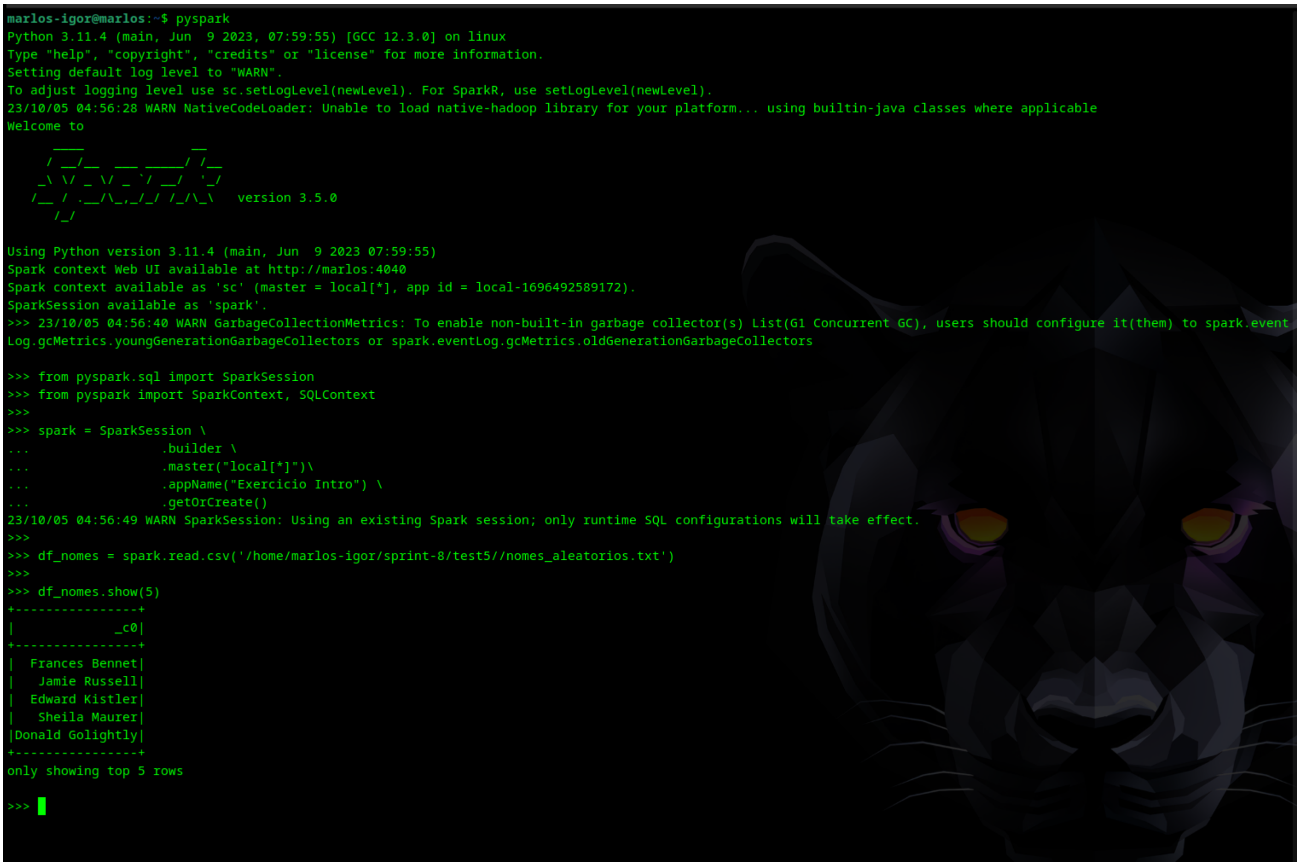
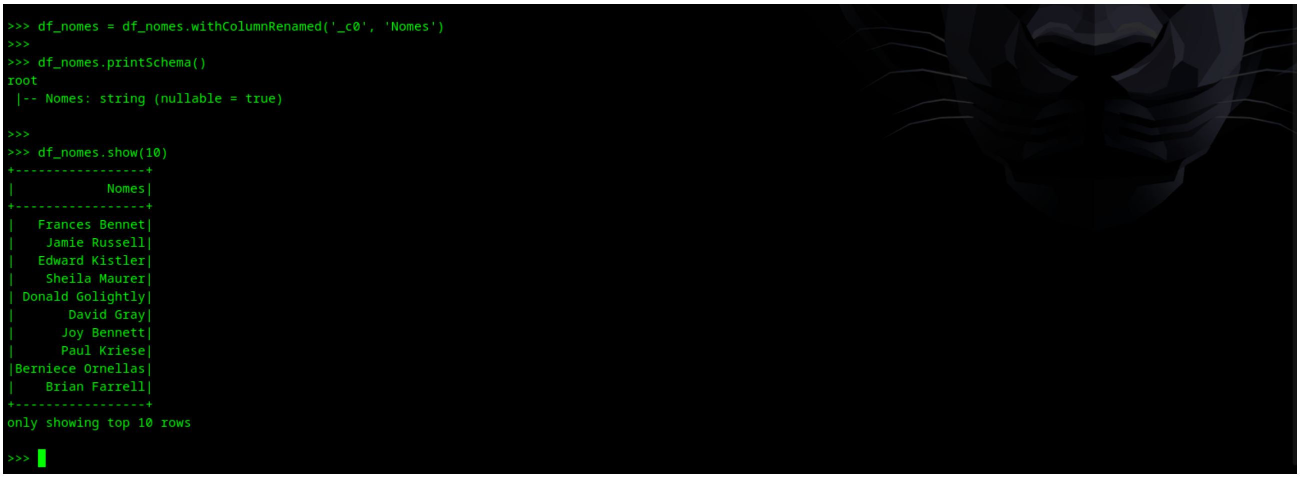
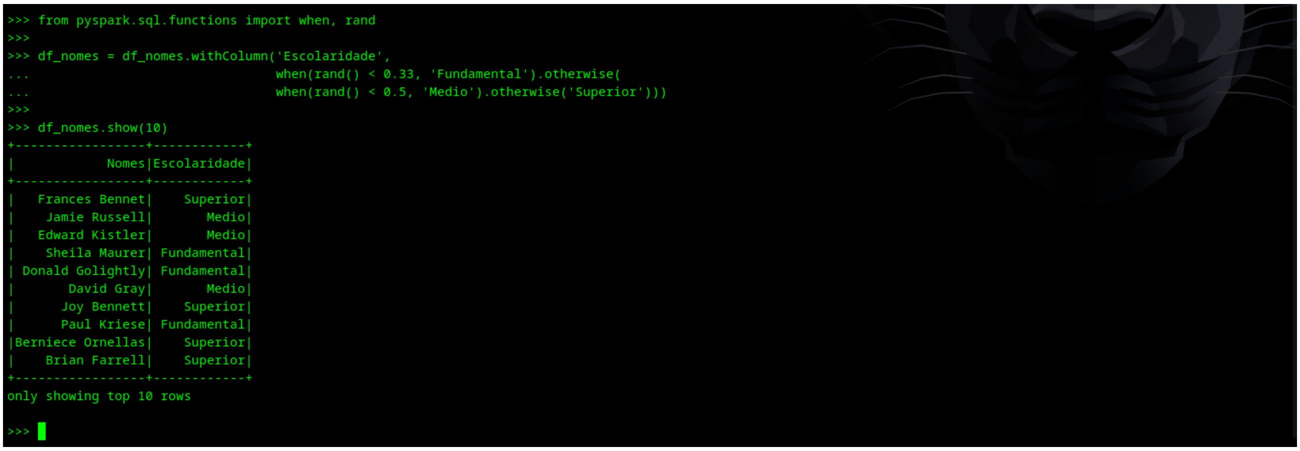


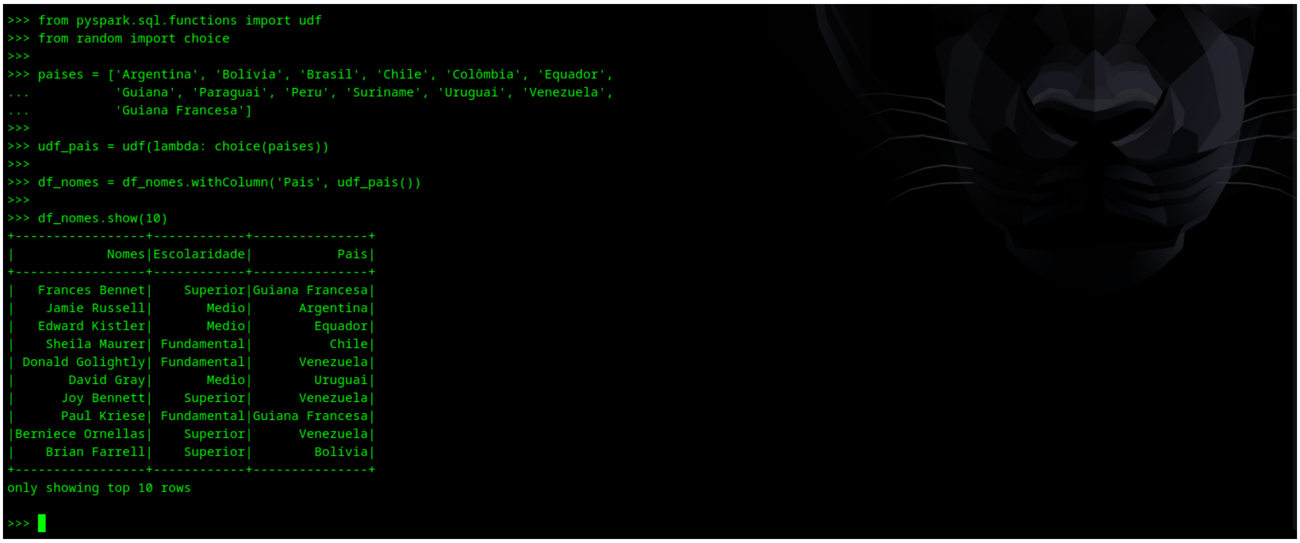
```
marlos-igor@marlos:~$ pyspark
Python 3.11.4 (main, Jun  9 2023, 07:59:55) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/10/05 04:56:28 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.0
      /_/

Using Python version 3.11.4 (main, Jun  9 2023 07:59:55)
Spark context Web UI available at http://marlos:4040
Spark context available as 'sc' (master = local[*], app id = local-1696492589172).
SparkSession available as 'spark'.
>>> 23/10/05 04:56:40 WARN GarbageCollectionMetrics: To enable non-built-in garbage collector(s) List(G1 Concurrent GC), users should configure it(them) to spark.event
Log.gcMetrics.youngGenerationGarbageCollectors or spark.eventLog.gcMetrics.oldGenerationGarbageCollectors

>>> from pyspark.sql import SparkSession
>>> from pyspark import SparkContext, SQLContext
>>>
>>> spark = SparkSession \
...                 .builder \
...                 .master("local[*]")\
...                 .appName("Exercicio Intro") \
...                 .getOrCreate()
23/10/05 04:56:49 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>>
>>> df_nomes = spark.read.csv('/home/marlos-igor/sprint-8/test5//nomes_aleatorios.txt')
>>>
>>> df_nomes.show(5)
+----------------+
|             _c0|
+----------------+
|  Frances Bennet|
|   Jamie Russell|
|  Edward Kistler|
|   Sheila Maurer|
|Donald Golightly|
+----------------+
only showing top 5 rows

>>>
```



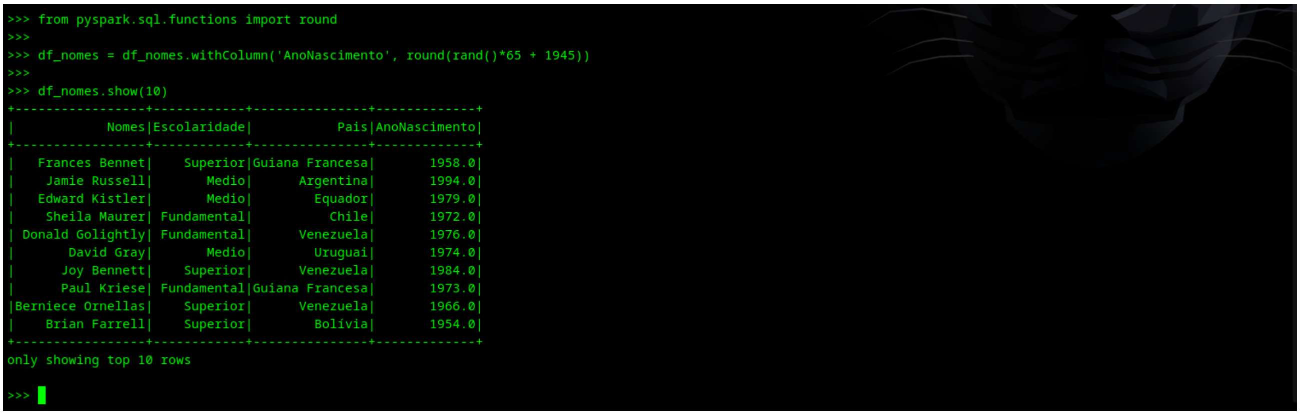
```
>>> df_nomes = df_nomes.withColumnRenamed('_c0', 'Nomes')
>>>
>>> df_nomes.printSchema()
root
 |-- Nomes: string (nullable = true)

>>>
>>> df_nomes.show(10)
+-----------------+
|            Nomes|
+-----------------+
|   Frances Bennet|
|    Jamie Russell|
|   Edward Kistler|
|    Sheila Maurer|
| Donald Golightly|
|       David Gray|
|      Joy Bennett|
|      Paul Kriese|
|Berniece Ornellas|
|    Brian Farrell|
+-----------------+
only showing top 10 rows

>>>
```



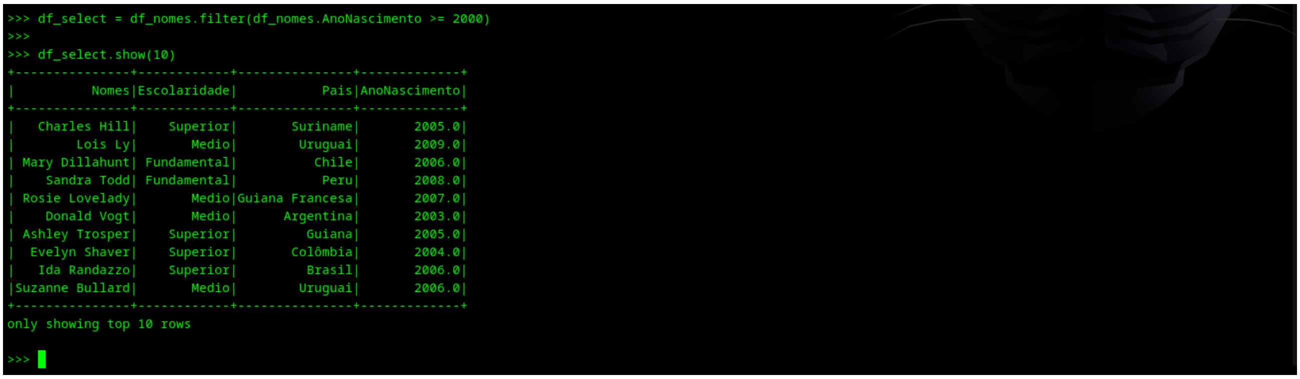
```
>>> from pyspark.sql.functions import when, rand
>>>
>>> df_nomes = df_nomes.withColumn('Escolaridade',
...                                   when(rand() < 0.33, 'Fundamental').otherwise(
...                                   when(rand() < 0.5, 'Medio').otherwise('Superior')))
>>>
>>> df_nomes.show(10)
+-----------------+------------+
|            Nomes|Escolaridade|
+-----------------+------------+
|   Frances Bennet|    Superior|
|    Jamie Russell|       Medio|
|   Edward Kistler|       Medio|
|    Sheila Maurer| Fundamental|
| Donald Golightly| Fundamental|
|       David Gray|       Medio|
|      Joy Bennett|    Superior|
|      Paul Kriese| Fundamental|
|Berniece Ornellas|    Superior|
|    Brian Farrell|    Superior|
+-----------------+------------+
only showing top 10 rows

>>>
```



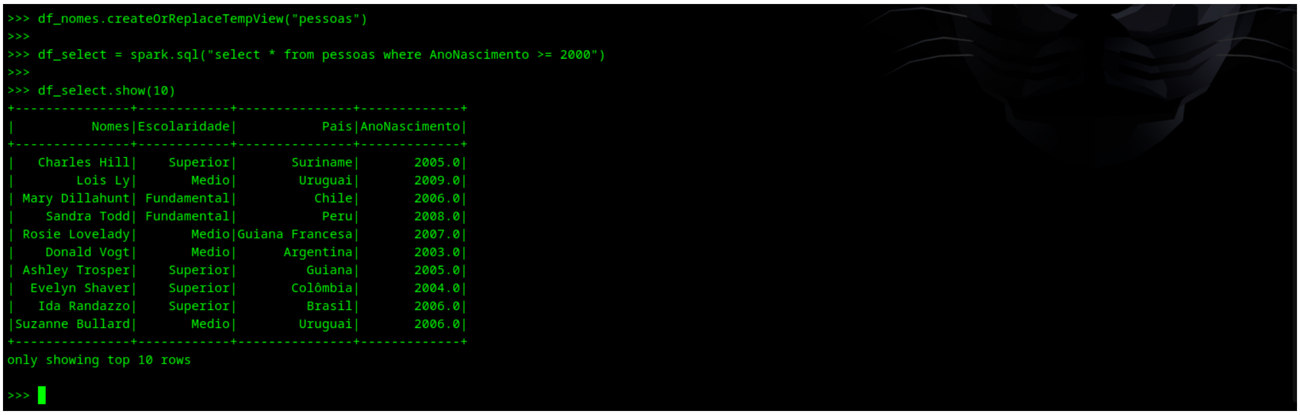
```
>>> from pyspark.sql.functions import udf
>>> from random import choice
>>>
>>> paises = ['Argentina', 'Bolívia', 'Brasil', 'Chile', 'Colômbia', 'Equador',
...           'Guiana', 'Paraguai', 'Peru', 'Suriname', 'Uruguai', 'Venezuela',
...           'Guiana Francesa']
>>>
>>> udf_pais = udf(lambda: choice(paises))
>>>
>>> df_nomes = df_nomes.withColumn('Pais', udf_pais())
>>>
>>> df_nomes.show(10)
+-----------------+------------+---------------+
|            Nomes|Escolaridade|           Pais|
+-----------------+------------+---------------+
|   Frances Bennet|    Superior|Guiana Francesa|
|    Jamie Russell|       Medio|      Argentina|
|   Edward Kistler|       Medio|        Equador|
|    Sheila Maurer| Fundamental|          Chile|
| Donald Golightly| Fundamental|      Venezuela|
|       David Gray|       Medio|        Uruguai|
|      Joy Bennett|    Superior|      Venezuela|
|      Paul Kriese| Fundamental|Guiana Francesa|
|Berniece Ornellas|    Superior|      Venezuela|
|    Brian Farrell|    Superior|        Bolívia|
+-----------------+------------+---------------+
only showing top 10 rows

>>> 
```



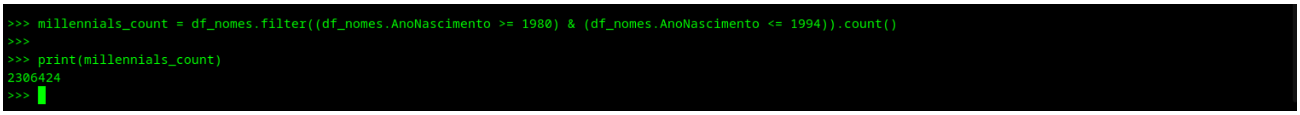
```
>>> from pyspark.sql.functions import round
>>>
>>> df_nomes = df_nomes.withColumn('AnoNascimento', round(rand()*65 + 1945))
>>>
>>> df_nomes.show(10)
+-----------------+------------+---------------+-------------+
|            Nomes|Escolaridade|           Pais|AnoNascimento|
+-----------------+------------+---------------+-------------+
|   Frances Bennet|    Superior|Guiana Francesa|       1958.0|
|    Jamie Russell|       Medio|      Argentina|       1994.0|
|   Edward Kistler|       Medio|        Equador|       1979.0|
|    Sheila Maurer| Fundamental|          Chile|       1972.0|
| Donald Golightly| Fundamental|      Venezuela|       1976.0|
|       David Gray|       Medio|        Uruguai|       1974.0|
|      Joy Bennett|    Superior|      Venezuela|       1984.0|
|      Paul Kriese| Fundamental|Guiana Francesa|       1973.0|
|Berniece Ornellas|    Superior|      Venezuela|       1966.0|
|    Brian Farrell|    Superior|        Bolívia|       1954.0|
+-----------------+------------+---------------+-------------+
only showing top 10 rows

>>> 
```



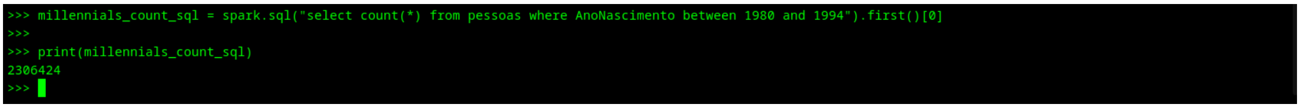
```
>>> df_select = df_nomes.filter(df_nomes.AnoNascimento >= 2000)
>>>
>>> df_select.show(10)
+---------------+------------+---------------+-------------+
|          Nomes|Escolaridade|           Pais|AnoNascimento|
+---------------+------------+---------------+-------------+
|   Charles Hill|    Superior|       Suriname|       2005.0|
|        Lois Ly|       Medio|        Uruguai|       2009.0|
| Mary Dillahunt| Fundamental|          Chile|       2006.0|
|    Sandra Todd| Fundamental|           Peru|       2008.0|
| Rosie Lovelady|       Medio|Guiana Francesa|       2007.0|
|    Donald Vogt|       Medio|      Argentina|       2003.0|
| Ashley Trosper|    Superior|         Guiana|       2005.0|
|  Evelyn Shaver|    Superior|       Colômbia|       2004.0|
|   Ida Randazzo|    Superior|         Brasil|       2006.0|
|Suzanne Bullard|       Medio|        Uruguai|       2006.0|
+---------------+------------+---------------+-------------+
only showing top 10 rows

>>> 
```



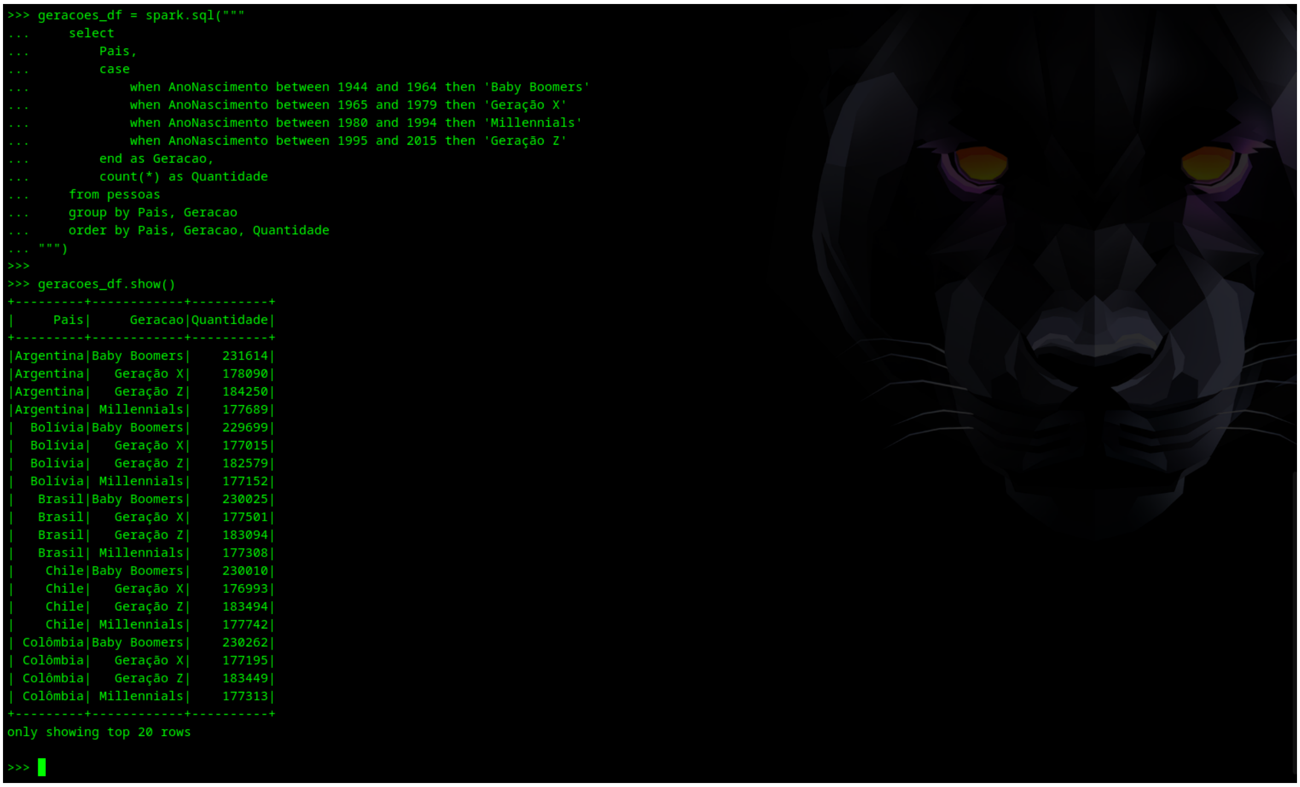
```
>>> df_nomes.createOrReplaceTempView("pessoas")
>>>
>>> df_select = spark.sql("select * from pessoas where AnoNascimento >= 2000")
>>>
>>> df_select.show(10)
+---------------+------------+---------------+-------------+
|          Nomes|Escolaridade|           Pais|AnoNascimento|
+---------------+------------+---------------+-------------+
|   Charles Hill|    Superior|       Suriname|       2005.0|
|        Lois Ly|       Medio|        Uruguai|       2009.0|
| Mary Dillahunt| Fundamental|          Chile|       2006.0|
|    Sandra Todd| Fundamental|           Peru|       2008.0|
| Rosie Lovelady|       Medio|Guiana Francesa|       2007.0|
|    Donald Vogt|       Medio|      Argentina|       2003.0|
| Ashley Trosper|    Superior|         Guiana|       2005.0|
|  Evelyn Shaver|    Superior|       Colômbia|       2004.0|
|   Ida Randazzo|    Superior|         Brasil|       2006.0|
|Suzanne Bullard|       Medio|        Uruguai|       2006.0|
+---------------+------------+---------------+-------------+
only showing top 10 rows

>>> 
```



```
>>> millennials_count = df_nomes.filter((df_nomes.AnoNascimento >= 1980) & (df_nomes.AnoNascimento <= 1994)).count()
>>>
>>> print(millennials_count)
2306424
>>>
```



```
>>> millennials_count_sql = spark.sql("select count(*) from pessoas where AnoNascimento between 1980 and 1994").first()[0]
>>>
>>> print(millennials_count_sql)
2306424
>>>
```



```
>>> geracoes_df = spark.sql("""
...     select
...         Pais,
...         case
...             when AnoNascimento between 1944 and 1964 then 'Baby Boomers'
...             when AnoNascimento between 1965 and 1979 then 'Geração X'
...             when AnoNascimento between 1980 and 1994 then 'Millennials'
...             when AnoNascimento between 1995 and 2015 then 'Geração Z'
...         end as Geracao,
...         count(*) as Quantidade
...     from pessoas
...     group by Pais, Geracao
...     order by Pais, Geracao, Quantidade
... """)
>>>
>>> geracoes_df.show()
+---------+------------+----------+
|     Pais|     Geracao|Quantidade|
+---------+------------+----------+
|Argentina|Baby Boomers|    231614|
|Argentina|   Geração X|    178090|
|Argentina|   Geração Z|    184250|
|Argentina| Millennials|    177689|
|  Bolivia|Baby Boomers|    229699|
|  Bolivia|   Geração X|    177015|
|  Bolivia|   Geração Z|    182579|
|  Bolivia| Millennials|    177152|
|   Brasil|Baby Boomers|    230025|
|   Brasil|   Geração X|    177501|
|   Brasil|   Geração Z|    183094|
|   Brasil| Millennials|    177308|
|    Chile|Baby Boomers|    230010|
|    Chile|   Geração X|    176993|
|    Chile|   Geração Z|    183494|
|    Chile| Millennials|    177742|
| Colômbia|Baby Boomers|    230262|
| Colômbia|   Geração X|    177195|
| Colômbia|   Geração Z|    183449|
| Colômbia| Millennials|    177313|
+---------+------------+----------+
only showing top 20 rows

>>>
```