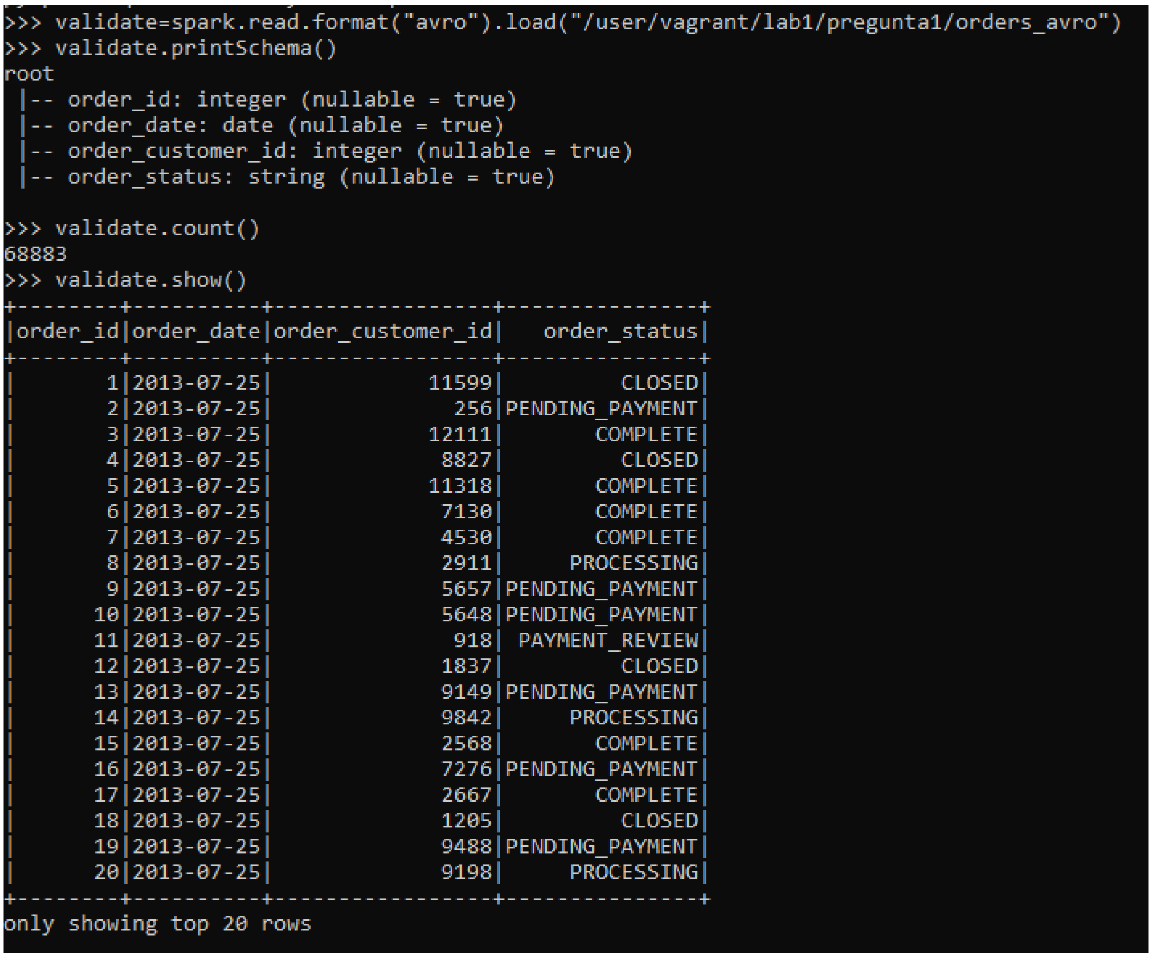
Taller de Preparación – Certificación Apache Spark Cloudera 2022

Nombre y Apellido: Javier Manuel Beltrán Reyes

Respuesta 1

```
from pyspark.sql.types import *

customSchema =
StructType([StructField("order_id",
IntegerType(),
True),StructField("order_date", DateType(),
True),StructField("order_customer_id",
IntegerType(),
True),StructField("order_status",
StringType(), True)])

orders=spark.read.option("inferSchema","true
").schema(customSchema).csv("/public/retail_
db/orders")

orders.write.format("avro").save("/user/vagr
ant/lab1/pregunta1/orders_avro")

validate=spark.read.format("avro").load("/us
er/vagrant/lab1/pregunta1/orders_avro")
validate.printSchema()
validate.count()
validate.show()
```



```
>>> validate=spark.read.format("avro").load("/user/vagrant/lab1/pregunta1/orders_avro")
>>> validate.printSchema()
root
 |-- order_id: integer (nullable = true)
 |-- order_date: date (nullable = true)
 |-- order_customer_id: integer (nullable = true)
 |-- order_status: string (nullable = true)

>>> validate.count()
68883
>>> validate.show()
+--------+----------+-----------------+---------------+
|order_id|order_date|order_customer_id|   order_status|
+--------+----------+-----------------+---------------+
|       1|2013-07-25|            11599|         CLOSED|
|       2|2013-07-25|              256|PENDING_PAYMENT|
|       3|2013-07-25|            12111|       COMPLETE|
|       4|2013-07-25|             8827|         CLOSED|
|       5|2013-07-25|            11318|       COMPLETE|
|       6|2013-07-25|             7130|       COMPLETE|
|       7|2013-07-25|             4530|       COMPLETE|
|       8|2013-07-25|             2911|     PROCESSING|
|       9|2013-07-25|             5657|PENDING_PAYMENT|
|      10|2013-07-25|             5648|PENDING_PAYMENT|
|      11|2013-07-25|              918| PAYMENT_REVIEW|
|      12|2013-07-25|             1837|         CLOSED|
|      13|2013-07-25|             9149|PENDING_PAYMENT|
|      14|2013-07-25|             9842|     PROCESSING|
|      15|2013-07-25|             2568|       COMPLETE|
|      16|2013-07-25|             7276|PENDING_PAYMENT|
|      17|2013-07-25|             2667|       COMPLETE|
|      18|2013-07-25|             1205|         CLOSED|
|      19|2013-07-25|             9488|PENDING_PAYMENT|
|      20|2013-07-25|             9198|     PROCESSING|
+--------+----------+-----------------+---------------+
only showing top 20 rows
```

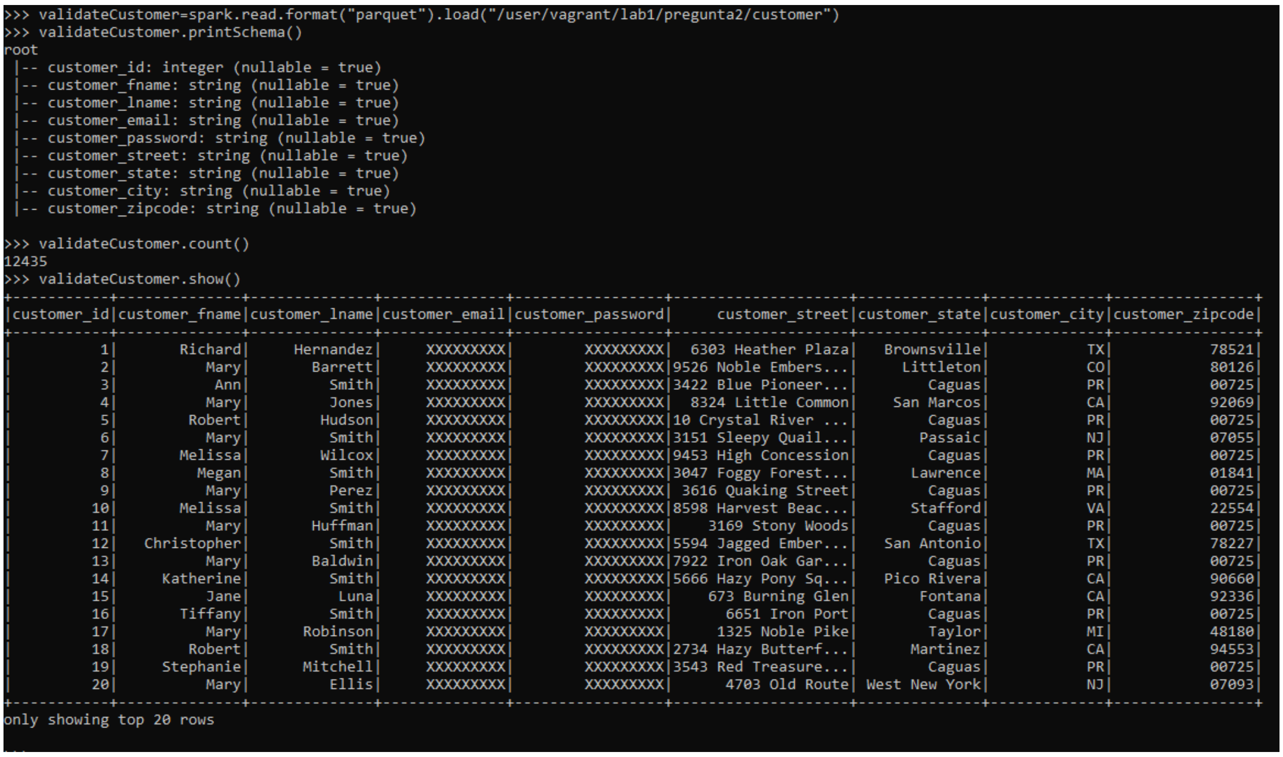
Respuesta 2

```
customerSchema =
StructType([StructField("customer_id",
IntegerType(),
True),StructField("customer_fname",
StringType(),
True),StructField("customer_lname",
StringType(),
True),StructField("customer_email",
StringType(),
True),StructField("customer_password",
StringType(),
True),StructField("customer_street",
StringType(),
True),StructField("customer_state",StringTyp
e(), True),StructField("customer_city",
StringType(),
True),StructField("customer_zipcode",
StringType(), True)])
```

```
customer=spark.read.schema(customerSchema).c
sv("/public/retail_db/customers/part-00000")
```

```
customer.write.format("parquet").save("/user
/vagrant/lab1/pregunta2/customer")
```

```
validateCustomer=spark.read.format("parquet"
).load("/user/vagrant/lab1/pregunta2/custome
r")
validateCustomer.printSchema()
validateCustomer.count()
validateCustomer.show()
```



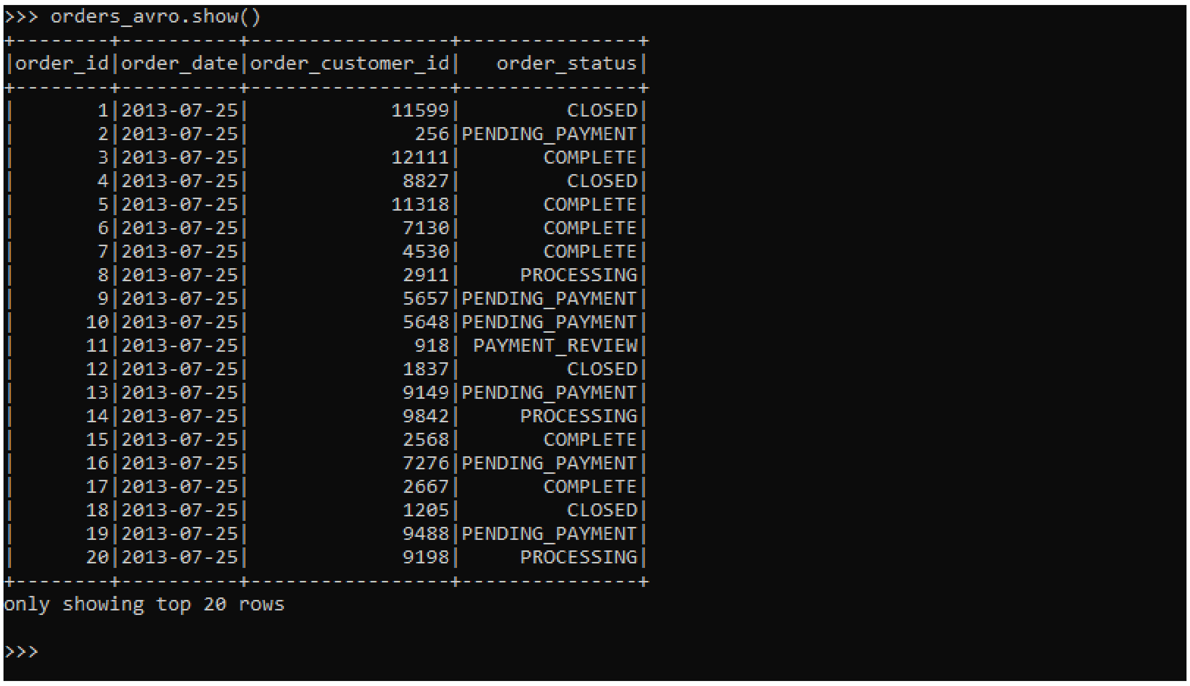
```
>>> validateCustomer=spark.read.format("parquet").load("/user/vagrant/lab1/pregunta2/customer")
>>> validateCustomer.printSchema()
root
 |-- customer_id: integer (nullable = true)
 |-- customer_fname: string (nullable = true)
 |-- customer_lname: string (nullable = true)
 |-- customer_email: string (nullable = true)
 |-- customer_password: string (nullable = true)
 |-- customer_street: string (nullable = true)
 |-- customer_state: string (nullable = true)
 |-- customer_city: string (nullable = true)
 |-- customer_zipcode: string (nullable = true)

>>> validateCustomer.count()
12435
>>> validateCustomer.show()
+-----------+--------------+--------------+--------------+-----------------+--------------------+--------------+-------------+----------------+
|customer_id|customer_fname|customer_lname|customer_email|customer_password|     customer_street|customer_state|customer_city|customer_zipcode|
+-----------+--------------+--------------+--------------+-----------------+--------------------+--------------+-------------+----------------+
|          1|       Richard|     Hernandez|     XXXXXXXXX|        XXXXXXXXX|  6303 Heather Plaza|   Brownsville|           TX|           78521|
|          2|          Mary|       Barrett|     XXXXXXXXX|        XXXXXXXXX|9526 Noble Embers...|     Littleton|           CO|           80126|
|          3|           Ann|         Smith|     XXXXXXXXX|        XXXXXXXXX|3422 Blue Pioneer...|        Caguas|           PR|           00725|
|          4|          Mary|         Jones|     XXXXXXXXX|        XXXXXXXXX|  8324 Little Common|    San Marcos|           CA|           92069|
|          5|        Robert|        Hudson|     XXXXXXXXX|        XXXXXXXXX|10 Crystal River ...|        Caguas|           PR|           00725|
|          6|          Mary|         Smith|     XXXXXXXXX|        XXXXXXXXX|3151 Sleepy Quail...|       Passaic|           NJ|           07055|
|          7|       Melissa|        Wilcox|     XXXXXXXXX|        XXXXXXXXX|9453 High Concession|        Caguas|           PR|           00725|
|          8|         Megan|         Smith|     XXXXXXXXX|        XXXXXXXXX|3047 Foggy Forest...|      Lawrence|           MA|           01841|
|          9|          Mary|         Perez|     XXXXXXXXX|        XXXXXXXXX| 3616 Quaking Street|        Caguas|           PR|           00725|
|         10|       Melissa|         Smith|     XXXXXXXXX|        XXXXXXXXX|8598 Harvest Beac...|      Stafford|           VA|           22554|
|         11|          Mary|       Huffman|     XXXXXXXXX|        XXXXXXXXX|    3169 Stony Woods|        Caguas|           PR|           00725|
|         12|   Christopher|         Smith|     XXXXXXXXX|        XXXXXXXXX|5594 Jagged Ember...|   San Antonio|           TX|           78227|
|         13|          Mary|       Baldwin|     XXXXXXXXX|        XXXXXXXXX|7922 Iron Oak Gar...|        Caguas|           PR|           00725|
|         14|     Katherine|         Smith|     XXXXXXXXX|        XXXXXXXXX|5666 Hazy Pony Sq...|   Pico Rivera|           CA|           90660|
|         15|          Jane|          Luna|     XXXXXXXXX|        XXXXXXXXX|    673 Burning Glen|       Fontana|           CA|           92336|
|         16|       Tiffany|         Smith|     XXXXXXXXX|        XXXXXXXXX|      6651 Iron Port|        Caguas|           PR|           00725|
|         17|          Mary|      Robinson|     XXXXXXXXX|        XXXXXXXXX|     1325 Noble Pike|        Taylor|           MI|           48180|
|         18|        Robert|         Smith|     XXXXXXXXX|        XXXXXXXXX|2734 Hazy Butterf...|      Martinez|           CA|           94553|
|         19|     Stephanie|      Mitchell|     XXXXXXXXX|        XXXXXXXXX|3543 Red Treasure...|        Caguas|           PR|           00725|
|         20|          Mary|         Ellis|     XXXXXXXXX|        XXXXXXXXX|      4703 Old Route| West New York|           NJ|           07093|
+-----------+--------------+--------------+--------------+-----------------+--------------------+--------------+-------------+----------------+
only showing top 20 rows
```

Respuesta 3

```
orders_avro=spark.read.format("avro").load("/user/vagrant/lab1/pregunta1/orders_avro/part-00000-e1522a05-76e7-4503-b6cf-65879531d383-c000.avro")
```

```
orders_avro.select("order_id","order_status").write.option("compression","gzip").parquet("/user/vagrant/lab1/pregunta3/resultado")
```

```
orders_avro.show()
```



```
>>> orders_avro.show()
+--------+----------+-----------------+---------------+
|order_id|order_date|order_customer_id|   order_status|
+--------+----------+-----------------+---------------+
|       1|2013-07-25|            11599|         CLOSED|
|       2|2013-07-25|              256|PENDING_PAYMENT|
|       3|2013-07-25|            12111|       COMPLETE|
|       4|2013-07-25|             8827|         CLOSED|
|       5|2013-07-25|            11318|       COMPLETE|
|       6|2013-07-25|             7130|       COMPLETE|
|       7|2013-07-25|             4530|       COMPLETE|
|       8|2013-07-25|             2911|     PROCESSING|
|       9|2013-07-25|             5657|PENDING_PAYMENT|
|      10|2013-07-25|             5648|PENDING_PAYMENT|
|      11|2013-07-25|              918| PAYMENT_REVIEW|
|      12|2013-07-25|             1837|         CLOSED|
|      13|2013-07-25|             9149|PENDING_PAYMENT|
|      14|2013-07-25|             9842|     PROCESSING|
|      15|2013-07-25|             2568|       COMPLETE|
|      16|2013-07-25|             7276|PENDING_PAYMENT|
|      17|2013-07-25|             2667|       COMPLETE|
|      18|2013-07-25|             1205|         CLOSED|
|      19|2013-07-25|             9488|PENDING_PAYMENT|
|      20|2013-07-25|             9198|     PROCESSING|
+--------+----------+-----------------+---------------+
only showing top 20 rows

>>>
```

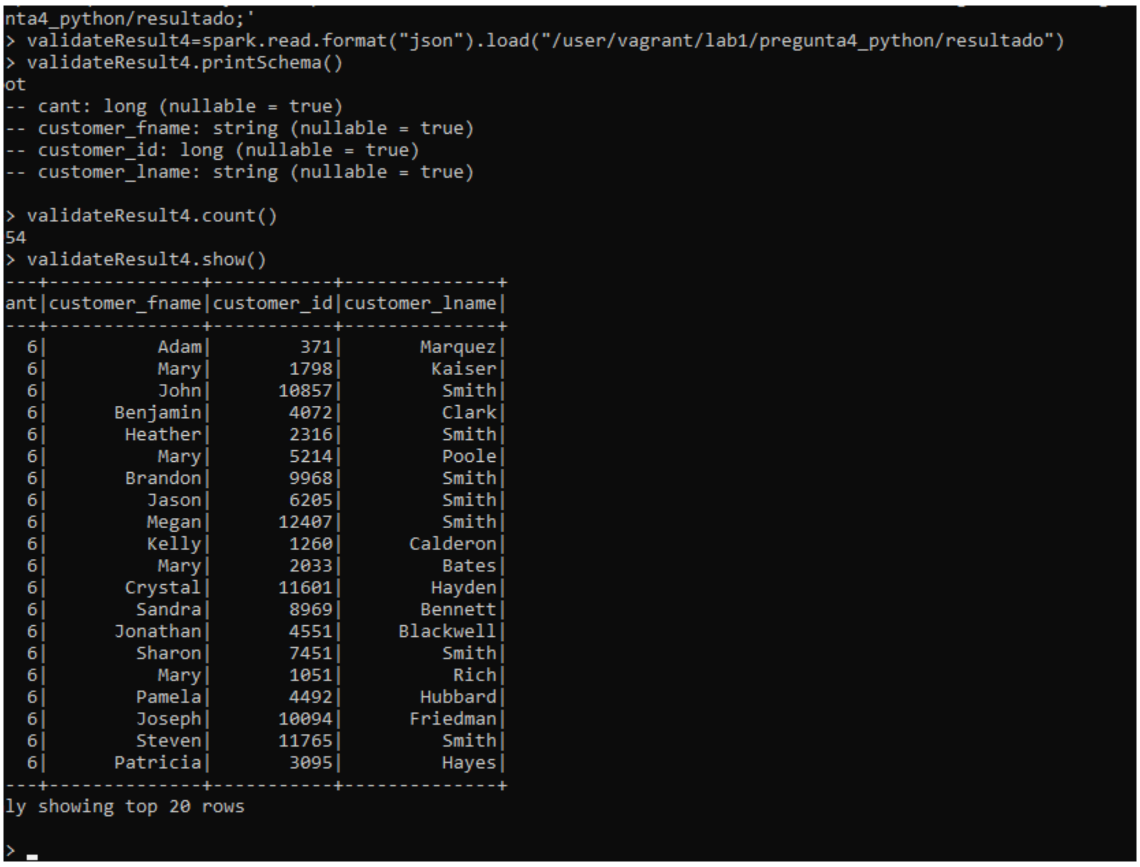
Respuesta 4

```
customer=spark.read.format("parquet").load("/user/vagrant/lab1/pregunta2/customer")
```

```
orders_avro=spark.read.format("avro").load("/user/vagrant/lab1/pregunta1/orders_avro/part-00000-e1522a05-76e7-4503-b6cf-65879531d383-c000.avro")
```

```
orders_avro.createOrReplaceTempView("orders_avro")
```

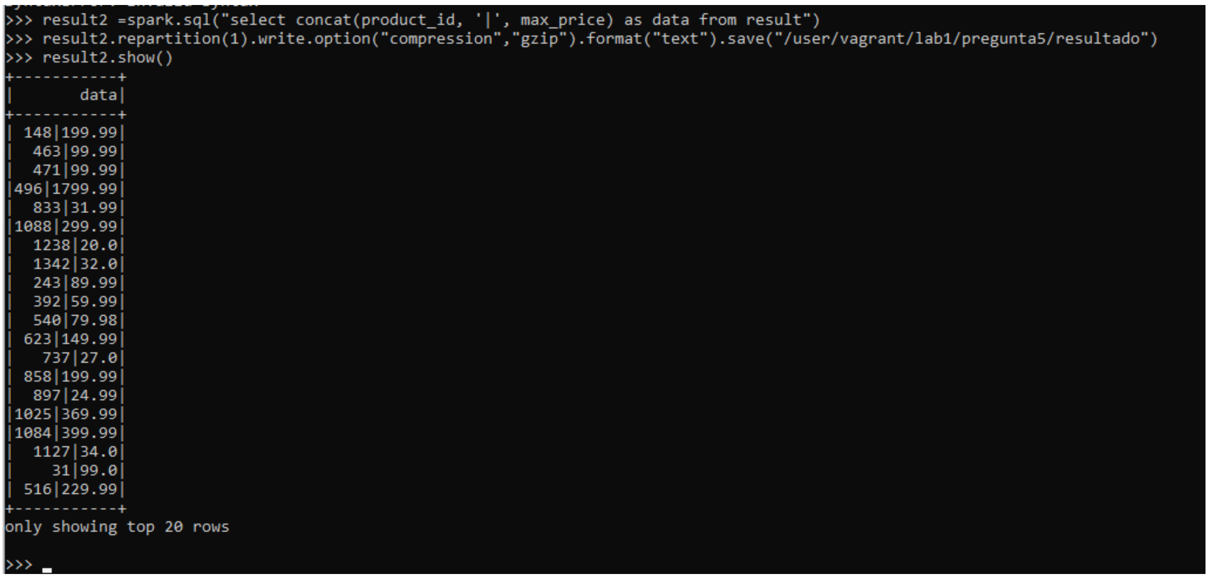
```
result=spark.sql("select
customer_id,customer_fname,customer_lname,co
unt(*) as cant from customer a inner join
orders_avro b on
a.customer_id=order_customer_id group by
customer_id,customer_lname,customer_fname
having count(*)>5")

result.sort(result.cant.desc()).write.format
("json").save("/user/vagrant/lab1/pregunta4_
python/resultado")

validateResult4=spark.read.format("json").lo
ad("/user/vagrant/lab1/pregunta4_python/resu
ltado")

validateResult4.printSchema()
validateResult4.count()
validateResult4.show()
```



```
nta4_python/resultado;'
> validateResult4=spark.read.format("json").load("/user/vagrant/lab1/pregunta4_python/resultado")
> validateResult4.printSchema()
ot
-- cant: long (nullable = true)
-- customer_fname: string (nullable = true)
-- customer_id: long (nullable = true)
-- customer_lname: string (nullable = true)

> validateResult4.count()
54
> validateResult4.show()
---+--------------+-----------+-------------+
ant|customer_fname|customer_id|customer_lname|
---+--------------+-----------+-------------+
  6|          Adam|        371|      Marquez|
  6|          Mary|       1798|       Kaiser|
  6|          John|      10857|        Smith|
  6|      Benjamin|       4072|        Clark|
  6|       Heather|       2316|        Smith|
  6|          Mary|       5214|        Poole|
  6|       Brandon|       9968|        Smith|
  6|         Jason|       6205|        Smith|
  6|         Megan|      12407|        Smith|
  6|         Kelly|       1260|     Calderon|
  6|          Mary|       2033|        Bates|
  6|       Crystal|      11601|       Hayden|
  6|        Sandra|       8969|      Bennett|
  6|      Jonathan|       4551|    Blackwell|
  6|        Sharon|       7451|        Smith|
  6|          Mary|       1051|         Rich|
  6|        Pamela|       4492|      Hubbard|
  6|        Joseph|      10094|     Friedman|
  6|        Steven|      11765|        Smith|
  6|      Patricia|       3095|        Hayes|
---+--------------+-----------+-------------+
ly showing top 20 rows

> _
```

Respuesta 5

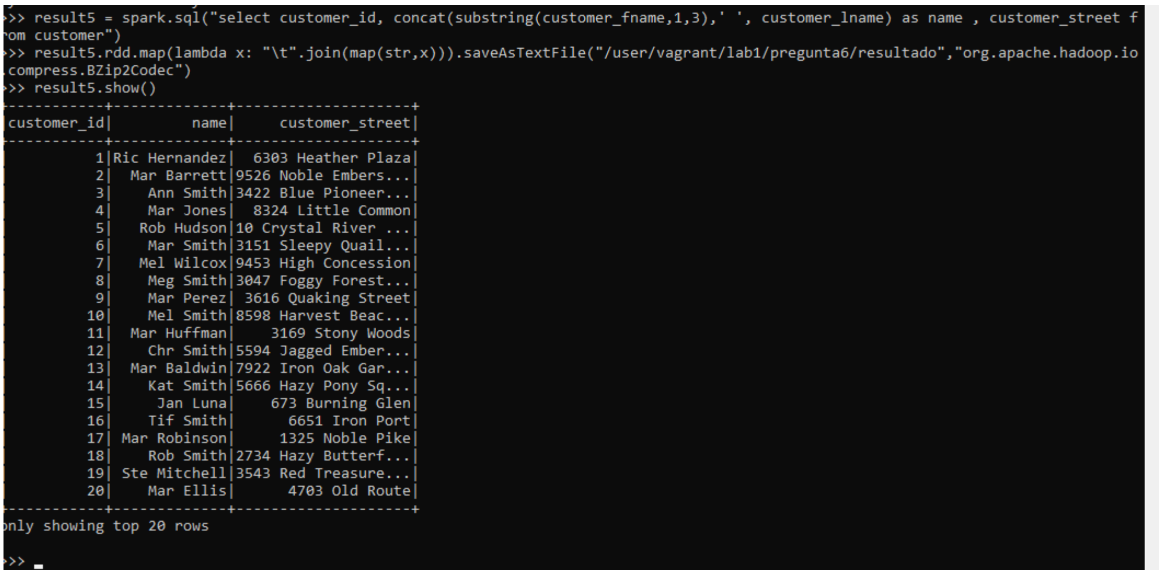
```
from pyspark.sql.types import *

ProductSchema=StructType([StructField("produ
ct_id",IntegerType(),True),StructField("prod
uct_category_id",IntegerType(),True),StructF
ield("product_name",StringType(),True),Struc
tField("product_description",StringType(),Tr
ue),StructField("product_price",FloatType(),
True),StructField("product_image",StringType
(),True)])

product =
spark.read.format("csv").option("inferSchema
","true").schema(ProductSchema).load("/publi
c/retail_db/products/part-00000")

product.createOrReplaceTempView("product")

result =spark.sql("select product_id,
max(product_price) as max_price from product
group by product_id")

result.createOrReplaceTempView("result")

result2 =spark.sql("select
concat(product_id, '|', max_price) as data
from result")

result2.repartition(1).write.option("compres
sion","gzip").format("text").save("/user/vag
rant/lab1/pregunta5/resultado")

result2.show()
```



```
>>> result2 =spark.sql("select concat(product_id, '|', max_price) as data from result")
>>> result2.repartition(1).write.option("compression","gzip").format("text").save("/user/vagrant/lab1/pregunta5/resultado")
>>> result2.show()
+-----------+
|       data|
+-----------+
| 148|199.99|
|  463|99.99|
|  471|99.99|
|496|1799.99|
|  833|31.99|
|1088|299.99|
|  1238|20.0|
|  1342|32.0|
|  243|89.99|
|  392|59.99|
|  540|79.98|
| 623|149.99|
|   737|27.0|
| 858|199.99|
|  897|24.99|
|1025|369.99|
|1084|399.99|
|  1127|34.0|
|    31|99.0|
| 516|229.99|
+-----------+
only showing top 20 rows

>>> _
```

Respuesta 6

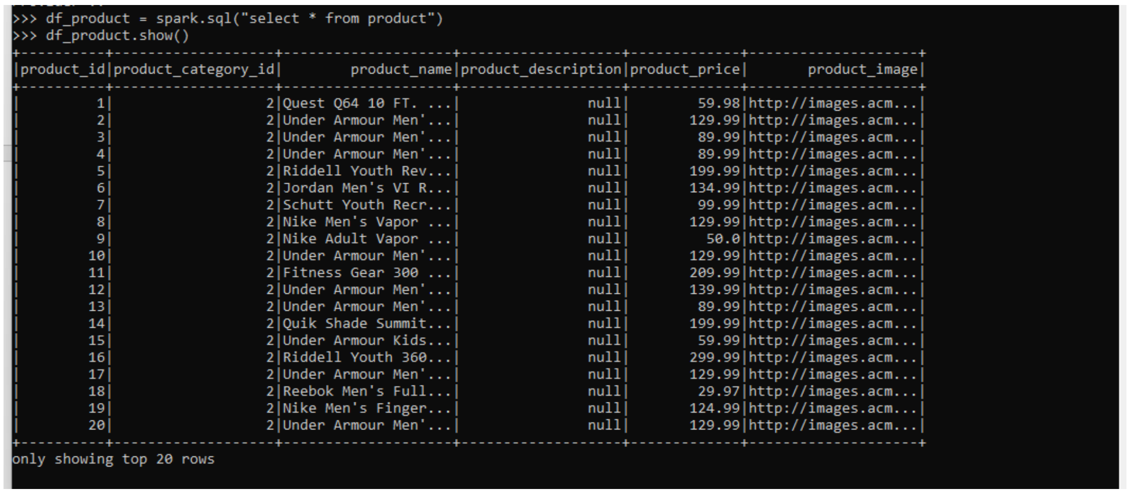
```
customer=spark.read.format("parquet").load("/user/vagrant/lab1/pregunta2/customer")

customer.createOrReplaceTempView("customer")

result5 = spark.sql("select customer_id, concat(substring(customer_fname,1,3),' ', customer_lname) as name , customer_street from customer")

result5.rdd.map(lambda x: "\t".join(map(str,x))).saveAsTextFile("/user/vagrant/lab1/pregunta6/resultado","org.apache.hadoop.io.compress.BZip2Codec")

result5.show()
```



```
>> result5 = spark.sql("select customer_id, concat(substring(customer_fname,1,3),' ', customer_lname) as name , customer_street f
om customer")
>> result5.rdd.map(lambda x: "\t".join(map(str,x))).saveAsTextFile("/user/vagrant/lab1/pregunta6/resultado","org.apache.hadoop.io
.compress.BZip2Codec")
>> result5.show()
-----------+------------+-------------------+
customer_id|        name|    customer_street|
-----------+------------+-------------------+
          1|Ric Hernandez|  6303 Heather Plaza|
          2|  Mar Barrett|9526 Noble Embers...|
          3|    Ann Smith|3422 Blue Pioneer...|
          4|    Mar Jones|  8324 Little Common|
          5|   Rob Hudson|10 Crystal River ...|
          6|    Mar Smith|3151 Sleepy Quail...|
          7|   Mel Wilcox|9453 High Concession|
          8|    Meg Smith|3047 Foggy Forest...|
          9|    Mar Perez| 3616 Quaking Street|
         10|    Mel Smith|8598 Harvest Beac...|
         11|  Mar Huffman|    3169 Stony Woods|
         12|    Chr Smith|5594 Jagged Ember...|
         13|  Mar Baldwin|7922 Iron Oak Gar...|
         14|    Kat Smith|5666 Hazy Pony Sq...|
         15|     Jan Luna|    673 Burning Glen|
         16|    Tif Smith|      6651 Iron Port|
         17| Mar Robinson|     1325 Noble Pike|
         18|    Rob Smith|2734 Hazy Butterf...|
         19| Ste Mitchell|3543 Red Treasure...|
         20|    Mar Ellis|      4703 Old Route|
-----------+------------+-------------------+
nly showing top 20 rows

>> _
```

Respuesta 7

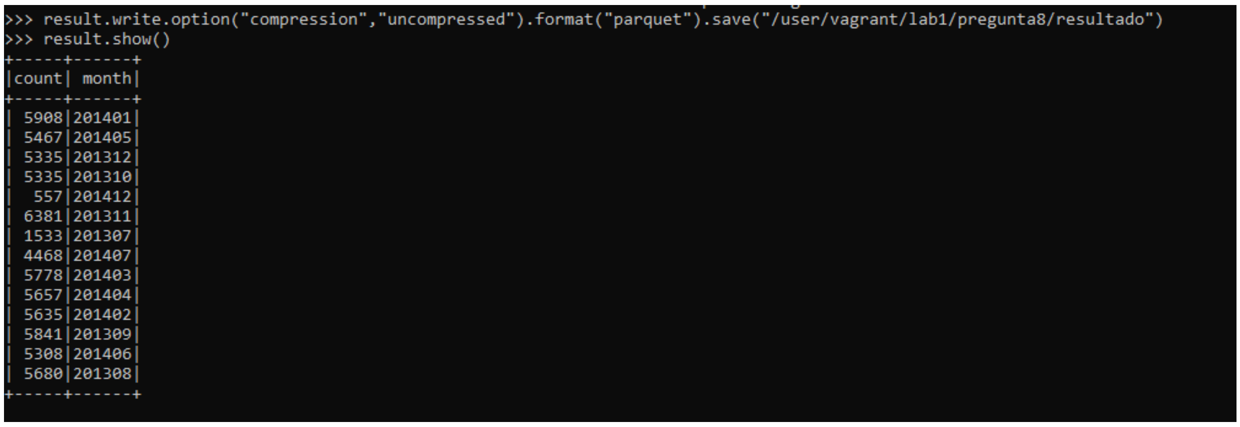
```
product =
spark.read.format("csv").option("inferSchema
","true").schema(ProductSchema).load("/publi
c/retail_db/products/part-00000")

product.write.format("hive").saveAsTable("pr
oduct")

df_product = spark.sql("select * from
product")

df_product.show()
```



```
>>> df_product = spark.sql("select * from product")
>>> df_product.show()
+----------+-------------------+--------------------+-------------------+-------------+--------------------+
|product_id|product_category_id|        product_name|product_description|product_price|       product_image|
+----------+-------------------+--------------------+-------------------+-------------+--------------------+
|         1|                  2|Quest Q64 10 FT. ...|               null|        59.98|http://images.acm...|
|         2|                  2|Under Armour Men'...|               null|       129.99|http://images.acm...|
|         3|                  2|Under Armour Men'...|               null|        89.99|http://images.acm...|
|         4|                  2|Under Armour Men'...|               null|        89.99|http://images.acm...|
|         5|                  2|Riddell Youth Rev...|               null|       199.99|http://images.acm...|
|         6|                  2|Jordan Men's VI R...|               null|       134.99|http://images.acm...|
|         7|                  2|Schutt Youth Recr...|               null|        99.99|http://images.acm...|
|         8|                  2|Nike Men's Vapor ...|               null|       129.99|http://images.acm...|
|         9|                  2|Nike Adult Vapor ...|               null|         50.0|http://images.acm...|
|        10|                  2|Under Armour Men'...|               null|       129.99|http://images.acm...|
|        11|                  2|Fitness Gear 300 ...|               null|       209.99|http://images.acm...|
|        12|                  2|Under Armour Men'...|               null|       139.99|http://images.acm...|
|        13|                  2|Under Armour Men'...|               null|        89.99|http://images.acm...|
|        14|                  2|Quik Shade Summit...|               null|       199.99|http://images.acm...|
|        15|                  2|Under Armour Kids...|               null|        59.99|http://images.acm...|
|        16|                  2|Riddell Youth 360...|               null|       299.99|http://images.acm...|
|        17|                  2|Under Armour Men'...|               null|       129.99|http://images.acm...|
|        18|                  2|Reebok Men's Full...|               null|        29.97|http://images.acm...|
|        19|                  2|Nike Men's Finger...|               null|       124.99|http://images.acm...|
|        20|                  2|Under Armour Men'...|               null|       129.99|http://images.acm...|
+----------+-------------------+--------------------+-------------------+-------------+--------------------+
only showing top 20 rows
```

Respuesta 8

```
from pyspark.sql.types import *

OrdersSchema=StructType([StructField("order_
id",IntegerType(),True),StructField("order_d
ate
",DateType(),True),StructField("order_custom
er_id",IntegerType(),True),StructField("orde
r_status",StringType(),True)])

orders=spark.read.format("csv").option("infe
rSchema",
"true").schema(OrdersSchema).load("/public/r
etail_db/orders/part-00000")

orders.write.format("hive").saveAsTable("ord
ers")

result=spark.sql("select count(*) as
count,date_format(order_date,'YYYYMM') as
month from orders group by
date_format(order_date, 'YYYYMM')")

result.write.option("compression","uncompres
sed").format("parquet").save("/user/vagrant/
lab1/pregunta8/resultado")

result.show()
```



```
>>> result.write.option("compression","uncompressed").format("parquet").save("/user/vagrant/lab1/pregunta8/resultado")
>>> result.show()
+-----+------+
|count| month|
+-----+------+
| 5908|201401|
| 5467|201405|
| 5335|201312|
| 5335|201310|
|  557|201412|
| 6381|201311|
| 1533|201307|
| 4468|201407|
| 5778|201403|
| 5657|201404|
| 5635|201402|
| 5841|201309|
| 5308|201406|
| 5680|201308|
+-----+------+
```

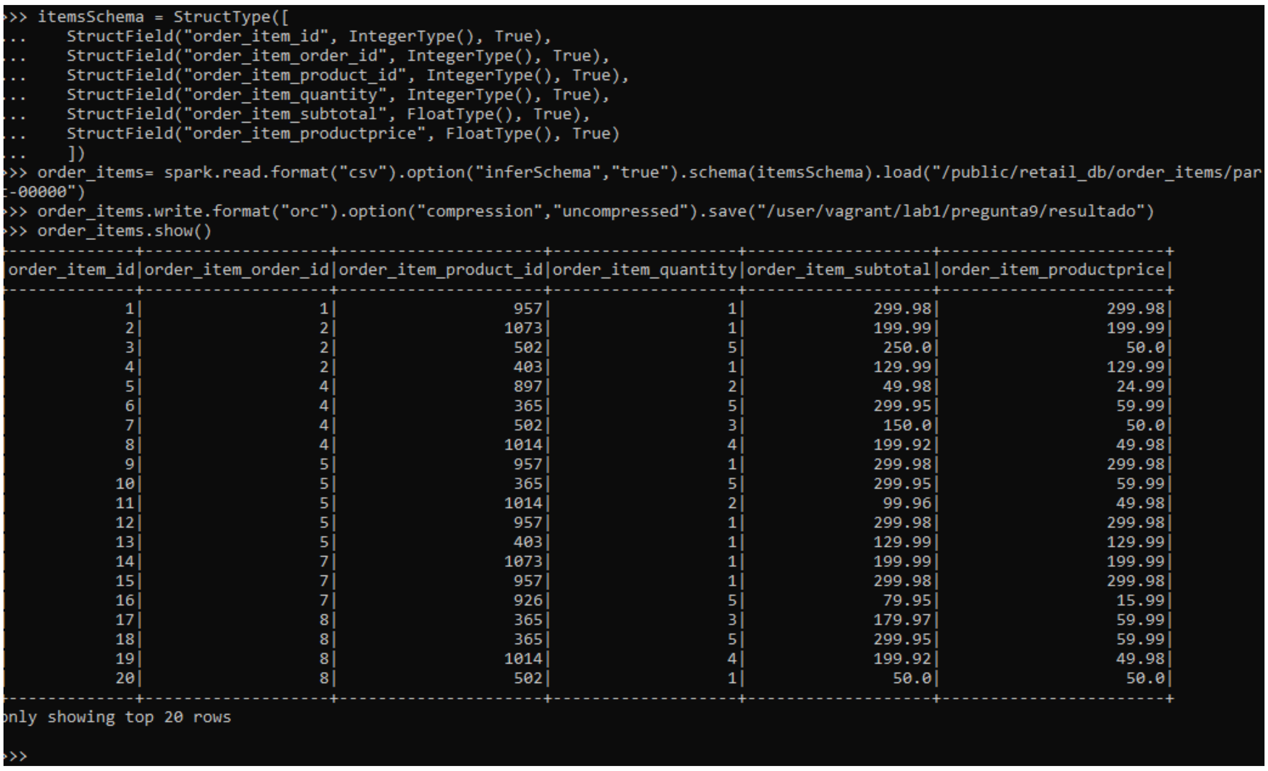
Respuesta 9

```
itemsSchema = StructType([
   StructField("order_item_id",
IntegerType(), True),
   StructField("order_item_order_id",
IntegerType(), True),
   StructField("order_item_product_id",
IntegerType(), True),
   StructField("order_item_quantity",
IntegerType(), True),
   StructField("order_item_subtotal",
FloatType(), True),
   StructField("order_item_productprice",
FloatType(), True)
   ])
```

```
order_items=
spark.read.format("csv").option("inferSchema
","true").schema(itemsSchema).load("/public/
retail_db/order_items/part-00000")
```

```
order_items.write.format("orc").option("comp
ression","uncompressed").save("/user/vagrant
/lab1/pregunta9/resultado")
```

```
order_items.show()
```



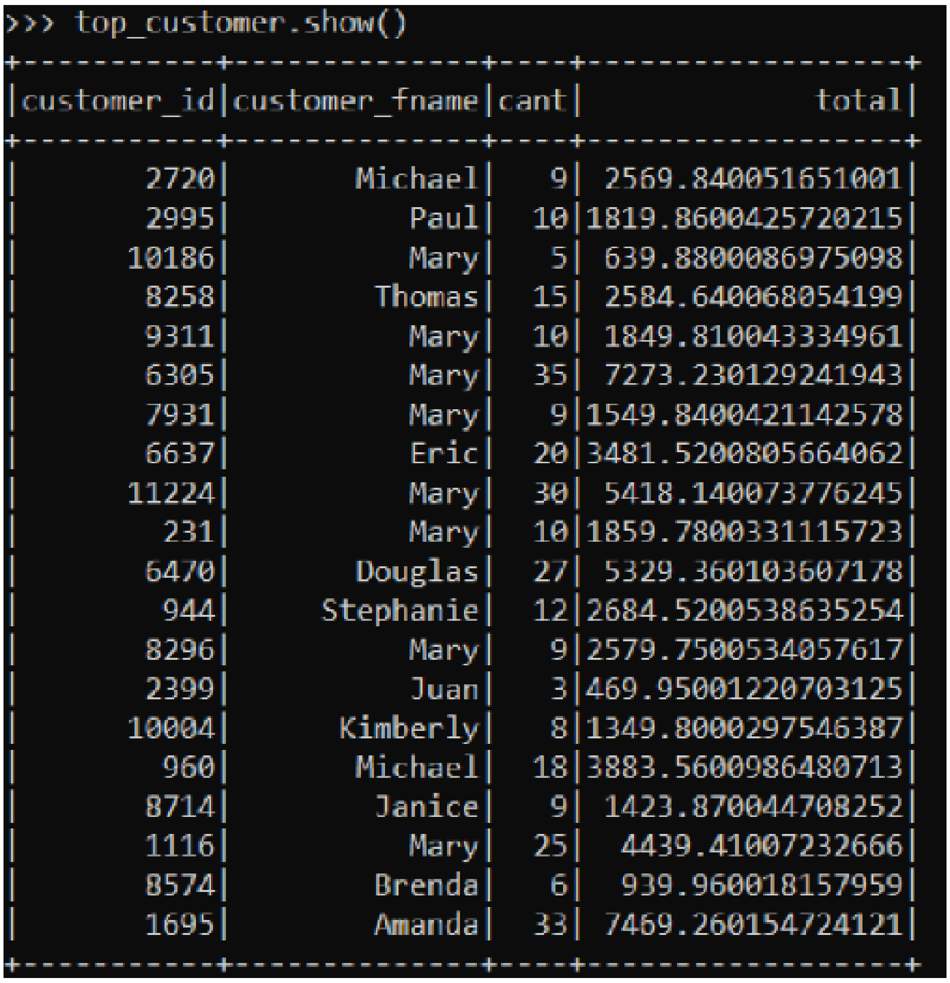
```
>> itemsSchema = StructType([
..     StructField("order_item_id", IntegerType(), True),
..     StructField("order_item_order_id", IntegerType(), True),
..     StructField("order_item_product_id", IntegerType(), True),
..     StructField("order_item_quantity", IntegerType(), True),
..     StructField("order_item_subtotal", FloatType(), True),
..     StructField("order_item_productprice", FloatType(), True)
..     ])
>> order_items= spark.read.format("csv").option("inferSchema","true").schema(itemsSchema).load("/public/retail_db/order_items/par
-00000")
>> order_items.write.format("orc").option("compression","uncompressed").save("/user/vagrant/lab1/pregunta9/resultado")
>> order_items.show()
-------------+-------------------+---------------------+-------------------+-------------------+-----------------------+
|order_item_id|order_item_order_id|order_item_product_id|order_item_quantity|order_item_subtotal|order_item_productprice|
-------------+-------------------+---------------------+-------------------+-------------------+-----------------------+
|            1|                  1|                  957|                  1|             299.98|                 299.98|
|            2|                  2|                 1073|                  1|             199.99|                 199.99|
|            3|                  2|                  502|                  5|              250.0|                   50.0|
|            4|                  2|                  403|                  1|             129.99|                 129.99|
|            5|                  4|                  897|                  2|              49.98|                  24.99|
|            6|                  4|                  365|                  5|             299.95|                  59.99|
|            7|                  4|                  502|                  3|              150.0|                   50.0|
|            8|                  4|                 1014|                  4|             199.92|                  49.98|
|            9|                  5|                  957|                  1|             299.98|                 299.98|
|           10|                  5|                  365|                  5|             299.95|                  59.99|
|           11|                  5|                 1014|                  2|              99.96|                  49.98|
|           12|                  5|                  957|                  1|             299.98|                 299.98|
|           13|                  5|                  403|                  1|             129.99|                 129.99|
|           14|                  7|                 1073|                  1|             199.99|                 199.99|
|           15|                  7|                  957|                  1|             299.98|                 299.98|
|           16|                  7|                  926|                  5|              79.95|                  15.99|
|           17|                  8|                  365|                  3|             179.97|                  59.99|
|           18|                  8|                  365|                  5|             299.95|                  59.99|
|           19|                  8|                 1014|                  4|             199.92|                  49.98|
|           20|                  8|                  502|                  1|               50.0|                   50.0|
-------------+-------------------+---------------------+-------------------+-------------------+-----------------------+
nly showing top 20 rows

>>
```

Respuesta 10

```
customer=spark.read.format("parquet").load("/user/vagrant/lab1/pregunta2/customer")

customer.createOrReplaceTempView("customer")

order_items=spark.read.format("orc").load("/user/vagrant/lab1/pregunta9/resultado")

order_items.createOrReplaceTempView("order_items")

orders=spark.read.format("avro").load("/user/vagrant/lab1/pregunta1/orders_avro")

orders.createOrReplaceTempView("orders")


top_customer = spark.sql("select
customer_id,customer_fname,count(*) as
cant,sum(order_item_subtotal) as total
from customer a inner join orders b  on
a.customer_id = b.order_customer_id inner
join order_items c on c.order_item_order_id
= b.order_id where customer_city like 'M%'
group by customer_id,customer_fname")
```

```
top_customer.write.format("parquet").option(
"compression","gzip").save("user/vagrant/lab
1/pregunta10/resultado")
```

```
top_customer.show()
```



```
>>> top_customer.show()
+-----------+--------------+----+------------------+
|customer_id|customer_fname|cant|             total|
+-----------+--------------+----+------------------+
|       2720|       Michael|   9| 2569.840051651001|
|       2995|          Paul|  10|1819.8600425720215|
|      10186|          Mary|   5| 639.8800086975098|
|       8258|        Thomas|  15| 2584.640068054199|
|       9311|          Mary|  10| 1849.810043334961|
|       6305|          Mary|  35| 7273.230129241943|
|       7931|          Mary|   9|1549.8400421142578|
|       6637|          Eric|  20|3481.5200805664062|
|      11224|          Mary|  30| 5418.140073776245|
|        231|          Mary|  10|1859.7800331115723|
|       6470|       Douglas|  27| 5329.360103607178|
|        944|     Stephanie|  12|2684.5200538635254|
|       8296|          Mary|   9|2579.7500534057617|
|       2399|          Juan|   3|469.95001220703125|
|      10004|      Kimberly|   8|1349.8000297546387|
|        960|       Michael|  18|3883.5600986480713|
|       8714|        Janice|   9| 1423.870044708252|
|       1116|          Mary|  25|  4439.41007232666|
|       8574|        Brenda|   6|  939.960018157959|
|       1695|        Amanda|  33| 7469.260154724121|
+-----------+--------------+----+------------------+
```