BSD2343 DATA WAREHOUSING

2021/2022 SEMESTER II

GROUP'S NAME: Jupyter 404

TITLE: ACCIDENT RISK INDEX

PREPARED FOR

DR AZUANA BINTI RAMLI

PREPARED BY

MATRIC ID	**NAME**	**SECTION**
SD20008	SARAH BATRISYIA BINTI MORSHIDI	01G
SD20009	SITI NUR AISYAH BINTI ANUAR	
SD20012	AHMAD MUHSIN BIN MOHD NIZAM	
SD20022	NIK NUR AIN BINTI NIK JID	
SD20057	SRI VEERA SIVA THACHAYAANI A/P VESVANATHAN	

TABLE OF CONTENTS

1.0 BACKGROUND

1.1 Project Background

Road traffic accidents are one of the most serious threats to human life. Despite widespread efforts to regulate and mitigate the problem, road traffic accidents continue to rise on a daily basis. According to the World Health Organization's (WHO) 2015 report on road safety, road traffic injuries kill over 1.25 million people each year and have a huge impact on human life and development. These events, in particular, constitute the leading cause of death among young persons aged 15 to 29. In low- and middle-income nations, the cost of deaths and injuries amounts to around 3% of the Gross Domestic Product (GDP). Despite the enormous human and economic costs, efforts to combat this global threat remain insufficient.

To address the preventable problem of poor road safety, many ministries, most notably legislation, planning, transportation, education, public information, and health, must work together in improving the built environment (e.g., safer road design, regulating sidewalks and traffic lights, introducing safe bicycle lanes), law enforcement and education to increase seatbelt use and helmet wearing while reducing speeding and drunk driving, better vehicle standards, and improved post-crash response are among the measures to ensure road safety. Road safety solutions that provide safer, more sustainable public transportation options are also very attractive and have the potential to create synergies between health, transportation, and carbon emission reduction aims.

On the one hand, economic progress has expanded global motorization, particularly in low- and middle-income countries. The rising number of motor vehicles necessitates additional roads and a greater requirement for improved road safety and protection measures. The number of motor vehicles increased by 16% globally in 2014; nevertheless, it is crucial to remember that the road network has not improved at the same rate. Using the fatality rate (deaths in road crashes per 100,000 people) as a road safety measure, the Southeast Asian Region continues to outperform Europe.This worrying circumstance emphasises the importance of promoting risk prevention efforts across borders. As previously said, the figures on road accidents and injuries show that low- and middle-income countries have the highest fatality rate, which is nearly twice that of developed countries. The road safety circumstances of low-, middle-, and high-income countries varies in the Asian region. According to WHO, the comparison of fatality rates (FR) demonstrates that average FR values for low-, middle-, and high-income nations.

Countries have devised and implemented numerous road safety measures in order to reduce traffic accidents. It is worth noting that industrialised countries have been successful in reducing road accidents. These accomplishments are the result of safer infrastructure, improved vehicle safety, and the implementation of a variety of other measures shown to reduce road traffic injuries. Having high-quality data to track the impact of these activities is also essential for demonstrating their success. However, developing and underdeveloped countries have yet to reach this level of achievement. Regular road inspections are an important step in ensuring the quality of roads and road surfaces.Taking into account some major factors such as institutional framework, alcohol usage and speeds, protective systems, vehicles infrastructure and roads and trauma management. The goal of this study is to statistically develop and investigate the relationships between the accident rate and the effectiveness of the traffic police.

1.2 Description of Data

The dataset that we used in our project is about accidents data that happened in the United Kingdom. This data consists of several tables which are population table, roads table, sample table, test table and train table.

For the population table, there are 10 columns.

Variable	Data Type	Description
postcode	string	The postcode of the area.
Rural Urban	string	This data indicates that the dataset is the total data from the rural and urban area.

Variable: All usual residents; measures: Value	integer	The number of population.
Variable: Males; measures: Value	integer	The number of males .
Variable: Females; measures: Value	integer	The number of females.
Variable: Lives in a household; measures: Value	integer	The number of residents that live in a household.
Variable: Lives in a communal establishment; measures: Value	integer	The number of residents that live in a communal establishment.
Variable: Schoolchild or full-time student aged 4 and over at their non-term time address; measures: Value	integer	The number of students aged 4 and above.
Variable: Area (Hectares); measures: Value	float	The area of the place is in hectares.
Variable: Density (number of persons per hectare); measures: Value	float	The number of persons per hectare.

For the roads table, there are 8 columns

Variable	Data Type	Description
WKT	string	Well-known text of the geometry of the road.
roadClassi	string	The road classification (A Road/motorway).
roadFuncti	string	The road function.
formOfWay	string	The way of the road.
length	float	The length of the road.
primaryRou	binary	1 indicates that it is a primary route. 0 indicates that it is a non-primary route.
distance to the nearest point on rd	float	The distance to the nearest point on the road.
postcode	string	The postcode of the area.

For the sample table, there are 2 columns

Variable	Data Type	Description
postcode	string	The postcode of the area.
Accident_risk_index	integer	The mean casualties at a postcode.

For the test and train table, there are 27 columns; both tables have the same variables.

Variable	Data Type	Description
Accident_ID	integer	The numbering of the datasets.
Police_Force	integer	The number of police forces in the area.
Number_of_Vehicles	integer	The number of vehicles.
Number_of_Casualties	integer	The number of casualties.
Date	date	Date when the accident happened.
Day_of_Week	integer	The day of the accident happened 1- Monday 2- Tuesday 3- Wednesday 4- Thursday 5- Friday 6- Saturday 7- Sunday
Time	time	Time the accident happened.
Local_Authority_(District)	integer	The number of local authorities within the district.
Local_Authority(Highway)	string	The local authority code of the highway.
1st_Road_Class	integer	The first road class.
1st_Road_Number	integer	The first road number.

Road_Type	string	The type of the roads.
Speed_limit	integer	The speed limit of the area.
2nd_Road_Class	integer	The second road class.
2nd_Road_Number	integer	The second road number.
Pedestrian_Crossing-Human_Control	string	Pedestrian crossing without using the facilities for crossing roads.
Pedestrian_Crossing-Physical_Facilities	string	Pedestrian crossing with the usage of facilities for crossing.
Light_Conditions	string	The light condition of the road.
Weather_Conditions	string	Weather conditions that consist of fine without high winds, fine with high winds, raining without high winds, raining with high winds and snowing without high winds.
Road_Surface_Conditions	string	The road surface consists of dry, wet/damp, frost/ice, snow and flood that over 3 cm of water.
Special_Conditions_at_Site	string	Whether there is roadworks or road surface defective.
Carriageway_Hazards	string	Carriageway hazards that involve during the accident.
Urban_or_Rural_Area	integer	1 indicates an urban area, 2 indicates a rural area.
Did_Police_Officer_Attend_Scene_of_Accident	string	Whether the police were on site when the accident happened.
state	string	States of the accident area.
postcode	string	Postcode of the accident area.
country	string	The country where the accidents occur which is the United Kingdom.

1.3 Problem to be Solved

Nowadays, the production of automobiles has been increasing quite significantly, the increase in the production of cars might just lead to multiple obstacles on the road and the road tends to be more vulnerable towards accidents. In this case study, we want to help in analysing factors that could lead to accident rates. Through this analysis, we hoped that it would help the authority to wisely arrange the roads and road users to take precautions to avoid casualties.

1.4 Objectives

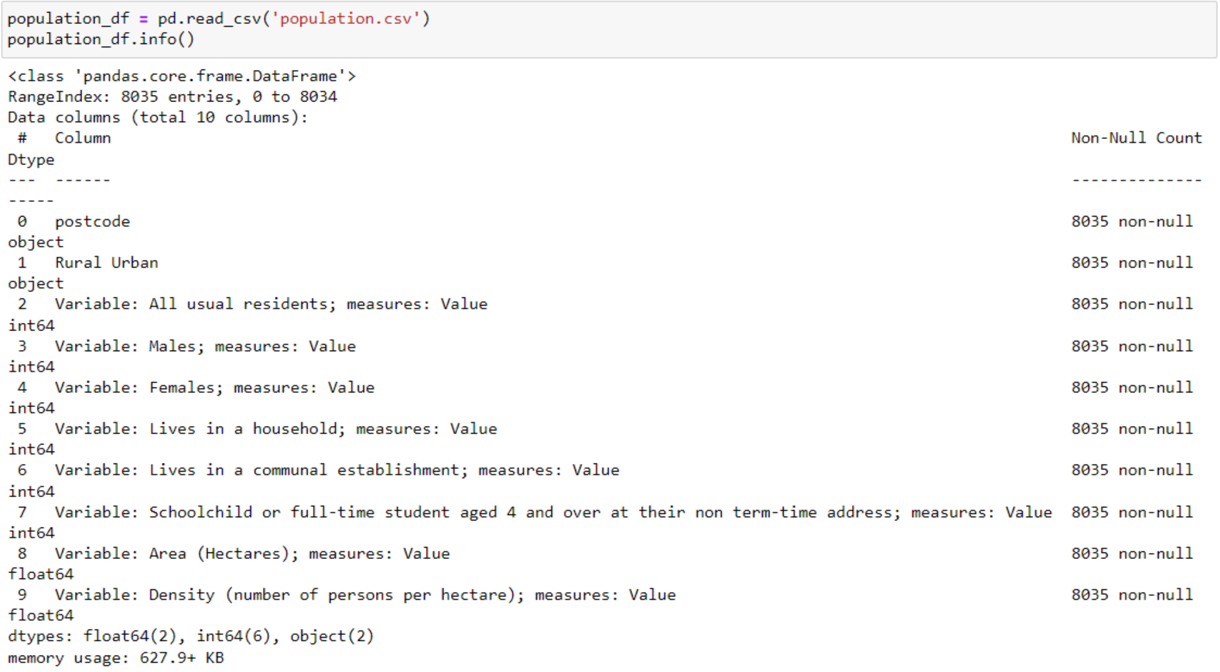
- To observe the state that has the highest accident rate.
- To identify factors that affect the accident rate.
- To observe the effectiveness of police authority on roads.

1.5 Data Schema

import pandas as pd

import numpy as np

- For population



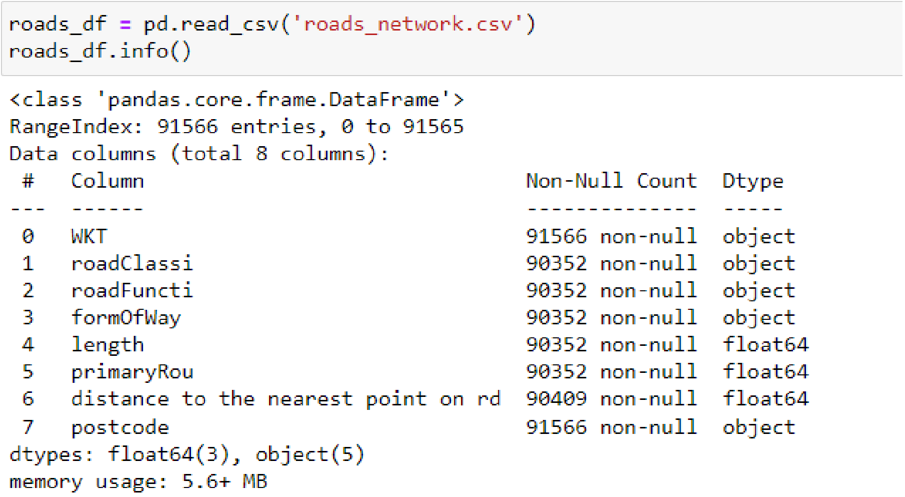
```
population_df = pd.read_csv('population.csv')
population_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8035 entries, 0 to 8034
Data columns (total 10 columns):
 #   Column                                                                                                      Non-Null Count
Dtype
---  ------                                                                                                      --------------
-----
 0   postcode                                                                                                    8035 non-null
object
 1   Rural Urban                                                                                                 8035 non-null
object
 2   Variable: All usual residents; measures: Value                                                              8035 non-null
int64
 3   Variable: Males; measures: Value                                                                            8035 non-null
int64
 4   Variable: Females; measures: Value                                                                          8035 non-null
int64
 5   Variable: Lives in a household; measures: Value                                                             8035 non-null
int64
 6   Variable: Lives in a communal establishment; measures: Value                                                8035 non-null
int64
 7   Variable: Schoolchild or full-time student aged 4 and over at their non term-time address; measures: Value  8035 non-null
int64
 8   Variable: Area (Hectares); measures: Value                                                                  8035 non-null
float64
 9   Variable: Density (number of persons per hectare); measures: Value                                          8035 non-null
float64
dtypes: float64(2), int64(6), object(2)
memory usage: 627.9+ KB
```

Figure 1.5.1: Population tables

Based on figure 1.5.1 above, the data schema for the population consists of ten attributes or columns. The data types for this data frame are string, integer and float. Two columns have the string data type, six columns have the integer data type and two columns have float data type.

- For roads network



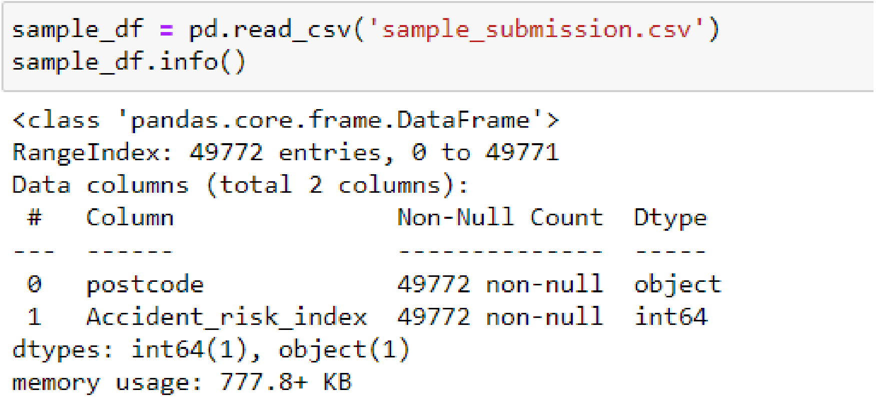
```
roads_df = pd.read_csv('roads_network.csv')
roads_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91566 entries, 0 to 91565
Data columns (total 8 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   WKT                              91566 non-null  object 
 1   roadClassi                       90352 non-null  object 
 2   roadFuncti                       90352 non-null  object 
 3   formOfWay                        90352 non-null  object 
 4   length                           90352 non-null  float64
 5   primaryRou                       90352 non-null  float64
 6   distance to the nearest point on rd  90409 non-null  float64
 7   postcode                         91566 non-null  object 
dtypes: float64(3), object(5)
memory usage: 5.6+ MB
```

Figure 1.5.2: Roads network table

Based on figure 1.5.2 above, the data schema for the roads network consists of eight attributes or columns. This data frame consists of only two data types which are string and float. Five columns have the string data type while three columns have float data type.

- For sample submission



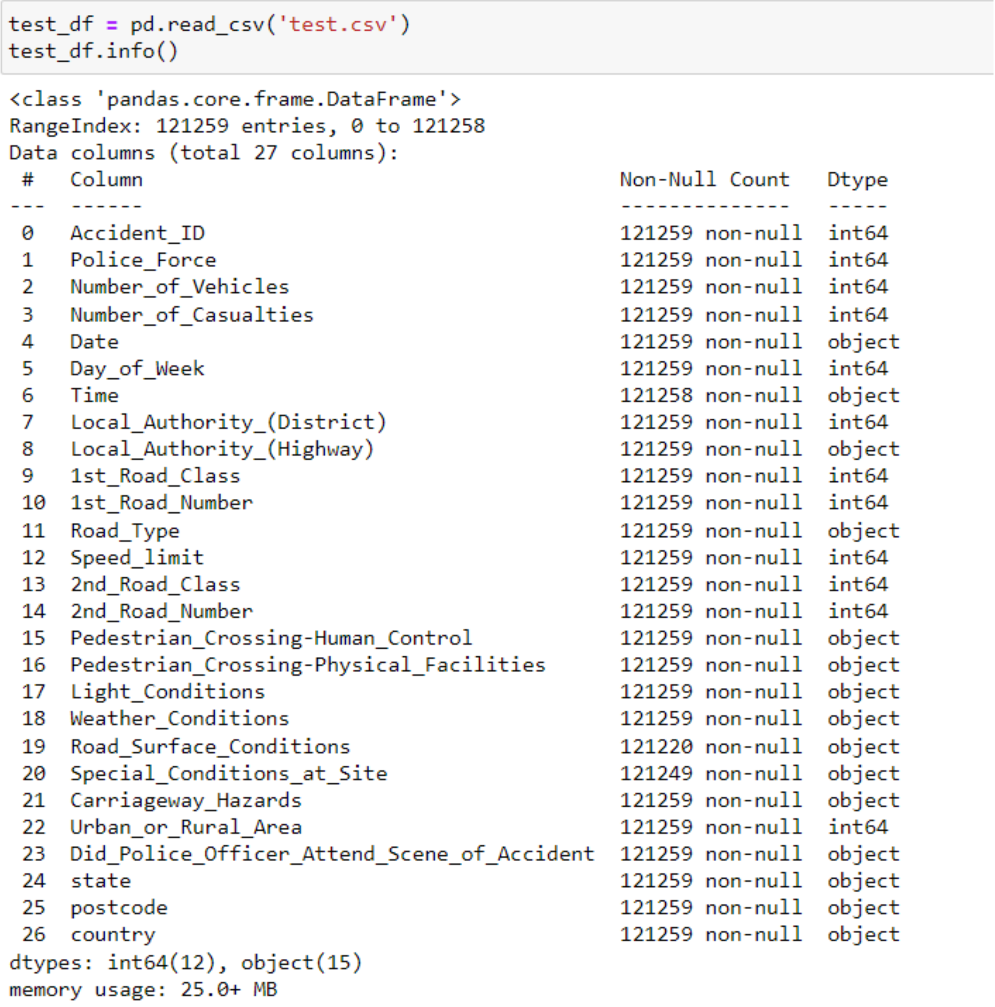
```
sample_df = pd.read_csv('sample_submission.csv')
sample_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49772 entries, 0 to 49771
Data columns (total 2 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   postcode             49772 non-null  object
 1   Accident_risk_index  49772 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 777.8+ KB
```

Figure 1.5.3: Sample table

Based on figure 1.5.3 above, the data schema for the sample submission consists of two attributes or columns. Postcode has string data type while Accident_risk_index has an integer data type.

- For test



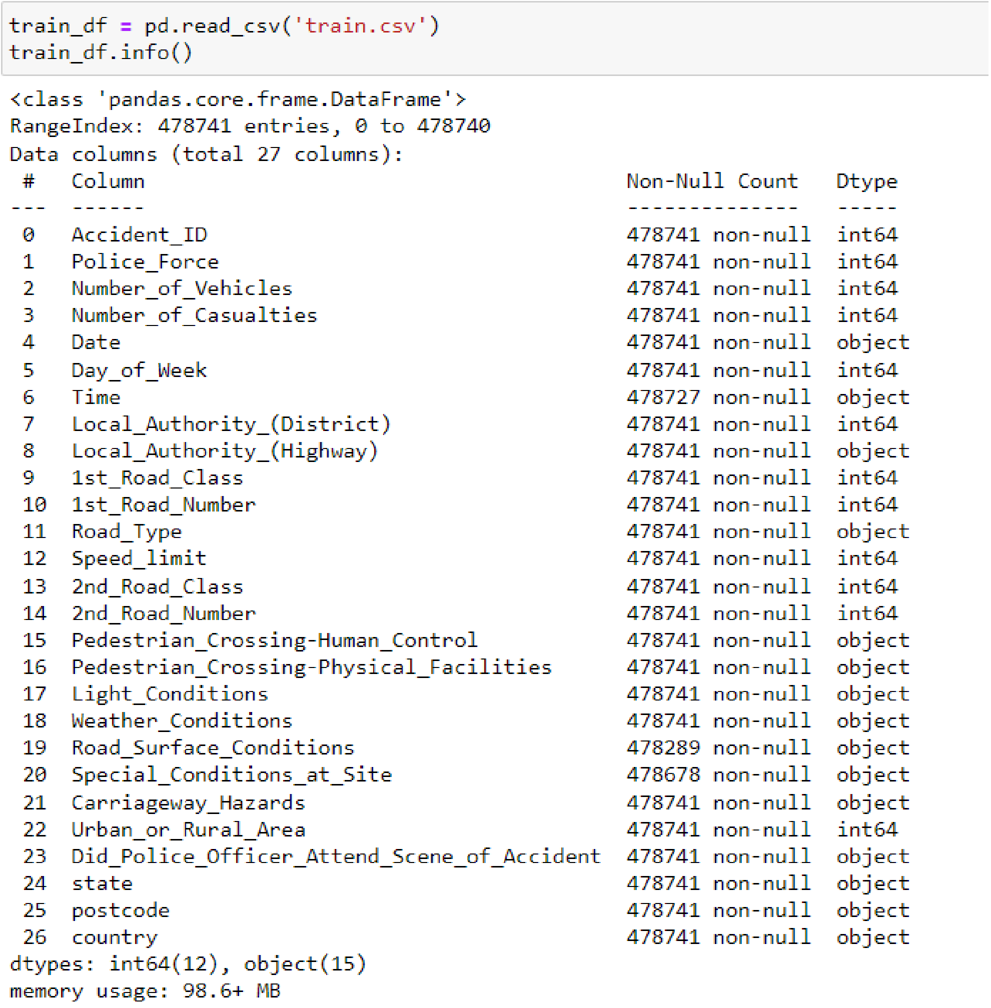
```python
test_df = pd.read_csv('test.csv')
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121259 entries, 0 to 121258
Data columns (total 27 columns):
 #   Column                                       Non-Null Count   Dtype 
---  ------                                       --------------   ----- 
 0   Accident_ID                                  121259 non-null  int64 
 1   Police_Force                                 121259 non-null  int64 
 2   Number_of_Vehicles                           121259 non-null  int64 
 3   Number_of_Casualties                         121259 non-null  int64 
 4   Date                                         121259 non-null  object
 5   Day_of_Week                                  121259 non-null  int64 
 6   Time                                         121258 non-null  object
 7   Local_Authority_(District)                   121259 non-null  int64 
 8   Local_Authority_(Highway)                    121259 non-null  object
 9   1st_Road_Class                               121259 non-null  int64 
 10  1st_Road_Number                              121259 non-null  int64 
 11  Road_Type                                    121259 non-null  object
 12  Speed_limit                                  121259 non-null  int64 
 13  2nd_Road_Class                               121259 non-null  int64 
 14  2nd_Road_Number                              121259 non-null  int64 
 15  Pedestrian_Crossing-Human_Control            121259 non-null  object
 16  Pedestrian_Crossing-Physical_Facilities      121259 non-null  object
 17  Light_Conditions                             121259 non-null  object
 18  Weather_Conditions                           121259 non-null  object
 19  Road_Surface_Conditions                      121220 non-null  object
 20  Special_Conditions_at_Site                   121249 non-null  object
 21  Carriageway_Hazards                          121259 non-null  object
 22  Urban_or_Rural_Area                          121259 non-null  int64 
 23  Did_Police_Officer_Attend_Scene_of_Accident  121259 non-null  object
 24  state                                        121259 non-null  object
 25  postcode                                     121259 non-null  object
 26  country                                      121259 non-null  object
dtypes: int64(12), object(15)
memory usage: 25.0+ MB
```

Figure 1.5.4: Test table

Based on figure 1.5.4 above, the data schema for the test consists of 27 attributes or columns. The data types for this data frame are string and integer. There are 15 columns that have the string data type while 12 columns have integer data type.

- For train



```
train_df = pd.read_csv('train.csv')
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 478741 entries, 0 to 478740
Data columns (total 27 columns):
 #   Column                                      Non-Null Count   Dtype 
---  ------                                      --------------   ----- 
 0   Accident_ID                                 478741 non-null  int64 
 1   Police_Force                                478741 non-null  int64 
 2   Number_of_Vehicles                          478741 non-null  int64 
 3   Number_of_Casualties                        478741 non-null  int64 
 4   Date                                        478741 non-null  object
 5   Day_of_Week                                 478741 non-null  int64 
 6   Time                                        478727 non-null  object
 7   Local_Authority_(District)                  478741 non-null  int64 
 8   Local_Authority_(Highway)                   478741 non-null  object
 9   1st_Road_Class                              478741 non-null  int64 
 10  1st_Road_Number                             478741 non-null  int64 
 11  Road_Type                                   478741 non-null  object
 12  Speed_limit                                 478741 non-null  int64 
 13  2nd_Road_Class                              478741 non-null  int64 
 14  2nd_Road_Number                             478741 non-null  int64 
 15  Pedestrian_Crossing-Human_Control           478741 non-null  object
 16  Pedestrian_Crossing-Physical_Facilities     478741 non-null  object
 17  Light_Conditions                            478741 non-null  object
 18  Weather_Conditions                          478741 non-null  object
 19  Road_Surface_Conditions                     478289 non-null  object
 20  Special_Conditions_at_Site                  478678 non-null  object
 21  Carriageway_Hazards                         478741 non-null  object
 22  Urban_or_Rural_Area                         478741 non-null  int64 
 23  Did_Police_Officer_Attend_Scene_of_Accident 478741 non-null  object
 24  state                                       478741 non-null  object
 25  postcode                                    478741 non-null  object
 26  country                                     478741 non-null  object
dtypes: int64(12), object(15)
memory usage: 98.6+ MB
```

Figure 1.5.5: Train table

Based on figure 1.5.5 above, the data schema for the test consists of 27 attributes or columns. This data frame consists only of string and integer. There are 15 columns that have the string data type while 12 columns have integer data type.

2.0 ARCHITECTURE

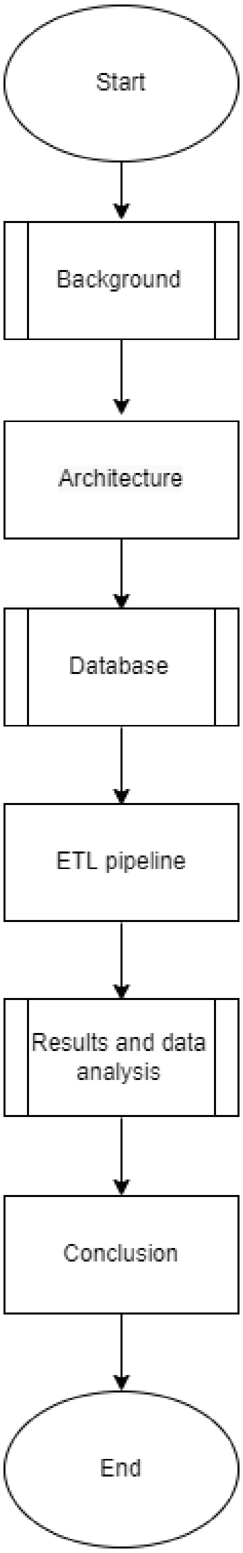


Figure 2.1 : Flow of the project

Figure 2.1 shows the general process of the project which our group will do six stages in this project to be done.

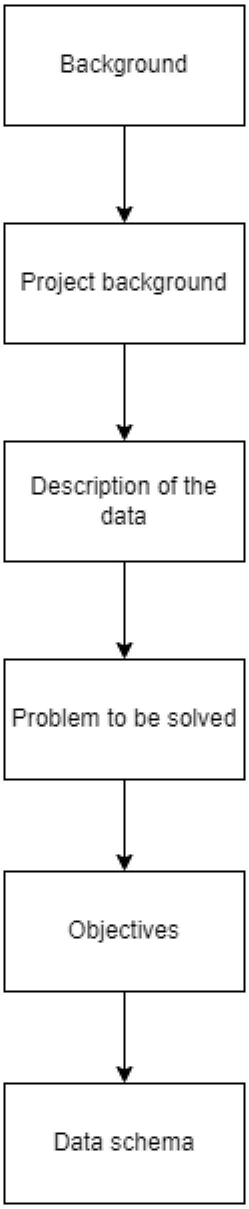


Figure 2.2 : Process of the background

Figure 2.2 shows the background process that includes project background, description of the dta, problem to be solved, objectives and data schema. The data used in this project was obtained from Kaggle. After that, the architecture was created so that the project will run according to the plan and run smoothly.

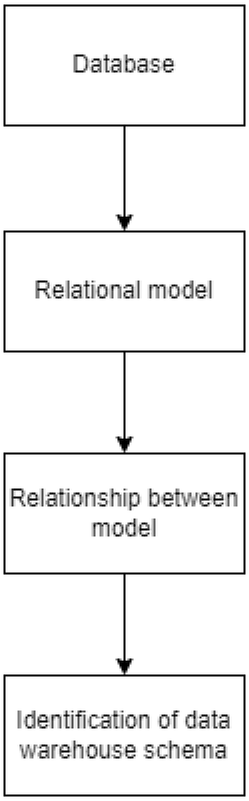

Database
Relational model
Relationship between model
Identification of data warehouse schema

Figure 2.3 : Process for the database

Figure 2.3 shows the process of the database including relational model, relationship between model and identification of the data warehouse schema. In this project, our group uses Microsoft Power BI to create the relational model. After that, we proceed with the Extract, Transform and Load (ETL) pipeline by using Jupyter Notebook and pgAdmin. The raw data will be extracted to the Jupyter Notebook and transformed by doing the cleaning. After that, we load the clean data to the pgAdmin to do the analysis.

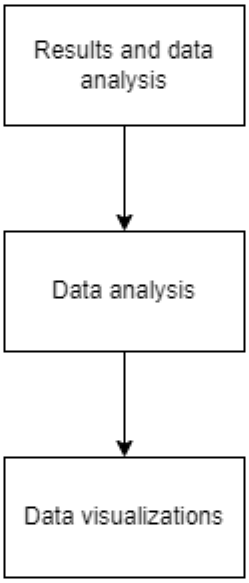


Figure 2.4 : Process for the results and data analysis

Figure 2.4 shows the results and data analysis that includes the data analysis and data visualisation. For the data analysis, we will use pgAdmin to do the analysis such as roll up and slicing. After that, we perform the data visualisation using the Microsoft Power BI. For the conclusion, we will conclude based on the data analysis and data visualisation obtained.

3.0 DATABASE

3.1 Relational Model

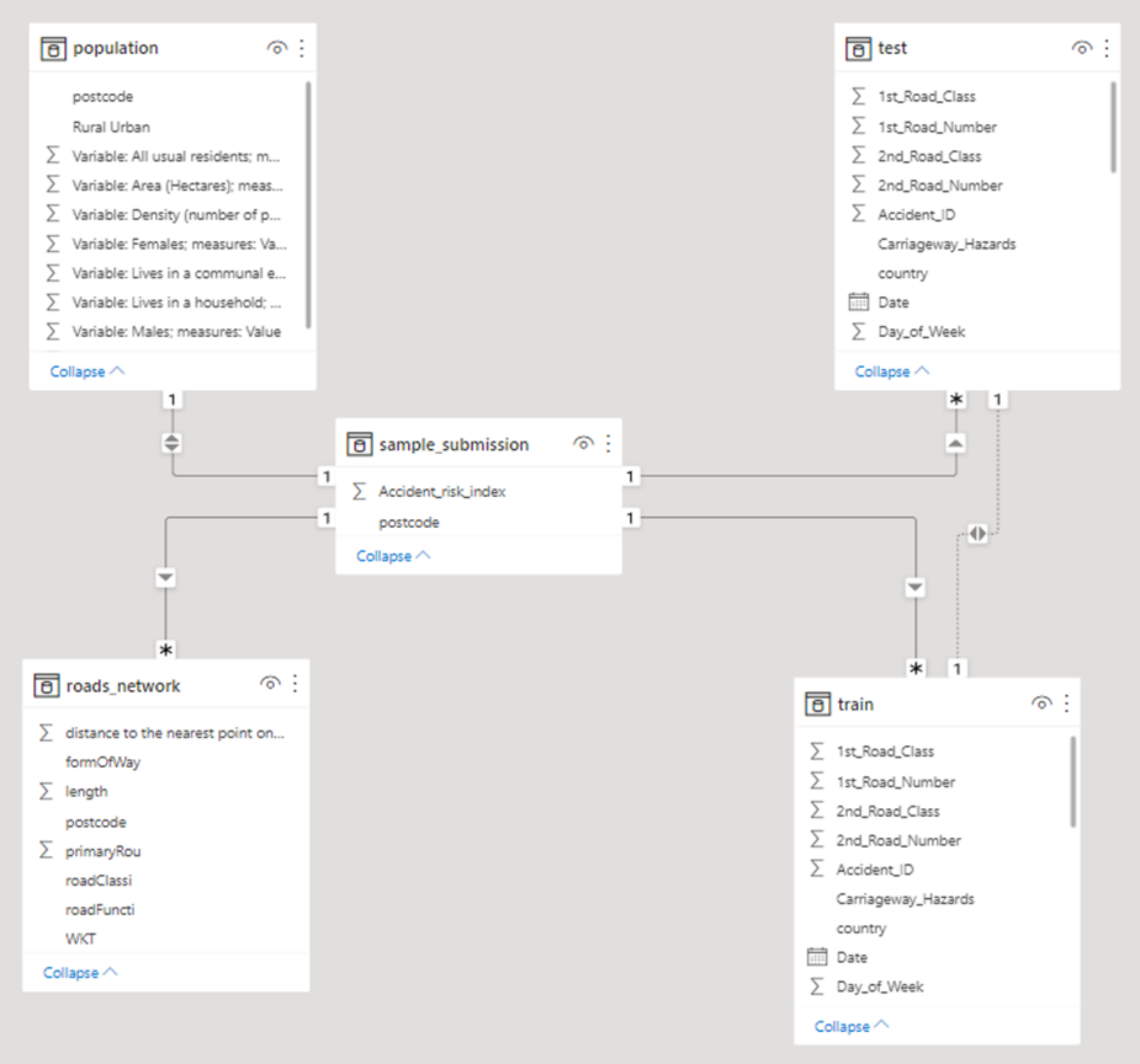


Figure 3.1 Relational Model

3.2 Relationship between Data

Data	Relationship
sample_submission -> population	one-to-one
sample_submission -> roads_network	one-to-many
sample_submission -> test	one-to-many
sample_submission -> train	one-to-many
test -> train	one-to-one

3.3 Identification of Data Warehouse Schema

Based on the figure 3.1 above, the data warehouse schema of these datasets is Snowflake Schema because it has one fact table that is connected to a four dimensions table. The fact table is sample_submission and the dimensional tables are population, roads_network, test, and train.

4.0 ETL PIPELINE

4.1 ETL Pipeline

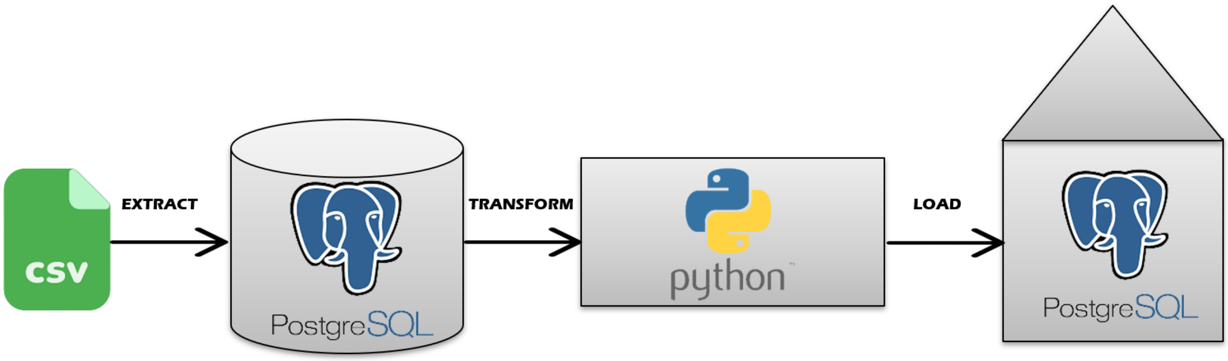


Figure 4.1 ETL Pipeline

Figure 4.1 shows the pipeline of Extract, Transform, Load (ETL) for the dataset of accident risk index. In the ETL process, it can extract data from various data sources, transform the data, and then load the data into the Data Warehouse System. In this project, we used PostgreSQL to extract the data from csv file, transform the data using Python by connecting the PostgreSQL with the Jupyter Notebook, and then load the clean data into the PostgreSQL back.

4.2 ETL Process

4.2.1 Extract

Before starting the ETL process the datasets need to be stored into a database which is PostgreSQL. Firstly, create a new database and then create tables by using the syntax below:

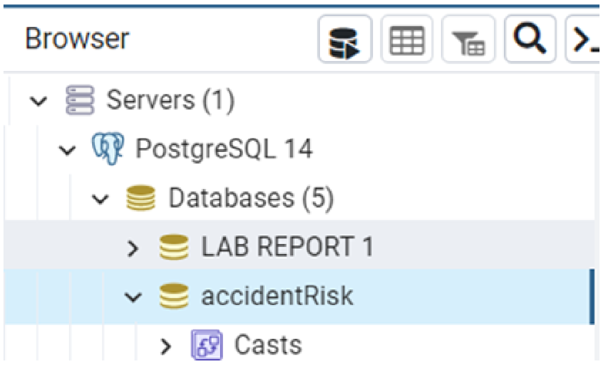


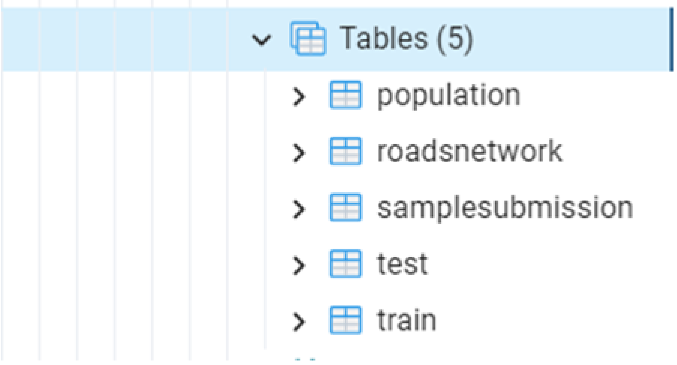
Figure 4.2.1.1 Database in PostgreSQL

Figure 4.2.1 shows that we have created a database named ‘accidentRisk’ in PostgreSQL.

Query to create tables:

```
CREATE TABLE population
 (
       Postcode text,
       RuralUrban text,
       Residents numeric,
       Males numeric,
       Females numeric,
       Household numeric,
       Cmmunal numeric,
       Students numeric,
       Area numeric,
       DensityPersons numeric
 );

CREATE TABLE roadsNetwork
 (
       WKT text,
       RoadClassi text,
       RoadFunction text,
       FormOfWay text,
       RoadLength numeric,
       PrimaryRou numeric,
       Distance numeric,
       Postcode text
 );

CREATE TABLE sampleSubmission
 (
       Postcode text,
       AcidentRiskIndex numeric
 );

CREATE TABLE test
 (
       AccidentID int,
       PoliceForce int,
       NoOfVehicles int,
       NoOfCasualties int,
       AccidentDate date,
       DayOfWeek int,
       AccidentTime time,
       AuthorityDistrict text,
       AuthorityHighway text,
       FirstRoadClass int,
       FirstRoadNumber int,
       RoadType text,
       SpeedLimit int,
       SecondRoadClass int,
       SecondRoadNumber int,
       PedestrianCrossing_HumanControl text,
       PedestrinCrossing_PhysicalFacilities text,
       LightCondition text,
       WeatherCondition text,
       RoadSurfaceCondition text,
       SpecialConditionAtSite text,
       Carriageway_Hazards text,
       UrbanOrRural int,
       PoliceOnSite text,
       State text,
       Postcode text,
       Country text
 );

CREATE TABLE train
 (
       AccidentID int,
       PoliceForce int,
       NoOfVehicles int,
       NoOfCasualties int,
       AccidentDate date,
       DayOfWeek int,
       AccidentTime time,
       AuthorityDistrict text,
       AuthorityHighway text,
       FirstRoadClass int,
       FirstRoadNumber int,
       RoadType text,
       SpeedLimit int,
       SecondRoadClass int,
       SecondRoadNumber int,
       PedestrianCrossing_HumanControl text,
       PedestrinCrossing_PhysicalFacilities text,
       LightCondition text,
       WeatherCondition text,
       RoadSurfaceCondition text,
       SpecialConditionAtSite text,
       Carriageway_Hazards text,
       UrbanOrRural int,
       PoliceOnSite text,
       State text,
       Postcode text,
       Country text
```

```
);
```



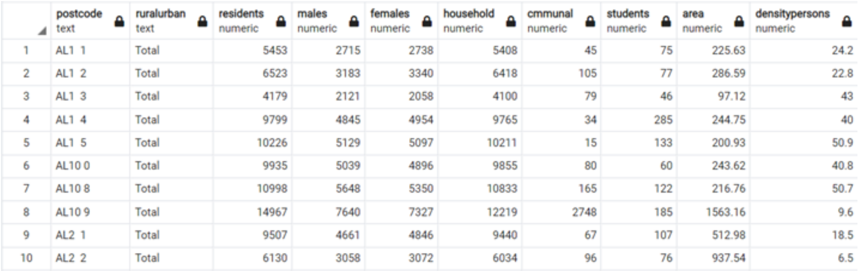
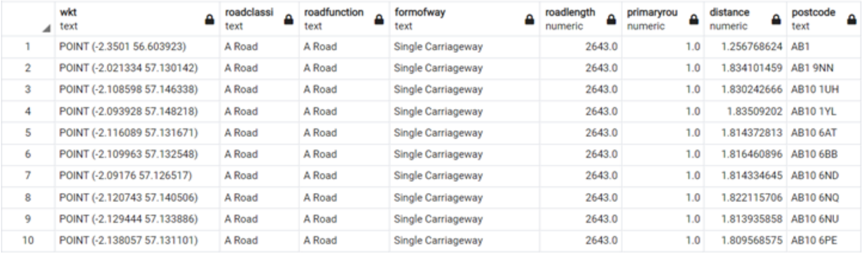
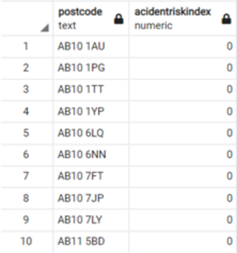
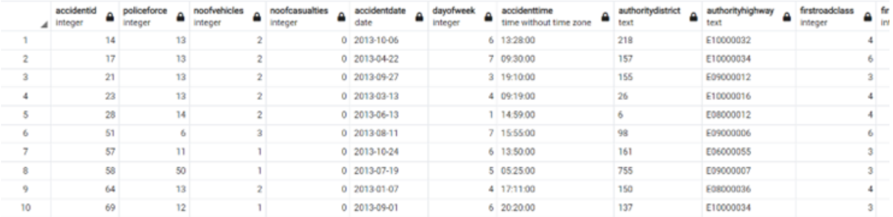
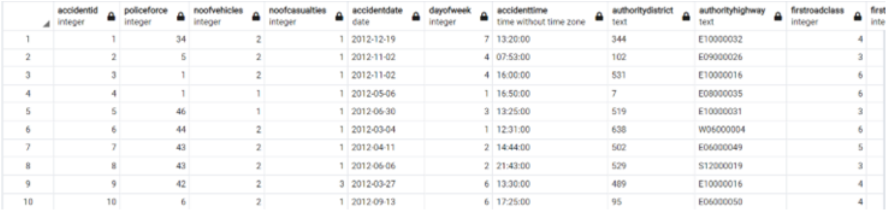
Tables (5)
population
roadsnetwork
samplesubmission
test
train

Figure 4.2.1.2 Tables successfully created

Query to copy the data from csv file into table:

```
COPY population
FROM 'D:\population.csv'
DELIMITER ','
CSV HEADER;

COPY roadsNetwork
FROM 'D:\roads_network.csv'
DELIMITER ','
CSV HEADER;

COPY sampleSubmission
FROM 'D:\sample_submission.csv'
DELIMITER ','
CSV HEADER;

COPY test
FROM 'D:\test.csv'
DELIMITER ','
CSV HEADER;

COPY train
FROM 'D:\train.csv'
DELIMITER ','
CSV HEADER;
```

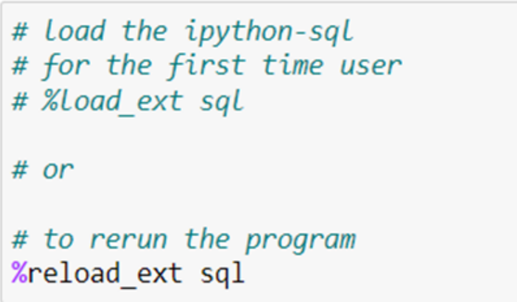
Run a query (Select * from {table_name}) to view the data in the table.

<table>
<tr><th>Query</th><th>Output</th></tr>
<tr><td>SELECT * FROM population;</td><td>

<table>
<tr><th></th><th>postcode
text</th><th>ruralurban
text</th><th>residents
numeric</th><th>males
numeric</th><th>females
numeric</th><th>household
numeric</th><th>cmmunal
numeric</th><th>students
numeric</th><th>area
numeric</th><th>densitypersons
numeric</th></tr>
<tr><td>1</td><td>AL1 1</td><td>Total</td><td>5453</td><td>2715</td><td>2738</td><td>5408</td><td>45</td><td>75</td><td>225.63</td><td>24.2</td></tr>
<tr><td>2</td><td>AL1 2</td><td>Total</td><td>6523</td><td>3183</td><td>3340</td><td>6418</td><td>105</td><td>77</td><td>286.59</td><td>22.8</td></tr>
<tr><td>3</td><td>AL1 3</td><td>Total</td><td>4179</td><td>2121</td><td>2058</td><td>4100</td><td>79</td><td>46</td><td>97.12</td><td>43</td></tr>
<tr><td>4</td><td>AL1 4</td><td>Total</td><td>9799</td><td>4845</td><td>4954</td><td>9765</td><td>34</td><td>285</td><td>244.75</td><td>40</td></tr>
<tr><td>5</td><td>AL1 5</td><td>Total</td><td>10226</td><td>5129</td><td>5097</td><td>10211</td><td>15</td><td>133</td><td>200.93</td><td>50.9</td></tr>
<tr><td>6</td><td>AL10 0</td><td>Total</td><td>9935</td><td>5039</td><td>4896</td><td>9855</td><td>80</td><td>60</td><td>243.62</td><td>40.8</td></tr>
<tr><td>7</td><td>AL10 8</td><td>Total</td><td>10998</td><td>5648</td><td>5350</td><td>10833</td><td>165</td><td>122</td><td>216.76</td><td>50.7</td></tr>
<tr><td>8</td><td>AL10 9</td><td>Total</td><td>14967</td><td>7640</td><td>7327</td><td>12219</td><td>2748</td><td>185</td><td>1563.16</td><td>9.6</td></tr>
<tr><td>9</td><td>AL2 1</td><td>Total</td><td>9507</td><td>4661</td><td>4846</td><td>9440</td><td>67</td><td>107</td><td>512.98</td><td>18.5</td></tr>
<tr><td>10</td><td>AL2 2</td><td>Total</td><td>6130</td><td>3058</td><td>3072</td><td>6034</td><td>96</td><td>76</td><td>937.54</td><td>6.5</td></tr>
</table>
</td></tr>
<tr><td>SELECT * FROM roadsnetwork;</td><td>

<table>
<tr><th></th><th>wkt
text</th><th>roadclassi
text</th><th>roadfunction
text</th><th>formofway
text</th><th>roadlength
numeric</th><th>primaryrou
numeric</th><th>distance
numeric</th><th>postcode
text</th></tr>
<tr><td>1</td><td>POINT (-2.3501 56.603923)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.256768624</td><td>AB1</td></tr>
<tr><td>2</td><td>POINT (-2.021334 57.130142)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.834101459</td><td>AB1 9NN</td></tr>
<tr><td>3</td><td>POINT (-2.108598 57.146338)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.830242666</td><td>AB10 1UH</td></tr>
<tr><td>4</td><td>POINT (-2.093928 57.148218)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.83509202</td><td>AB10 1YL</td></tr>
<tr><td>5</td><td>POINT (-2.116089 57.131671)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.814372813</td><td>AB10 6AT</td></tr>
<tr><td>6</td><td>POINT (-2.109963 57.132548)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.816460896</td><td>AB10 6BB</td></tr>
<tr><td>7</td><td>POINT (-2.09176 57.126517)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.814334645</td><td>AB10 6ND</td></tr>
<tr><td>8</td><td>POINT (-2.120743 57.140506)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.822115706</td><td>AB10 6NQ</td></tr>
<tr><td>9</td><td>POINT (-2.129444 57.133886)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.813935858</td><td>AB10 6NU</td></tr>
<tr><td>10</td><td>POINT (-2.138057 57.131101)</td><td>A Road</td><td>A Road</td><td>Single Carriageway</td><td>2643.0</td><td>1.0</td><td>1.809568575</td><td>AB10 6PE</td></tr>
</table>
</td></tr>
<tr><td>SELECT * FROM samplesubmission;</td><td>

<table>
<tr><th></th><th>postcode
text</th><th>acidentriskindex
numeric</th></tr>
<tr><td>1</td><td>AB10 1AU</td><td>0</td></tr>
<tr><td>2</td><td>AB10 1PG</td><td>0</td></tr>
<tr><td>3</td><td>AB10 1TT</td><td>0</td></tr>
<tr><td>4</td><td>AB10 1YP</td><td>0</td></tr>
<tr><td>5</td><td>AB10 6LQ</td><td>0</td></tr>
<tr><td>6</td><td>AB10 6NN</td><td>0</td></tr>
<tr><td>7</td><td>AB10 7FT</td><td>0</td></tr>
<tr><td>8</td><td>AB10 7JP</td><td>0</td></tr>
<tr><td>9</td><td>AB10 7LY</td><td>0</td></tr>
<tr><td>10</td><td>AB11 5BD</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>SELECT * FROM test;</td><td>

<table>
<tr><th></th><th>accidentid
integer</th><th>policeforce
integer</th><th>noofvehicles
integer</th><th>noofcasualties
integer</th><th>accidentdate
date</th><th>dayofweek
integer</th><th>accidenttime
time without time zone</th><th>authoritydistrict
text</th><th>authorityhighway
text</th><th>firstroadclass
integer</th><th>fir
in</th></tr>
<tr><td>1</td><td>14</td><td>13</td><td>2</td><td>0</td><td>2013-10-06</td><td>6</td><td>13:28:00</td><td>218</td><td>E10000032</td><td>4</td><td></td></tr>
<tr><td>2</td><td>17</td><td>13</td><td>2</td><td>0</td><td>2013-04-22</td><td>7</td><td>09:30:00</td><td>157</td><td>E10000034</td><td>6</td><td></td></tr>
<tr><td>3</td><td>21</td><td>13</td><td>2</td><td>0</td><td>2013-09-27</td><td>3</td><td>19:10:00</td><td>155</td><td>E09000012</td><td>3</td><td></td></tr>
<tr><td>4</td><td>23</td><td>13</td><td>2</td><td>0</td><td>2013-03-13</td><td>4</td><td>09:19:00</td><td>26</td><td>E10000016</td><td>4</td><td></td></tr>
<tr><td>5</td><td>28</td><td>14</td><td>2</td><td>0</td><td>2013-06-13</td><td>1</td><td>14:59:00</td><td>6</td><td>E08000012</td><td>4</td><td></td></tr>
<tr><td>6</td><td>51</td><td>6</td><td>3</td><td>0</td><td>2013-08-11</td><td>7</td><td>15:55:00</td><td>98</td><td>E09000006</td><td>6</td><td></td></tr>
<tr><td>7</td><td>57</td><td>11</td><td>1</td><td>0</td><td>2013-10-24</td><td>6</td><td>13:50:00</td><td>161</td><td>E06000055</td><td>3</td><td></td></tr>
<tr><td>8</td><td>58</td><td>50</td><td>1</td><td>0</td><td>2013-07-19</td><td>5</td><td>05:25:00</td><td>755</td><td>E09000007</td><td>3</td><td></td></tr>
<tr><td>9</td><td>64</td><td>13</td><td>2</td><td>0</td><td>2013-01-07</td><td>4</td><td>17:11:00</td><td>150</td><td>E08000036</td><td>4</td><td></td></tr>
<tr><td>10</td><td>69</td><td>12</td><td>1</td><td>0</td><td>2013-09-01</td><td>6</td><td>20:20:00</td><td>137</td><td>E10000034</td><td>3</td><td></td></tr>
</table>
</td></tr>
<tr><td>SELECT * FROM train;</td><td>

<table>
<tr><th></th><th>accidentid
integer</th><th>policeforce
integer</th><th>noofvehicles
integer</th><th>noofcasualties
integer</th><th>accidentdate
date</th><th>dayofweek
integer</th><th>accidenttime
time without time zone</th><th>authoritydistrict
text</th><th>authorityhighway
text</th><th>firstroadclass
integer</th><th>first
inte</th></tr>
<tr><td>1</td><td>1</td><td>34</td><td>2</td><td>1</td><td>2012-12-19</td><td>7</td><td>13:20:00</td><td>344</td><td>E10000032</td><td>4</td><td></td></tr>
<tr><td>2</td><td>2</td><td>5</td><td>2</td><td>1</td><td>2012-11-02</td><td>4</td><td>07:53:00</td><td>102</td><td>E09000026</td><td>3</td><td></td></tr>
<tr><td>3</td><td>3</td><td>1</td><td>2</td><td>1</td><td>2012-11-02</td><td>4</td><td>16:00:00</td><td>531</td><td>E10000016</td><td>6</td><td></td></tr>
<tr><td>4</td><td>4</td><td>1</td><td>1</td><td>1</td><td>2012-05-06</td><td>1</td><td>16:50:00</td><td>7</td><td>E08000035</td><td>6</td><td></td></tr>
<tr><td>5</td><td>5</td><td>46</td><td>1</td><td>1</td><td>2012-06-30</td><td>3</td><td>13:25:00</td><td>519</td><td>E10000031</td><td>3</td><td></td></tr>
<tr><td>6</td><td>6</td><td>44</td><td>2</td><td>1</td><td>2012-03-04</td><td>1</td><td>12:31:00</td><td>638</td><td>W06000004</td><td>6</td><td></td></tr>
<tr><td>7</td><td>7</td><td>43</td><td>2</td><td>1</td><td>2012-04-11</td><td>2</td><td>14:44:00</td><td>502</td><td>E06000049</td><td>5</td><td></td></tr>
<tr><td>8</td><td>8</td><td>43</td><td>2</td><td>1</td><td>2012-06-06</td><td>2</td><td>21:43:00</td><td>529</td><td>S12000019</td><td>3</td><td></td></tr>
<tr><td>9</td><td>9</td><td>42</td><td>2</td><td>3</td><td>2012-03-27</td><td>6</td><td>13:30:00</td><td>489</td><td>E10000016</td><td>4</td><td></td></tr>
<tr><td>10</td><td>10</td><td>6</td><td>2</td><td>1</td><td>2012-09-13</td><td>6</td><td>17:25:00</td><td>95</td><td>E06000050</td><td>4</td><td></td></tr>
</table>
</td></tr>
</table>

After the raw data has been extracted into PostgreSQL, we need to connect our PostgreSQL with Jupyter Notebook to proceed with the next process which transforms the data. Before starting the process, we are required to install a few packages.

1. Pip install ipython-sql
2. Pip install sqlalchemy
3. Pip install pyscopg2

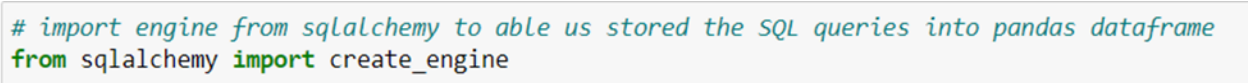
After install all these packages, we need to load ipython-sql using the following command:



```
# load the ipython-sql
# for the first time user
# %load_ext sql

# or

# to rerun the program
%reload_ext sql
```

Figure 4.2.1.3 Load ipython-sql

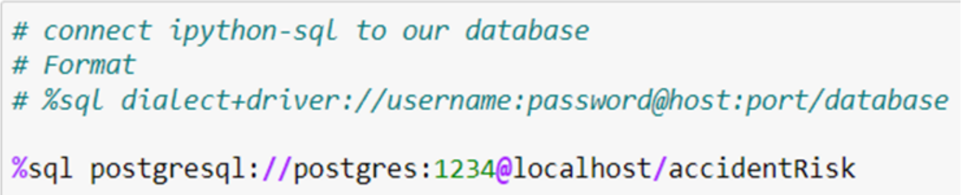
Call the create engine function:



```
# import engine from sqlalchemy to able us stored the SQL queries into pandas dataframe
from sqlalchemy import create_engine
```

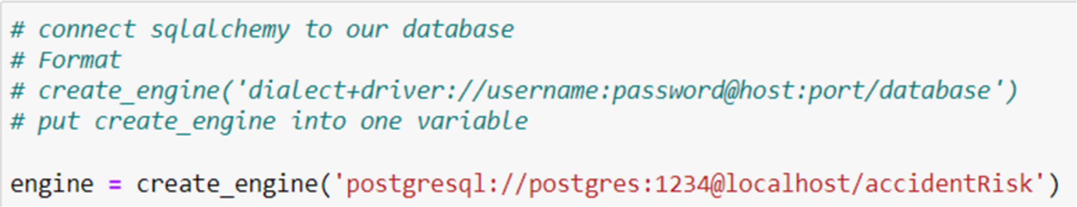
Figure 4.2.1.4 Call create engine

Connect ipython-sql and sqlalchemy with our database:



```
# connect ipython-sql to our database
# Format
# %sql dialect+driver://username:password@host:port/database

%sql postgresql://postgres:1234@localhost/accidentRisk
```

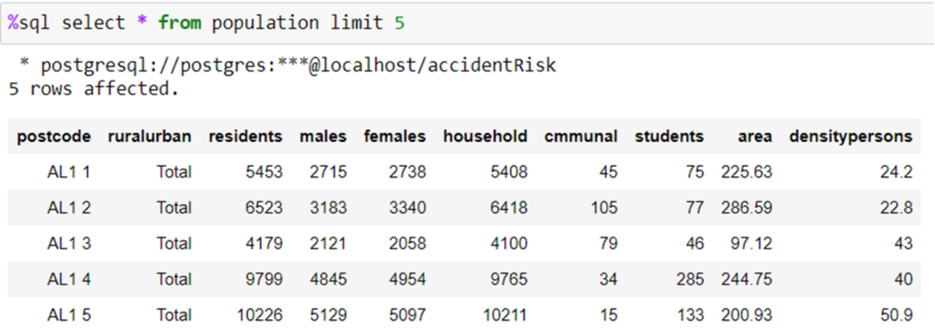
Figure 4.2.1.5 Connect ipython-sql



```
# connect sqlalchemy to our database
# Format
# create_engine('dialect+driver://username:password@host:port/database')
# put create_engine into one variable

engine = create_engine('postgresql://postgres:1234@localhost/accidentRisk')
```

Figure 4.2.1.6 Connect sqlalchemy

After all these commands successfully run without any error, we can check the connection between PostgreSQL and Python by print some of the data from the table:



```
%sql select * from population limit 5
```

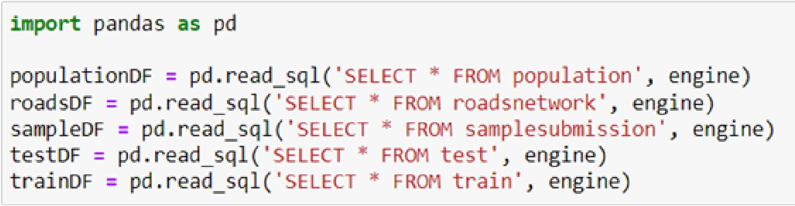
```
 * postgresql://postgres:***@localhost/accidentRisk
5 rows affected.
```

postcode	ruralurban	residents	males	females	household	cmmunal	students	area	densitypersons
AL1 1	Total	5453	2715	2738	5408	45	75	225.63	24.2
AL1 2	Total	6523	3183	3340	6418	105	77	286.59	22.8
AL1 3	Total	4179	2121	2058	4100	79	46	97.12	43
AL1 4	Total	9799	4845	4954	9765	34	285	244.75	40
AL1 5	Total	10226	5129	5097	10211	15	133	200.93	50.9

Figure 4.2.1.7 Check table

4.2.2 Transforms

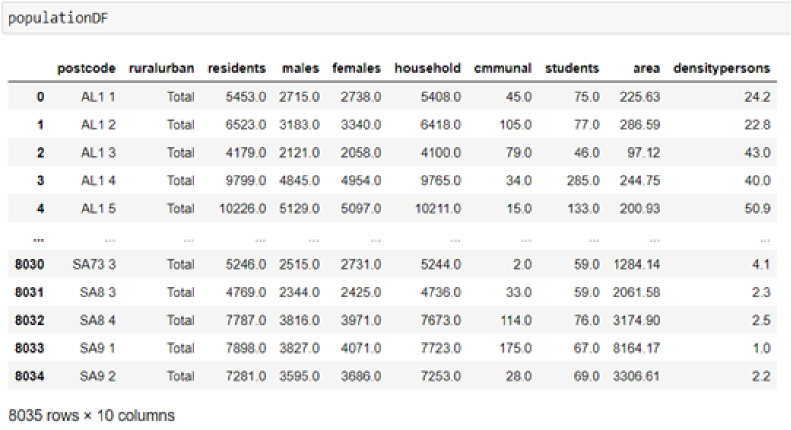
After the connecting process, then the data need to be clean as the data cleaning process is a crucial part in data processing. Some connectors were installed to ensure that data can be transferred from PostgreSQL to Python. To make it easy for the cleaning process, the data can be stored into data frame using pandas library:



```
import pandas as pd

populationDF = pd.read_sql('SELECT * FROM population', engine)
roadsDF = pd.read_sql('SELECT * FROM roadsnetwork', engine)
sampleDF = pd.read_sql('SELECT * FROM samplesubmission', engine)
testDF = pd.read_sql('SELECT * FROM test', engine)
trainDF = pd.read_sql('SELECT * FROM train', engine)
```

Figure 4.2.2.1 Store into Data Frames

Check the process:



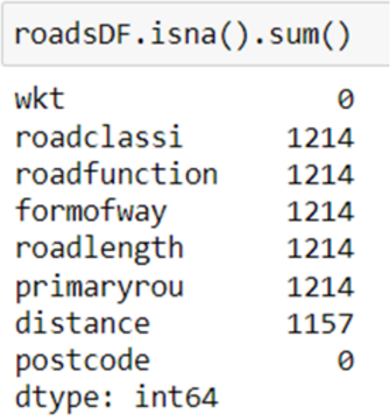
```
populationDF
```

	postcode	ruralurban	residents	males	females	household	cmmunal	students	area	densitypersons
0	AL1 1	Total	5453.0	2715.0	2738.0	5408.0	45.0	75.0	225.63	24.2
1	AL1 2	Total	6523.0	3183.0	3340.0	6418.0	105.0	77.0	286.59	22.8
2	AL1 3	Total	4179.0	2121.0	2058.0	4100.0	79.0	46.0	97.12	43.0
3	AL1 4	Total	9799.0	4845.0	4954.0	9765.0	34.0	285.0	244.75	40.0
4	AL1 5	Total	10226.0	5129.0	5097.0	10211.0	15.0	133.0	200.93	50.9
...	...	...	...	...	...	...	...	...	...	...
8030	SA73 3	Total	5246.0	2515.0	2731.0	5244.0	2.0	59.0	1284.14	4.1
8031	SA8 3	Total	4769.0	2344.0	2425.0	4736.0	33.0	59.0	2061.58	2.3
8032	SA8 4	Total	7787.0	3816.0	3971.0	7673.0	114.0	76.0	3174.90	2.5
8033	SA9 1	Total	7898.0	3827.0	4071.0	7723.0	175.0	67.0	8164.17	1.0
8034	SA9 2	Total	7281.0	3595.0	3686.0	7253.0	28.0	69.0	3306.61	2.2

8035 rows × 10 columns

Figure 4.2.2.2 Data Frame

After the data successfully stored into the data frame, then we can start the cleaning process by checking the null values first:

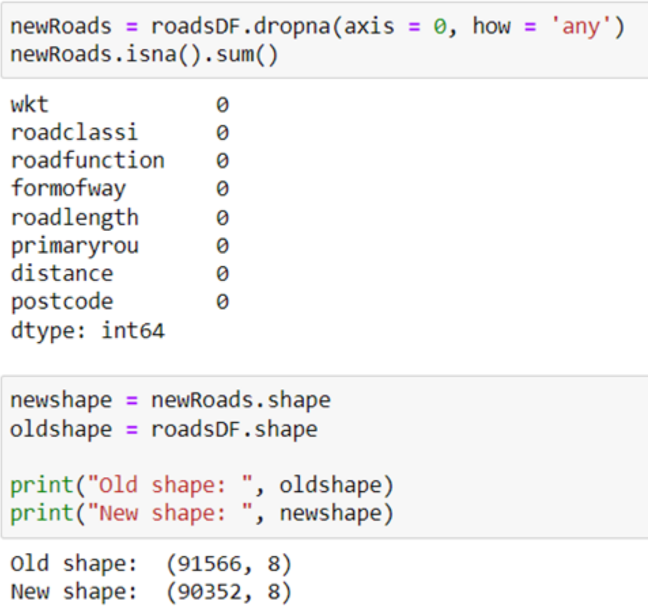


```
roadsDF.isna().sum()
```

```
wkt                0
roadclassi      1214
roadfunction    1214
formofway       1214
roadlength      1214
primaryrou      1214
distance        1157
postcode           0
dtype: int64
```

Figure 4.2.2.3 Checking Null

If the dataset contains the null values, drop the null values and check whether the null values has been dropped or not using these commands:



```
newRoads = roadsDF.dropna(axis = 0, how = 'any')
newRoads.isna().sum()
```

```
wkt             0
roadclassi      0
roadfunction    0
formofway       0
roadlength      0
primaryrou      0
distance        0
postcode        0
dtype: int64
```

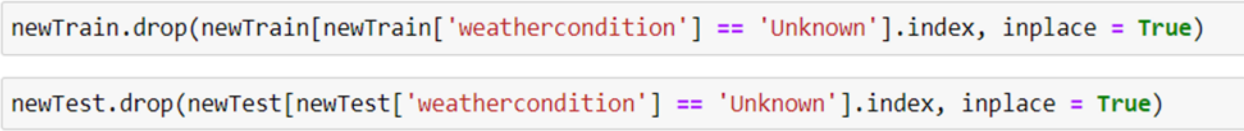
```
newshape = newRoads.shape
oldshape = roadsDF.shape

print("Old shape: ", oldshape)
print("New shape: ", newshape)
```

```
Old shape:  (91566, 8)
New shape:  (90352, 8)
```

Figure 4.2.2.4 Drop and Check Null Values

Repeat the process for all the data frames.

Other than checking for the null values, the cleaning process also required us to view and study the data in order for us to delete or drop the unnecessary data or column in the dataset. After viewing the data, we decide to remove the unknown condition of weather in the train and test table. This is because we want to analyze the weather condition that can cause the accident.

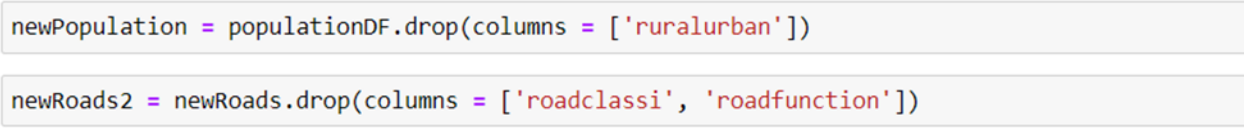


```
newTrain.drop(newTrain[newTrain['weathercondition'] == 'Unknown'].index, inplace = True)
```

```
newTest.drop(newTest[newTest['weathercondition'] == 'Unknown'].index, inplace = True)
```

Figure 4.2.2.5 Remove Unknown Weather

We also decide to drop the columns from the population and roads table that has no useful information.

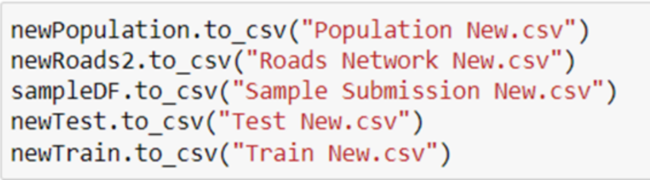


```
newPopulation = populationDF.drop(columns = ['ruralurban'])
```

```
newRoads2 = newRoads.drop(columns = ['roadclassi', 'roadfunction'])
```

Figure 4.2.2.6 Drop Columns

Last step in the transforms process, after the data has been cleaned, the data need to be store into the new csv file to proceed with the analysis with the cleaned data.



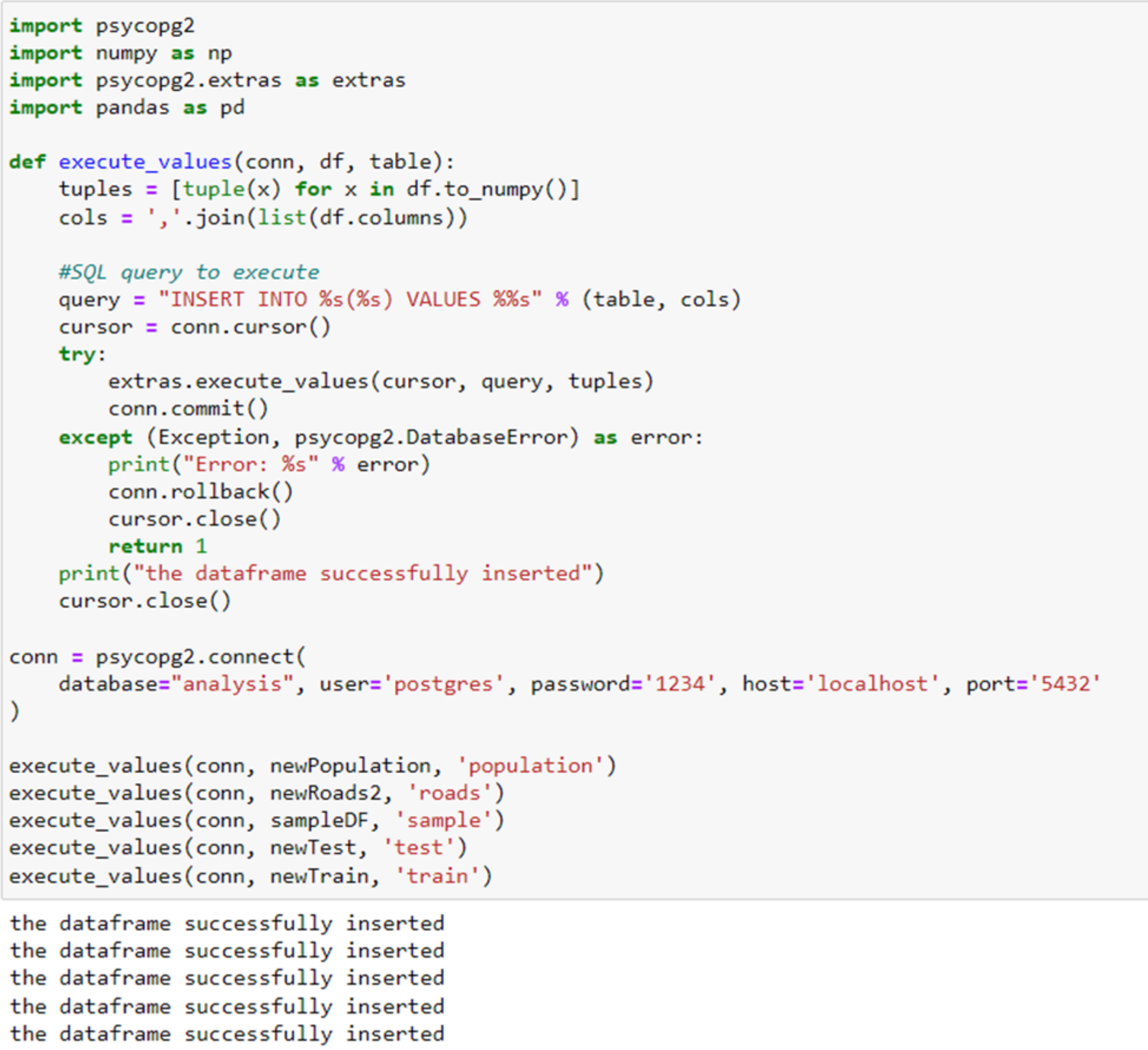
```
newPopulation.to_csv("Population New.csv")
newRoads2.to_csv("Roads Network New.csv")
sampleDF.to_csv("Sample Submission New.csv")
newTest.to_csv("Test New.csv")
newTrain.to_csv("Train New.csv")
```

Figure 4.2.2.7 Stored Cleaned Data

4.2.3 Load

After the transforms process, the data that has been cleaned needs to be loaded back into data warehouse tools to do the analysis which is PostgreSQL. There are two options to load the data, first we can load the data using the new csv file that has been created using the import option in the pgAdmin. Second option that we have is to import the data that has been cleaned from the Jupyter Notebook directly into PostgreSQL after the table of the data has been created in the pgAdmin.

After create database and table in pgAdmin, using the coding below, we can directly import the data from Jupyter Notebook into our PostgreSQL:

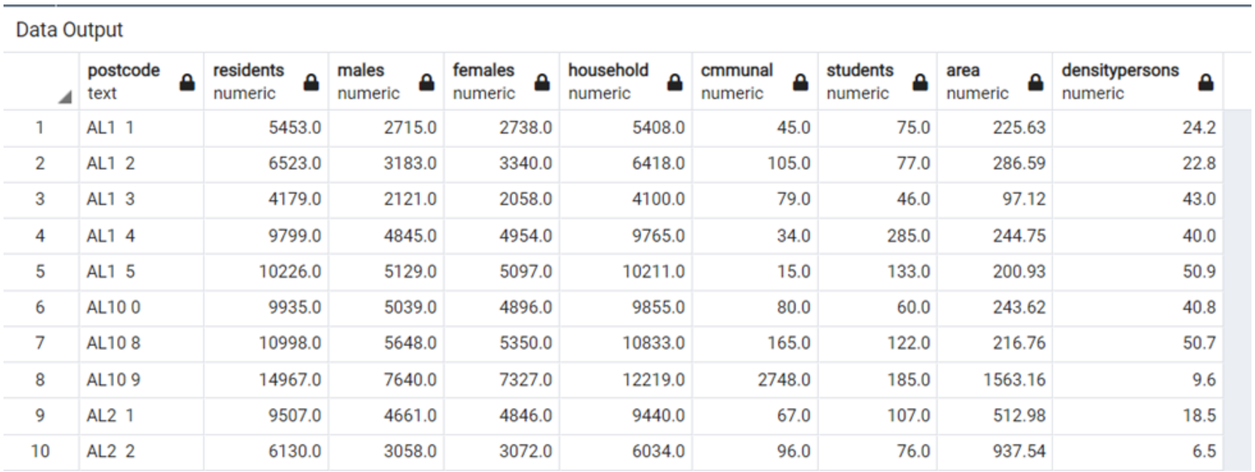


```
import psycopg2
import numpy as np
import psycopg2.extras as extras
import pandas as pd

def execute_values(conn, df, table):
    tuples = [tuple(x) for x in df.to_numpy()]
    cols = ','.join(list(df.columns))

    #SQL query to execute
    query = "INSERT INTO %s(%s) VALUES %%s" % (table, cols)
    cursor = conn.cursor()
    try:
        extras.execute_values(cursor, query, tuples)
        conn.commit()
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error: %s" % error)
        conn.rollback()
        cursor.close()
        return 1
    print("the dataframe successfully inserted")
    cursor.close()

conn = psycopg2.connect(
    database="analysis", user='postgres', password='1234', host='localhost', port='5432'
)

execute_values(conn, newPopulation, 'population')
execute_values(conn, newRoads2, 'roads')
execute_values(conn, sampleDF, 'sample')
execute_values(conn, newTest, 'test')
execute_values(conn, newTrain, 'train')
```

```
the dataframe successfully inserted
the dataframe successfully inserted
the dataframe successfully inserted
the dataframe successfully inserted
the dataframe successfully inserted
```

Figure 4.2.3.1 Import Data into PostgreSQL

View the data that has been inserted using SELECT * FROM population:



Data Output

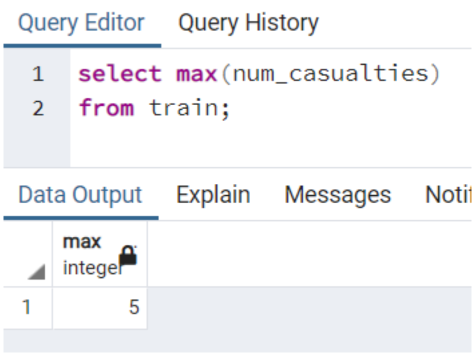
	postcode	residents	males	females	household	cmmunal	students	area	densitypersons
	text	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric
1	AL1 1	5453.0	2715.0	2738.0	5408.0	45.0	75.0	225.63	24.2
2	AL1 2	6523.0	3183.0	3340.0	6418.0	105.0	77.0	286.59	22.8
3	AL1 3	4179.0	2121.0	2058.0	4100.0	79.0	46.0	97.12	43.0
4	AL1 4	9799.0	4845.0	4954.0	9765.0	34.0	285.0	244.75	40.0
5	AL1 5	10226.0	5129.0	5097.0	10211.0	15.0	133.0	200.93	50.9
6	AL10 0	9935.0	5039.0	4896.0	9855.0	80.0	60.0	243.62	40.8
7	AL10 8	10998.0	5648.0	5350.0	10833.0	165.0	122.0	216.76	50.7
8	AL10 9	14967.0	7640.0	7327.0	12219.0	2748.0	185.0	1563.16	9.6
9	AL2 1	9507.0	4661.0	4846.0	9440.0	67.0	107.0	512.98	18.5
10	AL2 2	6130.0	3058.0	3072.0	6034.0	96.0	76.0	937.54	6.5

Figure 4.2.3.2 View Data

The data has been successfully inserted into PostgreSQL and we can start to do the analysis.

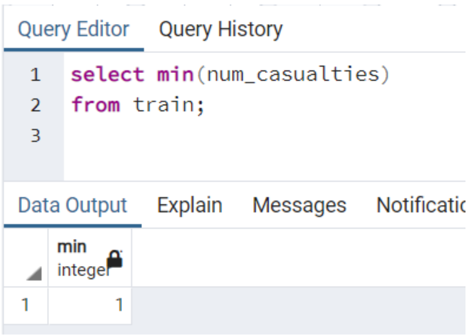
5.0 RESULTS AND DATA ANALYSIS

After going through the data integration, we have performed the data analysis using PostgreSQL and Power BI for the visualisation. Firstly, we identified the maximum and minimum number of casualties.



```sql
select max(num_casualties)
from train;
```

	max integer
1	5

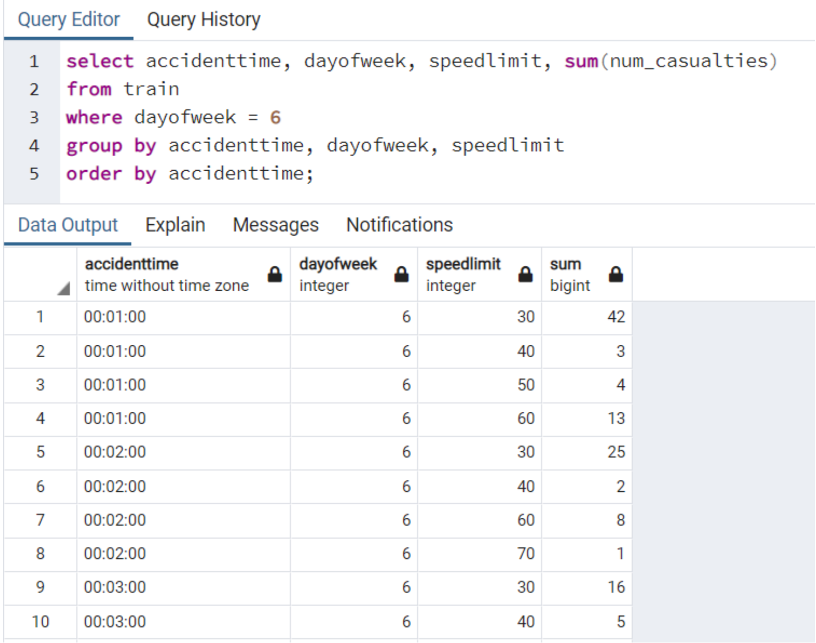


```sql
select min(num_casualties)
from train;
```

	min integer
1	1

We can see from the analysis above that the maximum number of casualties in these datasets is five and the minimum number of casualties is one.

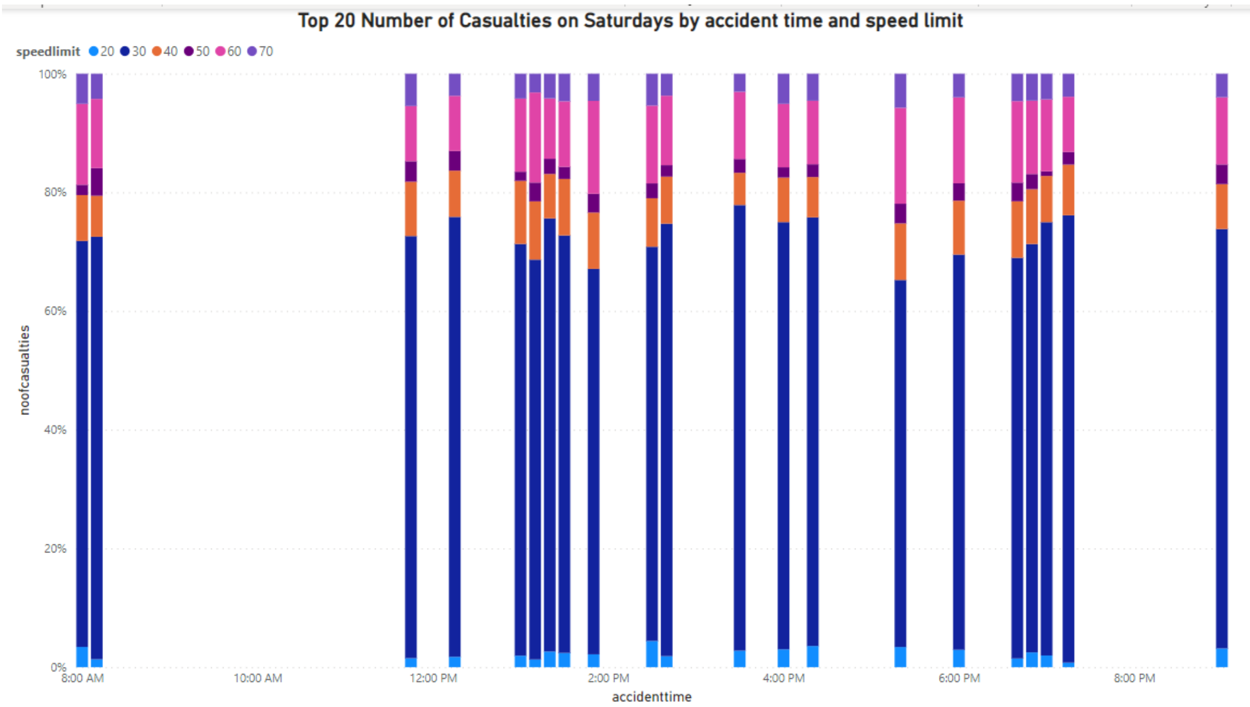
Then, we did dicing to see the number of casualties on Saturdays based on the accident time and speed limit.



```sql
select accidenttime, dayofweek, speedlimit, sum(num_casualties)
from train
where dayofweek = 6
group by accidenttime, dayofweek, speedlimit
order by accidenttime;
```

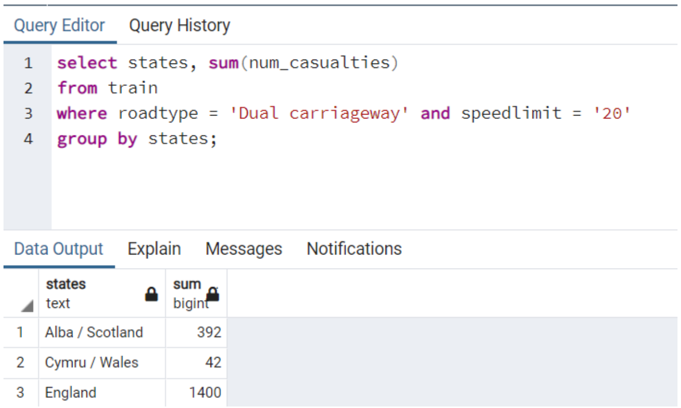
	accidenttime time without time zone	dayofweek integer	speedlimit integer	sum bigint
1	00:01:00	6	30	42
2	00:01:00	6	40	3
3	00:01:00	6	50	4
4	00:01:00	6	60	13
5	00:02:00	6	30	25
6	00:02:00	6	40	2
7	00:02:00	6	60	8
8	00:02:00	6	70	1
9	00:03:00	6	30	16
10	00:03:00	6	40	5

Visualisation:



Based on the visualisation above, we can see that most car accidents happened during the afternoon. Most accidents happen on roads that have a 30 speed limit. There were no casualties during 10.00 am. We can see that during that time there might not be a lot of vehicles on the roads. Thus, making it have less casualties involved.

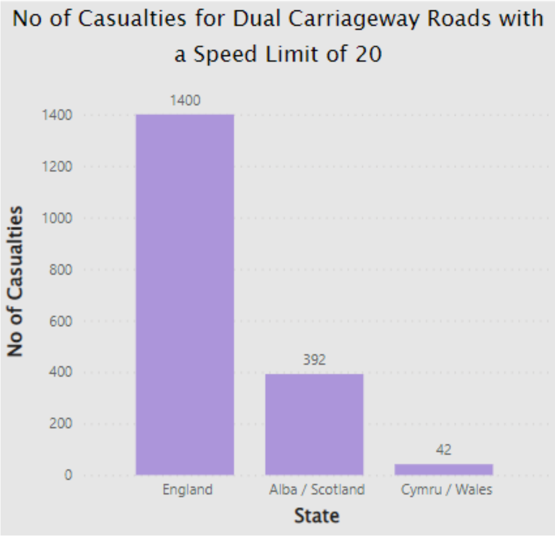
Next, we did the slicing operation to the train table. We wanted to see the number of casualties for dual carriageway roads with a speed limit of 20



```
select states, sum(num_casualties)
from train
where roadtype = 'Dual carriageway' and speedlimit = '20'
group by states;
```

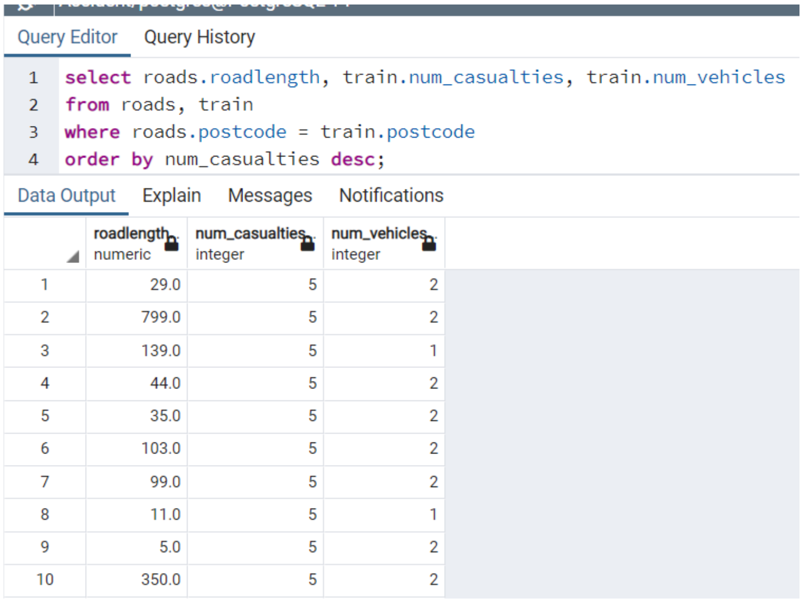
	states text	sum bigint
1	Alba / Scotland	392
2	Cymru / Wales	42
3	England	1400

Visualisation:



Based on the bar chart, England has the highest number of casualties which is 1400 casualties for the dual carriageway roads with a speed limit of 20.

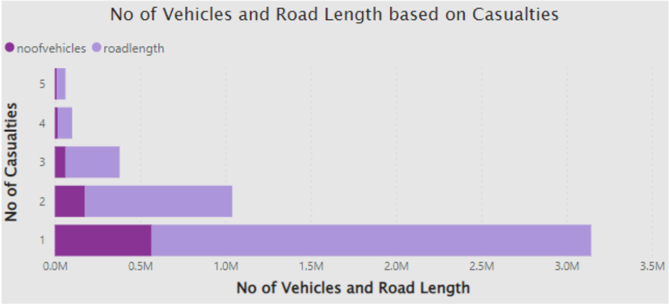
Next, we joined the train and roads table to see the number of casualties and number of vehicles involved based on its road length. We wanted to see whether road lengths had an effect on the total number of casualties.



```sql
select roads.roadlength, train.num_casualties, train.num_vehicles
from roads, train
where roads.postcode = train.postcode
order by num_casualties desc;
```

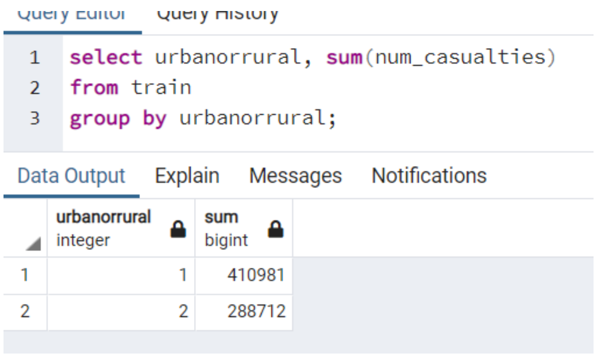
	roadlength numeric	num_casualties integer	num_vehicles integer
1	29.0	5	2
2	799.0	5	2
3	139.0	5	1
4	44.0	5	2
5	35.0	5	2
6	103.0	5	2
7	99.0	5	2
8	11.0	5	1
9	5.0	5	2
10	350.0	5	2

Visualisation:



Based on the visualisation, the longest road length has the highest number of vehicles involved in the accident with one number of casualties. Hence, the length of the road has an effect on the total number of casualties.

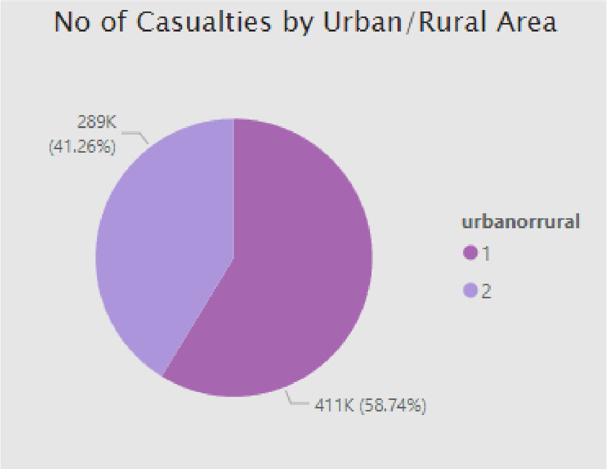
Next, we performed a roll-up operation to the train table to see the summation of number casualties in urban and rural areas.



```
select urbanorrural, sum(num_casualties)
from train
group by urbanorrural;
```

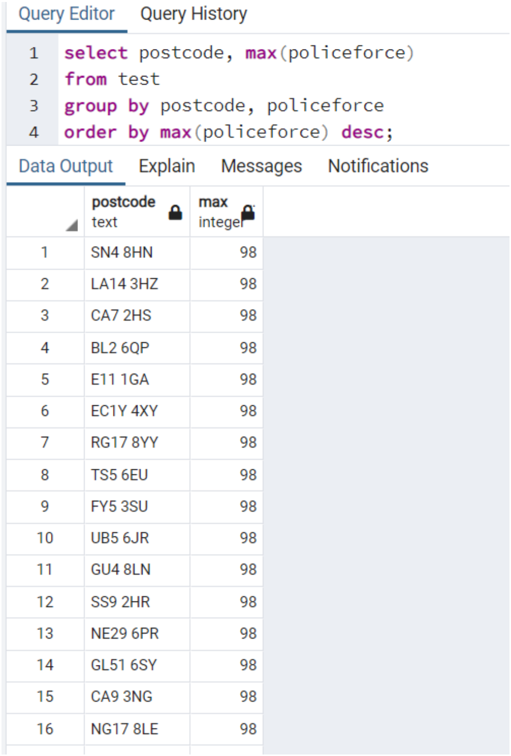
```
    urbanorrural   sum
    integer        bigint
1              1   410981
2              2   288712
```

Visualisation:



Based on the visualisation, urban 1 has the higher number of casualties which is 411k casualties while urban 2 has 289k number of casualties.

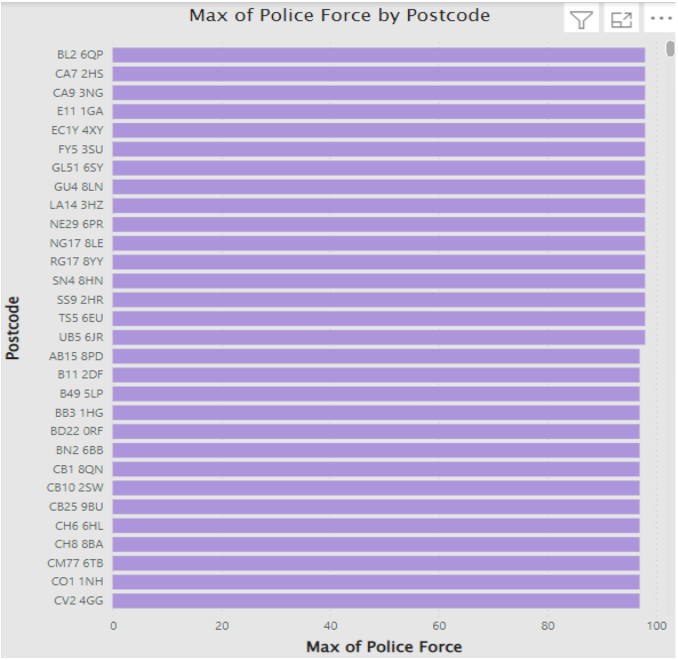
We used roll-up operation to the test table to see the maximum police force on roads according to its postcode order by maximum police force in a descending order.



```
select postcode, max(policeforce)
from test
group by postcode, policeforce
order by max(policeforce) desc;
```

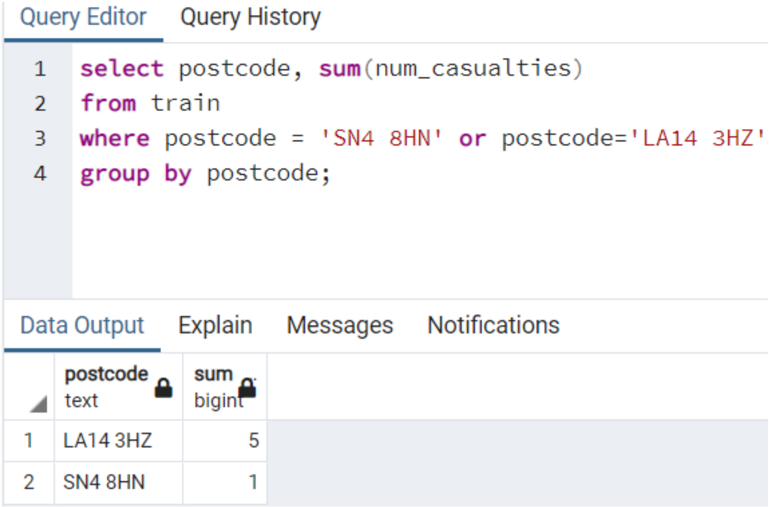
	postcode text	max integer
1	SN4 8HN	98
2	LA14 3HZ	98
3	CA7 2HS	98
4	BL2 6QP	98
5	E11 1GA	98
6	EC1Y 4XY	98
7	RG17 8YY	98
8	TS5 6EU	98
9	FY5 3SU	98
10	UB5 6JR	98
11	GU4 8LN	98
12	SS9 2HR	98
13	NE29 6PR	98
14	GL51 6SY	98
15	CA9 3NG	98
16	NG17 8LE	98

Visualisation:



The highest number of police force is 98, and we can see that there are 16 areas that are represented by the postcode as having the highest police force.

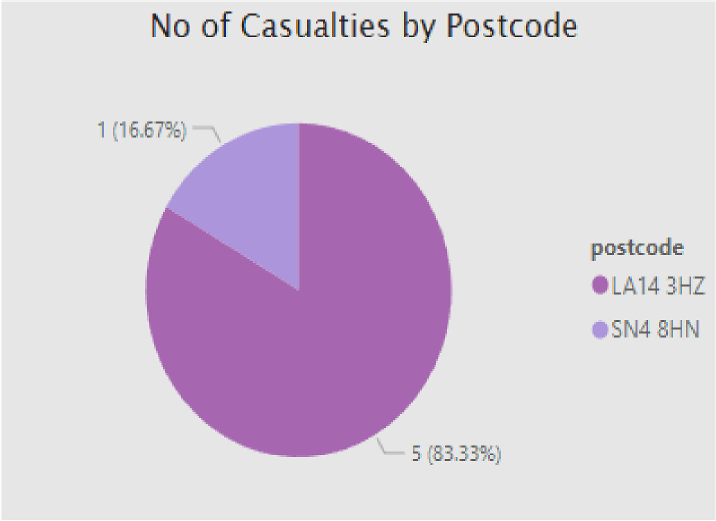
Then, we applied the slicing operation on the train table to see whether the number of police force affects the number of casualties.



```
select postcode, sum(num_casualties)
from train
where postcode = 'SN4 8HN' or postcode='LA14 3HZ'
group by postcode;
```

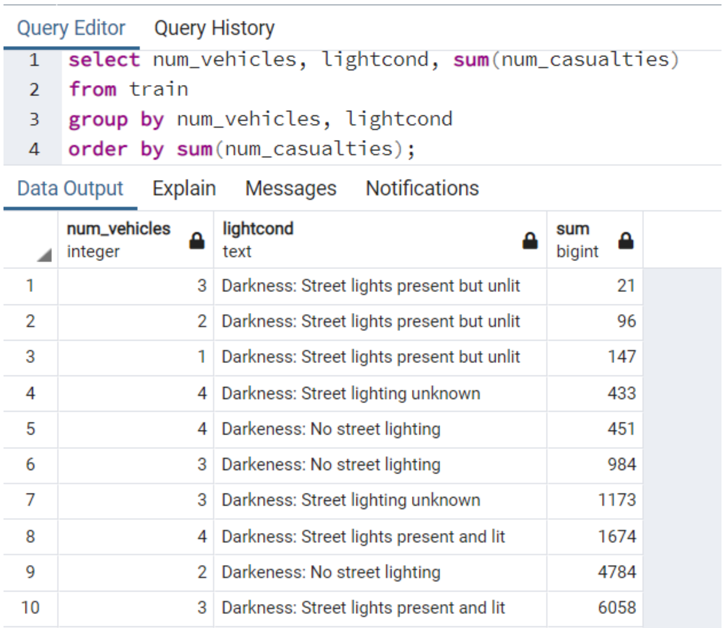
	postcode text	sum bigint
1	LA14 3HZ	5
2	SN4 8HN	1

Visualisation:



At the postcode of LA14 3HZ the number of casualties is five. Meanwhile, at the postcode SN4 8HN the number of casualties is one.

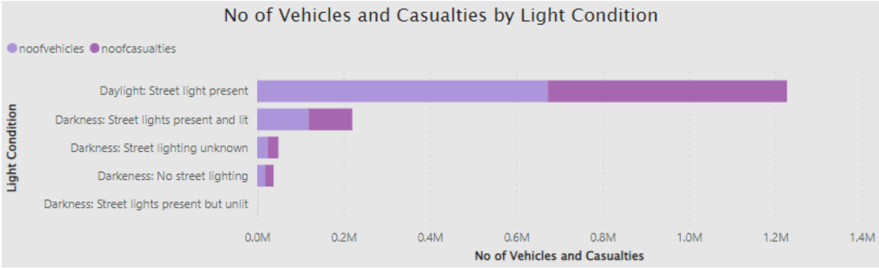
After that, we did the roll-up operation on the train table to see the total number of casualties based on the light condition of the road and the number of vehicles involved.



```sql
select num_vehicles, lightcond, sum(num_casualties)
from train
group by num_vehicles, lightcond
order by sum(num_casualties);
```

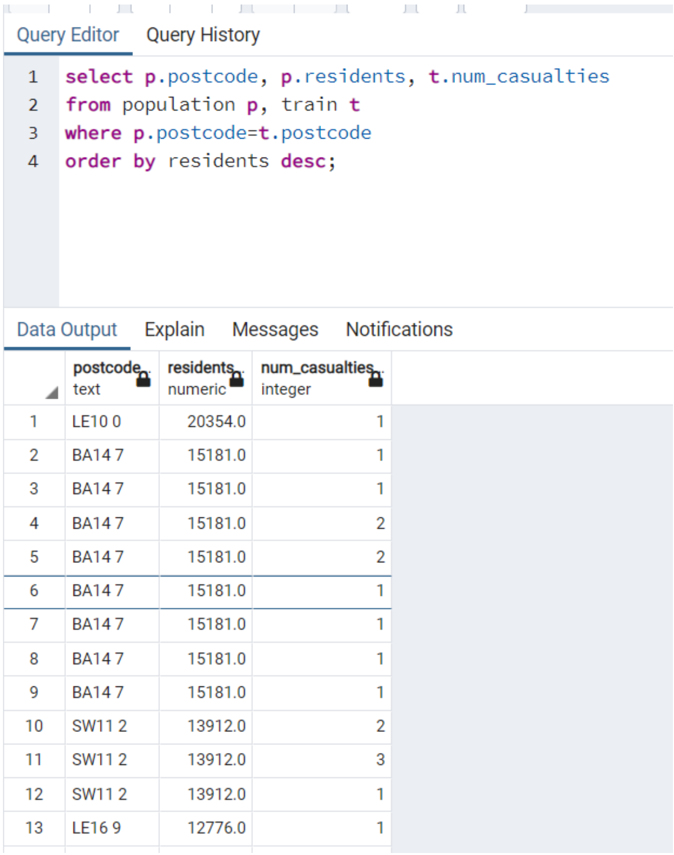
	num_vehicles integer	lightcond text	sum bigint
1	3	Darkness: Street lights present but unlit	21
2	2	Darkness: Street lights present but unlit	96
3	1	Darkness: Street lights present but unlit	147
4	4	Darkness: Street lighting unknown	433
5	4	Darkeness: No street lighting	451
6	3	Darkeness: No street lighting	984
7	3	Darkness: Street lighting unknown	1173
8	4	Darkness: Street lights present and lit	1674
9	2	Darkeness: No street lighting	4784
10	3	Darkness: Street lights present and lit	6058

Visualisation:



Based on the visualisation, the light condition during the daylight with the presence of street light causes the most casualties in total with 553910 casualties and 673101 vehicles involved.

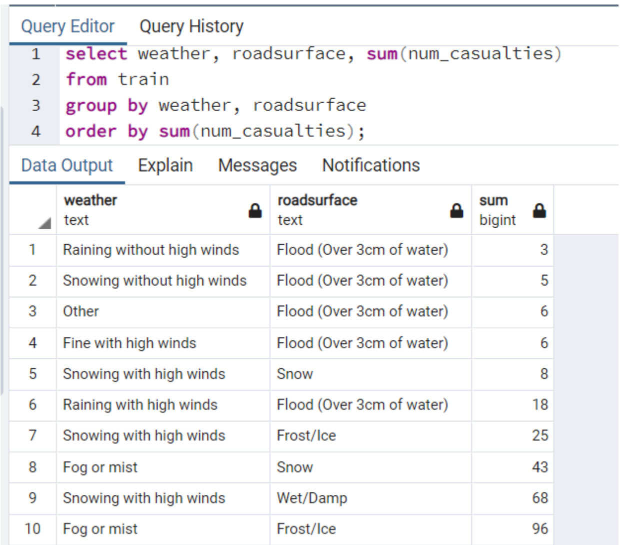
We joined the population table and train table to see the number of casualties based on the number of residents of the postcode area.



```sql
select p.postcode, p.residents, t.num_casualties
from population p, train t
where p.postcode=t.postcode
order by residents desc;
```

	postcode text	residents numeric	num_casualties integer
1	LE10 0	20354.0	1
2	BA14 7	15181.0	1
3	BA14 7	15181.0	1
4	BA14 7	15181.0	2
5	BA14 7	15181.0	2
6	BA14 7	15181.0	1
7	BA14 7	15181.0	1
8	BA14 7	15181.0	1
9	BA14 7	15181.0	1
10	SW11 2	13912.0	2
11	SW11 2	13912.0	3
12	SW11 2	13912.0	1
13	LE16 9	12776.0	1

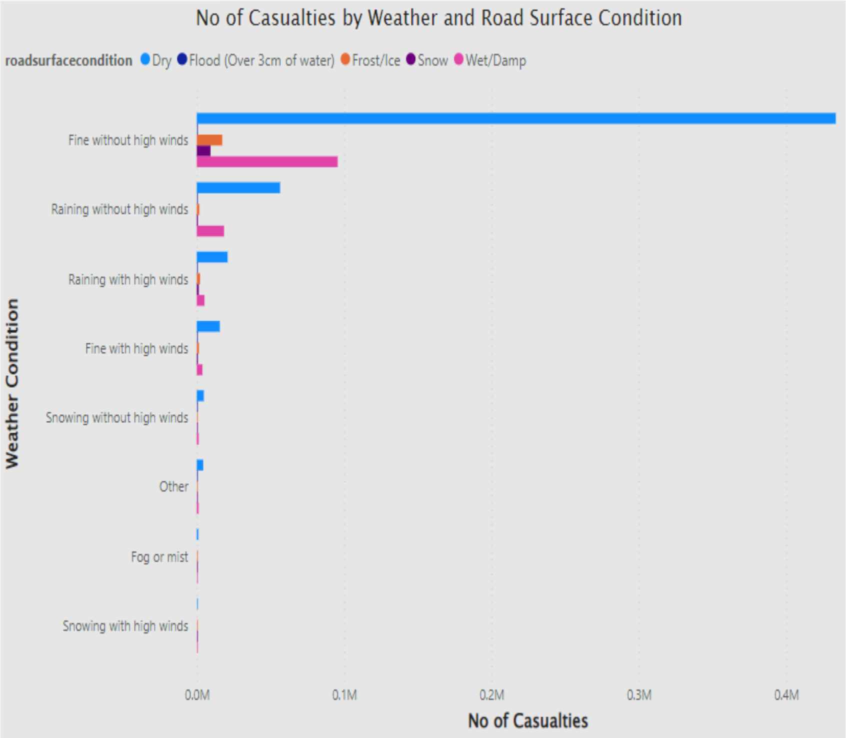
We applied the roll-up operation to the train table to see the summation of casualties based on the weather and the road surface.



```
select weather, roadsurface, sum(num_casualties)
from train
group by weather, roadsurface
order by sum(num_casualties);
```

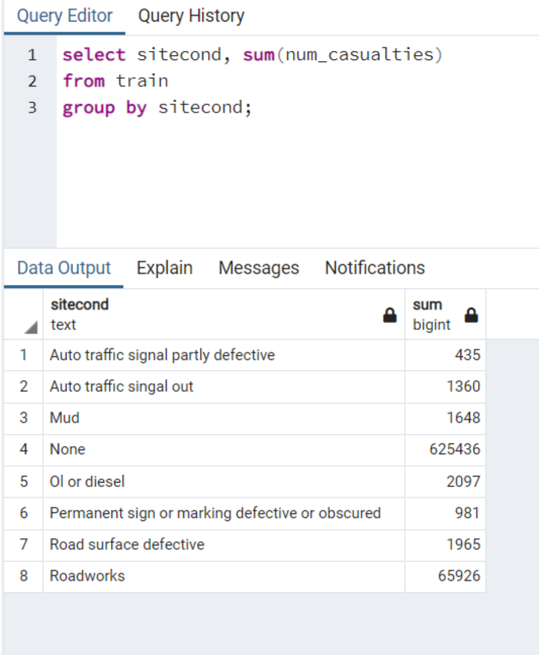
	weather text	roadsurface text	sum bigint
1	Raining without high winds	Flood (Over 3cm of water)	3
2	Snowing without high winds	Flood (Over 3cm of water)	5
3	Other	Flood (Over 3cm of water)	6
4	Fine with high winds	Flood (Over 3cm of water)	6
5	Snowing with high winds	Snow	8
6	Raining with high winds	Flood (Over 3cm of water)	18
7	Snowing with high winds	Frost/Ice	25
8	Fog or mist	Snow	43
9	Snowing with high winds	Wet/Damp	68
10	Fog or mist	Frost/Ice	96

Visualisation:



If we look at the visualisation roughly, during weather conditions fine without high winds cause the most number of casualties in any condition of road surface.

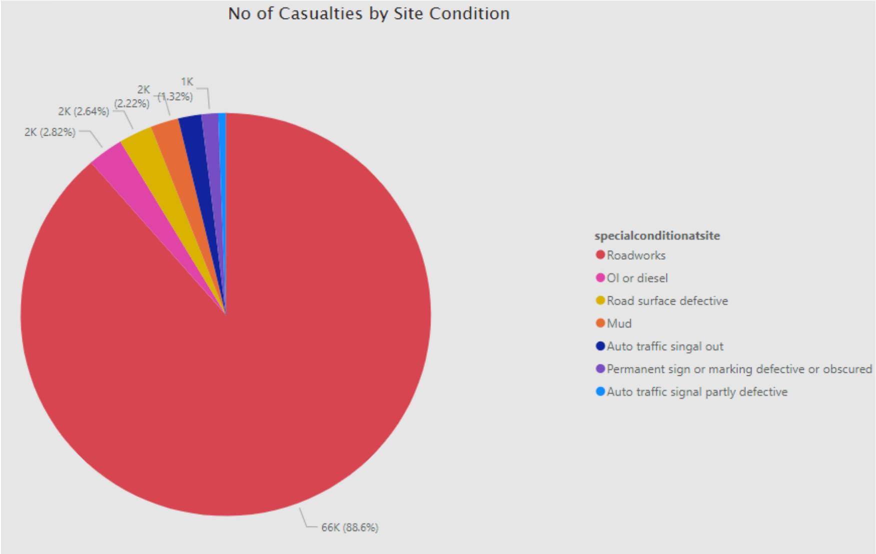
We observed the number of casualties from the train table based on the site condition using the roll-up operation.



```
select sitecond, sum(num_casualties)
from train
group by sitecond;
```

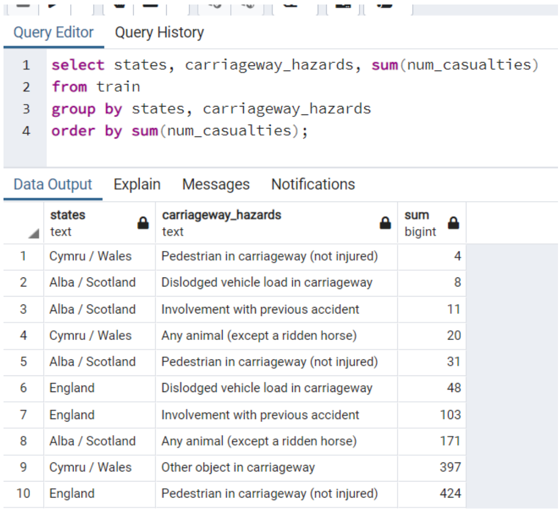
	sitecond text	sum bigint
1	Auto traffic signal partly defective	435
2	Auto traffic singal out	1360
3	Mud	1648
4	None	625436
5	Ol or diesel	2097
6	Permanent sign or marking defective or obscured	981
7	Road surface defective	1965
8	Roadworks	65926

Visualisation:



After filtering the 'None' site condition from the visualisation, we can see that roadworks site conditions cause the most number of casualties with 66K casualties.

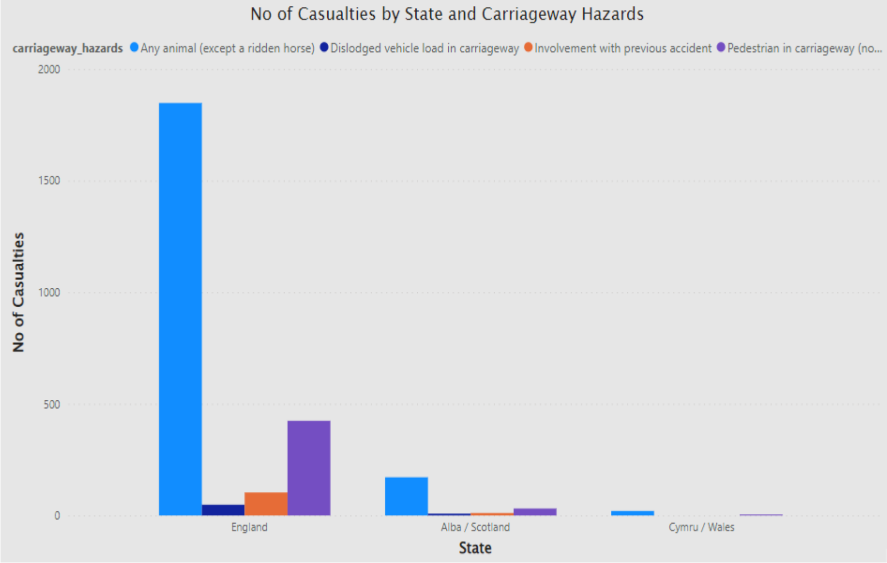
Then, we did the roll-up operation to the train table to see the number of casualties according to states and carriageway hazards.



```sql
select states, carriageway_hazards, sum(num_casualties)
from train
group by states, carriageway_hazards
order by sum(num_casualties);
```

	states text	carriageway_hazards text	sum bigint
1	Cymru / Wales	Pedestrian in carriageway (not injured)	4
2	Alba / Scotland	Dislodged vehicle load in carriageway	8
3	Alba / Scotland	Involvement with previous accident	11
4	Cymru / Wales	Any animal (except a ridden horse)	20
5	Alba / Scotland	Pedestrian in carriageway (not injured)	31
6	England	Dislodged vehicle load in carriageway	48
7	England	Involvement with previous accident	103
8	Alba / Scotland	Any animal (except a ridden horse)	171
9	Cymru / Wales	Other object in carriageway	397
10	England	Pedestrian in carriageway (not injured)	424

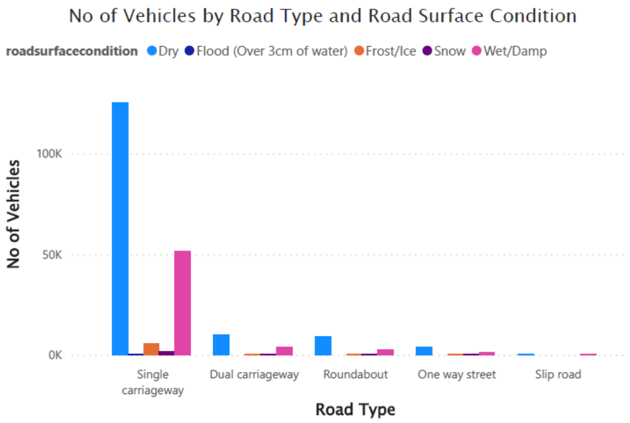
Visualisation:

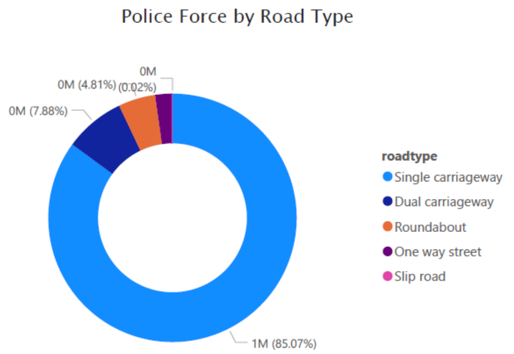


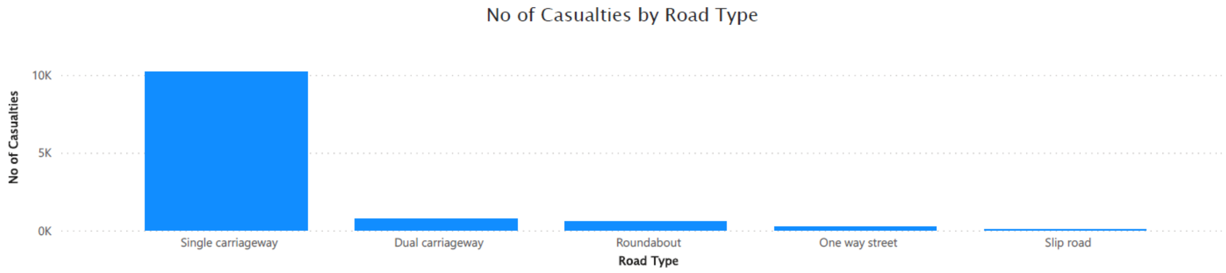
After filtering out the 'None' and 'Other objects in carriageway' from the carriageway hazards column, we can see that any animal except a ridden horse causes the most number of casualties in all states. And there is a clear visualisation that shows England has the most number of casualties compared to other states.

After gathering knowledge through data analysis, we will proceed on visualising the data to produce insights for our project.

Visualisation based on road type:



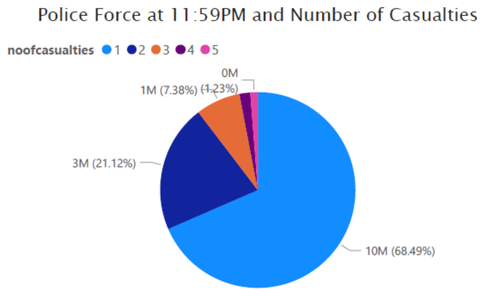


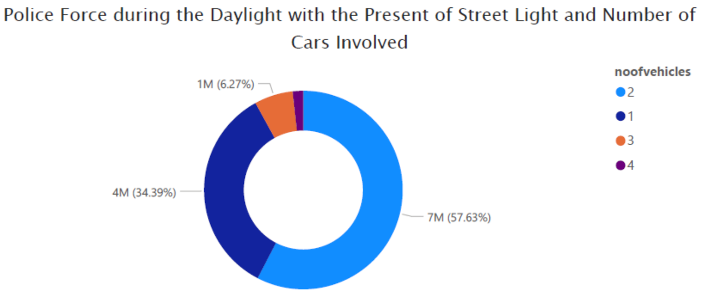


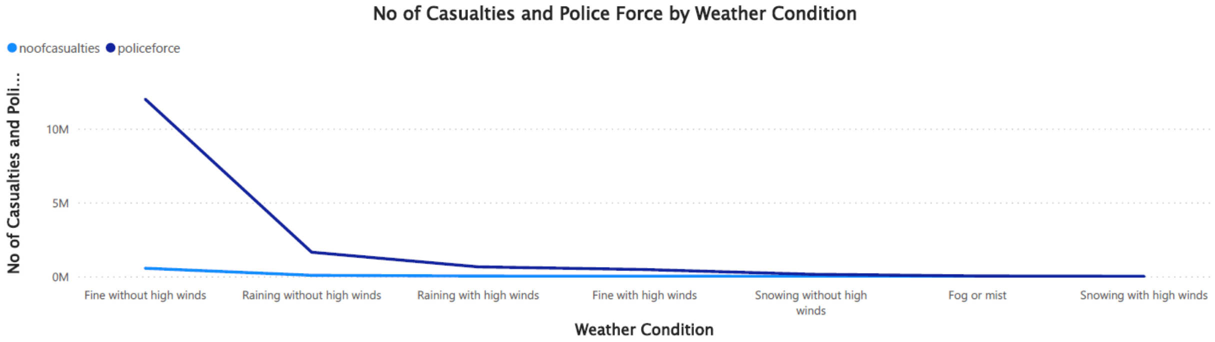
Interpretation:

Based on the dashboard above, we can see that most of the accidents that involve a big number of vehicles are from the single carriageway type of road with a dry road surface. Although single carriageway roads have a lot of police force on site, the number of casualties are still high. The least number of casualties are from slip roads. Based on our observation, we can say that single carriageway roads are quite busy seeing as they have the highest number of vehicles involved in accidents. The slip roads are the least busy as they have quite a small number of casualties and vehicles involved in the accidents.

Visualisation based on police:


Police Force at 11:59PM and Number of Casualties


Police Force during the Daylight with the Present of Street Light and Number of Cars Involved


No of Casualties and Police Force by Weather Condition

Interpretation:

Based on the dashboard above, we can see that during the weather conditions fine without high winds, the total number of police force is the highest with the 12M of police roughly and the number of casualties that happen during that time is much lower than the number of police. Even Though the number of casualties was lower than police during that time, the weather conditions were fine without high winds and still scored the highest number of casualties compared to other weather conditions.

As we know from the visualisation above, during the daylight with the present of the street light has the highest number of casualties. In this dashboard we want to see the number of police forces and the number of cars that were involved in the car accident in certain areas. We can see that the highest number of cars involved is two with 57.63% of police force. Now let's see during the midnight, there are 68.49% of police force at 11:59PM with the one casualtie happening is the highest.

6.0 CONCLUSION

Based on the analysis that had been made to achieve the objectives, we can clearly see that England has the highest accident rate. We can assume this result because England has the biggest authority in the United Kingdom compared to Scotland and Wales. Length of the roads is one of the factors that affect the accident rate, the longer the length of the roads, the accident rate of that area is also high. Other than that, the weather condition and the light condition on the roads also can cause the accident. We can see that even during normal weather conditions, which are fine without high winds, have the highest number of car accidents. Not only that, for the light condition during the normal daylight with the presence of the street light also has the highest number of car accidents. Not only that, the number of casualties increases during the afternoon. The time also plays a significant role in detecting accidents. Most cases also happen on roads that have a speed limit of 30. Although the speed limit is low, the number of casualties is the highest among other roads that have a higher speed limit.

As for the police force authority, we can see that even with the highest number of police authority in that area, the number of casualties is also the highest compared to other factors. From these datasets, we can conclude that car accidents are caused by the human mentality. This is because even with the many police on the roads and normal conditions of weather and light, the number of casualties is also high. Hence, as human beings we should know the importance of our life and be careful whenever we drive.

Through the entire process of this project, there are challenges that we faced while doing this study. One of the main challenge that we face is during the beginning of our project when we unable to choose which datasets is the best to study on. We go through so many datasets but unable to come up with the objective of the datasets clearly. With the current datasets also we faced challenges to execute our objectives clearly. However, after go through into the datasets so many times we can come out with a very clear objective.

The next challenge that we faced is we are unable to determine which tools to use for our ETL pipeline. After brainstorming with the team members we decided to use PostgreSQL as our database and Snowflake as our data warehouse. However, we come across some problems importing the data that has been cleaned into the Snowflake database because when we do the analysis in the Snowflake and then we do the visualisation in the Power BI, the results shown are different from each other. Hence, we decide to load our data back into PostgreSQL to do the analysis. This is because with the PostgreSQL we can directly import our cleaned data from Jupyter Notebook into PostgreSQL using some coding in Python language and the data from the PostgreSQL also can be directly imported to the Power BI for the visualisation. Directly load and transfer the data from transforming tools into the data warehouse tools and visualisation tools make it easier to ensure the data that is being transferred from one tool to another tools are the same data.

The last challenge that we faced is during the analysis process. This is because some of the dataset has many attributes but there are no strong correlations between the attributes. Hence, we are having a hard time to analyse and visualise the attributes together in order to reach our objectives.

7.0 REFERENCES

1. *ETL - Simple Pipeline.* (2020, November 24). Michael Fuchs Python Netlify App. https://michael-fuchs-python.netlify.app/2020/11/24/etl-simple-pipeline/

2. *Machine Hack Mar22*. (2022, April 1). Kaggle. Retrieved May 18, 2022, from https://www.kaggle.com/datasets/krishnadaskv/machine-hack-mar22-predict-accident-risk-score?select=roads_network.csv

3. *MachineHack*. (2022, March 11). Machine Hack. https://machinehack.com/hackathons/predict_accident_risk_score_for_unique_postcode/overview

4. *PostgreSQL ROLLUP*. (2020, July 17). PostgreSQL Tutorial. https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-rollup/

5. *Slicing and Dicing Data - SQL Queries Succinctly Ebook*. (n.d.). Syncfusion. Retrieved May 7, 2022, from https://www.syncfusion.com/succinctly-free-ebooks/sql-queries-succinctly/slicing-and-dicing-data

6. *Teleron, A.* (2019, November 4). PostgreSQL Integration with Jupyter Notebook. https://medium.com/analytics-vidhya/postgresql-integration-with-jupyter-notebook-deb97579a38d

BSD2343 DATA WAREHOUSING
8.0 GROUP PROJECT MARKING SCHEME

Rubric for CLO1

CLO1: **Acquire fundamental Big data and data warehousing concepts**								
CRITERIA	**LEVEL OF ACHIEVEMENT**						**WEIGHTAGE**	**SCORE**
	0 **Grossly Inadequate**	**1** **Inadequate**	**2** **Emerging**	**3** **Developing**	**4** **Good**	**5** **Excellent**		
Description and explanation of the selected project and problem to be solved.	No description and explanation about the project selected and problem to be solved in the report.	Poorly describe and explain about the project selected with no problem to be solved in the report.	Poorly describe and explain about the project selected and problem to be solved in the report.	Fairly describe and explain about the project selected and problem to be solved in the report.	Clearly describe and explain about the project selected and problem to be solved in the report.	Excellently describe and explain about the project selected and problem to be solved in the report.	1	
Explanation regarding the data schema and the relationship between data.	Failed to explain the data schema and the relationship between data.	Able to explain the data schema but failed to explain the relationship between data.	Poorly explain the data schema and the relationship between data.	Fairly explain the data schema and the relationship between data.	Good explanation on the data schema and the relationship between data.	Excellent explanation on the data schema and the relationship between data.	2	
Concluding remarks.	No concluding remarks provided.	Limited concluding remarks provided and inaccurate .	Concluding remarks provided but unclear and inaccurate.	Concluding remarks provided but partly inaccurate.	Clear and good concluding remarks provided.	Very clear and excellent concluding remarks provided.	1	
						TOTAL		**20**

Rubric for CLO2

CLO2: **Analyze real life problems using appropriate Big data and data warehousing concepts**								
CRITERIA	**LEVEL OF ACHIEVEMENT**						**WEIGHTAGE**	**SCORE**
	0 **Grossly Inadequate**	**1** **Inadequate**	**2** **Emerging**	**3** **Developing**	**4** **Good**	**5** **Excellent**		
Able to sketch the pipeline structure of the project.	Failed to provide the pipeline structure of the project.	Able to sketch the pipeline structure but the pipeline is wrong.	Able to sketch the pipeline structure but the pipeline is partly correct.	Able to sketch the pipeline structure correctly but no description on the pipeline.	Able to sketch the pipeline structure correctly but limited description on the pipeline.	Excellently sketch the pipeline structure with good description on the pipeline.	1	

Able to present database in relational model and map the relationship between data.	Failed to present database in relational model and map the relationship between data.	Able to present database in relational model but failed to map the relationship between data.	Poorly present database in relational model and map the relationship between data..	Fairly present database in relational model and map the relationship between data.	Clearly present database in relational model and map the relationship between data.	Excellently present database in relational model and map the relationship between data.	1	
Able to sketch and explain the ETL pipeline.	Failed to sketch and explain the ETL pipeline.	Able to sketch ETL pipeline but failed to provide any explanation on the pipeline.	Poorly sketch and explain the ETL pipeline.	Fairly sketch and explain the ETL pipeline.	Clearly sketch and explain the ETL pipeline.	Excellently sketch and explain the ETL pipeline.	1	
Perform all data analysis with data visualization.	No results obtained from the data analysis.	Poorly present data analysis and there is no data visualization.	Poorly present data analysis and data visualization.	Fairly present data analysis and data visualization.	Clear and good presentation of the data analysis and data visualization.	Excellent presentation of the data analysis and data visualization.	0.5	
Explanation on the data analysis results.	Unable to provide any explanation on the data analysis results.	Limited explanation on the data analysis results.	Able to explain the data analysis results but unclear and inaccurate.	Able to explain the data analysis results clearly but inaccurate.	Able to explain the data analysis results clearly.	Able to explain the data analysis results perfectly.	0.5	
						TOTAL		**20**

Rubric for CLO3

CLO3: **Build and integrate Big data in data warehouse by using appropriate software**								
CRITERIA	**LEVEL OF ACHIEVEMENT** E GA T HG I						E W R E	C S
	0 **Grossly Inadequate**	**1** **Inadequate**	**2** **Emerging**	**3** **Developing**	**4** **Good**	**5** **Excellent**		
Ability to extract the datasets from sources very well.	Unable to extract the datasets.	Barely able to extract the datasets from the sources.	Partly able to extract the datasets from the sources.	Able to extract the datasets from the sources in successful results.	Very good in extracting the datasets from the sources.	Able to extract the datasets from the sources excellently.	1	

Ability to construct pipeline structure of the project involving various tools from Big Data and Data Warehouse.	Unable to construct pipeline structure of the project involving various tools from Big Data and Data Warehouse.	Barely able to construct pipeline structure of the project involving various tools from Big Data and Data Warehouse.	Able to construct pipeline structure of the project but limited tools from Big Data and Data Warehouse.	Able to construct pipeline structure of the project involving various tools from Big Data and Data Warehouse in successful results.	Very good in constructing pipeline structure of the project involving various tools from Big Data and Data Warehouse.	Able to construct pipeline structure of the project involving various tools from Big Data and Data Warehouse excellently.	1	
Able to map the relationship between data by using appropriate tools.	Unable to map the relationship between data by using appropriate tools.	Barely able to map the relationship between data by using appropriate tools.	Able to map the relationship between data by using appropriate tools but the mapping is inaccurate.	Able to map the relationship between data by using appropriate tools correctly.	Very good in mapping map the relationship between data by using appropriate tools.	Able to map the relationship between data by using appropriate tools excellently.	1	
Ability to construct ETL pipeline by using appropriate tools.	Unable to construct ETL pipeline by using appropriate tools.	Barely able to construct ETL pipeline by using appropriate tools.	Able to construct ETL pipeline by using appropriate tools but the pipeline is inaccurate.	Able to construct ETL pipeline by using appropriate tools correctly.	Very good in constructing ETL pipeline by using appropriate tools.	Able to construct ETL pipeline by using appropriate tools excellently.	1	
Ability to visualise (table, graph, GUI and etc) the programming codes.	Unable to visualise (table, graph, GUI and etc) the programming codes.	Barely able to visualise (table, graph, GUI and etc) the programming codes.	Partly to visualise (table, graph, GUI and etc) the programming codes.	Ability to visualise (table, graph, GUI and etc) the programming codes in successful results.	Very good in visualising (table, graph, GUI and etc) the programming codes.	Excellently visualise (table, graph, GUI and etc) the programming codes.	1	
The code can be executed and easy to understand the codes constructed.	No code is constructed.	Only few codes can be executed and difficult to follow the structure and flow of the codes.	Some of the codes can be executed and fairly difficult to follow the structure and flow of the codes.	The code can be executed and fairly easy to follow the structure and flow of the codes.	The code can be executed and easily to follow the structure and flow of the codes.	The code can be executed and well easily to follow the structure and flow of the codes.	1	
						TOTAL		**30**

Rubric for CLO4

CLO4: **Work in group in order to complete the given assessments in specific time frame.**								
CRITERIA	**LEVEL OF ACHIEVEMENT**						**WEIGHTAGE**	**SCORE**
	0 **Grossly Inadequate**	**1** **Inadequate**	**2** **Emerging**	**3** **Developing**	**4** **Good**	**5** **Excellent**		
Every member in the group able to provide information related to this project and show understanding of the project.	All group members unable to provide information related to this project and show understanding of the project.	Some group members unable to provide information related to the project and show understanding of the project.	One or two group members unable to provide information related to the project and show understanding of the topic.	All group members able to provide adequate information related to this project but only show average understanding of the project.	All group members able to provide adequate information related to this project and show good understanding of the project.	All group members able to provide information related to this project and show excellent understanding of the project.	1	
Able to work together as a team towards goal achievement and submit report on time.	Unable to work together as a team and submit report on time.	Able to work together as a team, however, unable to submit report on time.	Able to work together as a team towards goal achievement, however, unable to submit quality report.	Able to work together as a team towards goal achievement and submit average quality report on time.	Able to work together as a team towards goal achievement and submit good quality report on time.	Able to work together as a team towards goal achievement and submit high quality report on time.	2	
Ability to assume alternate roles as a group leader and group members.	No clear evidence of ability to assume alternate roles as a group leader and group members demonstrated in practice.	Attempt to demonstrate in practice the ability to alternate roles as a group leader and group members but with limited effect and require improvements.	Able to demonstrate in practice the ability to assume alternate roles as a group leader and group members with some effect(s) and require minor improvements.	Able to demonstrate in practice the ability to assume alternate roles as a group leader and a group member to achieve the same goal.	Able to demonstrate in good practice the ability to assume alternate roles as a group leader and a group member to achieve the same goal.	Able to demonstrate in excellent practice the ability to assume alternate roles as a group leader and a group member to achieve the same goal.	1	
						TOTAL		**20**