In [2]:
```
import pandas as pd
import numpy as np
pd.pandas.set_option ("display.max_columns", None)
```

Task 1

Question 1. The two tables can be merged using the Accident_Index field. Write Python code to merge the two tables and store the results in a new csv file.

In [3]:
```
df_acc = pd.read_csv("Data/Task1/Accidents_2015.csv")
df_cas = pd.read_csv("Data/Task1/Casualties_2015.csv")

merge = df_acc.merge(df_cas, on = "Accident_Index")
merge
```

```
/Users/kike/opt/anaconda3/lib/python3.9/site-packages/IPython/core/interactiveshell.py:344
4: DtypeWarning: Columns (0) have mixed types.Specify dtype option on import or set low_me
mory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Out[3]:

	Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Forc
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	
...	...	...	...	...	...	
164515	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164516	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164517	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164518	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164519	2015984141415	314050.0	579638.0	-3.348646	55.103676	9

164520 rows × 47 columns

Question 2. The Accident_Severity variable needs to be recoded. Write Python code to replace the values in this column as: 1=Minor (1 should be converted to minor), 2=Medium, 3=Severe.

In [4]:
```
merg = merge.copy()
dic = {1:"minor", 2: "medium",3:"severe"}

merg["Accident_Severity"] = merg["Accident_Severity"].replace(dic)

merg["Accident_Severity"]
```

Out[4]:
```
0         severe
```

```
1         severe
2         severe
3         severe
4         medium
           ...
164515    severe
164516    severe
164517    severe
164518    severe
164519    severe
Name: Accident_Severity, Length: 164520, dtype: object
```

Question 3. Replace missing values in a set of attributes by -1 (this process has been already done for the example datasets). Then, write Python code to detect these values and report the names of the columns in each table that contain such values.

In [5]:
```
#replacing nulls with -1
merg = merge.copy()
merg.fillna(-1, inplace = True)
merg

#defining fucntion to search for -1
def MinusOneReporter(merg): #function to give back the names of columns that contain
-1 values
    columnNames = []
    for column in merg.columns:
        if merg[column].eq(-1).any():
            columnNames.append(column)
    return columnNames

MinusOneReporter(merg)
```

Out[5]:
```
['Location_Easting_OSGR',
 'Location_Northing_OSGR',
 'Longitude',
 'Latitude',
 'Time',
 'Junction_Detail',
 'Junction_Control',
 '2nd_Road_Class',
 '2nd_Road_Number',
 'Pedestrian_Crossing-Human_Control',
 'Pedestrian_Crossing-Physical_Facilities',
 'Road_Surface_Conditions',
 'Special_Conditions_at_Site',
 'Carriageway_Hazards',
 'Did_Police_Officer_Attend_Scene_of_Accident',
 'LSOA_of_Accident_Location',
 'Sex_of_Casualty',
 'Age_of_Casualty',
 'Age_Band_of_Casualty',
 'Pedestrian_Location',
 'Pedestrian_Movement',
 'Car_Passenger',
 'Bus_or_Coach_Passenger',
```

```
 'Pedestrian_Road_Maintenance_Worker',
 'Casualty_Home_Area_Type',
 'Casualty_IMD_Decile']
```

QUESTION 4 For all numerical variables, write a Python function to check if there are any clearly extreme values, or values that do not belong in that column. If you find any, remove these records from the dataset.

In [15]:

```
#using +-3 Z method
#Getting list of numerical data
numerics = [x for x in merg.columns if merg[x].dtypes != "object"]
numerics

#Atrubutes to work
df = merg[numerics]

#defining function
def out_det(df):
#Find outliers (using 3 sd metric to label as outlier)
    for i in numerics:
        std =  df[i].std()
        mean=  df[i].mean()

        df[i + " distance from mean"] = abs(df[i] -(mean))



        df.loc[df[i + " distance from mean"] > (3*std),  i + " distance from mean"] =
"Outlier"


    clean = df[~df.isin(["Outlier"]).any(axis=1)][numerics]
    clean.dropna(inplace= True)
    return clean




# checkin outliers
clean_df = out_det(df)
clean_df
```

```
/var/folders/5x/8y9rl70j7_s076nbjm7jn2dh0000gn/T/ipykernel_12892/3885882992.py:15: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  df[i + " distance from mean"] = abs(df[i] -(mean))
/Users/kike/opt/anaconda3/lib/python3.9/site-packages/pandas/core/indexing.py:1817: Settin
gWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_column(loc, value, pi)
```

Out[15]:

	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Force	Accident_Sev
4	524630.0	179040.0	-0.206022	51.496572	1	
5	525480.0	179530.0	-0.193610	51.500788	1	
7	527590.0	178660.0	-0.163542	51.492497	1	
8	524170.0	180930.0	-0.211980	51.513659	1	
11	523850.0	181450.0	-0.216407	51.518402	1	
...	...	...	...	...	...	
164502	319301.0	566593.0	-3.262676	54.987365	98	
164504	312087.0	570791.0	-3.376671	55.023855	98	
164505	320671.0	569791.0	-3.242159	55.016316	98	
164506	311731.0	586343.0	-3.387067	55.163502	98	
164508	311731.0	586343.0	-3.387067	55.163502	98	

108414 rows × 42 columns

QUESTION 5 Write Python code to create a new attribute (column) called is_minor, that checks whether a casualty was a minor or an adult. Being adult is defined as having an age of 18 or above. The column should only contain the values 'Yes' and 'No'.

In [9]:

```
merg = merge.copy()

merg = merg.assign(is_minor=np.where(merg['Age_of_Casualty'] >= 18, 'no', 'yes'))

merg
```

Out[9]:

	Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Forc
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	
...	...	...	...	...	...	
164515	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164516	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164517	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164518	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164519	2015984141415	314050.0	579638.0	-3.348646	55.103676	9

164520 rows × 48 columns

QUESTION 6.- Choose an attribute which is numeric and has some missing values. Then, calculate the average of all the available values in that column and fill the missing cells in the column with the average value. For example, the Location_Easting_OSGR variable has about 27 missing values - solve this with imputation of the average of the 'Location_Easting_OSGR' of all records.

In [10]:
```
merg = merge.copy()
print(merg.isnull().sum())


#Choosed :Location_Easting_OSGR

#Fill with mean
merg["Location_Easting_OSGR"] =
merg["Location_Easting_OSGR"].fillna(merg["Location_Easting_OSGR"].mean())
merg
```

```
Accident_Index                                  0
Location_Easting_OSGR                          37
Location_Northing_OSGR                         37
Longitude                                      37
Latitude                                       37
Police_Force                                    0
Accident_Severity                               0
Number_of_Vehicles                              0
Number_of_Casualties                            0
Date                                            0
Day_of_Week                                     0
Time                                           22
Local_Authority_(District)                      0
Local_Authority_(Highway)                       0
1st_Road_Class                                  0
1st_Road_Number                                 0
Road_Type                                       0
Speed_limit                                     0
Junction_Detail                                 0
Junction_Control                                0
2nd_Road_Class                                  0
2nd_Road_Number                                 0
Pedestrian_Crossing-Human_Control               0
Pedestrian_Crossing-Physical_Facilities         0
Light_Conditions                                0
Weather_Conditions                              0
Road_Surface_Conditions                         0
Special_Conditions_at_Site                      0
Carriageway_Hazards                             0
Urban_or_Rural_Area                             0
Did_Police_Officer_Attend_Scene_of_Accident     0
LSOA_of_Accident_Location                   11444
Vehicle_Reference                               0
Casualty_Reference                              0
Casualty_Class                                  0
Sex_of_Casualty                                 0
Age_of_Casualty                                 0
```

```
Age_Band_of_Casualty                        0
Casualty_Severity                           0
Pedestrian_Location                         0
Pedestrian_Movement                         0
Car_Passenger                               0
Bus_or_Coach_Passenger                      0
Pedestrian_Road_Maintenance_Worker          0
Casualty_Type                               0
Casualty_Home_Area_type                     0
Casualty_IMD_Decile                         0
dtype: int64
```

Out[10]:

	Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Forc
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	
...	...	...	...	...	...	
164515	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164516	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164517	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164518	2015984141415	314050.0	579638.0	-3.348646	55.103676	9
164519	2015984141415	314050.0	579638.0	-3.348646	55.103676	9

164520 rows × 47 columns