europe|        8958960|
+--------------------+-------------+---------------+
```

Query 2: What are the top 5 countries with highest density in 2023

spark.sql("SELECT Country, Continent, Population_2023/Area_km2 as Density FROM df ORDER BY Density DESC LIMIT 5").show()

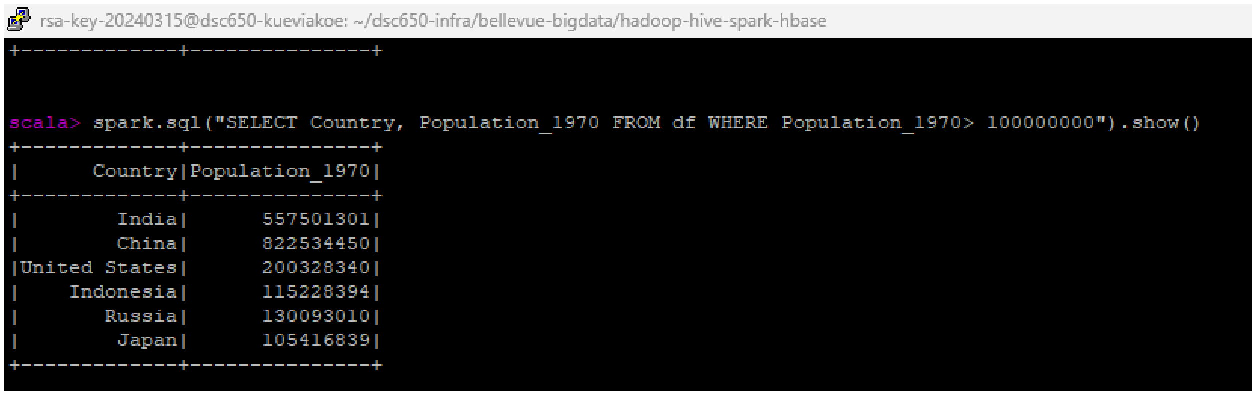


rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase

```
scala> spark.sql("SELECT Country, Continent, Population_2023/Area_km2 as Density FROM df ORDER BY Density DESC LIMIT 5").show()
+---------+---------+-----------------+
|  Country|Continent|          Density|
+---------+---------+-----------------+
|    Macau|     Asia| 21402.70516717325|
|   Monaco|   Europe|17968.811881188118|
|Singapore|     Asia| 8471.440845070423|
|Hong Kong|     Asia| 6785.877717391304|
|Gibraltar|   Europe| 4807.058823529412|
+---------+---------+-----------------+
```

Query 3: Find countries with a population greater than 100 million in 1970:

spark.sql("SELECT Country, Population_1970 FROM df WHERE Population_1970> 100000000").show()

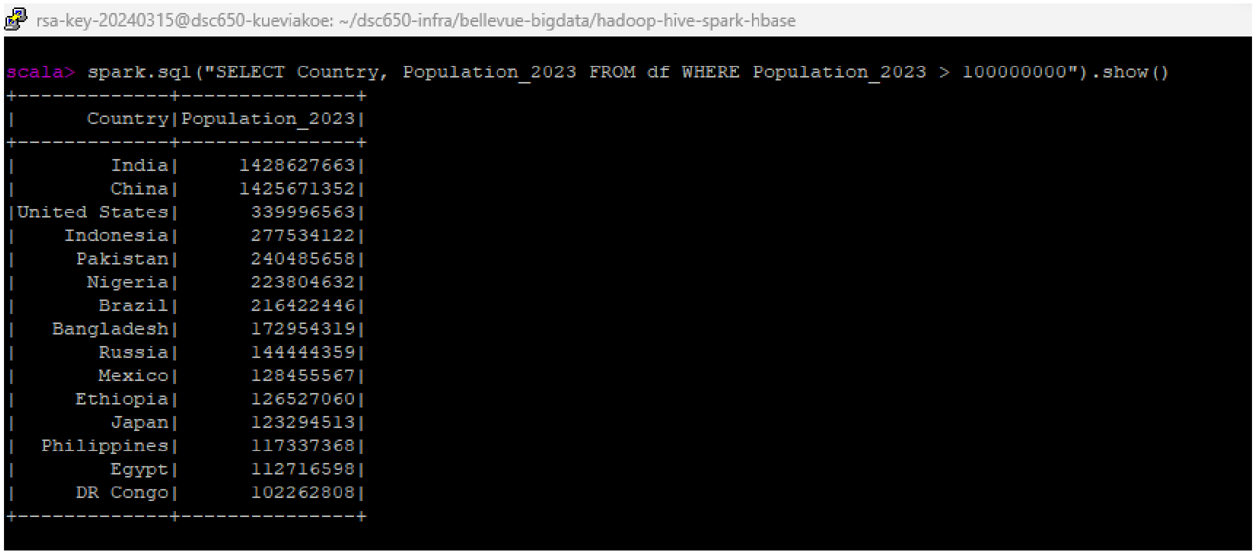


rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase

```
+-------------+---------------+


scala> spark.sql("SELECT Country, Population_1970 FROM df WHERE Population_1970> 100000000").show()
+-------------+---------------+
|      Country|Population_1970|
+-------------+---------------+
|        India|      557501301|
|        China|      822534450|
|United States|      200328340|
|    Indonesia|      115228394|
|       Russia|      130093010|
|        Japan|      105416839|
+-------------+---------------+
```

Query 4: Find countries with a population greater than 100 million in 2023:

spark.sql("SELECT Country, Population_2023 FROM df WHERE Population_2023 > 100000000").show()



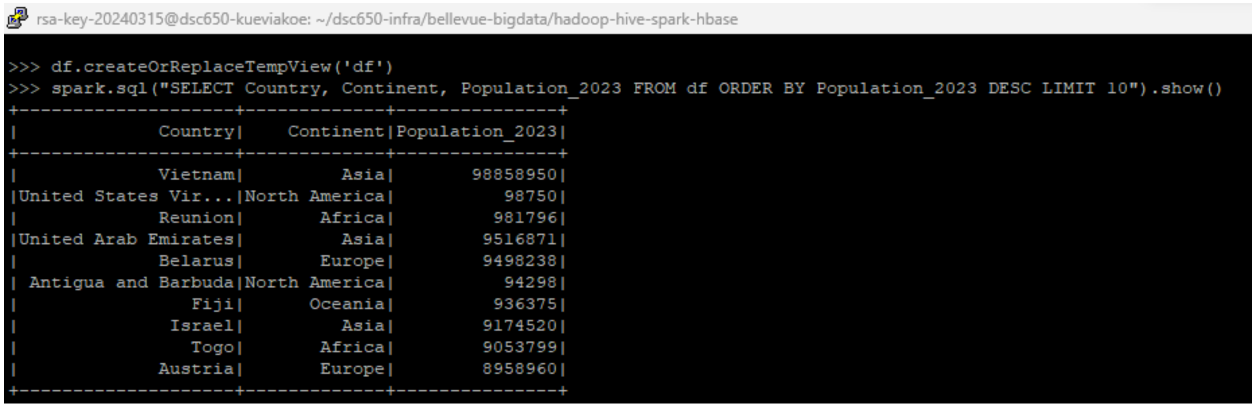
rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase

```
scala> spark.sql("SELECT Country, Population_2023 FROM df WHERE Population_2023 > 100000000").show()
+-------------+---------------+
|      Country|Population_2023|
+-------------+---------------+
|        India|     1428627663|
|        China|     1425671352|
|United States|      339996563|
|    Indonesia|      277534122|
|     Pakistan|      240485658|
|      Nigeria|      223804632|
|       Brazil|      216422446|
|   Bangladesh|      172954319|
|       Russia|      144444359|
|       Mexico|      128455567|
|     Ethiopia|      126527060|
|        Japan|      123294513|
|  Philippines|      117337368|
|        Egypt|      112716598|
|     DR Congo|      102262808|
+-------------+---------------+
```

Running the same queries using PySpark

Query 1: 10 most populated countries in 2023

spark.sql("SELECT Country, Continent, Population_2023 FROM df ORDER BY Population_2023 DESC LIMIT 10").show()

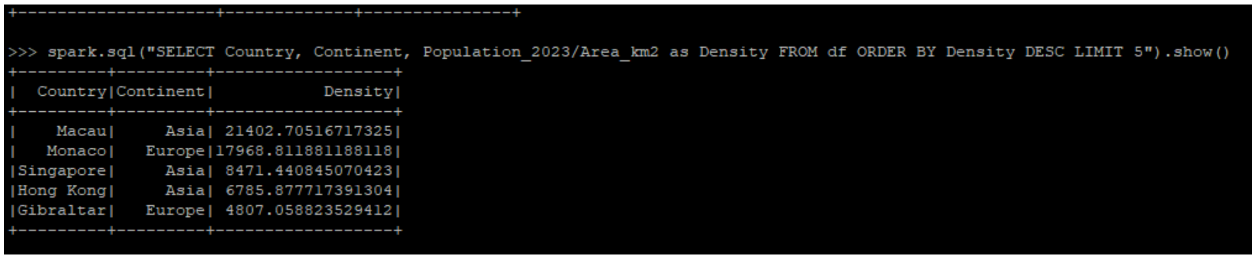


rsa-key-20240315@dsc650-kueviakoe: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase

```
>>> df.createOrReplaceTempView('df')
>>> spark.sql("SELECT Country, Continent, Population_2023 FROM df ORDER BY Population_2023 DESC LIMIT 10").show()
+--------------------+-------------+---------------+
|             Country|    Continent|Population_2023|
+--------------------+-------------+---------------+
|             Vietnam|         Asia|       98858950|
|United States Vir...|North America|          98750|
|             Reunion|       Africa|         981796|
|United Arab Emirates|         Asia|        9516871|
|             Belarus|       Europe|        9498238|
| Antigua and Barbuda|North America|          94298|
|                Fiji|      Oceania|         936375|
|              Israel|         Asia|        9174520|
|                Togo|       Africa|        9053799|
|             Austria|       Europe|        8958960|
+--------------------+-------------+---------------+
```

Query 2: What are the top 5 countries with highest density in 2023

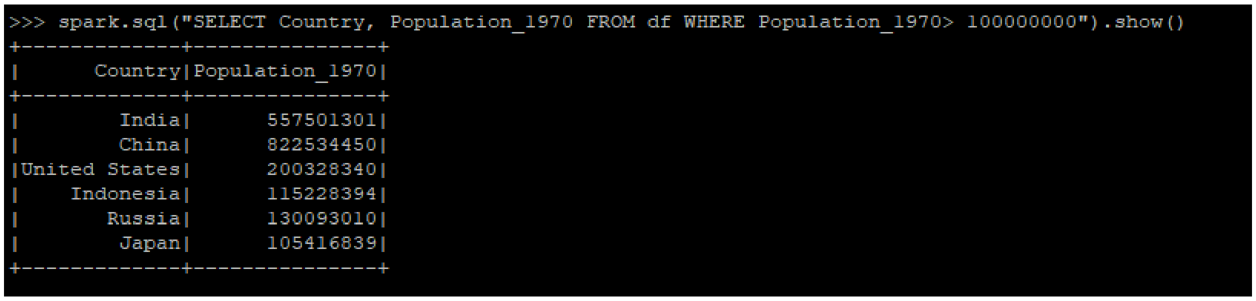
spark.sql("SELECT Country, Continent, Population_2023/Area_km2 as Density FROM df ORDER BY Density DESC LIMIT 5").show()



```
+-------------------+------------+--------------+

>>> spark.sql("SELECT Country, Continent, Population_2023/Area_km2 as Density FROM df ORDER BY Density DESC LIMIT 5").show()
+---------+---------+-----------------+
|  Country|Continent|          Density|
+---------+---------+-----------------+
|    Macau|     Asia| 21402.70516717325|
|   Monaco|   Europe|17968.811881188118|
|Singapore|     Asia| 8471.440845070423|
|Hong Kong|     Asia| 6785.877717391304|
|Gibraltar|   Europe| 4807.058823529412|
+---------+---------+-----------------+
```

Query 3: Find countries with a population greater than 100 million in 1970:

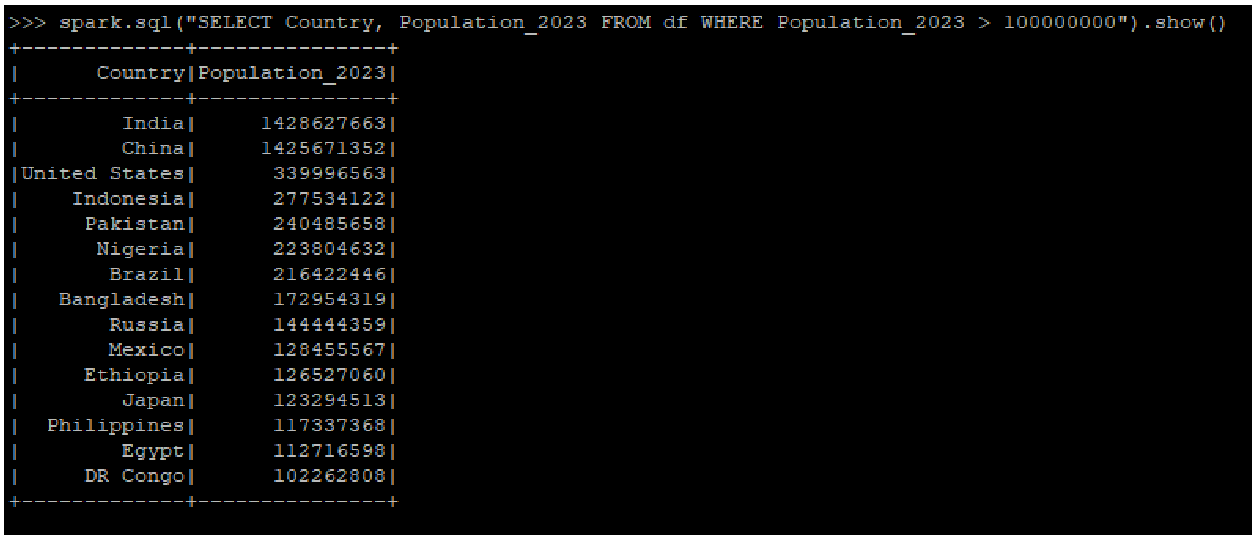
spark.sql("SELECT Country, Population_1970 FROM df WHERE Population_1970> 100000000").show()



```
>>> spark.sql("SELECT Country, Population_1970 FROM df WHERE Population_1970> 100000000").show()
+-------------+---------------+
|      Country|Population_1970|
+-------------+---------------+
|        India|      557501301|
|        China|      822534450|
|United States|      200328340|
|    Indonesia|      115228394|
|       Russia|      130093010|
|        Japan|      105416839|
+-------------+---------------+
```

Query 4: Find countries with a population greater than 100 million in 2023:

spark.sql("SELECT Country, Population_2023 FROM df WHERE Population_2023 > 100000000").show()



```
>>> spark.sql("SELECT Country, Population_2023 FROM df WHERE Population_2023 > 100000000").show()
+-------------+---------------+
|      Country|Population_2023|
+-------------+---------------+
|        India|     1428627663|
|        China|     1425671352|
|United States|      339996563|
|    Indonesia|      277534122|
|     Pakistan|      240485658|
|      Nigeria|      223804632|
|       Brazil|      216422446|
|   Bangladesh|      172954319|
|       Russia|      144444359|
|       Mexico|      128455567|
|     Ethiopia|      126527060|
|        Japan|      123294513|
|  Philippines|      117337368|
|        Egypt|      112716598|
|     DR Congo|      102262808|
+-------------+---------------+
```