In [1]:

```
spark
```

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
4	application_1719381822110_0005	spark	idle	Link	Link	None	✔

```
SparkSession available as 'spark'.

res1: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@79ecefec
```

In [2]:

```
val df_csv_data = spark.read.option("header", "true").option("inferSchema", "true").csv(
"s3://omkar-bucket/assessment3/*.csv")
```

```
df_csv_data: org.apache.spark.sql.DataFrame = [From Date: string, To Date: string ... 23
more fields]
```

In [3]:

```
df_csv_data.write.mode("overwrite").parquet("s3://omkar-bucket/assessment3/parquet_output
")
```

In [4]:

```
val df_parquet = spark.read.option("header",true).option("inferSchema", true).parquet("s3
://omkar-bucket/assessment3/parquet_output/")
```

```
df_parquet: org.apache.spark.sql.DataFrame = [From Date: string, To Date: string ... 23 m
ore fields]
```

In [5]:

```
df_parquet.show()
```

```
+-------------------+-------------------+-------------+------------+----------+--------
---+---------+-----------+-----------+----------+-------------+---------------+--------
-------+-------------------+-----------------+----------------+---------------+------+-
-------+--------+----------+---------+---------+--------------+-------------+
|          From Date|            To Date|PM2.5 (ug/m3)|PM10 (ug/m3)|NO (ug/m3)|NO2 (ug/m
3)|NOx (ppb)|NH3 (ug/m3)|SO2 (ug/m3)|CO (mg/m3)|Ozone (ug/m3)|Benzene (ug/m3)|Toluene (ug
/m3)|Eth-Benzene (ug/m3)|MP-Xylene (ug/m3)|O Xylene (ug/m3)|Temp (degree C)|RH (%)|WS (m/
s)|WD (deg)|SR (W/mt2)|BP (mmHg)|VWS (m/s)|Xylene (ug/m3)|AT (degree C)|
+-------------------+-------------------+-------------+------------+----------+--------
---+---------+-----------+-----------+----------+-------------+---------------+--------
-------+-------------------+-----------------+----------------+---------------+------+-
-------+--------+----------+---------+---------+--------------+-------------+
|2023-02-08 17:00:00|2023-02-08 18:00:00|        10.33|       56.33|      3.03|        8.
47|      7.0|       NULL|       2.63|      0.33|        97.17|            0.2|
1.63|               0.17|             0.03|            NULL|          20.67|   0.2|  128
.67|    60.0|      19.3|     NULL|     NULL|          NULL|         NULL|
|2023-02-08 18:00:00|2023-02-08 19:00:00|         12.0|       81.75|      3.45|       32.
75|    20.23|       NULL|       2.35|      0.76|        23.38|            0.3|
2.58|                0.4|             0.05|            NULL|          31.25|  0.28|  128
.25|    52.0|     17.58|     NULL|     NULL|          NULL|         NULL|
|2023-02-08 19:00:00|2023-02-08 20:00:00|         68.0|      158.75|      9.07|        47
.8|    32.83|       NULL|       2.45|      1.24|          3.9|           1.55|
8.07|                2.9|             0.25|            0.08|           30.5|  0.35|   12
```

```
8.5|    51.5|      16.98|      NULL|      NULL|             NULL|            NULL|
|2023-02-08 20:00:00|2023-02-08 21:00:00|        62.75|       223.25|      13.57|       51.
65|    38.52|         NULL|        2.33|       1.14|          2.55|             1.25|
6.27|                2.17|               0.22|              0.08|           31.25|  0.27|   12
8.5|   50.75|      16.6|      NULL|      NULL|             NULL|            NULL|
|2023-02-08 21:00:00|2023-02-08 22:00:00|        47.25|        160.0|      14.93|       49.
33|    38.33|         NULL|        1.62|        0.9|          4.33|              1.1|
6.35|                 2.1|                0.2|              0.08|            36.5|  0.38|   12
8.5|    49.5|      15.85|      NULL|      NULL|             NULL|            NULL|
|2023-02-08 22:00:00|2023-02-08 23:00:00|        37.25|        131.0|      11.45|       43.
33|    32.32|         NULL|        1.48|       0.52|         19.75|             0.68|
3.77|                1.15|                0.1|              NULL|            34.5|   0.4|   12
8.5|   47.75|      15.77|      NULL|      NULL|             NULL|            NULL|
|2023-02-08 23:00:00|2023-02-09 00:00:00|        26.25|       102.25|       8.72|       46.
38|    31.75|         NULL|        1.25|       0.64|           8.6|              0.5|
3.47|                0.93|                0.1|              0.05|            40.5|  0.15|   12
8.5|   46.75|      15.25|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 00:00:00|2023-02-09 01:00:00|        19.75|       126.75|      11.35|       45.
03|    33.17|         NULL|        1.75|       0.63|           8.5|              0.6|
4.22|                1.57|                0.1|              0.05|           44.25|  0.25|   12
8.5|   45.25|      14.97|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 01:00:00|2023-02-09 02:00:00|         14.0|       131.75|       6.97|       38.
72|    26.32|         NULL|         1.6|       0.42|          32.3|             0.43|
2.95|                0.75|               0.08|              0.03|            41.5|  0.23|   12
8.5|    44.5|      15.03|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 02:00:00|2023-02-09 03:00:00|         19.0|         83.5|       4.45|       33.
72|    21.55|         NULL|        1.05|       0.36|         37.75|             0.38|
2.25|                0.55|               0.03|              0.03|           43.75|  0.25|   12
8.5|   43.75|      14.85|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 03:00:00|2023-02-09 04:00:00|         18.0|         73.0|        3.4|       29.
85|    18.65|         NULL|         0.6|        0.4|         23.62|             0.32|
2.15|                0.45|               0.05|              0.08|            51.0|  0.32|  128
.25|    43.0|      14.2|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 04:00:00|2023-02-09 05:00:00|          9.5|         77.0|        3.4|       28.
65|     18.0|         NULL|        1.05|       0.39|         38.15|             0.33|
2.0|                0.43|               NULL|              0.08|            48.5|  0.35|   128
.5|   43.25|      14.5|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 05:00:00|2023-02-09 06:00:00|        18.75|        79.25|       3.38|        29
.8|     18.6|         NULL|        0.85|       0.49|         23.62|              0.5|
2.05|                0.43|               NULL|              NULL|            53.5|  0.47|   12
8.5|   44.25|      13.78|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 06:00:00|2023-02-09 07:00:00|        23.25|         83.5|       3.38|       29.
83|    18.62|         NULL|        0.92|       0.53|          43.8|             0.58|
2.05|                 0.4|               NULL|              0.05|            36.0|  0.55|   12
8.5|    45.5|      15.15|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 07:00:00|2023-02-09 08:00:00|         24.0|        106.0|        4.1|       29.
47|    19.02|         NULL|        0.95|       0.73|         30.82|             1.07|
2.42|                0.55|               0.03|              0.05|            42.5|  0.57|   12
8.5|    52.0|      14.72|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 08:00:00|2023-02-09 09:00:00|         44.0|        142.0|       4.22|       24.
18|    16.27|         NULL|        0.73|        0.6|         50.02|              1.0|
2.55|                 0.6|               0.05|              0.12|            45.5|   0.6|   12
8.5|    93.5|      15.4|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 09:00:00|2023-02-09 10:00:00|         44.0|       162.75|        4.5|       21.
32|    14.98|         NULL|        1.62|       0.67|         69.28|             0.88|
3.05|                0.65|               0.05|               0.1|            33.5|  0.57|   12
8.5|   157.5|      17.73|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 10:00:00|2023-02-09 11:00:00|         NULL|       168.25|       3.25|       11.
65|      8.8|         NULL|        2.35|       0.33|         97.05|             0.62|
2.88|                0.53|               0.03|              0.12|            24.5|  1.23|   12
8.5|   213.0|      18.75|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 11:00:00|2023-02-09 12:00:00|         16.0|        66.25|       3.05|        7.
47|     6.45|         NULL|        1.75|       0.31|        101.93|             0.45|
2.52|                0.28|               NULL|              0.05|           23.25|   1.3|   12
8.5|   253.5|      19.2|      NULL|      NULL|             NULL|            NULL|
|2023-02-09 12:00:00|2023-02-09 13:00:00|        14.75|         65.0|        3.1|        7.
92|     6.72|         NULL|        2.25|        0.3|          94.0|             0.38|
2.27|                0.25|               0.03|              NULL|           22.25|  1.48|   12
8.5|   264.5|      19.7|      NULL|      NULL|             NULL|            NULL|
+-------------------+-------------------+-------------+------------+-----------+--------
---+---------+-----------+-----------+----------+-------------+---------------+--------
-------+-------------------+-----------------+---------------+----------------+------+-
```

```
-------+---------+----------+---------+---------+-------------+-------------+
only showing top 20 rows
```

In [6]:

```
val sumPM10DF = df_parquet.groupBy("From Date").agg(sum("PM10 (ug/m3)").alias("Total_PM10
"))
sumPM10DF.show()
```

```
sumPM10DF: org.apache.spark.sql.DataFrame = [From Date: string, Total_PM10: double]
+-------------------+------------------+
|          From Date|        Total_PM10|
+-------------------+------------------+
|2023-02-19 17:00:00|          36309.48|
|2022-12-18 00:00:00| 53035.15000000001|
|2022-12-19 22:00:00| 62498.70999999999|
|2022-11-06 08:00:00|          37016.99|
|2012-08-08 05:00:00|             14.21|
|2012-11-22 02:00:00|              NULL|
|2013-02-04 08:00:00|              3.06|
|2013-03-07 08:00:00|            163.98|
|2013-03-10 12:00:00|             83.76|
|2013-03-17 17:00:00|              NULL|
|2013-04-04 04:00:00|            407.96|
|2013-05-08 21:00:00|172.60999999999999|
|2013-10-19 06:00:00|              NULL|
|2013-10-19 08:00:00|              NULL|
|2013-10-21 03:00:00|              9.12|
|2013-10-24 19:00:00|              6.73|
|2013-10-25 12:00:00|             40.23|
|2013-11-25 14:00:00|              1.85|
|2014-05-07 05:00:00|            306.63|
|2014-05-14 01:00:00|             63.75|
+-------------------+------------------+
only showing top 20 rows
```

In [7]:

```
df_parquet.printSchema()
```

```
root
 |-- From Date: string (nullable = true)
 |-- To Date: string (nullable = true)
 |-- PM2.5 (ug/m3): string (nullable = true)
 |-- PM10 (ug/m3): string (nullable = true)
 |-- NO (ug/m3): string (nullable = true)
 |-- NO2 (ug/m3): string (nullable = true)
 |-- NOx (ppb): string (nullable = true)
 |-- NH3 (ug/m3): string (nullable = true)
 |-- SO2 (ug/m3): string (nullable = true)
 |-- CO (mg/m3): string (nullable = true)
 |-- Ozone (ug/m3): string (nullable = true)
 |-- Benzene (ug/m3): string (nullable = true)
 |-- Toluene (ug/m3): string (nullable = true)
 |-- Eth-Benzene (ug/m3): string (nullable = true)
 |-- MP-Xylene (ug/m3): string (nullable = true)
 |-- O Xylene (ug/m3): string (nullable = true)
 |-- Temp (degree C): string (nullable = true)
 |-- RH (%): string (nullable = true)
 |-- WS (m/s): string (nullable = true)
 |-- WD (deg): string (nullable = true)
 |-- SR (W/mt2): string (nullable = true)
 |-- BP (mmHg): string (nullable = true)
 |-- VWS (m/s): string (nullable = true)
 |-- Xylene (ug/m3): string (nullable = true)
 |-- AT (degree C): string (nullable = true)
```

In [8]:

```
val avgPM25DF = df_parquet.groupBy("`From Date`").agg(avg($"`PM2.5 (ug/m3)`").alias("Avg
_PM25"))
avgPM25DF.show()
```

```
avgPM25DF: org.apache.spark.sql.DataFrame = [From Date: string, Avg_PM25: double]
+-------------------+------------------+
|          From Date|          Avg_PM25|
+-------------------+------------------+
|2023-02-19 17:00:00| 72.38812749003985|
|2022-12-18 00:00:00|134.44160869565218|
|2022-12-19 22:00:00| 158.1076348547718|
|2022-11-06 08:00:00| 96.90310185185184|
|2012-08-08 05:00:00|             49.28|
|2012-11-22 02:00:00|203.11666666666668|
|2013-02-04 08:00:00|232.30499999999998|
|2013-03-07 08:00:00|140.50166666666667|
|2013-03-10 12:00:00|            148.32|
|2013-03-17 17:00:00|           105.175|
|2013-04-04 04:00:00|           154.488|
|2013-05-08 21:00:00|            99.015|
|2013-10-19 06:00:00|           181.925|
|2013-10-19 08:00:00|           166.475|
|2013-10-21 03:00:00|           302.435|
|2013-10-24 19:00:00|            192.77|
|2013-10-25 12:00:00|           179.685|
|2013-11-25 14:00:00|             190.5|
|2014-05-07 05:00:00|146.81333333333333|
|2014-05-14 01:00:00| 52.89333333333334|
+-------------------+------------------+
only showing top 20 rows
```

In [18]:

```
val maxPM10DF = df_parquet.groupBy(date_format($"`From Date`", "yyyy-MM-dd").alias("Date
")).agg(sum($"`PM10 (ug/m3)`").alias("Max_PM10"))
maxPM10DF.show()
```

```
maxPM10DF: org.apache.spark.sql.DataFrame = [Date: string, Max_PM10: double]
+----------+------------------+
|      Date|          Max_PM10|
+----------+------------------+
|2013-03-14|            2280.5|
|2017-05-14|59428.029999999984|
|2010-09-24| 8900.239999999998|
|2011-01-29|3826.3200000000006|
|2016-08-08|          16474.36|
|2016-08-20|15348.859999999999|
|2019-09-29|         132535.67|
|2010-03-06|              NULL|
|2023-02-10| 964510.2899999999|
|2015-02-28| 9033.600000000002|
|2015-11-20| 88530.90999999999|
|2016-07-06|27803.090000000004|
|2013-11-08|451.15000000000003|
|2014-03-17| 5535.359999999999|
|2016-04-15| 46232.75999999998|
|2018-03-16|         273989.86|
|2020-06-20| 285331.1400000001|
|2010-06-15|              NULL|
|2014-07-24|1710.0600000000002|
|2014-08-29|           2221.35|
+----------+------------------+
only showing top 20 rows
```

In [10]:

```
val totalNOxDF = df_parquet.groupBy(date_format($"`From Date`", "yyyy-MM-dd").alias("Dat
e")).agg(sum($"`NOx (ppb)`").alias("Total_NOx"))
totalNOxDF.show()
```

```
totalNOxDF: org.apache.spark.sql.DataFrame = [Date: string, Total_NOx: double]
+----------+------------------+
|      Date|         Total_NOx|
+----------+------------------+
|2013-03-14| 6203.450000000002|
|2017-05-14|18009.849999999995|
|2010-09-24|10808.720000000008|
|2011-01-29|15406.469999999998|
|2016-08-08| 19545.46000000001|
|2016-08-20| 8315.269999999999|
|2019-09-29| 67615.86999999997|
|2010-03-06|           9228.27|
|2023-02-10|         235600.93|
|2015-02-28|5468.8600000000015|
|2015-11-20| 29525.05999999999|
|2016-07-06|14975.179999999998|
|2013-11-08|6287.8599999999915|
|2014-03-17|5193.9400000000005|
|2016-04-15|          24527.21|
|2018-03-16| 78227.90000000004|
|2020-06-20| 66671.37999999999|
|2010-06-15|           4652.34|
|2014-07-24| 8644.599999999997|
|2014-08-29|           3268.24|
+----------+------------------+
only showing top 20 rows
```

In [11]:

```
val avgTempDF = df_parquet.groupBy("`From Date`").agg(avg($"`Temp (degree C)`").alias("Av
g_Temp"))
avgTempDF.show()
```

```
avgTempDF: org.apache.spark.sql.DataFrame = [From Date: string, Avg_Temp: double]
+-------------------+------------------+
|          From Date|          Avg_Temp|
+-------------------+------------------+
|2023-02-19 17:00:00|  57.8602752293578|
|2022-12-18 00:00:00| 75.96326203208557|
|2022-12-19 22:00:00| 73.88394871794871|
|2022-11-06 08:00:00| 71.08162790697675|
|2012-08-08 05:00:00|173.08090909090907|
|2012-11-22 02:00:00| 78.07222222222224|
|2013-02-04 08:00:00|107.27333333333335|
|2013-03-07 08:00:00|            130.76|
|2013-03-10 12:00:00|145.86166666666665|
|2013-03-17 17:00:00|             34.67|
|2013-04-04 04:00:00| 52.63249999999999|
|2013-05-08 21:00:00|136.35142857142856|
|2013-10-19 06:00:00| 46.47833333333333|
|2013-10-19 08:00:00|45.913333333333334|
|2013-10-21 03:00:00|169.44166666666663|
|2013-10-24 19:00:00|119.41777777777777|
|2013-10-25 12:00:00|117.06666666666666|
|2013-11-25 14:00:00|122.68874999999998|
|2014-05-07 05:00:00|149.53166666666667|
|2014-05-14 01:00:00|           148.995|
+-------------------+------------------+
only showing top 20 rows
```

In [12]:

```
val maxCODF = df_parquet.groupBy("`From Date`").agg(max($"`CO (mg/m3)`").alias("Max_CO")
)
maxCODF.show()
```

```
maxCODF: org.apache.spark.sql.DataFrame = [From Date: string, Max_CO: string]
+-------------------+------+
|          From Date|Max_CO|
+-------------------+------+
|2010-01-01 20:00:00|  6.75|
|2010-01-05 05:00:00|  6.86|
|2010-01-05 06:00:00|  7.51|
|2010-01-07 02:00:00|   6.7|
|2010-01-07 22:00:00|  8.34|
|2010-01-08 03:00:00|  5.21|
|2010-01-10 22:00:00|  8.98|
|2010-01-11 05:00:00|  5.87|
|2010-01-11 15:00:00|  5.62|
|2010-01-12 12:00:00|  5.61|
|2010-01-14 11:00:00|  3.78|
|2010-01-16 20:00:00|   5.1|
|2010-01-18 07:00:00|   3.6|
|2010-01-19 01:00:00|  3.58|
|2010-01-19 10:00:00| 39.67|
|2010-01-20 05:00:00|  2.81|
|2010-01-20 06:00:00|  2.06|
|2010-01-22 01:00:00|  8.56|
|2010-01-22 18:00:00|  5.37|
|2010-01-23 01:00:00|  2.04|
+-------------------+------+
only showing top 20 rows
```

In [13]:

```
val sumNO2DF = df_parquet.groupBy(date_format($"`From Date`", "yyyy-MM-dd").alias("Date"
)).agg(sum($"`NO2 (ug/m3)`").alias("Sum_NO2"))
sumNO2DF.show()
```

```
sumNO2DF: org.apache.spark.sql.DataFrame = [Date: string, Sum_NO2: double]
+----------+------------------+
|      Date|           Sum_NO2|
+----------+------------------+
|2013-03-14| 4741.159999999998|
|2017-05-14|19435.119999999995|
|2010-09-24|6436.4199999999955|
|2011-01-29| 15709.45999999999|
|2016-08-08|14576.829999999996|
|2016-08-20|          13423.14|
|2019-09-29| 58750.55999999999|
|2010-03-06| 5508.689999999998|
|2023-02-10|          203561.0|
|2015-02-28| 4279.549999999999|
|2015-11-20|32965.669999999984|
|2016-07-06|          13580.98|
|2013-11-08| 4745.839999999998|
|2014-03-17|           3928.83|
|2016-04-15|21599.390000000003|
|2018-03-16|          58946.98|
|2020-06-20| 57952.14000000003|
|2010-06-15| 3597.129999999999|
|2014-07-24| 4733.970000000002|
|2014-08-29|2946.3700000000013|
+----------+------------------+
only showing top 20 rows
```

In [14]:

```
val minRHDF = df_parquet.groupBy(date_format($"`From Date`", "yyyy-MM-dd").alias("Date")
).agg(min($"`RH (%)`").alias("Min_RH"))
minRHDF.show()
```

```
minRHDF: org.apache.spark.sql.DataFrame = [Date: string, Min_RH: string]
+----------+------+
|      Date|Min_RH|
+----------+------+
|2010-01-01|  0.06|
|2010-01-02|  0.04|
|2010-01-03|  0.04|
|2010-01-04|  0.09|
|2010-01-05|  0.06|
|2010-01-06|  0.06|
|2010-01-07|  0.03|
|2010-01-08|  0.03|
|2010-01-09|  0.03|
|2010-01-10|  0.04|
|2010-01-11|  0.03|
|2010-01-12|  0.04|
|2010-01-13|  0.03|
|2010-01-14|  0.02|
|2010-01-15|  0.01|
|2010-01-16|  0.05|
|2010-01-17|  0.06|
|2010-01-18|  0.06|
|2010-01-19|  0.28|
|2010-01-20|  0.02|
+----------+------+
only showing top 20 rows
```

In [15]:

```
val avgWSDF = df_parquet.groupBy("`From Date`").agg(avg($"`WS (m/s)`").alias("Avg_WS"))
avgWSDF.show()
```

```
avgWSDF: org.apache.spark.sql.DataFrame = [From Date: string, Avg_WS: double]
+-------------------+------------------+
|          From Date|            Avg_WS|
+-------------------+------------------+
|2023-02-19 17:00:00|233.45175879396984|
|2022-12-18 00:00:00|245.31698863636365|
|2022-12-19 22:00:00|247.86244565217393|
|2022-11-06 08:00:00|278.26988372093024|
|2012-08-08 05:00:00|162.08272727272728|
|2012-11-22 02:00:00|         253.80625|
|2013-02-04 08:00:00|239.08153846153846|
|2013-03-07 08:00:00|166.77272727272728|
|2013-03-10 12:00:00| 270.0957142857143|
|2013-03-17 17:00:00|           91.5375|
|2013-04-04 04:00:00|           280.985|
|2013-05-08 21:00:00|37.919999999999995|
|2013-10-19 06:00:00| 54.79833333333334|
|2013-10-19 08:00:00|137.38333333333333|
|2013-10-21 03:00:00| 67.56333333333333|
|2013-10-24 19:00:00| 36.41857142857143|
|2013-10-25 12:00:00|159.20666666666665|
|2013-11-25 14:00:00| 64.10874999999999|
|2014-05-07 05:00:00|            13.705|
|2014-05-14 01:00:00|              6.56|
+-------------------+------------------+
only showing top 20 rows
```

In [16]:

```
val maxSRDF = df_parquet.groupBy("`From Date`").agg(max($"`SR (W/mt2)`").alias("Max_SR")
)
maxSRDF.show()
```

```
maxSRDF: org.apache.spark.sql.DataFrame = [From Date: string, Max_SR: string]
+-------------------+------+
|          From Date|Max_SR|
+-------------------+------+
|2010-01-01 20:00:00| 999.9|
|2010-01-05 05:00:00|994.72|
|2010-01-05 06:00:00|995.13|
|2010-01-07 02:00:00|998.16|
|2010-01-07 22:00:00|732.52|
|2010-01-08 03:00:00|999.13|
|2010-01-10 22:00:00| 91.91|
|2010-01-11 05:00:00| 87.88|
|2010-01-11 15:00:00|732.33|
|2010-01-12 12:00:00|732.79|
|2010-01-14 11:00:00|731.31|
|2010-01-16 20:00:00|732.61|
|2010-01-18 07:00:00| 89.26|
|2010-01-19 01:00:00|732.72|
|2010-01-19 10:00:00|732.45|
|2010-01-20 05:00:00|732.58|
|2010-01-20 06:00:00|732.64|
|2010-01-22 01:00:00|999.58|
|2010-01-22 18:00:00|998.86|
|2010-01-23 01:00:00|  8.15|
+-------------------+------+
only showing top 20 rows
```

In [17]:

```
val totalBenzeneDF = df_parquet.groupBy("`From Date`").agg(sum($"`Benzene (ug/m3)`").alia
s("Total_Benzene"))
totalBenzeneDF.show()
```

```
totalBenzeneDF: org.apache.spark.sql.DataFrame = [From Date: string, Total_Benzene: doubl
e]
+-------------------+------------------+
|          From Date|     Total_Benzene|
+-------------------+------------------+
|2023-02-19 17:00:00|           1745.31|
|2022-12-18 00:00:00|           1934.65|
|2022-12-19 22:00:00|1990.8899999999999|
|2022-11-06 08:00:00|           1475.08|
|2012-08-08 05:00:00|             116.4|
|2012-11-22 02:00:00|             96.88|
|2013-02-04 08:00:00|             40.92|
|2013-03-07 08:00:00|             75.19|
|2013-03-10 12:00:00|35.519999999999996|
|2013-03-17 17:00:00|              3.41|
|2013-04-04 04:00:00|             79.42|
|2013-05-08 21:00:00|39.089999999999996|
|2013-10-19 06:00:00|             21.98|
|2013-10-19 08:00:00|             22.86|
|2013-10-21 03:00:00|             53.07|
|2013-10-24 19:00:00|             87.66|
|2013-10-25 12:00:00|             52.19|
|2013-11-25 14:00:00|              32.4|
|2014-05-07 05:00:00|             34.84|
|2014-05-14 01:00:00|              34.8|
+-------------------+------------------+
only showing top 20 rows
```

In [19]:

```
val mysqlUrl = "jdbc:mysql://34.94.143.214/spark_emr"
```

```
mysqlUrl: String = jdbc:mysql://34.94.143.214/spark_emr
```

```
mysqlUrl: String = jdbc:mysql://54.94.145.214/spark_emr
```

In [20]:

```
val mysqlUser = "admin"
```

```
mysqlUser: String = admin
```

In [48]:

```
val mysqlPassword = "XXXXX"
```

```
mysqlPassword: String = XXXXX
```

Write in SQL

In [22]:

```
sumPM10DF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "spark_emr"
)).mode("append").save()
```

In [23]:

```
avgPM25DF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "avg_PM_25DF"
)).mode("append").save()
```

In [24]:

```
maxPM10DF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "max_PM_10DF"
)).mode("append").save()
```

In [25]:

```
totalNOxDF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "total_nox_df"
)).mode("append").save()
```

In [26]:

```
avgTempDF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "avg_temp_df"
)).mode("append").save()
```

In [27]:

```
maxCODF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "max_co_df"
)).mode("append").save()
```

In [28]:

```
sumNO2DF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "sum_no2_df"
)).mode("append").save()
```

In [29]:

```
minRHDF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "min_RH_df"
)).mode("append").save()
```

In [30]:

```
avgWSDF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "avg_ws_df"
)).mode("append").save()
```

In [31]:

```
maxSRDF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "max_sr_df"
)).mode("append").save()
```

In [32]:

```
totalBenzeneDF.write.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "total_benzene_df"
)).mode("append").save()
```

Read from SQL

In [37]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "spark_emr"
)).load().show(10)
```

```
+-------------------+------------------+
|          From Date|        Total_PM10|
+-------------------+------------------+
|2023-02-16 18:00:00|60347.759999999995|
|2023-03-05 17:00:00|          36095.38|
|2023-03-23 16:00:00|26050.019999999997|
|2023-01-19 22:00:00| 86513.67000000001|
|2023-01-28 00:00:00|52010.100000000006|
|2023-01-30 01:00:00|          41274.53|
|2023-01-05 01:00:00|           56422.3|
|2023-01-09 20:00:00| 93964.46999999999|
|              GJ015|              NULL|
|              RJ028|              NULL|
+-------------------+------------------+
only showing top 10 rows
```

In [38]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "avg_PM_25DF"
)).load().show(10)
```

```
+-------------------+-----------------+
|          From Date|         Avg_PM25|
+-------------------+-----------------+
|2023-02-10 10:00:00| 74.0163202247191|
|2023-02-16 18:00:00|74.19530054644808|
|2023-02-10 12:00:00|58.96759103641457|
|2023-02-13 19:00:00| 51.6024309392265|
|2023-02-16 04:00:00|84.89834269662919|
|2023-02-23 23:00:00|65.47857923497267|
|2023-03-11 22:00:00|79.65984293193719|
|2023-01-10 05:00:00|91.56450479233226|
|              BR010|             NULL|
|              HR001|             NULL|
+-------------------+-----------------+
only showing top 10 rows
```

In [39]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "max_PM_10DF"
)).load().show(10)
```

```
+----------+--------+
|      Date|Max_PM10|
+----------+--------+
|2010-09-28|   98.18|
|2010-02-12|    NULL|
|2011-01-23|   94.99|
|2010-09-24|   94.66|
|2010-01-23|    NULL|
|2011-01-27|   94.49|
|2011-01-29|   93.15|
|2011-07-16|    4.44|
|2013-03-14|    96.1|
|2012-01-12|    9.54|
+----------+--------+
only showing top 10 rows
```

In [40]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "total_nox_df"
)).load().show(10)
```

```
+----------+------------------+
|      Date|         Total_NOx|
+----------+------------------+
|2023-02-10|329828.98000000004|
|2023-01-21| 351930.5299999997|
|2011-01-23| 7603.879999999999|
|2022-10-05|179622.47999999998|
|2011-01-27|11014.810000000005|
|2011-01-29| 21226.91999999999|
|2011-07-16|20995.409999999996|
|2013-03-14|10786.090000000002|
|2012-01-12|          14251.07|
|2014-02-16| 7912.830000000002|
+----------+------------------+
only showing top 10 rows
```

In [41]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "avg_temp_df"
)).load().show(10)
```

```
+-------------------+-----------------+
|          From Date|         Avg_Temp|
+-------------------+-----------------+
|2023-02-16 18:00:00|72.03332247557005|
|2023-03-05 17:00:00|68.71694078947367|
|2023-03-23 16:00:00|67.85554878048781|
|2023-01-19 22:00:00|76.71377622377622|
|2023-01-28 00:00:00|83.35927835051547|
|2023-01-30 01:00:00|84.60905454545457|
|2023-01-05 01:00:00| 88.5828624535316|
|2023-01-09 20:00:00|84.55816546762591|
|               GJ015|              NULL|
|               RJ028|              NULL|
+--------------------+------------------+
only showing top 10 rows
```

In [42]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "max_co_df"
)).load().show(10)
```

```
+--------------------+-------+
|           From Date|Max_CO|
+--------------------+-------+
|2010-01-02 00:00:00|  5.06|
|2010-01-09 17:00:00|  5.61|
|2010-01-17 04:00:00|   1.7|
|2010-01-19 14:00:00|  6.63|
|2010-01-30 06:00:00| 41.17|
|2010-01-30 12:00:00|  5.31|
|2010-02-04 19:00:00|  2.24|
|2010-02-17 19:00:00|  5.67|
|2010-02-20 18:00:00|  1.66|
|2010-03-07 17:00:00|  4.45|
+--------------------+-------+
only showing top 10 rows
```

In [43]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "sum_no2_df"
)).load().show(10)
```

```
+----------+------------------+
|      Date|           Sum_NO2|
+----------+------------------+
|2013-11-08|10063.550000000007|
|2023-02-10| 271396.5600000002|
|2014-03-17| 7520.670000000001|
|2014-03-20|           5846.01|
|2014-11-01|13555.999999999998|
|2015-08-15|11423.780000000002|
|2011-01-23|10081.470000000001|
|2016-01-16|25550.979999999996|
|2016-03-17|22708.220000000005|
|2011-01-27|13201.529999999999|
+----------+------------------+
only showing top 10 rows
```

In [44]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "min_RH_df"
)).load().show(10)
```

```
+----------+------+
|      Date|Min_RH|
+----------+------+
|2010-03-06|  0.02|
|2012-03-04| -0.01|
|2012-10-21|  -0.0|
|2014-07-14|  -0.1|
|2014-12-11| -0.14|
|2015-02-09| -0.03|
|2015-02-27| -0.03|
|2016-04-22|  -0.0|
|2016-08-08|  -0.0|
|2016-08-20| -0.01|
+----------+------+
only showing top 10 rows
```

In [45]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "avg_ws_df"
)).load().show(10)
```

```
+-------------------+------------------+
|          From Date|            Avg_WS|
+-------------------+------------------+
|2023-02-10 10:00:00| 266.8008053691275|
|2023-02-10 23:00:00|223.77164285714284|
|2023-02-13 16:00:00| 244.7440604026846|
|2023-02-17 10:00:00|  249.421821192053|
|2023-03-03 05:00:00|221.19941379310345|
|2023-03-06 13:00:00| 271.3822580645161|
|2023-02-03 01:00:00|220.89655913978498|
|2023-02-04 11:00:00| 277.2291467576792|
|2023-01-14 17:00:00|220.35160142348755|
|2023-01-29 10:00:00| 235.0301742160279|
+-------------------+------------------+
only showing top 10 rows
```

In [46]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "max_sr_df"
)).load().show(10)
```

```
+-------------------+------+
|          From Date|Max_SR|
+-------------------+------+
|2010-01-14 07:00:00|731.71|
|2010-01-14 10:00:00|731.43|
|2010-01-18 07:00:00| 89.26|
|2010-01-20 05:00:00|732.58|
|2010-02-10 05:00:00|995.46|
|2010-02-14 16:00:00|993.72|
|2010-02-15 05:00:00|994.56|
|2010-02-16 05:00:00|996.29|
|2010-02-18 19:00:00|995.09|
|2010-02-22 23:00:00|995.25|
+-------------------+------+
only showing top 10 rows
```

In [47]:

```
spark.read.format("jdbc").options(
Map(
"url" -> mysqlUrl,
"user" -> mysqlUser,
"password" -> mysqlPassword,
"dbtable" -> "total_benzene_df"
)).load().show(10)
```

```
+-------------------+------------------+
|          From Date|     Total_Benzene|
+-------------------+------------------+
|2023-02-16 18:00:00|1946.0400000000004|
|2023-02-10 12:00:00|           1602.98|
|2023-03-05 17:00:00|           1737.29|
|2023-03-23 16:00:00|1680.6399999999999|
|2023-01-19 22:00:00|           2612.93|
|2023-01-28 00:00:00|           2007.98|
|2023-01-30 01:00:00|1635.1000000000004|
|2023-01-05 01:00:00|1623.5199999999998|
|2023-01-09 20:00:00|           2402.82|
|              GJ015|              NULL|
+-------------------+------------------+
only showing top 10 rows
```

In [ ]: