Spark:

To start with spark, I have worked the spark code on spark version 2.3.0.

```
[training@localhost ~]$ pyspark
Python 2.6.6 (r266:84292, Jun 18 2012, 14:18:47)
[GCC 4.4.6 20110731 (Red Hat 4.4.6-3)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
2018-04-17 21:58:05 WARN  Utils:66 - Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1, but we couldn't find any external IP address!
2018-04-17 21:58:05 WARN  Utils:66 - Set SPARK_LOCAL_IP if you need to bind to another address
2018-04-17 21:58:08 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
2018-04-17 21:58:11 WARN  MacAddressUtil:136 - Failed to find a usable hardware address from the network interfaces; using random bytes: aa:60:c8:3b:ec:ce:79:eb
2018-04-17 21:58:13 WARN  Utils:66 - Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.3.0
      /_/

Using Python version 2.6.6 (r266:84292, Jun 18 2012 14:18:47)
SparkSession available as 'spark'.
>>>
```

Once the spark is install and running, go to the pyspark_ph4.py to run the queries.

1. I have first imported SqlContext from pyspark used to run the sql queries.

```
>>> from pyspark import SparkContext
>>>
```

2. To reading the .csv format file from the given file path and storing into a dataframe, df

```
>>> df = spark.read.csv("file:///home/training/desktop")
>>> df.show()
```

3. To renaming the columns in the dataset file and stored in df1

```
>>> column = ['date','sender_firstname','sender_lastname','receiver','subject','id']
>>> df1 = df.toDF(*columns)
```

4. To renaming the data frame to df

```
>>> df = df.toDF(*columns)
```

5. Creating and naming the table as data, in which the data from the dataframe will be stored

```
>>> sqlContext.registerDataFrameAsTable(df, "data")
```

6. For queries to run on the table and store the result (query2_result.show() command under execution)

```
>>> query2_result=sqlContext.sql("select sender_firstname,sender_lastname, count(*) as count from data group by sender_firstname,sender_lastname order by count desc li
mit 15")
>>> query2_result.show()
[Stage 12:==============================>                    (116 + 2) / 200]
```