Project Report on
RETAIL DATASET

INDEX

DATA LOADING

The breakdown of data preparation is as follows.

1. Data loading
    a. Data loading from HDFS into Spark
    b. Data loading from S3 into Spark
    c. Data loading from MySql into Spark
    d. Data loading from Local file into Spark
2. Data cleaning (removing null and duplicate values)

Division of data sources is as follows.

HDFS	**S3**	**MySQL**	**Local**	**Kafka**
Linitem.csv	Customer.csv	Supplier.csv	Region.csv	Orders.csv
Partsupp.csv	Nation.csv	Part.csv		

Data loading from HDFS into Spark

Step 1: Make a new directory 'retail' to store the datasets.

```
hdfs dfs -mkdir /retail
```

Step 2: Copy the csv files into HDFS

```
hdfs dfs -copyFromLocal /home/ak/Downloads/Project_Gladiator/Retail/lineitem.csv /retail/
```

Confirm the file is copied:

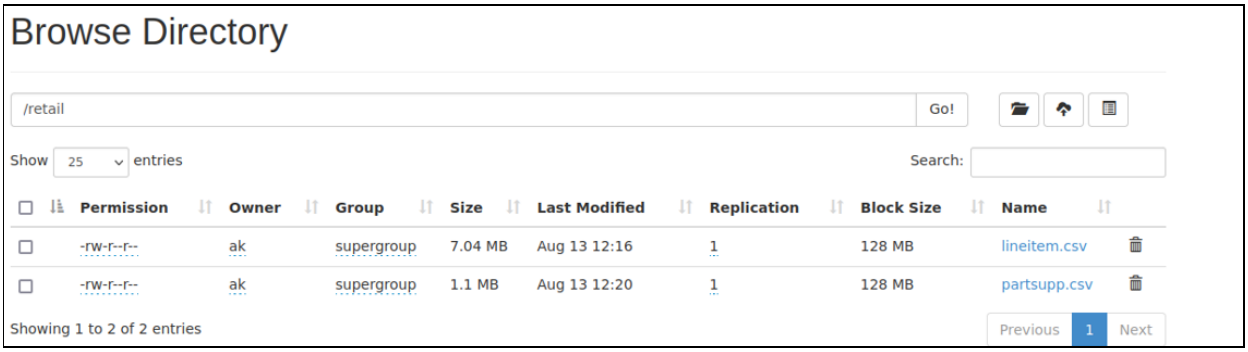
```
hdfs dfs -ls /retail
```

```
hdfs dfs -copyFromLocal /home/ak/Downloads/Project_Gladiator/Retail/partsupp.csv /retail/
```

Confirm the file is copied:

```
hdfs dfs -ls /retail
```

Lineitem.csv and partsupp.csv added to retail folder in HDFS :



Step 3: Defining the schema for the Lineitem.csv and Partsupp.csv

```
from pyspark.sql.types import IntegerType, DateType, StringType, DoubleType, StructType,
StructField

LineitemSchema = StructType([ \
        StructField('L_ORDERKEY', IntegerType(), True), \
        StructField('L_PARTKEY', IntegerType(), True), \
        StructField('L_SUPPKEY', IntegerType(), True), \
        StructField('L_LINENUMBER', IntegerType(), True), \
        StructField('L_QUANTITY', DoubleType(), True), \
        StructField('L_EXTENDEDPRICE', DoubleType(), True), \
        StructField('L_DISCOUNT', DoubleType(), True), \
        StructField('L_TAX', DoubleType(), True), \
        StructField('L_RETURNFLAG', StringType(), True), \
        StructField('L_LINESTATUS', StringType(), True), \
        StructField('L_SHIPDATE', DateType(), True), \
        StructField('L_COMMITDATE', DateType(), True), \
        StructField('L_RECEIPTDATE', DateType(), True), \
        StructField('L_SHIPINSTRUCT', StringType(), True), \
        StructField('L_SHIPMODE', StringType(), True), \
        StructField('L_COMMENT', StringType(), True) \
])


lineitem = spark.read.schema(LineitemSchema).option('header',
'true').csv('hdfs:///retail/lineitem.csv', sep='\t')

lineitem.show(5)
```

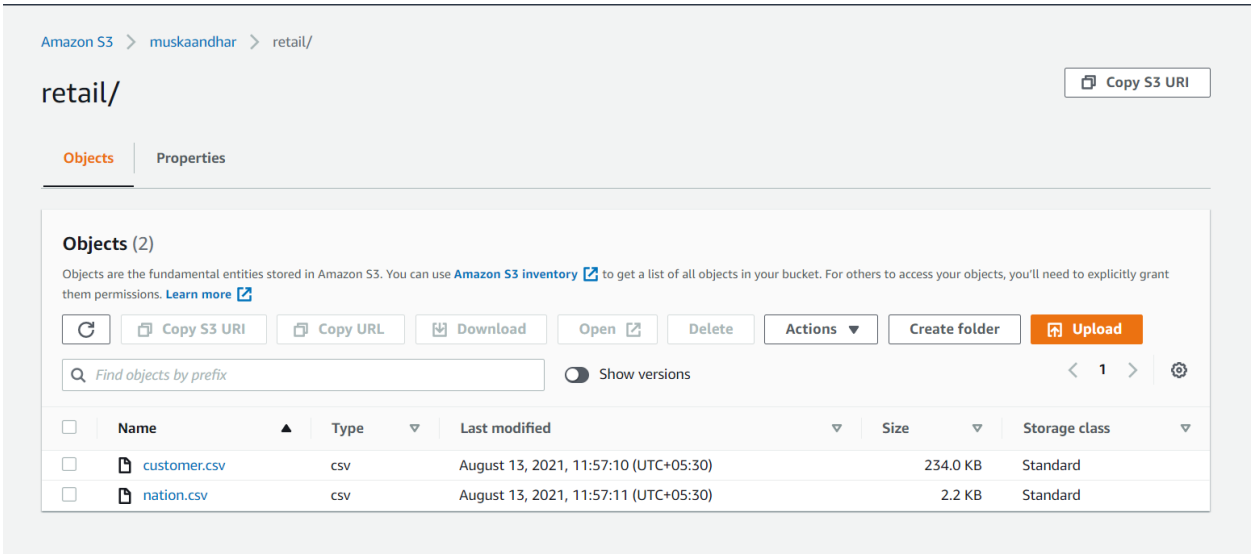
```
>>> lineitem.show(3)
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+---
----------+----------------+----------+--------------------+
|L_ORDERKEY|L_PARTKEY|L_SUPPKEY|L_LINENUMBER|L_QUANTITY|L_EXTENDEDPRICE|L_DISCOUNT|L_TAX|L_RETURNFLAG|L_LINESTATUS|L_SHIPDATE|L_COMMITDATE|L_R
ECEIPTDATE|   L_SHIPINSTRUCT|L_SHIPMODE|          L_COMMENT|
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+---
----------+----------------+----------+--------------------+
|         1|     1552|       93|           1|      17.0|       24710.35|      0.04| 0.02|           N|           O|1996-03-13|  1996-02-12|
1996-03-22|DELIVER IN PERSON|     TRUCK|egular courts abo...|
|         1|      674|       75|           2|      36.0|       56688.12|      0.09| 0.06|           N|           O|1996-04-12|  1996-02-28|
1996-04-20| TAKE BACK RETURN|      MAIL|ly final dependen...|
|         1|      637|       38|           3|       8.0|       12301.04|       0.1| 0.02|           N|           O|1996-01-29|  1996-03-05|
1996-01-31| TAKE BACK RETURN|   REG AIR|riously. regular,...|
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+---
----------+----------------+----------+--------------------+
only showing top 3 rows
```

```
PartsuppSchema = StructType([ \
        StructField('PS_PARTKEY', IntegerType(), True), \
        StructField('PS_SUPPKEY', IntegerType(), True), \
        StructField('PS_AVAILQTY', IntegerType(), True), \
        StructField('PS_SUPPLYCOST', DoubleType(), True), \
        StructField('PS_COMMENT', StringType(), True) \
])

partsupp =
spark.read.schema(PartsuppSchema).option('header','true').csv('hdfs:///retail/partsupp.csv',
sep='\t')

partsupp.printSchema()
partsupp.show(5)
```

```
+----------+----------+-----------+-------------+--------------------+
|PS_PARTKEY|PS_SUPPKEY|PS_AVAILQTY|PS_SUPPLYCOST|          PS_COMMENT|
+----------+----------+-----------+-------------+--------------------+
|         1|         2|       3325|       771.64|, even theodolite...|
|         1|        27|       8076|       993.49|ven ideas. quickl...|
|         1|        52|       3956|       337.09|after the fluffil...|
|         1|        77|       4069|       357.84|al, regular depen...|
|         2|         3|       8895|       378.49|nic accounts. fin...|
+----------+----------+-----------+-------------+--------------------+
```

Data loading from S3 into Spark

Step 1: Upload the documents in the S3 bucket.



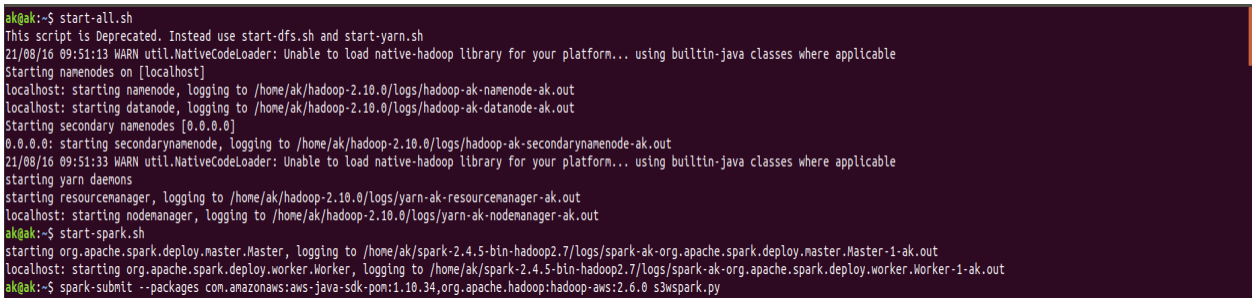
Step 2: Writing the code

```
from pyspark.sql import SparkSession

if __name__ == "__main__":
    spark = SparkSession\
        .builder\
        .appName("Connector")\
      .enableHiveSupport() \
        .getOrCreate()
    spark.sparkContext.setLogLevel('ERROR')

    access_id = (AWS Access Key)
    secretAccessKey = (AWS Secret Access Key)

    hadoop_conf = spark._jsc.hadoopConfiguration()

    hadoop_conf.set("fs.s3.awsAccessKeyId", access_id)
    hadoop_conf.set("fs.s3n.awsAccessKeyId", access_id)
    spark._jsc.hadoopConfiguration().set("fs.s3a.access.key", access_id)
    spark._jsc.hadoopConfiguration().set("fs.s3.awsSecretAccessKey",
secretAccessKey)
    spark._jsc.hadoopConfiguration().set("fs.s3n.awsSecretAccess.key",
secretAccessKey)
```

```
    spark._jsc.hadoopConfiguration().set("fs.s3a.secret.key",
secretAccessKey)
    spark._jsc.hadoopConfiguration().set("fs.s3a.access.key", access_id)
    spark._jsc.hadoopConfiguration().set("fs.s3n.impl",
"org.apache.hadoop.fs.s3native.NativeS3FileSystem")
    spark._jsc.hadoopConfiguration().set("fs.s3a.impl",
"org.apache.hadoop.fs.s3a.S3AFileSystem")
    spark._jsc.hadoopConfiguration().set("fs.s3.impl",
"org.apache.hadoop.fs.s3.S3FileSystem")


    filePath1 = "s3://muskaandhar/retail/customer.csv"
    df1 =
spark.read.option('inferschema','true').option('header','true').option('se
p','\t').csv(filePath1)

    df1.show()
    filePath2 = "s3://muskaandhar/retail/nation.csv"
    df2 =
spark.read.option('inferschema','true').option('header','true').option('se
p','\t').csv(filePath2)

    df2.show()
```

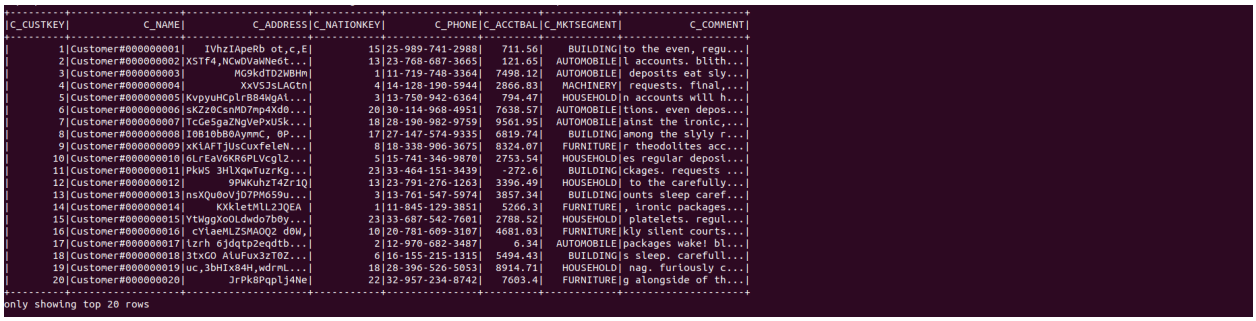
Step 3: Starting all Hadoop daemons

start-all.sh
start-spark.sh



Step 4: Run the python file containing the code with the following command:

spark-submit --packages
com.amazonaws:aws-java-sdk-pom:1.10.34,org.apache.hadoop:hadoop-aws:2.6.0 s3wspark.py



```
+---------+------------------+--------------------+----------+--------------+--------+------------+--------------------+
|C_CUSTKEY|            C_NAME|           C_ADDRESS|C_NATIONKEY|       C_PHONE|C_ACCTBAL|C_MKTSEGMENT|           C_COMMENT|
+---------+------------------+--------------------+----------+--------------+--------+------------+--------------------+
|        1|Customer#000000001|   IVhzIApeRb ot,c,E|        15|25-989-741-2988|   711.56|    BUILDING|to the even, regu...|
|        2|Customer#000000002|XSTf4,NCwDVaWNe6t...|        13|23-768-687-3665|   121.65|  AUTOMOBILE|l accounts. blith...|
|        3|Customer#000000003|        MC9kdTD2WBHm|         1|11-719-748-3364|  7498.12|  AUTOMOBILE| deposits eat sly...|
|        4|Customer#000000004|          XxVSJsLAGtn|         4|14-128-190-5944|  2866.83|   MACHINERY| requests. final,...|
|        5|Customer#000000005|KvpyuHCplrB84WgAi...|         3|13-750-942-6364|   794.47|   HOUSEHOLD|n accounts will h...|
|        6|Customer#000000006|sKZz0CsnMD7mp4Xd0...|        20|30-114-968-4951|  7638.57|  AUTOMOBILE|tions. even depos...|
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|        18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|
|        8|Customer#000000008|I0B10bB0AymmC, 0P...|        17|27-147-574-9335|  6819.74|    BUILDING|among the slyly r...|
|        9|Customer#000000009|xKiAFTjUsCuxfeleN...|         8|18-338-906-3675|  8324.07|   FURNITURE|r theodolites acc...|
|       10|Customer#000000010|6LrEaV6KR6PLVcgl2...|         5|15-741-346-9870|  2753.54|   HOUSEHOLD|es regular deposi...|
|       11|Customer#000000011|PkWS 3HlXqwTuzrKg...|        23|33-464-151-3439|  -272.6|    BUILDING|ckages. requests ...|
|       12|Customer#000000012|         9PWKuhzT4Zr1Q|        13|23-791-276-1263|  3396.49|   HOUSEHOLD| to the carefully...|
|       13|Customer#000000013|nsXQu0oVjD7PM659u...|         3|13-761-547-5974|  3857.34|    BUILDING|ounts sleep caref...|
|       14|Customer#000000014|     KXkletMlL2JQEA |         1|11-845-129-3851|  5266.3|   FURNITURE|, ironic packages...|
|       15|Customer#000000015|YtWggXoOLdwdo7b0y...|        23|33-687-542-7601|  2788.52|   HOUSEHOLD| platelets. regul...|
|       16|Customer#000000016| cYiaeMLZSMAOQ2 d0W,|        10|20-781-609-3107|  4681.03|   FURNITURE|kly silent courts...|
|       17|Customer#000000017|izrh 6jdqtp2eqdtb...|         2|12-970-682-3487|     6.34|  AUTOMOBILE|packages wake! bl...|
|       18|Customer#000000018|3txGO AiuFux3zT0Z...|         6|16-155-215-1315|  5494.43|    BUILDING|s sleep. carefull...|
|       19|Customer#000000019|uc,3bHIx84H,wdrmL...|        18|28-396-526-5053|  8914.71|   HOUSEHOLD| nag. furiously c...|
|       20|Customer#000000020|       JrPk8Pqplj4Ne|        22|32-957-234-8742|  7603.4|   FURNITURE|g alongside of th...|
+---------+------------------+--------------------+----------+--------------+--------+------------+--------------------+
only showing top 20 rows
```

```
+-----------+----------+-----------+--------------------+
|N_NATIONKEY|    N_NAME|N_REGIONKEY|          N_COMMENT|
+-----------+----------+-----------+--------------------+
|          0|   ALGERIA|          0| haggle. carefull...|
|          1| ARGENTINA|          1|al foxes promise ...|
|          2|    BRAZIL|          1|y alongside of th...|
|          3|    CANADA|          1|eas hang ironic, ...|
|          4|     EGYPT|          4|y above the caref...|
|          5|  ETHIOPIA|          0|ven packages wake...|
|          6|    FRANCE|          3|refully final req...|
|          7|   GERMANY|          3|l platelets. regu...|
|          8|     INDIA|          2|ss excuses cajole...|
|          9| INDONESIA|          2| slyly express as...|
|         10|      IRAN|          4|efully alongside ...|
|         11|      IRAQ|          4|nic deposits boos...|
|         12|     JAPAN|          2|ously. final, exp...|
|         13|    JORDAN|          4|ic deposits are b...|
|         14|     KENYA|          0| pending excuses ...|
|         15|   MOROCCO|          0|rns. blithely bol...|
|         16|MOZAMBIQUE|          0|s. ironic, unusua...|
|         17|      PERU|          1|platelets. blithe...|
|         18|     CHINA|          2|c dependencies. f...|
|         19|   ROMANIA|          3|ular asymptotes a...|
+-----------+----------+-----------+--------------------+
only showing top 20 rows
```

Data loading from MySql into Spark

Step 1: Create table in MySQL

create table supplier(s_suppkey int, s_name varchar(32), s_address varchar(128), s_nationkey int, s_phone varchar(16), s_acctbal float, s_comment varchar(255));

alter table supplier add constraint pk_supplier PRIMARY KEY(s_suppkey);

create table part(p_partkey int, p_name varchar(128), p_mfgr varchar(32), p_brand varchar(32), p_type varchar(128), p_size int, p_container varchar(128), p_retailprice int, p_comment varchar(255));

alter table supplier add constraint pk_supplier PRIMARY KEY(s_suppkey);

Step 2: Load files into tables

load data local infile '/home/ak/Downloads/Retail/supplier.csv' into table supplier fields terminated by '\t' lines terminated by '\n' ignore 1 lines;

load data local infile '/home/ak/Downloads/Retail/part.csv' into table supplier fields terminated by '\t' lines terminated by '\n' ignore 1 lines;

```
mysql> create table supplier(s_suppkey int, s_name varchar(32)mysql> create table supplier(, s_address varchar(128)       ), s_address varchar(128), s_nationkey int, s_phone varchar(16), s_acctbal float, s_comment varchar(255));
Query OK, 0 rows affected (0.36 sec)

mysql> alter table add constraint pk_supplier PRIMARY KEY(s_suppkey);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'add constraint pk_supplier PRIMARY KEY(s_suppkey)' at line 1
mysql> alter table supplier add constraint pk_supplier PRIMARY KEY(s_suppkey);
Query OK, 0 rows affected (0.31 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc supplier;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| s_suppkey   | int(11)      | NO   | PRI | NULL    |       |
| s_name      | varchar(32)  | YES  |     | NULL    |       |
| s_address   | varchar(128) | YES  |     | NULL    |       |
| s_nationkey | int(11)      | YES  |     | NULL    |       |
| s_phone     | varchar(16)  | YES  |     | NULL    |       |
| s_acctbal   | float        | YES  |     | NULL    |       |
| s_comment   | varchar(255) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql> load data local infile '/home/Retail/supplier.csv' into table supplier fields terminated by '\t' lines terminated by '\n' ignore 1 lines;
ERROR 2 (HY000): File '/home/Retail/supplier.csv' not found (Errcode: 2 - No such file or directory)
mysql> load data local infile '/home/ak/Retail/supplier.csv' into table supplier fields terminated by '\t' lines terminated by '\n' ignore 1 lines;
Query OK, 100 rows affected (0.28 sec)
Records: 100  Deleted: 0  Skipped: 0  Warnings: 0

mysql> select * from supplier limit 20;
+-----------+--------------------+------------------------------------------+-------------+-----------------+-----------+---------------------------------------------------------------------------------------------------------+
| s_suppkey | s_name             | s_address                                | s_nationkey | s_phone         | s_acctbal | s_comment                                                                                               |
+-----------+--------------------+------------------------------------------+-------------+-----------------+-----------+---------------------------------------------------------------------------------------------------------+
|         1 | Supplier#000000001 |  N kD4on9OM Ipw3,gf0JBoQDd7tgrzrddZ      |          17 | 27-918-335-1736 |   5755.94 | each slyly above the careful                                                                            |
|         2 | Supplier#000000002 | 89eJ5ksX3ImxJQBvxObC,                    |           5 | 15-679-861-2259 |   4032.68 |  slyly bold instructions. idle dependen                                                                 |
|         3 | Supplier#000000003 | q1,G3Pj6OjIuUYfUoH18BFTKP5aU9bEV3        |           1 | 11-383-516-1199 |    4192.4 | blithely silent requests after the express dependencies are sl                                          |
|         4 | Supplier#000000004 | Bk7ah4CK8SYQTepEmvMkkgMwg                |          15 | 25-843-787-7479 |   4641.08 | riously even requests above the exp                                                                     |
|         5 | Supplier#000000005 | Gcdm2rJRzl5qlTVzc                        |          11 | 21-151-690-3663 |   -283.84 | . slyly regular pinto bea                                                                               |
|         6 | Supplier#000000006 | tQxuVm7s7CnK                             |          14 | 24-696-997-4969 |   1365.79 | final accounts. regular dolphins use against the furiously ironic decoys.                               |
|         7 | Supplier#000000007 | s,4TicNGB4uO6PaSqNBUq                    |          23 | 33-990-965-2201 |   6820.35 | s unwind silently furiously regular courts. final requests are deposits. requests wake quietly blit     |
|         8 | Supplier#000000008 | 9Sq4bBH2FQEmaFOocY45sRTxo6yuoG           |          17 | 27-498-742-3860 |   7627.85 | al pinto beans. asymptotes haggl                                                                        |
|         9 | Supplier#000000009 | 1KhUgZegwM3ua7dsYmekYBsK                 |          10 | 20-403-398-8662 |   5302.37 | s. unusual, even requests along the furiously regular pac                                               |
|        10 | Supplier#000000010 | Saygah3gYWMp72i PY                       |          24 | 34-852-489-8585 |   3891.91 | ing waters. regular requests ar                                                                         |
|        11 | Supplier#000000011 | JfwTs,LZrV, M,9C                         |          18 | 28-613-996-1505 |   3393.08 | y ironic packages. slyly ironic accounts affix furiously; ironically unusual excuses across the flu     |
|        12 | Supplier#000000012 | aLIW  q0HYd                              |           8 | 18-179-925-7181 |   1432.69 | al packages nag alongside of the bold instructions. express, daring accounts                            |
|        13 | Supplier#000000013 | HK71HQyWoqRWOX8GI FpgAifW,2PoH           |           3 | 13-727-620-7813 |   9107.22 | requests engage regularly instructions. furiously special requests ar                                   |
|        14 | Supplier#000000014 | EXsnO5pTNj4iZRm                          |          15 | 25-656-247-5058 |   9189.82 | l accounts boost. fluffily bold warhorses wake                                                          |
|        15 | Supplier#000000015 | olXVbNBfVzRqgokr1T,Ie                    |           8 | 18-453-357-6394 |    308.56 |  across the furiously regular platelets wake even deposits. quickly express she                         |
|        16 | Supplier#000000016 | YjP5C55zHDXL7LalK27zfQnwejdpin4AMpvh     |          22 | 32-822-502-4215 |   2972.26 | ously express ideas haggle quickly dugouts? fu                                                          |
|        17 | Supplier#000000017 | c2d,ESHRSkK3WYnxpgw6aOqN0q               |          19 | 29-601-884-9219 |   1687.81 | eep against the furiously bold ideas. fluffily bold packa                                               |
|        18 | Supplier#000000018 | PGGVE5PWAMwKDZw                          |          16 | 26-729-551-1115 |   7040.82 | accounts snooze slyly furiously bold                                                                    |
```

```
mysql> desc part;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| p_partkey     | int(11)     | NO   | PRI | NULL    |       |
| p_name        | varchar(128) | YES |     | NULL    |       |
| p_mfgr        | varchar(32) | YES  |     | NULL    |       |
| p_brand       | varchar(32) | YES  |     | NULL    |       |
| p_type        | varchar(128) | YES |     | NULL    |       |
| p_size        | int(11)     | YES  |     | NULL    |       |
| p_container   | varchar(128) | YES |     | NULL    |       |
| p_retailprice | int(11)     | YES  |     | NULL    |       |
| p_comment     | varchar(255) | YES |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)

mysql> truncate table part;
Query OK, 0 rows affected (0.05 sec)

mysql> load data local infile '/home/ak/Downloads/Retail/part.csv' into table part fields terminated by '\t' lines terminated by '\n' ignore 1 lines;
Query OK, 2000 rows affected (0.18 sec)
Records: 2000  Deleted: 0  Skipped: 0  Warnings: 0

mysql> select * from part limit 10;
+-----------+-----------------------------------------+--------------+-------------+-------------------------+--------+-------------+---------------+-----------------------+
| p_partkey | p_name                                  | p_mfgr       | p_brand     | p_type                  | p_size | p_container | p_retailprice | p_comment             |
+-----------+-----------------------------------------+--------------+-------------+-------------------------+--------+-------------+---------------+-----------------------+
|         1 | goldenrod lavender spring chocolate lace | Manufacturer#1 | Brand#13 | PROMO BURNISHED COPPER  |      7 | JUMBO PKG   |           901 | ly. slyly ironi       |
|         2 | blush thistle blue yellow saddle        | Manufacturer#1 | Brand#13 | LARGE BRUSHED BRASS     |      1 | LG CASE     |           902 | lar accounts amo      |
|         3 | spring green yellow purple cornsilk     | Manufacturer#4 | Brand#42 | STANDARD POLISHED BRASS |     21 | WRAP CASE   |           903 | egular deposits hag   |
|         4 | cornflower chocolate smoke green pink   | Manufacturer#3 | Brand#34 | SMALL PLATED BRASS      |     14 | MED DRUM    |           904 | p furiously r         |
|         5 | forest brown coral puff cream           | Manufacturer#3 | Brand#32 | STANDARD POLISHED TIN   |     15 | SM PKG      |           905 |  wake carefully       |
|         6 | bisque cornflower lawn forest magenta   | Manufacturer#2 | Brand#24 | PROMO PLATED STEEL      |      4 | MED BAG     |           906 | sual a                |
|         7 | moccasin green thistle khaki floral     | Manufacturer#1 | Brand#11 | SMALL PLATED COPPER     |     45 | SM BAG      |           907 | lyly. ex              |
|         8 | misty lace thistle snow royal           | Manufacturer#4 | Brand#44 | PROMO BURNISHED TIN     |     41 | LG DRUM     |           908 | eposi                 |
|         9 | thistle dim navajo dark gainsboro       | Manufacturer#4 | Brand#43 | SMALL BURNISHED STEEL   |     12 | WRAP CASE   |           909 | ironic foxe           |
|        10 | linen pink saddle puff powder           | Manufacturer#5 | Brand#54 | LARGE BURNISHED STEEL   |     44 | LG CAN      |           910 | ithely final deposit  |
+-----------+-----------------------------------------+--------------+-------------+-------------------------+--------+-------------+---------------+-----------------------+
10 rows in set (0.00 sec)
```

Step 3: Create dataframes in Pyspark of the datasets in MySQL

>>> part_df = spark.read.format("jdbc").option("url", "jdbc:mysql://localhost:3306/project?useSSL=false").option("driver", "com.mysql.jdbc.Driver").option("dbtable", "part").option("user", "hiveuser").option("password", "hivepassword").option("partitionColumn","p_partkey").option("lowerBound",0).option("upperBound", 2000).option("numPartitions",1).load()

>>> part_df.show()

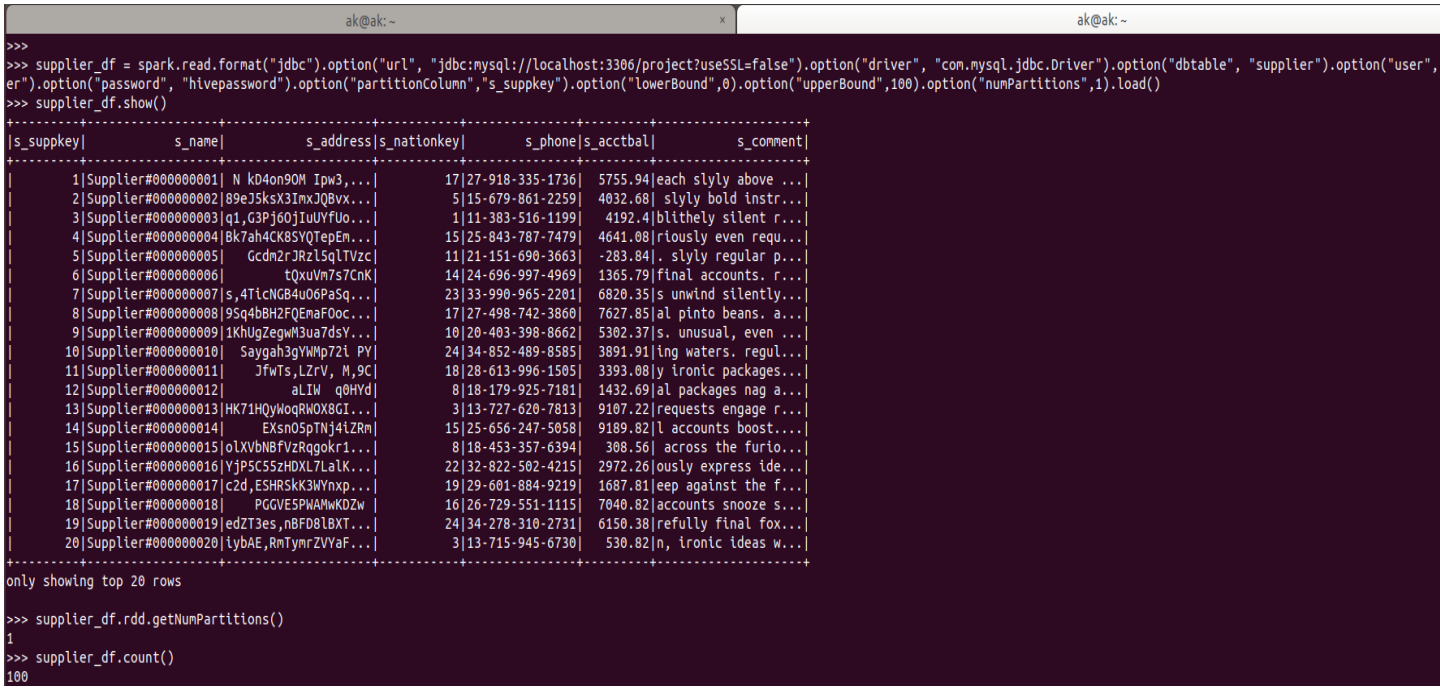
```
>>> part_df = spark.read.format("jdbc").option("url", "jdbc:mysql://localhost:3306/project?useSSL=false").option("driver", "com.mysql.jdbc.Driver").option("d
ion("password", "hivepassword").option("partitionColumn","p_partkey").option("lowerBound",0).option("upperBound",2000).option("numPartitions",1).load()
>>> part_df.show()
+---------+--------------------+--------------+--------+--------------------+------+-----------+-------------+--------------------+
|p_partkey|              p_name|        p_mfgr| p_brand|              p_type|p_size|p_container|p_retailprice|           p_comment|
+---------+--------------------+--------------+--------+--------------------+------+-----------+-------------+--------------------+
|        1|goldenrod lavende...|Manufacturer#1|Brand#13|PROMO BURNISHED C...|     7|  JUMBO PKG|          901|     ly. slyly ironi|
|        2|blush thistle blu...|Manufacturer#1|Brand#13| LARGE BRUSHED BRASS|     1|    LG CASE|          902|    lar accounts amo|
|        3|spring green yell...|Manufacturer#4|Brand#42|STANDARD POLISHED...|    21|  WRAP CASE|          903| egular deposits hag|
|        4|cornflower chocol...|Manufacturer#3|Brand#34|  SMALL PLATED BRASS|    14|   MED DRUM|          904|       p furiously r|
|        5|forest brown cora...|Manufacturer#3|Brand#32|STANDARD POLISHED...|    15|     SM PKG|          905|     wake carefully |
|        6|bisque cornflower...|Manufacturer#2|Brand#24|  PROMO PLATED STEEL|     4|    MED BAG|          906|              sual a|
|        7|moccasin green th...|Manufacturer#1|Brand#11|  SMALL PLATED COPPER|    45|     SM BAG|          907|            lyly. ex|
|        8|misty lace thistl...|Manufacturer#4|Brand#44|  PROMO BURNISHED TIN|    41|    LG DRUM|          908|              eposi|
|        9|thistle dim navaj...|Manufacturer#4|Brand#43|SMALL BURNISHED S...|    12|  WRAP CASE|          909|          ironic foxe|
|       10|linen pink saddle...|Manufacturer#5|Brand#54|LARGE BURNISHED S...|    44|     LG CAN|          910|ithely final deposit|
|       11|spring maroon sea...|Manufacturer#2|Brand#25|STANDARD BURNISHE...|    43|    WRAP BOX|          911|               ng gr|
|       12|cornflower wheat ...|Manufacturer#3|Brand#33|MEDIUM ANODIZED S...|    25|  JUMBO CASE|          912|             quickly|
|       13|ghost olive orang...|Manufacturer#5|Brand#55|MEDIUM BURNISHED ...|     1|  JUMBO PACK|          913|              osits.|
|       14|khaki seashell ro...|Manufacturer#1|Brand#13|SMALL POLISHED STEEL|    28|   JUMBO BOX|          914|            kages c|
|       15|blanched honeydew...|Manufacturer#1|Brand#15|LARGE ANODIZED BRASS|    45|    LG CASE|          915|            usual ac|
|       16|deep sky turquois...|Manufacturer#3|Brand#32|    PROMO PLATED TIN|     2|    MED PACK|          916|              unts a|
|       17|indian navy coral...|Manufacturer#4|Brand#43|ECONOMY BRUSHED S...|    16|     LG BOX|          917|    regular accounts|
|       18|turquoise indian ...|Manufacturer#1|Brand#11|SMALL BURNISHED S...|    42|  JUMBO PACK|          918|    s cajole slyly a|
|       19|chocolate navy ta...|Manufacturer#2|Brand#23|SMALL ANODIZED NI...|    33|    WRAP BOX|          919|         pending acc|
|       20|ivory navy honeyd...|Manufacturer#1|Brand#12|LARGE POLISHED NI...|    48|     MED BAG|          920|are across the as...|
+---------+--------------------+--------------+--------+--------------------+------+-----------+-------------+--------------------+
only showing top 20 rows
```

>>> supplier_df = spark.read.format("jdbc").option("url", "jdbc:mysql://localhost:3306/project?useSSL=false").option("driver", "com.mysql.jdbc.Driver").option("dbtable", "supplier").option("user", "hiveuser").option("password", "hivepassword").option("partitionColumn","s_suppkey").option("lowerBound",0).option("upperBound", 100).option("numPartitions",1).load()

>>> supplier_df.show()

>>> supplier_df.rdd.getNumPartitions()

>>> supplier_df.count()



Data loading from local file

Step 1: Read CSV data files

region=spark.read.option('delimiter','\t').option('header','true').csv("file:///home/ak/pglad/region.csv")
>>> region.printSchema()
>>> region.show()

Step 2: Casted R_REGIONKEY as integer

region1 = region.selectExpr("cast(R_REGIONKEY as Integer) as R_REGIONKEY","R_NAME","R_COMMENT")
>>> region1.show()
>>> region1.printSchema()

```
>>> region=spark.read.option('delimiter','\t').option('header','true').csv("file:///home/ak/pglad/region.csv")
>>> region.printSchema()
root
 |-- R_REGIONKEY: string (nullable = true)
 |-- R_NAME: string (nullable = true)
 |-- R_COMMENT: string (nullable = true)

>>> region.show()
+-----------+-----------+--------------------+
|R_REGIONKEY|     R_NAME|           R_COMMENT|
+-----------+-----------+--------------------+
|          0|     AFRICA|lar deposits. bli...|
|          1|    AMERICA|hs use ironic, ev...|
|          2|       ASIA|ges. thinly even ...|
|          3|     EUROPE|ly final courts c...|
|          4|MIDDLE EAST|uickly special ac...|
+-----------+-----------+--------------------+
```

Data Cleaning

Casting a data type :
lineitem.withColumn('L_QUANTITY', col('L_QUANTITY').cast('int'))
Before :

```
>>> lineitem.select('L_QUANTITY').show(3)
+----------+
|L_QUANTITY|
+----------+
|      17.0|
|      36.0|
|       8.0|
+----------+
only showing top 3 rows

>>> lineitem.select('L_QUANTITY').printSchema()
root
 |-- L_QUANTITY: double (nullable = true)
```

After :

```
>>> lineitem_casted = lineitem.withColumn('L_QUANTITY', col('L_QUANTITY').cast('int'))
>>> lineitem_casted.select('L_QUANTITY').show(3)
+----------+
|L_QUANTITY|
+----------+
|        17|
|        36|
|         8|
+----------+
only showing top 3 rows
```

Dropping Duplicates :
lineitem_casted.dropDuplicates()

```
>>> lineitem_casted.dropDuplicates().count()
60175
>>> lineitem_casted.count()
60175
>>>
```

Dropping NA :

lineitem.dropna()

```
>>> lineitem.dropna().count()
60175
>>> lineitem.count()
60175
>>>
```

test_df = spark.read.format("jdbc").option("url", "jdbc:mysql://localhost:3306/project?useSSL=false").option("driver", "com.mysql.jdbc.Driver").option("dbtable", "test").option("user", "hiveuser").option("password", "hivepassword").load()

>>> test_df.show()

>>> test_df.count()

>>> test_df.na.drop().show()

>>> test_df.na.drop().count()

>>> part_df.count()

>>> part_df.na.drop().count()

>>> supplier_df.count()

>>> supplier_df.na.drop().count()

```
>>> test_df.show()
+----+------+
|  id|    c2|
+----+------+
|   1|  2222|
|   2|  abxy|
|   2|  null|
|   6|  null|
|   4|   abc|
|null|abcxyz|
+----+------+

>>> test_df.count()
6
>>> test_df.na.drop().show()
+---+----+
| id|  c2|
+---+----+
|  1|2222|
|  2|abxy|
|  4| abc|
+---+----+

>>> test_df.na.drop().count()
3
>>> part_df.count()
2000
>>> part_df.na.drop().count()
2000
>>> supplier_df.count()
100
>>> supplier_df.na.drop().count()
100
>>>
```

Predefining User Schema and then Reading Data into Dataframe.

```
>>> RegionSchema = StructType([ \
...     StructField('R_REGIONKEY', IntegerType(), True), \
...     StructField('R_NAME', StringType(), True), \
...     StructField('R_COMMENT', StringType(), True) \
... ])
>>> OrderSchema = StructType([ \
...     StructField('O_ORDERKEY', IntegerType(), True), \
...     StructField('O_CUSTKEY', IntegerType(), True), \
...     StructField('O_ORDERSTATUS', StringType(), True), \
...     StructField('O_TOTALPRICE', DoubleType(), True), \
...     StructField('O_ORDERDATE', DateType(), True), \
...     StructField('O_ORDERPRIORITY', StringType(), True), \
...     StructField('O_CLERK', StringType(), True), \
...     StructField('O_SHIPPRIORITY', IntegerType(), True), \
...     StructField('O_COMMENT', StringType(), True) \
... ])
```

```
>>> NationSchema = StructType([ \
...     StructField('N_NATIONKEY', IntegerType(), True), \
...     StructField('N_NAME', StringType(), True), \
...     StructField('N_REGIONKEY', IntegerType(), True), \
...     StructField('N_COMMENT', StringType(), True) \
... ])
>>> CustomerSchema = StructType([ \
...     StructField('C_CUSTKEY', IntegerType(), True), \
...     StructField('C_NAME', StringType(), True), \
...     StructField('C_ADDRESS', StringType(), True), \
...     StructField('C_NATIONKEY', IntegerType(), True), \
...     StructField('C_PHONE', StringType(), True), \
...     StructField('C_ACCTBAL', DoubleType(), True), \
...     StructField('C_MKTSEGMENT', StringType(), True), \
...     StructField('C_COMMENT', StringType(), True) \
... ])
>>>
>>> PartsuppSchema = StructType([ \
...     StructField('PS_PARTKEY', IntegerType(), True), \
...     StructField('PS_SUPPKEY', IntegerType(), True), \
...     StructField('PS_AVAILQTY', IntegerType(), True), \
...     StructField('PS_SUPPLYCOST', DoubleType(), True), \
...     StructField('PS_COMMENT', StringType(), True) \
... ])
```

```
...
>>> RegionSchema = StructType([ \
...     StructField('R_REGIONKEY', IntegerType(), True), \
...     StructField('R_NAME', StringType(), True), \
...     StructField('R_COMMENT', StringType(), True) \
... ])
>>> OrderSchema = StructType([ \
...     StructField('O_ORDERKEY', IntegerType(), True), \
...     StructField('O_CUSTKEY', IntegerType(), True), \
...     StructField('O_ORDERSTATUS', StringType(), True), \
...     StructField('O_TOTALPRICE', DoubleType(), True), \
...     StructField('O_ORDERDATE', DateType(), True), \
...     StructField('O_ORDERPRIORITY', StringType(), True), \
...     StructField('O_CLERK', StringType(), True), \
...     StructField('O_SHIPPRIORITY', IntegerType(), True), \
...     StructField('O_COMMENT', StringType(), True) \
... ])
>>> NationSchema = StructType([ \
...     StructField('N_NATIONKEY', IntegerType(), True), \
...     StructField('N_NAME', StringType(), True), \
...     StructField('N_REGIONKEY', IntegerType(), True), \
...     StructField('N_COMMENT', StringType(), True) \
... ])
>>> CustomerSchema = StructType([ \
...     StructField('C_CUSTKEY', IntegerType(), True), \
...     StructField('C_NAME', StringType(), True), \
...     StructField('C_ADDRESS', StringType(), True), \
...     StructField('C_NATIONKEY', IntegerType(), True), \
...     StructField('C_PHONE', StringType(), True), \
...     StructField('C_ACCTBAL', DoubleType(), True), \
...     StructField('C_MKTSEGMENT', StringType(), True), \
...     StructField('C_COMMENT', StringType(), True) \
... ])
>>>
>>> PartsuppSchema = StructType([ \
...     StructField('PS_PARTKEY', IntegerType(), True), \
...     StructField('PS_SUPPKEY', IntegerType(), True), \
...     StructField('PS_AVAILQTY', IntegerType(), True), \
...     StructField('PS_SUPPLYCOST', DoubleType(), True), \
...     StructField('PS_COMMENT', StringType(), True) \
... ])
```

>>> orders_df = spark.read.schema(OrderSchema).option('header','true').csv('file:///home/ak/Retail/orders.csv', sep='\t')

>>> orders_df.printSchema()

>>> orders_df.show()

```
>>> orders_df = spark.read.schema(OrderSchema).option('header','true').csv('file:///home/ak/Retail/orders.csv', sep='\t')
>>> orders_df.printSchema()
root
 |-- O_ORDERKEY: integer (nullable = true)
 |-- O_CUSTKEY: integer (nullable = true)
 |-- O_ORDERSTATUS: string (nullable = true)
 |-- O_TOTALPRICE: double (nullable = true)
 |-- O_ORDERDATE: date (nullable = true)
 |-- O_ORDERPRIORITY: string (nullable = true)
 |-- O_CLERK: string (nullable = true)
 |-- O_SHIPPRIORITY: integer (nullable = true)
 |-- O_COMMENT: string (nullable = true)

>>> orders_df.show()
+----------+---------+-------------+------------+----------+---------------+---------------+--------------+--------------------+
|O_ORDERKEY|O_CUSTKEY|O_ORDERSTATUS|O_TOTALPRICE|O_ORDERDATE|O_ORDERPRIORITY|        O_CLERK|O_SHIPPRIORITY|           O_COMMENT|
+----------+---------+-------------+------------+----------+---------------+---------------+--------------+--------------------+
|         1|      370|            O|   172799.49| 1996-01-02|          5-LOW|Clerk#000000951|             0|nstructions sleep...|
|         2|      781|            O|    38426.09| 1996-12-01|       1-URGENT|Clerk#000000880|             0| foxes. pending a...|
|         3|     1234|            F|    205654.3| 1993-10-14|          5-LOW|Clerk#000000955|             0|sly final account...|
|         4|     1369|            O|    56000.91| 1995-10-11|          5-LOW|Clerk#000000124|             0|sits. slyly regul...|
|         5|      445|            F|   105367.67| 1994-07-30|          5-LOW|Clerk#000000925|             0|quickly. bold dep...|
|         6|      557|            F|     45523.1| 1992-02-21|4-NOT SPECIFIED|Clerk#000000058|             0|ggle. special, fi...|
|         7|      392|            O|   271885.66| 1996-01-10|         2-HIGH|Clerk#000000470|             0|ly special requests |
|        32|     1301|            O|   198665.57| 1995-07-16|         2-HIGH|Clerk#000000616|             0|ise blithely bold...|
|        33|      670|            F|   146567.24| 1993-10-27|       3-MEDIUM|Clerk#000000409|             0|uriously. furious...|
|        34|      611|            O|    73315.48| 1998-07-21|       3-MEDIUM|Clerk#000000223|             0|ly final packages...|
|        35|     1276|            O|   194641.93| 1995-10-23|4-NOT SPECIFIED|Clerk#000000259|             0|zzle. carefully e...|
|        36|     1153|            O|    42011.04| 1995-11-03|       1-URGENT|Clerk#000000358|             0| quick packages a...|
|        37|      862|            F|   131896.49| 1992-06-03|       3-MEDIUM|Clerk#000000456|             0|kly regular pinto...|
|        38|     1249|            O|    71553.08| 1996-08-21|4-NOT SPECIFIED|Clerk#000000604|             0|haggle blithely. ...|
|        39|      818|            O|   326565.37| 1996-09-20|       3-MEDIUM|Clerk#000000659|             0|ole express, iron...|
|        64|      322|            F|    35831.73| 1994-07-16|       3-MEDIUM|Clerk#000000661|             0|wake fluffily. so...|
|        65|      163|            P|    95469.44| 1995-03-18|       1-URGENT|Clerk#000000632|             0|ular requests are...|
|        66|     1292|            F|   104190.66| 1994-01-20|          5-LOW|Clerk#000000743|             0|y pending request...|
|        67|      568|            O|   182481.16| 1996-12-19|4-NOT SPECIFIED|Clerk#000000547|             0|symptotes haggle ...|
|        68|      286|            O|   301968.79| 1998-04-18|       3-MEDIUM|Clerk#000000440|             0| pinto beans slee...|
+----------+---------+-------------+------------+----------+---------------+---------------+--------------+--------------------+
only showing top 20 rows
```

>>> partsupp_df = spark.read.schema(PartsuppSchema).option('header','true').csv('file:///home/ak/Retail/partsupp.csv', sep='\t')

>>> partsupp_df.printSchema()

>>> partsupp_df.show()

```
>>> partsupp_df = spark.read.schema(PartsuppSchema).option('header','true').csv('file:///home/ak/Retail/partsupp.csv', sep='\t')
>>> partsupp_df.printSchema()
root
 |-- PS_PARTKEY: integer (nullable = true)
 |-- PS_SUPPKEY: integer (nullable = true)
 |-- PS_AVAILQTY: integer (nullable = true)
 |-- PS_SUPPLYCOST: double (nullable = true)
 |-- PS_COMMENT: string (nullable = true)

>>> partsupp_df.printSchema()
root
 |-- PS_PARTKEY: integer (nullable = true)
 |-- PS_SUPPKEY: integer (nullable = true)
 |-- PS_AVAILQTY: integer (nullable = true)
 |-- PS_SUPPLYCOST: double (nullable = true)
 |-- PS_COMMENT: string (nullable = true)

>>> partsupp_df.show()
+----------+----------+-----------+-------------+--------------------+
|PS_PARTKEY|PS_SUPPKEY|PS_AVAILQTY|PS_SUPPLYCOST|          PS_COMMENT|
+----------+----------+-----------+-------------+--------------------+
|         1|         2|       3325|       771.64|, even theodolite...|
|         1|        27|       8076|       993.49|ven ideas. quickl...|
|         1|        52|       3956|       337.09|after the fluffil...|
|         1|        77|       4069|       357.84|al, regular depen...|
|         2|         3|       8895|       378.49|nic accounts. fin...|
|         2|        28|       4969|       915.27|ptotes. quickly p...|
|         2|        53|       8539|       438.37|blithely bold ide...|
|         2|        78|       3025|       306.39|olites. deposits ...|
|         3|         4|       4651|       920.92|ilent foxes affix...|
|         3|        29|       4093|       498.13|ending dependenci...|
|         3|        54|       3917|        645.4|of the blithely r...|
|         3|        79|       9942|       191.92| unusual, ironic ...|
|         4|         5|       1339|       113.97| carefully unusua...|
|         4|        30|       6377|       591.18|ly final courts h...|
|         4|        55|       2694|        51.37|g, regular deposi...|
|         4|        80|       2480|       444.37|requests sleep qu...|
|         5|         6|       3735|       255.88|arefully even req...|
|         5|        31|       9653|        50.52|y stealthy deposi...|
|         5|        56|       1329|       219.83|iously regular de...|
|         5|        81|       6925|       537.98|sits. quickly flu...|
+----------+----------+-----------+-------------+--------------------+
only showing top 20 rows
```

>>> nation_df = spark.read.schema(NationSchema).option('header','true').csv('file:///home/ak/Retail/nation.csv', sep='\t')

>>> nation_df.show()

>>> nation_df.printSchema()

```
>>> nation_df = spark.read.schema(NationSchema).option('header','true').csv('file:///home/ak/Retail/nation.csv', sep='\t')
>>> nation_df.show()
+-----------+----------+-----------+--------------------+
|N_NATIONKEY|    N_NAME|N_REGIONKEY|           N_COMMENT|
+-----------+----------+-----------+--------------------+
|          0|   ALGERIA|          0| haggle. carefull...|
|          1| ARGENTINA|          1|al foxes promise ...|
|          2|    BRAZIL|          1|y alongside of th...|
|          3|    CANADA|          1|eas hang ironic, ...|
|          4|     EGYPT|          4|y above the caref...|
|          5|  ETHIOPIA|          0|ven packages wake...|
|          6|    FRANCE|          3|refully final req...|
|          7|   GERMANY|          3|l platelets. regu...|
|          8|     INDIA|          2|ss excuses cajole...|
|          9| INDONESIA|          2| slyly express as...|
|         10|      IRAN|          4|efully alongside ...|
|         11|      IRAQ|          4|nic deposits boos...|
|         12|     JAPAN|          2|ously. final, exp...|
|         13|    JORDAN|          4|ic deposits are b...|
|         14|     KENYA|          0| pending excuses ...|
|         15|   MOROCCO|          0|rns. blithely bol...|
|         16|MOZAMBIQUE|          0|s. ironic, unusua...|
|         17|      PERU|          1|platelets. blithe...|
|         18|     CHINA|          2|c dependencies. f...|
|         19|   ROMANIA|          3|ular asymptotes a...|
+-----------+----------+-----------+--------------------+
only showing top 20 rows

>>> nation_df.printSchema()
root
 |-- N_NATIONKEY: integer (nullable = true)
 |-- N_NAME: string (nullable = true)
 |-- N_REGIONKEY: integer (nullable = true)
 |-- N_COMMENT: string (nullable = true)

>>>
```

>>> region_df = spark.read.schema(RegionSchema).option('header','true').csv('file:///home/ak/Retail/region.csv', sep='\t')

>>> region_df.printSchema()

>>> region_df.show()

```
>>> region_df = spark.read.schema(RegionSchema).option('header','true').csv('file:///home/ak/Retail/region.csv', sep='\t')
>>> region_df.printSchema()
root
 |-- R_REGIONKEY: integer (nullable = true)
 |-- R_NAME: string (nullable = true)
 |-- R_COMMENT: string (nullable = true)

>>> region_df.show()
+-----------+-----------+--------------------+
|R_REGIONKEY|     R_NAME|           R_COMMENT|
+-----------+-----------+--------------------+
|          0|     AFRICA|lar deposits. bli...|
|          1|    AMERICA|hs use ironic, ev...|
|          2|       ASIA|ges. thinly even ...|
|          3|     EUROPE|ly final courts c...|
|          4|MIDDLE EAST|uickly special ac...|
+-----------+-----------+--------------------+

>>>
```

Data Storage Optimization

>>>part_df.write.option("header","true").parquet("/retail_project/clean_data/part")

>>>supplier_df.write.option("header","true").option("mode","overwrite").parquet("/retail_project/clean_data/supplier")

>>>orders_df.write.option("header","true").option("mode","overwrite").parquet("/retail_project/clean_data/orders")

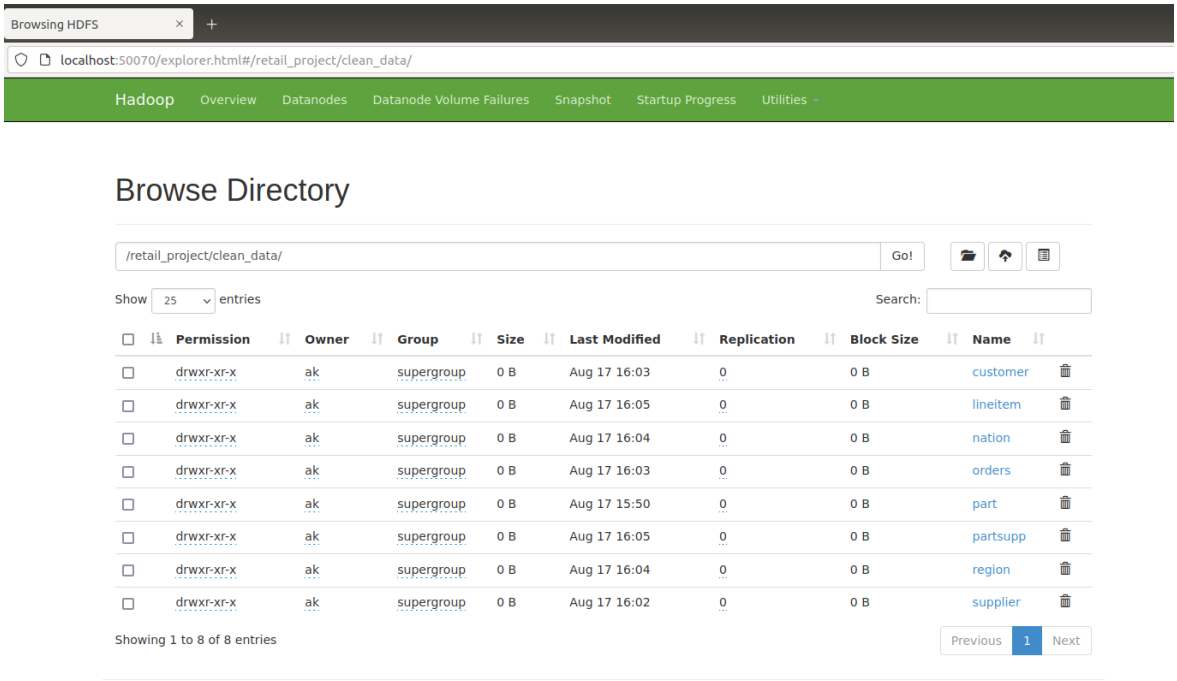
>>>customer_df.write.option("header","true").option("mode","overwrite").parquet("/retail_project/clean_data/customer")

>>>region_df.write.option("header","true").option("mode","overwrite").parquet("/retail_project/clean_data/region")

>>>nation_df.write.option("header","true").option("mode","overwrite").parquet("/retail_project/clean_data/nation")

>>>lineitem_df.write.option("header","true").option("mode","overwrite").parquet("/retail_project/clean_data/lineitem")

>>>partsupp_df.write.option("header","true").option("mode","overwrite").parquet("/retail_project/clean_data/partsupp")



Reading Data from HDFS as a Dataframe

>>> test_df=spark.read.option("header","true").parquet("/retail_project/clean_data/part")

>>> test_df.show()

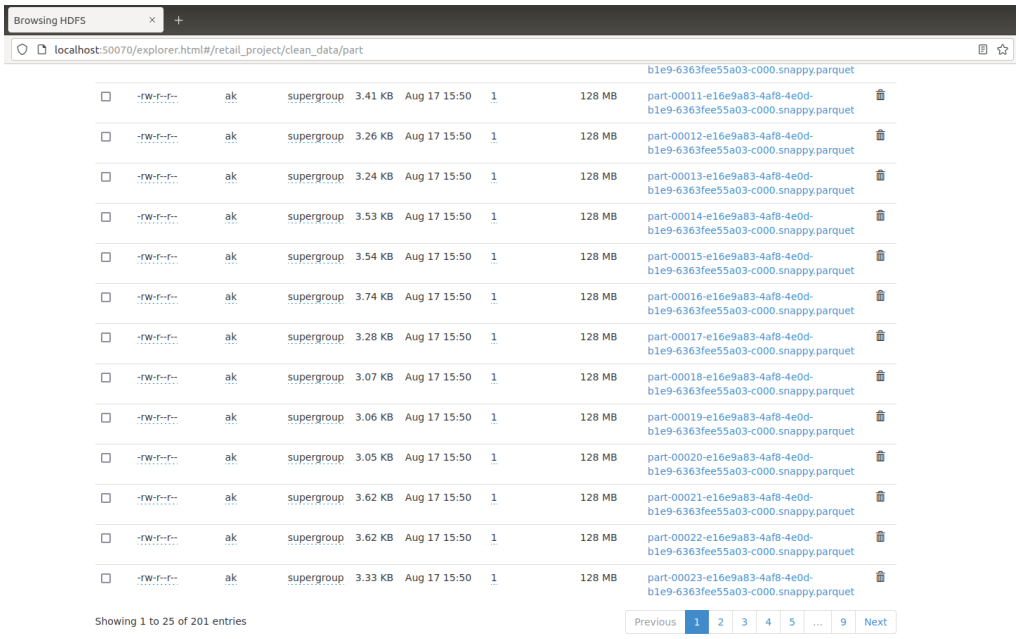
>>> test_df.printSchema()

>>> test_df.rdd.getNumPartitions()

>>> test_df.count()

>>> part_df.count()

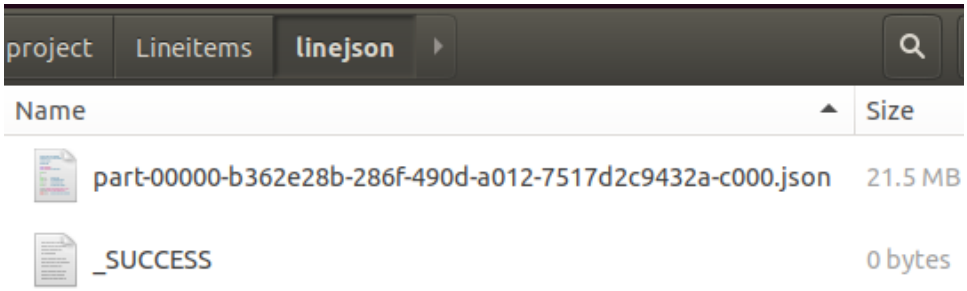
```
>>> test_df=spark.read.option("header","true").parquet("/retail_project/clean_data/part")
>>> test_df.show()
+---------+--------------------+-------------+--------+--------------------+------+----------+------------+--------------------+
|p_partkey|              p_name|       p_mfgr| p_brand|              p_type|p_size|p_container|p_retailprice|           p_comment|
+---------+--------------------+-------------+--------+--------------------+------+----------+------------+--------------------+
|      413|chiffon ivory law...|Manufacturer#3|Brand#33|  PROMO PLATED STEEL|    49|    MED PKG|        1313|r deposits wake. alw|
|      586|ivory rosy royal ...|Manufacturer#1|Brand#11|PROMO ANODIZED BRASS|    11|    LG DRUM|        1487| ts haggle fluffily |
|      614|dodger beige maro...|Manufacturer#2|Brand#23|STANDARD BURNISHE...|     6|   WRAP PKG|        1515|  ending accounts ca|
|      675|misty hot pale gh...|Manufacturer#1|Brand#11|LARGE BRUSHED COPPER|     2|     SM CAN|        1576|             ely re|
|      715|deep yellow coral...|Manufacturer#2|Brand#24|    LARGE BRUSHED TIN|    28|   WRAP BAG|        1616|   ons boost. furiou|
|      753|blanched plum nav...|Manufacturer#2|Brand#25|STANDARD ANODIZED...|    13|    LG DRUM|        1654|nal foxes. quickly f|
|      760|orange navy hot a...|Manufacturer#4|Brand#44|MEDIUM POLISHED C...|     6|     LG JAR|        1661|     jole according |
|      763|wheat seashell az...|Manufacturer#4|Brand#44|    LARGE BRUSHED TIN|    50|     SM PKG|        1664|        counts. regu|
|      985|drab rosy orange ...|Manufacturer#5|Brand#53|MEDIUM BURNISHED ...|    10|  JUMBO PKG|        1886|           s beside|
|     1013|sienna maroon cho...|Manufacturer#2|Brand#24|STANDARD BRUSHED ...|    14|  JUMBO BOX|         914|    uests. furiously|
|     1273|papaya dark cornf...|Manufacturer#2|Brand#25|MEDIUM BRUSHED CO...|    11|     LG JAR|        1174|        unusual plat|
|     1287|peach indian gold...|Manufacturer#4|Brand#41|   PROMO BRUSHED TIN|    33|     SM BOX|        1188|              e of |
|     1423|cream green flora...|Manufacturer#1|Brand#15|ECONOMY BRUSHED B...|    36|  WRAP PACK|        1324|  according to the r|
|     1424|forest metallic b...|Manufacturer#4|Brand#45|ECONOMY BURNISHED...|    25|    LG PACK|        1325|  ons. pinto beans w|
|     1428|aquamarine orange...|Manufacturer#3|Brand#34|ECONOMY POLISHED ...|    42|    MED PKG|        1329|             ld excu|
|     1552|plum chartreuse s...|Manufacturer#4|Brand#41|  SMALL POLISHED TIN|    10|  WRAP CASE|        1454|       onic deposits|
|     1710|cornsilk khaki ol...|Manufacturer#4|Brand#43|PROMO BURNISHED S...|     3|  JUMBO BOX|        1612|        ainst the de|
|     1727|dim puff frosted ...|Manufacturer#3|Brand#33|  SMALL PLATED BRASS|     6|   WRAP CAN|        1629|es. final deposit...|
|     1983|deep black beige ...|Manufacturer#2|Brand#25|  MEDIUM PLATED BRASS|    22|   WRAP CAN|        1885| lly about the slyly|
|        8|misty lace thistl...|Manufacturer#4|Brand#44| PROMO BURNISHED TIN|    41|    LG DRUM|         908|               eposi|
+---------+--------------------+-------------+--------+--------------------+------+----------+------------+--------------------+
only showing top 20 rows

>>> test_df.printSchema()
root
 |-- p_partkey: integer (nullable = true)
 |-- p_name: string (nullable = true)
 |-- p_mfgr: string (nullable = true)
 |-- p_brand: string (nullable = true)
 |-- p_type: string (nullable = true)
 |-- p_size: integer (nullable = true)
 |-- p_container: string (nullable = true)
 |-- p_retailprice: integer (nullable = true)
 |-- p_comment: string (nullable = true)

>>> test_df.rdd.getNumPartitions()
7
>>> test_df.count()
2000
>>> part_df.count()
2000
```



Writing in various formats and codecs and comparing their sizes.

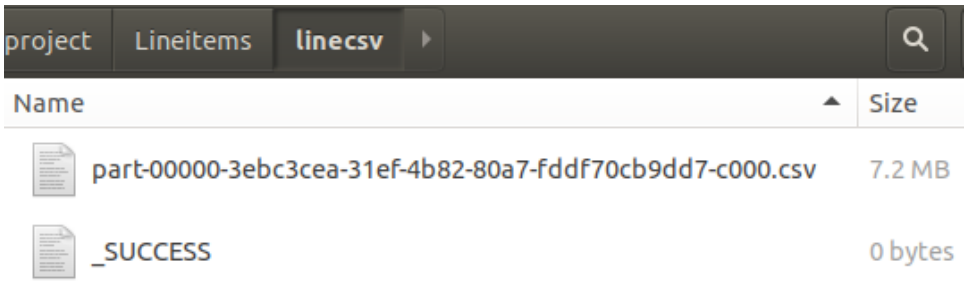
JSON

>>> lineitem_df.write.json("file:///home/ak/project/Lineitems/linejson")



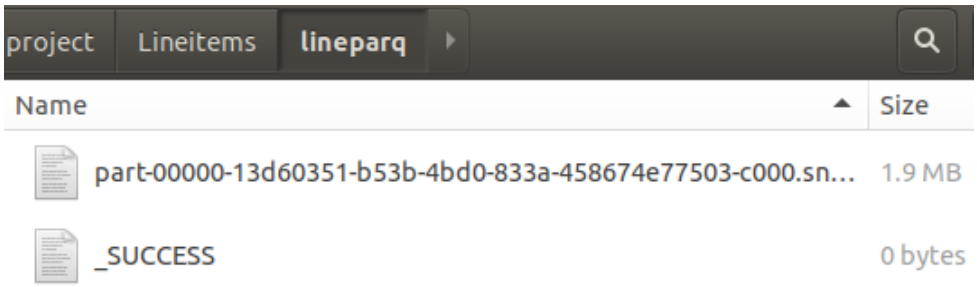
CSV

>>> lineitem_df.write.csv("file:///home/ak/project/Lineitems/linecsv")



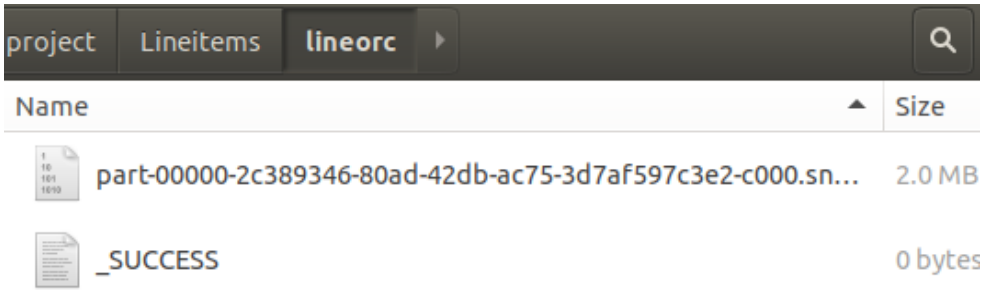
Parquet

>>> lineitem_df.write.parquet("file:///home/ak/project/Lineitems/lineparq")



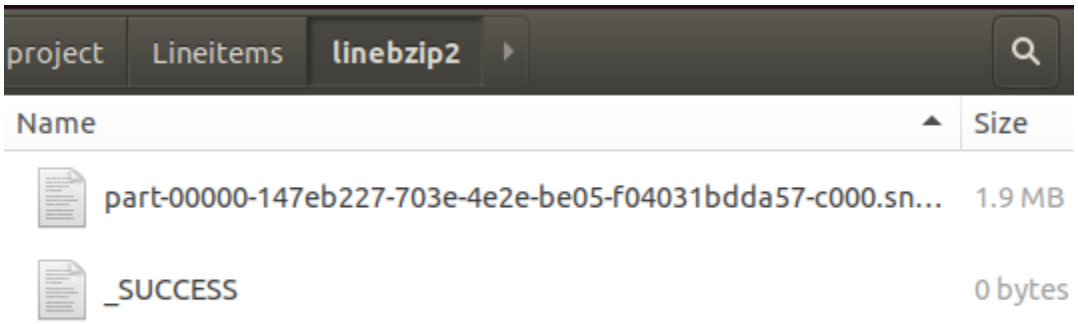
ORC

>>> lineitem_df.write.orc("file:///home/ak/project/Lineitems/lineorc")



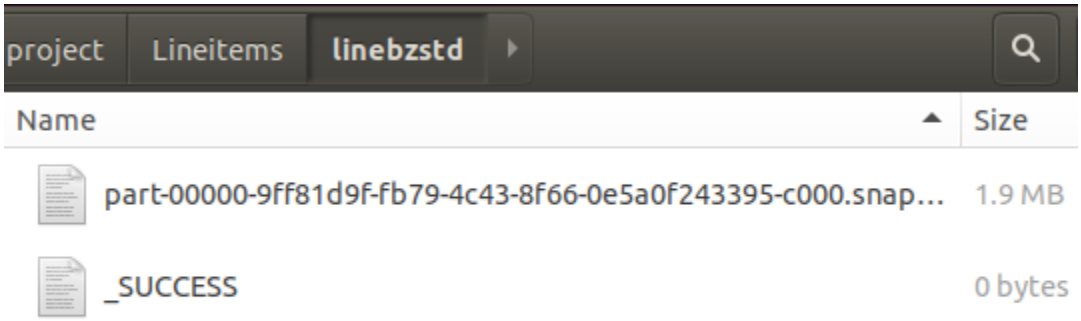
BZIP2

> lineitem_df.write.option("codec","bzip2").parquet("file:///home/ak/project/Lineitems/linebzip2")



ZSTD

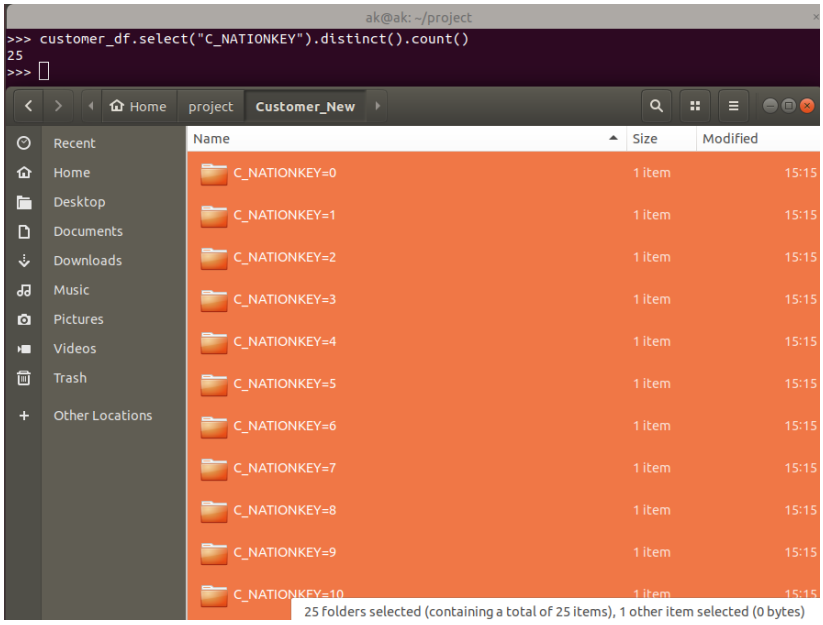
>>>lineitem_df.write.option("codec","zstd").parquet("file:///home/ak/project/Lineitems/linebzstd")



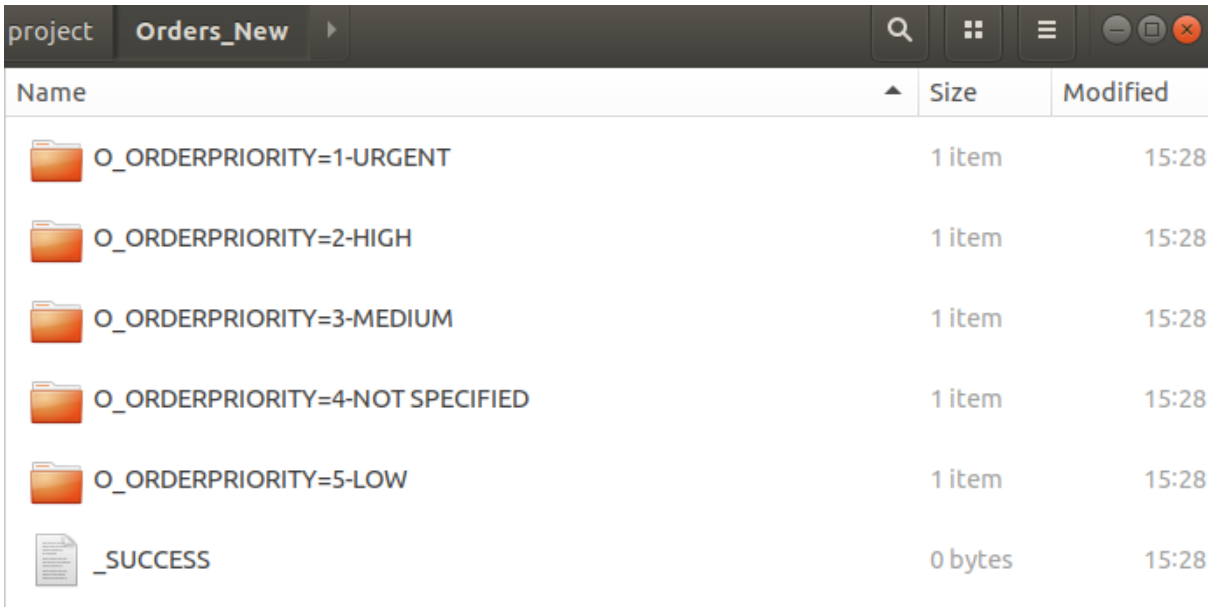
Partitioning

>>>customer_df.write.option("header","true").partitionBy("C_NATIONKEY").csv("file:///home/ak/project/Customer_New")

>>>customer_df.select("C_NATIONKEY").distinct().count()

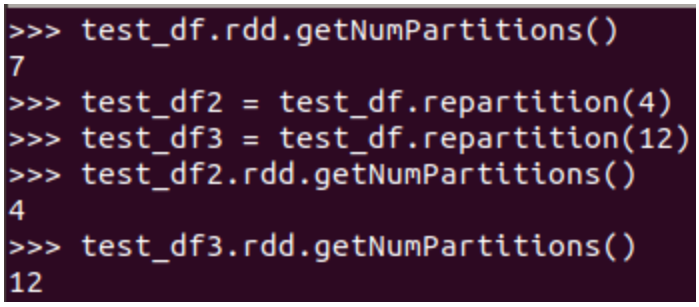


>>> orders_df.write.partitionBy("O_ORDERPRIORITY").json("file:///home/ak/project/Orders_New")



Repartitioning

```
>>> test_df.rdd.getNumPartitions()
>>> test_df2 = test_df.repartition(4)
>>> test_df3 = test_df.repartition(12)
>>> test_df2.rdd.getNumPartitions()
>>> test_df3.rdd.getNumPartitions()
```



Repartitioning By Column

```
>>> test_df.columns
>>> test_df.show(5)
>>> test_df.rdd.getNumPartitions()
>>> test_df2 = test_df.repartition("p_mfgr")
>>> test_df2.rdd.getNumPartitions()
>>> test_df2.show(5)
```

```
ak@ak: ~/project
>>> test_df.columns
['p_partkey', 'p_name', 'p_mfgr', 'p_brand', 'p_type', 'p_size', 'p_container', 'p_retailprice', 'p_comment']
>>> test_df.show(5)
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
|p_partkey|              p_name|       p_mfgr| p_brand|              p_type|p_size|p_container|p_retailprice|           p_comment|
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
|     1970|peru saddle blue ...|Manufacturer#5|Brand#54|  ECONOMY PLATED TIN|    50|   WRAP JAR|         1872|platelets wake b|
|     1493|metallic burlywoo...|Manufacturer#3|Brand#34|ECONOMY BRUSHED S...|    16|     SM PKG|         1394|          ccounts |
|      471|goldenrod honeyde...|Manufacturer#1|Brand#11| LARGE BRUSHED STEEL|    21|  JUMBO BAG|         1371|   sits nag caref|
|      511|red pale plum orc...|Manufacturer#1|Brand#15|STANDARD BRUSHED ...|     2|     LG JAR|         1412|       ss package|
|      794|ivory peach light...|Manufacturer#5|Brand#51| PROMO PLATED COPPER|    47|    MED PKG|         1695|             le bl|
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
only showing top 5 rows

>>> test_df.rdd.getNumPartitions()
7
>>> test_df2 = test_df.repartition("p_mfgr")
>>> test_df2.rdd.getNumPartitions()
200
>>> test_df2.show(5)
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
|p_partkey|              p_name|       p_mfgr| p_brand|              p_type|p_size|p_container|p_retailprice|           p_comment|
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
|      586|ivory rosy royal ...|Manufacturer#1|Brand#11|PROMO ANODIZED BRASS|    11|    LG DRUM|         1487|ts haggle fluffily |
|      675|misty hot pale gh...|Manufacturer#1|Brand#11|LARGE BRUSHED COPPER|     2|     SM CAN|         1576|             ely re|
|     1423|cream green flora...|Manufacturer#1|Brand#15|ECONOMY BRUSHED B...|    36|  WRAP PACK|         1324| according to the r|
|      605|grey burlywood li...|Manufacturer#1|Brand#11|LARGE POLISHED STEEL|    30|     LG JAR|         1506|  bold, final accou|
|     1147|smoke firebrick y...|Manufacturer#1|Brand#15|ECONOMY BURNISHED...|    14|   MED CASE|         1048|           the acco|
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
only showing top 5 rows
```

Broadcast Join and Bucketing

>>> nation_cust = customer_df.join(broadcast(nation_df),customer_df.C_NATIONKEY == nation_df.N_NATIONKEY,"inner")

>>> nation_cust.columns

>>> nation_cust.printSchema()

```
ak@ak: ~/project
>>> nation_cust = customer_df.join(broadcast(nation_df),cus
>>> nation_cust.columns
['C_CUSTKEY', 'C_NAME', 'C_ADDRESS', 'C_NATIONKEY', 'C_PHON
>>> nation_cust.printSchema()
root
 |-- C_CUSTKEY: integer (nullable = true)
 |-- C_NAME: string (nullable = true)
 |-- C_ADDRESS: string (nullable = true)
 |-- C_NATIONKEY: integer (nullable = true)
 |-- C_PHONE: string (nullable = true)
 |-- C_ACCTBAL: double (nullable = true)
 |-- C_MKTSEGMENT: string (nullable = true)
 |-- C_COMMENT: string (nullable = true)
 |-- N_NATIONKEY: integer (nullable = true)
 |-- N_NAME: string (nullable = true)
 |-- N_REGIONKEY: integer (nullable = true)
 |-- N_COMMENT: string (nullable = true)

>>>
```

>>> nc = nation_cust.alias('nc')

>>> id(nc)==id(nation_cust)

>>> nc.printSchema()

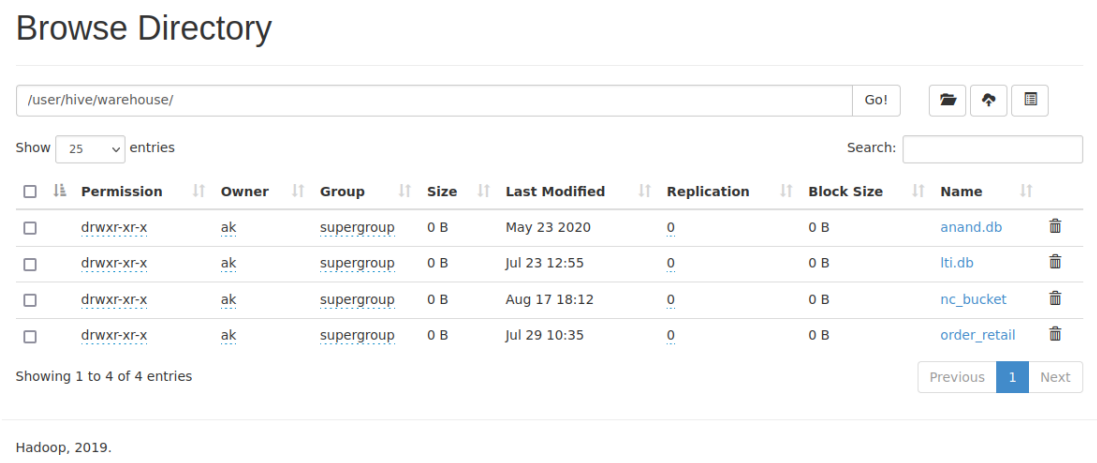
>>> nation_cust = nc.select(nc["C_CUSTKEY"].alias("CUSTKEY"), nc["C_NAME"].alias("CUSTNAME"), nc["C_NATIONKEY"].alias("NATIONKEY"), nc["N_NAME"].alias("NATION"), nc["N_REGIONKEY"].alias("REGIONKEY"))

>>> nation_cust.printSchema()

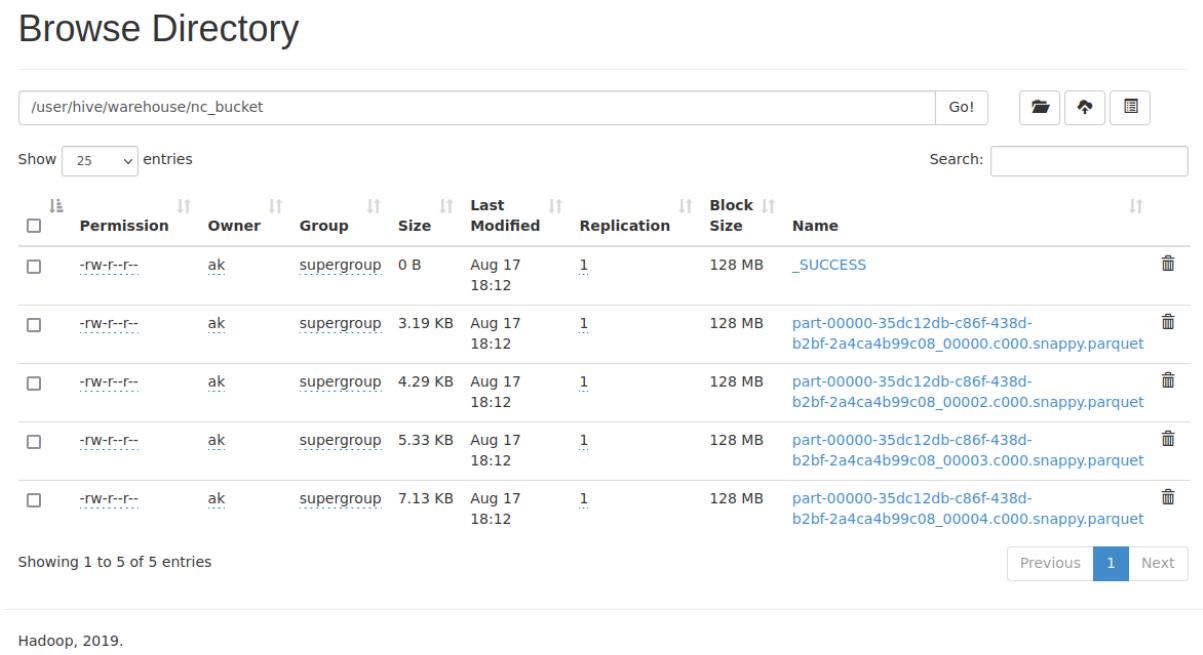
```
>>> nc = nation_cust.alias('nc')
>>> id(nc)==id(nation_cust)
False
>>> nc.printSchema()
root
 |-- C_CUSTKEY: integer (nullable = true)
 |-- C_NAME: string (nullable = true)
 |-- C_ADDRESS: string (nullable = true)
 |-- C_NATIONKEY: integer (nullable = true)
 |-- C_PHONE: string (nullable = true)
 |-- C_ACCTBAL: double (nullable = true)
 |-- C_MKTSEGMENT: string (nullable = true)
 |-- C_COMMENT: string (nullable = true)
 |-- N_NATIONKEY: integer (nullable = true)
 |-- N_NAME: string (nullable = true)
 |-- N_REGIONKEY: integer (nullable = true)
 |-- N_COMMENT: string (nullable = true)

>>> nation_cust = nc.select(nc["C_CUSTKEY"].alias("CUSTKEY"),
>>> nation_cust.printSchema()
root
 |-- CUSTKEY: integer (nullable = true)
 |-- CUSTNAME: string (nullable = true)
 |-- NATIONKEY: integer (nullable = true)
 |-- NATION: string (nullable = true)
 |-- REGIONKEY: integer (nullable = true)
```

>>> nation_cust.write.option("header","true").mode("overwrite").bucketBy(5,"NATION")\

.....sortBy("NATIONKEY").saveAsTable("NC_BUCKET", format="parquet")



Only 4 buckets were found in the table bucketed by Nation as one of the buckets was empty buckets and hence not created.




```
>>> nation_cust.write.option("header","true").mode("overwrite").bucketBy(5,"CUSTKEY")\
... .sortBy("NATIONKEY").saveAsTable("NC_BUCKET4", format="parquet")
```

Browse Directory

/user/hive/warehouse/nc_bucket4 Go!

Show 25 entries Search:

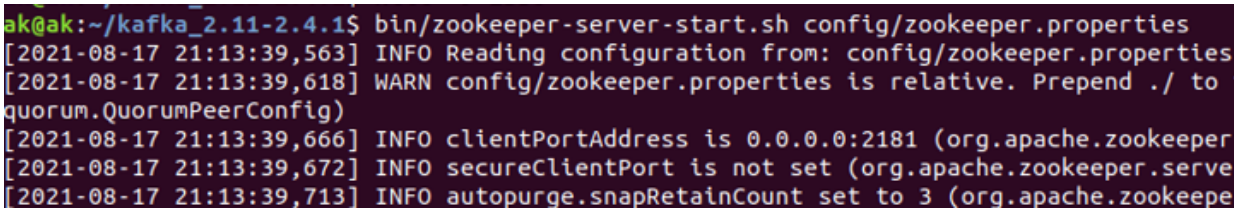
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	ak	supergroup	0 B	Aug 18 17:02	1	128 MB	_SUCCESS
-rw-r--r--	ak	supergroup	4.73 KB	Aug 18 17:02	1	128 MB	part-00000-6863d354-5f2a-4951-a287-231a5efa7817_00000.c000.snappy.parquet
-rw-r--r--	ak	supergroup	4.8 KB	Aug 18 17:02	1	128 MB	part-00000-6863d354-5f2a-4951-a287-231a5efa7817_00001.c000.snappy.parquet
-rw-r--r--	ak	supergroup	4.56 KB	Aug 18 17:02	1	128 MB	part-00000-6863d354-5f2a-4951-a287-231a5efa7817_00002.c000.snappy.parquet
-rw-r--r--	ak	supergroup	4.74 KB	Aug 18 17:02	1	128 MB	part-00000-6863d354-5f2a-4951-a287-231a5efa7817_00003.c000.snappy.parquet
-rw-r--r--	ak	supergroup	4.66 KB	Aug 18 17:02	1	128 MB	part-00000-6863d354-5f2a-4951-a287-231a5efa7817_00004.c000.snappy.parquet

Showing 1 to 6 of 6 entries Previous 1 Next

Kafka + Spark Structured Streaming + S3

Starting Zookeeper and Kafka

bin/zookeeper-server-start.sh config/zookeeper.properties



bin/kafka-server-start.sh config/server.properties

Create topic

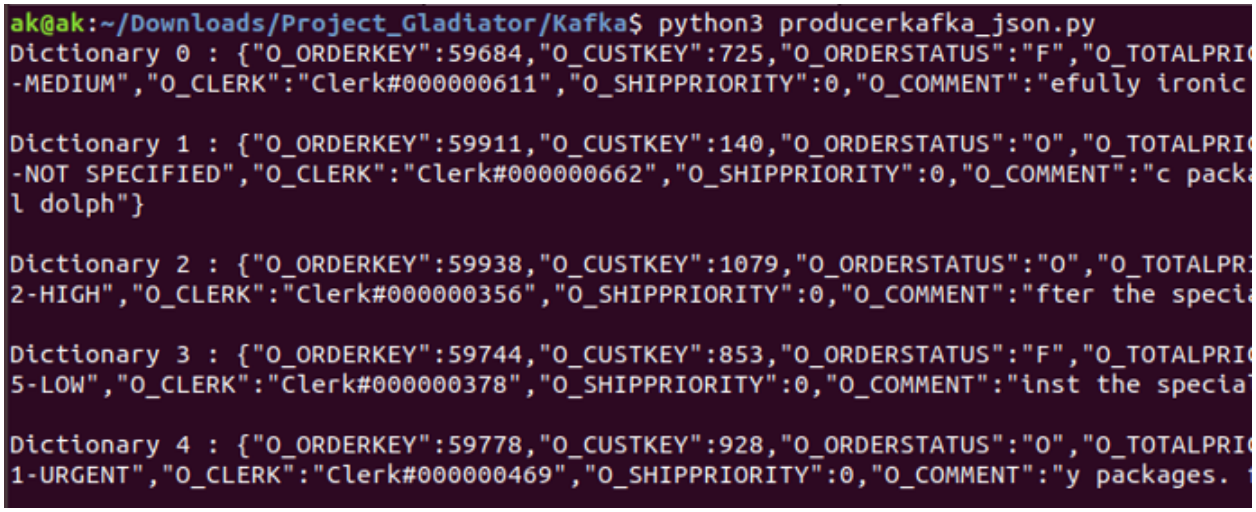
bin/kafka-topics.sh --create --zookeeper localhost:2181 --partitions 1 --replication-factor 1 --topic gladiator

Create and run Kafka Producer

producerkafka_json.py

```
from kafka import KafkaProducer
import json
import time

def ser_json(x):
    return x.encode('utf-8')

producer = KafkaProducer(bootstrap_servers=['localhost:9092'],value_serializer=ser_json)


if __name__ == "__main__":
        with open('/home/ak/Downloads/final_order/final_order_json.json', 'r') as lines:
                for line in lines:
                        print(f"Dictionary {count-1} : {line}")
                        producer.send('gladiator', line)
                        time.sleep(5)
```

python3 producerkafka_json.py



Spark Structured Streaming

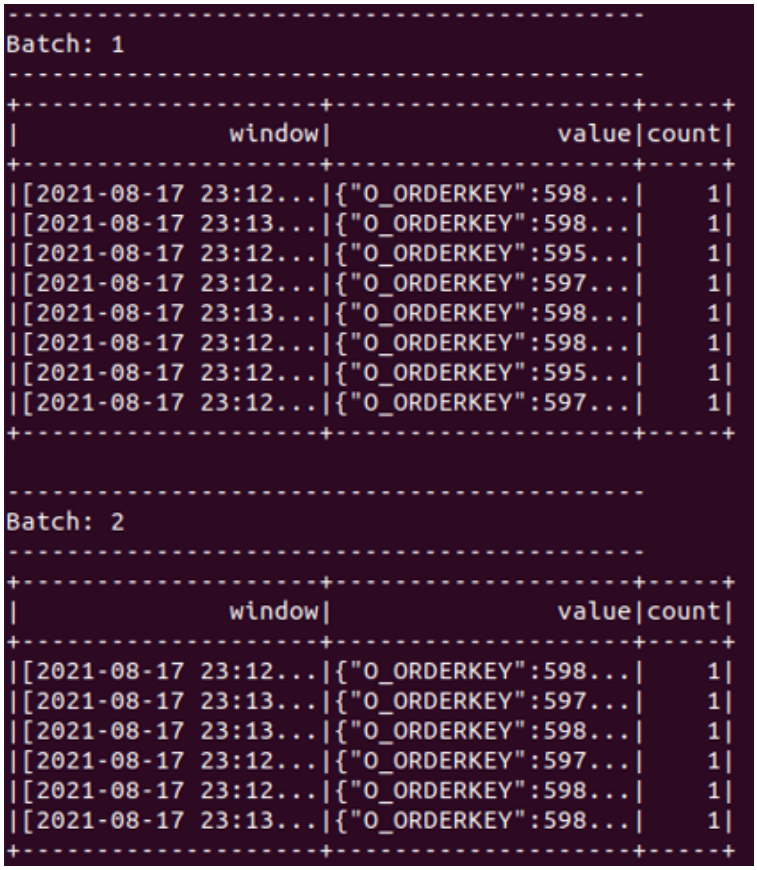
pyspark --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.1

-- Read data from producer stream
lines = spark.readStream.format("kafka").option("kafka.bootstrap.servers", "localhost:9092").option("subscribe", "gladiator").option("startingOffsets", "earliest").load()

-- Cast timestamp and value
newline = lines.selectExpr('CAST(timestamp AS TIMESTAMP)', 'CAST(value AS STRING)')

--Windowing and Watermarking
from pyspark.sql.functions import *
windowedCount = newline.withWatermark("timestamp", "5 minutes").groupBy(window(newline.timestamp, "5 minutes", "3 minutes"), newline.value).count()

-- Outputting the aggregated data on console
windowedCount.writeStream.outputMode("update").format("console").start().awaitTermination()



```
-------------------------------------------
Batch: 1
-------------------------------------------
+--------------------+--------------------+-----+
|              window|               value|count|
+--------------------+--------------------+-----+
|[2021-08-17 23:12...|{"O_ORDERKEY":598...|    1|
|[2021-08-17 23:13...|{"O_ORDERKEY":598...|    1|
|[2021-08-17 23:12...|{"O_ORDERKEY":595...|    1|
|[2021-08-17 23:12...|{"O_ORDERKEY":597...|    1|
|[2021-08-17 23:13...|{"O_ORDERKEY":598...|    1|
|[2021-08-17 23:12...|{"O_ORDERKEY":598...|    1|
|[2021-08-17 23:12...|{"O_ORDERKEY":595...|    1|
|[2021-08-17 23:12...|{"O_ORDERKEY":597...|    1|
+--------------------+--------------------+-----+


-------------------------------------------
Batch: 2
-------------------------------------------
+--------------------+--------------------+-----+
|              window|               value|count|
+--------------------+--------------------+-----+
|[2021-08-17 23:12...|{"O_ORDERKEY":598...|    1|
|[2021-08-17 23:13...|{"O_ORDERKEY":597...|    1|
|[2021-08-17 23:13...|{"O_ORDERKEY":598...|    1|
|[2021-08-17 23:12...|{"O_ORDERKEY":597...|    1|
|[2021-08-17 23:12...|{"O_ORDERKEY":598...|    1|
|[2021-08-17 23:13...|{"O_ORDERKEY":598...|    1|
+--------------------+--------------------+-----+
```

Spark Structured Streaming to S3

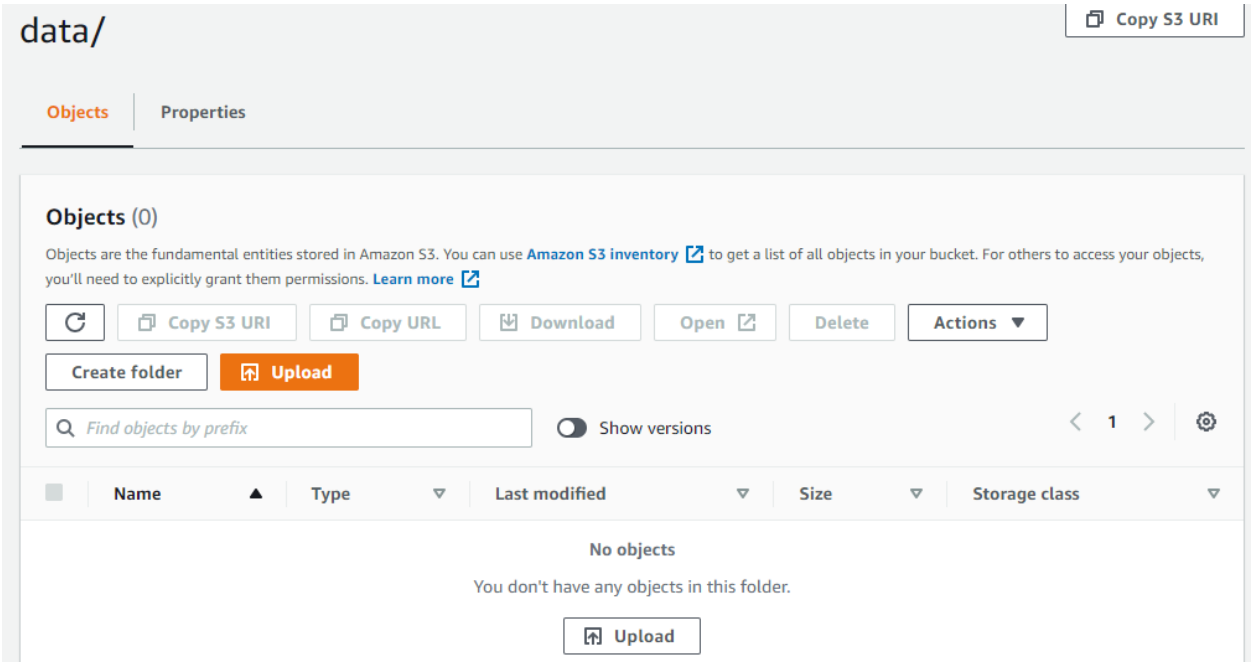
pyspark --packages
org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5,com.amazonaws:aws-java-sdk-pom:1.10.34,
org.apache.hadoop:hadoop-aws:2.6.0

-- Read data from producer stream
lines = spark.readStream.format("kafka").option("kafka.bootstrap.servers",
"localhost:9092").option("subscribe", "gladiator").option("startingOffsets", "latest").load()

-- Cast timestamp and value
newline = lines.selectExpr('CAST(timestamp AS TIMESTAMP)', 'CAST(value AS STRING)')

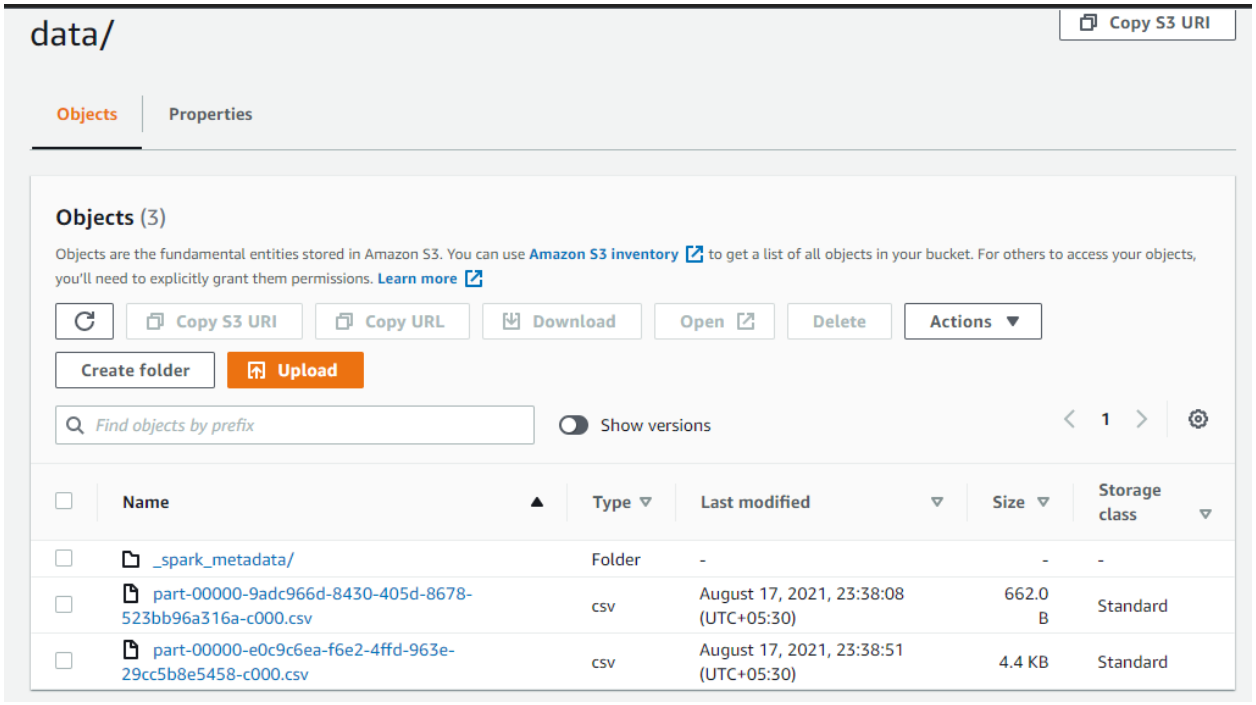
```
-- Send data to S3
query = newline.writeStream.outputMode("append")\
                        .format("csv")\
                        .option("path", "s3a://sheldonrodrigues/spark_streaming_data/data")\
                        .option("checkpointLocation",
"s3a://sheldonrodrigues/spark_streaming_data/checkpoint")\
                        .start()\
                        .awaitTermination()
```

Before :



After :


ak@ak: ~/kafka_2.11-2.4.1
ak@ak: ~/Downloads/Projec...
ak@ak: ~
ak@ak: ~/kafka_2.11-2.4.1
>>> lines = spark.readStream.format("kafka").option("kafka.bootstrap.servers", "localhost:9092").option("subscribe", "gla
tingOffsets", "latest").load()
>>> newline = lines.selectExpr('CAST(timestamp AS TIMESTAMP)', 'CAST(value AS STRING)')
>>> query = newline.writeStream.outputMode("append")\
... .format("csv")\
... .option("path", "s3a://sheldonrodrigues/spark_streaming_data/data")\
... .option("checkpointLocation", "s3a://sheldonrodrigues/spark_streaming_data/checkpoint")\
... .start()\
... .awaitTermination()


data/
Copy S3 URI
Objects
Properties
Objects (3)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Show versions
1
Name
Type
Last modified
Size
Storage class
_spark_metadata/
Folder
-
-
-
part-00000-9adc966d-8430-405d-8678-523bb96a316a-c000.csv
csv
August 17, 2021, 23:38:08 (UTC+05:30)
662.0 B
Standard
part-00000-e0c9c6ea-f6e2-4ffd-963e-29cc5b8e5458-c000.csv
csv
August 17, 2021, 23:38:51 (UTC+05:30)
4.4 KB
Standard

HIVE

External Table :

Create external table schema :

```
CREATE EXTERNAL TABLE retail.orders (
        o_orderkey int,
        o_custkey int,
        o_orderstatus STRING,
        o_totalprice int,
        o_orderdate STRING,
        o_orderpriority STRING,
        o_clerk STRING,
        o_shippriority int,
        o_comment STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION 'file:///home/ak/Downloads/Project_Gladiator/Hive';
```

```
hive> describe retail.orders;
OK
o_orderkey              int
o_custkey               int
o_orderstatus           string
o_totalprice            int
o_orderdate             string
o_orderpriority         string
o_clerk                 string
o_shippriority          int
o_comment               string
Time taken: 0.236 seconds, Fetched: 9 row(s)
hive>
```

```
hive> describe formatted retail.orders;
OK
# col_name              data_type               comment

o_orderkey              int
o_custkey               int
o_orderstatus           string
o_totalprice            int
o_orderdate             string
o_orderpriority         string
o_clerk                 string
o_shippriority          int
o_comment               string

# Detailed Table Information
Database:               retail
Owner:                  ak
CreateTime:             Wed Aug 18 15:58:27 IST 2021
LastAccessTime:         UNKNOWN
Retention:              0
Location:               file:/home/ak/Downloads/Project_Gladiator/Hive
Table Type:             EXTERNAL_TABLE
Table Parameters:
```

Copy orders.csv to external table location :
cp orders.csv /home/ak/Downloads/Project_Gladiator/Hive

Ignore headers from csv :

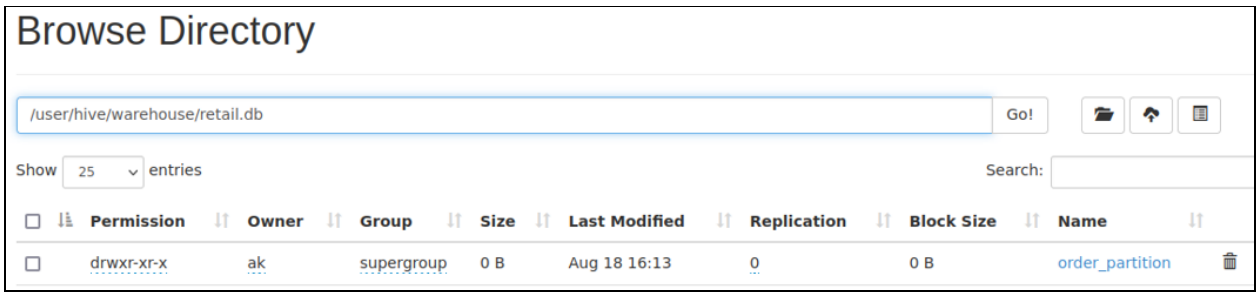
ALTER TABLE orders SET TBLPROPERTIES ("skip.header.line.count"="1");

Check data in table :
select * from orders LIMIT 5;

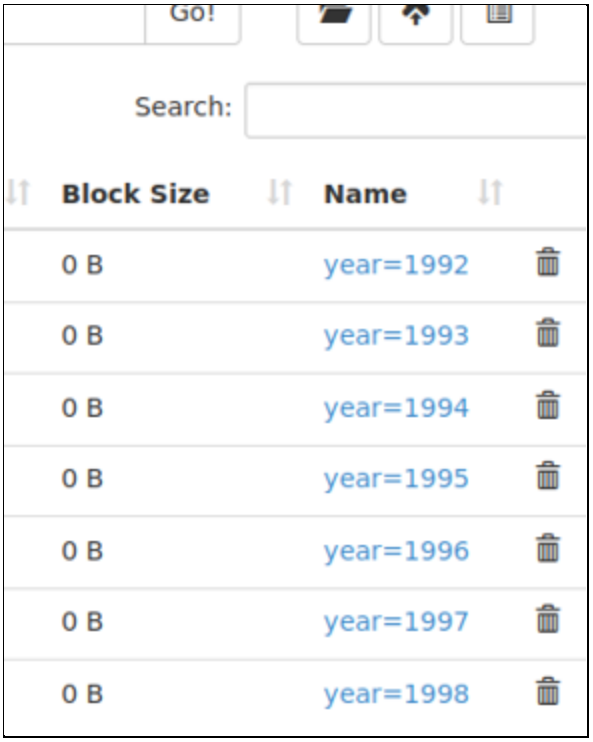
```
hive> select * from orders limit 5;
OK
1       370     O       172799  1996-01-02      5-LOW   Clerk#000000951 0       nstructions sleep furiously
2       781     O       38426   1996-12-01      1-URGENT        Clerk#000000880 0       foxes. pending acco
ot
3       1234    F       205654  1993-10-14      5-LOW   Clerk#000000955 0       sly final accounts boost. ca
ully. depos
4       1369    O       56000   1995-10-11      5-LOW   Clerk#000000124 0       sits. slyly regular warthogs
tes acro
5       445     F       105367  1994-07-30      5-LOW   Clerk#000000925 0       quickly. bold deposits sleep
Time taken: 0.521 seconds, Fetched: 5 row(s)
```

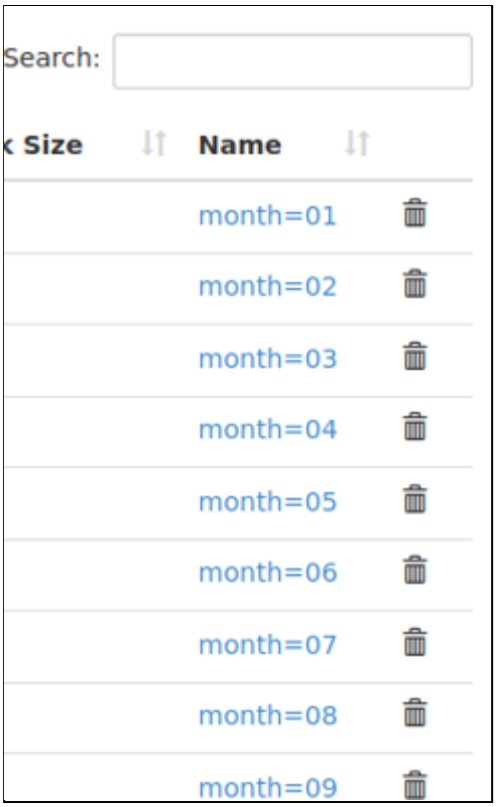
Managed table using partitions :
Create schema :
CREATE TABLE retail.order_partition (o_orderkey int, o_custkey int, o_orderstatus STRING, o_totalprice int, o_orderdate STRING, o_orderpriority STRING, o_clerk STRING, o_shippriority int, o_comment STRING) partitioned by (year string, month string);

```
hive> describe formatted order_partition;
OK
# col_name              data_type

o_orderkey              int
o_custkey               int
o_orderstatus           string
o_totalprice            int
o_orderdate             string
o_orderpriority         string
o_clerk                 string
o_shippriority          int
o_comment               string

# Partition Information
# col_name              data_type

year                    string
month                   string
```



Insert data into table :
INSERT INTO TABLE orders_partition PARTITION(year,month) select o_orderkey, o_custkey, o_orderstatus, o_totalprice, o_orderdate, o_orderpriority, o_clerk, o_shippriority, o_comment, substring(o_orderdate, 1, 4), substring(o_orderdate, 6, 2) from orders;

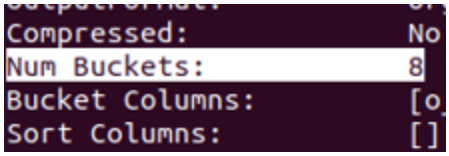




Bucketing :

Creating table for bucketing :
create table orders_bucket_1(o_orderkey int, o_custkey int, o_orderstatus string, o_totalprice double, o_orderpriority string, o_clerk string, o_orderdate string, o_shippriority int, o_comment string) clustered by (o_orderdate) into 8 buckets;

describe formatted orders_bucket_1;



Insert data into table :
insert overwrite table orders_bucket_1 select o_orderkey, o_custkey, o_orderstatus, o_totalprice, o_orderdate, o_orderpriority, o_clerk, o_shippriority, o_comment from orders;

DELTA TABLES

```
>>> orders.write.format("delta").save("file:///home/ak/pglad")
>>> deltaTable = DeltaTable.forPath(spark, 'file:///home/ak/pglad')

>>> deltaTable.update(col("O_ORDERSTATUS") == "F", { "O_ORDERSTATUS": lit("Fail") })
>>> deltadf2 = deltaTable.toDF().show()
```

```
>>> deltaTable.update(col("O_ORDERSTATUS") == "F", { "O_ORDERSTATUS": lit("Fail") })
>>> deltadf2 = deltaTable.toDF().show()
+----------+---------+-------------+-----------+-----------+---------------+---------------+-------------+--------------------+
|O_ORDERKEY|O_CUSTKEY|O_ORDERSTATUS|O_TOTALPRICE|O_ORDERDATE|O_ORDERPRIORITY|        O_CLERK|O_SHIPPRIORITY|           O_COMMENT|
+----------+---------+-------------+-----------+-----------+---------------+---------------+-------------+--------------------+
|         1|      370|            O|  172799.49| 1996-01-02|          5-LOW|Clerk#000000951|            0|nstructions sleep...|
|         2|      781|            O|   38426.09| 1996-12-01|       1-URGENT|Clerk#000000880|            0| foxes. pending a...|
|         3|     1234|         Fail|   205654.3| 1993-10-14|          5-LOW|Clerk#000000955|            0|sly final account...|
|         4|     1369|            O|   56000.91| 1995-10-11|          5-LOW|Clerk#000000124|            0|sits. slyly regul...|
|         5|      445|         Fail|  105367.67| 1994-07-30|          5-LOW|Clerk#000000925|            0|quickly. bold dep...|
|         6|      557|         Fail|    45523.1| 1992-02-21|4-NOT SPECIFIED|Clerk#000000058|            0|ggle. special, fi...|
|         7|      392|            O|  271885.66| 1996-01-10|         2-HIGH|Clerk#000000470|            0|ly special requests |
|        32|     1301|            O|  198665.57| 1995-07-16|         2-HIGH|Clerk#000000616|            0|ise blithely bold...|
|        33|      670|         Fail|  146567.24| 1993-10-27|       3-MEDIUM|Clerk#000000409|            0|uriously. furious...|
|        34|      611|            O|   73315.48| 1998-07-21|       3-MEDIUM|Clerk#000000223|            0|ly final packages...|
|        35|     1276|            O|  194641.93| 1995-10-23|4-NOT SPECIFIED|Clerk#000000259|            0|zzle. carefully e...|
|        36|     1153|            O|   42011.04| 1995-11-03|       1-URGENT|Clerk#000000358|            0| quick packages a...|
|        37|      862|         Fail|  131896.49| 1992-06-03|       3-MEDIUM|Clerk#000000456|            0|kly regular pinto...|
|        38|     1249|            O|   71553.08| 1996-08-21|4-NOT SPECIFIED|Clerk#000000604|            0|haggle blithely. ...|
|        39|      818|            O|  326565.37| 1996-09-20|       3-MEDIUM|Clerk#000000659|            0|ole express, iron...|
|        64|      322|         Fail|   35831.73| 1994-07-16|       3-MEDIUM|Clerk#000000661|            0|wake fluffily. so...|
|        65|      163|            P|   95469.44| 1995-03-18|       1-URGENT|Clerk#000000632|            0|ular requests are...|
|        66|     1292|         Fail|  104190.66| 1994-01-20|          5-LOW|Clerk#000000743|            0|y pending request...|
|        67|      568|            O|  182481.16| 1996-12-19|4-NOT SPECIFIED|Clerk#000000547|            0|symptotes haggle ...|
|        68|      286|            O|  301968.79| 1998-04-18|       3-MEDIUM|Clerk#000000440|            0| pinto beans slee...|
+----------+---------+-------------+-----------+-----------+---------------+---------------+-------------+--------------------+
```

>>> deltaTable.delete("O_ORDERPRIORITY='5-LOW'")
>>> deltadf3 = deltaTable.toDF().show()

```
>>> deltaTable.delete("O_ORDERPRIORITY='5-LOW'")
>>> deltadf3 = deltaTable.toDF().show()
+----------+---------+-------------+-----------+-----------+---------------+---------------+-------------+--------------------+
|O_ORDERKEY|O_CUSTKEY|O_ORDERSTATUS|O_TOTALPRICE|O_ORDERDATE|O_ORDERPRIORITY|        O_CLERK|O_SHIPPRIORITY|           O_COMMENT|
+----------+---------+-------------+-----------+-----------+---------------+---------------+-------------+--------------------+
|         2|      781|            O|   38426.09| 1996-12-01|       1-URGENT|Clerk#000000880|            0| foxes. pending a...|
|         6|      557|         Fail|    45523.1| 1992-02-21|4-NOT SPECIFIED|Clerk#000000058|            0|ggle. special, fi...|
|         7|      392|            O|  271885.66| 1996-01-10|         2-HIGH|Clerk#000000470|            0|ly special requests |
|        32|     1301|            O|  198665.57| 1995-07-16|         2-HIGH|Clerk#000000616|            0|ise blithely bold...|
|        33|      670|         Fail|  146567.24| 1993-10-27|       3-MEDIUM|Clerk#000000409|            0|uriously. furious...|
|        34|      611|            O|   73315.48| 1998-07-21|       3-MEDIUM|Clerk#000000223|            0|ly final packages...|
|        35|     1276|            O|  194641.93| 1995-10-23|4-NOT SPECIFIED|Clerk#000000259|            0|zzle. carefully e...|
|        36|     1153|            O|   42011.04| 1995-11-03|       1-URGENT|Clerk#000000358|            0| quick packages a...|
|        37|      862|         Fail|  131896.49| 1992-06-03|       3-MEDIUM|Clerk#000000456|            0|kly regular pinto...|
|        38|     1249|            O|   71553.08| 1996-08-21|4-NOT SPECIFIED|Clerk#000000604|            0|haggle blithely. ...|
|        39|      818|            O|  326565.37| 1996-09-20|       3-MEDIUM|Clerk#000000659|            0|ole express, iron...|
|        64|      322|         Fail|   35831.73| 1994-07-16|       3-MEDIUM|Clerk#000000661|            0|wake fluffily. so...|
|        65|      163|            P|   95469.44| 1995-03-18|       1-URGENT|Clerk#000000632|            0|ular requests are...|
|        67|      568|            O|  182481.16| 1996-12-19|4-NOT SPECIFIED|Clerk#000000547|            0|symptotes haggle ...|
|        68|      286|            O|  301968.79| 1998-04-18|       3-MEDIUM|Clerk#000000440|            0| pinto beans slee...|
|        69|      845|         Fail|  204110.73| 1994-06-04|4-NOT SPECIFIED|Clerk#000000330|            0| depths atop the ...|
|        71|       34|            O|  260603.38| 1998-01-24|4-NOT SPECIFIED|Clerk#000000271|            0| express deposits...|
|        96|     1078|         Fail|   64364.3| 1994-04-17|         2-HIGH|Clerk#000000395|            0|oost furiously. p...|
|        97|      211|         Fail|  100572.55| 1993-01-29|       3-MEDIUM|Clerk#000000547|            0|hang blithely alo...|
|        98|     1045|         Fail|    71721.4| 1994-09-25|       1-URGENT|Clerk#000000448|            0|c asymptotes. qui...|
+----------+---------+-------------+-----------+-----------+---------------+---------------+-------------+--------------------+
```

Business Queries

Q1. **Which manufacturer gets the most returned items?**

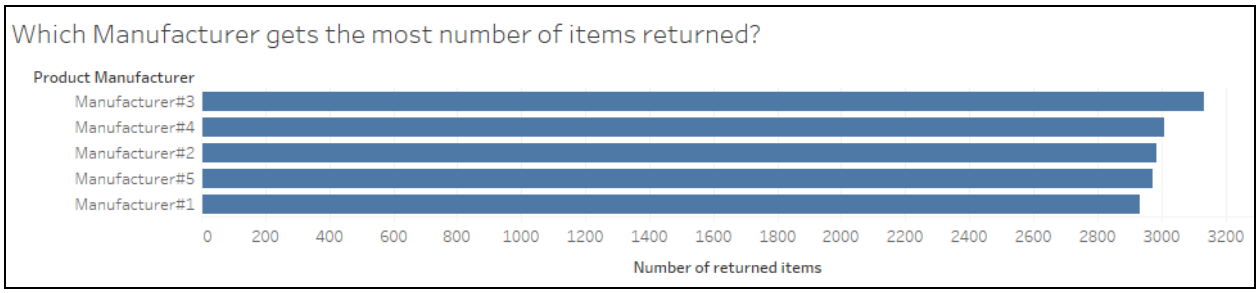
> lineitem_casted = lineitem.withColumn('L_QUANTITY', col('L_QUANTITY').cast('int'))
> lineitem_return_instruction = lineitem_casted.filter('L_SHIPINSTRUCT=="TAKE BACK RETURN"')

```
>>> lineitem_casted = lineitem.withColumn('L_QUANTITY', col('L_QUANTITY').cast('int'))
>>> lineitem_return_instruction = lineitem_casted.filter('L_SHIPINSTRUCT=="TAKE BACK RETURN"')
>>> lineitem_return_instruction.select('L_SHIPINSTRUCT').show(3)
+----------------+
|  L_SHIPINSTRUCT|
+----------------+
|TAKE BACK RETURN|
|TAKE BACK RETURN|
|TAKE BACK RETURN|
+----------------+
only showing top 3 rows
```

FOR MANUFACTURER
> part_line_joined=part.join(lineitem_return_instruction, part.P_PARTKEY==lineitem_return_instruction.L_PARTKEY, 'inner')

> manufacturer_with_most_returned_items = part_line_joined.groupBy('P_MFGR').count().sort('count', ascending=False).withColumnRenamed('count', 'most returned items')
> manufacturer_with_most_returned_items = manufacturer_with_most_returned_items.limit(5)

```
+--------------------+------------------------+
|Product Manufacturer|Number of returned items|
+--------------------+------------------------+
|      Manufacturer#3|                    3133|
|      Manufacturer#4|                    3008|
|      Manufacturer#2|                    2986|
|      Manufacturer#5|                    2973|
|      Manufacturer#1|                    2934|
+--------------------+------------------------+
```

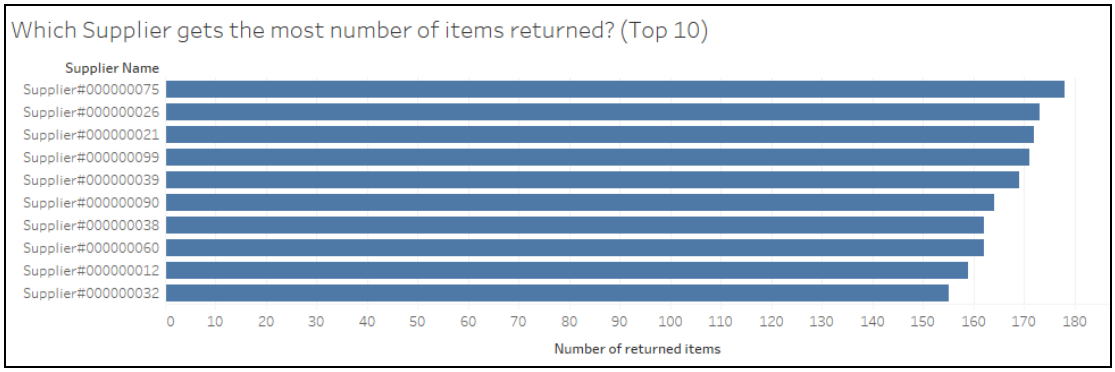


Q2. **Which suppliers get the most returned parts?**

> supp_line_joined = supplier.join(lineitem_return_instruction, supplier.S_SUPPKEY==lineitem_return_instruction.L_SUPPKEY, inner')

> supplier_with_most_returned_items = supp_line_joined.groupBy('S_NAME').count().sort('count', ascending=False).withColumnRenamed('count', 'most returned items')
> supplier_with_most_returned_items = supplier_with_most_returned_items.limit(5)

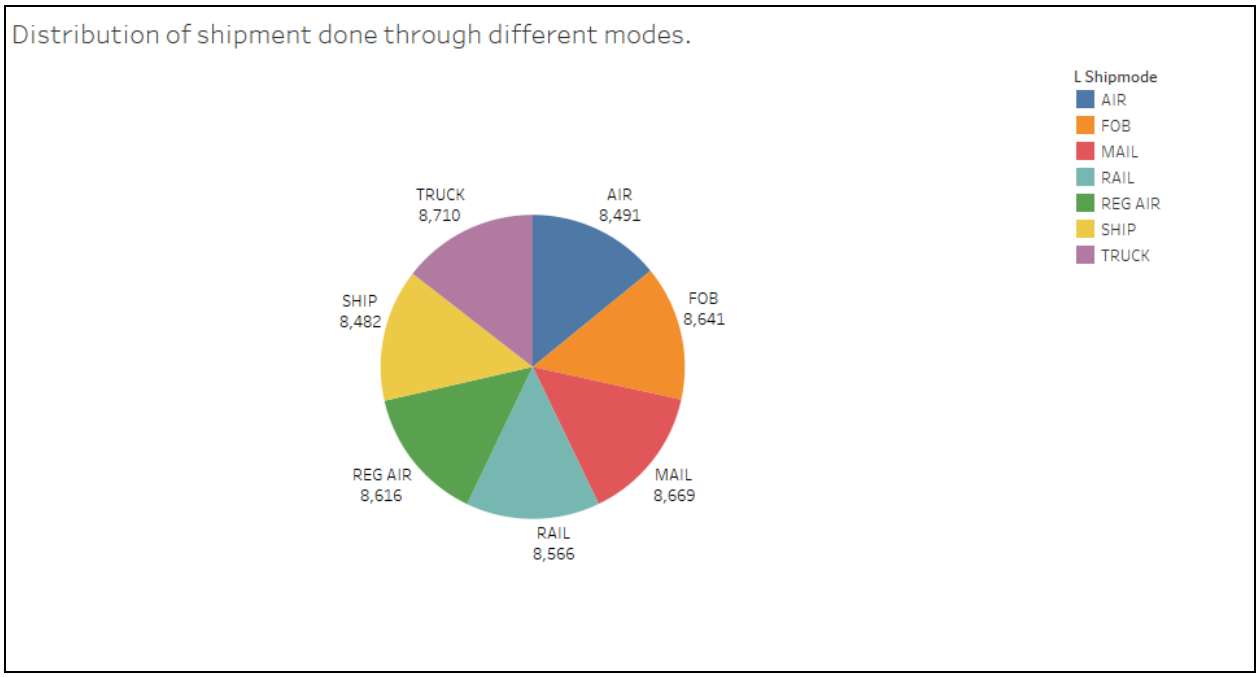
Supplier Name	Number of returned items
Supplier#000000075	178
Supplier#000000095	174
Supplier#000000026	173
Supplier#000000021	172
Supplier#000000099	171
Supplier#000000009	171
Supplier#000000089	170
Supplier#000000019	170
Supplier#000000084	169
Supplier#000000039	169



Q3.**Distribution of shipments done through different modes**

shipmodes_distribution = lineitem.groupBy(lineitem.L_SHIPMODE).count().sort('count', ascending=False)
shipmodes_distribution.show()

```
+----------------+--------------------+
|Mode of Shipment|Number of Shipments|
+----------------+--------------------+
|           TRUCK|                8710|
|            MAIL|                8669|
|             FOB|                8641|
|         REG AIR|                8616|
|            RAIL|                8566|
|             AIR|                8491|
|            SHIP|                8482|
+----------------+--------------------+
```



Q.4 Which nation orders the most amount of items?

> orders_cust_join = orders.join(customer, orders.O_CUSTKEY==customer.C_CUSTKEY, 'inner').select('C_NATIONKEY')

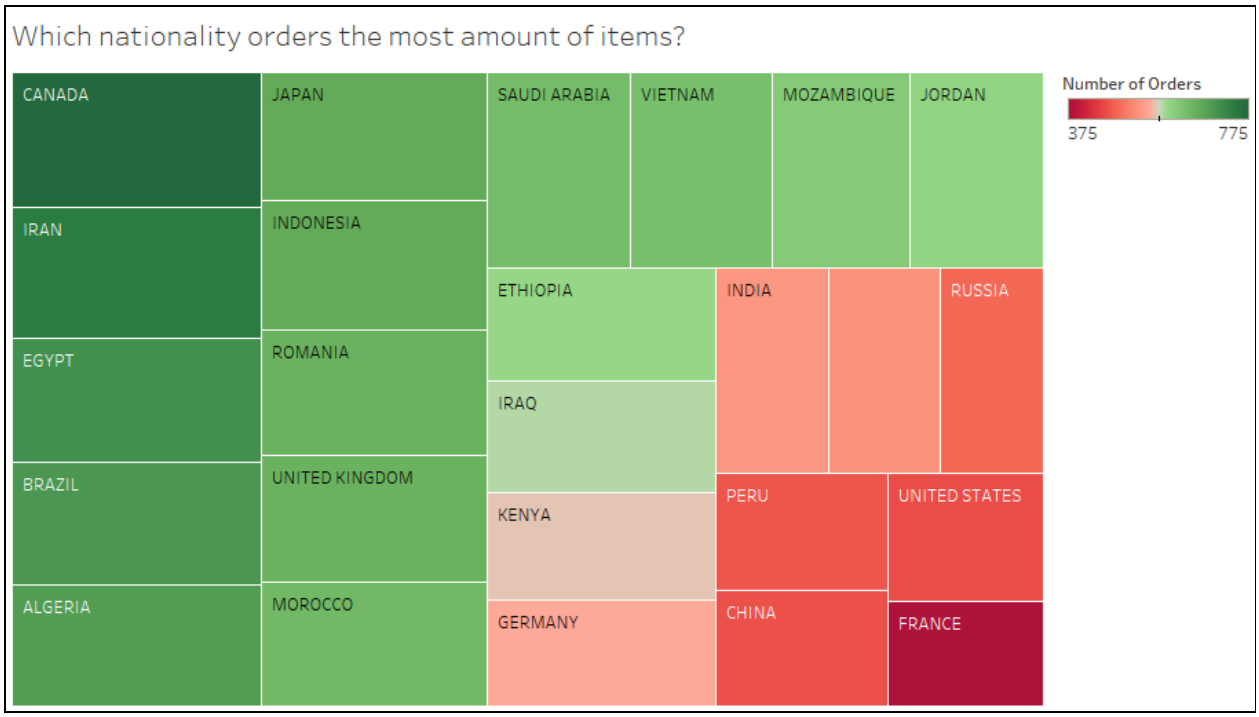
```
>>> orders_cust_join = orders.join(customer, orders.O_CUSTKEY==customer.C
>>> orders_cust_join.printSchema()
root
 |-- C_NATIONKEY: integer (nullable = true)

>>> orders_cust_join.show(5)
+-----------+
|C_NATIONKEY|
+-----------+
|         12|
|         18|
|          1|
|         10|
|         20|
+-----------+
only showing top 5 rows
```

>>> nations_with_most_orders = orders_cust_join.join(nation, orders_cust_join.C_NATIONKEY==nation.N_NATIONKEY, inner').select('N_NAME')

```
>>> nations_with_most_orders = orders_cust_join.
>>> nations_with_most_orders.printSchema()
root
 |-- N_NAME: string (nullable = true)

>>> nations_with_most_orders.show(5)
+------------+
|      N_NAME|
+------------+
|       JAPAN|
|       CHINA|
|   ARGENTINA|
|        IRAN|
|SAUDI ARABIA|
+------------+
only showing top 5 rows
```

> nations_with_most_orders = nations_with_most_orders.groupBy('N_NAME').count().sort('count', ascending=False).withColumnRenamed('N_NAME', ‘Country').withColumnRenamed('count', ‘Number of Orders').limit(5)

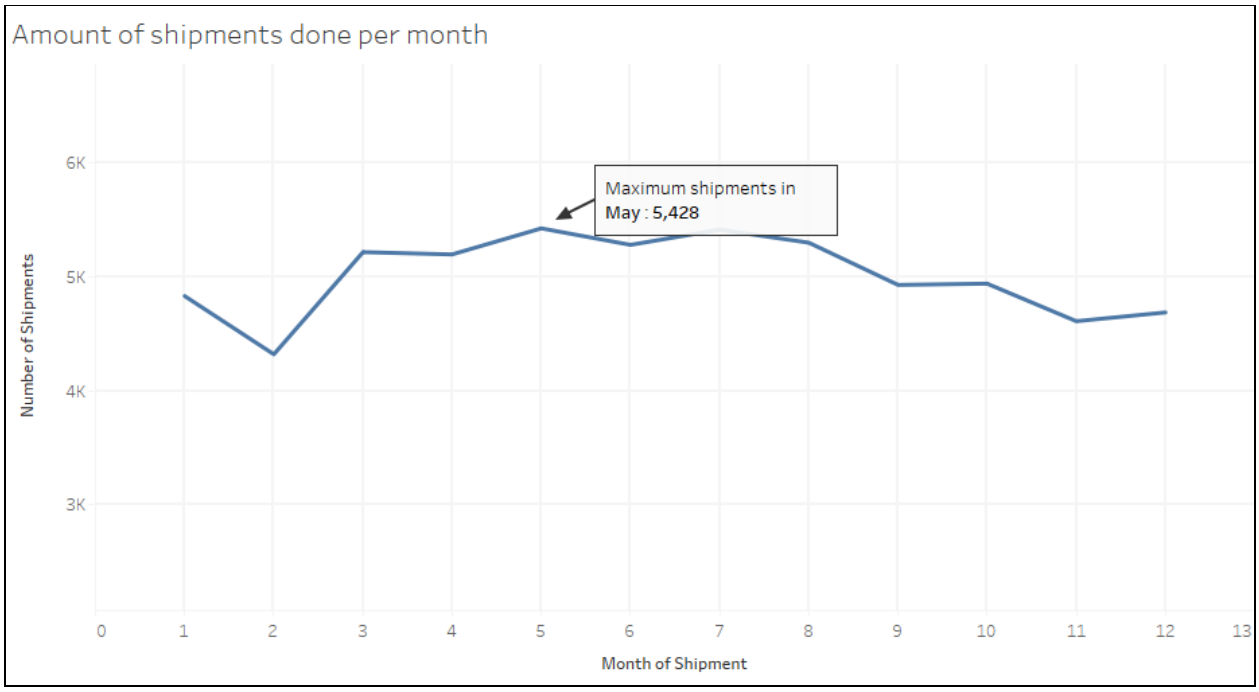
Country	Number of Orders
CANADA	775
IRAN	745
EGYPT	712
BRAZIL	700
ALGERIA	691



Q5. **In which month are the most shipments done?**

```
> q5 = lineitem.withColumn('new_month',
month('L_SHIPDATE')).groupBy('new_month').count().sort('count', ascending=False)
> q5.show()
```

Month	Number of Shipments
5	5428
7	5416
8	5302
6	5283
3	5219
4	5199
10	4942
9	4931
1	4833
12	4689
11	4612
2	4321



>>>cust1= customer.join(nation, customer.C_NATIONKEY==nation.N_NATIONKEY, "left").select("C_NAME","C_NATIONKEY", "N_NATIONKEY")

>>> spark.conf.set("spark.sql.crossJoin.enabled","true")

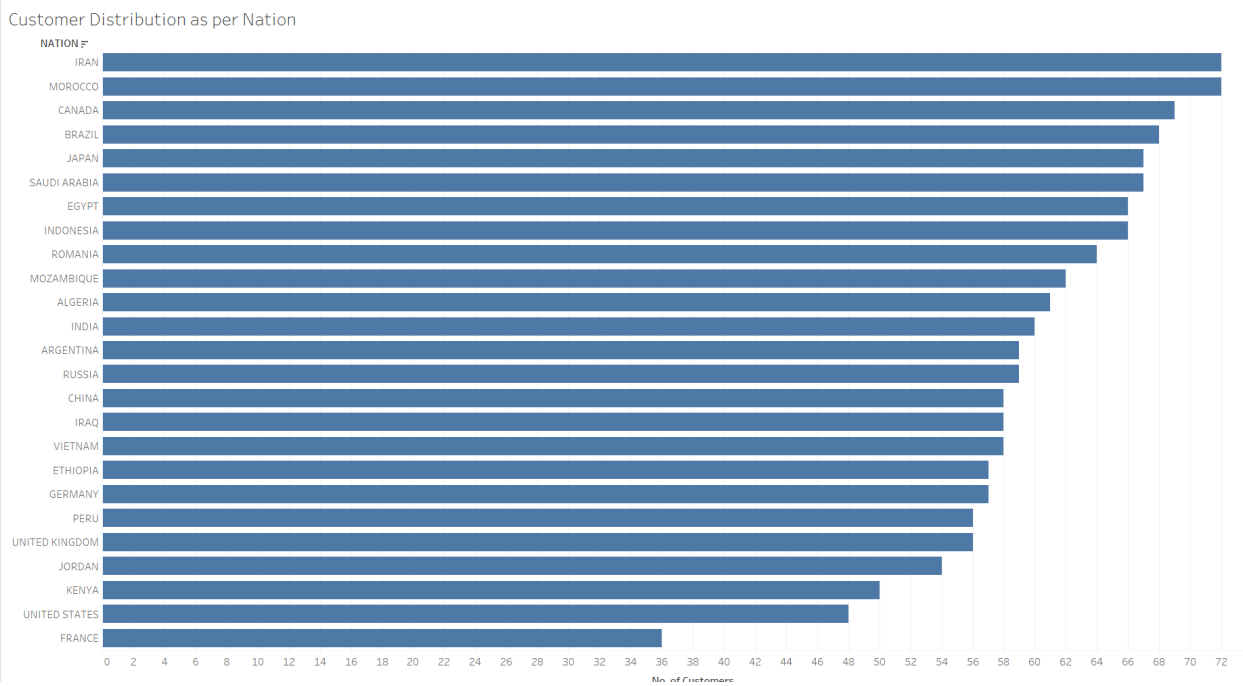
Q.6 Which nation has the most customers?

>>> nations_with_most_customers = cust1.join(nation, cust1.C_NATIONKEY==nation.N_NATIONKEY,'left').select("N_NAME")
>>> nations_with_most_customers.printSchema()
root
 |-- N_NAME: string (nullable = true)
nations_with_most_customers.show(5)

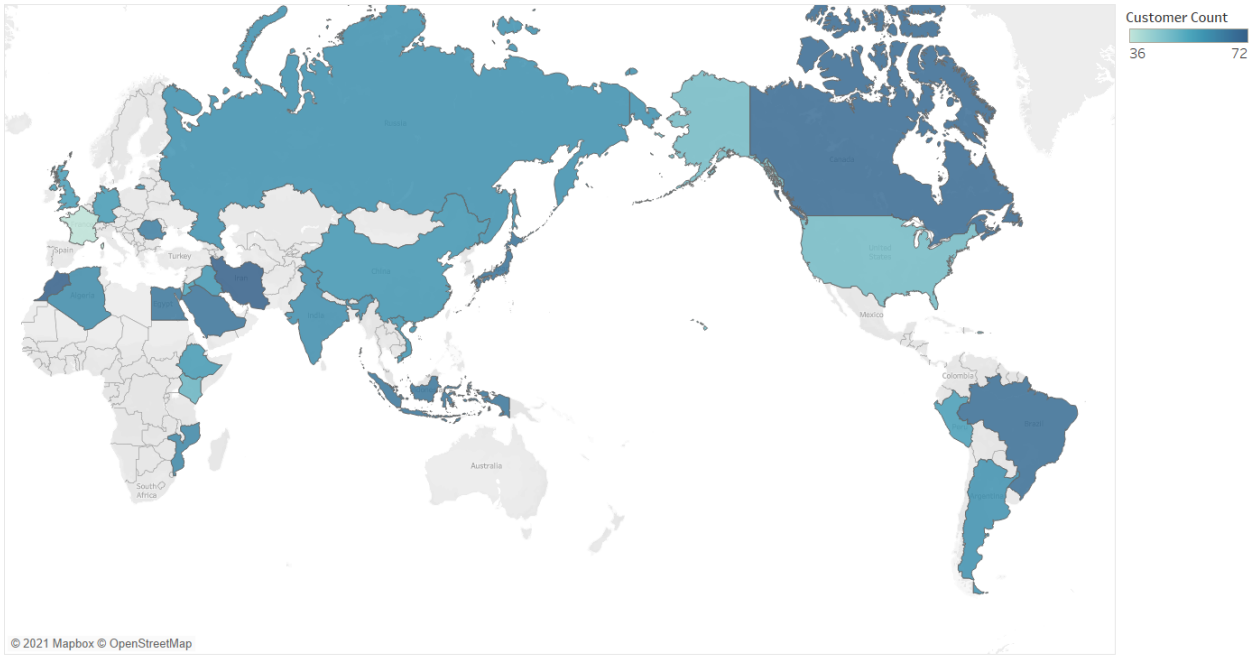
```
+--------------+---------------+
|        NATION|No_of_Customers|
+--------------+---------------+
|          IRAN|             72|
|       MOROCCO|             72|
|        CANADA|             69|
|        BRAZIL|             68|
|         JAPAN|             67|
|  SAUDI ARABIA|             67|
|         EGYPT|             66|
|     INDONESIA|             66|
|       ROMANIA|             64|
|    MOZAMBIQUE|             62|
|       ALGERIA|             61|
|         INDIA|             60|
|        RUSSIA|             59|
|     ARGENTINA|             59|
|         CHINA|             58|
|       VIETNAM|             58|
|          IRAQ|             58|
|      ETHIOPIA|             57|
|       GERMANY|             57|
|UNITED KINGDOM|             56|
+--------------+---------------+
only showing top 20 rows
```

>>>nations_with_most_customers=nations_with_most_customers.groupBy("N_NAME").count().sort('count', ascending=False).limit(3)
>>> nations_with_most_customers.show()

```
+--------------+---------------+
|        NATION|No_of_Customers|
+--------------+---------------+
|          IRAN|             72|
|       MOROCCO|             72|
|        CANADA|             69|
```



Customer Distribution as per Nation



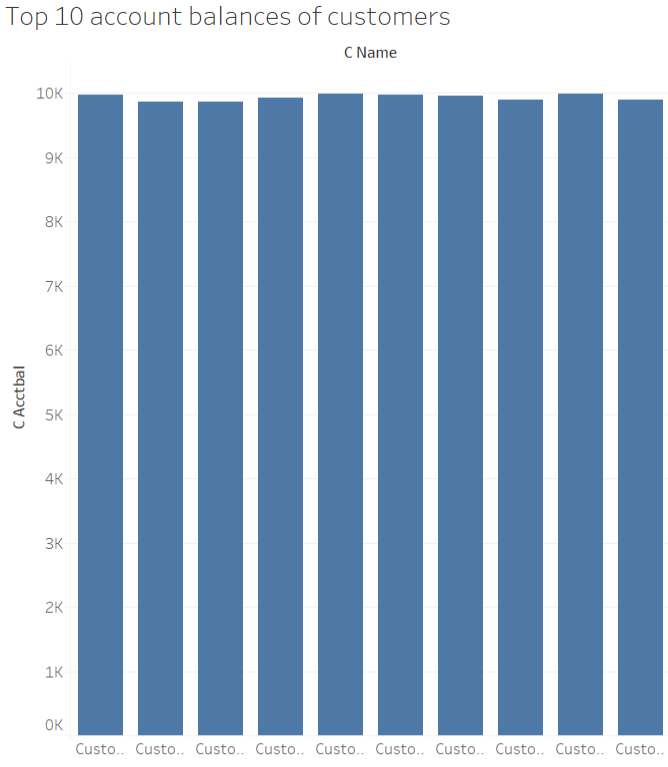
Map based on Longitude (generated) and Latitude (generated). Color shows distinct count of C_CUSTKEY. Details are shown for N_NAME.

Q.7 Top 10 account balances:

>>>

customer.sort(customer.C_ACCTBAL.desc()).limit(10).select(customer.C_ACCTBAL).show()

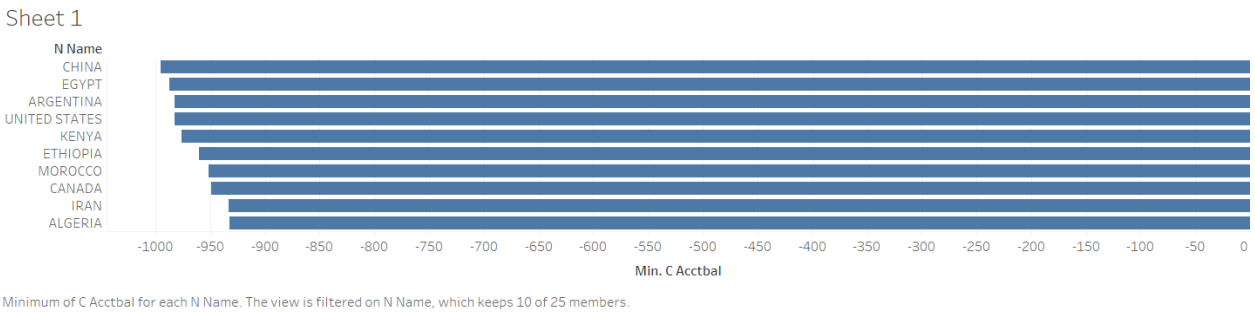
```
>>> customer.sort(customer.C_ACCTBAL.desc()).limit(10).select(customer.C_ACCTBAL
).show()
+---------+
|C_ACCTBAL|
+---------+
|  9987.71|
|  9983.38|
|  9977.62|
|   9967.6|
|  9963.15|
|  9931.71|
|  9904.28|
|  9889.89|
|  9874.12|
|  9871.66|
+---------+
```



Q.8 top 10 debt accounts by nation

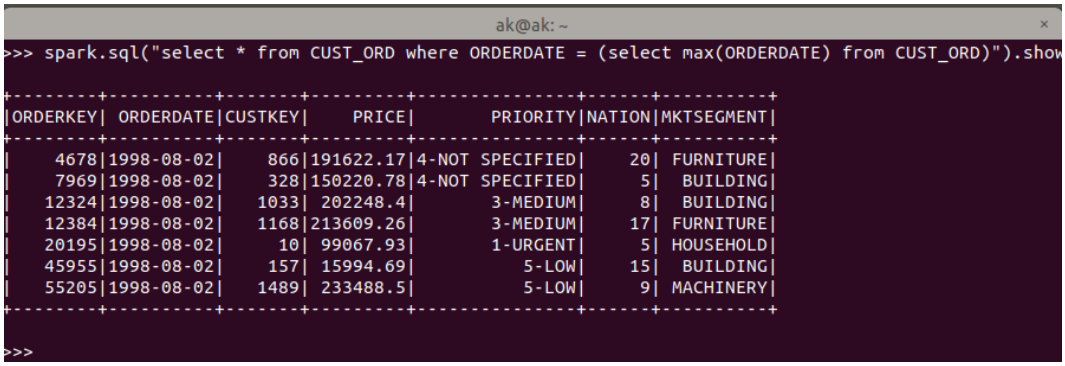
>>> cust2 = customer.join(nation,customer.C_NATIONKEY==nation.N_NATIONKEY, "inner")
>>> cust2.sort(cust2.C_ACCTBAL.asc()).limit(10).select(cust2.C_NAME, cust2.N_NAME, cust2.C_ACCTBAL).show()

```
>>> cust2 = customer.join(nation,customer.C_NATIONKEY==nation.N_NATIONKEY, "inne
r")
>>> cust2.sort(cust2.C_ACCTBAL.asc()).limit(10).select(cust2.C_NAME, cust2.N_NAM
E, cust2.C_ACCTBAL).show()
+------------------+-------------+---------+
|            C_NAME|       N_NAME|C_ACCTBAL|
+------------------+-------------+---------+
|Customer#000000294|        CHINA|  -994.79|
|Customer#000000128|        EGYPT|  -986.96|
|Customer#000001234|    ARGENTINA|  -982.32|
|Customer#000001235|UNITED STATES|  -982.05|
|Customer#000000834|        KENYA|  -976.25|
|Customer#000000885|     ETHIOPIA|  -959.94|
|Customer#000001013|      MOROCCO|  -951.53|
|Customer#000000875|       CANADA|  -949.28|
|Customer#000001244|     ETHIOPIA|   -942.8|
|Customer#000001187|         IRAN|  -932.96|
+------------------+-------------+---------+
```



Q.9 Orders placed on the last day of records

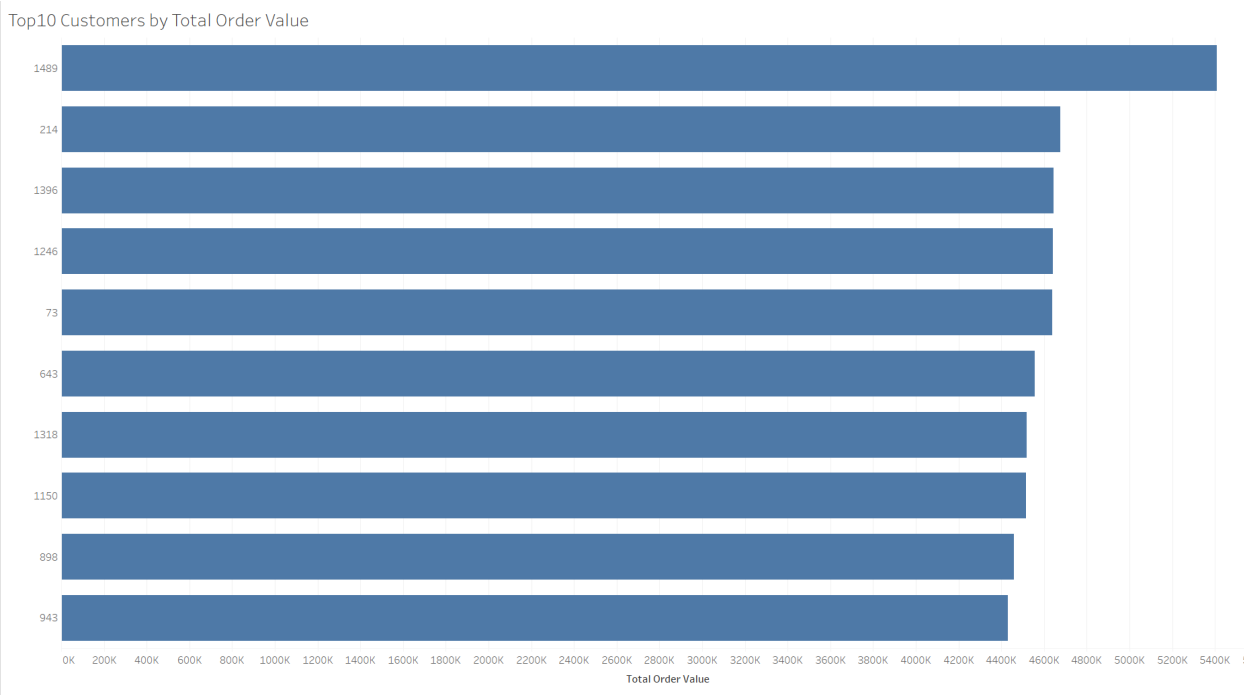
>>> spark.sql("select * from CUST_ORD where ORDERDATE = (select max(ORDERDATE) from CUST_ORD)").show()



Q.10 Top 10 customers by Highest Total Order Value

>>> spark.sql("select CUSTKEY, count(ORDERKEY), sum(PRICE), NATION, MKTSEGMENT from CUST_ORD group by CUSTKEY,NATION,MKTSEGMENT order by sum(PRICE) desc, count(ORDERKEY) desc limit 10").show()

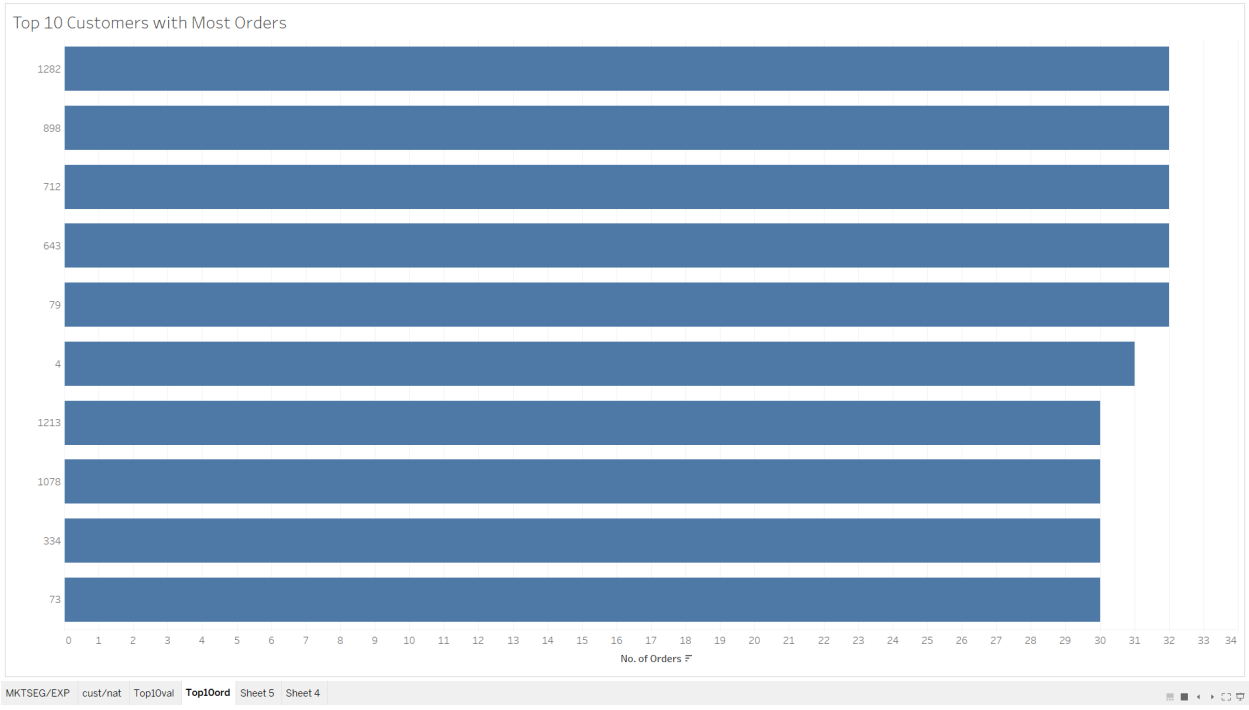
```
+-------+---------------+-----------------+------+----------+
|CUSTKEY|count(ORDERKEY)|       sum(PRICE)|NATION|MKTSEGMENT|
+-------+---------------+-----------------+------+----------+
|   1489|             29|       5408941.28|     9| MACHINERY|
|    214|             25|4674894.729999999|     8| MACHINERY|
|   1396|             28|4644936.890000001|     3|  BUILDING|
|   1246|             27|4642942.330000001|     4|  BUILDING|
|     73|             30|       4638819.21|     0|  BUILDING|
|    643|             32|4555789.040000001|     0| FURNITURE|
|   1318|             29|4520525.109999999|    14|  BUILDING|
|   1150|             28|       4516089.38|    21| MACHINERY|
|    898|             32|        4460059.6|     3|AUTOMOBILE|
|    943|             29|4432159.859999999|     5| FURNITURE|
+-------+---------------+-----------------+------+----------+
```



Q.11 Top 10 customers with Highest No. of Orders

>>> spark.sql("select CUSTKEY, count(ORDERKEY), sum(PRICE), NATION, MKTSEGMENT from CUST_ORD group by CUSTKEY,NATION,MKTSEGMENT order by count(ORDERKEY) desc, sum(PRICE) desc limit 10").show()

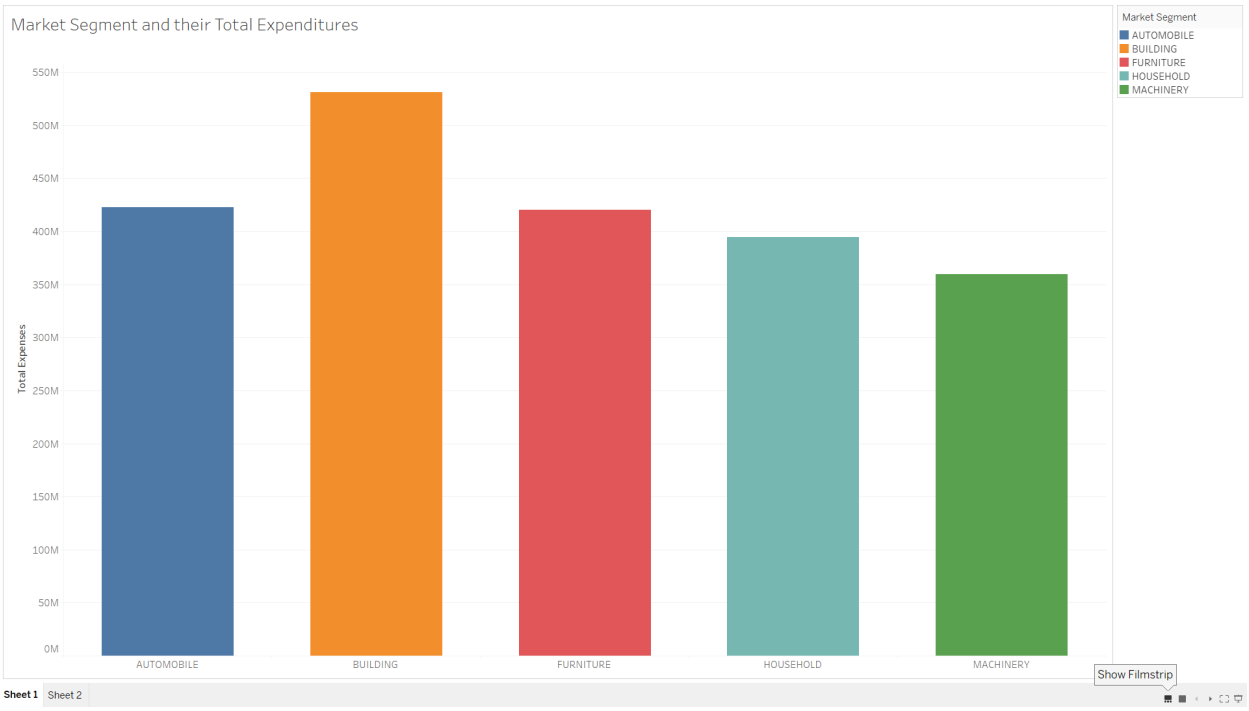
```
+-------+---------------+------------------+------+----------+
|CUSTKEY|count(ORDERKEY)|        sum(PRICE)|NATION|MKTSEGMENT|
+-------+---------------+------------------+------+----------+
|    643|             32| 4555789.040000001|     0| FURNITURE|
|    898|             32|         4460059.6|     3|AUTOMOBILE|
|     79|             32|        4412540.44|    15| MACHINERY|
|    712|             32|4152639.3199999994|     6|  BUILDING|
|   1282|             32|        3744680.68|    14| FURNITURE|
|      4|             31|4134567.3900000006|     4| MACHINERY|
|     73|             30|        4638819.21|     0|  BUILDING|
|    334|             30| 4246946.100000001|     4|  BUILDING|
|   1213|             30|        3813785.64|     7| HOUSEHOLD|
|   1078|             30|3607874.9900000007|    19| MACHINERY|
+-------+---------------+------------------+------+----------+
```



Q.12 Market segment in order of Expenditures

>>>cust_ord.groupBy('MKTSEGMENT').sum('PRICE').orderBy(sum('PRICE').desc()).show()

```
File  Edit  View  Search  Terminal  Tabs  Help
                                ak@ak: ~
>>> cust_ord.groupBy('MKTSEGMENT').sum('PRICE').orderBy(sum('PRICE').desc()).show()
+----------+--------------------+
|MKTSEGMENT|          sum(PRICE)|
+----------+--------------------+
|  BUILDING|5.3090349559999925E8|
|AUTOMOBILE| 4.2250410147999996E8|
| FURNITURE|4.1995199946000046E8|
| HOUSEHOLD| 3.9444706985999992E8|
| MACHINERY| 3.5959016361999999E8|
+----------+--------------------+

>>>
```

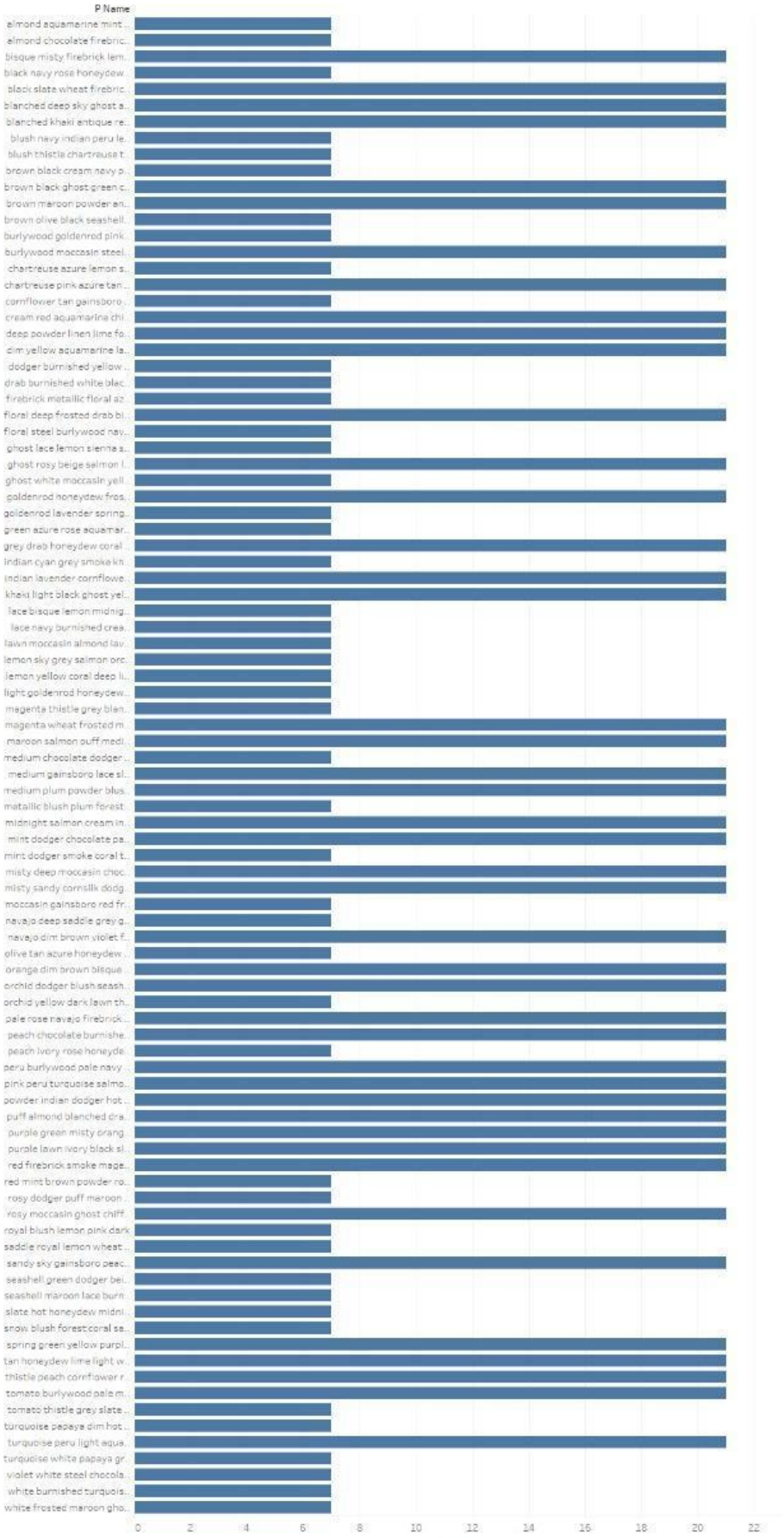


Q13 Get the records where Part Size is either 7 or 21.

>>> spark.sql("select * from PART p where P_size in ('7','21')").show()

```
>>> spark.sql("select * from PART p where P_size in ('7','21')").show()
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
|P_partkey|              P_name|       P_mfgr| P_brand|              P_type|P_size|P_container|P_retailprice|           P_comment|
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
|        1|goldenrod lavende...|Manufacturer#1|Brand#13|PROMO BURNISHED C...|     7|  JUMBO PKG|       901.00|     ly. slyly ironi|
|        3|spring green yell...|Manufacturer#4|Brand#42|STANDARD POLISHED...|    21|  WRAP CASE|       903.00| egular deposits hag|
|       29|lemon sky grey sa...|Manufacturer#3|Brand#33| PROMO PLATED COPPER|     7|    LG DRUM|       929.02|    carefully fluffi|
|       41|burlywood goldenr...|Manufacturer#2|Brand#23|ECONOMY ANODIZED TIN|     7|   WRAP JAR|       941.04|uriously. furious...|
|       81|misty sandy corns...|Manufacturer#5|Brand#53| ECONOMY BRUSHED TIN|    21|    MED BAG|       981.08|      ove the furiou|
|       89|ghost lace lemon ...|Manufacturer#5|Brand#53|STANDARD BURNISHE...|     7|    MED JAR|       989.08|      y final pinto |
|      139|floral steel burl...|Manufacturer#3|Brand#32|MEDIUM BRUSHED STEEL|     7|     SM BOX|      1039.13|               ter t|
|      150|pale rose navajo ...|Manufacturer#3|Brand#35|   LARGE BRUSHED TIN|    21|     SM BAG|      1050.15|        ironic foxes|
|      180|seashell maroon l...|Manufacturer#3|Brand#33|STANDARD BURNISHE...|     7|   WRAP BAG|      1080.18|            oss the |
|      194|brown black cream...|Manufacturer#5|Brand#51|ECONOMY POLISHED ...|     7|     SM CAN|      1094.19|    y special accoun|
|      221|chartreuse azure ...|Manufacturer#5|Brand#51|PROMO BRUSHED COPPER|     7|    LG DRUM|      1121.22|ld asymptotes sle...|
|      226|blush navy indian...|Manufacturer#3|Brand#33|PROMO ANODIZED STEEL|     7|     SM PKG|      1126.22| ular packages. some|
|      227|dim yellow aquama...|Manufacturer#4|Brand#44|   SMALL BRUSHED TIN|    21|    LG PACK|      1127.22|             silent r|
|      281|red mint brown po...|Manufacturer#1|Brand#12|  SMALL PLATED STEEL|     7|    MED CAN|      1181.28|               regula|
|      303|almond chocolate ...|Manufacturer#4|Brand#45|ECONOMY PLATED BRASS|     7| JUMBO CASE|      1203.30|       yly according|
|      329|mint dodger choco...|Manufacturer#3|Brand#31|MEDIUM ANODIZED N...|    21|     LG PKG|      1229.32|. quickly pending pa|
|      343|blush thistle cha...|Manufacturer#3|Brand#35|PROMO ANODIZED NI...|     7|     SM JAR|      1243.34|s-- ironic foxes ...|
|      364|puff almond blanc...|Manufacturer#4|Brand#43|LARGE BRUSHED NICKEL|    21|    MED PKG|      1264.36|           ach caref|
|      392|medium plum powde...|Manufacturer#4|Brand#42|MEDIUM BRUSHED CO...|    21|     SM BOX|      1292.39|        even request|
|      429|red firebrick smo...|Manufacturer#4|Brand#45|ECONOMY BRUSHED C...|    21|   MED DRUM|      1329.42|     carefully even |
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+
only showing top 20 rows
```

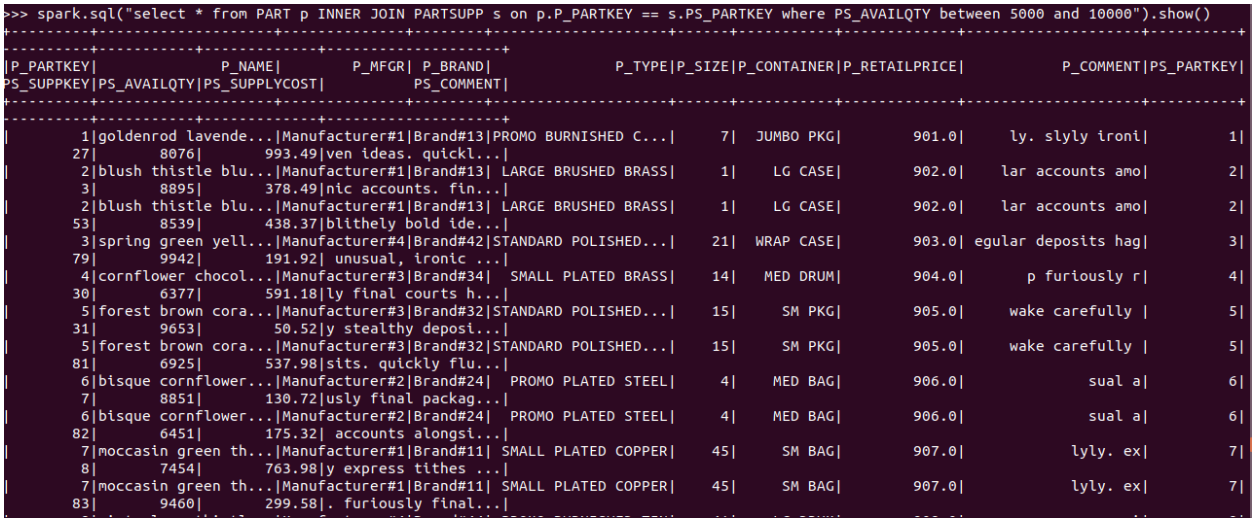
Sheet 1



Sum of P Size for each P Name. The view is filtered on sum of P Size and P Name. The sum of P Size filter ranges from 7 to 21. The P Name filter keeps 92 of 2,000 members.

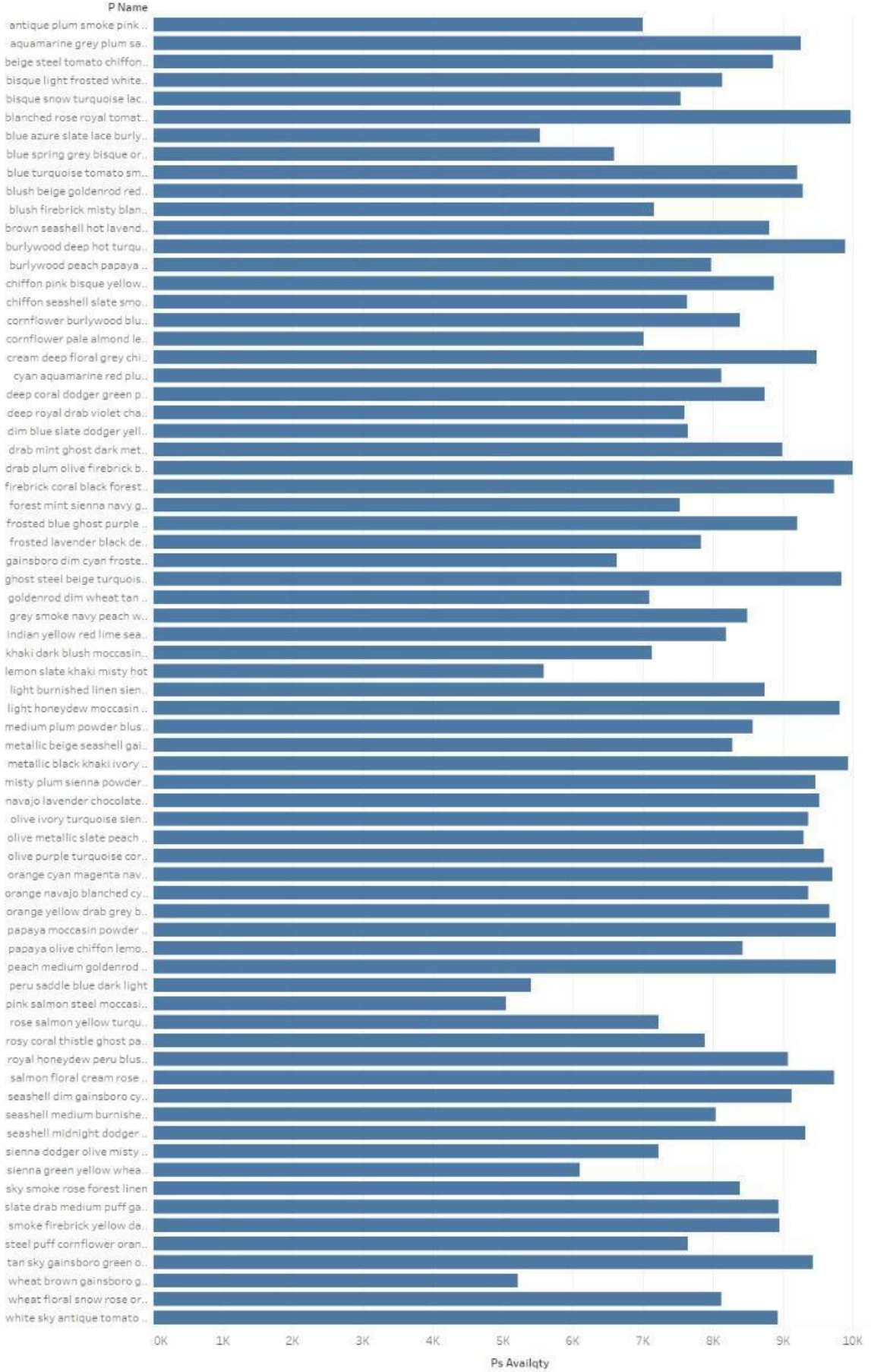
Q14. Get the records where the available qty is between 5000 and 10000.

>>> spark.sql("select * from PART p INNER JOIN PARTSUPP s on p.P_PARTKEY == s.PS_PARTKEY where PS_AVAILQTY between 5000 and 10000").show()



```
>>> spark.sql("select * from PART p INNER JOIN PARTSUPP s on p.P_PARTKEY == s.PS_PARTKEY where PS_AVAILQTY between 5000 and 10000").show()
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+----------+
---------+-----------+------------+--------------------+
|P_PARTKEY|              P_NAME|       P_MFGR| P_BRAND|              P_TYPE|P_SIZE|P_CONTAINER|P_RETAILPRICE|           P_COMMENT|PS_PARTKEY|
PS_SUPPKEY|PS_AVAILQTY|PS_SUPPLYCOST|          PS_COMMENT|
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+----------+
---------+-----------+------------+--------------------+
|        1|goldenrod lavende...|Manufacturer#1|Brand#13|PROMO BURNISHED C...|     7|  JUMBO PKG|        901.0|     ly. slyly ironi|         1|
       27|        8076|       993.49|ven ideas. quickl...|
|        2|blush thistle blu...|Manufacturer#1|Brand#13| LARGE BRUSHED BRASS|     1|    LG CASE|        902.0|    lar accounts amo|         2|
        3|        8895|       378.49|nic accounts. fin...|
|        2|blush thistle blu...|Manufacturer#1|Brand#13| LARGE BRUSHED BRASS|     1|    LG CASE|        902.0|    lar accounts amo|         2|
       53|        8539|       438.37|blithely bold ide...|
|        3|spring green yell...|Manufacturer#4|Brand#42|STANDARD POLISHED...|    21|  WRAP CASE|        903.0| egular deposits hag|         3|
       79|        9942|       191.92| unusual, ironic ...|
|        4|cornflower chocol...|Manufacturer#3|Brand#34|  SMALL PLATED BRASS|    14|   MED DRUM|        904.0|       p furiously r|         4|
       30|        6377|       591.18|ly final courts h...|
|        5|forest brown cora...|Manufacturer#3|Brand#32|STANDARD POLISHED...|    15|     SM PKG|        905.0|     wake carefully |         5|
       31|        9653|        50.52|y stealthy deposi...|
|        5|forest brown cora...|Manufacturer#3|Brand#32|STANDARD POLISHED...|    15|     SM PKG|        905.0|     wake carefully |         5|
       81|        6925|       537.98|sits. quickly flu...|
|        6|bisque cornflower...|Manufacturer#2|Brand#24|  PROMO PLATED STEEL|     4|    MED BAG|        906.0|              sual a|         6|
        7|        8851|       130.72|usly final packag...|
|        6|bisque cornflower...|Manufacturer#2|Brand#24|  PROMO PLATED STEEL|     4|    MED BAG|        906.0|              sual a|         6|
       82|        6451|       175.32| accounts alongsi...|
|        7|moccasin green th...|Manufacturer#1|Brand#11| SMALL PLATED COPPER|    45|     SM BAG|        907.0|            lyly. ex|         7|
        8|        7454|       763.98|y express tithes ...|
|        7|moccasin green th...|Manufacturer#1|Brand#11| SMALL PLATED COPPER|    45|     SM BAG|        907.0|            lyly. ex|         7|
       83|        9460|       299.58|. furiously final...|
```

Sheet 2



Sum of Ps Availqty for each P Name. The view is filtered on P Name, which keeps 71 of 2,000 members.

Q15. Get the records for the second highest supplycost.

>>> spark.sql("select max(PS_SUPPLYCOST) from PARTSUPP where PS_SUPPLYCOST < (select max(PS_SUPPLYCOST) from PARTSUPP)").show()

```
>>> spark.sql("select * from PART p INNER JOIN PARTSUPP s on p.P_PARTKEY == s.PS_PARTKEY where PS_AVAILQTY between 5000 and 10000").show()
+---------+--------------------+-------------+--------+--------------------+------+-----------+------------+--------------------+---------+
---------+-----------+------------+--------------------+
|P_PARTKEY|              P_NAME|       P_MFGR| P_BRAND|              P_TYPE|P_SIZE|P_CONTAINER|P_RETAILPRICE|           P_COMMENT|PS_PARTKEY|
PS_SUPPKEY|PS_AVAILQTY|PS_SUPPLYCOST|          PS_COMMENT|
+---------+--------------------+-------------+--------+--------------------+------+-----------+------------+--------------------+---------+
---------+-----------+------------+--------------------+
|        1|goldenrod lavende...|Manufacturer#1|Brand#13|PROMO BURNISHED C...|     7|  JUMBO PKG|       901.0|     ly. slyly ironi|         1|
       27|        8076|       993.49|ven ideas. quickl...|
|        2|blush thistle blu...|Manufacturer#1|Brand#13| LARGE BRUSHED BRASS|     1|    LG CASE|       902.0|    lar accounts amo|         2|
        3|        8895|       378.49|nic accounts. fin...|
|        2|blush thistle blu...|Manufacturer#1|Brand#13| LARGE BRUSHED BRASS|     1|    LG CASE|       902.0|    lar accounts amo|         2|
       53|        8539|       438.37|blithely bold ide...|
|        3|spring green yell...|Manufacturer#4|Brand#42|STANDARD POLISHED...|    21|  WRAP CASE|       903.0| egular deposits hag|         3|
       79|        9942|       191.92| unusual, ironic ...|
|        4|cornflower chocol...|Manufacturer#3|Brand#34|  SMALL PLATED BRASS|    14|   MED DRUM|       904.0|       p furiously r|         4|
       30|        6377|       591.18|ly final courts h...|
|        5|forest brown cora...|Manufacturer#3|Brand#32|STANDARD POLISHED...|    15|     SM PKG|       905.0|     wake carefully |         5|
       31|        9653|        50.52|y stealthy deposi...|
|        5|forest brown cora...|Manufacturer#3|Brand#32|STANDARD POLISHED...|    15|     SM PKG|       905.0|     wake carefully |         5|
       81|        6925|       537.98|sits. quickly flu...|
|        6|bisque cornflower...|Manufacturer#2|Brand#24|  PROMO PLATED STEEL|     4|    MED BAG|       906.0|              sual a|         6|
        7|        8851|       130.72|usly final packag...|
|        6|bisque cornflower...|Manufacturer#2|Brand#24|  PROMO PLATED STEEL|     4|    MED BAG|       906.0|              sual a|         6|
       82|        6451|       175.32| accounts alongsi...|
|        7|moccasin green th...|Manufacturer#1|Brand#11| SMALL PLATED COPPER|    45|     SM BAG|       907.0|            lyly. ex|         7|
        8|        7454|       763.98|y express tithes ...|
|        7|moccasin green th...|Manufacturer#1|Brand#11| SMALL PLATED COPPER|    45|     SM BAG|       907.0|            lyly. ex|         7|
       83|        9460|       299.58|. furiously final...|
```

Sheet 1

P Name	
frosted cornflower khaki ..	999.9

Maximum of Ps Supplycost broken down by P Name. The view is filtered on P Name, which keeps frosted cornflower khaki salmon metallic.

Q16. Get the records where the 4th manufacturer provides the LG BOX as a container with a supplycost between 500 and 1000.

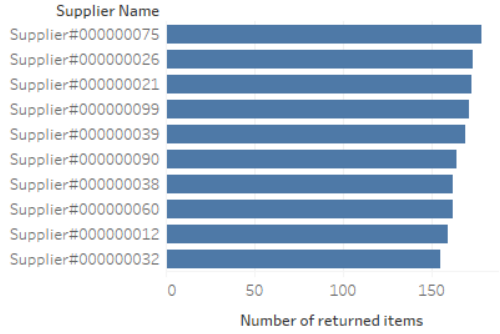
>>> spark.sql("select * from PART p INNER JOIN PARTSUPP s on p.P_PARTKEY == s.PS_PARTKEY where P_MFGR in ('Manufacturer#4') and P_CONTAINER = 'LG BOX' and PS_SUPPLYCOST between 500 and 1000").show()

```
>>> spark.sql("select * from PART p INNER JOIN PARTSUPP s on p.P_PARTKEY == s.PS_PARTKEY where P_MFGR in ('Manufacturer#4') and P_CONTAINER =
'LG BOX' and PS_SUPPLYCOST between 500 and 1000").show()
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+----------+-
---------+-----------+-------------+--------------------+
|P_PARTKEY|              P_NAME|       P_MFGR| P_BRAND|              P_TYPE|P_SIZE|P_CONTAINER|P_RETAILPRICE|           P_COMMENT|PS_PARTKEY|P
S_SUPPKEY|PS_AVAILQTY|PS_SUPPLYCOST|          PS_COMMENT|
+---------+--------------------+-------------+--------+--------------------+------+-----------+-------------+--------------------+----------+-
---------+-----------+-------------+--------------------+
|       17|indian navy coral...|Manufacturer#4|Brand#43|ECONOMY BRUSHED S...|    16|     LG BOX|       917.01|   regular accounts|        17|
      18|       8555|       995.35|are furiously fin...|
|       17|indian navy coral...|Manufacturer#4|Brand#43|ECONOMY BRUSHED S...|    16|     LG BOX|       917.01|   regular accounts|        17|
      43|       7737|       648.75|e blithely expres...|
|       17|indian navy coral...|Manufacturer#4|Brand#43|ECONOMY BRUSHED S...|    16|     LG BOX|       917.01|   regular accounts|        17|
      68|       3123|       555.04|ly bold accounts....|
|       30|cream misty steel...|Manufacturer#4|Brand#42|  PROMO ANODIZED TIN|    17|     LG BOX|       930.03|      carefully bus|        30|
      31|       4767|       989.05|ts. slyly final p...|
|       30|cream misty steel...|Manufacturer#4|Brand#42|  PROMO ANODIZED TIN|    17|     LG BOX|       930.03|      carefully bus|        30|
      56|        535|       743.26|sual instructions...|
|       30|cream misty steel...|Manufacturer#4|Brand#42|  PROMO ANODIZED TIN|    17|     LG BOX|       930.03|      carefully bus|        30|
      81|       7756|       568.86| special foxes ac...|
|       30|cream misty steel...|Manufacturer#4|Brand#42|  PROMO ANODIZED TIN|    17|     LG BOX|       930.03|      carefully bus|        30|
       6|       7945|       583.84| sleep. bold, reg...|
|      464|azure magenta lac...|Manufacturer#4|Brand#42|LARGE BURNISHED N...|    34|     LG BOX|      1364.46|l decoys. close exc|       464|
      65|       6084|       504.88|egular theodolite...|
|      612|midnight azure mo...|Manufacturer#4|Brand#42|  PROMO PLATED STEEL|    19|     LG BOX|      1512.61|        se quickly;|       612|
      75|       7469|       788.99| carefully specia...|
|      612|midnight azure mo...|Manufacturer#4|Brand#42|  PROMO PLATED STEEL|    19|     LG BOX|      1512.61|        se quickly;|       612|
       6|       9998|       888.14|refully final dep...|
|     1699|tan lemon goldenr...|Manufacturer#4|Brand#44|ECONOMY BRUSHED S...|    26|     LG BOX|      1600.69|            s along|      1699|
     100|       2775|       878.92|phins wake quickl...|
|     1699|tan lemon goldenr...|Manufacturer#4|Brand#44|ECONOMY BRUSHED S...|    26|     LG BOX|      1600.69|            s along|      1699|
      41|        518|       954.55|ly carefully pend...|
```

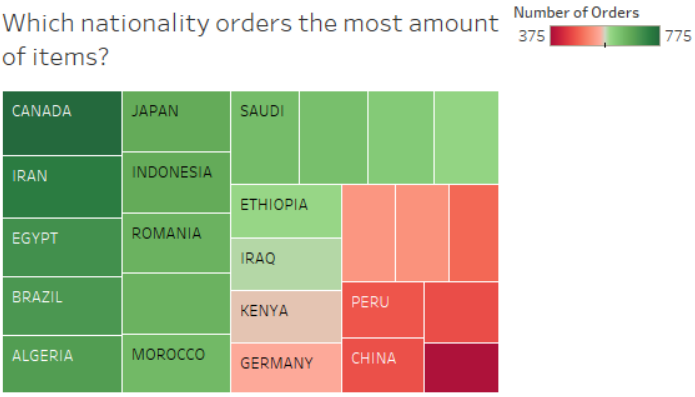
VISUALISATION

Retail

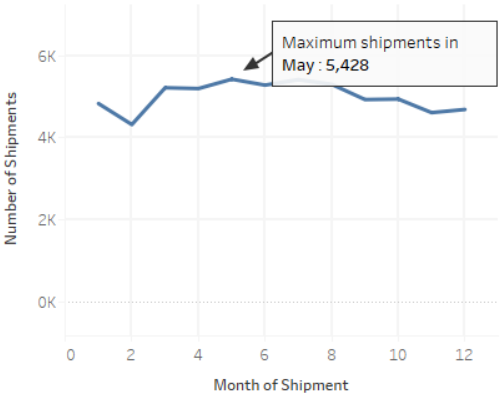
Which Supplier gets the most number of items returned? (Top 10)



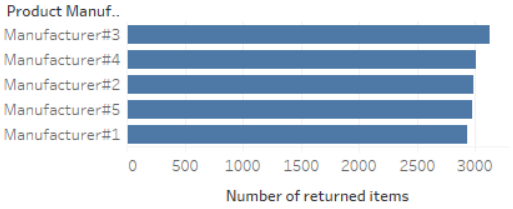
Which nationality orders the most amount of items?



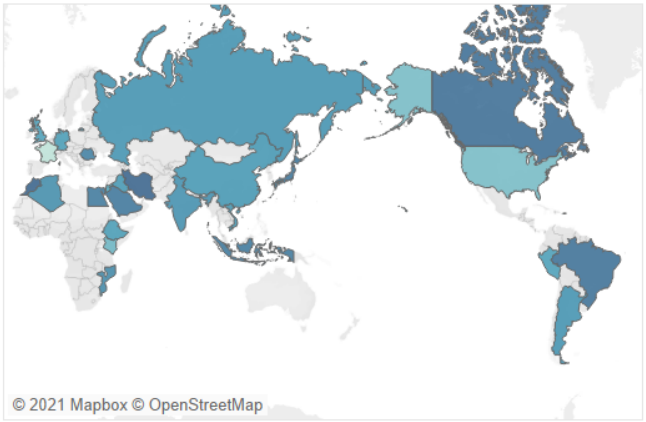
Amount of shipments done per month



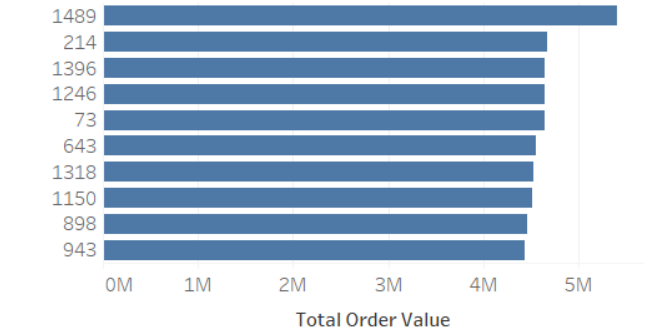
Which Manufacturer gets the most number of items returned?



Customer Distribution as per Nation



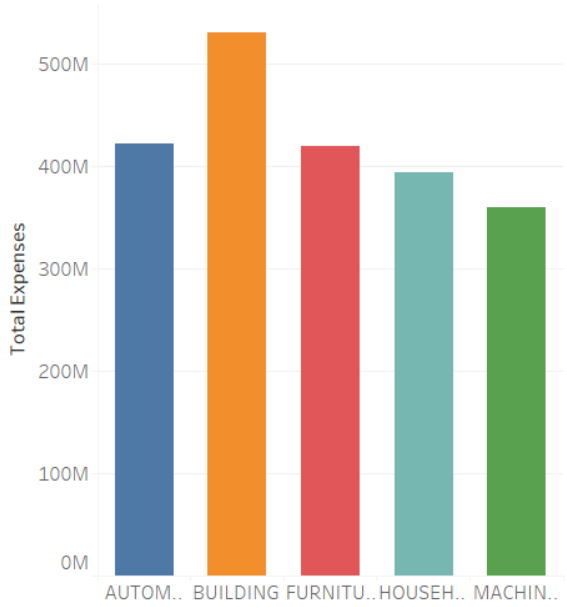
Top10 Customers by Total Order Value



Customer Count

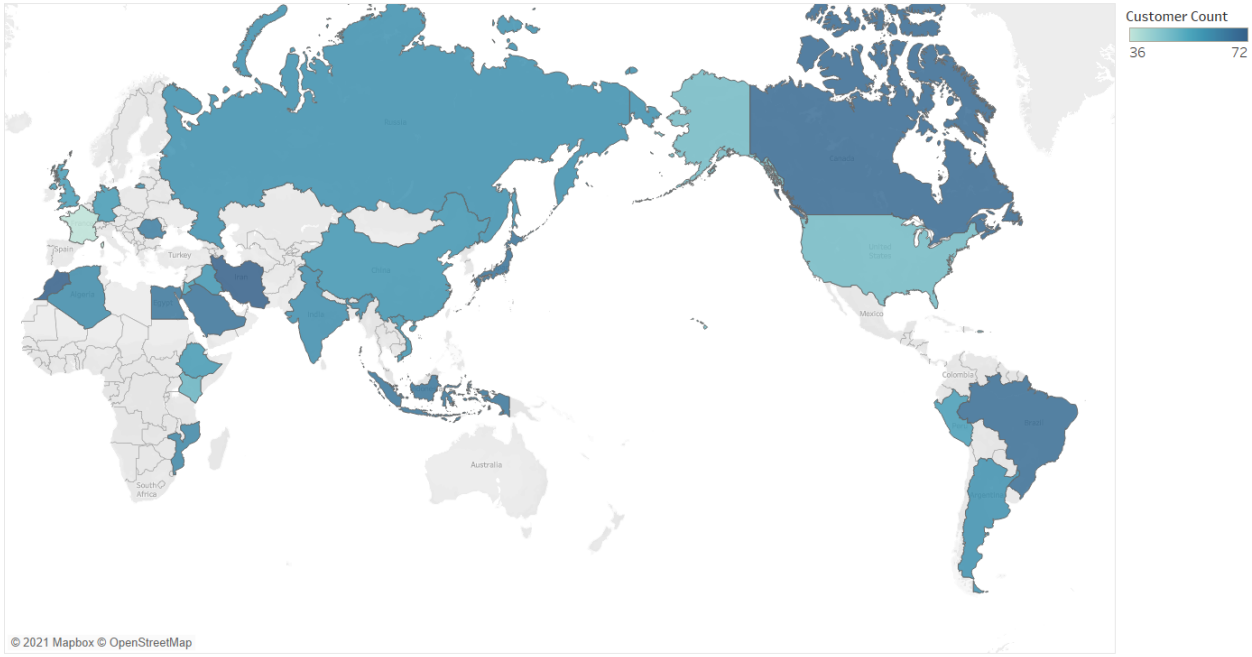
36 72

Market Segment and their Total Expenditures



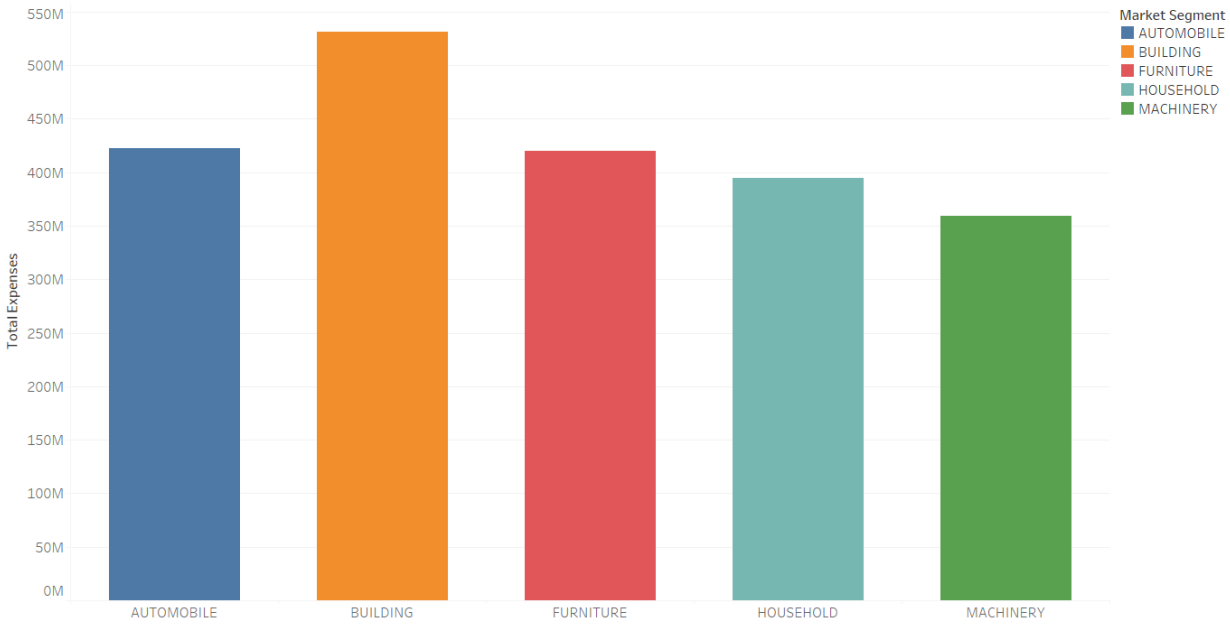
Market Segment
- AUTOMOBILE
- BUILDING
- FURNITURE
- HOUSEHOLD
- MACHINERY

Customer Distribution as per Nation



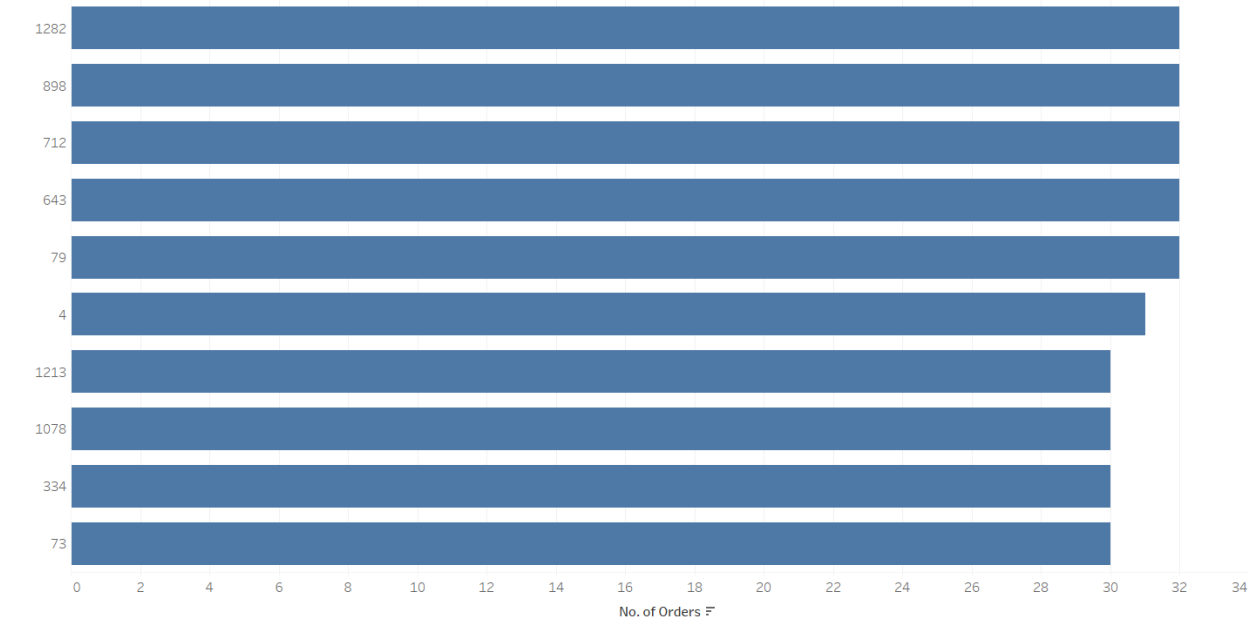
Map based on Longitude (generated) and Latitude (generated). Color shows distinct count of C_CUSTKEY. Details are shown for N_NAME.

Market Segment and their Total Expenditures



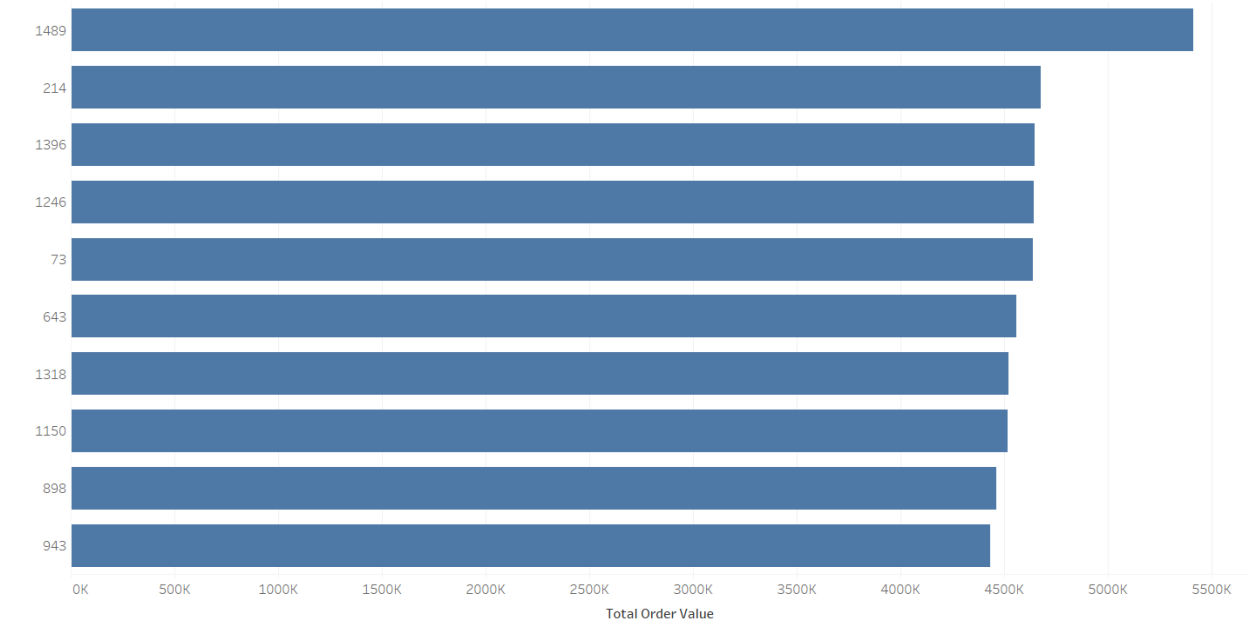
Sum of O_TOTALPRICE for each C_MKTSEGMENT. Color shows details about C_MKTSEGMENT.

Top 10 Customers with Most Orders



Distinct count of O_ORDERKEY for each C_CUSTKEY. The view is filtered on C_CUSTKEY, which keeps 10 of 1,500 members.

Top10 Customers by Total Order Value



Sum of O_TOTALPRICE for each C_CUSTKEY. The view is filtered on C_CUSTKEY, which keeps 10 of 1,500 members.