MET CS 689 Designing and Implementing a Data Warehouse
Assignment 5: Semi-Structured Data Warehousing with Apache Spark

This submission template is a convenient document for you to provide the screenshots and explanations for Assignment 5.0. This submission template is intended to be used in conjunction with the Assignment 5.0 Instructions document. The instructions document illustrates how to correctly execute each SQL construct, explains important theoretical and practical details, and contains the complete set of instructions on how to complete this lab.

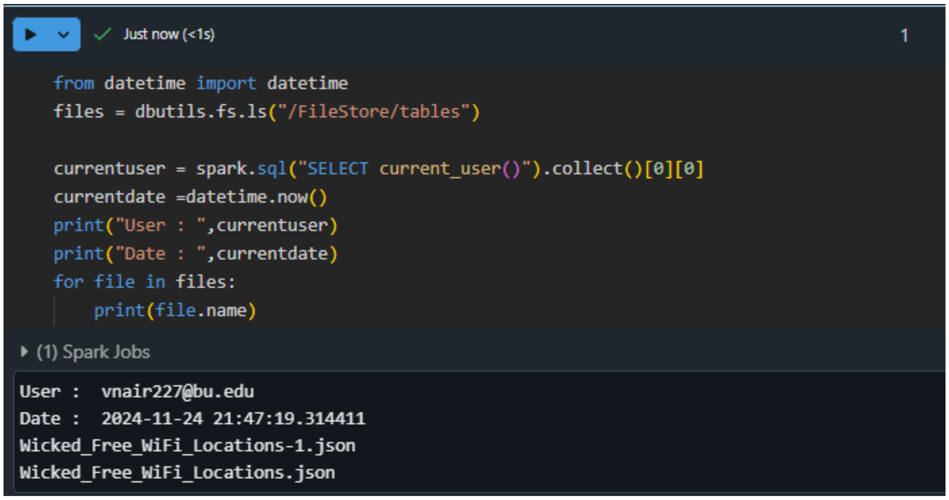
Name: Varun Nair

Date: 11/24/2024

Section Two

The screenshots needs to show your user name and the date loaded.

15. Screenshot of the loaded file.



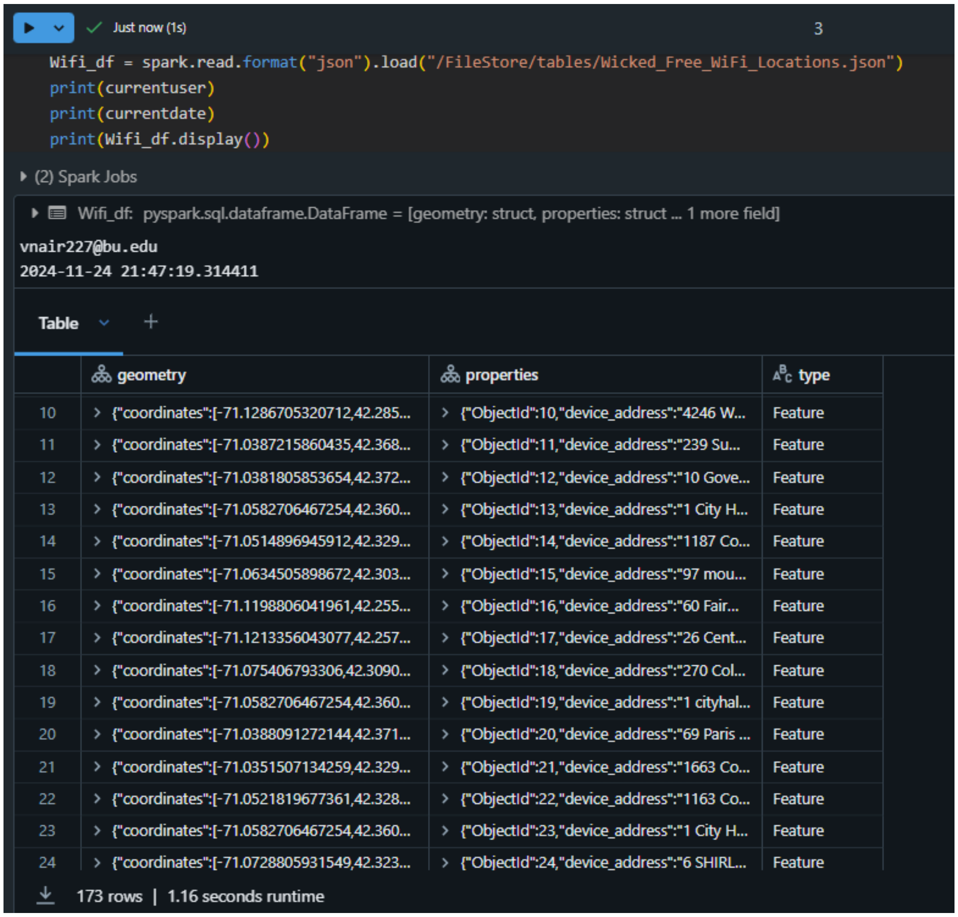
```
from datetime import datetime
files = dbutils.fs.ls("/FileStore/tables")

currentuser = spark.sql("SELECT current_user()").collect()[0][0]
currentdate =datetime.now()
print("User : ",currentuser)
print("Date : ",currentdate)
for file in files:
    print(file.name)
```

```
User :  vnair227@bu.edu
Date :  2024-11-24 21:47:19.314411
Wicked_Free_WiFi_Locations-1.json
Wicked_Free_WiFi_Locations.json
```

Copyright 2022 Boston University. All Rights Reserved.

16. Screenshot of the loaded data frame.



```
Wifi_df = spark.read.format("json").load("/FileStore/tables/Wicked_Free_WiFi_Locations.json")
print(currentuser)
print(currentdate)
print(Wifi_df.display())
```

```
Wifi_df: pyspark.sql.dataframe.DataFrame = [geometry: struct, properties: struct ... 1 more field]
vnair227@bu.edu
2024-11-24 21:47:19.314411
```

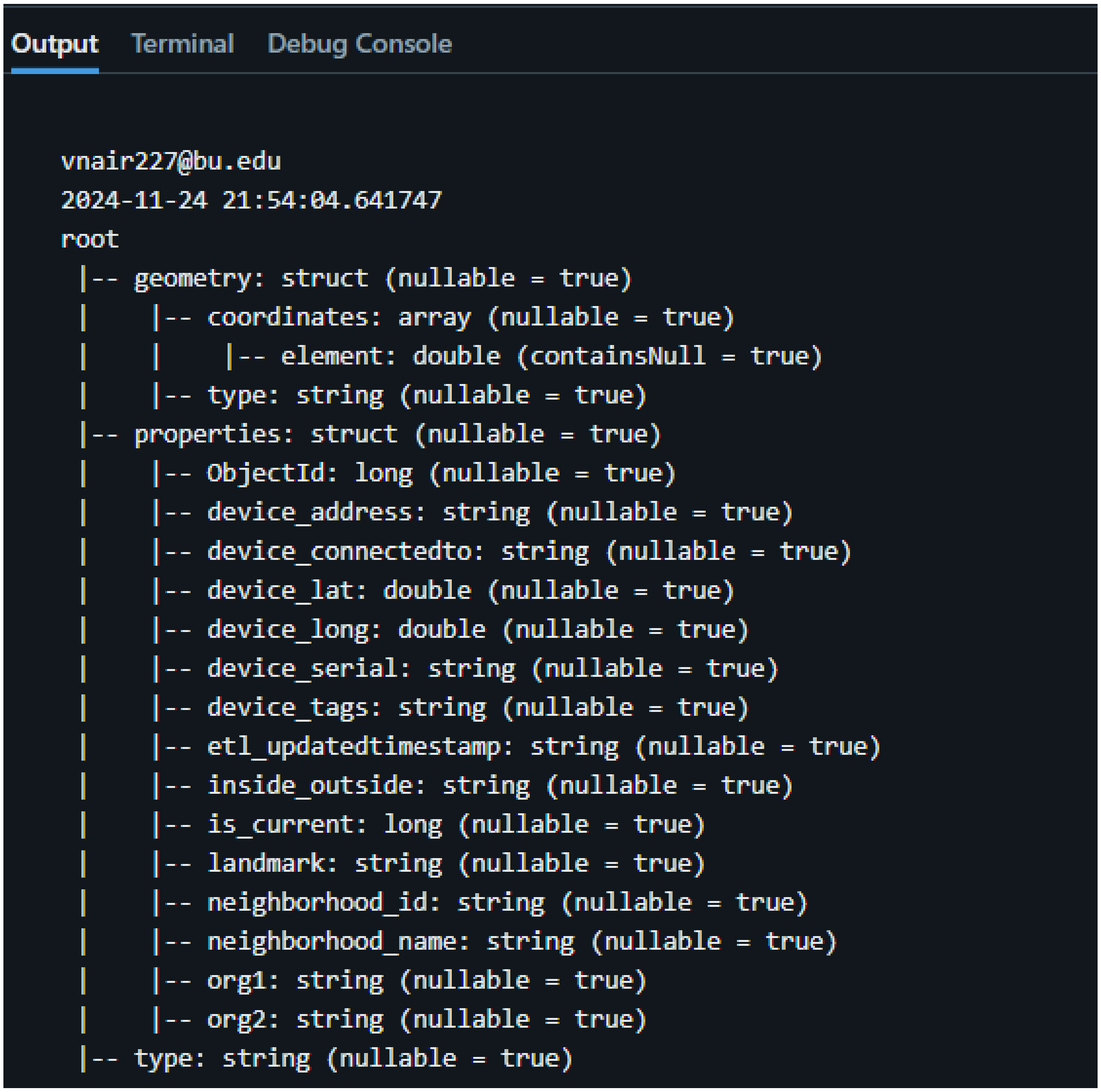
	geometry	properties	type
10	{"coordinates":[-71.1286705320712,42.285...	{"ObjectId":10,"device_address":"4246 W...	Feature
11	{"coordinates":[-71.0387215860435,42.368...	{"ObjectId":11,"device_address":"239 Su...	Feature
12	{"coordinates":[-71.0381805853654,42.372...	{"ObjectId":12,"device_address":"10 Gove...	Feature
13	{"coordinates":[-71.0582706467254,42.360...	{"ObjectId":13,"device_address":"1 City H...	Feature
14	{"coordinates":[-71.0514896945912,42.329...	{"ObjectId":14,"device_address":"1187 Co...	Feature
15	{"coordinates":[-71.0634505898672,42.303...	{"ObjectId":15,"device_address":"97 mou...	Feature
16	{"coordinates":[-71.1198806041961,42.255...	{"ObjectId":16,"device_address":"60 Fair...	Feature
17	{"coordinates":[-71.1213356043077,42.257...	{"ObjectId":17,"device_address":"26 Cent...	Feature
18	{"coordinates":[-71.075406793306,42.3090...	{"ObjectId":18,"device_address":"270 Col...	Feature
19	{"coordinates":[-71.0582706467254,42.360...	{"ObjectId":19,"device_address":"1 cityhal...	Feature
20	{"coordinates":[-71.0388091272144,42.371...	{"ObjectId":20,"device_address":"69 Paris ...	Feature
21	{"coordinates":[-71.0351507134259,42.329...	{"ObjectId":21,"device_address":"1663 Co...	Feature
22	{"coordinates":[-71.0521819677361,42.328...	{"ObjectId":22,"device_address":"1163 Co...	Feature
23	{"coordinates":[-71.0582706467254,42.360...	{"ObjectId":23,"device_address":"1 City H...	Feature
24	{"coordinates":[-71.0728805931549,42.323...	{"ObjectId":24,"device_address":"6 SHIRL...	Feature

173 rows | 1.16 seconds runtime

17. Provide the query command and the resulting data set

```
print(currentuser)
print(datetime.now())
Wifi_df.printSchema()
```

Screenshot :



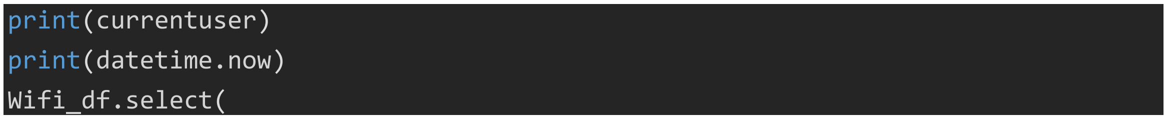
```
vnair227@bu.edu
2024-11-24 21:54:04.641747
root
 |-- geometry: struct (nullable = true)
 |    |-- coordinates: array (nullable = true)
 |    |    |-- element: double (containsNull = true)
 |    |-- type: string (nullable = true)
 |-- properties: struct (nullable = true)
 |    |-- ObjectId: long (nullable = true)
 |    |-- device_address: string (nullable = true)
 |    |-- device_connectedto: string (nullable = true)
 |    |-- device_lat: double (nullable = true)
 |    |-- device_long: double (nullable = true)
 |    |-- device_serial: string (nullable = true)
 |    |-- device_tags: string (nullable = true)
 |    |-- etl_updatedtimestamp: string (nullable = true)
 |    |-- inside_outside: string (nullable = true)
 |    |-- is_current: long (nullable = true)
 |    |-- landmark: string (nullable = true)
 |    |-- neighborhood_id: string (nullable = true)
 |    |-- neighborhood_name: string (nullable = true)
 |    |-- org1: string (nullable = true)
 |    |-- org2: string (nullable = true)
 |-- type: string (nullable = true)
```

18. Briefly describe the structure of the data frame.

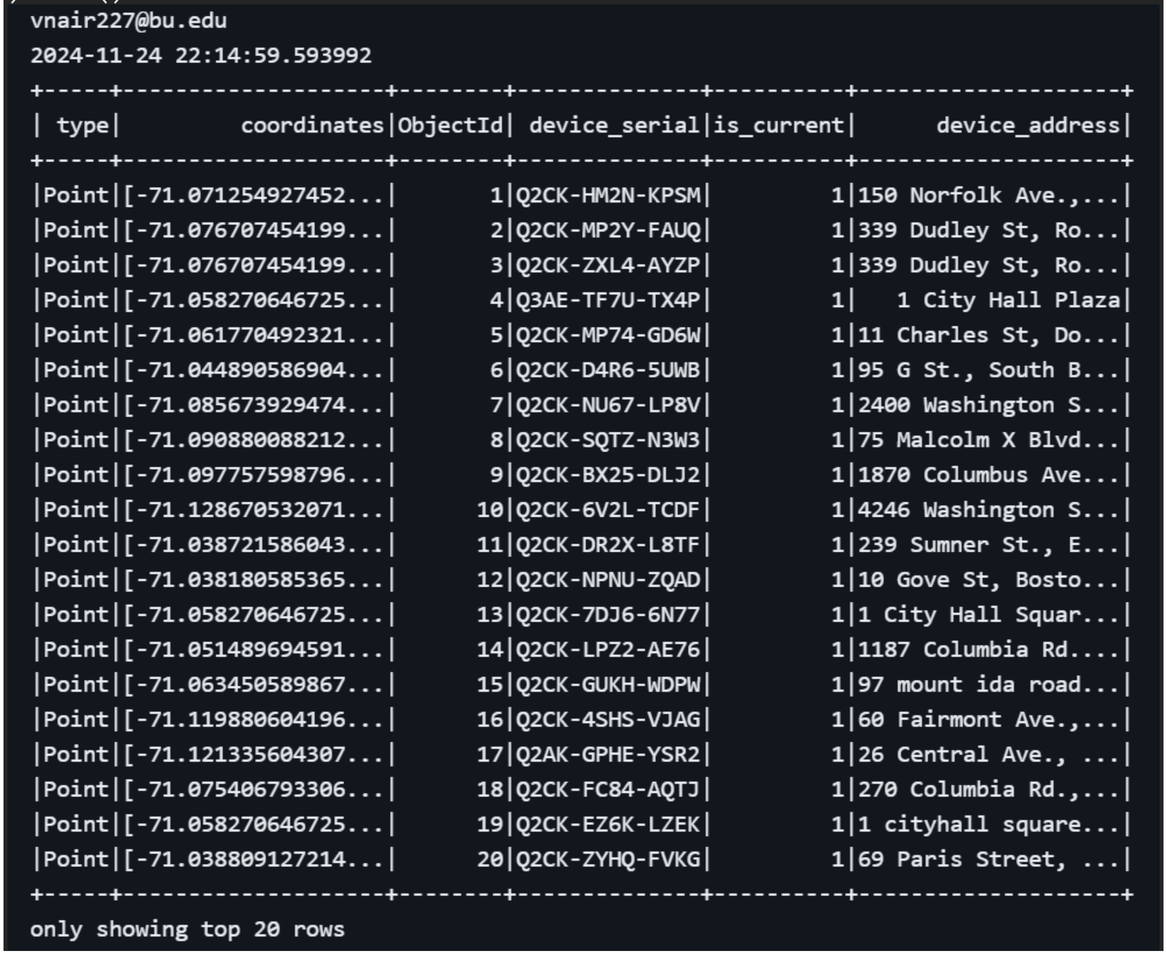
There are 3 main fields (geometry,properties and type). Each field has a list of subfields giving more details about the location.

- Geometry has the latitude and longitude in the coordinates subfield and a type subfield classifying the type of coordinate (Point)
- Properties field has several subfields having details about the location itself like address,tags,what it is connected to,landmarks,neighbourhood name and ID.
- The type field just has a String value in it calling each row a feature.

21. Provide the query command and the resulting data set



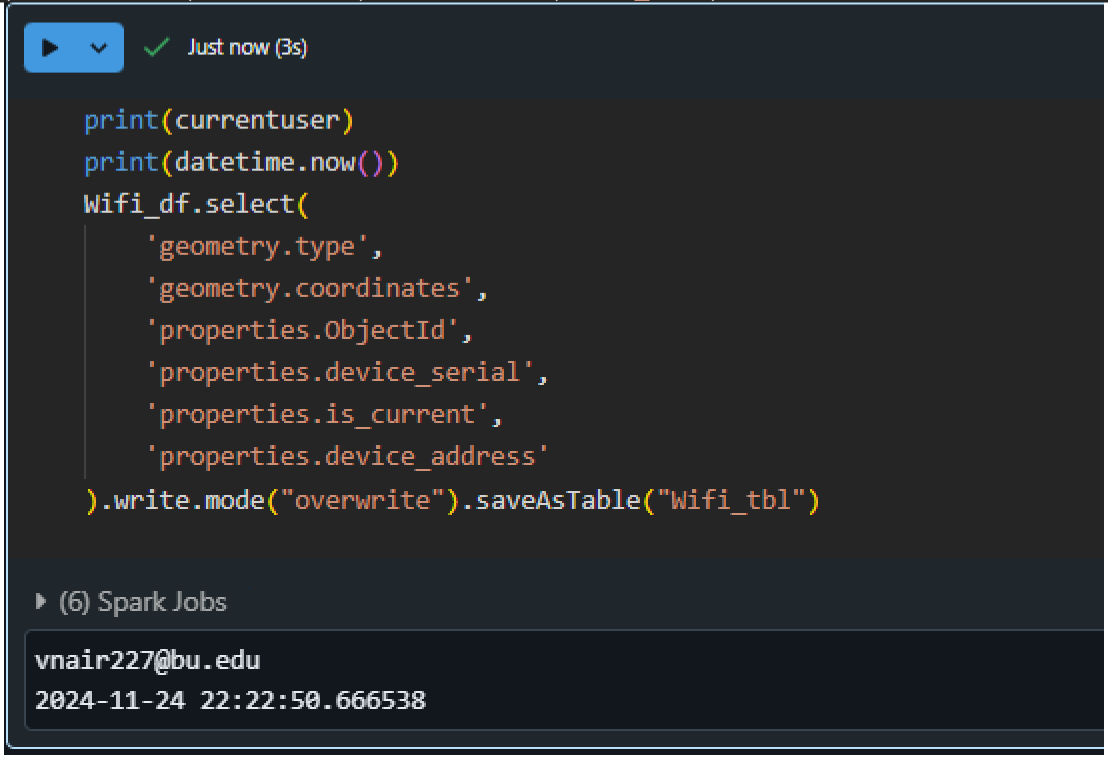
```
print(currentuser)
print(datetime.now)
Wifi_df.select(
    'geometry.type',
    'geometry.coordinates',
    'properties.ObjectId',
    'properties.device_serial',
    'properties.is_current',
    'properties.device_address'
).show()
```



```
vnair227@bu.edu
2024-11-24 22:14:59.593992
+-----+--------------------+--------+--------------+----------+--------------------+
| type|         coordinates|ObjectId| device_serial|is_current|      device_address|
+-----+--------------------+--------+--------------+----------+--------------------+
|Point|[-71.071254927452...|       1|Q2CK-HM2N-KPSM|         1|150 Norfolk Ave.,...|
|Point|[-71.076707454199...|       2|Q2CK-MP2Y-FAUQ|         1|339 Dudley St, Ro...|
|Point|[-71.076707454199...|       3|Q2CK-ZXL4-AYZP|         1|339 Dudley St, Ro...|
|Point|[-71.058270646725...|       4|Q3AE-TF7U-TX4P|         1|   1 City Hall Plaza|
|Point|[-71.061770492321...|       5|Q2CK-MP74-GD6W|         1|11 Charles St, Do...|
|Point|[-71.044890586904...|       6|Q2CK-D4R6-5UWB|         1|95 G St., South B...|
|Point|[-71.085673929474...|       7|Q2CK-NU67-LP8V|         1|2400 Washington S...|
|Point|[-71.090880088212...|       8|Q2CK-SQTZ-N3W3|         1|75 Malcolm X Blvd...|
|Point|[-71.097757598796...|       9|Q2CK-BX25-DLJ2|         1|1870 Columbus Ave...|
|Point|[-71.128670532071...|      10|Q2CK-6V2L-TCDF|         1|4246 Washington S...|
|Point|[-71.038721586043...|      11|Q2CK-DR2X-L8TF|         1|239 Sumner St., E...|
|Point|[-71.038180585365...|      12|Q2CK-NPNU-ZQAD|         1|10 Gove St, Bosto...|
|Point|[-71.058270646725...|      13|Q2CK-7DJ6-6N77|         1|1 City Hall Squar...|
|Point|[-71.051489694591...|      14|Q2CK-LPZ2-AE76|         1|1187 Columbia Rd....|
|Point|[-71.063450589867...|      15|Q2CK-GUKH-WDPW|         1|97 mount ida road...|
|Point|[-71.119880604196...|      16|Q2CK-4SHS-VJAG|         1|60 Fairmont Ave.,...|
|Point|[-71.121335604307...|      17|Q2AK-GPHE-YSR2|         1|26 Central Ave., ...|
|Point|[-71.075406793306...|      18|Q2CK-FC84-AQTJ|         1|270 Columbia Rd.,...|
|Point|[-71.058270646725...|      19|Q2CK-EZ6K-LZEK|         1|1 cityhall square...|
|Point|[-71.038809127214...|      20|Q2CK-ZYHQ-FVKG|         1|69 Paris Street, ...|
+-----+--------------------+--------+--------------+----------+--------------------+
only showing top 20 rows
```

22. Provide the query command and the resulting data set

```
Wifi_df.select(
    'geometry.type',
    'geometry.coordinates',
    'properties.ObjectId',
    'properties.device_serial',
    'properties.is_current',
    'properties.device_address'
).write.mode("overwrite").saveAsTable("Wifi_tbl")
```

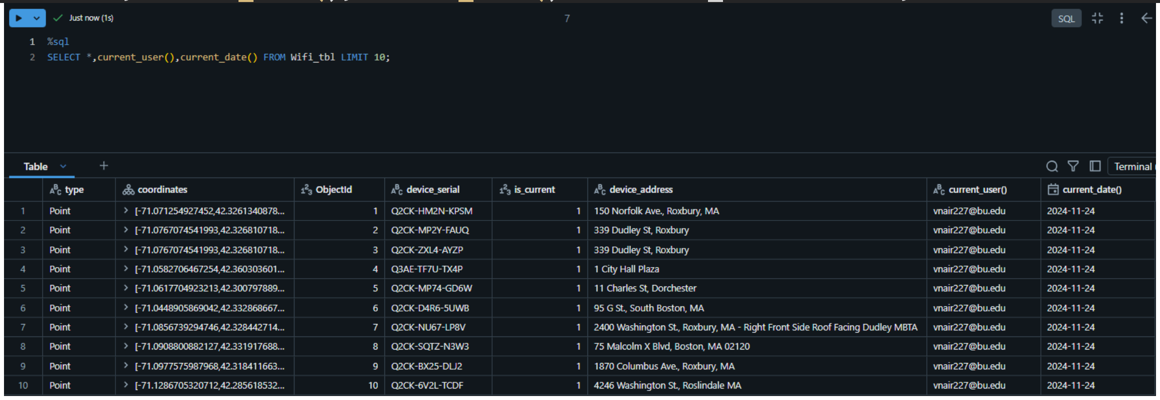


```
print(currentuser)
print(datetime.now())
Wifi_df.select(
    'geometry.type',
    'geometry.coordinates',
    'properties.ObjectId',
    'properties.device_serial',
    'properties.is_current',
    'properties.device_address'
).write.mode("overwrite").saveAsTable("Wifi_tbl")
```

```
vnair227@bu.edu
2024-11-24 22:22:50.666538
```

23. Provide the query command and the resulting data set

```
%sql
SELECT *,current_user(),current_date() FROM Wifi_tbl LIMIT 10;
```

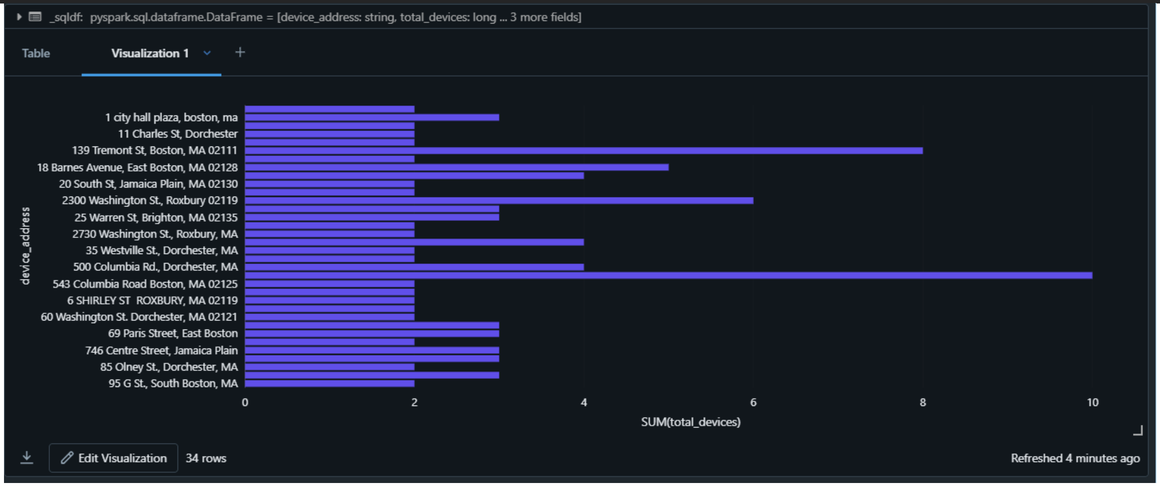


```
%sql
SELECT *,current_user(),current_date() FROM Wifi_tbl LIMIT 10;
```

	type	coordinates	ObjectId	device_serial	is_current	device_address	current_user()	current_date()
1	Point	[-71.071254927452,42.3261340878...	1	Q2CK-HM2N-KPSM	1	150 Norfolk Ave., Roxbury, MA	vnair227@bu.edu	2024-11-24
2	Point	[-71.0767074541993,42.326810718...	2	Q2CK-MP2Y-FAUQ	1	339 Dudley St, Roxbury	vnair227@bu.edu	2024-11-24
3	Point	[-71.0767074541993,42.326810718...	3	Q2CK-ZXL4-AYZP	1	339 Dudley St, Roxbury	vnair227@bu.edu	2024-11-24
4	Point	[-71.0582706467254,42.360303601...	4	Q3AE-TF7U-TX4P	1	1 City Hall Plaza	vnair227@bu.edu	2024-11-24
5	Point	[-71.0617704923213,42.300797889...	5	Q2CK-MP74-GD6W	1	11 Charles St, Dorchester	vnair227@bu.edu	2024-11-24
6	Point	[-71.0448905869042,42.332868667...	6	Q2CK-D4R6-5UWB	1	95 G St., South Boston, MA	vnair227@bu.edu	2024-11-24
7	Point	[-71.0856739294746,42.328442714...	7	Q2CK-NU67-LP8V	1	2400 Washington St., Roxbury, MA - Right Front Side Roof Facing Dudley MBTA	vnair227@bu.edu	2024-11-24
8	Point	[-71.0908800882127,42.331917688...	8	Q2CK-SQTZ-N3W3	1	75 Malcolm X Blvd, Boston, MA 02120	vnair227@bu.edu	2024-11-24
9	Point	[-71.0977575987968,42.318411663...	9	Q2CK-BX25-DLJ2	1	1870 Columbus Ave., Roxbury, MA	vnair227@bu.edu	2024-11-24
10	Point	[-71.1286705320712,42.285618532...	10	Q2CK-6V2L-TCDF	1	4246 Washington St., Roslindale MA	vnair227@bu.edu	2024-11-24

24. Provide the query command and the resulting data set including chart

```
%sql
SELECT
    device_address,
    COUNT(*) AS total_devices,
    SUM(CASE WHEN is_current = 1 THEN 1 ELSE 0 END) AS current_devices,
    ROUND((SUM(CASE WHEN is_current = 1 THEN 1 ELSE 0 END) * 100.0) /
COUNT(*), 2) AS percentage_current,
    RANK() OVER (ORDER BY COUNT(*) DESC) AS address_rank
FROM Wifi_tbl
GROUP BY device_address
HAVING COUNT(*) > 1
ORDER BY address_rank;
```



```
_sqldf:  pyspark.sql.dataframe.DataFrame = [device_address: string, total_devices: long ... 3 more fields]
```

Table
Visualization 1
+
device_address
1 city hall plaza, boston, ma
11 Charles St, Dorchester
139 Tremont St, Boston, MA 02111
18 Barnes Avenue, East Boston, MA 02128
20 South St, Jamaica Plain, MA 02130
2300 Washington St., Roxbury 02119
25 Warren St, Brighton, MA 02135
2730 Washington St., Roxbury, MA
35 Westville St., Dorchester, MA
500 Columbia Rd., Dorchester, MA
543 Columbia Road Boston, MA 02125
6 SHIRLEY ST ROXBURY, MA 02119
60 Washington St. Dorchester, MA 02121
69 Paris Street, East Boston
746 Centre Street, Jamaica Plain
85 Olney St., Dorchester, MA
95 G St., South Boston, MA
0
2
4
6
8
10
SUM(total_devices)
Edit Visualization
34 rows
Refreshed 4 minutes ago

25. Very briefly explain what you have discovered based on your data set from the query above.

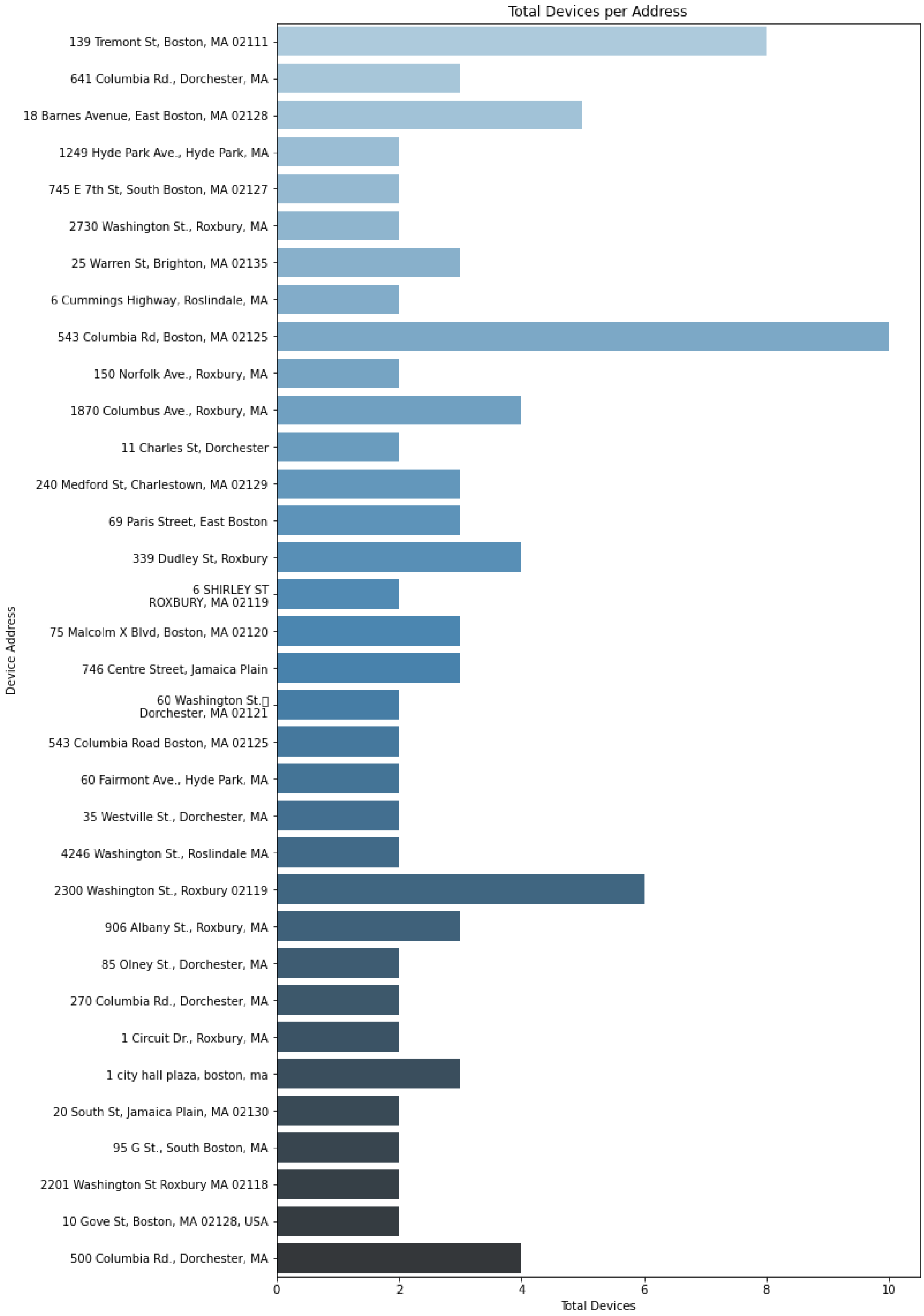
The query lists out the total number of devices that are connected to the wifi at a specific location (at the time the data was collected). It also filters out the addresses which had only 1 device connected.

26. Provide the query command and the resulting data set including chart

```
from pyspark.sql import functions as f
import matplotlib.pyplot as plt
import seaborn as sns

agg = wifi_tbl.groupBy("device_address")\
    .agg(
        f.count("*").alias("total_devices"),
        f.sum(f.when(f.col("is_current")==1,1).otherwise(0)).alias("current_de
vices"),
        f.round(
            (f.sum(f.when(f.col("is_current") == 1, 1).otherwise(0)) * 100.0)
/ f.count("*"), 2
        ).alias("percentage_current")
    )
```

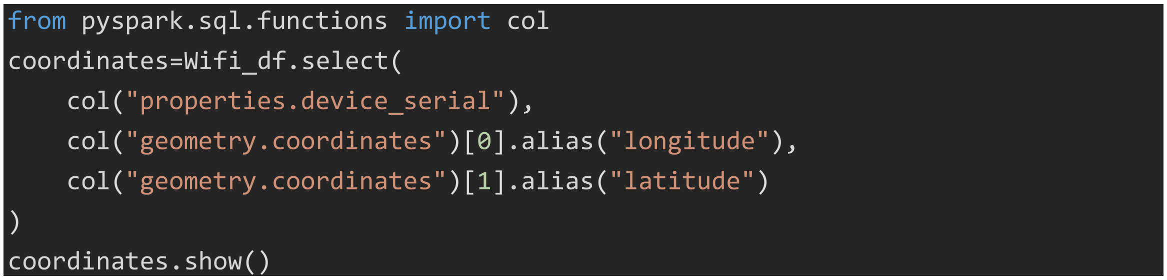
```
agg=agg.toPandas()
aggmorethanone =agg[agg.total_devices>1]
plt.figure(figsize=(10,20))
sns.barplot(x='total_devices', y='device_address', data=aggmorethanone,
palette='Blues_d')
plt.title('Total Devices per Address')
plt.xlabel('Total Devices')
plt.ylabel('Device Address')
plt.show()
```


Total Devices per Address
Device Address
139 Tremont St, Boston, MA 02111
641 Columbia Rd., Dorchester, MA
18 Barnes Avenue, East Boston, MA 02128
1249 Hyde Park Ave., Hyde Park, MA
745 E 7th St, South Boston, MA 02127
2730 Washington St., Roxbury, MA
25 Warren St, Brighton, MA 02135
6 Cummings Highway, Roslindale, MA
543 Columbia Rd, Boston, MA 02125
150 Norfolk Ave., Roxbury, MA
1870 Columbus Ave., Roxbury, MA
11 Charles St, Dorchester
240 Medford St, Charlestown, MA 02129
69 Paris Street, East Boston
339 Dudley St, Roxbury
6 SHIRLEY ST
ROXBURY, MA 02119
75 Malcolm X Blvd, Boston, MA 02120
746 Centre Street, Jamaica Plain
60 Washington St.□
Dorchester, MA 02121
543 Columbia Road Boston, MA 02125
60 Fairmont Ave., Hyde Park, MA
35 Westville St., Dorchester, MA
4246 Washington St., Roslindale MA
2300 Washington St., Roxbury 02119
906 Albany St., Roxbury, MA
85 Olney St., Dorchester, MA
270 Columbia Rd., Dorchester, MA
1 Circuit Dr., Roxbury, MA
1 city hall plaza, boston, ma
20 South St, Jamaica Plain, MA 02130
95 G St., South Boston, MA
2201 Washington St Roxbury MA 02118
10 Gove St, Boston, MA 02128, USA
500 Columbia Rd., Dorchester, MA
0
2
4
6
8
10
Total Devices

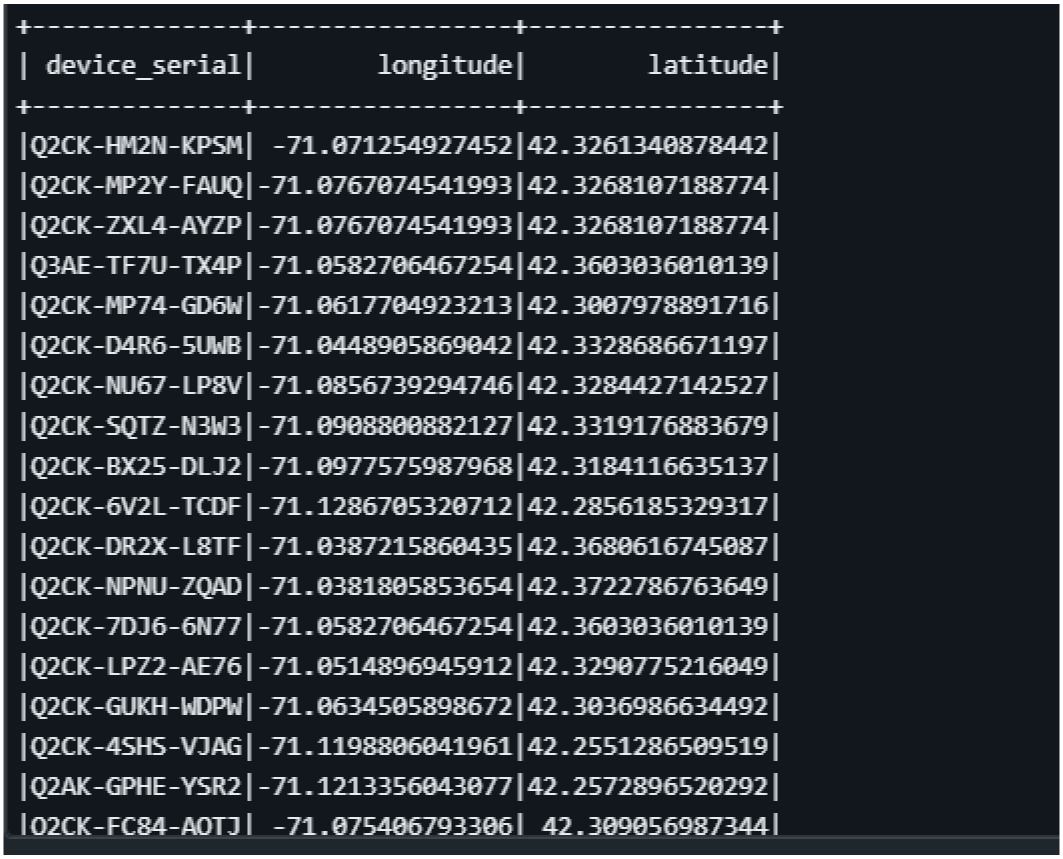
27: Very briefly explain what you have discovered based on your data set from the query above.

The query lists out the total number of devices that are connected to the wifi at a specific location (at the time the data was collected). It also filters out the addresses which had only 1 device connected.

Extra Credit (2 points): Note how one of the columns in the original data frame is an array of coordinates. Look to use the explode function to extract those coordinates into a separate flattened data frame.



```python
from pyspark.sql.functions import col
coordinates=Wifi_df.select(
    col("properties.device_serial"),
    col("geometry.coordinates")[0].alias("longitude"),
    col("geometry.coordinates")[1].alias("latitude")
)
coordinates.show()
```



```
+--------------+-----------------+----------------+
| device_serial|        longitude|        latitude|
+--------------+-----------------+----------------+
|Q2CK-HM2N-KPSM| -71.071254927452|42.3261340878442|
|Q2CK-MP2Y-FAUQ|-71.0767074541993|42.3268107188774|
|Q2CK-ZXL4-AYZP|-71.0767074541993|42.3268107188774|
|Q3AE-TF7U-TX4P|-71.0582706467254|42.3603036010139|
|Q2CK-MP74-GD6W|-71.0617704923213|42.3007978891716|
|Q2CK-D4R6-5UWB|-71.0448905869042|42.3328686671197|
|Q2CK-NU67-LP8V|-71.0856739294746|42.3284427142527|
|Q2CK-SQTZ-N3W3|-71.0908800882127|42.3319176883679|
|Q2CK-BX25-DLJ2|-71.0977575987968|42.3184116635137|
|Q2CK-6V2L-TCDF|-71.1286705320712|42.2856185329317|
|Q2CK-DR2X-L8TF|-71.0387215860435|42.3680616745087|
|Q2CK-NPNU-ZQAD|-71.0381805853654|42.3722786763649|
|Q2CK-7DJ6-6N77|-71.0582706467254|42.3603036010139|
|Q2CK-LPZ2-AE76|-71.0514896945912|42.3290775216049|
|Q2CK-GUKH-WDPW|-71.0634505898672|42.3036986634492|
|Q2CK-4SHS-VJAG|-71.1198806041961|42.2551286509519|
|Q2AK-GPHE-YSR2|-71.1213356043077|42.2572896520292|
|Q2CK-FC84-AOTJ| -71.075406793306| 42.309056987344|
```

Use the **Ask your Facilitator Discussion Board** if you have any questions regarding the how to approach this assignment.

Save your assignment as ***lastnameFirstname_lassignment5.doc*** and submit it in the *Assignments* section of the course.

For help uploading files please refer to the *Technical Support* page in the syllabus.

Criterion	A	B	C	D	F	Letter Grade
Correctness and Completeness of Results (70%)	All steps' results are entirely complete and correct	About ¾ of the steps' results are correct and complete	About half of the steps' results are correct and complete	About ¼ of the steps' results are correct and complete	Virtually none of the step's results are correct and complete	
Constitution of SQL/Python and Explanations (30%)	Excellent use and integration of appropriate SQL/Python constructs and supporting explanations	Good use and integration of appropriate SQL/Python constructs and supporting explanations	Mediocre use and integration of appropriate SQL/Python constructs and supporting explanations	Substandard use and integration of appropriate SQL/Python constructs and supporting explanations	Virtually all SQL/Python constructs and supporting explanations are unsuitable or improperly integrated	
					Assignment Grade:	