Cairo University
Faculty Computers & Artificial Intelligence
Information Systems Department
Spring 2020

Course: IS422 / Data Warehousing

Delivery Date: Saturday 13 June 2020

Project ID: PID23868003

Instructor: Dr. Hoda Mokhtar

Final Assessment Project

Transportation Data Warehouse

(UK Accidents for 10 years)

Name	ID
Mira Mohamed Abd-Elraheem	20170305
Nehal Akram Ahmed	20170318
Nouran Qassem Mohamed	20170322
Nourhan Ihab El-Khodary	20170324

Table of Contents

List of Illustration

Abstract

Decreasing accidents and decreasing its casualties are the aims of every country all over the world. The United Kingdom has an institute which is responsible for this. We are going to design a data warehouse to help them take decisions and take precautions, using some technologies and tools specified throughout the report.

First, we show in detail every step we have done to built and design the data warehouse, starting with understanding the data set of UK accidents' history for 10 years and setting the main targets of the project to creating the cube of the data warehouse and generating the reports. Then summarizing the steps, specifying what type of queries needed in some steps and their complexities, performance and time taken, and representing some limitations that could appear.

Second, we represent the data model that we are working on in this project, explaining the dimensions, fact table and measures of the data warehouse. Then we connect all what mentioned and done above with the standard data warehouse architecture. Also, we represent the tool we use in generating the dashboard and why we used this tool. Then, we show a sample of the implemented queries that where used in the project.

Finally, we represent a sample of reports that could be generated from the data warehouse in different forms from static reports, charts and dashboards. Each of these reports represent one of the targets mentioned and based on the reports, we conclude the results and give a suggestion.

Chapter 1: Introduction

Description of the project idea

Every country tries to decrease the accidents' numbers and finds what needs to be done to minimize the casualties and catastrophes, one of these countries is the United Kingdom (UK).

So, we build a data warehouse to help in finding out what should be done. It will show what conditions lead to accidents like the lights, roads, weather conditions. Based on the results, authorities should act on what needs or could be fixed. It also provides information on the police and emergency points where should it be added.

Problem Statement

Accidents increase on a daily base caused by environmental conditions or mis using and mis placing the available resources. Also, not applying the right and required actions that helps in decreasing the accidents.

But there are institutes and governments that are trying to find solutions for the current problem such as the National Health Service in the UK. It is a state funded system that guarantees care for all starting from ambulances, emergency to long hospital stays all for free. They have a department concerning the emergency points in the city. They want to prioritize their care (Where to add emergency points) depending on where the most sever, and most destructive accidents happen and when and under certain conditions.

Technology and Tools

To achieve what was mentioned above we need some tools that we used to collect the data, build the data warehouse and analyze the results.

- **SQL Server Database Engine:** to collect the data and apply the ETL phase.
- **Microsoft Visual Studio:** to build the cube of the data warehouse.
- **SQL Server Analysis Service:** to browse the built cube write the MDX queries that are used for the reports.
- **Microsoft Excel:** for generating the static report and chart.
- **Microsoft Power (BI):** for generating the dashboard and dynamic report.

Chapter 2: Requirements and Analysis

Project Life Cycle

When building a data warehouse, four main milestones should be done which are data acquisition, extract transform and load (ETL), building the cube and analyzing the results.

First, we started by collecting the data which is about accidents happened in the UK. The dataset provides detailed road safety data in United Kingdom from 2005 to 2014 about road accidents showing the vehicles with its properties of each vehicle and its conditions during the accidents, the casualties in the accident (severity, personal information, conditions) under certain circumstances such as the weather, time, date, road conditions, location…etc.

The Data set contains 3 main tables: Accidents, Vehicles, and Casualties. Besides, many other tables which are referenced to the 3 tabled showing its options as shown in **Figure 1**.

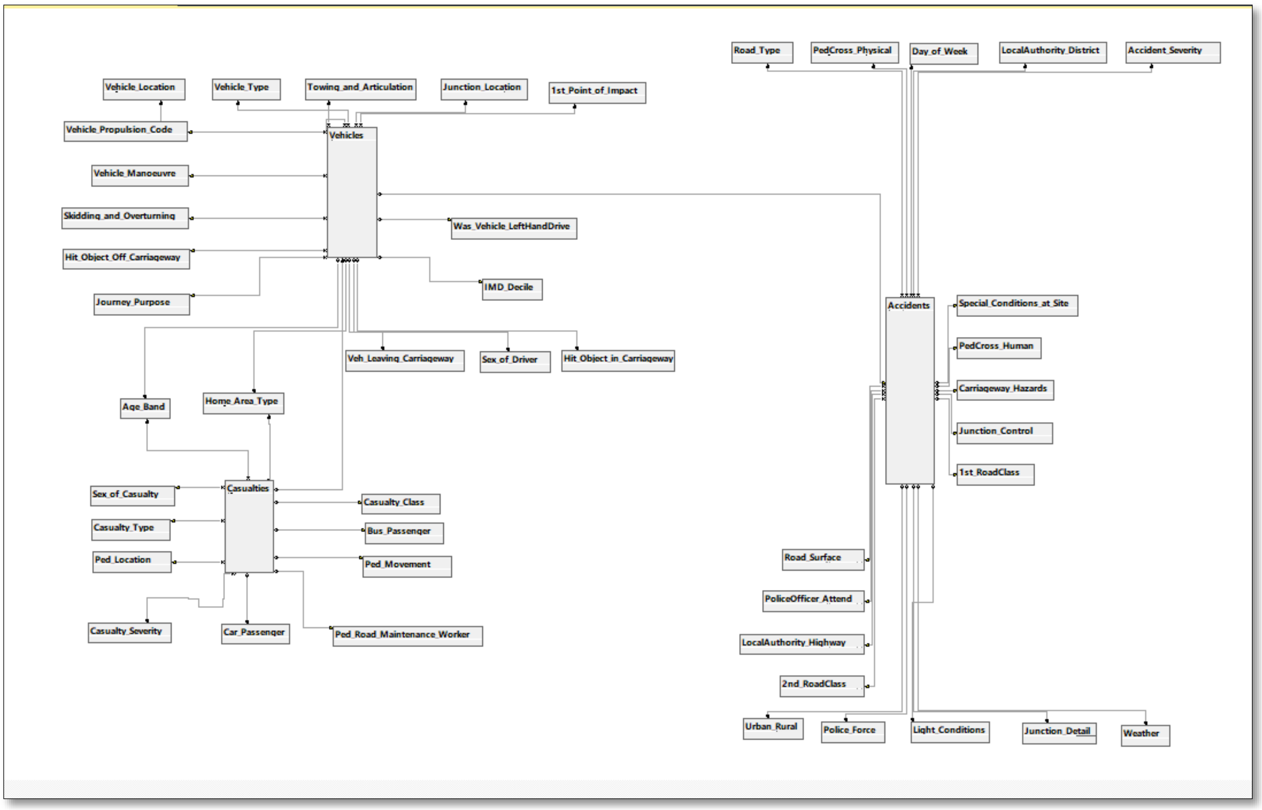

Figure 1: Full ERD

The Full diagram is partitioned in **Figure 2** to be easier to describe later.

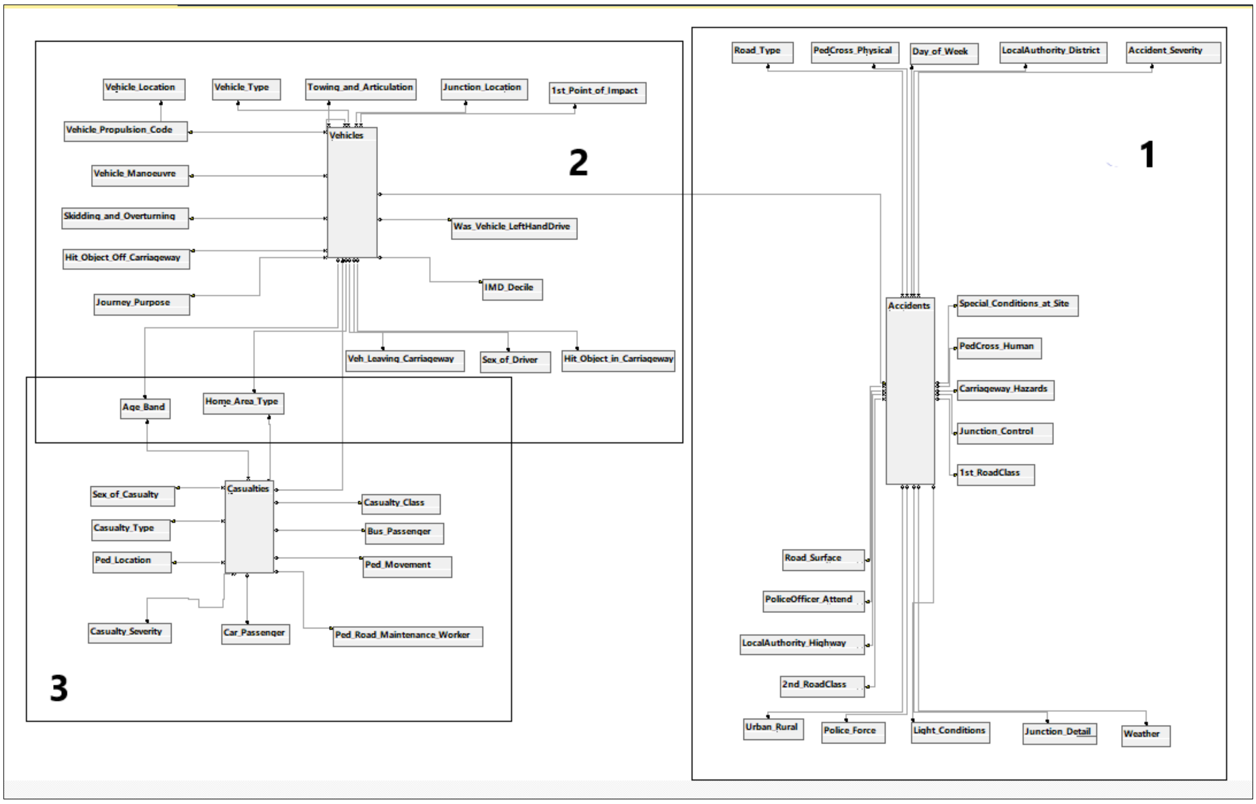

Figure 2: Full ERD Partitioned

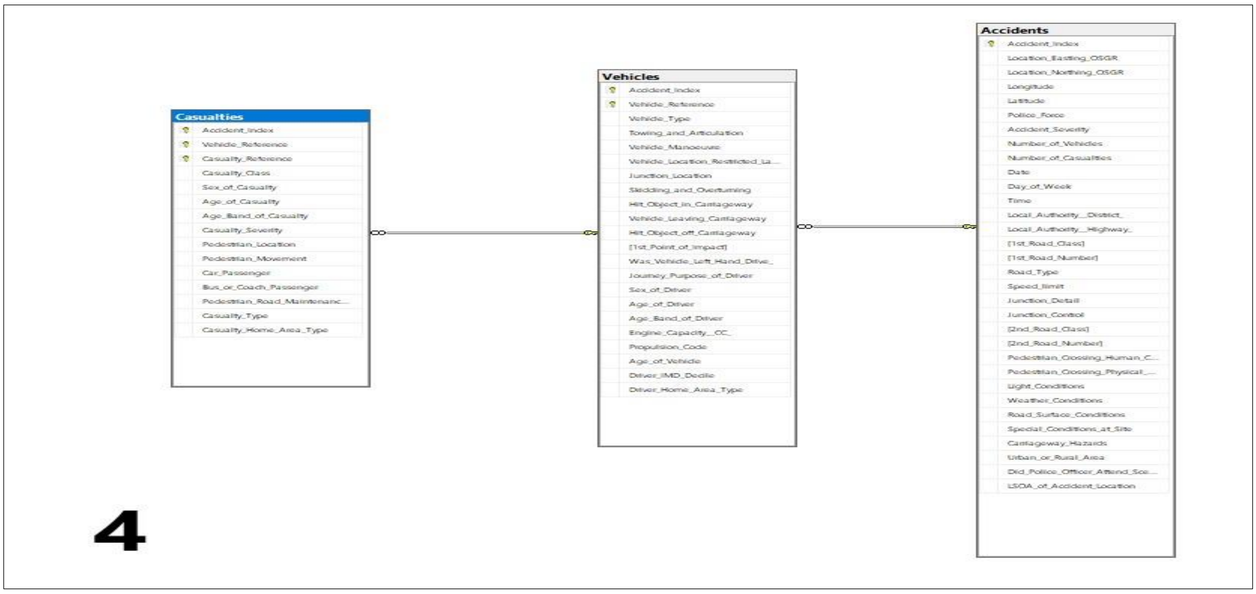

Figure 3: Relation of 3 main tables

Accidents Table:

It provides data about the accident identifying time, date, location, weather, number of casualties and other conditions such as how was the surrounding of the accident in general, also, is there was a police officer and from which police force as shown in **Figure 4.**

It also references on different tables which describe the different conditions. Each table is identified by a code and is described. Accidents' table is identified by Accident_index.

The table contains approximate 1,500,000 record and 32 attributes containing the primary key.

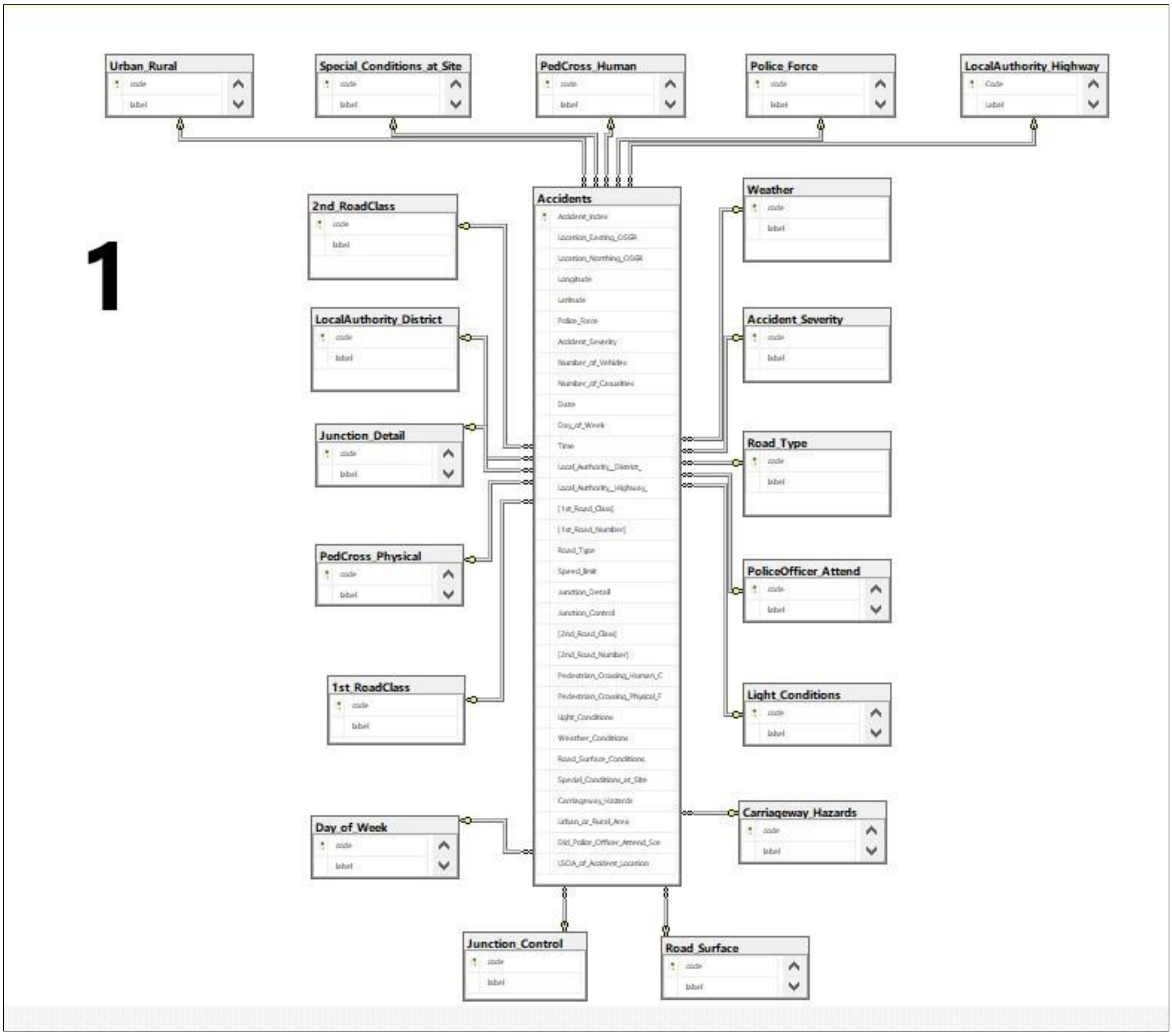

Figure 4: Accidents' ERD

Vehicle’s Table:

It provides data about the vehicles in an accident identifying its properties and condition such as its type, how was it towed, vehicle movement… also some information about the driver and what was the purpose of his journey. As well as, what did it hit and where were it hit as shown in **Figure 5**.

It also references on different tables which describe the different conditions. Each table is identified by a code and is described. Vehicle’s table is identified by a composite key of the Accident_index which is referenced on Accidents table and vehicle reference which is a unique key for each vehicle in the accident. The relation is shown in **Figure 3**.

The table contains approximate 3,000,000 record and 22 attributes containing the composite primary key.

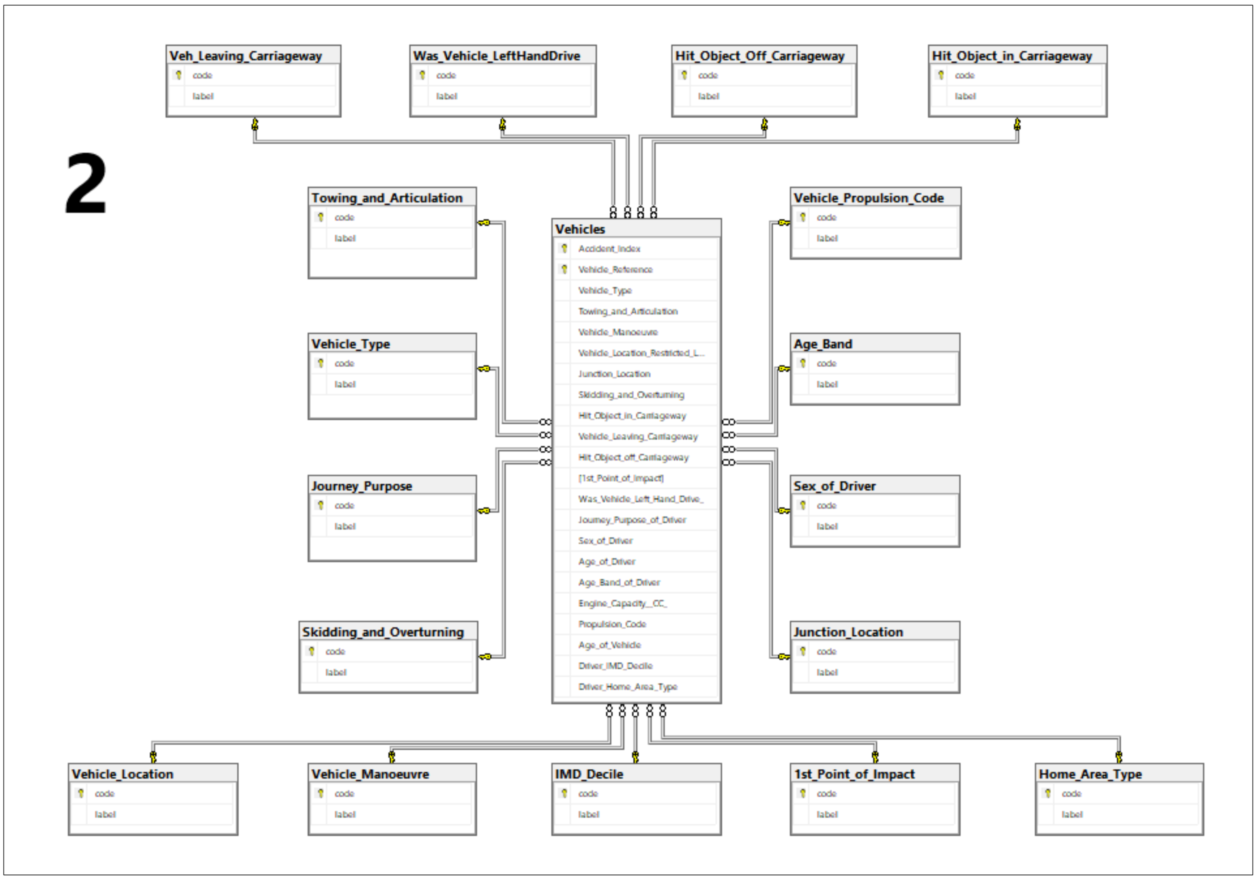

Figure 5: Vehicles’ ERD

Casualties Table:

It provides data about the casualties in an accident identifying some of his personal information such as age, gender… etc, and his severity. As well as, if he was a passenger, driver or a Pedestrian where was he? As shown in **Figure 6**.

It also references on different tables which describe the different conditions. Each table is identified by a code and is described. Casualties' table is identified by a composite key of the composite key which is referenced on Vehicle table and casualty reference which is a unique key for each casualty in the accident and the vehicle which hit or was in. The relation is shown in **Figure 3**.

The table contains approximate 2,500,000 record and 15 attributes containing the composite primary key.

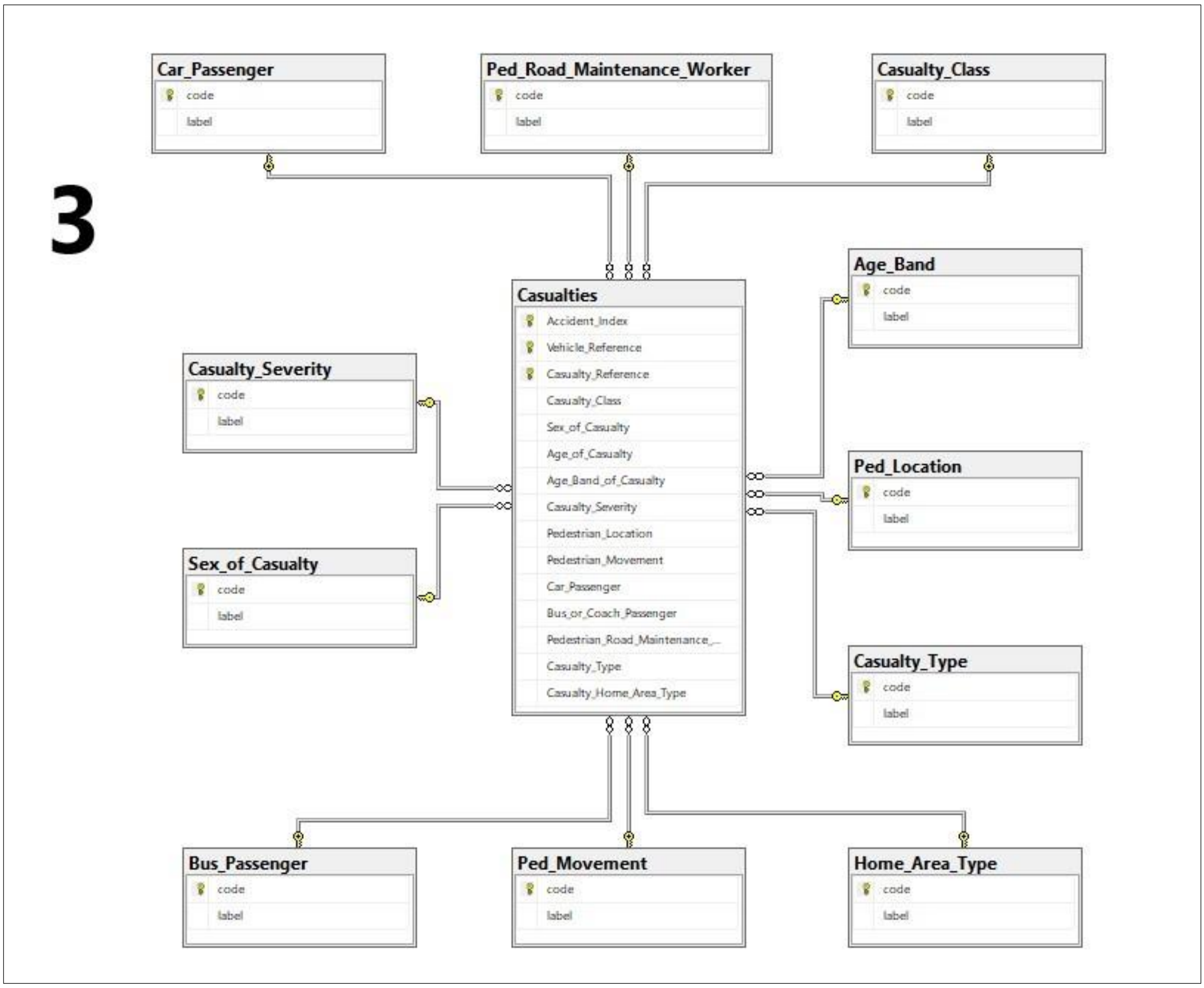

Figure 6: Casualties' ERD

After understanding the dataset we identify our target for the data warehouse and based on we build the data warehouse.

The Main targets:

1- Where should we put emergency points?
    - Measures: measure the number of vehicles, accidents, casualties.
    - Dimensions: Time, Location, accident conditions.

2- Identify which roads needs wider and more lanes and modification.
    - Measures: measure the number of vehicles in accidents.
    - Dimensions: Time, Location, Road, accident conditions.

3- Identify the conditions the accidents happened in that causes most sever accidents.
    - Measures: measures the number of accidents.
    - Dimensions: Time, Location, accident conditions.

Second was the extract, transform and load (ETL) phase, we began by extracting the data from the operational systems by converting the cvs files to relational database in SQL Server. Then we transformed the data to accommodate the requirements and format of the data warehouse. We removed the nulls, change the date time format that all records have the same format. In the end of this phase, we loaded the needed data but first we create data warehouse database by creating the dimensions and fact tables and identifying their granularity. Then, load the transformed data to the data warehouse database using queries as shown in the next chapter.

After loading the data to the data warehouse database, we built the cube using visual studio (VS) following these steps.

First, we have created Analysis Service Multidimensional and Data Mining project as shown in **Figure 7**.

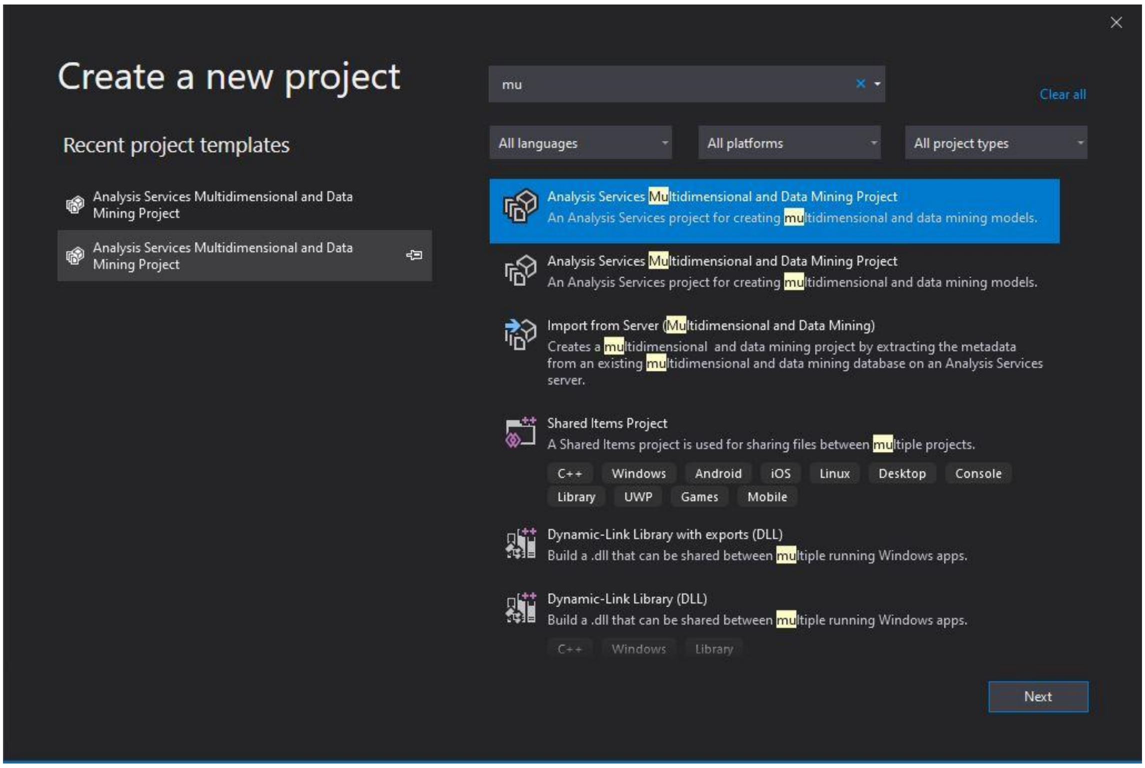

Figure 7: Create New Project in VS

Once the project is created, we create a new data source from the solution explorer that is shown in **Figure 8**.

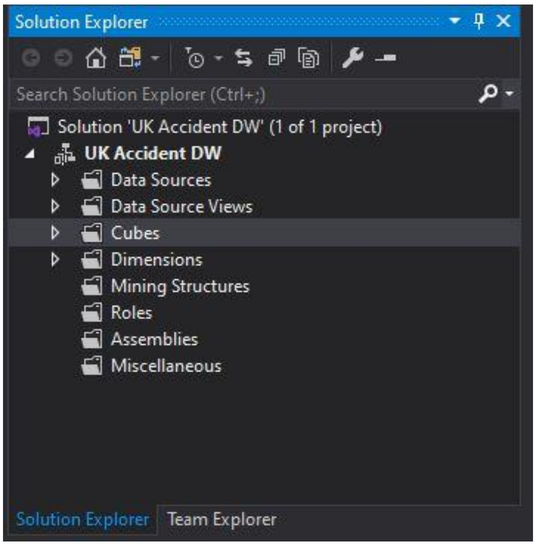

Figure 8: Solution Explorer

From where we stablished a connection with the data source as shown in **Figure 9** which is the data warehouse database that we have created in phase 2 as mentioned above.

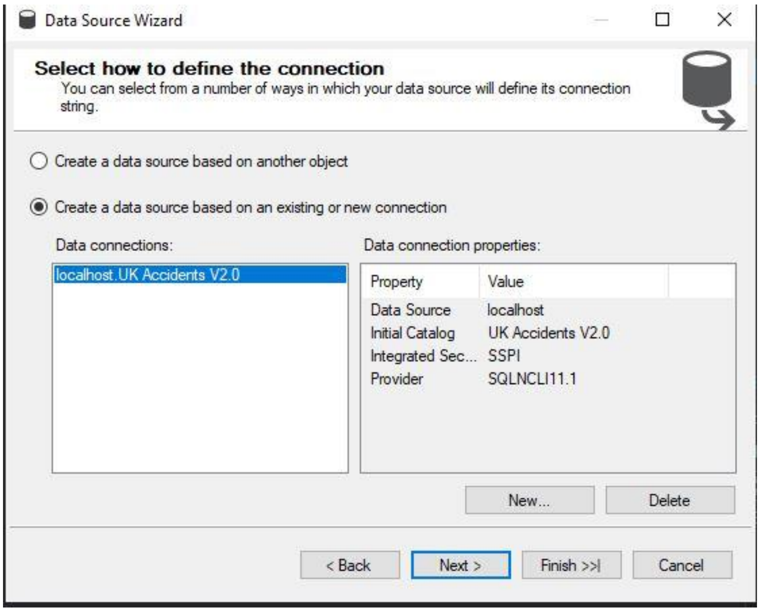

Figure 9: Data Source

Then, we create a new Data Source View from the Solution Explorer **Figure 8**. We add the tables that are going to be used in building the cube as shown in **Figure 10**.

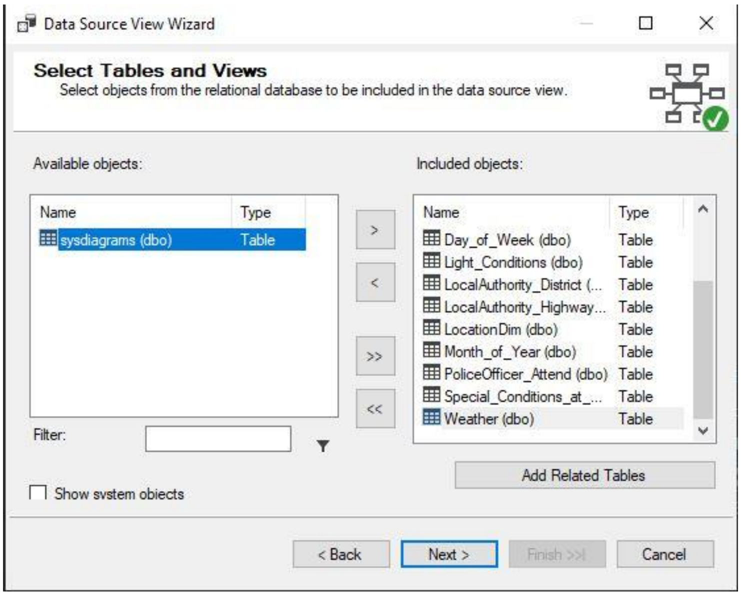

Figure 10: Data Source View

Now that the connection is established and the tables that we are going to use are identified, we start building the cube.

From the Solution Explorer **Figure 8**, create a new cube. Then we identify the fact table and measures.

Then, we select the main dimensions of the data warehouse. Now, we have the main cube as shown in **Figure 11**.

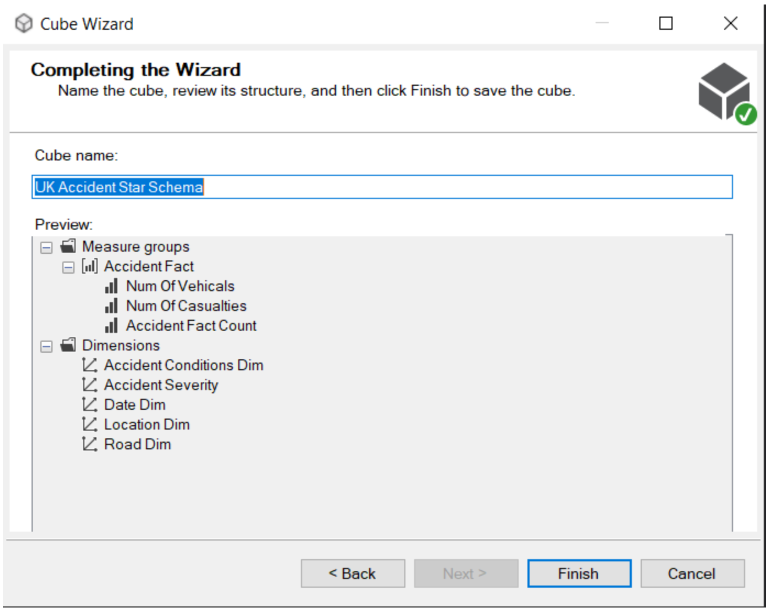

Figure 11: Cube Identification

Now that fact table, measures and all dimensions are identified, we start creating the hierarchy. We have a hierarchy in the date dimension. We select Date dimension and edit as shown in **Figure 12**.

We drag and drop the attributes that will form the hierarchy as shown in **Figure 13**. Then, we organize the relations and levels of the hierarchy from Attribute Relationships as shown in **Figure 14**. Finally, as values are not distinct such as month and day, we change the properties of each to select only the distinct values as from KeyColumns property shown in **Figure 15**.

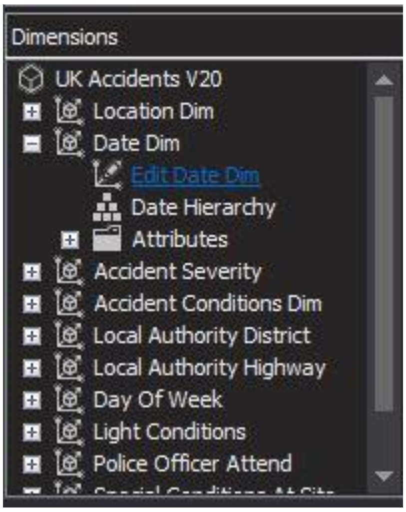

Figure 12: Editing Dimensions

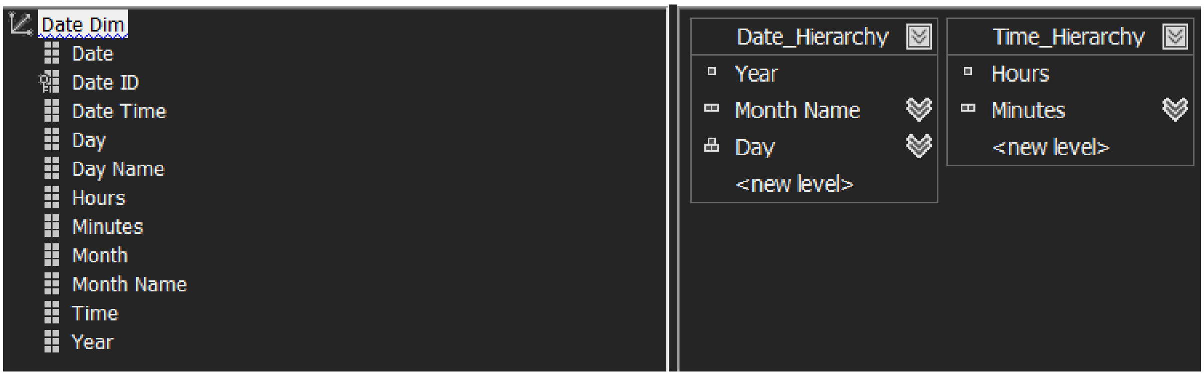

Figure 13: Identifying Hierarchy Attributes

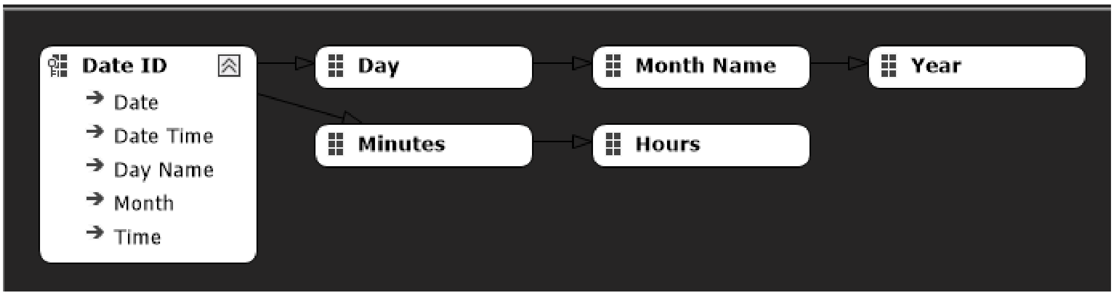

Figure 14: Identifying Hierarchy Structure

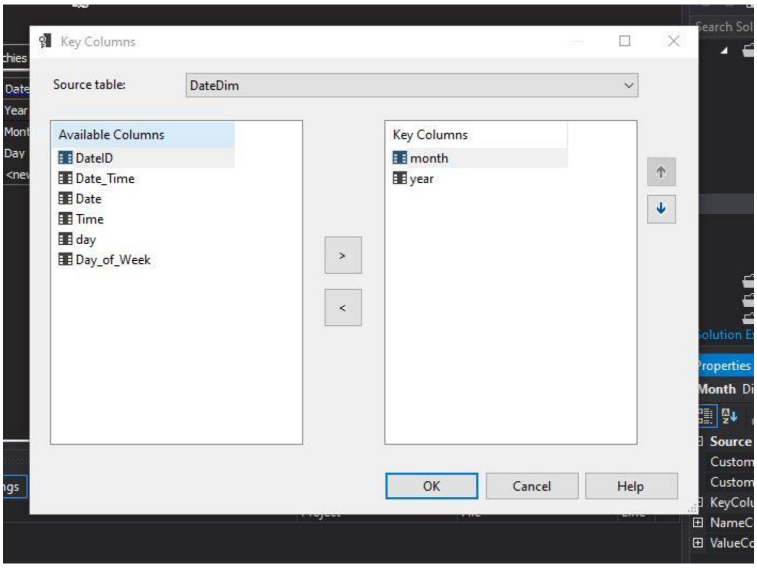

Figure 15: Identifying Hierarchy Keys

The cube is ready to be processed. From Solution Explorer **Figure 8**, process the cube. The connection is confirmed as shown in **Figure 16** and the processing is successful as shown in **figure 17**.

Figure 16: Connection Confirmation

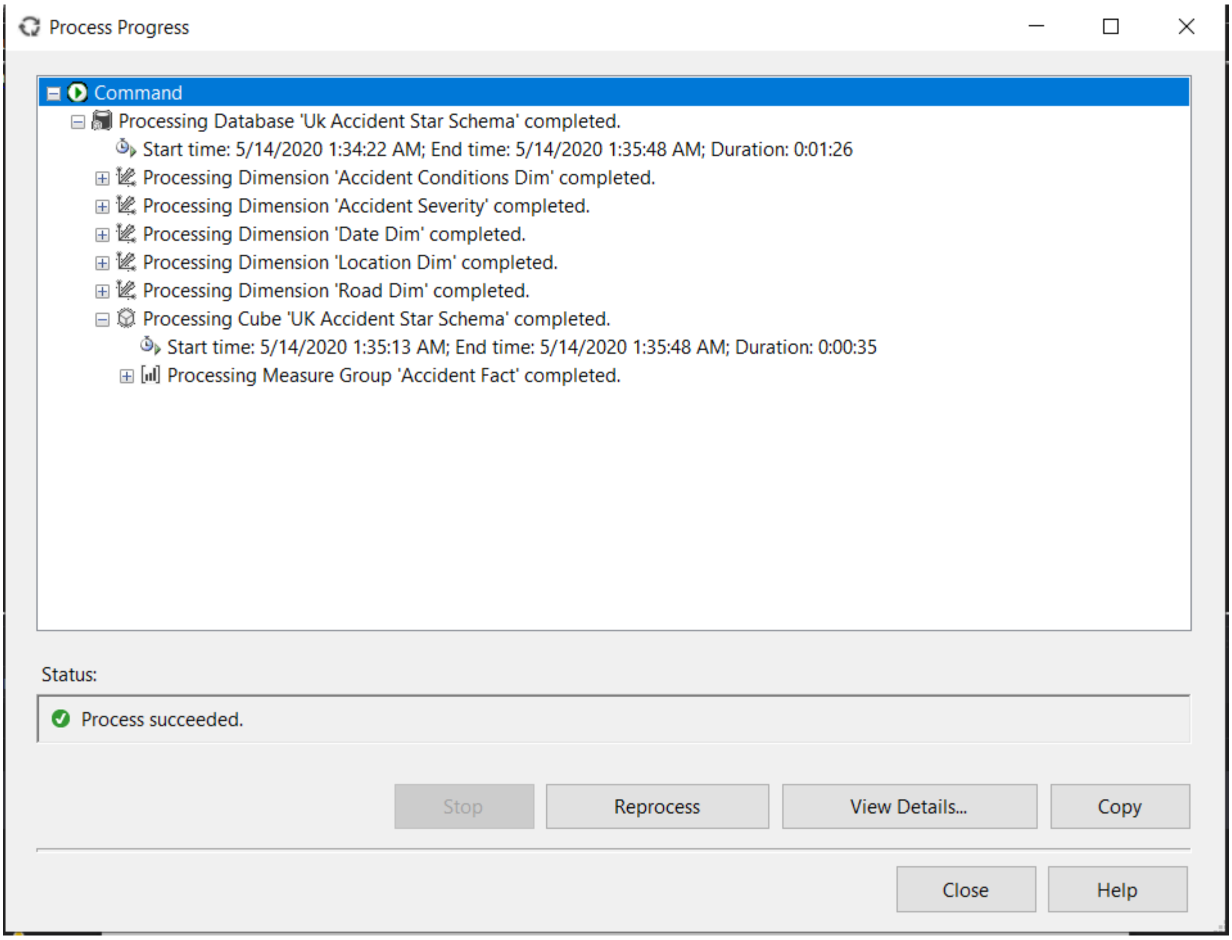

Figure 17: Successful Cube Processing

Finally, after the cube is processed, we write MDX queries to generate reports to be analyzed as shown in Chapter 4. Then, these reports are presented to specialist. Based on these reports, a decision is chosen, and actions will be done.

Project Functionalities and Capabilities

In the previous section, we represented the steps of building a data warehouse from the start in details, to summarize and show the main functions and building blocks.

1. Understanding the dataset.
2. Identifying the targets for the data warehouse project.
3. Extract the needed data that is suitable for the targets.
4. Transform the data: removing the nulls, reformatting the data, adjusting the wrong data … etc.
5. Create the star schema database of the data warehouse.
6. Building the cube: identifying the fact table, dimensions, measures and hierarchies of the cube.
7. Generate reports to analyze and based on this analysis make decisions.

In some of these steps, they are based on writing queries. So, to finish these steps in an efficient way, we must know the types of the queries in each step and how do they perform.

First, we use queries in the ETL phase:

- When transforming the data, we use DML (delete and update) queries. They are not complex, and they don’t take much time if the data size increased.
- To load the data, we must create the data warehouse database to load in it the data. So, we write DDL (create) queries. These queries are simple, does not take time.
- When loading the data, we use DML (select) queries. These queries are very complex because query has inner joins as shown in chapter 3. It takes time to load the data due to its large scale.

Second, we use queries in the Analysis phase:

- To generate reports, we write MDX (select) queries. These queries are not complex as it is a star schema (not normalized). The query performance is high and does not take much time.

Finally, the MDX queries are limited to the granularity and the data in the data warehouse. We have 3 main views, each view could use specific MDX queries, not all queries available for all views.

Chapter 3: Implementation

Data Model

After going through the steps and milestones of the project and knowing what is included in the dataset. This dataset is transformed to a star schema and this is the data model as shown in **Figure 18**.

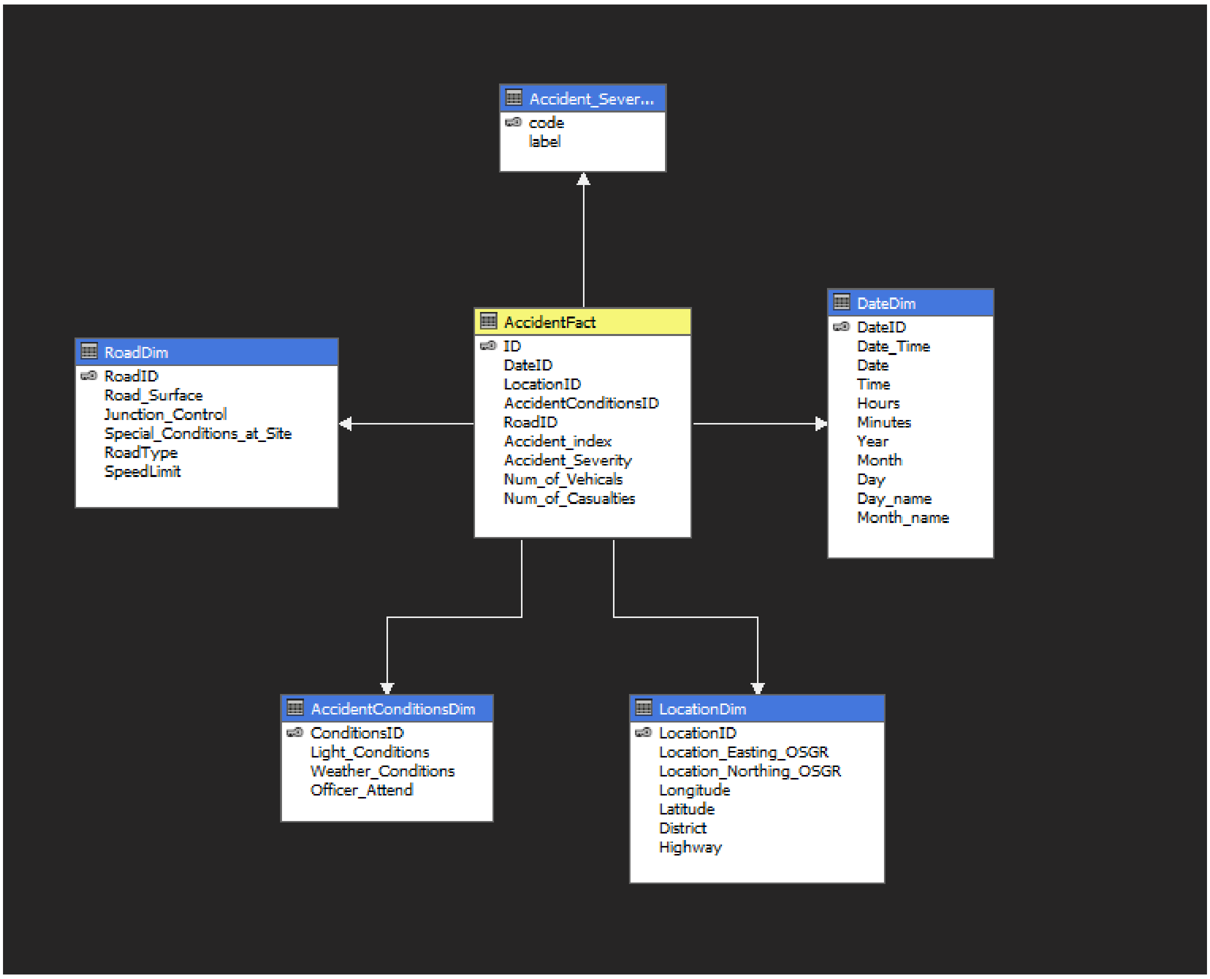

Figure 18: Star Schema Model

The schema contains 4 main dimensions (Date, Location, Accident Conditions, Road), 1 dimension to show the type of accident (Accident Severity Type. We also have 3 measures (Accident Fact Count, Number of vehicles, Number of Casualties).

First the Dimensions:

Date Dimension

This dimension describes the date and time of an accident. It shows the date which includes (year, month, month names, day, day names, time in hours and minutes).

It has a hierarchy for the date. First, it is the date and time. Second, it is divided to date alone and time alone. Finally, the date is further divided by year, month, day and the time hierarchy divided to hours, minutes.

Location Dimension

This dimension describes the location. It describes the location based on the longitude and latitude, location easting and location northing based on OSGR (X-Y Coordinates). Besides the location of the accident in which district and which highway.

Accident Conditions Dimension

This dimension **(Junk Dimension)** describes some conditions that an accident might go through such as (Lighting Conditions, Weather Conditions, did a police officer appear at the scene or is he/she report the accidents?).

Road Dimension

This dimension **(Junk Dimension)** describes how the road was when the accident occurred such as (how is the road surface, junction control, special conditions at site, the road type, speed limit of the road).

Second the fact table and the measures:

Accident Fact Table

The fact table contains foreign keys for the dimension (Location, Date, Accident Conditions, RoadDim). It also contains 1 degenerated dimension (DD) one that is the Accident_index which is the accident key in the case files. Accident_Severity shows the type of severity of the accident it references on Accident Severity table that identifies the severity whether it is (fatal, serious or slight). It also contains 3 measures that we will indulge in detail later in this heading.

Measures:

1. **Number of vehicles:** We count the number of vehicles in each accident from table vehicles with aggregate function **Count**. The equation is similar to:
   If found vehicle with accident_ index X
   Counter ++

2. **Number of Casualties:** We count the number of casualties in each accident from table Casualties with aggregate function **Count**. The equation is similar to:
   If found casualty with accident_ index X
   Counter ++

3. **Accident Count:** We count how many accidents happened under certain conditions.

Project Architecture

Any data warehouse project is preferable to be built using the standard architecture style. It is as shown in **Figure 19** has five building blocks: Operational Systems, Extract Transform and Load (ETL), Data warehouse database, Data Marts and Analysis.

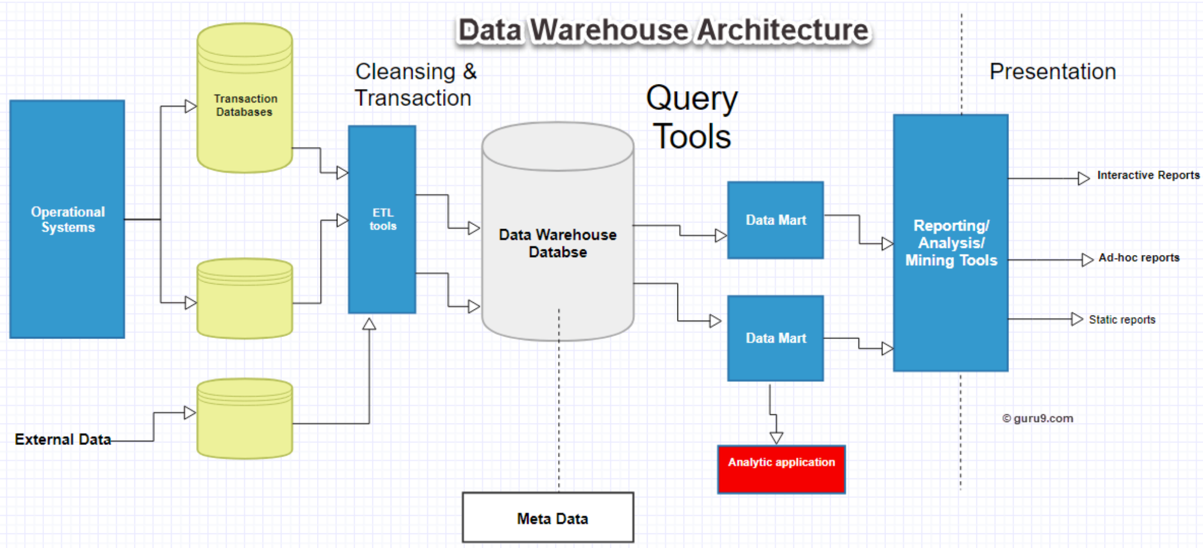

Figure 19: Data Warehouse Architecture

Operational Systems

It is the phase when we collect the data from databases, files, papers entered which is phase 1 in our project. It is application oriented, easily accessed, is entered daily and highly detailed.

Extract, Transform & Load (ETL)

It is the phase that we extract the data from all the operation system transform the data so all the data have the same format, no nulls in the data and all the data are correctly entered then load the transformed data to the data warehouse database which is phase 2 in our project.

Data Warehouse Database

It is when the needed data is added, and the cube is built which is phase 3 in our project. It contains the summarized data of the operational systems. It is subject oriented, all queries available on this data warehouse could be executed but only for the authorized person.

Data Marts

It is the deprived version of the data warehouse. Every department in an enterprise needs certain queries to execute (not all queries that could be applied on the data warehouse). So, data marts provide this by giving access to the only needed and authorized for data.

Analysis

It is when the decision maker executes the available queries for the data mart and generating reports which he will analyze and based on this analysis he will make the appropriate decision.

Visualization Module

When the decision maker requests a report, it is generated on tools that are connected to the data mart to extract the data and represent it in form of charts and reports. There are many tools such as Microsoft Excel, Visual Studio Reporting, Pentaho, Power Designer (BI) and many more.

In our project we are going to use Power BI. It provides all the visuals as shown in **Figure 20** that we need, it is flexible and easy to use. It also provides to generate easily dynamic reports that provide the decision maker options to view the data and charts as shown in **Figures 21 & 22**, it shows that when select a type for a certain KPI it shows the intersection of this type in the other KPIs.

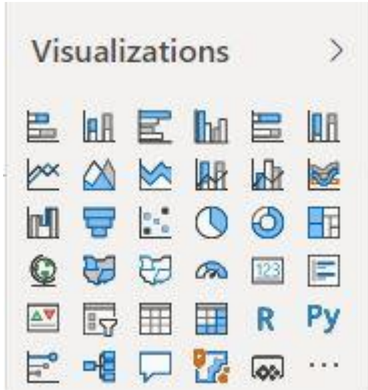

Figure 20: Power (BI) Visuals

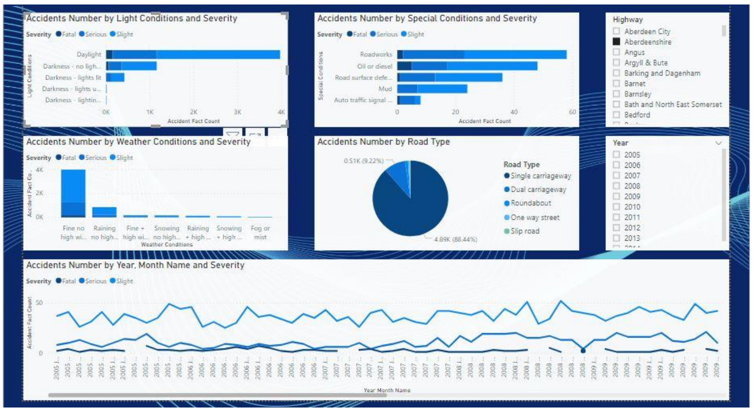

Figure 22: Example 1 for BI

Figure 21: Example 2 for BI

Implementation Code

As mention before in chapter 2, there are two phases in which queries are implemented: the ETL phase (DML and DDL queries) and in the data marts phase (MDX queries).

In the ETL phase:

- During the transformation we removed the nulls present in the location represented in values of the latitude and longitude, and time that included null values as shown:

```
Delete from [dbo].[Accidents] where Time is null


Delete from Accidents where [Location_Easting_OSGR] is NULL or
[Location_Northing_OSGR] is NULL
or [Longitude] is NULL or [Latitude] is NULL
```

- When creating the dimensions and fact tables using the DML as shown (Examples):

```
Create Table AccidentFact
(
	ID                   int not null Primary Key identity(1,1),
	DateID               int not null,
	LocationID           int not null,
	AccidentConditionsID int not null,
	RoadID               int not null,
	Accident_index       nvarchar(50) not null,
	Accident_Severity    int not null,
	Num_of_Vehicals      int not null,
	Num_of_Casualties    int not null
)


create table DateDim
(
	DateID     int not null PRIMARY KEY identity (1,1),
	Date_Time  DATETIME2 (3) NOT NULL,
	Date       DATETIME2(3) NOT NULL,
	Time       DATETIME2(7) NOT NULL,
	Hours      int not null,
	Minutes    int not null,
	Year       int not null,
	Month      int not null,
	Day        int not null,
	Day_name   varchar(10) not null,
	Month_name varchar(15) not null
)
```

- When loading the data to the data warehouse database using DDL queries as shown below examples to the fact table query and the date dimension query:

Fact Table Query:

```
Select Accidents.Accident_Index, Accident_Severity, DateID,
LocationID, ConditionsID, RoadID,
COUNT([Casualty_Reference]) as 'Num_of_Casualties',
COUNT(Vehicles.[Vehicle_Reference]) as 'Num_of_Vehicals'

from Accidents, DateDim, [dbo].[LocationDim] ,
AccidentConditionsDim, [dbo].[Vehicles],
[dbo].[Casualties], [dbo].[RoadDim], [dbo].[Light_Conditions],
[dbo].[Weather], [dbo].[PoliceOfficer_Attend], [dbo].[Road_Surface],
[dbo].[Junction_Control],[dbo].[Road_Type],
[dbo].[Special_Conditions_at_Site]

where
Accidents.Date = DateDim.Date And Accidents.Time = DateDim.Time AND

Accidents.Latitude = [dbo].[LocationDim].Latitude and
Accidents.Longitude = [dbo].[LocationDim].Longitude And

Accidents.Light_Conditions = [dbo].[Light_Conditions].[code] AND
[dbo].[Light_Conditions].[label] =
AccidentConditionsDim.Light_Conditions And
Accidents.[Weather_Conditions] = [dbo].[Weather].[code] AND
[dbo].[Weather].[label] = AccidentConditionsDim.[Weather_Conditions]
AND
Accidents.[Did_Police_Officer_Attend_Scene_of_Accident] =
[dbo].[PoliceOfficer_Attend].[code] and
[dbo].[PoliceOfficer_Attend].[label] =
[dbo].[AccidentConditionsDim].[Officer_Attend] and

Accidents.[Road_Surface_Conditions] = [dbo].[Road_Surface].[code]
and
[dbo].[Road_Surface].[label] = [dbo].[RoadDim].[Road_Surface] and
Accidents.[Junction_Control] = [dbo].[Junction_Control].[code] and
[dbo].[Junction_Control].[label] =
[dbo].[RoadDim].[Junction_Control] and
Accidents.Special_Conditions_at_Site =
[dbo].[Special_Conditions_at_Site].[code] and
[dbo].[Special_Conditions_at_Site].label =
[dbo].[RoadDim].Special_Conditions_at_Site and
Accidents.[Road_Type] = [dbo].[Road_Type].[code] and
[dbo].[Road_Type].label = [dbo].[RoadDim].[RoadType] And
Accidents.[Speed_limit] = [dbo].[RoadDim].[SpeedLimit] AND

Accidents.Accident_Index = Vehicles.Accident_Index And
Accidents.Accident_Index = Casualties.Accident_Index

Group By Accidents.Accident_Index, Accident_Severity, DateID,
LocationID, ConditionsID, RoadID
```

Date Dimension Query:

```
select Distinct Cast(((DATEADD(day, 0, DATEDIFF(day, 0,Date)) +
DATEADD(day, 0 -DATEDIFF(day, 0, cast(Time as datetime)), cast(Time
as datetime)))) as datetime) as 'DateTime',Date, Time, YEAR(Date) as
'year', MONTH(Date) as 'month', DAY(Date) as 'day', DatePart
(HOUR,[Time]) as 'hour' , DatePart ( MINUTE,[Time]) as
'minute',[dbo].[Day_of_Week].[label] as 'day_of_week'
,[dbo].[Month_of_Year].[label] as 'month_name'

from Accidents , [Day_of_Week] , [dbo].[Month_of_Year]

where [dbo].[Month_of_Year].[code] = [dbo].[Accidents].[Month] and
[dbo].[Day_of_Week].[code] = [Day_of_Week]
```

In the data marts phase: the MDX queries for the reports.

- Represents the number of accidents for all districts every year when it was dark and there were no lights.

```
select [Location Dim].[District].Allmembers ON rows,

Crossjoin([Date Dim].[Date Hierarchy].[Year].Allmembers,
[Accident Conditions Dim].[Light Conditions].&[Darkness – no
lighting]) ON columns

FROM [UK Accidents Star Schema]

WHERE [Measures].[Accident Fact Count]
```

- Represents the number of casualties caused by accidents throughout October in 2005 for all special conditions and if there were none it will not show.

```
select [Date Dim].[Date Hierarchy].[Month
Name].&[2005]&[10].Children ON COLUMNS,

Except ([Road Dim].[Special Conditions].Allmembers,
{[Road Dim].[Special Conditions].&[None]}) ON ROWS

FROM [UK Accidents Star Schema]

WHERE ([Measures].[Num Of Casualties])
```

- Represents number of vehicles hit in accidents in 2008 and 2009. Showing which weather condition caused more accidents.

```
select Union({[Date Dim].[Date Hierarchy].[Year].&[2009]},[Date
Dim].[Date Hierarchy].[Year].&[2008]) on columns,
[Accident Conditions Dim].[Weather Conditions].Allmembers On rows

from [UK Accidents Star Schema]

where [Measures].[Num Of Vehicals]
```

- Represents throughout 2010 in each month, how many accidents happened depending on the severity.

```
select DESCENDANTS([Date Dim].[Date Hierarchy].[Year].&[2010], [Date
Dim].[Date Hierarchy].[Month Name], self ) on rows,
[Accident Severity].[Severity].Allmembers on columns

from [UK Accidents Star Schema]

where [Measures].[Accident Fact Count]
```

Chapter 4: Project Output

Report

The report represents the number of casualties in 3 years (2009, 2010, 2011) in the districts and the type of road. To analyze which districts, have the highest number of casualties and which road causes more accidents as shown in **Figure 23**.

Num Of Casualties	Column Labels			
Row Labels	2009	2010	2011	Grand Total
Barnet	2594	2801	2619	8014
Dual carriageway	644	893	700	2237
One way street	6	18		24
Single carriageway	1944	1890	1919	5753
Birmingham	7960	7273	6104	21337
Dual carriageway	1971	1941	1733	5645
One way street	185	165	138	488
Single carriageway	5804	5167	4233	15204
Bradford	4208	3873	3436	11517
Bristol, City of	2513	2347	2181	7041
Cardiff	1945	2047	2022	6014
Cheshire East	3148	3313	3074	9535
Cheshire West and Chester	2216	2572	2607	7395
Cornwall	3157	3790	3577	10524
County Durham	3048	2949	2743	8740
Croydon	2006	1996	2213	6215
Doncaster	2652	2402	2371	7425
East Riding of Yorkshire	1963	2201	2191	6355
Edinburgh, City of	2384	2345	2270	6999
Glasgow City	3114	2863	2754	8731
Kirklees	3280	3000	2658	8938
Lambeth	2261	2253	2336	6850
Leeds	5622	4769	5039	15430
Leicester	2447	2207	2292	6946
Liverpool	3706	3983	3376	11065
Manchester	4270	3625	3570	11465
Northumberland	1931	2119	1959	6009
Nottingham	2099	2125	1998	6222
Sheffield	3293	2988	2750	9031

Figure 23: Report

Conclusion

Most accidents happen in the single carriageway in most of the districts and the highest district is Birmingham. That said it is suggested that we should add another carriageway to decrease the accidents and the priority is for Birmingham.

Chart

This report shows which Highway needs more emergency points. It views the count of Accidents for 3 years (2012, 2013, 2014) in the Highways that have accidents more than 900 and no police officer attended to the scene of accident as shown in **Figure 24**.

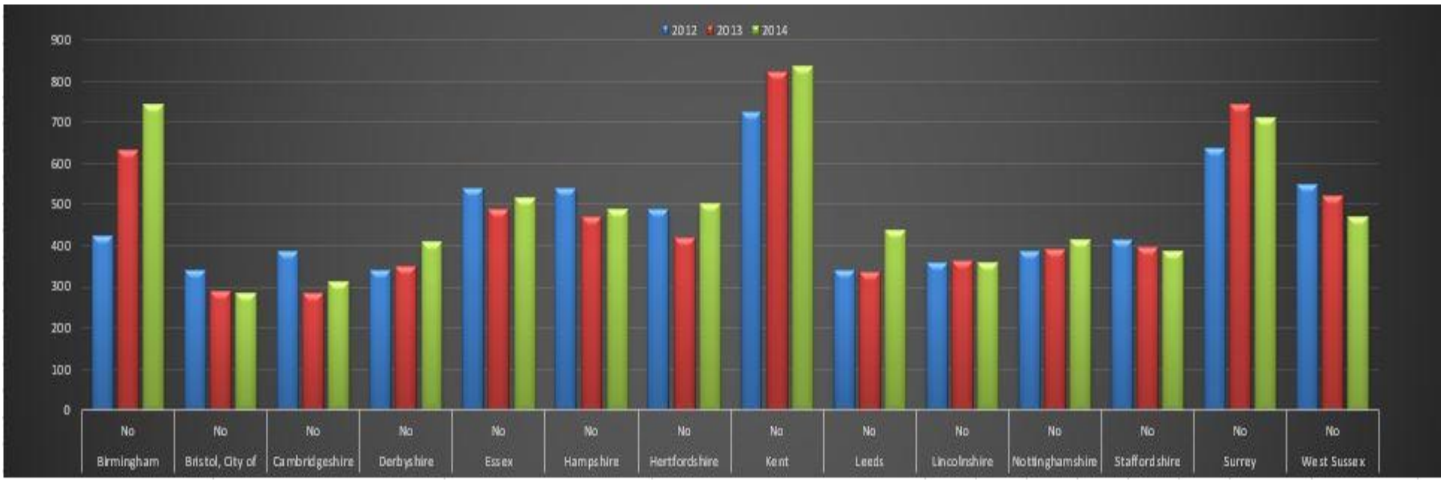

Figure 24: Chart

Conclusion

We conclude from the report that Kent is the highest highway having accidents and no police officer attends. That makes it the priority when constructing emergency points, comes after Surrey and Birmingham.

Dashboard

The dashboard summarizes the all the causes of the accidents and severity by showing trends and representing the KPIs to the government. It shows what caused the increasing of accidents, so they should try to apply safer and take special precautions to decrease the accidents. Also, they could see the performance for all the years and highways or choose a specific year or specific highway or both that gives them flexibility to see the performance specifically.

The dashboard contains on trend, shows 4 KPIs and have two slicers (year, highway) as shown in **Figure 25**.

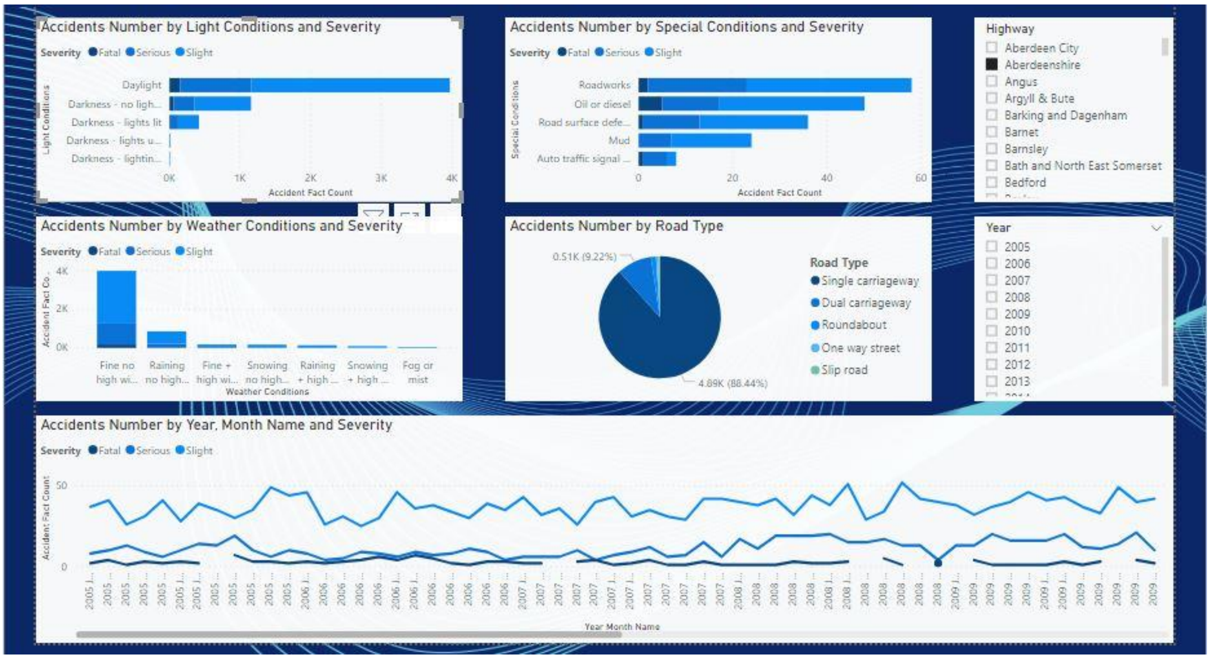

Figure 25: Dashboard

Trends: Shows the number of accidents and severity throughout a range of time as shown in Figure 26.

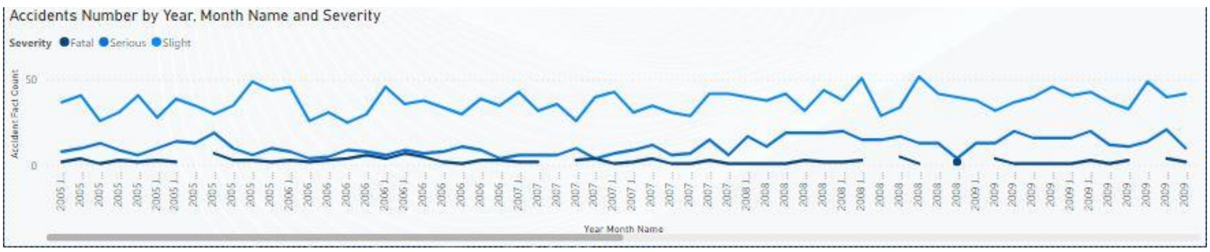

Figure 26: Trend in Dashboard

KPIs:

- Special Conditions with respect to the severity measuring the number of accidents as shown in **Figure 27**.

- Light Conditions with respect to the severity measuring the number of accidents as shown in **Figure 28**.

- Weather Conditions with respect to the severity measuring the number of accidents as shown in **Figure 29**.

- Road type with respect to the severity measuring the number of accidents as shown in **Figure 30**.

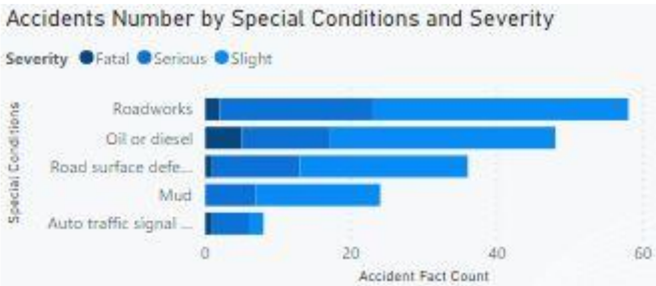

Figure 28: KPI 1

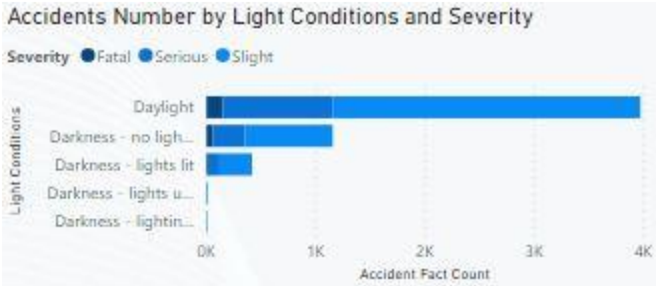

Figure 27: KPI 2

Figure 30: KPI 3

Figure 29: KPI 3

Slicers:

- Year as shown in **Figure 31**.
- Highway as shown in **Figure 32**.

Figure 31: Slicer 1

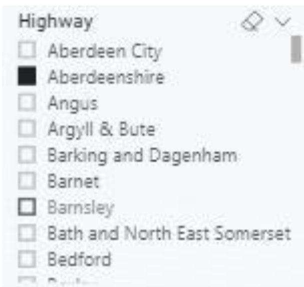

Figure 32: Slicer 2

References

- Dataset
  Ben Fedit (2018). UK Accidents 10 years history.
  https://www.kaggle.com/benoit72/uk-accidents-10-years-history-with-many-variables