Output

Task 1

```
TASK 1: Retrieve all records for particular customer = Donna Smith
Using Dataframe
+-----------+--------------+--------------+--------------+----------------+--------------+--------------+--------------+----------------+
|customer_id|customer_fname|customer_lname|customer_email|customer_password|customer_street| customer_city|customer_state|customer_zipcode|
+-----------+--------------+--------------+--------------+----------------+--------------+--------------+--------------+----------------+
|       6896|         Donna|         Smith|       *******|         *******|       *******|     Santa Ana|            CA|         *******|
|       7087|         Donna|         Smith|       *******|         *******|       *******|      Brooklyn|            NY|         *******|
|       7541|         Donna|         Smith|       *******|         *******|       *******| Pompano Beach|            FL|         *******|
|       8328|         Donna|         Smith|       *******|         *******|       *******|   Crown Point|            IN|         *******|
|       8826|         Donna|         Smith|       *******|         *******|       *******|    Greensboro|            NC|         *******|
|       8905|         Donna|         Smith|       *******|         *******|       *******|      Augusta|            GA|         *******|
|        567|         Donna|         Smith|       *******|         *******|       *******|     Las Vegas|            NV|         *******|
|        759|         Donna|         Smith|       *******|         *******|       *******|        Lawton|            OK|         *******|
|        796|         Donna|         Smith|       *******|         *******|       *******|   Tallahassee|            FL|         *******|
|       1896|         Donna|         Smith|       *******|         *******|       *******|       Chicago|            IL|         *******|
|       2068|         Donna|         Smith|       *******|         *******|       *******|       Atlanta|            GA|         *******|
|       2305|         Donna|         Smith|       *******|         *******|       *******|      Temecula|            CA|         *******|
|       2425|         Donna|         Smith|       *******|         *******|       *******|   Los Angeles|            CA|         *******|
|       3176|         Donna|         Smith|       *******|         *******|       *******|        Caguas|            PR|         *******|
|       4441|         Donna|         Smith|       *******|         *******|       *******|Rancho Cordova|            CA|         *******|
|       5938|         Donna|         Smith|       *******|         *******|       *******|        Caguas|            PR|         *******|
|       6191|         Donna|         Smith|       *******|         *******|       *******|    Saint Paul|            MN|         *******|
|       9327|         Donna|         Smith|       *******|         *******|       *******|        Caguas|            PR|         *******|
|       9331|         Donna|         Smith|       *******|         *******|       *******|     Clementon|            NJ|         *******|
|       9455|         Donna|         Smith|       *******|         *******|       *******|      El Monte|            CA|         *******|
+-----------+--------------+--------------+--------------+----------------+--------------+--------------+--------------+----------------+
only showing top 20 rows

Number of rows in result1DF = 24
Using SparkSQL
+-----------+--------------+--------------+--------------+----------------+--------------+--------------+--------------+----------------+
|customer_id|customer_fname|customer_lname|customer_email|customer_password|customer_street| customer_city|customer_state|customer_zipcode|
+-----------+--------------+--------------+--------------+----------------+--------------+--------------+--------------+----------------+
|       6896|         Donna|         Smith|       *******|         *******|       *******|     Santa Ana|            CA|         *******|
|       7087|         Donna|         Smith|       *******|         *******|       *******|      Brooklyn|            NY|         *******|
|       7541|         Donna|         Smith|       *******|         *******|       *******| Pompano Beach|            FL|         *******|
|       8328|         Donna|         Smith|       *******|         *******|       *******|   Crown Point|            IN|         *******|
|       8826|         Donna|         Smith|       *******|         *******|       *******|    Greensboro|            NC|         *******|
|       8905|         Donna|         Smith|       *******|         *******|       *******|      Augusta|            GA|         *******|
|        567|         Donna|         Smith|       *******|         *******|       *******|     Las Vegas|            NV|         *******|
|        759|         Donna|         Smith|       *******|         *******|       *******|        Lawton|            OK|         *******|
|        796|         Donna|         Smith|       *******|         *******|       *******|   Tallahassee|            FL|         *******|
|       1896|         Donna|         Smith|       *******|         *******|       *******|       Chicago|            IL|         *******|
|       2068|         Donna|         Smith|       *******|         *******|       *******|       Atlanta|            GA|         *******|
|       2305|         Donna|         Smith|       *******|         *******|       *******|      Temecula|            CA|         *******|
|       2425|         Donna|         Smith|       *******|         *******|       *******|   Los Angeles|            CA|         *******|
|       3176|         Donna|         Smith|       *******|         *******|       *******|        Caguas|            PR|         *******|
|       4441|         Donna|         Smith|       *******|         *******|       *******|Rancho Cordova|            CA|         *******|
|       5938|         Donna|         Smith|       *******|         *******|       *******|        Caguas|            PR|         *******|
|       6191|         Donna|         Smith|       *******|         *******|       *******|    Saint Paul|            MN|         *******|
|       9327|         Donna|         Smith|       *******|         *******|       *******|        Caguas|            PR|         *******|
|       9331|         Donna|         Smith|       *******|         *******|       *******|     Clementon|            NJ|         *******|
|       9455|         Donna|         Smith|       *******|         *******|       *******|      El Monte|            CA|         *******|
+-----------+--------------+--------------+--------------+----------------+--------------+--------------+--------------+----------------+
only showing top 20 rows

Number of rows in result1DF_SQL = 24
```

Task 2

```
TASK 2: List count of orders based on status and month
Using Dataframe
+---------------+-----------+-----+
|   order_status|order_month|count|
+---------------+-----------+-----+
|       CANCELED|          7|  127|
|       COMPLETE|          7| 1934|
|     PROCESSING|         11|  714|
|       COMPLETE|          5| 1854|
|         CLOSED|          8|  637|
|     PROCESSING|          9|  676|
|     PROCESSING|          2|  700|
| PAYMENT_REVIEW|          2|   57|
|       COMPLETE|          3| 1967|
|       CANCELED|          3|  117|
|     PROCESSING|          4|  719|
|PENDING_PAYMENT|         10| 1172|
|        PENDING|          3|  605|
|     PROCESSING|          8|  692|
|        ON_HOLD|          3|  323|
|       COMPLETE|          9| 1933|
|         CLOSED|         11|  686|
|       COMPLETE|         12| 1898|
|     PROCESSING|         12|  685|
| PAYMENT_REVIEW|          4|   52|
+---------------+-----------+-----+
only showing top 20 rows

Number of rows in result2DF = 108
Using SparkSQL
+---------------+-----------+-----+
|   order_status|order_month|count|
+---------------+-----------+-----+
|       CANCELED|          7|  127|
|       COMPLETE|          7| 1934|
|     PROCESSING|         11|  714|
|       COMPLETE|          5| 1854|
|         CLOSED|          8|  637|
|     PROCESSING|          9|  676|
|     PROCESSING|          2|  700|
| PAYMENT_REVIEW|          2|   57|
|       COMPLETE|          3| 1967|
|       CANCELED|          3|  117|
|     PROCESSING|          4|  719|
|PENDING_PAYMENT|         10| 1172|
|        PENDING|          3|  605|
|     PROCESSING|          8|  692|
|        ON_HOLD|          3|  323|
|       COMPLETE|          9| 1933|
|         CLOSED|         11|  686|
|       COMPLETE|         12| 1898|
|     PROCESSING|         12|  685|
| PAYMENT_REVIEW|          4|   52|
+---------------+-----------+-----+
only showing top 20 rows

Number of rows in result2DF_SQL = 108
```

Task 3

```
TASK 3: List count of orders based on status and month for particular customer = Donna Smith
Using Dataframe
+---------------+-----------+-----+
|   order_status|order_month|count|
+---------------+-----------+-----+
|       CANCELED|          7|    1|
|       COMPLETE|          7|    3|
|       COMPLETE|          5|    2|
|     PROCESSING|          2|    3|
|       COMPLETE|          3|    2|
|     PROCESSING|          4|    1|
|PENDING_PAYMENT|         10|    3|
|        PENDING|          3|    3|
|     PROCESSING|          8|    1|
|       COMPLETE|          9|    5|
|       COMPLETE|         12|    4|
|         CLOSED|         11|    3|
|       CANCELED|          6|    1|
|        PENDING|          1|    1|
|PENDING_PAYMENT|         12|    4|
|PENDING_PAYMENT|          6|    2|
|       COMPLETE|          2|    3|
|        PENDING|          2|    3|
|        PENDING|         10|    2|
|       COMPLETE|         10|    3|
+---------------+-----------+-----+
only showing top 20 rows

Number of rows in result3DF = 53
Using SparkSQL
+---------------+-----------+-----+
|   order_status|order_month|count|
+---------------+-----------+-----+
|       CANCELED|          7|    1|
|       COMPLETE|          7|    3|
|       COMPLETE|          5|    2|
|     PROCESSING|          2|    3|
|       COMPLETE|          3|    2|
|     PROCESSING|          4|    1|
|PENDING_PAYMENT|         10|    3|
|        PENDING|          3|    3|
|     PROCESSING|          8|    1|
|       COMPLETE|          9|    5|
|       COMPLETE|         12|    4|
|         CLOSED|         11|    3|
|       CANCELED|          6|    1|
|        PENDING|          1|    1|
|PENDING_PAYMENT|         12|    4|
|PENDING_PAYMENT|          6|    2|
|       COMPLETE|          2|    3|
|        PENDING|          2|    3|
|        PENDING|         10|    2|
|       COMPLETE|         10|    3|
+---------------+-----------+-----+
only showing top 20 rows

Number of rows in result3DF_SQL = 53
```

Task 4

```
TASK 4. List count of orders based on customer and status
Using Dataframe
+-----------+---------------+-----+
|customer_id|   order_status|count|
+-----------+---------------+-----+
|       5204|       COMPLETE|    5|
|      10796|       COMPLETE|    4|
|       8161|       COMPLETE|    1|
|       8151|       COMPLETE|    1|
|       8206|     PROCESSING|    2|
|       7553|PENDING_PAYMENT|    2|
|        197|        PENDING|    2|
|       2334|        ON_HOLD|    1|
|      10114|PENDING_PAYMENT|    3|
|       6151|         CLOSED|    1|
|       4193|     PROCESSING|    2|
|      12268|PENDING_PAYMENT|    1|
|       8971|     PROCESSING|    2|
|       8376|     PROCESSING|    2|
|       4386|       COMPLETE|    3|
|       1385|        PENDING|    1|
|       4952|        PENDING|    1|
|      10660|       COMPLETE|    1|
|       9626|         CLOSED|    1|
|      11118|        ON_HOLD|    1|
+-----------+---------------+-----+
only showing top 20 rows

Number of rows in result4DF = 43460
Using SparkSQL
+-----------+---------------+-----+
|customer_id|   order_status|count|
+-----------+---------------+-----+
|       5204|       COMPLETE|    5|
|      10796|       COMPLETE|    4|
|       8161|       COMPLETE|    1|
|       8151|       COMPLETE|    1|
|       8206|     PROCESSING|    2|
|       7553|PENDING_PAYMENT|    2|
|        197|        PENDING|    2|
|       2334|        ON_HOLD|    1|
|      10114|PENDING_PAYMENT|    3|
|       6151|         CLOSED|    1|
|       4193|     PROCESSING|    2|
|      12268|PENDING_PAYMENT|    1|
|       8971|     PROCESSING|    2|
|       8376|     PROCESSING|    2|
|       4386|       COMPLETE|    3|
|       1385|        PENDING|    1|
|       4952|        PENDING|    1|
|      10660|       COMPLETE|    1|
|       9626|         CLOSED|    1|
|      11118|        ON_HOLD|    1|
+-----------+---------------+-----+
only showing top 20 rows

Number of rows in result4DF_SQL = 43460
```

Task 5

```
TASK 5. Find the customers who have placed orders
Using Dataframe
+-----------+--------------+--------------+
|customer_id|customer_fname|customer_lname|
+-----------+--------------+--------------+
|       5016|         James|        Rangel|
|      11404|         Helen|          Cook|
|       8645|        Andrew|      Johnston|
|       7192|          Mary|        Bowers|
|      11178|         Laura|         Smith|
|        982|          Mary|       Robbins|
|       7160|          Mary|         Allen|
|       9322|         Kevin|         Smith|
|       9000|         Amber|         White|
|        117|          Mary|         Smith|
|       7538|          Mary|         Baker|
|       8348|          Mary|        Maddox|
|       6199|          Mary|        Jacobs|
|      11014|       Heather|         Smith|
|       6321|          Mary|        Mcneil|
|       6020|          Mary|         Smith|
|       8327|          Mary|          Paul|
|       5421|          Mary|       Carlson|
|       9367|          Ryan|         Smith|
|      10903|          Mary|         Smith|
+-----------+--------------+--------------+
only showing top 20 rows

Number of rows in result5DF = 68883
Using SparkSQL
+-----------+--------------+--------------+
|customer_id|customer_fname|customer_lname|
+-----------+--------------+--------------+
|       5016|         James|        Rangel|
|      11404|         Helen|          Cook|
|       8645|        Andrew|      Johnston|
|       7192|          Mary|        Bowers|
|      11178|         Laura|         Smith|
|        982|          Mary|       Robbins|
|       7160|          Mary|         Allen|
|       9322|         Kevin|         Smith|
|       9000|         Amber|         White|
|        117|          Mary|         Smith|
|       7538|          Mary|         Baker|
|       8348|          Mary|        Maddox|
|       6199|          Mary|        Jacobs|
|      11014|       Heather|         Smith|
|       6321|          Mary|        Mcneil|
|       6020|          Mary|         Smith|
|       8327|          Mary|          Paul|
|       5421|          Mary|       Carlson|
|       9367|          Ryan|         Smith|
|      10903|          Mary|         Smith|
+-----------+--------------+--------------+
only showing top 20 rows

Number of rows in result5DF_SQL = 68883
```

Task 6

```
TASK 6. Find the customers who have not placed orders yet
Using Dataframe
+-----------+--------------+--------------+
|customer_id|customer_fname|customer_lname|
+-----------+--------------+--------------+
|       6613|         Ashley|         Smith|
|       7011|          Kevin|         Smith|
|       7552|           Carl|         Smith|
|       8243|           Gary|        Walker|
|       8343|           Mary|        Bolton|
|       8575|           Mary|       Mueller|
|       8778|           Mary|         Smith|
|       8882|        Kenneth|         Smith|
|       9060|        Matthew|         Patel|
|       9315|           Mary|         Lewis|
|        219|           Mary|       Harrell|
|        339|           Mary|        Greene|
|        469|          Randy|         Smith|
|       1187|       Dorothy|       Vazquez|
|       1481|          Grace|         Smith|
|       1808|         Albert|       Ellison|
|       2073|          Donna|      Stephens|
|       2096|           Jose|        Tanner|
|       2450|          James|         Smith|
|       4555|           Mary|         Smith|
+-----------+--------------+--------------+
only showing top 20 rows

Number of rows in result6DF = 30
Using SparkSQL
+-----------+--------------+--------------+
|customer_id|customer_fname|customer_lname|
+-----------+--------------+--------------+
|       6613|         Ashley|         Smith|
|       7011|          Kevin|         Smith|
|       7552|           Carl|         Smith|
|       8243|           Gary|        Walker|
|       8343|           Mary|        Bolton|
|       8575|           Mary|       Mueller|
|       8778|           Mary|         Smith|
|       8882|        Kenneth|         Smith|
|       9060|        Matthew|         Patel|
|       9315|           Mary|         Lewis|
|        219|           Mary|       Harrell|
|        339|           Mary|        Greene|
|        469|          Randy|         Smith|
|       1187|       Dorothy|       Vazquez|
|       1481|          Grace|         Smith|
|       1808|         Albert|       Ellison|
|       2073|          Donna|      Stephens|
|       2096|           Jose|        Tanner|
|       2450|          James|         Smith|
|       4555|           Mary|         Smith|
+-----------+--------------+--------------+
only showing top 20 rows

Number of rows in result6DF_SQL = 30
```

Task 7

```
TASK 7(A). Find top 5 customer with highest number of orders
Using Dataframe
+-----------+-----+
|customer_id|count|
+-----------+-----+
|        569|   16|
|       5897|   16|
|       6316|   16|
|      12431|   16|
|        221|   15|
+-----------+-----+

Using SparkSQL
+-----------+-----+
|customer_id|count|
+-----------+-----+
|        569|   16|
|       5897|   16|
|       6316|   16|
|      12431|   16|
|        221|   15|
+-----------+-----+

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
TASK 7(B). Find top 5 customer with highest sum of total orders
Using Dataframe
+-----------+---------+
|customer_id|order_sum|
+-----------+---------+
|       5624|   708692|
|        569|   655204|
|       4249|   644299|
|       5004|   634073|
|       5897|   618281|
+-----------+---------+

Using SparkSQL
+-----------+---------+
|customer_id|order_sum|
+-----------+---------+
|       5624|   708692|
|        569|   655204|
|       4249|   644299|
|       5004|   634073|
|       5897|   618281|
+-----------+---------+
```

Task 8

```
TASK 8. Find the customer who did not order in last 1 month or for long time
Using Dataframe
+-----------+
|customer_id|
+-----------+
|       5016|
|      11404|
|       8645|
|       7192|
|      11178|
|        982|
|       7160|
|       9322|
|       9000|
|        117|
|       7538|
|       8348|
|       6199|
|      11014|
|       6321|
|       6020|
|       8327|
|       5421|
|       9367|
|      10903|
+-----------+
only showing top 20 rows

Number of rows in result8DF = 68883
Using SparkSQL
+-----------+
|customer_id|
+-----------+
|       5016|
|      11404|
|       8645|
|       7192|
|      11178|
|        982|
|       7160|
|       9322|
|       9000|
|        117|
|       7538|
|       8348|
|       6199|
|      11014|
|       6321|
|       6020|
|       8327|
|       5421|
|       9367|
|      10903|
+-----------+
only showing top 20 rows

Number of rows in result8DF_SQL = 68883
```

Task 9

```
TASK 9. Find the last order date for all customers
Using Dataframe
+-----------+-------------------+
|customer_id|    max(order_date)|
+-----------+-------------------+
|       7754|2014-06-14 00:00:00|
|       6336|2014-07-01 00:00:00|
|       9465|2014-06-17 00:00:00|
|       1591|2014-07-19 00:00:00|
|       2659|2014-07-11 00:00:00|
|       7240|2014-06-20 00:00:00|
|       4935|2014-06-23 00:00:00|
|       6397|2014-06-23 00:00:00|
|      10817|2014-07-15 00:00:00|
|       5518|2014-07-09 00:00:00|
|       3749|2014-07-05 00:00:00|
|       3794|2014-06-26 00:00:00|
|       1238|2014-06-27 00:00:00|
|       5300|2014-06-30 00:00:00|
|       9900|2014-06-27 00:00:00|
|      10362|2014-06-30 00:00:00|
|       3175|2014-07-01 00:00:00|
|       5803|2014-07-22 00:00:00|
|       7833|2014-07-04 00:00:00|
|       2366|2014-07-06 00:00:00|
+-----------+-------------------+
only showing top 20 rows

Number of rows in result9DF = 12405
Using SparkSQL
+-----------+-------------------+
|customer_id|    max(order_date)|
+-----------+-------------------+
|       7754|2014-06-14 00:00:00|
|       6336|2014-07-01 00:00:00|
|       9465|2014-06-17 00:00:00|
|       1591|2014-07-19 00:00:00|
|       2659|2014-07-11 00:00:00|
|       7240|2014-06-20 00:00:00|
|       4935|2014-06-23 00:00:00|
|       6397|2014-06-23 00:00:00|
|      10817|2014-07-15 00:00:00|
|       5518|2014-07-09 00:00:00|
|       3749|2014-07-05 00:00:00|
|       3794|2014-06-26 00:00:00|
|       1238|2014-06-27 00:00:00|
|       5300|2014-06-30 00:00:00|
|       9900|2014-06-27 00:00:00|
|      10362|2014-06-30 00:00:00|
|       3175|2014-07-01 00:00:00|
|       5803|2014-07-22 00:00:00|
|       7833|2014-07-04 00:00:00|
|       2366|2014-07-06 00:00:00|
+-----------+-------------------+
only showing top 20 rows

Number of rows in result9DF_SQL = 12405
```

Task 10

```
TASK 10. Find open and close number of orders for a customer
Using Dataframe
open_orders = 105

closed_orders = 15

Using SparkSQL
+-----------+
|open_orders|
+-----------+
|        105|
+-----------+

+-------------+
|closed_orders|
+-------------+
|           15|
+-------------+
```

Task 11

```
TASK 11. Find number of customers in every state
Using Dataframe
+--------------+-----+
|customer_state|count|
+--------------+-----+
|SC            |41   |
|AZ            |213  |
|LA            |63   |
|MN            |39   |
|NJ            |219  |
|DC            |42   |
|OR            |119  |
|VA            |136  |
|RI            |15   |
|KY            |35   |
|MI            |254  |
|NV            |103  |
|WI            |64   |
|ID            |9    |
|CA            |2012 |
|CT            |73   |
|MT            |7    |
|NC            |150  |
|MD            |164  |
|DE            |23   |
+--------------+-----+
only showing top 20 rows

Number of rows in result11DF = 44
Using SparkSQL
+--------------+-----+
|customer_state|count|
+--------------+-----+
|SC            |41   |
|AZ            |213  |
|LA            |63   |
|MN            |39   |
|NJ            |219  |
|DC            |42   |
|OR            |119  |
|VA            |136  |
|RI            |15   |
|KY            |35   |
|MI            |254  |
|NV            |103  |
|WI            |64   |
|ID            |9    |
|CA            |2012 |
|CT            |73   |
|MT            |7    |
|NC            |150  |
|MD            |164  |
|DE            |23   |
+--------------+-----+
only showing top 20 rows

Number of rows in result11DF_SQL = 44
```

Task 12

```
TASK 12. Number of customers in every city
Using Dataframe
+---------------+-----+
|customer_city  |count|
+---------------+-----+
|Hanover        |9    |
|Caguas         |4584 |
|Corona         |25   |
|Tempe          |35   |
|Bowling Green  |8    |
|Springfield    |3    |
|Lawrenceville  |12   |
|Palatine       |8    |
|North Las Vegas|12   |
|Phoenix        |64   |
|Bountiful      |8    |
|Plainfield     |13   |
|Levittown      |8    |
|Waukegan       |9    |
|Pittsburg      |4    |
|Hollywood      |24   |
|Toa Alta       |3    |
|Mount Prospect |7    |
|Toms River     |11   |
|Brighton       |13   |
+---------------+-----+
only showing top 20 rows

Number of rows in result12DF = 562
Using SparkSQL
+---------------+-----+
|customer_city  |count|
+---------------+-----+
|Hanover        |9    |
|Caguas         |4584 |
|Corona         |25   |
|Tempe          |35   |
|Bowling Green  |8    |
|Springfield    |3    |
|Lawrenceville  |12   |
|Palatine       |8    |
|North Las Vegas|12   |
|Phoenix        |64   |
|Bountiful      |8    |
|Plainfield     |13   |
|Levittown      |8    |
|Waukegan       |9    |
|Pittsburg      |4    |
|Hollywood      |24   |
|Toa Alta       |3    |
|Mount Prospect |7    |
|Toms River     |11   |
|Brighton       |13   |
+---------------+-----+
only showing top 20 rows

Number of rows in result12DF_SQL = 562
```

Task 13

```
TASK 13. Latest 5 orders
Using Dataframe
+--------+--------------------+----------------+---------------+
|order_id|          order_date|order_customer_id|   order_status|
+--------+--------------------+----------------+---------------+
|   57595|2014-07-24 00:00:00|            9102|       COMPLETE|
|   57599|2014-07-24 00:00:00|            4500|PENDING_PAYMENT|
|   57596|2014-07-24 00:00:00|            2634|PENDING_PAYMENT|
|   57597|2014-07-24 00:00:00|            4574|PENDING_PAYMENT|
|   57598|2014-07-24 00:00:00|             138|        PENDING|
+--------+--------------------+----------------+---------------+

Using SparkSQL
+--------+--------------------+----------------+---------------+
|order_id|          order_date|order_customer_id|   order_status|
+--------+--------------------+----------------+---------------+
|   57595|2014-07-24 00:00:00|            9102|       COMPLETE|
|   57599|2014-07-24 00:00:00|            4500|PENDING_PAYMENT|
|   57596|2014-07-24 00:00:00|            2634|PENDING_PAYMENT|
|   57597|2014-07-24 00:00:00|            4574|PENDING_PAYMENT|
|   57598|2014-07-24 00:00:00|             138|        PENDING|
+--------+--------------------+----------------+---------------+
```

Task 14

```
TASK 14. Count of orders based on city
Using Dataframe
+---------------+---------------+
|  customer_city|count(order_id)|
+---------------+---------------+
|        Hanover|             52|
|         Caguas|          25487|
|         Corona|            160|
|          Tempe|            182|
|  Bowling Green|             45|
|    Springfield|             21|
|       Palatine|             50|
|North Las Vegas|             64|
|  Lawrenceville|             55|
|      Bountiful|             39|
|        Phoenix|            358|
|     Plainfield|             60|
|      Levittown|             47|
|       Waukegan|             57|
|      Hollywood|            104|
|      Pittsburg|             22|
| Mount Prospect|             50|
|       Toa Alta|             16|
|     Toms River|             67|
|       Brighton|             70|
+---------------+---------------+
only showing top 20 rows

Number of rows in result14DF = 562
Using SparkSQL
+---------------+---------------+
|  customer_city|count_of_orders|
+---------------+---------------+
|        Hanover|             52|
|         Caguas|          25487|
|         Corona|            160|
|          Tempe|            182|
|  Bowling Green|             45|
|    Springfield|             21|
|       Palatine|             50|
|North Las Vegas|             64|
|  Lawrenceville|             55|
|      Bountiful|             39|
|        Phoenix|            358|
|     Plainfield|             60|
|      Levittown|             47|
|       Waukegan|             57|
|      Hollywood|            104|
|      Pittsburg|             22|
| Mount Prospect|             50|
|       Toa Alta|             16|
|     Toms River|             67|
|       Brighton|             70|
+---------------+---------------+
only showing top 20 rows

Number of rows in result14DF_SQL = 562
```

Enhancement

Task 1 - Read the input dataset in the parquet format

Steps:

1. Use sqoop to write data from MySQL to HDFS in csv format

```
$ sqoop import --connect jdbc:mysql://ms.itversity.com:3306/retail_db --username retail_user --password itversity --table orders --warehouse-dir /user/itv001183/dataset

$ sqoop import --connect jdbc:mysql://ms.itversity.com:3306/retail_db --username retail_user --password itversity --table customers --warehouse-dir /user/itv001183/dataset
```

2. Read the csv dataset as a Spark dataframe and use Spark to write it into a different HDFS location as a parquet file

```
// Read Orders dataset from HDFS and store output in Parquet format on HDFS
val tempOrders = spark
  .read
  .option("inferSchema", "true")
  .csv(inputPathOrders)
  .withColumnRenamed("_c0", "order_id")
  .withColumnRenamed("_c1", "order_date")
  .withColumnRenamed("_c2", "order_customer_id")
  .withColumnRenamed("_c3", "order_status")

tempOrders
  .write
  .format("parquet")
  .mode("overwrite")
  .save(parquetPathOrders)

// Read Customers dataset from HDFS and store output in Parquet format on HDFS
val tempCustomers = spark
  .read
  .option("inferSchema", "true")
  .csv(inputPathCustomers)
  .withColumnRenamed("_c0", "customer_id")
  .withColumnRenamed("_c1", "customer_fname")
  .withColumnRenamed("_c2", "customer_lname")
  .withColumnRenamed("_c3", "customer_email")
  .withColumnRenamed("_c4", "customer_password")
  .withColumnRenamed("_c5", "customer_street")
  .withColumnRenamed("_c6", "customer_city")
  .withColumnRenamed("_c7", "customer_state")
  .withColumnRenamed("_c8", "customer_zipcode")

tempCustomers
  .write
  .format("parquet")
  .mode("overwrite")
  .save(parquetPathCustomers)
```

3. Read the parquet dataset as the input dataframe in Spark

```
// Read Orders dataset from HDFS
val ordersDF = spark
  .read
  .option("inferSchema", "true")
  .parquet(parquetPathOrders)

ordersDF.show()

ordersDF.createOrReplaceTempView("orders")

// Read Customers dataset from HDFS
val customersDF = spark
  .read
  .option("inferSchema", "true")
  .parquet(parquetPathCustomers)

customersDF.show()

customersDF.createOrReplaceTempView("customers")
```

Task 2 - Masking

```scala
// Read Customers dataset from HDFS
val customersDF = spark
  .read
  .option("inferSchema", "true")
  .parquet(parquetPathCustomers)
  .withColumn("customer_email", lit("*******"))       // customer_email is already masked, but still masking for uniformity
  .withColumn("customer_password", lit("********"))    // customer_password is already masked, but still masking for uniformity
  .withColumn("customer_street", lit("*******"))
  .withColumn("customer_zipcode", lit("*******"))
```