In [0]:

```
from pyspark.sql import SparkSession
spark= SparkSession.builder.appName("guru").getOrCreate()
```

Extraction

In [0]:

```
df=spark.read.format('csv').option('header',True).option('inferSchema',True) .
```

In [0]:

```
df.display()
```

In [0]:

```
df.printSchema()
```

```
root
 |-- Violation_ID: string (nullable = true)
 |-- Violation_Type: string (nullable = true)
 |-- Fine_Amount: integer (nullable = true)
 |-- Location: string (nullable = true)
 |-- Date: date (nullable = true)
 |-- Time: timestamp (nullable = true)
 |-- Vehicle_Type: string (nullable = true)
 |-- Vehicle_Color: string (nullable = true)
 |-- Vehicle_Model_Year: integer (nullable = true)
 |-- Registration_State: string (nullable = true)
 |-- Driver_Age: integer (nullable = true)
 |-- Driver_Gender: string (nullable = true)
 |-- License_Type: string (nullable = true)
 |-- Penalty_Points: integer (nullable = true)
 |-- Weather_Condition: string (nullable = true)
 |-- Road_Condition: string (nullable = true)
 |-- Officer_ID: string (nullable = true)
 |-- Issuing_Agency: string (nullable = true)
 |-- License_Validity: string (nullable = true)
 |-- Number_of_Passengers: integer (nullable = true)
 |-- Helmet_Worn: string (nullable = true)
 |-- Seatbelt_Worn: string (nullable = true)
 |-- Traffic_Light_Status: string (nullable = true)
 |-- Speed_Limit: integer (nullable = true)
 |-- Recorded_Speed: integer (nullable = true)
 |-- Alcohol_Level: double (nullable = true)
 |-- Breathalyzer_Result: string (nullable = true)
 |-- Towed: string (nullable = true)
 |-- Fine_Paid: string (nullable = true)
 |-- Payment_Method: string (nullable = true)
 |-- Court_Appearance_Required: string (nullable = true)
 |-- Previous_Violations: integer (nullable = true)
 |-- Comments: string (nullable = true)
```

In [0]:

```
df.dropDuplicates()
```

```
Out[6]: DataFrame[Violation_ID: string, Violation_Type: string, Fine_Amount: in
t, Location: string, Date: date, Time: timestamp, Vehicle_Type: string, Vehicl
e_Color: string, Vehicle_Model_Year: int, Registration_State: string, Driver_Ag
e: int, Driver_Gender: string, License_Type: string, Penalty_Points: int, Weath
er_Condition: string, Road_Condition: string, Officer_ID: string, Issuing_Agenc
y: string, License_Validity: string, Number_of_Passengers: int, Helmet_Worn: st
ring, Seatbelt_Worn: string, Traffic_Light_Status: string, Speed_Limit: int, Re
corded_Speed: int, Alcohol_Level: double, Breathalyzer_Result: string, Towed: s
tring, Fine_Paid: string, Payment_Method: string, Court_Appearance_Required: st
ring, Previous_Violations: int, Comments: string]
```

```
In [0]: df.display()
```

```
In [0]: from pyspark.sql.functions import *
```

```
In [0]: df.select([sum(when(col(c).isNull(), 1).otherwise(0)).alias(c) for c in df.col
```

Violation_ID	Violation_Type	Fine_Amount	Location	Date	Time	Vehicle_Type	Ve
0	0	0	0	0	0	0	

FINDING NULL VALUE COUNT

```
In [0]: df.select([sum(when(col(c).isNull(),1).otherwise(0)).alias(c) for c in df.colu
```

Violation_ID	Violation_Type	Fine_Amount	Location	Date	Time	Vehicle_Type	Ve
0	0	0	0	0	0	0	

MOST COMMON VIOLATION TYPE

```
In [0]: df.groupBy("Violation_Type").count().display()
```

Violation_Type	count
Overloading	399
Signal Jumping	446
Over-speeding	448
Driving Without License	451
Wrong Parking	454
Drunk Driving	488
No Helmet	463
No Seatbelt	440
Using Mobile Phone	411

```
In [0]: df.groupBy("Violation_Type").count().orderBy('count', ascending=False).show(1)
```

```
+--------------+-----+
|Violation_Type|count|
+--------------+-----+
| Drunk Driving|  488|
+--------------+-----+
only showing top 1 row
```

MOST COMMON TRAFFICE VIOLATION AS PER LOCATION

```
In [0]: df.groupBy('location').count().orderBy('count', ascending=False).show(10)
```

```
+-------------+-----+
|     location|count|
+-------------+-----+
|      Gujarat|  520|
|  Maharashtra|  504|
|       Punjab|  503|
|  West Bengal|  503|
|   Tamil Nadu|  500|
|Uttar Pradesh|  499|
|        Delhi|  492|
|    Karnataka|  479|
+-------------+-----+
```

VIOLATION HAPPENED IN GUJARAT WHO HAVE MAXIMUM VIOLATION COUNT

```
In [0]: df.filter(col('location')=='Gujarat').groupBy('violation_type').count().show()
```

```
+--------------------+-----+
|      violation_type|count|
+--------------------+-----+
|         Overloading|   51|
|      Signal Jumping|   63|
|       Over-speeding|   50|
|Driving Without L...|   68|
|       Wrong Parking|   57|
|       Drunk Driving|   68|
|           No Helmet|   60|
|         No Seatbelt|   60|
|  Using Mobile Phone|   43|
+--------------------+-----+
```

TOTAL FINE AMOUNT PAID

```
In [0]: df.select(sum('Fine_Amount').alias('Totat_Amount')).show()
```

```
+------------+
|Totat_Amount|
+------------+
|    10119285|
+------------+
```

In [0]:

```
df.groupBy('Fine_Paid').agg(sum('Fine_Amount').alias('total_fines')).show()
```

```
+---------+-----------+
|Fine_Paid|total_fines|
+---------+-----------+
|       No|    5187461|
|      Yes|    4931824|
+---------+-----------+
```

Violation count by road condition

In [0]:

```
df.groupBy("Road_Condition").count().orderBy("count", ascending=False).show()
```

```
+------------------+-----+
|    Road_Condition|count|
+------------------+-----+
|          Slippery|  833|
|Under Construction|  821|
|               Dry|  810|
|               Wet|  775|
|          Potholes|  761|
+------------------+-----+
```

Violation count by weather condition

In [0]:

```
df.groupBy('Weather_Condition').count().orderBy('count', ascending=False).show
```

```
+-----------------+-----+
|Weather_Condition|count|
+-----------------+-----+
|            Rainy|  817|
|           Cloudy|  807|
|       Dust Storm|  801|
|            Clear|  798|
|            Foggy|  777|
+-----------------+-----+
```

In [0]:

```
df.filter(col('Weather_Condition')=='Rainy').groupBy('Violation_Type').count()
```

```
+--------------------+-----+
|      Violation_Type|count|
+--------------------+-----+
|       Drunk Driving|  109|
|           No Helmet|  100|
|       Over-speeding|   91|
|Driving Without L...|   89|
|         No Seatbelt|   89|
|         Overloading|   86|
|       Wrong Parking|   85|
|      Signal Jumping|   84|
|  Using Mobile Phone|   84|
+--------------------+-----+
```

```
In [0]: df.filter(df["Fine_Paid"] == False).groupBy("Location").agg(sum("Fine_Amount")
```

```
+-------------+------------+
|     Location|Unpaid_Fines|
+-------------+------------+
|   Tamil Nadu|      701498|
|      Gujarat|      685253|
|Uttar Pradesh|      684321|
|  West Bengal|      673041|
|  Maharashtra|      663090|
|       Punjab|      627970|
|    Karnataka|      582543|
|        Delhi|      569745|
+-------------+------------+
```

```
In [0]: gujarat_data =df.filter((col('Fine_Paid')==False) & (col('location')=='Gujarat
```

```
In [0]: gujarat_data.display()
```

Violation_ID	Violation_Type	Fine_Amount	Location	Date	
VLT100014	No Seatbelt	4098	Gujarat	2023-01-15	2025-05-23T20:39:
VLT100048	No Seatbelt	3287	Gujarat	2023-02-18	2025-05-23T14:37:
VLT100050	Over-speeding	4845	Gujarat	2023-02-20	2025-05-23T08:18:
VLT100060	Overloading	397	Gujarat	2023-03-02	2025-05-23T02:56:
VLT100062	No Seatbelt	2253	Gujarat	2023-03-04	2025-05-23T22:54:
VLT100076	Overloading	4679	Gujarat	2023-03-18	2025-05-23T15:10:
VLT100096	Over-speeding	4657	Gujarat	2023-04-07	2025-05-23T12:56:

LOADING DATA INTO CSV FILE --(GUJARAT WHOSE FINE NOT PAID)

In [0]:

```
gujarat_data.coalesce(1).write.mode("overwrite").option("header", True).csv("c
```