Week 5-8 : Graded Assignment Part 1 - Analysing IoT Data with Spark Sql

Submitted by: Jophy Joseph August 07, 2022

In [1]:
```
!sudo apt update
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
!wget -q https://dlcdn.apache.org/spark/spark-3.3.0/spark-3.3.0-bin-hadoop3.tgz
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease [3,626 B]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:3 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic InRelease
Ign:5 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64  InRelease
Get:6 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64  InRelease [1,581 B]
Hit:7 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64  Release
Get:8 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:9 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64  Packages [902 kB]
Hit:11 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Hit:12 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Get:13 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2,905 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Hit:15 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [3,336 kB]
Get:17 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1,528 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2,306 kB]
Fetched 11.2 MB in 4s (2,923 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
25 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

In [2]:
```
!tar xf spark-3.3.0-bin-hadoop3.tgz
!pip install -q findspark
!pip install pyspark
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyspark
  Downloading pyspark-3.3.0.tar.gz (281.3 MB)
     |████████████████████████████████| 281.3 MB 48 kB/s
Collecting py4j==0.10.9.5
  Downloading py4j-0.10.9.5-py2.py3-none-any.whl (199 kB)
     |████████████████████████████████| 199 kB 49.0 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.3.0-py2.py3-none-any.whl size=281764026 sha256=1252ffb50a5c44b453f4cdb66a9fba3f5fef5162d0936ba9a307d0e9df73cab7
  Stored in directory: /root/.cache/pip/wheels/7a/8e/1b/f73a52650d2e5f337708d9f6a1750d451a7349a867f928b885
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.5 pyspark-3.3.0
```

In [3]:
```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.3.0-bin-hadoop3"
```

In [4]:
```
import findspark
findspark.init()
findspark.find()
```

Out[4]:
```
'/content/spark-3.3.0-bin-hadoop3'
```

In [5]:
```
from pyspark.sql import DataFrame, SparkSession
from typing import List
import pyspark.sql.types as T
import pyspark.sql.functions as F


spark = SparkSession \
       .builder \
       .appName("Part-1: Working with SparkSQL") \
       .getOrCreate()

spark
```

Out[5]: **SparkSession - in-memory**

SparkContext

Spark UI

Version	`v3.3.0`
Master	`local[*]`
AppName	`Part-1: Working with SparkSQL`

Task 1: Read the data into a Dataframe.

In [6]:
```python
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [7]:
```python
file_location = "/content/drive/My Drive/data/iot_devices.json"
file_type = "json"

# CSV options
infer_schema = True
first_row_is_header = True
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
iot_df = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)
```

In [8]:
```python
iot_df.show(20, False)
```

```
+-------------+---------+----+----+-------------+---------+------------------------+--------+---------------+--------+------+---------+-------+----+-------------
+
|battery_level|c02_level|cca2|cca3|cn           |device_id|device_name             |humidity|ip             |latitude|lcd   |longitude|scale  |temp|timestamp    
|
+-------------+---------+----+----+-------------+---------+------------------------+--------+---------------+--------+------+---------+-------+----+-------------
+
|8            |868      |US  |USA |United States|1        |meter-gauge-1xbYRYcj    |51      |68.161.225.1   |38.0    |green |-97.0    |Celsius|34  |1458444054093
|
|7            |1473     |NO  |NOR |Norway       |2        |sensor-pad-2n2Pea       |70      |213.161.254.1  |62.47   |red   |6.15     |Celsius|11  |1458444054119
|
|2            |1556     |IT  |ITA |Italy        |3        |device-mac-36TWSKiT     |44      |88.36.5.1      |42.83   |red   |12.83    |Celsius|19  |1458444054120
|
|6            |1080     |US  |USA |United States|4        |sensor-pad-4mzWkz       |32      |66.39.173.154  |44.06   |yellow|-121.32  |Celsius|28  |1458444054121
|
|4            |931      |PH  |PHL |Philippines  |5        |therm-stick-5gimpUrBB   |62      |203.82.41.9    |14.58   |green |120.97   |Celsius|25  |1458444054122
|
|3            |1210     |US  |USA |United States|6        |sensor-pad-6al7RTAobR   |51      |204.116.105.67 |35.93   |yellow|-85.46   |Celsius|27  |1458444054122
|
|3            |1129     |CN  |CHN |China        |7        |meter-gauge-7GeDoanM    |26      |220.173.179.1  |22.82   |yellow|108.32   |Celsius|18  |1458444054123
|
|0            |1536     |JP  |JPN |Japan        |8        |sensor-pad-8xUD6pzsQI   |35      |210.173.177.1  |35.69   |red   |139.69   |Celsius|27  |1458444054123
|
|3            |807      |JP  |JPN |Japan        |9        |device-mac-9GcjZ2pw     |85      |118.23.68.227  |35.69   |green |139.69   |Celsius|13  |1458444054124
```

```
|
|7            |1470     |US  |USA |United States|10       |sensor-pad-10BsywSYUF   |56      |208.109.163.218|33.61   |red   |-111.89  |Celsius|26  |1458444054125
|
|3            |1544     |IT  |ITA |Italy        |11       |meter-gauge-11dlMTZty   |85      |88.213.191.34  |42.83   |red   |12.83    |Celsius|16  |1458444054125
|
|0            |1260     |US  |USA |United States|12       |sensor-pad-12Y2kIm0o    |92      |68.28.91.22    |38.0    |yellow|-97.0    |Celsius|12  |1458444054126
|
|6            |1007     |IN  |IND |India        |13       |meter-gauge-13GrojanSGBz|92      |59.144.114.250 |28.6    |yellow|77.2     |Celsius|13  |1458444054127
|
|1            |1346     |NO  |NOR |Norway       |14       |sensor-pad-14QL93sBR0j  |90      |193.156.90.200 |59.95   |yellow|10.75    |Celsius|16  |1458444054127
|
|9            |1259     |US  |USA |United States|15       |device-mac-15se6mZ      |70      |67.185.72.1    |47.41   |yellow|-122.0   |Celsius|13  |1458444054128
|
|4            |1425     |US  |USA |United States|16       |sensor-pad-16aXmIJZtdO  |53      |68.85.85.106   |38.0    |red   |-97.0    |Celsius|15  |1458444054128
|
|0            |1466     |US  |USA |United States|17       |meter-gauge-17zb8Fghhl  |98      |161.188.212.254|39.95   |red   |-75.16   |Celsius|31  |1458444054129
|
|4            |1096     |CN  |CHN |China        |18       |sensor-pad-18XULN9Xv    |25      |221.3.128.242  |25.04   |yellow|102.72   |Celsius|31  |1458444054130
|
|9            |1531     |US  |USA |United States|19       |meter-gauge-19eg1BpfCO  |75      |64.124.180.215 |38.0    |red   |-97.0    |Celsius|29  |1458444054130
|
|7            |1155     |US  |USA |United States|20       |sensor-pad-20gFNfBgqr   |33      |66.153.162.66  |33.94   |yellow|-78.92   |Celsius|10  |1458444054131
|
+-------------+---------+----+----+-------------+---------+------------------------+--------+---------------+--------+------+---------+-------+----+-------------
+
only showing top 20 rows
```

Task 2. Convert the Dataframe into a temporary view called iot.

In [9]:

```python
iot_df.createOrReplaceTempView('iot')
```

In [10]:

```python
spark.sql("select * from iot").show(5)
```

```
+-------------+---------+----+----+-------------+---------+--------------------+--------+---------------+--------+------+---------+-------+----+-------------+
|battery_level|c02_level|cca2|cca3|           cn|device_id|         device_name|humidity|             ip|latitude|   lcd|longitude|  scale|temp|    timestamp|
+-------------+---------+----+----+-------------+---------+--------------------+--------+---------------+--------+------+---------+-------+----+-------------+
|            8|      868|  US| USA|United States|        1|meter-gauge-1xbYRYcj|      51|   68.161.225.1|    38.0| green|    -97.0|Celsius|  34|1458444054093|
|            7|     1473|  NO| NOR|       Norway|        2|   sensor-pad-2n2Pea|      70|  213.161.254.1|   62.47|   red|     6.15|Celsius|  11|1458444054119|
|            2|     1556|  IT| ITA|        Italy|        3| device-mac-36TWSKiT|      44|      88.36.5.1|   42.83|   red|    12.83|Celsius|  19|1458444054120|
|            6|     1080|  US| USA|United States|        4|   sensor-pad-4mzWkz|      32|  66.39.173.154|   44.06|yellow|  -121.32|Celsius|  28|1458444054121|
|            4|      931|  PH| PHL|  Philippines|        5|therm-stick-5gimp...|      62|    203.82.41.9|   14.58| green|   120.97|Celsius|  25|1458444054122|
+-------------+---------+----+----+-------------+---------+--------------------+--------+---------------+--------+------+---------+-------+----+-------------+
only showing top 5 rows
```

```
In [11]: iot_df.printSchema()
```

```
root
 |-- battery_level: long (nullable = true)
 |-- c02_level: long (nullable = true)
 |-- cca2: string (nullable = true)
 |-- cca3: string (nullable = true)
 |-- cn: string (nullable = true)
 |-- device_id: long (nullable = true)
 |-- device_name: string (nullable = true)
 |-- humidity: long (nullable = true)
 |-- ip: string (nullable = true)
 |-- latitude: double (nullable = true)
 |-- lcd: string (nullable = true)
 |-- longitude: double (nullable = true)
 |-- scale: string (nullable = true)
 |-- temp: long (nullable = true)
 |-- timestamp: long (nullable = true)
```

Task 3. Count how many devices are there from each country and display the output.

```
In [12]: spark.sql("select cca3 as Country_name, count(distinct(device_id)) as Number_of_devices from iot group by cca3 order by Number_of_devices desc").show()
```

```
+------------+-----------------+
|Country_name|Number_of_devices|
+------------+-----------------+
|         USA|            70405|
|         CHN|            14455|
|         JPN|            12100|
|         KOR|            11879|
|         DEU|             7942|
|         GBR|             6486|
|         CAN|             6041|
|         RUS|             5989|
|         FRA|             5305|
|         BRA|             3224|
|         AUS|             3119|
|         ITA|             2915|
|         SWE|             2880|
|         POL|             2744|
|         NLD|             2488|
|         ESP|             2310|
|         TWN|             2128|
|         IND|             1867|
|         CZE|             1507|
|         NOR|             1487|
+------------+-----------------+
only showing top 20 rows
```

Task 4. Display all the countries whose carbon dioxide level is more than 1400. Sort the output in descending order.

In [13]:
```
spark.sql("select Cca2 as Country_Code, Cca3 as Country_name, sum(c02_level) as CO2_level  from iot  where c02_level > 1400  group by Cca2, Cca3 order by CO2_lev
```

```
+------------+------------+---------+
|Country_Code|Country_name|CO2_level|
+------------+------------+---------+
|          US|         USA| 26242891|
|          CN|         CHN|  5424312|
|          KR|         KOR|  4415118|
|          JP|         JPN|  4399107|
|          DE|         DEU|  2950796|
|          GB|         GBR|  2488574|
|          CA|         CAN|  2343270|
|          RU|         RUS|  2262936|
|          FR|         FRA|  2030583|
|          BR|         BRA|  1284892|
|          AU|         AUS|  1153899|
|          SE|         SWE|  1086146|
|          IT|         ITA|  1070505|
|          PL|         POL|   995721|
|          NL|         NLD|   970297|
|          ES|         ESP|   878143|
|          TW|         TWN|   813253|
|          IN|         IND|   666101|
|          NO|         NOR|   598248|
|          UA|         UKR|   559605|
+------------+------------+---------+
only showing top 20 rows
```

Task 5. Select all countries' devices with high-levels of C02 and group by cca3 and order by device_ids (Hint: For high CO2 level, the LCD status will be RED).

In [14]:
```
spark.sql("select cca3,cn,device_id from iot where lcd = 'red' group by cca3,cn,device_id order by device_id ").show()
```

```
+----+----------------+---------+
|cca3|              cn|device_id|
+----+----------------+---------+
| NOR|          Norway|        2|
| ITA|           Italy|        3|
| JPN|           Japan|        8|
| USA|   United States|       10|
| ITA|           Italy|       11|
| USA|   United States|       16|
| USA|    United States|       17|
| USA|    United States|       19|
| JPN|            Japan|       22|
| CAN|           Canada|       24|
| KOR|Republic of Korea|       27|
| KOR|Republic of Korea|       28|
| UKR|          Ukraine|       47|
| SWE|           Sweden|       53|
| USA|    United States|       54|
| USA|    United States|       57|
| USA|    United States|       64|
| CZE|   Czech Republic|       66|
| IND|            India|       77|
| KOR|Republic of Korea|       78|
+----+-----------------+---------+
only showing top 20 rows
```

Task 6. Find out all devices in countries whose batteries need replacements.

In [15]:
```
# Assumption - devices with battery level < 3 would need replacement
spark.sql("select cca3, device_id, battery_level from iot where battery_level < 3  order by battery_level ").show()
```

```
+----+---------+-------------+
|cca3|device_id|battery_level|
+----+---------+-------------+
| JPN|   106121|            0|
| DEU|   106203|            0|
| AUS|   106061|            0|
| RUS|   106125|            0|
| AUS|   106067|            0|
| VNM|   106139|            0|
| USA|   106076|            0|
| CAN|   106152|            0|
| USA|   106088|            0|
| FRA|   106162|            0|
| DEU|   106105|            0|
| USA|   106168|            0|
| GBR|   106112|            0|
| USA|   106172|            0|
| CHN|   106199|            0|
| KOR|   106177|            0|
| MYS|   106085|            0|
| ROU|   106182|            0|
| USA|   106110|            0|
| LBY|   106195|            0|
+----+---------+-------------+
only showing top 20 rows
```