

Analyzing Road Safety in the UK

Business problem:

The UK Department of Transport provides open datasets on road safety and casualties, and one can use these datasets to analyze how safe the roads in the UK are. This project will help you answer a few questions using their 2015 dataset. The dataset has 3 tables i.e Accident, vehicle, Vehicle_type Approach/Project Idea Use aggregate functions in SQL and Python to answer the following sample questions:

1. Evaluate the median severity value of accidents caused by various Motorcycles.
2. Evaluate Accident Severity and Total Accidents per Vehicle Type
3. Calculate the Average Severity by vehicle type.
4. Calculate the Average Severity and Total Accidents by Motorcycle.

Github source code link: <https://lnkd.in/dbEAKY2G> (<https://lnkd.in/dbEAKY2G>)

In [1]:

```
1 import pandas as pd
2 import numpy as np
```

Reading Datasets

1.Accident_2015.csv

2.Vehicles_2015.csv

In [2]:

```
1 #Accident_2015.csv
2 df_Accident_2015=pd.read_csv('https://raw.githubusercontent.com/ptyadana/SQL-Data-Analy
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_11200\686553673.py:2: DtypeWarning: Columns (0) have mixed types. Specify dtype option on import or set low_memory=False.

```
df_Accident_2015=pd.read_csv('https://raw.githubusercontent.com/ptyadana/SQL-Data-Analysis-and-Visualization-Projects/master/Database%20Clinics%20-%20MySQL/02.UK%20Road%20Safty%20Accidents%202015/datasets/Accidents_2015.csv')
```

In [3]:

```
1 df_Accident_2015
```

Out[3]:

	Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Force
0	201501BS70001	525130.0	180050.0	-0.198465	51.505538	
1	201501BS70002	526530.0	178560.0	-0.178838	51.491836	
2	201501BS70004	524610.0	181080.0	-0.205590	51.514910	
3	201501BS70005	524420.0	181080.0	-0.208327	51.514952	
4	201501BS70008	524630.0	179040.0	-0.206022	51.496572	
...
140051	2015984139115	312087.0	570791.0	-3.376671	55.023855	9
140052	2015984139715	320671.0	569791.0	-3.242159	55.016316	9
140053	2015984140215	311731.0	586343.0	-3.387067	55.163502	9

In [4]:

```
1 df_Accident_2015.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 140056 entries, 0 to 140055
Data columns (total 32 columns):
```

#	Column	Non-Null Count	Dtype
0	Accident_Index	140056 non-null	object
1	Location_Easting_OSGR	140029 non-null	float64
2	Location_Northing_OSGR	140029 non-null	float64
3	Longitude	140029 non-null	float64
4	Latitude	140029 non-null	float64
5	Police_Force	140056 non-null	int64
6	Accident_Severity	140056 non-null	int64
7	Number_of_Vehicles	140056 non-null	int64
8	Number_of_Casualties	140056 non-null	int64
9	Date	140056 non-null	object
10	Day_of_Week	140056 non-null	int64
11	Time	140038 non-null	object
12	Local_Authority_(District)	140056 non-null	int64
13	Local_Authority_(Highway)	140056 non-null	object
14	1st_Road_Class	140056 non-null	int64
15	1st_Road_Number	140056 non-null	int64
16	Road_Type	140056 non-null	int64
17	Speed_limit	140056 non-null	int64
18	Junction_Detail	140056 non-null	int64
19	Junction_Control	140056 non-null	int64
20	2nd_Road_Class	140056 non-null	int64
21	2nd_Road_Number	140056 non-null	int64
22	Pedestrian_Crossing-Human_Control	140056 non-null	int64
23	Pedestrian_Crossing-Physical_Facilities	140056 non-null	int64
24	Light_Conditions	140056 non-null	int64
25	Weather_Conditions	140056 non-null	int64
26	Road_Surface_Conditions	140056 non-null	int64
27	Special_Conditions_at_Site	140056 non-null	int64
28	Carriageway_Hazards	140056 non-null	int64
29	Urban_or_Rural_Area	140056 non-null	int64
30	Did_Police_Officer_Attend_Scene_of_Accident	140056 non-null	int64
31	LSOA_of_Accident_Location	131159 non-null	object

```
dtypes: float64(4), int64(23), object(5)
```

```
memory usage: 34.2+ MB
```

In [5]:

```
1 #Converting to csv
2 df_Accident_2015.to_csv('Accident_2015.csv',index=False)
```

In [6]:

```
1 #Vehicle_2015.csv
2 df_Vehicle_2015=pd.read_csv('https://raw.githubusercontent.com/ptyadana/SQL-Data-Analysis')
```

In [7]:

```
1 df_Vehicle_2015.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 257845 entries, 0 to 257844
Data columns (total 23 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Accident_Index                           257845 non-null  object
1   Vehicle_Reference                         257845 non-null  int64
2   Vehicle_Type                             257845 non-null  int64
3   Towing_and_Articulation                 257845 non-null  int64
4   Vehicle_Manoeuvre                       257845 non-null  int64
5   Vehicle_Location-Restricted_Lane        257845 non-null  int64
6   Junction_Location                       257845 non-null  int64
7   Skidding_and_Overturning                257845 non-null  int64
8   Hit_Object_in_Carriageway               257845 non-null  int64
9   Vehicle_Leaving_Carriageway             257845 non-null  int64
10  Hit_Object_off_Carriageway               257845 non-null  int64
11  1st_Point_of_Impact                      257845 non-null  int64
12  Was_Vehicle_Left_Hand_Drive?             257845 non-null  int64
13  Journey_Purpose_of_Driver                  257845 non-null  int64
14  Sex_of_Driver                           257845 non-null  int64
15  Age_of_Driver                            257845 non-null  int64
16  Age_Band_of_Driver                       257845 non-null  int64
17  Engine_Capacity_(CC)                     257845 non-null  int64
18  Propulsion_Code                          257845 non-null  int64
19  Age_of_Vehicle                           257845 non-null  int64
20  Driver_IMD_Decile                        257845 non-null  int64
21  Driver_Home_Area_Type                     257845 non-null  int64
22  Vehicle_IMD_Decile                       257845 non-null  int64
dtypes: int64(22), object(1)
memory usage: 45.2+ MB
```

In [8]:

```
1 #Converting to csv
2 df_Vehicle_2015.to_csv('Vehicle_2015.csv',index=False)
```