Uploaded the dataset to “HDFS” from FTP :

```
[therohitsaha08gmail@sl-cdp-prod-en10 ~]$ hdfs dfs -ls Retail_Business
Found 5 items
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-12 12:47 Retail_Business/categories
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-12 09:21 Retail_Business/customers-tab-delimited
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-12 14:00 Retail_Business/orders_parquet
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-14 16:34 Retail_Business/products_avro
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-15 10:24 Retail_Business/result
[therohitsaha08gmail@sl-cdp-prod-en10 ~]$
```

Task 2.2 --> Explore the customer records saved in the "customers-tab-delimited" directory on HDFS

```
>>> customer_df = spark.read.format("csv").option("delimiter", "\t").option("header", "false").option("inferSchema", "true").load("Retail_Business/customers-tab-delimited/part-m-00000").toDF("customer_id","customer_fname","customer_lname","customer_email" ,"customer_password" ,"customer_street" ,"customer_city" ,"customer_state" ,"customer_zipcode")
>>> customer_df.show()
+-----------+--------------+--------------+--------------+-----------------+--------------------+-------------+--------------+---------------+
|customer_id|customer_fname|customer_lname|customer_email|customer_password|     customer_street|customer_city|customer_state|customer_zipcode|
+-----------+--------------+--------------+--------------+-----------------+--------------------+-------------+--------------+---------------+
|          1|       Richard|     Hernandez|     XXXXXXXXX|        XXXXXXXXX|  6303 Heather Plaza|  Brownsville|            TX|          78521|
|          2|          Mary|       Barrett|     XXXXXXXXX|        XXXXXXXXX|9526 Noble Embers...|    Littleton|            CO|          80126|
|          3|           Ann|         Smith|     XXXXXXXXX|        XXXXXXXXX|3422 Blue Pioneer...|      Caguas|            PR|            725|
|          4|          Mary|         Jones|     XXXXXXXXX|        XXXXXXXXX|  8324 Little Common|   San Marcos|            CA|          92069|
|          5|        Robert|        Hudson|     XXXXXXXXX|        XXXXXXXXX|10 Crystal River ...|      Caguas|            PR|            725|
|          6|          Mary|         Smith|     XXXXXXXXX|        XXXXXXXXX|3151 Sleepy Quail...|     Passaic|            NJ|           7055|
|          7|       Melissa|         Wilcox|     XXXXXXXXX|        XXXXXXXXX|9453 High Concession|      Caguas|            PR|            725|
|          8|         Megan|         Smith|     XXXXXXXXX|        XXXXXXXXX|3047 Foggy Forest...|    Lawrence|            MA|           1841|
|          9|          Mary|         Perez|     XXXXXXXXX|        XXXXXXXXX| 3616 Quaking Street|      Caguas|            PR|            725|
|         10|       Melissa|         Smith|     XXXXXXXXX|        XXXXXXXXX|8598 Harvest Beac...|    Stafford|            VA|          22554|
|         11|          Mary|       Huffman|     XXXXXXXXX|        XXXXXXXXX|    3169 Stony Woods|      Caguas|            PR|            725|
|         12|   Christopher|         Smith|     XXXXXXXXX|        XXXXXXXXX|5594 Jagged Ember...| San Antonio|            TX|          78227|
|         13|          Mary|       Baldwin|     XXXXXXXXX|        XXXXXXXXX|7922 Iron Oak Gar...|      Caguas|            PR|            725|
|         14|     Katherine|         Smith|     XXXXXXXXX|        XXXXXXXXX|5666 Hazy Pony Sq...| Pico Rivera|            CA|          90660|
|         15|          Jane|          Luna|     XXXXXXXXX|        XXXXXXXXX|    673 Burning Glen|     Fontana|            CA|          92336|
|         16|       Tiffany|         Smith|     XXXXXXXXX|        XXXXXXXXX|      6651 Iron Port|      Caguas|            PR|            725|
|         17|          Mary|      Robinson|     XXXXXXXXX|        XXXXXXXXX|    1325 Noble Pike|       Taylor|            MI|          48180|
|         18|        Robert|         Smith|     XXXXXXXXX|        XXXXXXXXX|2734 Hazy Butterf...|     Martinez|            CA|          94553|
|         19|     Stephanie|      Mitchell|     XXXXXXXXX|        XXXXXXXXX|3543 Red Treasure...|      Caguas|            PR|            725|
|         20|          Mary|         Ellis|     XXXXXXXXX|        XXXXXXXXX|     4703 Old Route|West New York|            NJ|           7093|
+-----------+--------------+--------------+--------------+-----------------+--------------------+-------------+--------------+---------------+
only showing top 20 rows
```

2.2.1 Show the client information for those who live in California
2.2.2 Save the results in the result/scenario1/solution folder
2.2.3 Only records with the state value "CA" should be included in the result
2.2.4 Only the customer's entire name should be included in the output
Example: “Robert Hudson”

```
>>> from pyspark.sql.functions import col,concat,lit,from_unixtime,date_format
>>> df_california = customer_df.withColumn('customer_fullname',concat('customer_fname', lit(' '), 'customer_lname')).drop('customer_fname', 'customer_lname').filter(col("customer_state") == 'CA')
>>> df_california.show()
+-----------+--------------+-----------------+--------------------+-------------+--------------+---------------+-----------------+
|customer_id|customer_email|customer_password|     customer_street| customer_city|customer_state|customer_zipcode|customer_fullname|
+-----------+--------------+-----------------+--------------------+-------------+--------------+---------------+-----------------+
|          4|     XXXXXXXXX|        XXXXXXXXX|  8324 Little Common|    San Marcos|            CA|          92069|       Mary Jones|
|         14|     XXXXXXXXX|        XXXXXXXXX|5666 Hazy Pony Sq...|   Pico Rivera|            CA|          90660|  Katherine Smith|
|         15|     XXXXXXXXX|        XXXXXXXXX|    673 Burning Glen|      Fontana|            CA|          92336|        Jane Luna|
|         18|     XXXXXXXXX|        XXXXXXXXX|2734 Hazy Butterf...|     Martinez|            CA|          94553|     Robert Smith|
|         35|     XXXXXXXXX|        XXXXXXXXX|   9456 Sleepy Jetty|    Oceanside|            CA|          92056|  Margaret Wright|
|         40|     XXXXXXXXX|        XXXXXXXXX|   7358 Rocky Villas|    Long Beach|            CA|          90805|       Mary Smith|
|         44|     XXXXXXXXX|        XXXXXXXXX|     1356 Easy Plaza|         Napa|            CA|          94558|     Howard Smith|
|         50|     XXXXXXXXX|        XXXXXXXXX|938 Rustic Pine R...|San Bernardino|            CA|          92410|        Mary Kim|
|         59|     XXXXXXXXX|        XXXXXXXXX|     2306 Green Lane|     Sunnyvale|            CA|          94086|    Douglas James|
|         70|     XXXXXXXXX|        XXXXXXXXX| 5553 Cinder Harbour|  Los Angeles|            CA|          90042|     Mary Simmons|
|         72|     XXXXXXXXX|        XXXXXXXXX|5332 Heather Hill...|        Vista|            CA|          92084|  Frank Gillespie|
|         76|     XXXXXXXXX|        XXXXXXXXX|7605 Tawny Horse ...|  Los Angeles|            CA|          90016|     Joseph Young|
|         89|     XXXXXXXXX|        XXXXXXXXX|9126 Wishing Expr...|    Escondido|            CA|          92027|       Sean Smith|
|        106|     XXXXXXXXX|        XXXXXXXXX|     2783 Foggy Mews|         Napa|            CA|          94558|   Lauren Freeman|
|        114|     XXXXXXXXX|        XXXXXXXXX|4566 Burning Deer...|   Bellflower|            CA|          90706|     Alice Warner|
|        115|     XXXXXXXXX|        XXXXXXXXX|    1613 Broad Beach|  West Covina|            CA|          91790|       Mary Smith|
|        125|     XXXXXXXXX|        XXXXXXXXX|9831 Sunny Cloud ...|      Oxnard|            CA|          93030|   Mary Gallagher|
|        139|     XXXXXXXXX|        XXXXXXXXX|     9468 Red Corner|    San Diego|            CA|          92104|   Daniel Maxwell|
|        149|     XXXXXXXXX|        XXXXXXXXX|9729 Emerald Pony...|       Colton|            CA|          92324|  Shirley Mcclain|
|        156|     XXXXXXXXX|        XXXXXXXXX|2291 Thunder Leaf...|  Los Angeles|            CA|          90001|       Mary Smith|
+-----------+--------------+-----------------+--------------------+-------------+--------------+---------------+-----------------+
only showing top 20 rows

>>> df_california.write.format("csv").option("delimiter", "\t").option("header", "true").save("Retail_Business/result/scenario1/solution")
```

Task 2.3 --> Explore the order records saved in the “orders parquet” directory on HDFS

```
>>> order_df = spark.read.parquet(r'Retail_Business/orders_parquet/741ca897-c70e-4633-b352-5dc3414c5680.parquet')
>>> order_df.show()
+--------+-------------+-----------------+---------------+
|order_id|   order_date|order_customer_id|   order_status|
+--------+-------------+-----------------+---------------+
|       1|1374710400000|            11599|         CLOSED|
|       2|1374710400000|              256|PENDING_PAYMENT|
|       3|1374710400000|            12111|       COMPLETE|
|       4|1374710400000|             8827|         CLOSED|
|       5|1374710400000|            11318|       COMPLETE|
|       6|1374710400000|             7130|       COMPLETE|
|       7|1374710400000|             4530|       COMPLETE|
|       8|1374710400000|             2911|     PROCESSING|
|       9|1374710400000|             5657|PENDING_PAYMENT|
|      10|1374710400000|             5648|PENDING_PAYMENT|
|      11|1374710400000|              918| PAYMENT_REVIEW|
|      12|1374710400000|             1837|         CLOSED|
|      13|1374710400000|             9149|PENDING_PAYMENT|
|      14|1374710400000|             9842|     PROCESSING|
|      15|1374710400000|             2568|       COMPLETE|
|      16|1374710400000|             7276|PENDING_PAYMENT|
|      17|1374710400000|             2667|       COMPLETE|
|      18|1374710400000|             1205|         CLOSED|
|      19|1374710400000|             9488|PENDING_PAYMENT|
|      20|1374710400000|             9198|     PROCESSING|
+--------+-------------+-----------------+---------------+
only showing top 20 rows
```

2.3.1 Show all orders with the order status value "COMPLETE"
2.3.2 The output should be in JSON format
2.3.3 Save the data in the "result/scenario2/solution" directory on HDFS
2.3.4 The "order date" column should be in the "YYYY-MM-DD" format
2.3.5 Use GZIP compression to compress the output
2.3.6 Only the column names listed below should be included in the output:
2.3.6.1 Order number
2.3.6.2 Order date
2.3.6.3 Current situation

convert epoch time(in ms) to date format :

```
>>> orderdate_df = order_df.withColumn("order_date", from_unixtime(col("order_date")/1000, 'yyyy-MM-dd'))
>>> orderdate_df.show()
+--------+----------+-----------------+---------------+
|order_id|order_date|order_customer_id|   order_status|
+--------+----------+-----------------+---------------+
|       1|2013-07-25|            11599|         CLOSED|
|       2|2013-07-25|              256|PENDING_PAYMENT|
|       3|2013-07-25|            12111|       COMPLETE|
|       4|2013-07-25|             8827|         CLOSED|
|       5|2013-07-25|            11318|       COMPLETE|
|       6|2013-07-25|             7130|       COMPLETE|
|       7|2013-07-25|             4530|       COMPLETE|
|       8|2013-07-25|             2911|     PROCESSING|
|       9|2013-07-25|             5657|PENDING_PAYMENT|
|      10|2013-07-25|             5648|PENDING_PAYMENT|
|      11|2013-07-25|              918| PAYMENT_REVIEW|
|      12|2013-07-25|             1837|         CLOSED|
|      13|2013-07-25|             9149|PENDING_PAYMENT|
|      14|2013-07-25|             9842|     PROCESSING|
|      15|2013-07-25|             2568|       COMPLETE|
|      16|2013-07-25|             7276|PENDING_PAYMENT|
|      17|2013-07-25|             2667|       COMPLETE|
|      18|2013-07-25|             1205|         CLOSED|
|      19|2013-07-25|             9488|PENDING_PAYMENT|
|      20|2013-07-25|             9198|     PROCESSING|
+--------+----------+-----------------+---------------+
only showing top 20 rows
```

```
>>> complete_order_df = orderdate_df.filter(col("order_status") == 'COMPLETE').select("order_id","order_date","order_status")
>>> complete_order_df.show()
+--------+----------+------------+
|order_id|order_date|order_status|
+--------+----------+------------+
|       3|2013-07-25|    COMPLETE|
|       5|2013-07-25|    COMPLETE|
|       6|2013-07-25|    COMPLETE|
|       7|2013-07-25|    COMPLETE|
|      15|2013-07-25|    COMPLETE|
|      17|2013-07-25|    COMPLETE|
|      22|2013-07-25|    COMPLETE|
|      26|2013-07-25|    COMPLETE|
|      28|2013-07-25|    COMPLETE|
|      32|2013-07-25|    COMPLETE|
|      35|2013-07-25|    COMPLETE|
|      45|2013-07-25|    COMPLETE|
|      56|2013-07-25|    COMPLETE|
|      63|2013-07-25|    COMPLETE|
|      65|2013-07-25|    COMPLETE|
|      67|2013-07-25|    COMPLETE|
|      71|2013-07-25|    COMPLETE|
|      72|2013-07-25|    COMPLETE|
|      76|2013-07-25|    COMPLETE|
|      80|2013-07-25|    COMPLETE|
+--------+----------+------------+
only showing top 20 rows

>>> complete_order_df.write.format("json").option("compression", "gzip").option("header", "true").save("Retail_Business/result/scenario2/solution")
```

Task 2.4 --> Explore the customer records saved in the "customers-tab-delimited" directory on HDFS

2.4.1 Produce a list of all consumers who live in the city of "Caguas"

2.4.2 Save the data in the result/scenario3/solution directory on HDFS

2.4.3 The result should only contain records with the value "Caguas" for the customer city

2.4.4 Use snappy compression to compress the output

2.4.5 Save the file in the orc format

```
>>> df_Caguas = customer_df.filter(col("customer_city") == 'Caguas')
>>> df_Caguas.show()
+-----------+-------------+-------------+-------------+-----------------+--------------------+-------------+--------------+----------------+
|customer_id|customer_fname|customer_lname|customer_email|customer_password|     customer_street|customer_city|customer_state|customer_zipcode|
+-----------+-------------+-------------+-------------+-----------------+--------------------+-------------+--------------+----------------+
|          3|          Ann|        Smith|    XXXXXXXXX|       XXXXXXXXX|3422 Blue Pioneer...|       Caguas|            PR|             725|
|          5|       Robert|       Hudson|    XXXXXXXXX|       XXXXXXXXX|10 Crystal River ...|       Caguas|            PR|             725|
|          7|      Melissa|       Wilcox|    XXXXXXXXX|       XXXXXXXXX|9453 High Concession|       Caguas|            PR|             725|
|          9|         Mary|        Perez|    XXXXXXXXX|       XXXXXXXXX| 3616 Quaking Street|       Caguas|            PR|             725|
|         11|         Mary|      Huffman|    XXXXXXXXX|       XXXXXXXXX|    3169 Stony Woods|       Caguas|            PR|             725|
|         13|         Mary|      Baldwin|    XXXXXXXXX|       XXXXXXXXX|7922 Iron Oak Gar...|       Caguas|            PR|             725|
|         16|      Tiffany|        Smith|    XXXXXXXXX|       XXXXXXXXX|      6651 Iron Port|       Caguas|            PR|             725|
|         19|    Stephanie|     Mitchell|    XXXXXXXXX|       XXXXXXXXX|3543 Red Treasure...|       Caguas|            PR|             725|
|         21|      William|    Zimmerman|    XXXXXXXXX|       XXXXXXXXX|3323 Old Willow M...|       Caguas|            PR|             725|
|         24|         Mary|        Smith|    XXXXXXXXX|       XXXXXXXXX| 9417 Emerald Towers|       Caguas|            PR|             725|
|         27|         Mary|      Vincent|    XXXXXXXXX|       XXXXXXXXX|1768 Sleepy Zephy...|       Caguas|            PR|             725|
|         30|      Barbara|        Smith|    XXXXXXXXX|       XXXXXXXXX|   2455 Merry Hollow|       Caguas|            PR|             725|
|         32|        Alice|        Smith|    XXXXXXXXX|       XXXXXXXXX|   2082 Hidden Green|       Caguas|            PR|             725|
|         34|         Mary|        Smith|    XXXXXXXXX|       XXXXXXXXX|3330 Easy Berry R...|       Caguas|            PR|             725|
|         36|     Michelle|        Carey|    XXXXXXXXX|       XXXXXXXXX| 6336 Fallen Village|       Caguas|            PR|             725|
|         39|         Juan|     Mckinney|    XXXXXXXXX|       XXXXXXXXX|7274 Blue Wagon  ...|       Caguas|            PR|             725|
|         43|         Mary|      Herring|    XXXXXXXXX|       XXXXXXXXX|   4575 Thunder Dale|       Caguas|            PR|             725|
|         47|         Lori|       Fuller|    XXXXXXXXX|       XXXXXXXXX|      357 Noble Lane|       Caguas|            PR|             725|
|         49|       Martha|        Smith|    XXXXXXXXX|       XXXXXXXXX|    7449 Merry Chase|       Caguas|            PR|             725|
|         51|      Jessica|        Smith|    XXXXXXXXX|       XXXXXXXXX|8344 Dewy Fawn Farms|       Caguas|            PR|             725|
+-----------+-------------+-------------+-------------+-----------------+--------------------+-------------+--------------+----------------+
only showing top 20 rows

>>> df_Caguas.write.format("orc").option("compression", "snappy").option("header", "true").save("Retail_Business/result/scenario3/solution")
```

Task 2.5 --> Explore all the category records stored in the “categories” directory on HDFS

2.5.1 Save the result files in CSV format

2.5.2 Save the data in the result/scenario4/solution directory on HDFS

2.5.3 Use lz4 compression to compress the output

```
>>> category_df = spark.read.format("csv").option("delimiter", ",").option("header", "false").option("inferSchema", "true").load("Retail_Business/categories/part-m-00000").toDF("category_id","category_department_id","category_name
")
>>> category_df.show()
+-----------+----------------------+------------------+
|category_id|category_department_id|     category_name|
+-----------+----------------------+------------------+
|          1|                     2|          Football|
|          2|                     2|            Soccer|
|          3|                     2|Baseball & Softball|
|          4|                     2|        Basketball|
|          5|                     2|         Lacrosse|
|          6|                     2|  Tennis & Racquet|
|          7|                     2|          Hockey|
|          8|                     2|       More Sports|
|          9|                     3|  Cardio Equipment|
|         10|                     3| Strength Training|
|         11|                     3|Fitness Accessories|
|         12|                     3|      Boxing & MMA|
|         13|                     3|       Electronics|
|         14|                     3|    Yoga & Pilates|
|         15|                     3| Training by Sport|
|         16|                     3|   As Seen on  TV!|
|         17|                     4|            Cleats|
|         18|                     4|    Men's Footwear|
|         19|                     4|  Women's Footwear|
|         20|                     4|    Kids' Footwear|
+-----------+----------------------+------------------+
only showing top 20 rows

>>> category_df.write.format("csv").option("compression", "lz4").option("header", "true").save("Retail_Business/result/scenario4/solution")
```

Task 2.6 --> Explore all product records that are saved in the “products_avro” database

```
>>> products_df = spark.read.format("avro").load("Retail_Business/products_avro")
>>> products_df.show()
+----------+--------------------+--------------------+-------------------+-------------+--------------------+
|product_id|product_category_id|        product_name|product_description|product_price|       product_image|
+----------+--------------------+--------------------+-------------------+-------------+--------------------+
|      1009|                  45|Diamond Fear No E...|                   |       599.99|http://images.acm...|
|      1010|                  46|DBX Vector Series...|                   |        19.98|http://images.acm...|
|      1011|                  46|Old Town Canoe Sa...|                   |       499.99|http://images.acm...|
|      1012|                  46|Pelican Trailblaz...|                   |       299.99|http://images.acm...|
|      1013|                  46|Perception Sport ...|                   |       349.99|http://images.acm...|
|      1014|                  46|O'Brien Men's Neo...|                   |        49.98|http://images.acm...|
|      1015|                  46|GoPro HERO3+ Blac...|                   |       399.99|http://images.acm...|
|      1016|                  46|Field & Stream 12...|                   |       549.99|http://images.acm...|
|      1017|                  46|DBX Vector Series...|                   |        19.98|http://images.acm...|
|      1018|                  46|Coleman Scanoe Canoe|                   |       399.99|http://images.acm...|
|      1019|                  46|O'Brien Youth Neo...|                   |        44.98|http://images.acm...|
|      1020|                  46|Field & Stream Ea...|                   |       549.99|http://images.acm...|
|      1021|                  46|Old Town Trip 10 ...|                   |       499.99|http://images.acm...|
|      1022|                  46|O'Brien Women's N...|                   |        49.98|http://images.acm...|
|      1023|                  46|Quest Pioneer Adj...|                   |        29.99|http://images.acm...|
|      1024|                  46|Columbia Women's ...|                   |        21.99|http://images.acm...|
|      1025|                  46|Future Beach Trop...|                   |       369.99|http://images.acm...|
|      1026|                  46|Quantum Smoke PT ...|                   |       119.99|http://images.acm...|
|      1027|                  46|DBX Vector Series...|                   |        34.99|http://images.acm...|
|      1028|                  46|YETI Tundra 65 Ch...|                   |       399.99|http://images.acm...|
+----------+--------------------+--------------------+-------------------+-------------+--------------------+
only showing top 20 rows
```

2.6.1 Only products with a price of more than 1000.0 should be included in the output

2.6.2 Save the output files in parquet format

2.6.3 Remove data from the table if the product price is lesser than 1000.0

2.6.4 Save the data in the result/scenario5/solution directory on HDFS

2.6.5 Use snappy compression to compress the output

```
>>> df_prod1000 = products_df.filter(col("product_price") > 1000)
>>> df_prod1000.show()
+----------+------------------+--------------------+-------------------+-------------+--------------------+
|product_id|product_category_id|        product_name|product_description|product_price|       product_image|
+----------+------------------+--------------------+-------------------+-------------+--------------------+
|      1048|                47|Spalding Beast 60...|                   |      1099.99|http://images.acm...|
|       496|                22|  SOLE F85 Treadmill|                   |      1799.99|http://images.acm...|
|        66|                 4|  SOLE F85 Treadmill|                   |      1799.99|http://images.acm...|
|       199|                10|  SOLE F85 Treadmill|                   |      1799.99|http://images.acm...|
|       208|                10| SOLE E35 Elliptical|                   |      1999.99|http://images.acm...|
+----------+------------------+--------------------+-------------------+-------------+--------------------+

>>> df_prod1000.write.parquet("Retail_Business/result/scenario5/solution", compression="snappy")
```

Task 2.7 --> Explore the “products_avro” stored in product records
2.7.1 Only products with a price of more than 1000.0 should be in the output
2.7.2 The pattern "Treadmill" appears in the product name
2.7.3 Save the output files in parquet format
2.7.4Save the data in the result/scenario6/solution directory on HDFS
2.7.5 Use GZIP compression to compress the output

```
>>> df_prod = products_df.filter(col("product_price") > 1000).filter(col("product_name").like("%Treadmill%"))
>>> df_prod.show()
+----------+------------------+------------------+-------------------+-------------+--------------------+
|product_id|product_category_id|      product_name|product_description|product_price|       product_image|
+----------+------------------+------------------+-------------------+-------------+--------------------+
|       496|                22|SOLE F85 Treadmill|                   |      1799.99|http://images.acm...|
|        66|                 4|SOLE F85 Treadmill|                   |      1799.99|http://images.acm...|
|       199|                10|SOLE F85 Treadmill|                   |      1799.99|http://images.acm...|
+----------+------------------+------------------+-------------------+-------------+--------------------+

>>> df_prod.write.parquet("Retail_Business/result/scenario6/solution", compression="gzip")
```

Task 2.8 --> Explore the order records that are saved in the “orders parquet” table on HDFS
2.8.1 Output all PENDING orders in July 2013
2.8.2 Output files should be in JSON format
2.8.3 Save the data in the result/scenario7/solution directory on HDFS.
2.8.4 Only entries with the order status value of "PENDING" should be included in the result
2.8.5 Order date should be in the YYYY-MM-DD format
2.8.6 Use snappy compression to compress the output, which should just contain the order date and order status

```
>>> final_df = orderdate_df.filter(col("order_date").like("2013-07%")).filter(col("order_status") == 'PENDING').select("order_date", "order_status").orderBy("order_date")
>>> final_df.show()
+----------+------------+
|order_date|order_status|
+----------+------------+
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-25|     PENDING|
|2013-07-26|     PENDING|
|2013-07-26|     PENDING|
|2013-07-26|     PENDING|
|2013-07-26|     PENDING|
|2013-07-26|     PENDING|
|2013-07-26|     PENDING|
|2013-07-26|     PENDING|
+----------+------------+
only showing top 20 rows

>>> final_df.write.format("json").option("compression", "snappy").option("header", "true").save("Retail_Business/result/scenario7/solution")
```

All Results in HDFS Directory:

```
[therohitsaha08gmail@sl-cdp-prod-en10 ~]$ hdfs dfs -ls Retail_Business/result
Found 7 items
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-14 12:44 Retail_Business/result/scenario1
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-14 12:56 Retail_Business/result/scenario2
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-14 15:48 Retail_Business/result/scenario3
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-14 16:23 Retail_Business/result/scenario4
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-14 17:32 Retail_Business/result/scenario5
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-14 17:15 Retail_Business/result/scenario6
drwxrwxr-x   - therohitsaha08gmail hadoop          0 2024-02-15 10:24 Retail_Business/result/scenario7
```