(base) C:\spark-2.3.4-bin-hadoop2.6>pyspark --help

Usage: bin\pyspark.cmd [options]

Options:

--master MASTER\_URL spark://host:port, mesos://host:port, yarn,

k8s://https://host:port, or local (Default: local[\*]).

--deploy-mode DEPLOY\_MODE Whether to launch the driver program locally ("client") or

on one of the worker machines inside the cluster ("cluster")

(Default: client).

--class CLASS\_NAME Your application's main class (for Java / Scala apps).

--name NAME A name of your application.

--jars JARS Comma-separated list of jars to include on the driver

and executor classpaths.

--packages Comma-separated list of maven coordinates of jars to include

on the driver and executor classpaths. Will search the local

maven repo, then maven central and any additional remote

repositories given by --repositories. The format for the

coordinates should be groupId:artifactId:version.

--exclude-packages Comma-separated list of groupId:artifactId, to exclude while

resolving the dependencies provided in --packages to avoid

dependency conflicts.

--repositories Comma-separated list of additional remote repositories to

search for the maven coordinates given with --packages.

--py-files PY\_FILES Comma-separated list of .zip, .egg, or .py files to place

on the PYTHONPATH for Python apps.

--files FILES Comma-separated list of files to be placed in the working

directory of each executor. File paths of these files

in executors can be accessed via SparkFiles.get(fileName).

--conf PROP=VALUE Arbitrary Spark configuration property.

--properties-file FILE Path to a file from which to load extra properties. If not

specified, this will look for conf/spark-defaults.conf.

at java.lang.Thread.run(Thread.java:748)aitResult(ThreadUtils.scala:201)

--driver-memory MEM Memory for driver (e.g. 1000M, 2G) (Default: 1024M).nds]

--driver-java-options Extra Java options to pass to the driver.cala:219)

--driver-library-path Extra library path entries to pass to the driver.3)

--driver-class-path Extra class path entries to pass to the driver. Note that

jars added with --jars are automatically included in the

classpath.: Futures timed out after [10 seconds]. This timeout is controlled by spark.executor.heartbeatInterval

20/09/17 22:10:40 WARN NettyRpcEnv: Ignored message: HeartbeatResponse(true)createRpcTimeoutException(RpcTimeout.scala:47)

--executor-memory MEM Memory per executor (e.g. 1000M, 2G) (Default: 1G).Else(RpcTimeout.scala:62)

20/09/17 22:10:41 WARN SparkContext: Killing executors is not supported by current scheduler.out.scala:58)

--proxy-user NAME User to impersonate when submitting the application.la:36)

This argument does not work with --principal / --keytab.

at org.apache.spark.rpc.RpcEndpointRef.askSync(RpcEndpointRef.scala:92)

--help, -h Show this help message and exit.$executor$Executor$$reportHeartBeat(Executor.scala:785)

--verbose, -v Print additional debug output.un$run$1.apply$mcV$sp(Executor.scala:814)

--version, Print the version of current Spark.n$1.apply(Executor.scala:814)

at org.apache.spark.executor.Executor$$anon$2$$anonfun$run$1.apply(Executor.scala:814)

Cluster deploy mode only:k.util.Utils$.logUncaughtExceptions(Utils.scala:1992)

--driver-cores NUM Number of cores used by the driver, only in cluster mode

(Default: 1).unnableAdapter.call(Executors.java:511)

at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308)

Spark standalone or Mesos with cluster deploy mode only:or$ScheduledFutureTask.access$301(ScheduledThreadPoolExecutor.java:180)

--supervise If given, restarts the driver on failure.tureTask.run(ScheduledThreadPoolExecutor.java:294)

--kill SUBMISSION\_ID If given, kills the driver specified.PoolExecutor.java:1149)

--status SUBMISSION\_ID If given, requests the status of the driver specified.a:624)

at java.lang.Thread.run(Thread.java:748)

Spark standalone and Mesos only:imeoutException: Futures timed out after [10 seconds]

--total-executor-cores NUM Total cores for all executors.y(Promise.scala:219)

at scala.concurrent.impl.Promise$DefaultPromise.result(Promise.scala:223)

Spark standalone and YARN only:.ThreadUtils$.awaitResult(ThreadUtils.scala:201)

--executor-cores NUM Number of cores per executor. (Default: 1 in YARN mode,

or all available cores on the worker in standalone mode)

20/09/17 16:41:50 WARN NettyRpcEnv: Ignored message: HeartbeatResponse(false)

YARN-only:

--queue QUEUE\_NAME The YARN queue to submit to (Default: "default").

--num-executors NUM Number of executors to launch (Default: 2).

If dynamic allocation is enabled, the initial number of

executors will be at least NUM.

--archives ARCHIVES Comma separated list of archives to be extracted into the

working directory of each executor.

--principal PRINCIPAL Principal to be used to login to KDC, while running on

secure HDFS.

--keytab KEYTAB The full path to the file that contains the keytab for the

principal specified above. This keytab will be copied to

the node running the Application Master via the Secure

Distributed Cache, for renewing the login tickets and the

delegation tokens periodically.

(base) C:\spark-2.3.4-bin-hadoop2.6>

Details - Duration 40 minutes

* Data set [URL 892](https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2)
* Choose language of your choice Python or Scala
* Data is available in HDFS file system under /public/crime/csv
* You can check properties of files using hadoop fs -ls -h /public/crime/csv
* Structure of data (ID,Case Number,Date,Block,IUCR,Primary Type,Description,Location Description,Arrest,Domestic,Beat,District,Ward,Community Area,FBI Code,X Coordinate,Y Coordinate,Year,Updated On,Latitude,Longitude,Location)
* File format - text file
* Delimiter - “,”
* Get monthly count of primary crime type, sorted by month in ascending and number of crimes per type in descending order
* Store the result in HDFS path /user/<YOUR\_USER\_ID>/solutions/solution01/crimes\_by\_type\_by\_month
* Output File Format: TEXT
* Output Columns: Month in YYYYMM format, crime count, crime type
* Output Delimiter: \t (tab delimited)
* Output Compression: gzip

export SPARK\_MAJOR\_VERSION=2

$ pyspark --master yarn --conf spark.ui.port=12181 --num-executors 4 --executor-cores 2 --executor-memory 2G

>>> from pyspark import SQLContext

>>> from pyspark.sql import SparkSession

>>> from pyspark.sql.functions import \*

>>> from pyspark.sql.window import \*

>>> crimeFile = spark.read.format('csv').option('sep', ',').schema('ID INT, Case\_Number STRING, Date STRING, Block STRING, IUCR STRING, Primary\_Type STRING, Description STRING, Location\_Description STRING, Arrest STRING, Domestic STRING, Beat STRING, District STRING, Ward STRING, Community STRING, Area STRING, FBI\_code STRING, X\_cordinate STRING, Y\_cordinate STRING, YEAR STRING, Updated\_On STRING, Latitude STRING, Longitude STRING, Location STRING').load('/public/crime/csv')

>>> crimeFile.show()

>>> crimeFile1=crimeFile.na.drop(subset=['ID'])

>>> crimeFile1.show()

<https://stackoverflow.com/questions/37262762/filter-pyspark-dataframe-column-with-none-value/37262973>

>>> crimeFile1.select('Primary\_Type').distinct().show(40,truncate=False)

+---------------------------------+

|Primary\_Type |

+---------------------------------+

|OFFENSE INVOLVING CHILDREN |

|STALKING |

|PUBLIC PEACE VIOLATION |

|OBSCENITY |

|NON-CRIMINAL (SUBJECT SPECIFIED) |

|ARSON |

|DOMESTIC VIOLENCE |

|GAMBLING |

|CRIMINAL TRESPASS |

|ASSAULT |

|NON - CRIMINAL |

|LIQUOR LAW VIOLATION |

|MOTOR VEHICLE THEFT |

|THEFT |

|BATTERY |

|ROBBERY |

|HOMICIDE |

|RITUALISM |

|PUBLIC INDECENCY |

|CRIM SEXUAL ASSAULT |

|HUMAN TRAFFICKING |

|INTIMIDATION |

|PROSTITUTION |

|DECEPTIVE PRACTICE |

|CONCEALED CARRY LICENSE VIOLATION|

|SEX OFFENSE |

|CRIMINAL DAMAGE |

|NARCOTICS |

|NON-CRIMINAL |

|OTHER OFFENSE |

|KIDNAPPING |

|BURGLARY |

|WEAPONS VIOLATION |

|OTHER NARCOTIC VIOLATION |

|INTERFERENCE WITH PUBLIC OFFICER |

+---------------------------------+

>>> crimeFile1.select('Primary\_Type').distinct().count()

35

>>> crimeFile1.select('Primary\_Type', 'Date').show()

+-----------------+--------------------+

| Primary\_Type| Date|

+-----------------+--------------------+

| CRIMINAL DAMAGE|07/24/2007 10:11:...|

| BATTERY|07/24/2007 01:00:...|

| BATTERY|07/24/2007 02:20:...|

| THEFT|07/21/2007 12:30:...|

| CRIMINAL DAMAGE|07/26/2007 03:00:...|

| CRIMINAL DAMAGE|07/20/2007 11:40:...|

| THEFT|07/25/2007 10:20:...|

| THEFT|07/26/2007 03:00:...|

|WEAPONS VIOLATION|07/25/2007 06:50:...|

| BURGLARY|07/24/2007 11:30:...|

| THEFT|07/22/2007 12:05:...|

| BATTERY|07/25/2007 02:10:...|

| THEFT|07/23/2007 09:00:...|

| OTHER OFFENSE|04/23/2007 12:01:...|

| THEFT|07/23/2007 08:00:...|

| THEFT|07/23/2007 07:00:...|

| BATTERY|07/23/2007 10:00:...|

| NARCOTICS|07/19/2007 11:00:...|

| BATTERY|07/25/2007 08:35:...|

| OTHER OFFENSE|07/25/2007 04:00:...|

+-----------------+--------------------+

>>> from pyspark.sql.types import DateType

>>> from pyspark.sql.functions import to\_date

>>> crimeFile1.select('Primary\_Type', date\_format(unix\_timestamp('Date', "MM/dd/YYYY").cast("timestamp"), 'MM/YYYY').alias('Month')).show()

+-----------------+-------+

| Primary\_Type| Month|

+-----------------+-------+

| CRIMINAL DAMAGE|12/2007|

| BATTERY|12/2007|

| BATTERY|12/2007|

| THEFT|12/2007|

| CRIMINAL DAMAGE|12/2007|

| CRIMINAL DAMAGE|12/2007|

| THEFT|12/2007|

| THEFT|12/2007|

|WEAPONS VIOLATION|12/2007|

| BURGLARY|12/2007|

| THEFT|12/2007|

| BATTERY|12/2007|

| THEFT|12/2007|

| OTHER OFFENSE|12/2007|

| THEFT|12/2007|

| THEFT|12/2007|

| BATTERY|12/2007|

| NARCOTICS|12/2007|

| BATTERY|12/2007|

| OTHER OFFENSE|12/2007|

+-----------------+-------+

>>> crimeFile1=crimeFile1.withColumn('Month', date\_format(unix\_timestamp('Date', "MM/dd/YYYY").cast("timestamp"), 'MM/YYYY'))

>>> crimeFile1.show()

>>> from pyspark.sql.window import \*

>>> from itertools import \*

>>> spec = Window.partitionBy('Primary\_Type').orderBy(crimeFile1.Month, crimeFile1.Primary\_Type.desc())

>>> crimeFile1.select('Month', f.count('Primary\_Type').over(spec).alias('crime\_count'), 'Primary\_Type').orderBy('Month', crimeFile1.Primary\_Type.desc()).show()

1. Launch pyspark with optmal resources
2. Read data from hdfs into RDD
3. Convert each element into tuple ((month, crime\_type), 1)
4. Perform aggregation to get ((month, crime\_type), count)
5. Convert each element into ((month, -count), month + ‘\t’ + count + ‘\t’ + crime\_type)
6. Sort the data
7. Discard the key and get the value – month + ‘\t’ + count + ‘\t’ + crime\_type
8. Save into HDFS using compression and make sure number of files are either one or two .

Through API core::

>>> crimeData = sc.textFile("/public/crime/csv")

>>> for i in crimeData.take(10): print(i)

...

ID,Case Number,Date,Block,IUCR,Primary Type,Description,Location Description,Arrest,Domestic,Beat,District,Ward,Community Area,FBI Code,X Coordinate,Y Coordinate,Year,Updated On,Latitude,Longitude,Location

5679862,HN487108,07/24/2007 10:11:00 PM,054XX S ABERDEEN ST,1320,CRIMINAL DAMAGE,TO VEHICLE,STREET,false,false,0934,009,16,61,14,1169912,1868555,2007,04/15/2016 08:55:02 AM,41.794811309,-87.652466989,"(41.794811309, -87.652466989)"

5679863,HN488302,07/24/2007 01:00:00 PM,082XX S TALMAN AVE,0460,BATTERY,SIMPLE,STREET,false,false,0835,008,18,70,08B,1160134,1850078,2007,04/15/2016 08:55:02 AM,41.744314668,-87.688830696,"(41.744314668, -87.688830696)"

5679864,HN487195,07/24/2007 02:20:00 PM,034XX N MILWAUKEE AVE,0486,BATTERY,DOMESTIC BATTERY SIMPLE,RESIDENCE,false,true,1731,017,30,16,08B,1148204,1922490,2007,04/15/2016 08:55:02 AM,41.943259605,-87.730682304,"(41.943259605, -87.730682304)"

5679865,HN484199,07/21/2007 12:30:00 PM,035XX W BELMONT AVE,0820,THEFT,$500 AND UNDER,STREET,false,false,1412,014,35,21,06,1152064,1921048,2007,04/15/2016 08:55:02 AM,41.9392273,-87.716532899,"(41.9392273, -87.716532899)"

5679866,HN489706,07/26/2007 03:00:00 AM,048XX W ARGYLE ST,1305,CRIMINAL DAMAGE,CRIMINAL DEFACEMENT,RESIDENCE,false,false,1623,016,45,12,14,1143169,1932807,2007,04/15/2016 08:55:02 AM,41.971665985,-87.74893007,"(41.971665985, -87.74893007)"

5679868,HN479171,07/20/2007 11:40:00 PM,023XX S CALIFORNIA AVE,1320,CRIMINAL DAMAGE,TO VEHICLE,STREET,false,false,1033,010,12,30,14,1158058,1888522,2007,04/15/2016 08:55:02 AM,41.849852628,-87.695392076,"(41.849852628, -87.695392076)"

5679870,HN487786,07/25/2007 10:20:00 AM,001XX E 71ST ST,0890,THEFT,FROM BUILDING,GROCERY FOOD STORE,false,false,0323,003,6,69,06,1178576,1857978,2007,04/15/2016 08:55:02 AM,41.765594255,-87.62101766,"(41.765594255, -87.62101766)"

5679872,HN489597,07/26/2007 03:00:00 AM,024XX N FRANCISCO AVE,0810,THEFT,OVER $500,APARTMENT,false,false,1414,014,35,22,06,1156634,1915948,2007,04/15/2016 08:55:02 AM,41.925141071,-87.699875328,"(41.925141071, -87.699875328)"

5679873,HN488796,07/25/2007 06:50:00 PM,098XX S PARNELL AVE,143A,WEAPONS VIOLATION,UNLAWFUL POSS OF HANDGUN,PARK PROPERTY,true,false,2223,022,21,73,15,1174363,1839821,2007,04/15/2016 08:55:02 AM,41.715863801,-87.636998103,"(41.715863801, -87.636998103)"

>>>

>>> crimeDataHead = crimeData.first()

>>> crimeDataNoHeader = crimeData.filter(lambda b: b != crimeDataHead)

>>> for i in crimeDataNoHeader.take(10): print(i)

...

5679862,HN487108,07/24/2007 10:11:00 PM,054XX S ABERDEEN ST,1320,CRIMINAL DAMAGE,TO VEHICLE,STREET,false,false,0934,009,16,61,14,1169912,1868555,2007,04/15/2016 08:55:02 AM,41.794811309,-87.652466989,"(41.794811309, -87.652466989)"

5679863,HN488302,07/24/2007 01:00:00 PM,082XX S TALMAN AVE,0460,BATTERY,SIMPLE,STREET,false,false,0835,008,18,70,08B,1160134,1850078,2007,04/15/2016 08:55:02 AM,41.744314668,-87.688830696,"(41.744314668, -87.688830696)"

5679864,HN487195,07/24/2007 02:20:00 PM,034XX N MILWAUKEE AVE,0486,BATTERY,DOMESTIC BATTERY SIMPLE,RESIDENCE,false,true,1731,017,30,16,08B,1148204,1922490,2007,04/15/2016 08:55:02 AM,41.943259605,-87.730682304,"(41.943259605, -87.730682304)"

5679865,HN484199,07/21/2007 12:30:00 PM,035XX W BELMONT AVE,0820,THEFT,$500 AND UNDER,STREET,false,false,1412,014,35,21,06,1152064,1921048,2007,04/15/2016 08:55:02 AM,41.9392273,-87.716532899,"(41.9392273, -87.716532899)"

5679866,HN489706,07/26/2007 03:00:00 AM,048XX W ARGYLE ST,1305,CRIMINAL DAMAGE,CRIMINAL DEFACEMENT,RESIDENCE,false,false,1623,016,45,12,14,1143169,1932807,2007,04/15/2016 08:55:02 AM,41.971665985,-87.74893007,"(41.971665985, -87.74893007)"

5679868,HN479171,07/20/2007 11:40:00 PM,023XX S CALIFORNIA AVE,1320,CRIMINAL DAMAGE,TO VEHICLE,STREET,false,false,1033,010,12,30,14,1158058,1888522,2007,04/15/2016 08:55:02 AM,41.849852628,-87.695392076,"(41.849852628, -87.695392076)"

5679870,HN487786,07/25/2007

>>> def getCrimeDetails(rec):

... data=rec.split(",")

... dateWithTime = data[2]

... date = dateWithTime.split(" ")[0]

... month = int(date.split("/")[2] + date.split("/")[0])

... crimeType = data[5]

... return((month, crimeType), 1)

>>> crimeFileMap = crimeFileNoHeader.map(lambda rec: getCrimeDetails(rec))

>>> for i in crimeFileMap.take(10): print(i)

...

((200707, u'CRIMINAL DAMAGE'), 1)

((200707, u'BATTERY'), 1)

((200707, u'BATTERY'), 1)

((200707, u'THEFT'), 1)

((200707, u'CRIMINAL DAMAGE'), 1)

((200707, u'CRIMINAL DAMAGE'), 1)

((200707, u'THEFT'), 1)

((200707, u'THEFT'), 1)

((200707, u'WEAPONS VIOLATION'), 1)

((200707, u'BURGLARY'), 1)

>>>

>>> from operator import add

>>> monthlyCrimeType = crimeFileMap.reduceByKey(add)

>>> for i in monthlyCrimeType.take(10): print(i)

...

((200606, u'OTHER NARCOTIC VIOLATION'), 1)

((200911, u'KIDNAPPING'), 34)

((200607, u'NARCOTICS'), 4701)

((200505, u'INTERFERENCE WITH PUBLIC OFFICER'), 51)

((200312, u'DECEPTIVE PRACTICE'), 1154)

((200607, u'KIDNAPPING'), 30)

((201410, u'BURGLARY'), 1359)

((200105, u'GAMBLING'), 94)

((201303, u'OFFENSE INVOLVING CHILDREN'), 215)

((201007, u'WEAPONS VIOLATION'), 326)

>>>

>>> monthlyCrimeTimeMap = monthlyCrimeType.map(lambda rec: ((rec[0][0], -rec[1]), str(rec[0][0]) + "\t" + str(rec[1]) + "\t" + str(rec[0][1])))

>>> for i in monthlyCrimeTimeMap.take(10): print(i)

...

((200606, -1), '200606\t1\tOTHER NARCOTIC VIOLATION')

((201306, -1477), '201306\t1477\tBURGLARY')

((200801, -1082), '200801\t1082\tCRIMINAL TRESPASS')

((200607, -4701), '200607\t4701\tNARCOTICS')

((200505, -51), '200505\t51\tINTERFERENCE WITH PUBLIC OFFICER')

((200312, -1154), '200312\t1154\tDECEPTIVE PRACTICE')

((200607, -30), '200607\t30\tKIDNAPPING')

((201410, -1359), '201410\t1359\tBURGLARY')

((200803, -311), '200803\t311\tWEAPONS VIOLATION')

((201007, -326), '201007\t326\tWEAPONS VIOLATION')

>>> monthlyCrimeCountSorted = monthlyCrimeTimeMap.sortByKey(lambda a: a[0])

>>> for i in monthlyCrimeCountSorted.take(10): print(i)

...

((200101, -7866), '200101\t7866\tTHEFT')

((200101, -6525), '200101\t6525\tBATTERY')

((200101, -4714), '200101\t4714\tNARCOTICS')

((200101, -3966), '200101\t3966\tCRIMINAL DAMAGE')

((200101, -2799), '200101\t2799\tOTHER OFFENSE')

((200101, -2123), '200101\t2123\tASSAULT')

((200101, -2095), '200101\t2095\tMOTOR VEHICLE THEFT')

((200101, -1934), '200101\t1934\tBURGLARY')

((200101, -1396), '200101\t1396\tROBBERY')

((200101, -1387), '200101\t1387\tDECEPTIVE PRACTICE')

>>> FinalResultCountCrimeType = monthlyCrimeCountSorted.map(lambda a: a[1])

>>> for i in FinalResultCountCrimeType.take(10): print(i)

...

200101 7866 THEFT

200101 6525 BATTERY

200101 4714 NARCOTICS

200101 3966 CRIMINAL DAMAGE

200101 2799 OTHER OFFENSE

200101 2123 ASSAULT

200101 2095 MOTOR VEHICLE THEFT

200101 1934 BURGLARY

200101 1396 ROBBERY

200101 1387 DECEPTIVE PRACTICE

>>>

>>> FinalResultCountCrimeType.coalesce(2).saveAsTextFile("/user/saurabhbhanwala/solutions/solution01/crimebyType\_by\_month\_RDD", compressionCodecClass="org.apache.hadoop.io.compress.GzipCodec")

>>> import os

>>> os.system("hdfs dfs -ls /user/saurabhbhanwala/solutions/solution01/crimebyType\_by\_month\_RDD")

Found 3 items

-rw-r--r-- 2 saurabhbhanwala hdfs 0 2020-09-20 04:43 /user/saurabhbhanwala/solutions/solution01/crimebyType\_by\_month\_RDD/\_SUCCESS

-rw-r--r-- 2 saurabhbhanwala hdfs 15220 2020-09-20 04:43 /user/saurabhbhanwala/solutions/solution01/crimebyType\_by\_month\_RDD/part-00000.gz

-rw-r--r-- 2 saurabhbhanwala hdfs 11658 2020-09-20 04:43 /user/saurabhbhanwala/solutions/solution01/crimebyType\_by\_month\_RDD/part-00001.gz

0

crimeFile = spark.read.format('csv').option('sep', ',').schema('ID INT, Case\_Number STRING, Date STRING, Block STRING, IUCR STRING, Primary\_Type STRING, Description STRING, Location\_Description STRING, Arrest STRING, Domestic STRING, Beat STRING, District STRING, Ward STRING, Community STRING, Area STRING, FBI\_code STRING, X\_cordinate STRING, Y\_cordinate STRING, YEAR STRING, Updated\_On STRING, Latitude STRING, Longitude STRING, Location STRING').load('/public/crime/csv')

>>> crimeFile = spark.read.format('csv').option('sep', ',').schema('ID INT, Case\_Number STRING, Date STRING, Block STRING, IUCR STRING, Primary\_Type STRING, Description STRING, Location\_Description STRING, Arrest STRING, Domestic STRING, Beat STRING, District STRING, Ward STRING, Community STRING, Area STRING, FBI\_code STRING, X\_cordinate STRING, Y\_cordinate STRING, YEAR STRING, Updated\_On STRING, Latitude STRING, Longitude STRING, Location STRING').load('/public/crime/csv')

>>> crimeFile1=crimeFile.na.drop(subset=['ID'])

>>> crimeFile1.show()

20/09/20 05:18:44 WARN DFSClient: Slow ReadProcessor read fields took 30052ms (threshold=30000ms); ack: seqno: 151 reply: SUCCESS reply: SUCCESS downstreamAckTimeNanos: 12302909 flag: 0 flag: 0, targets: [DatanodeInfoWithStorage[172.16.1.107:50010,DS-a12c4ae3-3f6a-42fc-83ff-7779a9fc0482,DISK], DatanodeInfoWithStorage[172.16.1.108:50010,DS-736614f7-27de-46b8-987f-d669be6a32a3,DISK]]

+-------+-----------+--------------------+--------------------+----+-----------------+--------------------+--------------------+------+--------+----+--------+----+---------+----+--------+-----------+-----------+--------------------+------------+-------------+--------------------+--------+

| ID|Case\_Number| Date| Block|IUCR| Primary\_Type| Description|Location\_Description|Arrest|Domestic|Beat|District|Ward|Community|Area|FBI\_code|X\_cordinate|Y\_cordinate| YEAR| Updated\_On| Latitude| Longitude|Location|

+-------+-----------+--------------------+--------------------+----+-----------------+--------------------+--------------------+------+--------+----+--------+----+---------+----+--------+-----------+-----------+--------------------+------------+-------------+--------------------+--------+

|5679862| HN487108|07/24/2007 10:11:...| 054XX S ABERDEEN ST|1320| CRIMINAL DAMAGE| TO VEHICLE| STREET| false| false|0934| 009| 16| 61| 14| 1169912| 1868555| 2007|04/15/2016 08:55:...|41.794811309|-87.652466989|(41.794811309, -8...| null|

|5679863| HN488302|07/24/2007 01:00:...| 082XX S TALMAN AVE|0460| BATTERY| SIMPLE| STREET| false| false|0835| 008| 18| 70| 08B| 1160134| 1850078| 2007|04/15/2016 08:55:...|41.744314668|-87.688830696|(41.744314668, -8...| null|

|5679864| HN487195|07/24/2007 02:20:...|034XX N MILWAUKEE...|0486| BATTERY|DOMESTIC BATTERY ...| RESIDENCE| false| true|1731| 017| 30| 16| 08B| 1148204| 1922490| 2007|04/15/2016 08:55:...|41.943259605|-87.730682304|(41.943259605, -8...| null|

|5679865| HN484199|07/21/2007 12:30:...| 035XX W BELMONT AVE|0820| THEFT| $500 AND UNDER| STREET| false| false|1412| 014| 35| 21| 06| 1152064| 1921048| 2007|04/15/2016 08:55:...| 41.9392273|-87.716532899|(41.9392273, -87....| null|

|5679866| HN489706|07/26/2007 03:00:...| 048XX W ARGYLE ST|1305| CRIMINAL DAMAGE| CRIMINAL DEFACEMENT| RESIDENCE| false| false|1623| 016| 45| 12| 14| 1143169| 1932807| 2007|04/15/2016 08:55:...|41.971665985| -87.74893007|(41.971665985, -8...| null|

|5679868| HN479171|07/20/2007 11:40:...|023XX S CALIFORNI...|1320| CRIMINAL DAMAGE| TO VEHICLE| STREET| false| false|1033| 010| 12| 30| 14| 1158058| 1888522| 2007|04/15/2016 08:55:...|41.849852628|-87.695392076|(41.849852628, -8...| null|

|5679870| HN487786|07/25/2007 10:20:...| 001XX E 71ST ST|0890| THEFT| FROM BUILDING| GROCERY FOOD STORE| false| false|0323| 003| 6| 69| 06| 1178576| 1857978| 2007|04/15/2016 08:55:...|41.765594255| -87.62101766|(41.765594255, -8...| null|

|5679872| HN489597|07/26/2007 03:00:...|024XX N FRANCISCO...|0810| THEFT| OVER $500| APARTMENT| false| false|1414| 014| 35| 22| 06| 1156634| 1915948| 2007|04/15/2016 08:55:...|41.925141071|-87.699875328|(41.925141071, -8...| null|

|5679873| HN488796|07/25/2007 06:50:...| 098XX S PARNELL AVE|143A|WEAPONS VIOLATION|UNLAWFUL POSS OF ...| PARK PROPERTY| true| false|2223| 022| 21| 73| 15| 1174363| 1839821| 2007|04/15/2016 08:55:...|41.715863801|-87.636998103|(41.715863801, -8...| null|

|5679874| HN486104|07/24/2007 11:30:...| 017XX W PIERCE AVE|0620| BURGLARY| UNLAWFUL ENTRY| APARTMENT| false| false|1433| 014| 1| 24| 05| 1164550| 1910426| 2007|04/15/2016 08:55:...|41.909824195|-87.670945023|(41.909824195, -8...| null|

|5679875| HN485891|07/22/2007 12:05:...| 016XX N WESTERN AVE|0810| THEFT| OVER $500| STREET| false| false|1434| 014| 1| 24| 06| 1160114| 1910738| 2007|04/15/2016 08:55:...|41.910773194|-87.687232407|(41.910773194, -8...| null|

|5679876| HN488180|07/25/2007 02:10:...|035XX W FULLERTON...|0460| BATTERY| SIMPLE| SIDEWALK| false| false|1413| 014| 26| 22| 08B| 1152506| 1915731| 2007|04/15/2016 08:55:...|41.924628284| -87.71504934|(41.924628284, -8...| null|

|5679877| HN485606|07/23/2007 09:00:...|015XX N ARTESIAN AVE|0820| THEFT| $500 AND UNDER| STREET| false| false|1423| 014| 1| 24| 06| 1159792| 1910263| 2007|04/15/2016 08:55:...|41.909476407| -87.68842844|(41.909476407, -8...| null|

|5679878| HN459531|04/23/2007 12:01:...|026XX N SACRAMENT...|2825| OTHER OFFENSE|HARASSMENT BY TEL...| APARTMENT| false| true|1411| 014| 35| 22| 26| 1155923| 1917476| 2007|04/15/2016 08:55:...|41.929348403| -87.70244657|(41.929348403, -8...| null|

|5679879| HN485432|07/23/2007 08:00:...| 030XX W PALMER BLVD|0820| THEFT| $500 AND UNDER| STREET| false| false|1414| 014| 35| 22| 06| 1155945| 1914325| 2007|04/15/2016 08:55:...|41.920701371|-87.702450917|(41.920701371, -8...| null|

|5679881| HN486824|07/23/2007 07:00:...| 010XX N STATE ST|0820| THEFT| $500 AND UNDER| SMALL RETAIL STORE| false| false|1824| 018| 42| 8| 06| 1176172| 1907536| 2007|04/15/2016 08:55:...|41.901639684| -87.62833826|(41.901639684, -8...| null|

|5679882| HN485055|07/23/2007 10:00:...|082XX S DR MARTIN...|0430| BATTERY|AGGRAVATED: OTHER...| STREET| false| false|0631| 006| 6| 44| 04B| 1180332| 1850554| 2007|04/15/2016 08:55:...|41.745181902|-87.614808669|(41.745181902, -8...| null|

|5679885| HN477072|07/19/2007 11:00:...| 003XX W OAK ST|2027| NARCOTICS| POSS: CRACK| CHA APARTMENT| true| false|1823| 018| 27| 8| 18| 1173484| 1907095| 2007|04/15/2016 08:55:...|41.900489748|-87.638224621|(41.900489748, -8...| null|

|5679887| HN488976|07/25/2007 08:35:...| 050XX W GLADYS AVE|0460| BATTERY| SIMPLE| SIDEWALK| false| false|1533| 015| 29| 25| 08B| 1142579| 1897865| 2007|04/15/2016 08:55:...|41.875792556|-87.751970712|(41.875792556, -8...| null|

|5679888| HN489702|07/25/2007 04:00:...|044XX S SPAULDING...|2825| OTHER OFFENSE|HARASSMENT BY TEL...| APARTMENT| false| true|0821| 008| 14| 58| 26| 1155096| 1875110| 2007|04/15/2016 08:55:...|41.813108297|-87.706622217|(41.813108297, -8...| null|

+-------+-----------+--------------------+--------------------+----+-----------------+--------------------+--------------------+------+--------+----+--------+----+---------+----+--------+-----------+-----------+--------------------+------------+-------------+--------------------+--------+

>>> crimeFile1.select('Primary\_Type').distinct().show(40,truncate=False)

+---------------------------------+

|Primary\_Type |

+---------------------------------+

|OFFENSE INVOLVING CHILDREN |

|STALKING |

|PUBLIC PEACE VIOLATION |

|OBSCENITY |

|NON-CRIMINAL (SUBJECT SPECIFIED) |

|ARSON |

|DOMESTIC VIOLENCE |

|GAMBLING |

|CRIMINAL TRESPASS |

|ASSAULT |

|NON - CRIMINAL |

|LIQUOR LAW VIOLATION |

|MOTOR VEHICLE THEFT |

|THEFT |

|BATTERY |

|ROBBERY |

|HOMICIDE |

|RITUALISM |

|PUBLIC INDECENCY |

|CRIM SEXUAL ASSAULT |

|HUMAN TRAFFICKING |

|INTIMIDATION |

|PROSTITUTION |

|DECEPTIVE PRACTICE |

|CONCEALED CARRY LICENSE VIOLATION|

|SEX OFFENSE |

|CRIMINAL DAMAGE |

|NARCOTICS |

|NON-CRIMINAL |

|OTHER OFFENSE |

|KIDNAPPING |

|BURGLARY |

|WEAPONS VIOLATION |

|OTHER NARCOTIC VIOLATION |

|INTERFERENCE WITH PUBLIC OFFICER |

+---------------------------------+

>>> crimeFile1=crimeFile1.withColumn('Month', date\_format(unix\_timestamp('Date', "MM/dd/YYYY").cast("timestamp"), 'YYYY/MM'))

>>> crimeFile1.show()

+-------+-----------+--------------------+--------------------+----+-----------------+--------------------+--------------------+------+--------+----+--------+----+---------+----+--------+-----------+-----------+--------------------+------------+-------------+--------------------+--------+-------+

| ID|Case\_Number| Date| Block|IUCR| Primary\_Type| Description|Location\_Description|Arrest|Domestic|Beat|District|Ward|Community|Area|FBI\_code|X\_cordinate|Y\_cordinate| YEAR| Updated\_On| Latitude| Longitude|Location| Month|

+-------+-----------+--------------------+--------------------+----+-----------------+--------------------+--------------------+------+--------+----+--------+----+---------+----+--------+-----------+-----------+--------------------+------------+-------------+--------------------+--------+-------+

|5679862| HN487108|07/24/2007 10:11:...| 054XX S ABERDEEN ST|1320| CRIMINAL DAMAGE| TO VEHICLE| STREET| false| false|0934| 009| 16| 61| 14| 1169912| 1868555| 2007|04/15/2016 08:55:...|41.794811309|-87.652466989|(41.794811309, -8...| null|2007/12|

|5679863| HN488302|07/24/2007 01:00:...| 082XX S TALMAN AVE|0460| BATTERY| SIMPLE| STREET| false| false|0835| 008| 18| 70| 08B| 1160134| 1850078| 2007|04/15/2016 08:55:...|41.744314668|-87.688830696|(41.744314668, -8...| null|2007/12|

|5679864| HN487195|07/24/2007 02:20:...|034XX N MILWAUKEE...|0486| BATTERY|DOMESTIC BATTERY ...| RESIDENCE| false| true|1731| 017| 30| 16| 08B| 1148204| 1922490| 2007|04/15/2016 08:55:...|41.943259605|-87.730682304|(41.943259605, -8...| null|2007/12|

|5679865| HN484199|07/21/2007 12:30:...| 035XX W BELMONT AVE|0820| THEFT| $500 AND UNDER| STREET| false| false|1412| 014| 35| 21| 06| 1152064| 1921048| 2007|04/15/2016 08:55:...| 41.9392273|-87.716532899|(41.9392273, -87....| null|2007/12|

|5679866| HN489706|07/26/2007 03:00:...| 048XX W ARGYLE ST|1305| CRIMINAL DAMAGE| CRIMINAL DEFACEMENT| RESIDENCE| false| false|1623| 016| 45| 12| 14| 1143169| 1932807| 2007|04/15/2016 08:55:...|41.971665985| -87.74893007|(41.971665985, -8...| null|2007/12|

|5679868| HN479171|07/20/2007 11:40:...|023XX S CALIFORNI...|1320| CRIMINAL DAMAGE| TO VEHICLE| STREET| false| false|1033| 010| 12| 30| 14| 1158058| 1888522| 2007|04/15/2016 08:55:...|41.849852628|-87.695392076|(41.849852628, -8...| null|2007/12|

|5679870| HN487786|07/25/2007 10:20:...| 001XX E 71ST ST|0890| THEFT| FROM BUILDING| GROCERY FOOD STORE| false| false|0323| 003| 6| 69| 06| 1178576| 1857978| 2007|04/15/2016 08:55:...|41.765594255| -87.62101766|(41.765594255, -8...| null|2007/12|

|5679872| HN489597|07/26/2007 03:00:...|024XX N FRANCISCO...|0810| THEFT| OVER $500| APARTMENT| false| false|1414| 014| 35| 22| 06| 1156634| 1915948| 2007|04/15/2016 08:55:...|41.925141071|-87.699875328|(41.925141071, -8...| null|2007/12|

|5679873| HN488796|07/25/2007 06:50:...| 098XX S PARNELL AVE|143A|WEAPONS VIOLATION|UNLAWFUL POSS OF ...| PARK PROPERTY| true| false|2223| 022| 21| 73| 15| 1174363| 1839821| 2007|04/15/2016 08:55:...|41.715863801|-87.636998103|(41.715863801, -8...| null|2007/12|

|5679874| HN486104|07/24/2007 11:30:...| 017XX W PIERCE AVE|0620| BURGLARY| UNLAWFUL ENTRY| APARTMENT| false| false|1433| 014| 1| 24| 05| 1164550| 1910426| 2007|04/15/2016 08:55:...|41.909824195|-87.670945023|(41.909824195, -8...| null|2007/12|

|5679875| HN485891|07/22/2007 12:05:...| 016XX N WESTERN AVE|0810| THEFT| OVER $500| STREET| false| false|1434| 014| 1| 24| 06| 1160114| 1910738| 2007|04/15/2016 08:55:...|41.910773194|-87.687232407|(41.910773194, -8...| null|2007/12|

|5679876| HN488180|07/25/2007 02:10:...|035XX W FULLERTON...|0460| BATTERY| SIMPLE| SIDEWALK| false| false|1413| 014| 26| 22| 08B| 1152506| 1915731| 2007|04/15/2016 08:55:...|41.924628284| -87.71504934|(41.924628284, -8...| null|2007/12|

|5679877| HN485606|07/23/2007 09:00:...|015XX N ARTESIAN AVE|0820| THEFT| $500 AND UNDER| STREET| false| false|1423| 014| 1| 24| 06| 1159792| 1910263| 2007|04/15/2016 08:55:...|41.909476407| -87.68842844|(41.909476407, -8...| null|2007/12|

|5679878| HN459531|04/23/2007 12:01:...|026XX N SACRAMENT...|2825| OTHER OFFENSE|HARASSMENT BY TEL...| APARTMENT| false| true|1411| 014| 35| 22| 26| 1155923| 1917476| 2007|04/15/2016 08:55:...|41.929348403| -87.70244657|(41.929348403, -8...| null|2007/12|

|5679879| HN485432|07/23/2007 08:00:...| 030XX W PALMER BLVD|0820| THEFT| $500 AND UNDER| STREET| false| false|1414| 014| 35| 22| 06| 1155945| 1914325| 2007|04/15/2016 08:55:...|41.920701371|-87.702450917|(41.920701371, -8...| null|2007/12|

|5679881| HN486824|07/23/2007 07:00:...| 010XX N STATE ST|0820| THEFT| $500 AND UNDER| SMALL RETAIL STORE| false| false|1824| 018| 42| 8| 06| 1176172| 1907536| 2007|04/15/2016 08:55:...|41.901639684| -87.62833826|(41.901639684, -8...| null|2007/12|

|5679882| HN485055|07/23/2007 10:00:...|082XX S DR MARTIN...|0430| BATTERY|AGGRAVATED: OTHER...| STREET| false| false|0631| 006| 6| 44| 04B| 1180332| 1850554| 2007|04/15/2016 08:55:...|41.745181902|-87.614808669|(41.745181902, -8...| null|2007/12|

|5679885| HN477072|07/19/2007 11:00:...| 003XX W OAK ST|2027| NARCOTICS| POSS: CRACK| CHA APARTMENT| true| false|1823| 018| 27| 8| 18| 1173484| 1907095| 2007|04/15/2016 08:55:...|41.900489748|-87.638224621|(41.900489748, -8...| null|2007/12|

|5679887| HN488976|07/25/2007 08:35:...| 050XX W GLADYS AVE|0460| BATTERY| SIMPLE| SIDEWALK| false| false|1533| 015| 29| 25| 08B| 1142579| 1897865| 2007|04/15/2016 08:55:...|41.875792556|-87.751970712|(41.875792556, -8...| null|2007/12|

|5679888| HN489702|07/25/2007 04:00:...|044XX S SPAULDING...|2825| OTHER OFFENSE|HARASSMENT BY TEL...| APARTMENT| false| true|0821| 008| 14| 58| 26| 1155096| 1875110| 2007|04/15/2016 08:55:...|41.813108297|-87.706622217|(41.813108297, -8...| null|2007/12|

+-------+-----------+--------------------+--------------------+----+-----------------+--------------------+--------------------+------+--------+----+--------+----+---------+----+--------+-----------+-----------+--------------------+------------+-------------+--------------------+--------+-------+

only showing top 20 rows

>>> crimeFile1.groupBy('Primary\_Type', 'Month').agg(f.count('Primary\_Type').alias('crime\_count')).orderBy('Month', crimeFile1.Primary\_Type.desc()).show()

+--------------------+------+-----------+

| Primary\_Type| Month|crime\_count|

+--------------------+------+-----------+

| WEAPONS VIOLATION|200112| 4274|

| THEFT|200112| 99264|

| STALKING|200112| 203|

| SEX OFFENSE|200112| 2229|

| ROBBERY|200112| 18441|

| RITUALISM|200112| 8|

|PUBLIC PEACE VIOL...|200112| 2750|

| PUBLIC INDECENCY|200112| 9|

| PROSTITUTION|200112| 6026|

| OTHER OFFENSE|200112| 29656|

|OTHER NARCOTIC VI...|200112| 6|

|OFFENSE INVOLVING...|200112| 2237|

| OBSCENITY|200112| 19|

| NARCOTICS|200112| 50567|

| MOTOR VEHICLE THEFT|200112| 27549|

|LIQUOR LAW VIOLATION|200112| 1637|

| KIDNAPPING|200112| 933|

| INTIMIDATION|200112| 279|

|INTERFERENCE WITH...|200112| 406|

| HOMICIDE|200112| 667|

+--------------------+------+-----------+

Scenario 2:

Details - Duration 15 to 20 minutes

* Data is available in local file system /data/retail\_db
* Source directories: /data/retail\_db/orders and /data/retail\_db/customers
* Source delimiter: comma (“,”)
* Source Columns - orders - order\_id, order\_date, order\_customer\_id, order\_status
* Source Columns - customers - customer\_id, customer\_fname, customer\_lname and many more
* Get the customers who have not placed any orders, sorted by customer\_lname and then customer\_fname
* Target Columns: customer\_lname, customer\_fname
* Number of files - 1
* Target Directory: /user/<YOUR\_USER\_ID>/solutions/solutions02/inactive\_customers
* Target File Format: TEXT
* Target Delimiter: comma (“, ”)
* Compression: N/A

>>> orders = sc.textFile("/user/saurabhbhanwala/retail\_db/orders")

>>> customer = sc.textFile("/user/saurabhbhanwala/retail\_db/customers")

>>> for i in orders.take(10): print(i)

...

1,2013-07-25 00:00:00.0,11599,CLOSED

2,2013-07-25 00:00:00.0,256,PENDING\_PAYMENT

3,2013-07-25 00:00:00.0,12111,COMPLETE

4,2013-07-25 00:00:00.0,8827,CLOSED

5,2013-07-25 00:00:00.0,11318,COMPLETE

6,2013-07-25 00:00:00.0,7130,COMPLETE

7,2013-07-25 00:00:00.0,4530,COMPLETE

8,2013-07-25 00:00:00.0,2911,PROCESSING

9,2013-07-25 00:00:00.0,5657,PENDING\_PAYMENT

10,2013-07-25 00:00:00.0,5648,PENDING\_PAYMENT

>>> for i in customers.take(10): print(i)

...

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'customers' is not defined

>>> for i in customer.take(10): print(i)

...

1,Richard,Hernandez,XXXXXXXXX,XXXXXXXXX,6303 Heather Plaza,Brownsville,TX,78521

2,Mary,Barrett,XXXXXXXXX,XXXXXXXXX,9526 Noble Embers Ridge,Littleton,CO,80126

3,Ann,Smith,XXXXXXXXX,XXXXXXXXX,3422 Blue Pioneer Bend,Caguas,PR,00725

4,Mary,Jones,XXXXXXXXX,XXXXXXXXX,8324 Little Common,San Marcos,CA,92069

5,Robert,Hudson,XXXXXXXXX,XXXXXXXXX,"10 Crystal River Mall ",Caguas,PR,00725

6,Mary,Smith,XXXXXXXXX,XXXXXXXXX,3151 Sleepy Quail Promenade,Passaic,NJ,07055

7,Melissa,Wilcox,XXXXXXXXX,XXXXXXXXX,9453 High Concession,Caguas,PR,00725

8,Megan,Smith,XXXXXXXXX,XXXXXXXXX,3047 Foggy Forest Plaza,Lawrence,MA,01841

9,Mary,Perez,XXXXXXXXX,XXXXXXXXX,3616 Quaking Street,Caguas,PR,00725

10,Melissa,Smith,XXXXXXXXX,XXXXXXXXX,8598 Harvest Beacon Plaza,Stafford,VA,22554

>>> ordersmap=orders.map(lambda o: (int(o.split(",")[0]), int(o.split(",")[2])))

>>> for i in ordersmap.take(10): print(i)

...

(1, 11599)

(2, 256)

(3, 12111)

(4, 8827)

(5, 11318)

(6, 7130)

(7, 4530)

(8, 2911)

(9, 5657)

(10, 5648)

>>> customersmap = customer.map(lambda c: (int(c.split(",")[0]), (c.split(",")[1], c.split(",")[2])))

>>> for i in customersmap.take(10): print(i)

...

(1, (u'Richard', u'Hernandez'))

(2, (u'Mary', u'Barrett'))

(3, (u'Ann', u'Smith'))

(4, (u'Mary', u'Jones'))

(5, (u'Robert', u'Hudson'))

(6, (u'Mary', u'Smith'))

(7, (u'Melissa', u'Wilcox'))

(8, (u'Megan', u'Smith'))

(9, (u'Mary', u'Perez'))

(10, (u'Melissa', u'Smith'))

>>>

>>> orders = sc.textFile("/user/saurabhbhanwala/retail\_db/orders")

>>> customer = sc.textFile("/user/saurabhbhanwala/retail\_db/customers")

>>> ordersMap = orders.map(lambda o: (int(o.split(",")[2]), 1))

>>> customerMap = customer.map(lambda c: (int(c.split(",")[0]), (c.split(",")[1], c.split(",")[2])))

>>> customerOrderJoin = customerMap.leftOuterJoin(ordersMap)

>>> for i in customerOrderjoin.take(10): print(i)

...

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'customerOrderjoin' is not defined

>>> for i in customerOrderJoin.take(10): print(i)

...

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

(8192, ((u'Kathryn', u'Marshall'), 1))

>>> inActiveCustomers = customerOrderJoin.filter(lambda cust: cust[1][1] == None)

>>> for i in inActiveCustomers.take(10): print(i)

...

>>> for i in inActiveCustomers.take(10): print(i)

...

(12392, ((u'Alan', u'Wolf'), None))

(1808, ((u'Albert', u'Ellison'), None))

(2096, ((u'Jose', u'Tanner'), None))

(10060, ((u'Mary', u'Shaw'), None))

(9060, ((u'Matthew', u'Patel'), None))

(6072, ((u'Stephen', u'Smith'), None))

(7552, ((u'Carl', u'Smith'), None))

(469, ((u'Randy', u'Smith'), None))

(1481, ((u'Grace', u'Smith'), None))

(2073, ((u'Donna', u'Stephens'), None))

>>> CustomersNoOrders = inActiveCustomers.map(lambda co: co[1][0][0] + "," + co[1][0][1])

>>> for i in CustomersNoOrders.take(10): print(i)

...

Alan,Wolf

Albert,Ellison

Mary,Shaw

Jose,Tanner

Matthew,Patel

Stephen,Smith

Carl,Smith

Randy,Smith

Grace,Smith

Donna,Stephens

CustomersNoOrders.saveAsTextFile("/user/saurabhbhanwala/solutions/solution02/inactive\_orders")

>>> os.system("hdfs dfs -ls /user/saurabhbhanwala/solutions/solution02/inactive\_orders")

Found 5 items

-rw-r--r-- 2 saurabhbhanwala hdfs 0 2020-09-21 03:33 /user/saurabhbhanwala/solutions/solution02/inactive\_orders/\_SUCCESS

-rw-r--r-- 2 saurabhbhanwala hdfs 86 2020-09-21 03:33 /user/saurabhbhanwala/solutions/solution02/inactive\_orders/part-00000

-rw-r--r-- 2 saurabhbhanwala hdfs 66 2020-09-21 03:33 /user/saurabhbhanwala/solutions/solution02/inactive\_orders/part-00001

-rw-r--r-- 2 saurabhbhanwala hdfs 70 2020-09-21 03:33 /user/saurabhbhanwala/solutions/solution02/inactive\_orders/part-00002

-rw-r--r-- 2 saurabhbhanwala hdfs 150 2020-09-21 03:33 /user/saurabhbhanwala/solutions/solution02/inactive\_orders/part-00003

0

>>> os.system("hdfs dfs -tail /user/saurabhbhanwala/solutions/solution02/inactive\_orders/part-00000")

Alan,Wolf

Albert,Ellison

Mary,Shaw

Jose,Tanner

Matthew,Patel

Stephen,Smith

Carl,Smith

0

>>> os.system("hdfs dfs -ls /user/saurabhbhanwala/solutions/solution02/inactive\_ordersSingle")

Found 2 items

-rw-r--r-- 2 saurabhbhanwala hdfs 0 2020-09-21 03:39 /user/saurabhbhanwala/solutions/solution02/inactive\_ordersSingle/\_SUCCESS

-rw-r--r-- 2 saurabhbhanwala hdfs 372 2020-09-21 03:39 /user/saurabhbhanwala/solutions/solution02/inactive\_ordersSingle/part-00000

0

====

With data frame and SQL

>>> orders = sc.textFile("/user/saurabhbhanwala/retail\_db/orders")

>>> customers = sc.textFile("/user/saurabhbhanwala/retail\_db/customers")

>>> from pyspark.sql import Row

>>> ordersDF = orders.map(lambda o: Row(orders\_customer\_order = int(o.split(",")[2]))).toDF

>>> customersDF = customers.map(lambda c:Row(customer\_id = int(c.split(",")[0]), lname = (c.split(",")[2]), fname = (c.split(",")[1]))).toDF()

>>> ordersDF = orders.map(lambda o: Row(orders\_customer\_order = int(o.split(",")[2]))).toDF()

>>> ordersDF.registerTempTable("orders\_DF")

>>> customersDF.registerTempTable("customers\_DF")

>>> sqlContext.setConf("spark.sql.shuffle.partitions", "1")

>>> sqlContext.sql("use saurabhbhanwala")

DataFrame[]

>>> sqlContext.sql("show tables").show(truncate=False)

+---------------+--------------+-----------+

|database |tableName |isTemporary|

+---------------+--------------+-----------+

|saurabhbhanwala|categories |false |

|saurabhbhanwala|customers |false |

|saurabhbhanwala|department |false |

|saurabhbhanwala|employee |false |

|saurabhbhanwala|order\_items |false |

|saurabhbhanwala|orders |false |

|saurabhbhanwala|partition\_user|false |

|saurabhbhanwala|products |false |

|saurabhbhanwala|weather |false |

| |customers\_df |true |

| |orders\_df |true |

+---------------+--------------+-----------+

>>> sqlContext.sql("select \* from orders\_DF limit 5").show(truncate=False)

+---------------------+

|orders\_customer\_order|

+---------------------+

|11599 |

|256 |

|12111 |

|8827 |

|11318 |

+---------------------+

>>> sqlContext.sql("select \* from customers\_DF limit 5").show(truncate=False)

+-----------+-------+---------+

|customer\_id|fname |lname |

+-----------+-------+---------+

|1 |Richard|Hernandez|

|2 |Mary |Barrett |

|3 |Ann |Smith |

|4 |Mary |Jones |

|5 |Robert |Hudson |

+-----------+-------+---------+

>>> CustomersNotActive = sqlContext.sql("select fname,lname from customers\_DF cd left outer join orders\_DF od on cd.customer\_id = od.orders\_customer\_order where od.orders\_customer\_order is null order by fname,lname")

>>> CustomersNotActive.show()

+-------+--------+

| fname| lname|

+-------+--------+

| Alan| Wolf|

| Albert| Ellison|

| Amanda| Smith|

| Ashley| Smith|

| Carl| Smith|

|Carolyn| Green|

| Donna|Stephens|

|Dorothy| Vazquez|

| Emma| Smith|

| Gary| Walker|

| Grace| Smith|

| James| Smith|

| Joan| Smith|

| Jose| Tanner|

|Kenneth| Smith|

| Kevin| Smith|

| Mary| Bolton|

| Mary| Greene|

| Mary| Harrell|

| Mary| Lewis|

+-------+--------+

only showing top 20 rows

>>> CustomersNotActiveFormatted = CustomersNotActive.rdd.map(lambda cn: cn[0] + "," + cn[1])

>>> CustomersNotActiveFormatted.coalesce(1).saveAsTextFile("/user/saurabhbhanwala/solutions/solution02/inactive\_usersSQL")

>>>